

**USO DE TÉCNICAS DE OPTIMIZACIÓN GLOBAL PARA RESOLVER
PROBLEMAS DE INVERSIÓN SÍSMICA: PRIMER GRUPO DE
EXPLORACIÓN**

**EDGAR ANDRÉS TÉLLEZ CORTÉS
JOHAO ALEXANDER CORONEL SILVA
JUAN CAMILO ARIAS PINTO**



**UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERÍAS FÍSICO - MECÁNICAS
ESCUELA DE INGENIERÍAS ELÉCTRICA, ELECTRÓNICA Y
TELECOMUNICACIONES
BUCARAMANGA
2015**

**USO DE TÉCNICAS DE OPTIMIZACIÓN GLOBAL PARA RESOLVER
PROBLEMAS DE INVERSIÓN SÍSMICA: PRIMER GRUPO DE
EXPLORACIÓN**

**EDGAR ANDRÉS TÉLLEZ CORTÉS
JOHAO ALEXANDER CORONEL SILVA
JUAN CAMILO ARIAS PINTO**

**TRABAJO DE GRADO PARA OPTAR AL TÍTULO DE
INGENIERO ELECTRÓNICO**

MODALIDAD: SEMINARIO DE INVESTIGACIÓN

Directores:

ÓSCAR MAURICIO REYES TORRES
Ingeniero Electrónico, Ph.D

SERGIO ALBERTO ABREO CARRILLO
Ingeniero Electrónico, Ph.D(c)



**UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERÍAS FÍSICO - MECÁNICAS
ESCUELA DE INGENIERÍAS ELÉCTRICA, ELECTRÓNICA Y
TELECOMUNICACIONES
BUCARAMANGA
2015**

CONTENIDO

	pág
INTRODUCCIÓN	16
1 DESCRIPCIÓN Y PLANTEAMIENTO DEL PROBLEMA	18
2 MARCO CONCEPTUAL	23
2.0.1 Geofísica	23
2.0.2 Sismología	23
2.0.3 Traza sísmica	23
2.0.4 Modelo del subsuelo	24
2.0.5 Problema directo	24
2.0.6 Problema inverso	25
2.0.7 Exploración sísmica	25
2.0.8 Inversión sísmica	26
2.0.9 Optimización	27
2.0.10 Técnica heurística	27
2.0.11 Técnicas metaheurísticas	27
2.0.12 Métricas de comparación	28
2.0.13 Correlación	28
2.0.14 Correlación cruzada	29
2.0.15 Error absoluto	30
3 MÉTODOS DE OPTIMIZACIÓN GLOBAL	32
3.1 OPTIMIZACIÓN MEDIANTE ENJAMBRE DE PARTÍCULAS (PSO)	33
3.1.1 Analogía con la naturaleza	33
3.1.2 Parámetros del algoritmo	33
3.1.3 Pseudocódigo PSO	34

3.1.4	Implementación del algoritmo	34
3.1.5	Evolución de las partículas	36
3.1.6	Descripción de las funciones multivariable utilizadas	37
3.1.6.1	Función básica	38
3.1.6.2	Función Ackley	38
3.1.6.3	Función Beale	38
3.1.6.4	Función Goldstein Price	39
3.1.6.5	Función Easom	39
3.1.6.6	Función Eggholder	40
3.1.6.7	Función Holder Table	40
3.1.6.8	Función Esfera	41
3.1.6.9	Función Step	41
3.1.6.10	Función Rastrigin	42
3.1.6.11	Función Schwefel	42
3.1.6.12	Función Rosenbrock	43
3.1.6.13	Función Stochastic	43
3.1.7	Implementación sobre funciones multivariadas	44
3.1.8	Análisis de Resultados	45
3.1.9	Ventajas del algoritmo	57
3.1.10	Desventajas del algoritmo	58
3.1.11	Discusión	58
3.2	OPTIMIZACIÓN MEDIANTE ALGORITMOS GENÉTICOS (GA)	59
3.2.1	Analogía con la naturaleza	59
3.2.2	Pseudocódigo GA	61
3.2.3	Implementación del algoritmo	61
3.2.4	Evolución de individuos	61
3.2.5	Implementación sobre funciones multivariable	63
3.2.6	Análisis de resultados	64
3.2.7	Ventajas del algoritmo	70

3.2.8	Desventajas del algoritmo	70
3.2.9	Discusión	71
4	INTEGRACIÓN DE ALGORITMOS DE OPTIMIZACIÓN EN EL PROCESO DE INVERSIÓN SÍSMICA	73
4.1	SEISMIC UNIX (SU)	73
4.2	COMUNICACIÓN ENTRE SU Y MATLAB	75
4.3	PSO EN INVERSIÓN SÍSMICA	76
4.3.1	Implementación	78
4.3.2	Análisis de resultados	79
4.4	GA EN INVERSIÓN SÍSMICA	82
4.4.1	Implementación	82
4.4.2	Análisis de resultados	84
4.5	DISCUSIÓN	85
5	CONCLUSIONES	87
6	RECOMENDACIONES	90
	CITAS BIBLIOGRÁFICAS	92
	BIBLIOGRAFÍA	96
	ANEXOS	98

LISTA DE FIGURAS

	pág
Figura 1 Esquema desarrollo seminario	19
Figura 2 Modelo de velocidades para varias capas del subsuelo	20
Figura 3 Ejemplo adquisición sísmica terrestre	21
Figura 4 Trazas sísmicas obtenidas	21
Figura 1 Traza sísmica generada mediante el software SU	23
Figura 2 Modelo del subsuelo	24
Figura 3 Esquema modelado problema directo	25
Figura 4 Problema inverso	25
Figura 5 Proceso exploración sísmica	26
Figura 6 Ubicación aproximada del mínimo global para la función Ackley .	27
Figura 7 Ejemplo correlación	29
Figura 8 Ejemplo correlación cruzada	30
Figura 9 Ejemplo error absoluto	31
Figura 10 Comportamiento de una partícula	36
Figura 11 Evolución de las partículas	37
Figura 12 Función Básica	38
Figura 13 Función Ackley	39
Figura 14 Función Beale	39
Figura 15 Función Goldstein Price	40
Figura 16 Función Easom	40
Figura 17 Función Eggholder	41

Figura 18 Función Holder Table	41
Figura 19 Función Esfera	42
Figura 20 Función Step	42
Figura 21 Función Rastrigin	43
Figura 22 Función Schwefel	43
Figura 23 Función Rosenbrock	44
Figura 24 Función Stochastic	44
Figura 25 Posiciones iniciales y finales del primer experimento con la función básica	46
Figura 26 Posiciones iniciales y finales del segundo experimento con la función básica	47
Figura 27 Posiciones finales con 10, 20 y 30 partículas en la función básica	48
Figura 28 Movimiento oscilatorio alrededor del mínimo global	51
Figura 29 Dinámica del enjambre para $w = 0,1$	52
Figura 30 Dinámica del enjambre para $w = 0,5$	52
Figura 31 Dinámica del enjambre para $w = 0,9$	52
Figura 32 Agrupación del enjambre para $c1 = 2,1$ y $c2 = 1,0$	54
Figura 33 Agrupación del enjambre para $c1 = 1,0$ y $c2 = 2,1$	55
Figura 34 Agrupación del enjambre para $c1 = 1,8$ y $c2 = 2,1$	55
Figura 35 Comportamiento de los individuos	61
Figura 36 Modelos creados	74
Figura 37 Trazas sísmicas adquiridas	75
Figura 38 Operaciones que se realizan para las métricas de correlación y correlación cruzada	78
Figura 39 Ilustración de la estrategia implementada	80
Figura 40 Trazas generadas en las tres primeras ejecuciones del algoritmo PSO	82
Figura 41 Tendencia del error medio cuadrático para las cuatro ejecuciones del algoritmo PSO	83
Figura 42 Trazas generadas en cada ejecución del algoritmo GA	84

Figura 43 Tendencia del error medio cuadrático para las cuatro ejecuciones
del algoritmo GA 85

LISTA DE TABLAS

	pág
Tabla 1 Resultados para la función básica	45
Tabla 2 Posición de la mejor partícula en 10 ejecuciones en la función básica	48
Tabla 3 Resultados para la función Ackley con $w = 0,1$	53
Tabla 4 Resultados para la función Ackley con $w = 0,3$	53
Tabla 5 Resultados para la función Ackley con $w = 0,5$	54
Tabla 6 Resultados para la función Ackley con $c1 = 1,9$ y $c2 = 1,8$	55
Tabla 7 Resultados para la función Ackleys con $c1 = 1,8$ y $c2 = 1,9$	56
Tabla 8 Parámetros bajo los cuales se obtuvieron los mejores resultados en la implementación de PSO	57
Tabla 9 Resultados para la función Beale	63
Tabla 10 Resultados para la función Ackley con $ni=10$	66
Tabla 11 Resultados para la función Ackley con $ni=20$	66
Tabla 12 Resultados para la función Ackley con $ni=30$	67
Tabla 13 Resultados para la función Básica con $ni=10$	67
Tabla 14 Resultados para la función Básica con $ni=20$	67
Tabla 15 Resultados para la función Básica con $ni=30$	68
Tabla 16 Parámetros bajo los cuales se obtuvieron los mejores resultados en la implementación de GA	68
Tabla 17 Resultados para la función Eggholder con $Tc=1$	68
Tabla 18 Resultados para la función Eggholder con $Tc=0,7$	69
Tabla 19 Resultados para la función Eggholder con $Tc=0,5$	69
Tabla 20 Resultados para la función Eggholder con $Tc=0,3$	69

Tabla 21 <i>Sloths</i> encontrados por el algoritmo PSO en inversión sísmica . . .	82
Tabla 22 <i>Sloths</i> encontrados por el algoritmo GA en inversión sísmica . . .	84

LISTA DE ANEXOS

	pág
ANEXO A.1 Descripción del código PSO	95
ANEXO A.2 Descripción del código GA	99
ANEXO A.3 Memorias del seminario de investigación	102

TÍTULO

USO DE TÉCNICAS DE OPTIMIZACIÓN GLOBAL PARA RESOLVER PROBLEMAS DE INVERSIÓN SÍSMICA *.

AUTORES:

EDGAR ANDRÉS TÉLLEZ CORTÉS**
JOHAO ALEXANDER CORONEL SILVA**
JUAN CAMILO ARIAS PINTO**

PALABRAS CLAVE:

Algoritmos genéticos, Inversión sísmica, Metaheurística, Optimización, Optimización mediante Enjambre de partículas.

RESUMEN:

Este documento presenta el desarrollo del trabajo de grado denominado *Uso de técnicas de optimización global para resolver problemas de inversión sísmica*, bajo la metodología de seminario de investigación. El objetivo principal de este trabajo consiste en el estudio de técnicas metaheurísticas aplicadas en problemas de inversión sísmica, así como su implementación y análisis.

Como primer paso hacia la comprensión del problema propuesto, se hace una revisión de conceptos básicos en Geofísica, inversión sísmica, optimización y técnicas metaheurísticas, así como la selección de un modelo matemático del proceso de inversión sísmica. Seguidamente, se presenta la discusión acerca de dos técnicas metaheurísticas conocidas como: *Optimización mediante Enjambre de Partículas y Algoritmos Genéticos*, las cuales fueron implementadas y puestas a prueba sobre funciones matemáticas multivariable. Estas técnicas son aplicadas posteriormente en el proceso de inversión sísmica, utilizando diferentes tipos de métricas de comparación. Los resultados más relevantes de la integración de las técnicas metaheurísticas al problema propuesto, fueron registradas mediante tablas, gráficas y bases de datos digitales, así como las actas del seminario que resumen todo lo que se hizo durante cada sesión.

Finalmente se presentan análisis, discusiones, recomendaciones y conclusiones acerca del uso de las técnicas metaheurísticas en problemas de inversión sísmica.

* Trabajo de Grado.

** Facultad de Ingenierías Físico-Mecánicas. Escuela de Ingenierías Eléctrica, Electrónica y de Telecomunicaciones. Directores Dr.Ing. Oscar Mauricio Reyes Torres y Ph.D(c) Sergio Alberto Abreo Carrillo.

TITLE

**USING GLOBAL OPTIMIZATION TECHNIQUES TO SOLVE
SEISMIC INVERSION PROBLEMS *.**

AUTHORS:

EDGAR ANDRÉS TÉLLEZ CORTÉS**
JOHAO ALEXANDER CORONEL SILVA**
JUAN CAMILO ARIAS PINTO**

KEYWORDS:

Genetic Algorithms, Metaheuristics, Optimization, Seismic Inversion, Particle Swarm Optimization.

ABSTRACT:

This document shows the development of the degree work entitled *Using Global Optimization Techniques to solve Seismic Inversion Problems*, as a research seminar. The main objective of this work is to study metaheuristic techniques in the context of seismic inversion problems, and its implementation and analysis.

As a first step toward understanding the proposed problem, a review of basic concepts in geophysics, seismic inversion, optimization and metaheuristic techniques is showed, as well as the selection of a mathematical model of the seismic inversion process. Then, it is presented the discussion about two metaheuristic techniques: Particle Swarm Optimization and Genetic Algorithms, which were implemented and tested on multivariable mathematical functions. These techniques are later applied in the process of seismic inversion, using different kind of performance measurements. The main results of the integration of the metaheuristic techniques to the proposed problem were stored in tables, graphs and digital databases, as well as the proceedings of the seminar which summarize all what was done in every sessions.

Finally, it is presented the analysis, discussion, recommendations and conclusions about the use of metaheuristic techniques in seismic inversion problems.

* Degree Work.

** Facultad de Ingenierías Físico-Mecánicas. Escuela de Ingenierías Eléctrica, Electrónica y de Telecomunicaciones. Directed by Dr.Ing Oscar Mauricio Reyes Torres and Dr.Ing Sergio Alberto Abreo Carrillo.

INTRODUCCIÓN

El contenido de este documento presenta el desarrollo del trabajo de grado denominado *Uso de técnicas de Optimización Global para resolver problemas de inversión sísmica*, en la modalidad de seminario de investigación. Este seminario se propuso con el objetivo de estudiar, analizar e implementar algoritmos de optimización global basados en técnicas metaheurísticas, para aplicarlos a problemas de inversión sísmica. Permitiendo la obtención de los parámetros de un modelo matemático de referencia (modelo de velocidades de una sección del subsuelo), a partir del ajuste de datos modelados con datos observados.

El documento consta de cuatro capítulos que reflejan el desarrollo del seminario. El primer capítulo expone una revisión de conceptos básicos en geofísica, inversión sísmica, optimización y técnicas metaheurísticas, donde se hace referencia a las principales fuentes bibliográficas que se estudiaron para el desarrollo del seminario. Posteriormente se hace el planteamiento de la problemática estudiada, con el fin de aclarar el contexto bajo el cual se desarrolla este trabajo.

En el segundo capítulo se presenta el estudio y describe la implementación de dos técnicas metaheurísticas conocidas como: *Optimización mediante Enjambre de Partículas (Particle Swarm Optimization, PSO)* y *Algoritmos Genéticos (Genetic Algorithms, GA)*. El estudio se centra en aspectos como: la analogía que tienen estas técnicas con la naturaleza, la configuración de sus parámetros y su respectivo pseudocódigo. La implementación se describe a partir de la puesta a prueba de estas técnicas sobre funciones matemáticas multivariable. Finalmente se hace una discusión referente al análisis de desempeño de cada técnica.

El tercer capítulo muestra la integración de dos técnicas metaheurísticas estudiadas con el proceso de inversión sísmica. Se presenta el uso de herramientas de software como Seismic Unix y Matlab, para realizar la integración mencionada. Al final del capítulo se muestra cómo se adquirieron los datos modelados y se definen las métricas de comparación utilizadas.

El cuarto capítulo comprende las conclusiones y recomendaciones relacionadas con el ajuste adecuado de parámetros y la implementación de estrategias, para la aplicación de técnicas metaheurísticas en inversión sísmica.

Finalmente se anexan los apéndices que describen los conceptos básicos necesarios para la comprensión de la problemática estudiada y los códigos implementados a lo largo del seminario de investigación. Además se crearon documentos complementarios que

permiten ver en detalle toda la información generada durante el seminario de investigación, los cuales se encuentran disponibles en la página del grupo de investigación CPS (Conectividad y Procesamiento de Señales) de la Universidad Industrial de Santander. Los documentos complementarios están conformados por: *Actas* semanales que explican brevemente los temas tratados, discusiones generadas, compromisos para sesiones posteriores, asistentes y asignación de roles como expositores y relator; *memorias* que explican a fondo temas tratados, discusiones, análisis a partir de las implementaciones y avances que tuvieron lugar en cada una de las sesiones semanales; y *anexos* que contienen archivos correspondientes a las implementaciones de las técnicas metaheurísticas en inversión, así como tablas que corresponden al registro de datos provenientes de la ejecución de los algoritmos sobre las distintas funciones de prueba. Estos documentos complementarios constituyen un conjunto de referencias para la realización de trabajos futuros en esta área.

1. DESCRIPCIÓN Y PLANTEAMIENTO DEL PROBLEMA

Este capítulo contiene los elementos mínimos necesarios (resultado de una revisión bibliográfica de conceptos básicos) que permiten entender el problema discutido en el seminario de investigación. La primera parte de esta revisión se hace de manera secuencial donde se presentan las definiciones de las ciencias correspondientes al tema de estudio, hasta abordar el problema de interés conocido como inversión sísmica. La segunda y tercera parte de esta revisión se hace de manera general, presentando la definición y el objetivo que tiene la optimización y las métricas de comparación. Finalmente se presenta el planteamiento del problema para comprender de manera general el trabajo que se realizó durante el seminario de investigación.

El problema planteado en el seminario de investigación, corresponde al proceso conocido como inversión sísmica. Este problema se aborda usando dos técnicas metaheurísticas para la obtención de los parámetros del modelo matemático, definido en la ecuación (1), según Caicedo en [3]. Las métricas de comparación usadas en la ecuación de búsqueda de las metaheurísticas corresponden a correlación, correlación cruzada y error absoluto.

El diagrama mostrado en la figura 1 muestra de manera general el desarrollo realizado en el seminario de investigación, la explicación de este diagrama es: El modelo matemático representado por

$$(\partial_x \Theta)^2 + (\partial_y \Theta)^2 + (\partial_z \Theta)^2 = \frac{1}{u^2}, \quad (1)$$

donde:

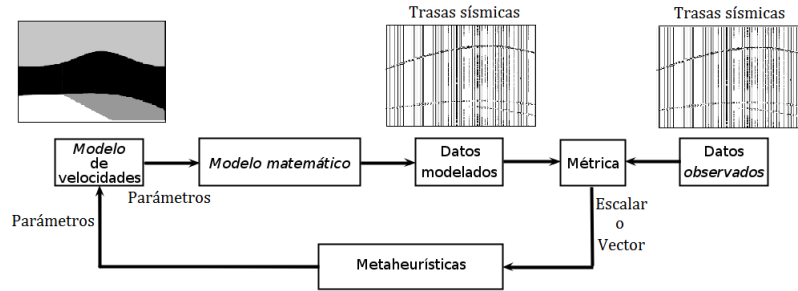
Θ : Representa los frentes de onda.

u : Representa la velocidad de propagación de los frentes de onda en cada capa.

corresponde a la aproximación lineal de la ecuación Eikonal (1) para altas frecuencias. Con este modelo matemático, se realizó la simulación del proceso conocido como adquisición sísmica por medio del software SU, que permite ejecutar de distintas maneras la adquisición, como por ejemplo: ecuación de onda, trazado de rayos, entre otros. Finalmente, de acuerdo con las discusiones del seminario, se decidió trabajar con la herramienta *triseis*, debido a su bajo costo computacional y a la facilidad de utilización de la misma. Esta herramienta simula una adquisición sísmica con trazado de rayos a partir de la ecuación (2) definida por:

$$\textit{triseis} < \textit{modelo} > \textit{sismograma}(xs - zs - xg - zg)[\textit{parámetros} - \textit{opcionales}] \quad (2)$$

Figura 1: Esquema general del desarrollo del seminario de investigación.



donde:

- $xs = 3$ [km] (coordenada x de la fuente).
- $zs = 0$ [km] (profundidad z de la fuente).
- $xg = -2950$ [m] a 2950 [m] (coordenada x de los receptores).
- $zg = 0$ [km] (profundidad de los geófonos).

Parámetros opcionales:

- Geometría del modelo = Fija
- Dimensiones del modelo = (10×7) [km] (largo \times profundidad)
- Número de capas del modelo = 4
- Número de geófonos = 60
- Espacio entre geófonos = 100[m]
- Número de fuentes = 1
- Número de disparos = 1
- Número de rayos que emana la fuente = 201
- Rango que emana la fuente = 65° de lado a lado
- Técnica utilizada = Trazado de rayos
- Muestras tomadas = 2,000
- Tiempo de muestreo = 0,0073[s]

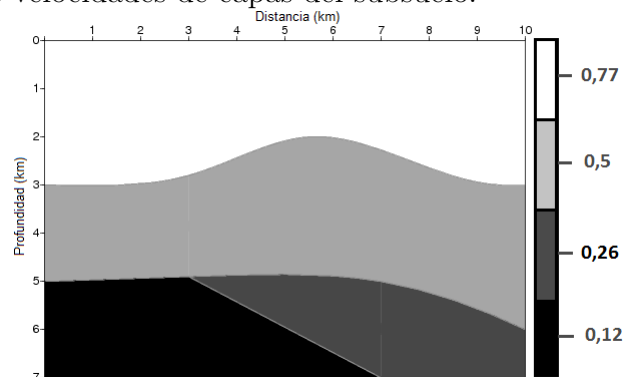
El conjunto de trazas producto de la adquisición sísmica, corresponde a los datos modelados. A partir de estos datos y de los datos observados (pertenecientes a una adquisición simulada sobre un modelo de velocidades conocido, denominado modelo de referencia) se hace una comparación por medio de las métricas que arrojan, dependiendo del tipo, un escalar o un vector como indicador de la diferencia entre las trazas. El indicador se entrega a las metaheurísticas, que lo usan para ajustar la búsqueda de un conjunto de parámetros que más se aproxime a los parámetros del modelo de referencia. Los parámetros entregados conforman las lentitudes (inverso al cuadrado de la velocidad de cada capa) del modelo de velocidades, que al aplicar el proceso de adquisición sísmica, se convierten en los nuevos datos modelados (denominado sintéticos). Finalmente, se espera que el indicador entregado por las métricas sea cercano a cero para el

caso del error absoluto y cercano a 1 para el caso de correlación y correlación cruzada como evidencia de una alta similitud entre los datos modelados y los datos observados.

La selección de las métricas de comparación se hizo conforme a las discusiones presentadas a lo largo del seminario de investigación. Las métricas elegidas son correlación, correlación cruzada y error absoluto. Para ello se tuvo en cuenta que cada una de ellas proporcionan comparaciones, punto a punto a través de un indicador conocido como coeficiente; comparaciones teniendo en cuenta todo el conjunto de datos y el desfase de los mismos; y comparaciones punto a punto a partir de la diferencia entre datos, respectivamente. Las técnicas de optimización global, basadas en métodos metaheurísticos, escogidas fueron: *optimización mediante enjambre de partículas (PSO)* y *algoritmos genéticos (GA)*. La elección de éstas técnicas se fundamentó en una revisión teórica del estado del arte, acerca del uso de los métodos directos, heurísticos y metaheurísticos en problemas de inversión sísmica.

Los modelos de velocidades que se tomaron como referencia, corresponden a modelos de cuatro capas con geometría fija, y con valores constantes de velocidades en la región de las capas del subsuelo. La imagen de uno de los modelos de referencia, se muestra en la figura 2. Las trazas sísmicas, producto de la adquisición sísmica, ejemplificada por la figura 3, sobre el modelo de la figura 2, se muestra en la figura 4.

Figura 2: Modelo de velocidades de capas del subsuelo.



Inicialmente, las técnicas de optimización global se probaron con funciones matemáticas multivariable, permitiendo verificar el funcionamiento de las técnicas, ajustar el valor de sus parámetros y evaluar el desempeño de ellas en aspectos como: la precisión del mínimo encontrado y el número de iteraciones empleadas para su convergencia. Posteriormente, se implementaron estas técnicas en inversión sísmica de modo que, en vez de buscar un mínimo bidimensional en una función matemática multivariable, permitieran la búsqueda de los parámetros del modelo matemático. Estos parámetros representan el valor de la lentitud de propagación de las ondas mecánicas en cada capa del modelo

Figura 3: Ejemplo de una adquisición sísmica terrestre.

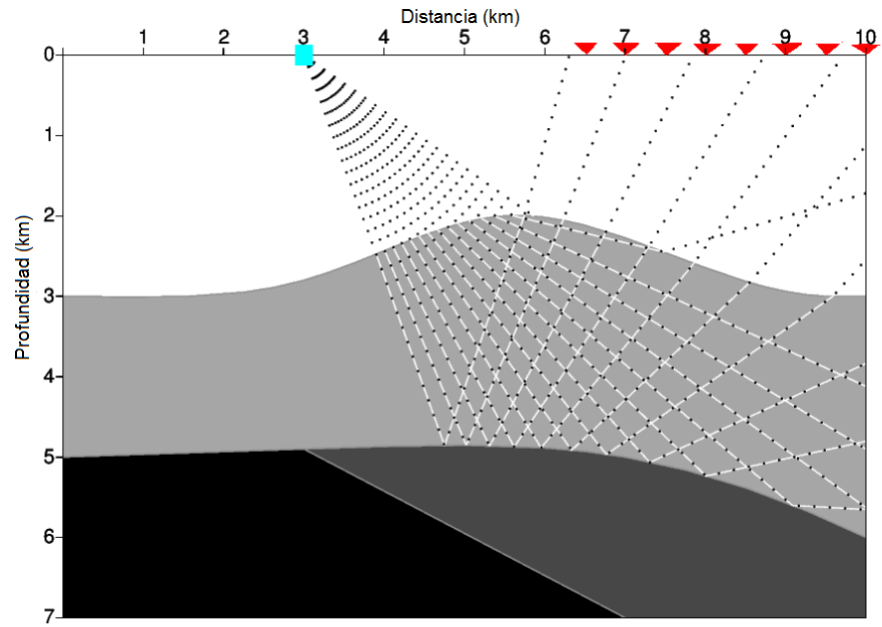
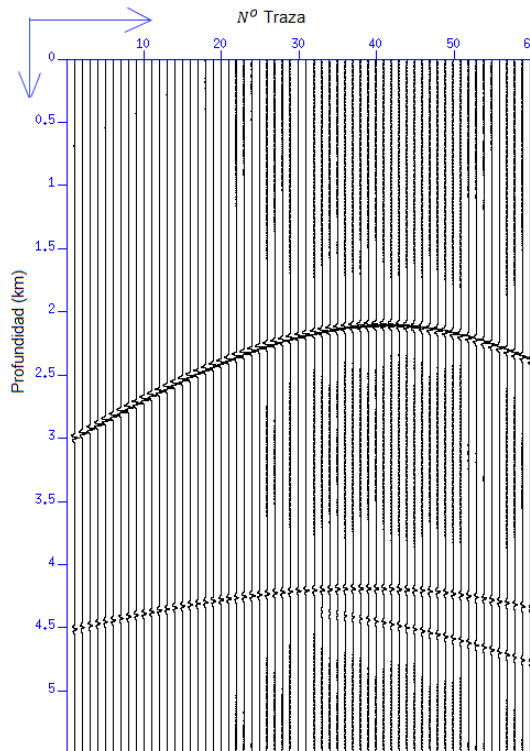


Figura 4: Trazas sísmicas obtenidas por una adquisición sísmica sobre el modelo de referencia.



del subsuelo. La lentitud de propagación se define como el cuadrado del inverso de la velocidad asociada a cada capa del modelo (ecuación 3).

$$Sloth = 1/V^2 \tag{3}$$

donde:

Sloth: Corresponde a la lentitud de la capa del modelo.

V: Corresponde a la velocidad de propagación del frente de onda de cada capa del modelo.

2. MARCO CONCEPTUAL

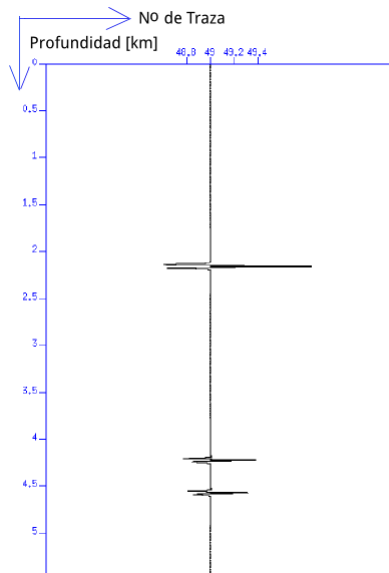
2.0.1. Geofísica Se define como la ciencia que estudia la composición, la estructura y las propiedades de la Tierra. Se basa en la experimentación e interpretación de los conceptos y métodos relacionados con áreas de estudio de la física como: la mecánica clásica, el electromagnetismo, los fenómenos ondulatorios y la termodinámica. Esta comprende el estudio de los fenómenos presentes tanto en el interior como en la superficie terrestre, dividiéndose así en geofísica interna que estudia y analiza las características y propiedades del interior de la tierra y geofísica externa que estudia y analiza las propiedades físicas de la capa superficial de la estructura de la Tierra [1].

2.0.2. Sismología Es la ciencia que se encarga del estudio de la propagación de ondas acústicas al interior y en la superficie de la Tierra. Estas ondas son generadas tanto por el fenómeno de la tectónica de placas, como por las detonaciones controladas mediante el uso de cargas explosivas de bajo poder [2]. Sus principales objetivos son:

- Estudiar los mecanismos de propagación de las ondas sísmicas en el interior de la Tierra para determinar su estructura.
- Establecer las causas que dan origen a los sismos de baja y alta intensidad.

2.0.3. Traza sísmica Es la representación gráfica (figura 1) de la trayectoria que presenta uno o varios frentes de ondas acústicas, al viajar desde un emisor (fuente) a un receptor (geófono) a través de las capas del subsuelo [3].

Figura 1: Traza sísmica generada mediante el software SU.



2.0.4. Modelo del subsuelo Es una representación aproximada de la estructura y composición de las capas que se encuentran en el subsuelo (figura 2). Muchos de estos modelos contienen parámetros asociados a las propiedades físicas del medio como: porosidad, velocidad, densidad, impedancia acústica, impedancia eléctrica, entre otros [4].

Figura 2: Modelo del subsuelo.



Fuente: [12]

2.0.5. Problema directo Es la representación de un fenómeno físico por medio de un modelo matemático, sobre el cual se busca describir el comportamiento del fenómeno a partir de sus parámetros conocidos.

Un ejemplo de problema directo corresponde a la relación de la tensión eléctrica presente entre los terminales de un trozo de alambre conductor de electricidad y la corriente eléctrica que lo atraviesa. El sistema de este problema se muestra en la figura 3 (a). La relación anterior se puede expresar matemáticamente mediante

$$V = I \times M, \quad (1)$$

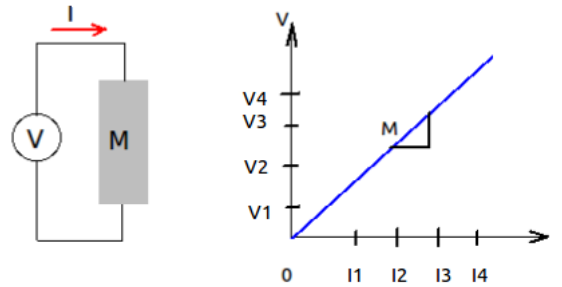
donde:

- V : Es el nivel de tensión presente entre los terminales de un trozo de alambre conductor de electricidad.
- I : Es la corriente eléctrica que fluye por el trozo de alambre conductor.
- M : Representa un parámetro denominado resistencia eléctrica, constante para el trozo de alambre utilizado.

A partir de la ecuación (1) se puede conocer la relación entre la tensión y la corriente eléctrica, en vista de que el valor del parámetro M es conocido. Por tal razón, es posible observar el comportamiento del fenómeno físico bajo consideración asignando valores

tabulados para la variable independiente I . La figura 3 (b) muestra la relación entre la tensión y la corriente sobre el alambre conductor.

Figura 3: Esquema modelado problema directo.

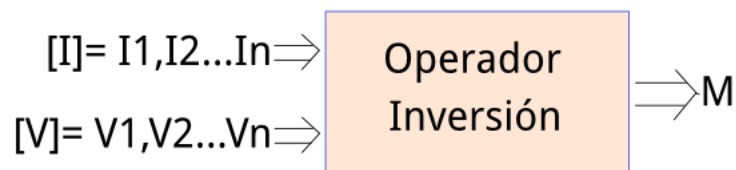


(a) Sistema del problema directo. (b) Relación entre la tensión y la corriente para un alambre conductor M conocido.

2.0.6. Problema inverso El proceso inverso consiste en la búsqueda de los parámetros de un modelo matemático dado, que mejor se ajusten a los datos observados para más entradas definidas. De esta manera, es posible encontrar el conjunto de parámetros que mejor ajustan el modelo al comportamiento real del fenómeno físico.

Un ejemplo de problema inverso corresponde a la búsqueda del valor del parámetro M , a partir del registro de datos experimentales de la corriente eléctrica I que circula por un trozo de alambre, y la tensión entre los extremos del mismo. La figura 4 muestra una representación gráfica del proceso realizado para la obtención del parámetro M a partir de los datos observados.

Figura 4: Esquema para la obtención de parámetros M , a partir de datos observados $[V]$ e $[I]$.

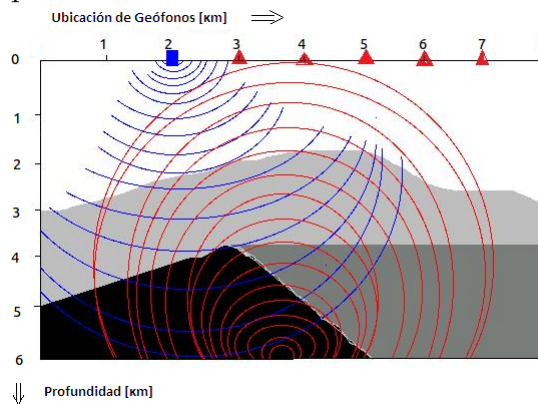


(a) Esquema problema inverso

2.0.7. Exploración sísmica Es un método físico para determinar la ubicación correcta de las estructuras geológicas en el interior del subsuelo, por medio de la explosión

de una fuente artificial. La propagación de los frentes de onda experimentan fenómenos de reflexión y refracción, debido a las discontinuidades existentes en las capas del subsuelo y a la elasticidad que estas presentan. Estos fenómenos de reflexión y refracción permiten que algunos de los frentes de onda regresen a la superficie de la Tierra siendo detectados por sensores conocidos como geófonos que almacenan su información [12]. La figura 5 muestra una representación gráfica de la exploración sísmica.

Figura 5: Proceso de exploración sísmica.



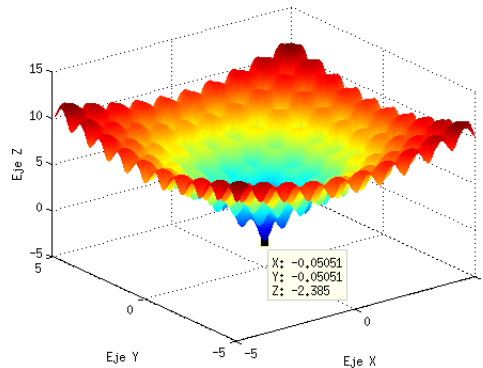
2.0.8. Inversión sísmica Se define como el problema inverso que permite reconstruir la estructura del subsuelo y obtener sus propiedades, a partir de un conjunto de datos obtenidos mediante el proceso de exploración sísmica. Estos datos son conocidos como trazas sísmicas.

En la inversión sísmica se identifican dos áreas de estudio, de acuerdo con lo expuesto en [8]. La *sismología* utiliza la variación de la amplitud de ondas mecánicas respecto al tiempo y distancia recorridas, para obtener un modelo definido en función de parámetros asociados a la elasticidad del material por capa [2]. Por otra parte, la *geomagnetotelúrica* utiliza mediciones relacionadas con la impedancia relativa y la fase, detectadas en determinadas ubicaciones e intervalos de tiempo, de modo tal que la reconstrucción de capas bajo el subsuelo describe cada una de éstas en función de los parámetros eléctricos, magnéticos y telúricos.

Según Ginzky en [7], en inversión sísmica es común usar dos tipos de información: la que proviene de pozos petrolíferos, conocida como registro de pozos y la información que proviene de las pruebas hechas en el área de interés, conocida como información sísmica. La inclusión de diferentes tipos de información permite obtener un modelo más preciso del subsuelo con todas las características no disponibles mediante modelado directo.

2.0.9. Optimización Se define como el procedimiento para determinar el mejor conjunto de valores que hacen mínima o máxima una función objetivo, satisfaciendo una o varias condiciones relacionadas con el proceso en particular [8]. Un ejemplo que permite observar la ubicación del mínimo global dentro de varios mínimos locales que posee la función objetivo, se muestra en la figura 6.

Figura 6: Ubicación aproximada del mínimo global para la función Ackley.



2.0.10. Técnica heurística Es una metodología o conjunto de pasos ordenados, basados en el uso de reglas empíricas para la búsqueda de una respuesta aproximada a un problema en particular.

Las técnicas heurísticas, a pesar de proveer un alto grado de confianza en su solución, no garantizan como óptimo su resultado. Por tal razón es fundamental evaluar la calidad de la solución por medio del análisis del peor de los casos, o contrastando con otros métodos de optimización o cotas que indiquen la solución del problema. De igual manera es importante conocer los casos en que son aplicables y los límites de su uso. El uso de estas técnicas permite dar solución a problemas donde los métodos exactos tienen un alto costo computacional.

2.0.11. Técnicas metaheurísticas Al igual que las técnicas heurísticas, éstas consisten en un conjunto de pasos ordenados, inspirados basan en la experiencia, para encontrar una respuesta que si bien no constituye una solución definitiva, se considera como una alternativa o indicio hacia la solución. La diferencia con las heurísticas radica en que éstas se diseñan con el propósito de mejorar el rendimiento de las mismas, ampliando su aplicación a los problemas que implican el uso separado de varias heurísticas.

Las metaheurísticas son conocidas como técnicas de propósito general creadas a partir de la integración de áreas como: genética, biología, matemática, física, inteligencia ar-

tificial, redes naturales, entre otras. Entre las técnicas metaheurísticas más conocidas, se destacan las técnicas inspiradas en la naturaleza, basadas en una población, basadas en la trayectoria, técnicas de búsqueda local y con memoria [11], [10].

2.0.12. Métricas de comparación Son medidas de similitud entre dos o más conjuntos de señales, vectores o datos que permiten estimar qué tan parecidas son la una de la otra a partir de un valor, coeficiente o vector. El uso de estas métricas es común en problemas de reconocimiento de patrones, donde se realiza una comparación entre los datos experimentales con los de referencia para determinar si el dato de entrada corresponde a alguno almacenado en la memoria del sistema. Por lo general, las métricas de comparación definen la diferencia entre los datos, representados por vectores característicos, por medio de un indicador conocido como distancia entre vectores.

Las definiciones de distancias más utilizadas para problemas de comparación tales como: distancia Euclídea, distancia Euclídea ponderada, distancia Mahalanobis, entre otras [11]. Otros tipos de métricas de comparación corresponden a medidas probabilísticas de dispersión como correlación y correlación cruzada [12]. El cálculo del error experimental o error absoluto también puede ser denominado como otro tipo de métrica, puesto que es un indicador asociado a la diferencia entre un dato experimental con uno de referencia.

Teniendo en cuenta que en el seminario de investigación sólo se utilizaron tres métricas de comparación, se considera necesario explicar únicamente estas métricas.

2.0.13. Correlación Es una medida de comparación muy usada en probabilidad y estadística, donde se caracteriza la relación que tiene una variable con otra. Esta operación puede ser representada mediante una línea de ajuste que indique el grado de similitud entre variables. La línea de ajuste consta de componentes como fuerza, sentido y forma, donde se clasifica la relación de fase, la relación proporcional o inversa y el ajuste lineal o curvo entre variables [13].

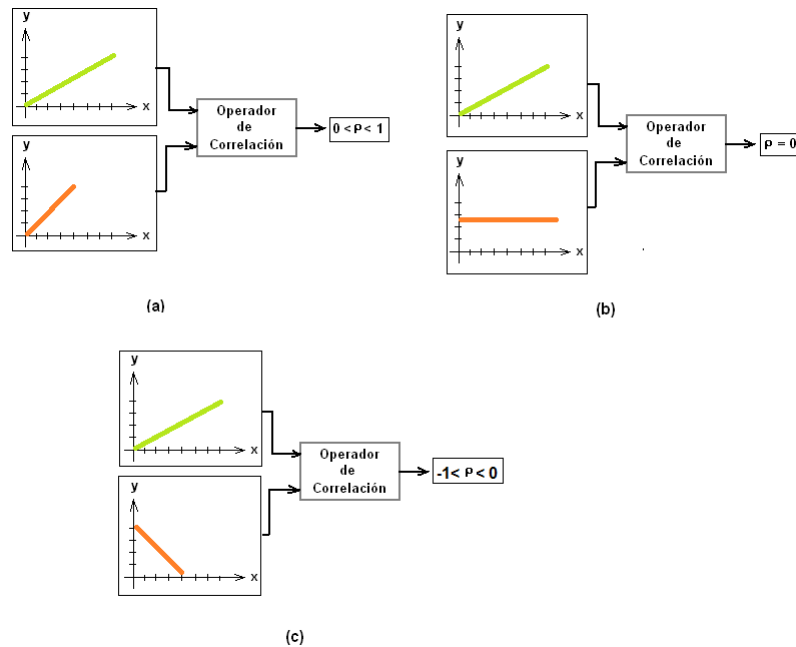
El tipo de comparación que realiza esta medida se hace componente a componente (punto a punto), donde el grado de similitud de las señales comparadas, se indica por medio un coeficiente de correlación (ρ). Este coeficiente se conoce como coeficiente de Pearson [14] e indica, en un rango de 0 a 1, la relación existente entre las variables a comparar. La interpretación según [13] y [14] se presenta de la siguiente manera:

- $\rho = 1$; Correlación positiva perfecta, con relación lineal y proporcional.
- $0 < \rho < 1$; Existe correlación positiva.

- $\rho = 0$; No existe relación lineal, podría existir una relación curva.
- $-1 < \rho < 0$; Existe correlación negativa.
- $\rho = -1$; Correlación negativa perfecta, con relación inversa.

Una ejemplo que permite ver de manera gráfica la estimación del coeficiente de correlación a partir de la comparación entre dos rectas con diferentes pendientes, teniendo en cuenta la interpretación de los valores de ρ , se muestra en la figura 7. Para el caso de la figura 7(a) se tiene un coeficiente de correlación ρ positivo en vista de que ambas rectas presentan cierta similitud directa donde algunos puntos coinciden. La figura 7(b) atribuye un coeficiente de correlación ρ de cero debido a que no hay ninguna similitud entre ambas rectas, que para este caso son perpendiculares entre sí. La figura 7(c) presenta un coeficiente de correlación ρ negativo en vista de que existe cierta similitud entre las rectas, pero de manera inversa.

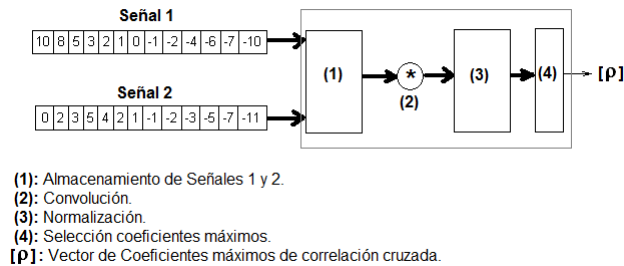
Figura 7: Ejemplo de comparación por correlación. a) Correlación positiva, b) Correlación nula, c) Correlación negativa.



2.0.14. Correlación cruzada Es una medida de similitud que comúnmente se aplica para la búsqueda de características similares de una señal desconocida a partir de una de referencia. Es una operación muy similar a la convolución, solo que difiere en que ninguna de las señales se invierte. La correlación cruzada entrega un vector de datos correspondientes a la sumatoria del producto punto entre cada componente de ambas señales, para cada posición de la trayectoria de operación. En vista de que el indicador no es un valor sino un vector, se deben normalizar sus componentes y luego

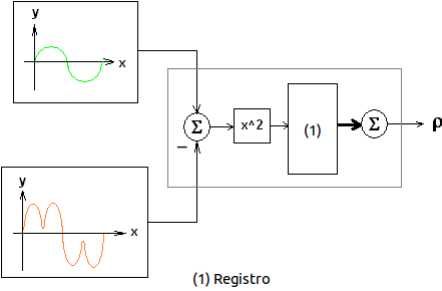
seleccionar los valores máximos correspondientes a cada trayectoria de operación, de modo que se pueda ver esta medida de similitud como los coeficientes de correlación cruzada. Estos se interpretan de igual manera a como se hace en correlación punto a punto [15]. A diferencia de la correlación, esta medida compara las señales en todo su conjunto, considerando los desplazamientos en tiempo de las mismas [16]. Un ejemplo que permite ver la secuencia utilizada en el proceso de comparación por medio de la métrica de correlación cruzada, se muestra en la figura 8. En esta representación se tienen como señales a comparar, dos vectores de datos que en primera medida se almacenan. Seguidamente se procede a calcular la convolución entre ambos vectores, recordando que ninguno de estos se invierte. Como tercer paso, se procede a realizar la normalización de las componentes del vector resultante con el fin de obtener coeficientes de correlación; para luego conformar un nuevo vector que contendrá los máximos coeficientes de correlación obtenidos a partir de la comparación entre los dos vectores de entrada. La elección de los máximos coeficientes de correlación cruzada se debe a que estos valores representan el instante en el que la similitud entre los vectores de entrada fue máxima.

Figura 8: Ejemplo de comparación de funciones por correlación cruzada.



2.0.15. Error absoluto Se define como una cuantificación de la desviación que presenta un dato experimental con respecto a uno de referencia. El error absoluto consiste en la diferencia componente a componente entre dos datos, indicando la magnitud del error a causa de su baja similitud [17]. Una ilustración del proceso que se realiza para calcular el error absoluto, se muestra en la figura 9. Los procesos para el cálculo del error absoluto consisten en hacer una diferencia de las señales de entrada, elevar al cuadrado cada componente de la señal resultante, realizar el registro de la señal y por ultimo, entregar el indicador del error absoluto entre las señales comparadas.

Figura 9: Error absoluto entre dos funciones.



3. MÉTODOS DE OPTIMIZACIÓN GLOBAL

Este capítulo describe las técnicas de optimización global denominadas: optimización mediante enjambre de partículas (*Particle Swarm Optimization, PSO*) y algoritmos genéticos (*Genetic Algorithms, GA*). Inicialmente se exponen los aspectos como la analogía con la naturaleza, la configuración de los parámetros y el pseudocódigo. Seguidamente, se describe la implementación en funciones matemáticas multivariable, para verificar el correcto desempeño de cada técnica. Y finalmente, se analiza el desempeño de las técnicas y se presentan las discusiones acerca de su implementación en funciones matemáticas multivariable. Estas discusiones son el punto de partida para la aplicación de las técnicas estudiadas en los problemas de inversión sísmica.

El interés por estudiar estas técnicas de optimización global, surge debido a la gran cabida que se le ha dado a estos algoritmos en la resolución de diversos problemas de optimización, en especial, en ingenierías. De igual manera, el interés va encaminado hacia la comprensión de la implementación de PSO y GA en problemas de inversión sísmica tal y como se menciona en el texto guía del seminario de investigación (*Global Optimization Methods in Geophysical Inversion* [8]). La elección de la técnica algoritmos genéticos GA, surgió a partir de una pre-selección de algoritmos de optimización estudiados en la primer fase del seminario de investigación, la cual consistía en la revisión teórica general asociada al problema de inversión sísmica enfocada en algoritmos de optimización global. Los algoritmos pre-seleccionados fueron: recocido simulado (SA), colonia artificial de abejas (ABC), algoritmos genéticos (GA) y colonia de hormigas. La elección de estos algoritmos se basó en la gran aplicabilidad que tienen en distintos problemas de optimización y de igual manera, la facilidad que se tiene a la hora de interpretar su funcionamiento, en vista de que son algoritmos bioinspirados. El descarte de los algoritmos colonia de hormigas y colonia artificial de abejas (ABC), se hizo partiendo de la gran similitud que guardan con el algoritmo PSO, ya elegido por el seminario de investigación, en el estilo de contar con agentes de búsqueda para encontrar el mínimo global del problema a optimizar. En cuanto a la técnica de recocido simulado (SA), se descartó por tratarse de un algoritmo basado en búsqueda por entornos, lo cual hace parte también de una de las características con las que PSO cuenta. Finalmente, la elección de GA se hace teniendo en cuenta que es una técnica evolucionista que representa una buena alternativa para comparar con respecto a la implementación de los problemas de inversión sísmica. Es importante mencionar que la técnica de optimización global PSO ya ha sido estudiada en los trabajos de grado de la Universidad Industrial de Santander, en la escuela de ingenierías Eléctrica, Electrónica y de Telecomunicaciones. Algunos de los usos que se le ha dado a este algoritmo, se ha encaminado en la solución de sistemas de ecuaciones no lineales según se trata en [2], [1], [9]. Por esta razón, los trabajos relacionados con la implementación de esta técnica, junto con el grupo de investigación CEMOS de la Universidad Industrial de Santander, fueron un gran referente para el desarrollo que se llevó a cabo en este seminario de investigación.

3.1. OPTIMIZACIÓN MEDIANTE ENJAMBRE DE PARTÍCULAS (PSO)

Es una técnica metaheurística que utiliza un grupo de partículas para explorar el espacio de solución de un problema o proceso. Las partículas recorren distintas posiciones dentro del área de posibles soluciones, de acuerdo al movimiento del enjambre, recordando la mejor posición individual y la mejor posición de grupo. La mejor posición individual representa las coordenadas que toma una partícula, para encontrar el menor valor de la función objetivo dentro de su propio recorrido. Esto significa que cada partícula tiene una mejor posición individual. Y dentro de este grupo de posiciones individuales, aquella que entrega el menor valor de la función objetivo, se define como la mejor posición de grupo o mejor posición global [5].

3.1.1. Analogía con la naturaleza Se basa explícitamente en el comportamiento que tienen bandadas de aves, bancos de peces, enjambre de abejas, entre otros; en la búsqueda del área donde se concentra la mayor cantidad de alimento. Para el caso de enjambre de abejas, cada una de éstas tiene su rumbo individual, pero pendiente y en comunicación con todo el enjambre. Como solución de búsqueda de alimento, se tiene el espacio que tenga el mayor área de flores y consigo de polen. Esta área es conocida como mínimo global, mientras que las de menor cantidad de alimento son conocidas como mínimos locales. Con lo anterior, el enjambre no está exento de llegar a mínimos locales, sin embargo, se espera que su capacidad de exploración; la comunicación entre partículas; y la capacidad de redirigir el enjambre en caso de encontrar una mejor solución, garanticen una mayor probabilidad de llegada al mínimo global. En otras palabras, no estancarse en una solución local.

3.1.2. Parámetros del algoritmo El algoritmo asigna cada partícula como una posición definida a un punto en un espacio de D-dimensiones. Cada partícula ajusta su vuelo mediante la ecuación de actualización de velocidad según Shi y Eberhart en [10]. Esta ecuación esta definida por

$$\vec{V}_i = (w * \vec{V}_i) + c_1 * rand * (\vec{P}_i - \vec{X}_i) + c_2 * rand * (\vec{G} - \vec{X}_i), \quad (1)$$

donde:

w: Peso de inercia que controla el desplazamiento de las partículas.

c_1 y c_2 : Hace referencia a los pesos que se incluyen en la ecuación de actualización de velocidad para darle mayor importancia a la búsqueda local o global del enjambre. Se definen mediante las variables c_1 para búsqueda local y c_2 para búsqueda global.

\vec{P}_i : Mejor posición individual.

\vec{G} : Mejor posición de grupo.

La ecuación de actualización de posición para cada partícula esta definida por

$$\vec{X}_i = \vec{X}_i + \vec{V}_i * dt, \quad (2)$$

donde:

dt : Radio de avance, entendido como el paso que utilizan las partículas para avanzar dentro del espacio de búsqueda. Un valor adecuado permite mantener las partículas dentro del espacio de búsqueda predeterminado.

\vec{V}_i : Vector velocidad actual de la i -ésima partícula.

\vec{X}_i : Vector posición actual de la i -ésima partícula.

Cabe mencionar que la ecuación de actualización de posición (2), no es la misma ecuación contemplada según Shi y Eberhart en [10]. La modificación que se hizo a criterio propio durante el seminario de investigación, fue la de añadir una nueva variable denominada radio de avance (dt), que permitiera hacer una juste con respecto al paso que hace una partícula al actualizar su posición.

Para comprender el comportamiento de las partículas considere el esquema de la figura 10. En la figura 10(a), se muestran las posibles alternativas que tiene una partícula para desplazarse. La magnitud de cada vector, indica la tendencia a tomar una determinada dirección. En la figura 10(b), se observa el vector resultante (vector rojo) que orienta la exploración local; también se observa el vector resultante que orienta la búsqueda global (vector celeste). En la figura 10(c), se observa que la suma de los efectos local, global e inercial (vector morado) determina la dirección y la velocidad de desplazamiento. Finalmente, en la figura 10(d) se observa la posición final de la partícula.

3.1.3. Pseudocódigo PSO La estructura del pseudocódigo se divide en dos partes: la declaración de variables y desarrollo del algoritmo. Esta última se subdivide en etapas denominadas: inicialización, actualización, cumplimiento de criterios y finalización. El pseudocódigo se puede observar en Algorithm 1.

3.1.4. Implementación del algoritmo La implementación se realizó mediante el software Matlab, el código utilizado conserva una estructura semejante a la expuesta en Algorithm 1. Adicionalmente se aplican dos criterios conocidos como: criterio de parada y de restitución de partículas, para detener el algoritmo y reubicar las partículas para los casos donde se salgan del espacio de solución. La descripción detallada del código implementado para PSO junto con las funciones que se requieren para su funcionamiento, se encuentran en la sección 6.

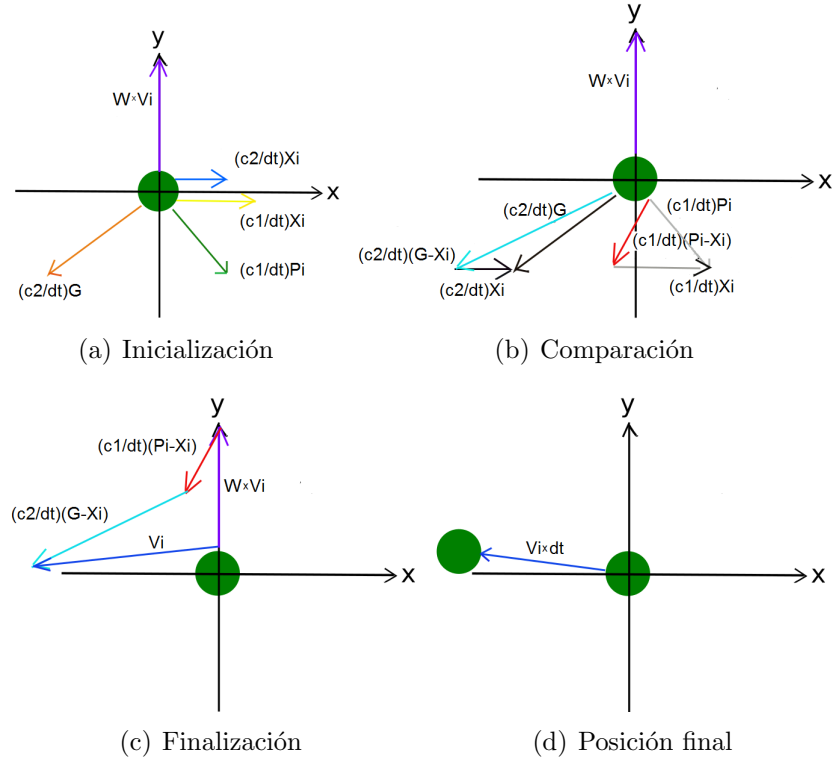
```

Data: w, dt, ns, c1, c2, ε1, ε2, saturación
Result: g
1 Inicio;
  /* Definir las fronteras de la función de referencia */
2 Xmín, Xmáx, Ymín, Ymáx;
  /* Ubicar las partículas en el área de exploración */
3 for i=1:ns do
4   | Ubicar cada partícula de manera aleatoria dentro del espacio de solución
5 end
  /* Inicializar vectores velocidad para cada partícula */
6  $\vec{V}_x = [0 \ 0 \ 0 \ \dots \ 0] : \vec{V}_y = [0 \ 0 \ 0 \ \dots \ 0]$ ;
  /* Definir función objetivo */
7  $f(x) = \sum_{i=1}^n x_i^2$ ;
  /* Evaluar la población inicial */
8 for j=1 : ns do
9   | Evaluar cada partícula en la función objetivo
10 end
11 Encontrar la mejor posición inicial;
  /* Obtener el mínimo global inicial y su posición */
12 g=argmin f(x);
  /* Actualizar la posición y velocidad para cada partícula */
13 for k=2:saturación do
14   for i=1 : ns do
15     for j=1:2 do
16       |  $V_i = (w * \vec{V}_i) + c_1 * rand * (\vec{P}_i - \vec{X}_i) + c_2 * rand * (\vec{G} - \vec{X}_i)$ ;
17       |  $\vec{X}_i = \vec{X}_i + \vec{V}_i * dt$ ;
18     end
19     Restituir las partículas que se salgan del espacio de solución, a la posición
        anterior;
        /* Actualizar el mínimo local */
20     if la población actual es mejor que la población anterior then
21       | Actualizar el mínimo local
22     end
23   end
  /* Actualizar el mínimo global */
24   if la mejor posición actual es mejor que la mejor posición anterior then
25     | Actualizar el mínimo global
26   end
  /* Criterio de Parada */
27   Garantizar que la máxima diferencia entre las posiciones de las partículas en
        la iteración anterior y actual, este por debajo del umbral ε2; y que la
        variación en la posición de la mejor partícula, en cinco iteraciones
        consecutivas, este por debajo del umbral ε1 ;
28 end
29 Mínimo global = g

```

Algorithm 1: Algoritmo PSO

Figura 10: Comportamiento de una partícula.

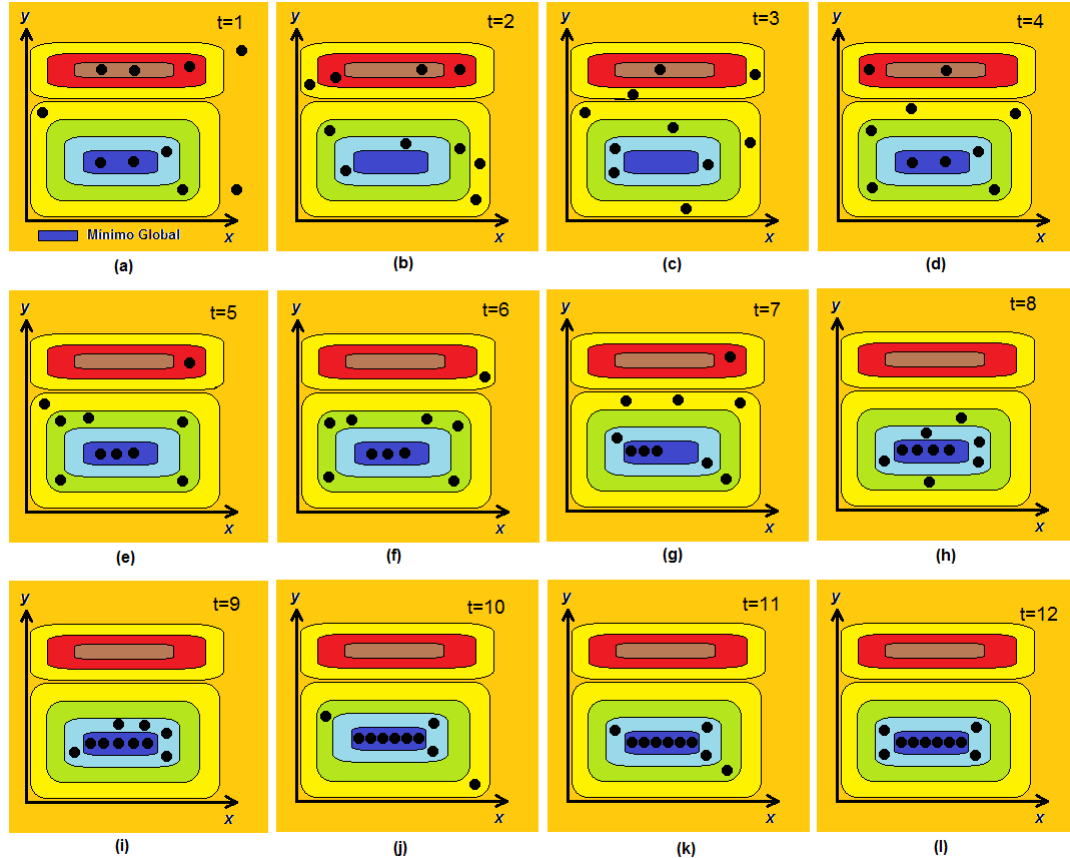


3.1.5. Evolución de las partículas Inicialmente, las partículas se distribuyen aleatoriamente dentro del espacio de solución de la función objetivo, mediante una distribución uniforme. La velocidad inicial del enjambre se considera cero y la posición inicial de cada partícula se fija de acuerdo a la distribución inicial, para la implementación que se hizo del algoritmo. Inicializada la velocidad y posición de cada partícula, éstas se evalúan en la función objetivo para así obtener una primera medida de mejor posición local y global de manera temporal. Luego de esto, es posible visualizar (por medio de gráficas o las ubicaciones de las partículas) la evolución de las partículas en la búsqueda del mínimo global del espacio de solución. En este proceso, se presentan comportamientos como: si la posición actual de una partícula cualquiera respecto a las demás posiciones del enjambre y las anteriores a ésta se encuentra más cerca al mínimo global, ésta partícula permanecerá en el mismo punto hasta que otra encuentre una mejor posición. Si se encuentra una mejor posición, las partículas restantes se desplazarán hacia ella. El comportamiento de las partículas cuando se desplazan hacia la mejor posición se conoce como convergencia, ésta es una de las características que se espera tenga el algoritmo. Para el caso donde la mejor posición cambia en medio del proceso de convergencia, el enjambre tendrá la posibilidad de redirigir su trayectoria con dirección en el nuevo punto.

Finalmente, se espera que la mayoría de las partículas alcancen el mínimo global, tal y como se ejemplifica en la figura 11. En la figura 11(a), se muestra la posición inicial de

las partículas. Luego, desde la figura 11(b) hasta la figura 11(k), se observa el recorrido de las partículas en el espacio de la función objetivo. La cantidad de partículas que se logran ubicar en el mínimo global y la rapidez con que lo hacen, dependen del valor que se define para los factores exploratorios y locales, el peso de inercia y el radio de avance. El ultimo paso se observa en la figura 11(l).

Figura 11: Evolución de las partículas.

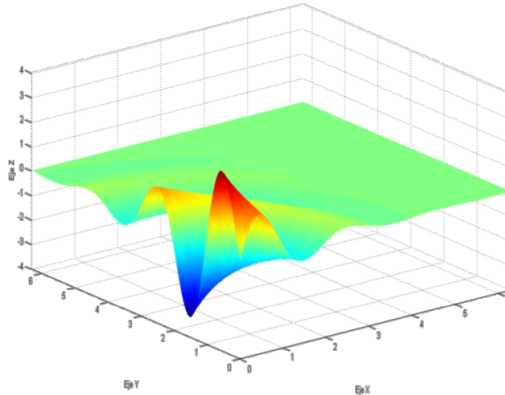


3.1.6. Descripción de las funciones multivariable utilizadas El uso de funciones de prueba se hizo con el fin de poner a prueba el funcionamiento de los algoritmos y a partir de su evaluación, poder caracterizar el comportamiento de los mismos frente a las distintas funciones objetivo y los distintos retos que estas ofrecen en la búsqueda del mínimo global contenida en cada una de ellas. El total de funciones de prueba implementadas fue de 13, conocidas como: función Ackley, Beale, Goldstein-Price, Easom, Eggholder, Holder-table, Esfera, Step, Rastrigin, Schwefel, Rosenbrock, Stochastic y función Básica denominada así en el seminario. A continuación se presenta una descripción correspondiente a los distintos tipos de funciones de prueba y de igual manera un análisis frente a los retos que estas pueden ofrecer.

3.1.6. Función básica La función consta de un sólo mínimo global y varios mínimos locales tal y como se observa en la figura 12. Como características principales, se tiene la presencia de una planicie y algunas elevaciones que contienen puntos máximos y zonas de estancamiento dentro de las fronteras establecidas para dicha función. El nivel de complejidad teniendo en cuenta las características morfológicas de la función, es bajo.

El desarrollo de las pruebas busca comprobar que el enjambre tiene la capacidad de comunicación entre partículas y consigo la habilidad de redirigir el enjambre hacia un mejor punto. Esto es posible gracias a la forma de la función, en vista de que posee áreas con mínimos locales y áreas donde la profundidad es constante.

Figura 12: Función de prueba Básica.

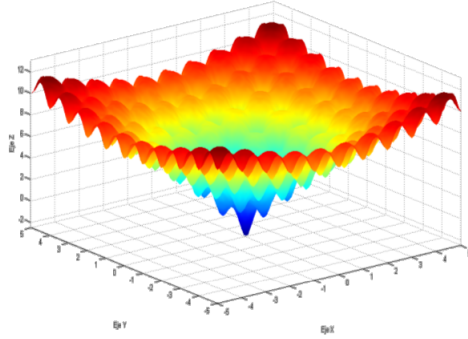


$$f(x, y) = 10 \sin(3\sqrt{x^2 + y^2})e^{(-\sqrt{(x+1)^2 + (y-1)^2})}$$

3.1.6. Función Ackley Se puede considerar como una función con un nivel de complejidad bajo puesto que el acceso a su mínimo global, que es único, se puede realizar sin ninguna dificultad desde cualquier punto dentro del espacio de solución (figura 13). A pesar de esto, la función ofrece curvas descendientes hacia el centro, lo cual podría ser un reto para el enjambre dado que la función da la sensación de que se pueden encontrar distintos puntos mínimos en pequeñas áreas y sin embargo el mejor punto se encontrará en el centro de la función.

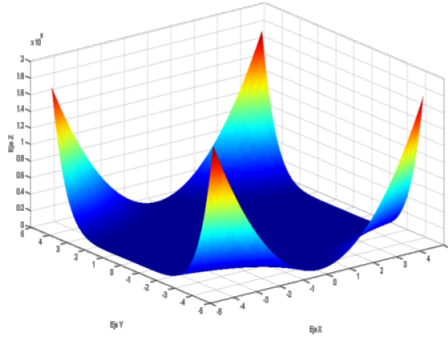
3.1.6. Función Beale Se puede considerar como una función con nivel de complejidad intermedio dado que presenta una gran planicie en el centro de la función lo cual ofrece un gran reto para el enjambre debido a que los puntos por donde transitan las partículas no presentan una mejora en el nivel de profundidad (figura 14). De ahí que la capacidad de exploración global deberá permitir al enjambre encontrar una mínima diferencia en amplitud con respecto a los distintos puntos alrededor del mínimo global de interés y de esta manera hacer converger el algoritmo.

Figura 13: Función Ackley.



$$f(x, y) = -20e^{(-0.2\sqrt{0.5(x^2+y^2)})} - e^{(0.5(\cos(2\pi x)+\cos(2\pi y)))} + e + 20$$

Figura 14: Función Beale.

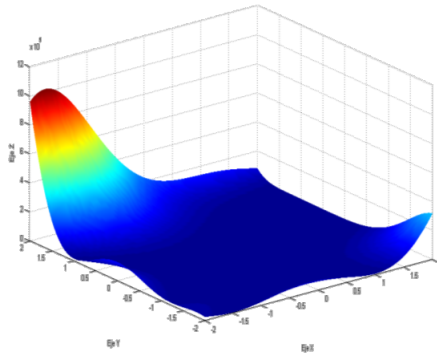


$$f(x, y) = (1.5 - x + xy^2) + (2.25 - x + xy^2)^2 + (2.625 - x + xy^3)^2$$

3.1.6. Función Goldstein Price Se puede considerar como una función con nivel de complejidad intermedio puesto que es muy similar en cuanto a las características de interés presentadas en la función Beale. La única diferencia con respecto a la función anterior radica en que solo uno de los puntos externos presenta un punto máximo, mientras que la función anterior presenta en sus cuatro extremos puntos máximos (figura 15).

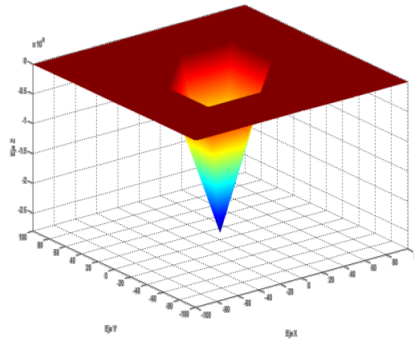
3.1.6. Función Easom Esta función también puede ser considerada con un nivel de complejidad intermedio, dado que presenta un único mínimo global en medio de una gran planicie. El reto que ofrece esta función se relaciona con el hecho de poder encontrar verdaderamente el mínimo global dado que la zona que presenta una menor amplitud con respecto a los demás puntos, desciende rápidamente (figura 16). En caso de no poseer una gran cantidad de información del área y una alta resolución en el avance por parte de las partículas, es posible que el enjambre tome un punto cercano al punto de interés como mínimo global o para el caso más desfavorable, caiga en un óptimo local.

Figura 15: Función Goldstein Price.



$$f(x, y) = (1 + (x + y + 1))^2(19 - 14x + 3x^2 - 14y + 6xy + 3y^2)$$

Figura 16: Función Easom.

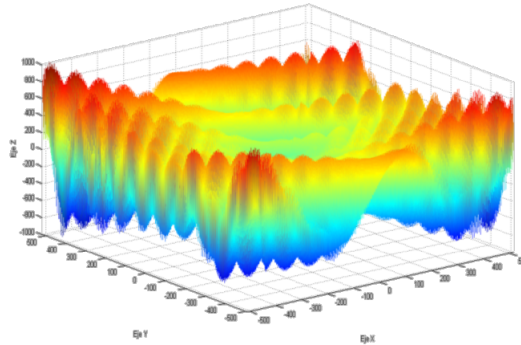


$$f(x, y) = -\cos(x) \cos(y) e^{-((x-\pi)^2 + (y-\pi)^2)}$$

3.1.6. Función Eggholder Se puede considerar como una función con un nivel de complejidad alto dado que presenta varios mínimos globales y un solo mínimo global que a no ser de una gran reconocimiento del espacio de solución, se puede confundir con uno de los muchos mínimos locales. Este tipo de función verdaderamente presenta un gran reto para el enjambre dado que presenta muchas zonas tentativas (figura 17), donde se evidencia un cambio brusco en el paso de un máximo a un mínimo y un descenso diagonal hacia puntos de menor amplitud.

3.1.6. Función Holder Table Se puede considerar como una función con un nivel de complejidad intermedio donde se puede encontrar un mezcla de los retos que ofrecen las funciones Easom y Eggholder en cuanto a la precisión en la búsqueda del mínimo global, en un área de descenso rápido, y la elección del mínimo global teniendo en cuenta el reconocimiento de todo el espacio de solución. Además de esto, la función presenta cuatro mínimos globales ubicados en los extremos de la función (figura 18),

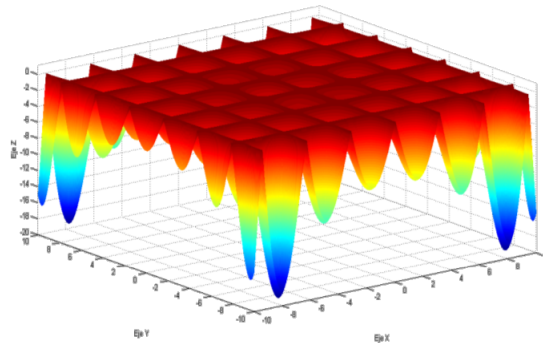
Figura 17: Función EggHolder.



$$f(x, y) = -(y + 47) \sin(\sqrt{|y + x \frac{\pi}{2} + 47|}) - x \sin(\sqrt{|x - (y + 47)|})$$

sin embargo la óptima búsqueda de dichos puntos depende del buen reconocimiento de la función en las zonas en donde tentativamente se pueden encontrar varios mínimos globales.

Figura 18: Función Holder Table.

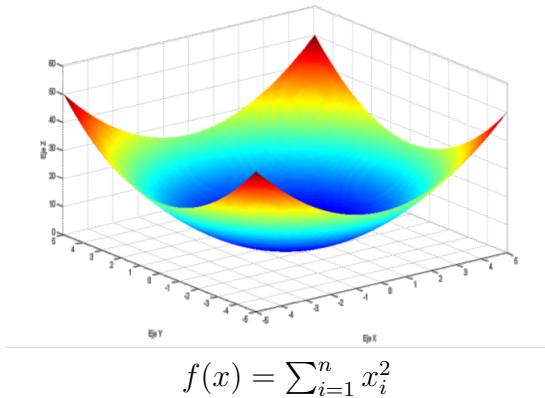


$$f(x, y) = -|\sin(x)\cos(y)e^{1-\frac{\sqrt{x^2+y^2}}{\pi}}|$$

3.1.6. Función Esfera Esta función presenta un nivel de complejidad bajo dado que solo presenta un mínimo global sin la presencia de otros mínimos. El acceso al punto de interés no presenta ningún tipo de dificultad, sin embargo, presenta un reto para el enjambre en cuanto a la precisión del punto de interés dado que se encuentra en una zona donde cualquier punto alrededor puede ser tomado como mínimo global sin serlo estrictamente. Este tipo de función (figura 19) más que todo es usada para la verificación del correcto funcionamiento del algoritmo.

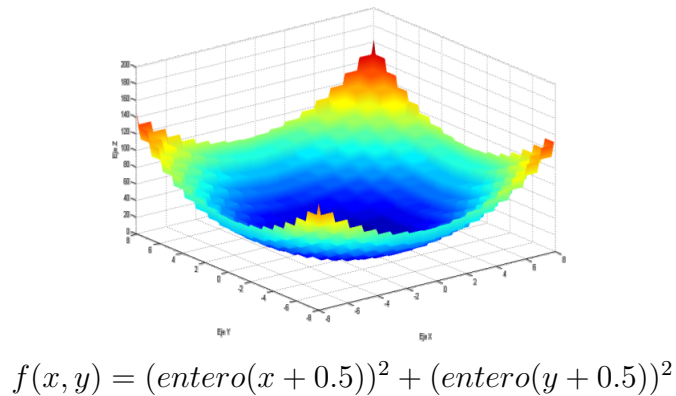
3.1.6. Función Step Es una representación discretizada de la función Esfera, donde entero: Es una función que aproxima al entero más cercano, por tal razón se considera con un nivel de complejidad bajo. El reto que quizás puede ofrecer se atribuye al hecho

Figura 19: Función Esfera.



de que el enjambre debe reconocer muy bien el espacio de solución y tener la capacidad de progresivamente ir transitando hacia las zonas con menor amplitud hasta llegar al punto de interés. (Figura 20).

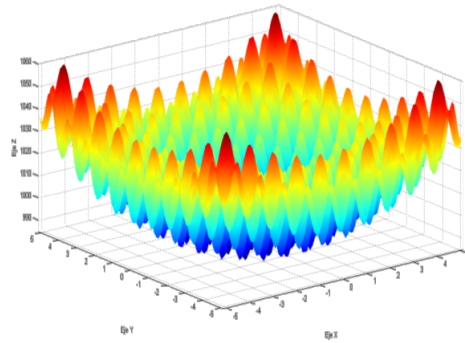
Figura 20: Función Step.



3.1.6. Función Rastrigin Se puede considerar con un nivel de complejidad alto debido a que presenta una gran cantidad de mínimos locales y un único mínimo global, que a no ser de un gran reconocimiento de todo el espacio de solución puede confundirse con los mínimos locales que se encuentran alrededor, $x_i \in [-5.12, 5.12]$. El reto que presenta dicha función es similar al que ofrece la función Eggholder, donde se evidencia constantemente un cambio entre puntos máximos y mínimos. (Figura 21)

3.1.6. Función Schwefel Esta función también considerada como una función con un nivel de complejidad alto, ofrece el mismo reto de la función Rastrigin. La única diferencia con respecto a esta se debe a que el cambio entre los puntos máximos y mínimos es más lento y presentan una mayor área. Estas características hacen interesante la función puesto que presenta un único mínimo global con una pequeña diferencia

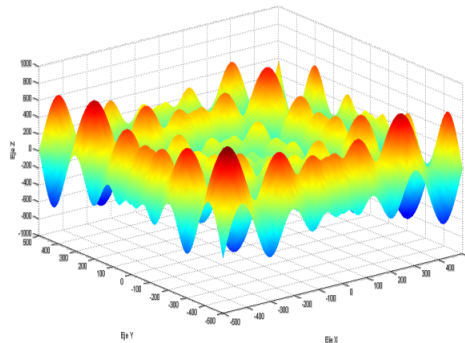
Figura 21: Función Rastrigin.



$$f(x) = A_n + \sum_{i=1}^n [x_i^2 - 10 \cos(2\pi x_i)]$$

de amplitud con respecto a los mínimos locales ubicados en los extremos de la función. (Figura 22).

Figura 22: Función Schwefel.

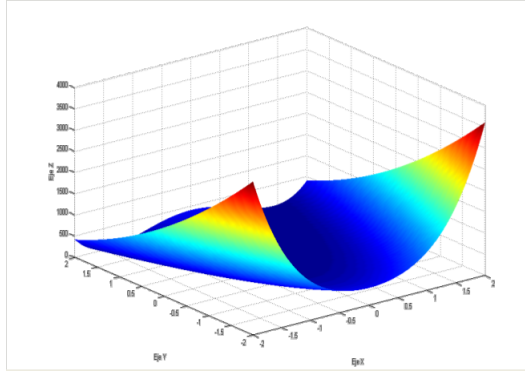


$$f(x, y) = (-x \sin(\sqrt{|x|}) + (-y \sin(\sqrt{|y|})))$$

3.1.6. Función Rosenbrock Se puede considerar como una función con un nivel de complejidad intermedio. A diferencia de las funciones que presentan una gran planicie y en medio de ella el mínimo global, este tipo de función presenta un área con forma de parábola para los puntos más bajos en amplitud (figura 23). el reto que ofrece esta función corresponde a la precisión que se debe tener para llegar al punto exacto de interés y no converger prematuramente a causa de no evidenciar un cambio notorio en amplitud en los puntos de la planicie y de la zona más baja del espacio de solución.

3.1.6. Función Stochastic Para este tipo de función se puede considerar un nivel de complejidad alto. La función presenta un fácil acceso por cualquier punto externo hacia el mínimo global que es único y esta centrado (figura 24), sin embargo presenta el reto de contener una gran variedad en las amplitudes de los puntos, evidenciando un

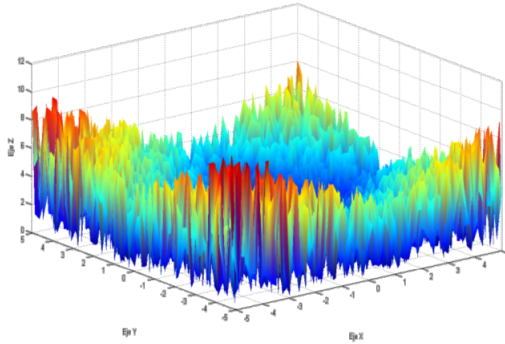
Figura 23: Función Rosenbrock.



$$f(x) = \sum_{i=1}^{n-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$$

cambio brusco entre los máximos y mínimos. El reto que representa para el algoritmo se relaciona con el hecho de encontrar verdaderamente el mínimo global y no los posibles mínimos locales situados alrededor del punto de interés o en cualquier otro punto de la función. El factor $e1$ es un número generado aleatoriamente.

Figura 24: Función Stochastic.



$$f(x, y) = e1 * (abs|x - 1|) + e1 * (abs|y - \frac{1}{2}|)$$

3.1.7. Implementación sobre funciones multivariadas La implementación del algoritmo PSO se realizó con las 13 funciones de prueba descritas en la sección 3.1.6. Cabe resaltar que el mínimo global de cada una de estas funciones se podría encontrar por medio de métodos directos, como por ejemplo la derivación. Sin embargo, la manera en como se procedió a calcular el mínimo de cada función se hizo con el fin de caracterizar el comportamiento del algoritmo.

En las implementaciones realizadas no se utilizaron topologías de agrupación del enjambre para la distribución inicial de las partículas de búsqueda. De igual manera, se varió un solo parámetro a la vez, dejando los demás fijos con valores recomendados en

la literatura [2], [1], [9] o con valores que se consideraron adecuados, en vista de que los resultados del mínimo global encontrado eran aceptables.

Dentro de las modificaciones hechas en la implementación, se destaca el uso de animaciones y de la herramienta *contour*, programadas en Matlab. Estas permitieron ver la evolución de las partículas en las diferentes funciones de prueba desde una vista superior. Esto con la finalidad de analizar mejor el comportamiento del algoritmo a medida que se acercaba a la solución.

Para registrar los resultados de la implementación del algoritmo PSO en funciones de prueba, se elaboró una hoja de cálculo en Excel que contiene tablas como la 1. Estas tablas están ubicadas de manera vertical por funciones, mostrando los resultados de cada una de ellas. La hoja de cálculo se encuentra en la carpeta Tablas en el CD anexo. En la parte lateral izquierda de la tabla se registra el valor de los parámetros con los que se realizó el experimento y, en la parte superior, el nombre de las variables como: el número de iteraciones (k) y las coordenadas (pg_x , pg_y y $f(x, y)$) del mínimo global encontrado por el algoritmo. Cada experimento se hizo 10 veces.

Tabla 1: Resultados para la función básica.

Parámetros	k	pg_x	pg_y	$f(x, y)$
$np = 30$	42	0,0376	1,5159	-3,0970
$c1 = 1,9$	79	0,2342	1,5293	-2,6043
$c2 = 1,8$	42	0,0001	1,5204	-3,2019
$w = 0,1$	64	1,2320	0,9765	-1,0730
$\epsilon 1 = 0,1$	44	0,4883	1,4254	-2,0878
$\epsilon 2 = 0,1$	46	0,9677	1,2133	-1,3795
$dt = 0,5$	38	0,5128	1,4141	-2,0422
$pg_x = 0$	50	0,0182	1,5187	-3,1507
$pg_y = 1,5387$	54	0,2174	1,5254	-2,6448
$f(x, y)_{teorico} = -3,1965$	90	0,4663	1,4594	-2,1367
<i>Promedio</i>	54,9			

3.1.8. Análisis de Resultados Como se mencionó al inicio de este capítulo, la implementación del algoritmo se hizo sobre un conjunto de funciones multivariantes con el fin de poner a prueba el funcionamiento de los algoritmos y, a partir de su evaluación, poder caracterizar el comportamiento de los mismos frente a los distintos retos que las funciones ofrecen. El computador utilizado para realizar las pruebas concernientes a la implementación del algoritmo de optimización global PSO en funciones multivariable, tiene las siguientes especificaciones:

- Procesador Intel(R) Core i3-2310M CPU @ 2,10GHz
- Memoria RAM DDR3 de 2 GB
- Sistema operativo Ubuntu 12,04 de 32 bits

A continuación se presenta el análisis de los resultados obtenidos con de la implementación del algoritmo sobre las funciones multivariable utilizadas, junto con la descripción del pre-análisis de los factores más influyentes en el comportamiento del enjambre, necesarios para que las pruebas realizadas se encaminaran en la obtención de resultados cercanos a los valores de referencia.

Ubicación inicial de las partículas La ubicación inicial con una distribución uniforme, evita sesgar el punto de partida y permite el cubrimiento uniforme del espacio de solución desde el inicio. En este orden de ideas, es más fácil y rápido para el algoritmo encontrar el mínimo global, si inicialmente se cuenta con que una de las partículas tome un valor cercano al punto de interés que se esta buscando. A partir de esto, se ve la importancia de ser imparciales en la ubicación inicial de las partículas y la manera como se distribuyen las mismas en el espacio de solución.

Un ejemplo de lo anterior se puede observar en las figuras 25 y 26, donde un mismo experimento con los mismos parámetros, presenta dos casos distintos de convergencia del enjambre hasta la iteración final, debido a la ubicación inicial de las partículas. En la figura 25, se puede observar que la convergencia de las partículas es más compacta debido a que la posición inicial de una de estas estuvo cercana al mínimo global, favoreciendo así la atracción del enjambre hacia la mejor posición. Un caso contrario se evidencia en el experimento mostrado en la figura 26, en el que todas las partículas en su posición inicial quedaron lejos de la mejor solución y, la convergencia final de las mismas no fue tan compacta a pesar de llegar a la solución.

Figura 25: Posiciones iniciales (a y b) y finales (c), del primer experimento con la función básica.

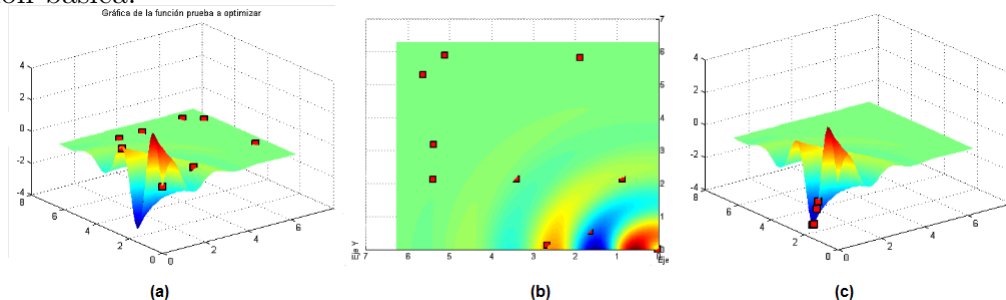
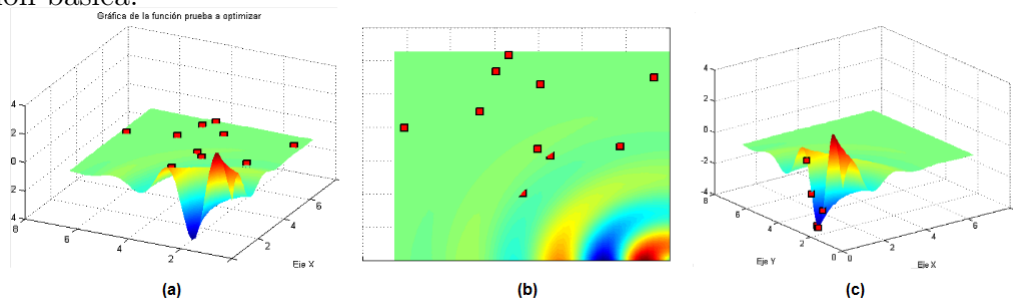


Figura 26: Posiciones iniciales (a y b) y finales (c), del segundo experimento con la función básica.



Tamaño de la población El tamaño del enjambre es otro factor que influye en el comportamiento y la precisión del algoritmo en la búsqueda del mínimo global. Por tal razón, se hicieron pruebas en las que se incrementó sistemáticamente el número de partículas con el fin de establecer la influencia de este parámetro en la convergencia del algoritmo. A partir de esto se observó que si el número de partículas en el enjambre es muy bajo (alrededor de 10 partículas), la información del espacio de solución y el reconocimiento del mismo carece de diversidad. Un caso contrario ocurre en enjambres de mayor tamaño (alrededor de 15 a 20 partículas) donde la mayor cantidad de información permite examinar un mayor número de zonas del espacio de solución y, a partir de esto, dirigir el enjambre hacia zonas más profundas a medida que éste va haciendo un reconocimiento más detallado del espacio. Esto se puede observar en la figuras 27, donde se muestra la convergencia final para una cantidad de 10, 15 y 20 partículas, respectivamente.

La cantidad de partículas en el enjambre también influye en la calidad o precisión de los resultados encontrados (refiriéndonos al mínimo global) puesto que un número de partículas bajo (menor a 10 partículas) no ofrece la misma información que un número alto (20 a 30 partículas), donde dicha información es necesaria para la depuración del área que la técnica metaheurística permite localizar. En la tabla 2 (lateral izquierda) se observa que para un número de partículas de 20, la posición donde se ubica el mínimo global oscila entre valores cercanos al ideal que tiene como coordenadas $x = 0$ e $y = 1,52$. Las constantes ni , np y w corresponden al número de iteraciones, número de partículas y peso de inercia respectivamente.

Por otra parte, la reducción a la mitad en el número de partículas lanzadas junto con un número alto de iteraciones permitidas y con un valor constante (30 iteraciones) como se evidencia en la tabla 2 (tabla central), presenta resultados similares a los mostrados en la tabla lateral izquierda. En ésta posición del mínimo varió con respecto al experimento previo en 10^{-3} ; pasando de una posición en el eje vertical (eje y) de 1,5167 a 1,5177.

El lanzamiento de pocas partículas, de acuerdo con los datos obtenidos en la tabla

Figura 27: Posiciones finales para los experimentos con 10, 20 y 30 partículas en la función básica.

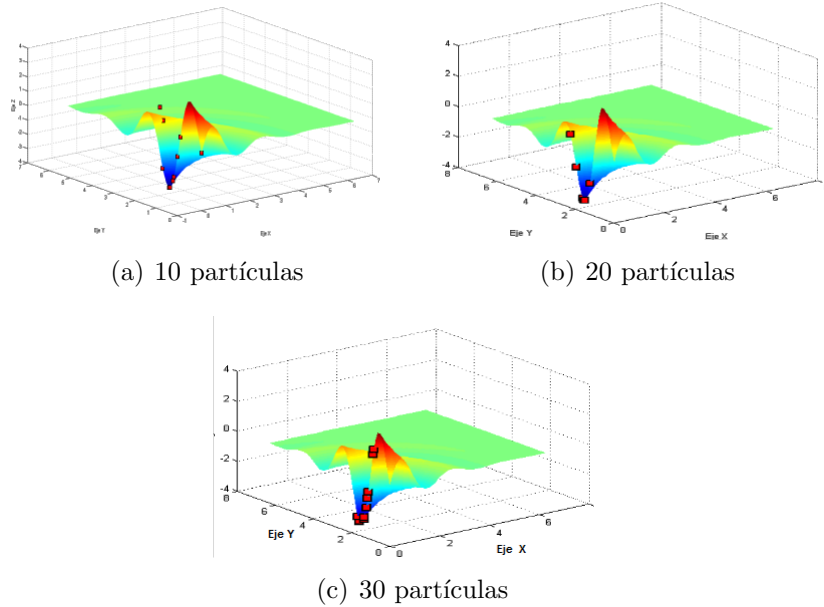


Tabla 2: Posición de la mejor partícula en 10 ejecuciones para $[ni=30, np=20, w=0,7]$, $[ni=30, np=10, w=0,7]$ y $[ni=30, np=5, w=0,7]$ en la función básica.

pg_x	pg_y	pg_x	pg_y	pg_x	pg_y
0	1,4979	0	1,5216	0	1,4476
0	1,5192	0	1,4877	0	5,6585
0	1,5205	0	1,5215	0	1,3154
0	1,5142	0	1,5222	0	1,7644
0	1,5174	0	1,5339	0	1,7327
0	1,52	0	1,518	0	1,5051
0	1,5212	0	1,5209	0	1,5388
0	1,5193	0	1,5192	0	5,6627
0	1,5187	0	1,5197	0	1,4559
0	1,5183	0	1,5124	0	1,4851

2 (lateral derecha), evidencia un distanciamiento notorio del mínimo encontrado con respecto a los experimentos anteriores. Esto se debe a que el enjambre no todas las veces llegó al mínimo global, sino que se estancó en mínimos locales a causa de la poca información, por tanto este algoritmo requiere de una cantidad de partículas (por ejemplo 30) para obtener un mejor desempeño. En este caso se debe a la cantidad de información que permite reconocer de mejor manera el espacio de solución. Otro ajuste que se puede implementar corresponde a la reducción del área de exploración en caso de utilizar una cantidad baja de partículas, esta alternativa fue consultada en [24], pero no fue implementada dado que implica manipular el espacio de solución a conveniencia del resultado que se busca obtener.

Cabe aclarar que para la asignación del número de partículas, se optó por realizar pruebas en donde el número de iteraciones fuese un valor constante, de modo que se convirtiera en la limitante para detener el algoritmo. Las tablas correspondientes a estas pruebas son la 2. Otro aspecto que hay que resaltar es la consideración que se tomó en el seminario de investigación con respecto al tamaño del enjambre. En trabajos similares con PSO desarrollados en la Universidad Industrial de Santander [2], [1], [9], se consideran tamaños de la población entre 20 y 3000 partículas. Sin embargo, los tamaños que se consideraron viables para este estudio estuvieron en el rango de 10 a 30 partículas, en vista de que los resultados obtenidos bajo estos parámetros eran cercanos a los de referencia y el tiempo de cómputo no era excesivo (no superaba las 100 iteraciones).

Radio de avance Otro factor que influye en el comportamiento del enjambre corresponde al radio de avance que controla el paso que da una partícula en cada actualización para avanzar dentro del espacio de solución. En pruebas preliminares, se observó que un radio de avance del orden de 10^{-2} y 10^{-3} , ocasiona que las partículas se muevan muy lento, produciendo un número de iteraciones superiores a 100 y un gasto innecesario en memoria de las posiciones recorridas. Las funciones donde es fácil que este fenómeno se presente son: Beale, Goldstein-Price, Easom, Holder-table y Rosenbrock. La característica en común de este tipo de funciones es una gran planicie que para el caso de pasos del orden de 10^{-3} , da la sensación al algoritmo de que no hay posibilidades de encontrar un mejor punto y por tal razón este se detiene. Otra característica corresponde a la existencia de áreas profundas, lo cual ocasiona que las partículas no tengan la capacidad suficiente para salir de ese hueco y por el contrario encuentren el punto más profundo de esa área pensando que es el mejor punto en todo el espacio de solución.

Adicionalmente se hicieron pruebas preliminares en donde se asignó un radio de avance de uno. Para este caso, se presentó que el avance de cada partícula al momento de hacer su actualización es muy brusco (distancias grandes de una posición a otra), lo que provoca que en muchas ocasiones las partículas se salgan del espacio de solución. Las funciones donde no se presentó este fenómeno fueron: Easom, Eggholder, Holder-table y Schewefel. La razón por la que este parámetro no afecta a este tipo de funciones corresponde al hecho de que son funciones con fronteras muy amplias, a comparación de las demás funciones, donde el espacio de solución es pequeño, lo que ocasiona la salida frecuente de las partículas dentro de ésta área. Por esta razón se define un valor para éste parámetro del orden de 10^{-1} , de modo que el radio de avance garantice la permanencia de las partículas dentro del espacio de solución. De igual manera, se garantiza un avance eficiente, de modo que no produzca gastos innecesarios en iteraciones y memoria para los puntos recorridos.

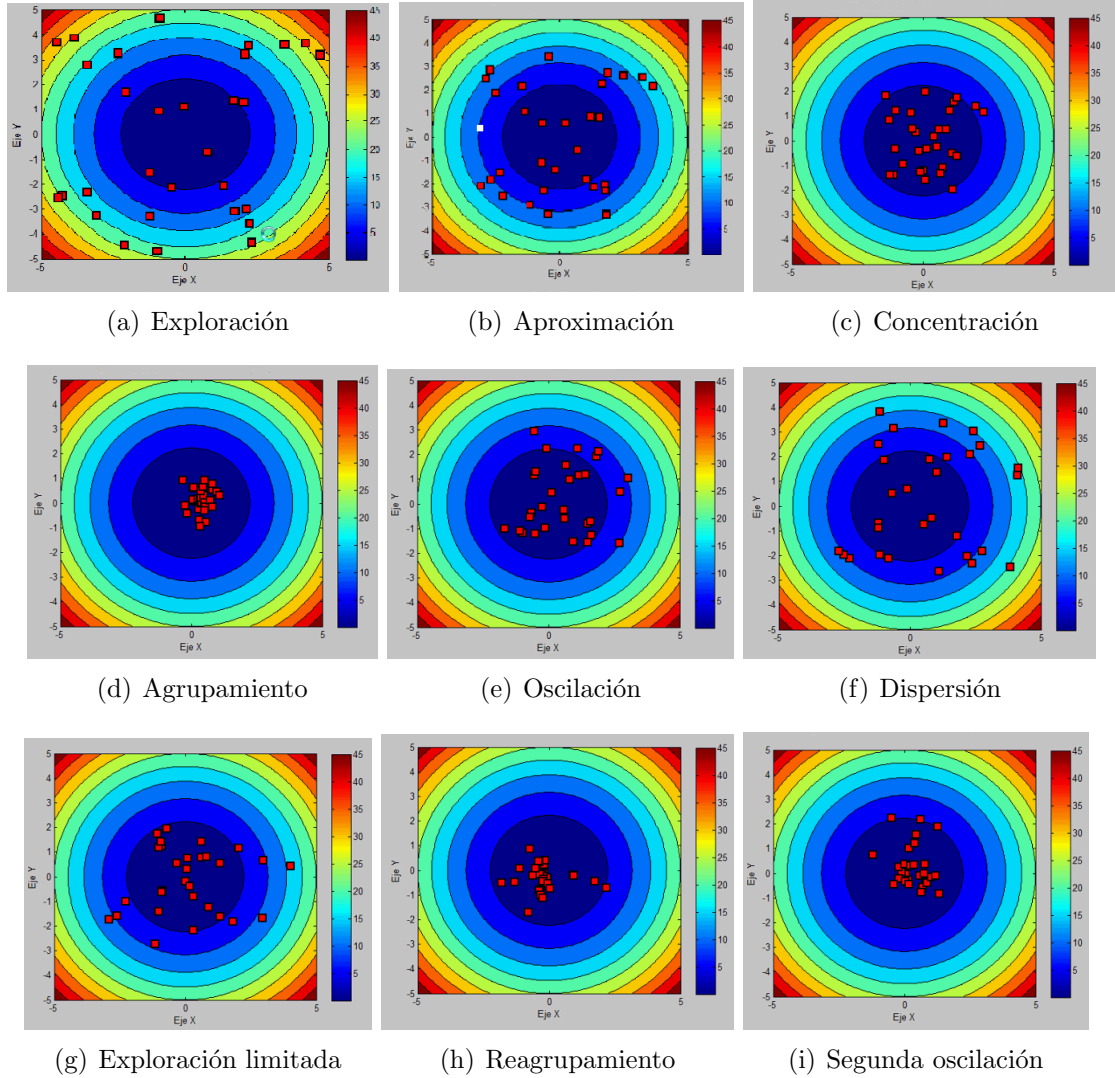
Valor de los ϵ_1 y ϵ_2 en el criterio de parada Otro factor que influye de manera directa en la parada de la ejecución del algoritmo, corresponde a los ϵ_1 y ϵ_2 necesarios para la verificación de si el algoritmo convergió y encontró de manera repetitiva el mínimo global. Este criterio de parada consiste en garantizar que la máxima diferencia entre las posiciones de las partículas en la iteración anterior y actual, este por debajo del umbral ϵ_2 ; y que la variación en la posición de la mejor partícula, en cinco iteraciones consecutivas, este por debajo del umbral ϵ_1 . Para la asignación de un valor adecuado se realizaron pruebas preliminares donde se llegó a la conclusión de que el valor de los ϵ correspondientes a los criterios de parada deben ser del mismo orden que el delta de tiempo dt que establece el radio de avance de las partículas. De no ser así, el criterio de parada podría estar deteniendo el algoritmo en tan solo un promedio de 10 ó 20 iteraciones. La razón corresponde al hecho de que el orden de estos parámetros (dt , ϵ_1 y ϵ_2) debe ser el mismo, puesto que los ϵ son tomados como un umbral que verifica la diferencia máxima de las partículas entre una iteración y otra. Por esta razón se debe permitir el movimiento de las partículas y, de igual manera permitir que el mínimo encontrado sea consecutivo en cinco iteraciones seguidas.

El radio de avance debe ser del mismo orden porque al presentar un orden menor, el criterio de parada pensará que tanto la diferencia, como el mínimo, no registran un cambio de posición considerable y, por lo tanto, detiene el algoritmo. Siendo la verdadera razón que el avance de cada partícula es muy pequeño por lo que la comparación entre una iteración y otra no es considerable. Para el caso en que dt es mayor a los umbrales ϵ , ocurrirán casos en donde se presenten iteraciones mayores a 100, a causa de la salida de las partículas del espacio de solución, y del brusco avance en cada actualización, ocasionando que el criterio de parada se demore en detener el algoritmo.

Variación en el peso de inercia El peso de inercia es otro factor que determina el comportamiento del enjambre, éste parámetro es el encargado de controlar el impacto de la historia previa de las velocidades, en el cálculo de la actual de cada partícula en enjambre. Teniendo en cuenta esto, el peso de inercia influye directamente en la manera como se aproxima el enjambre a la mejor solución logrando, en cierto modo, tener control de la rapidez que experimenta el mismo en la convergencia hacia el punto de interés. Un ejemplo que representa la influencia de este parámetro podría mostrarse por medio de la secuencia de la ubicación de las partículas en la búsqueda del mínimo global. Sin embargo, es dificultoso, por motivos de tamaño, presentar en imágenes el paso a paso de las iteraciones debido a que superan las 30, por tal razón, es mejor remitirse a la carpeta Tablas del CD de anexo donde se presenta de manera análoga lo expuesto anteriormente.

En vista de que la rapidez de convergencia del algoritmo se ve influenciada por la variación del peso de inercia, ocurren casos en los que el enjambre encuentra la solución y debido a la alta inercia que lleva en su movimiento, se aleja de ella. Para evitar

Figura 28: Movimiento oscilatorio alrededor del mínimo global.

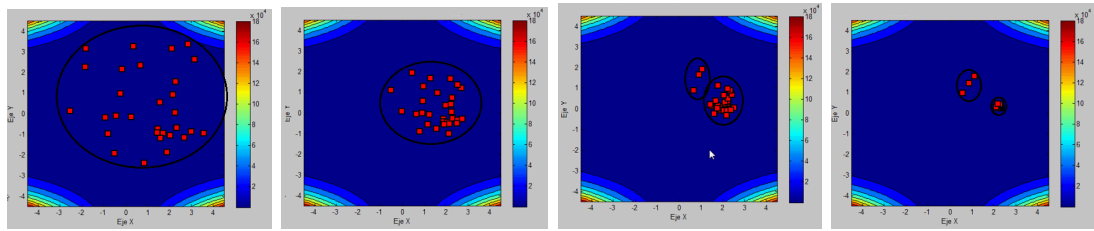


casos como éste, es indispensable la implementación de los criterios de parada que lo garantizan para que el enjambre converja en el punto exacto y no oscile alrededor de él. Una explicación del comportamiento del enjambre, sin implementar el criterio de parada, se puede observar en la figura 28. En esta se evidencia la dispersión transitoria y el movimiento oscilatorio alrededor del mínimo encontrado a causa del peso de inercia definido para el enjambre.

El peso de inercia influye en la dinámica de actualización de la velocidad y la posición de cada una de las partículas, así como su capacidad de exploración global y local. Un valor mayor a 0,5 en el peso de inercia w mejora la exploración global de las partículas dentro del espacio de solución delimitado según el tipo de función objetivo. Este comportamiento se observa en la figura 30 secuencia (a-c). Otro ejemplo en donde se

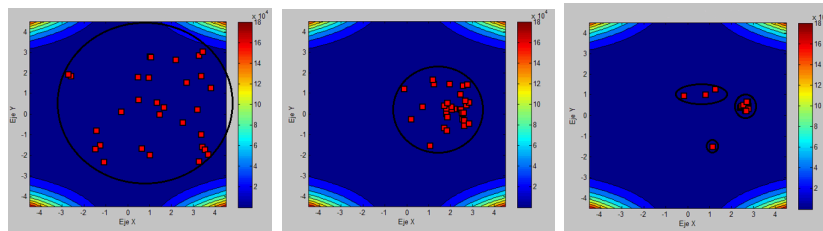
evidencia mayor exploración global, se encuentra en la figura 31 secuencia (a-d). Un valor menor a 0,5 para el peso de inercia permite una búsqueda más detallada en las zonas por donde se desplaza cada partícula del enjambre. Un ejemplo que evidencia lo mencionando anteriormente, se muestra en la figura 29 secuencia (a-d). En esta figura se observa que el peso de inercia es inferior a 0,5, lo cual facilita la búsqueda local y un agrupamiento mayor de partículas en la zona donde se localiza el mínimo global.

Figura 29: Dinámica del enjambre para $w = 0,1$.



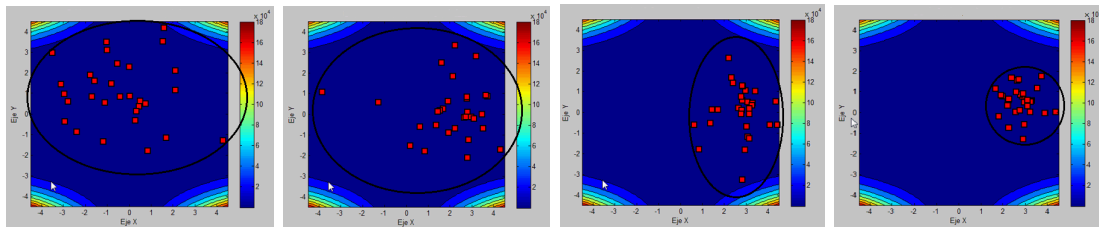
(a) Exploración después de 4 iteraciones (b) Concentración después de 10 iteraciones (c) Agrupamiento después de 35 iteraciones (d) Posición final después de 52 iteraciones

Figura 30: Dinámica del enjambre para $w = 0,5$.



(a) Inicio transcurridas 5 iteraciones (b) Agrupamiento después de 20 iteraciones (c) Posiciones finales después de 55 iteraciones

Figura 31: Dinámica del enjambre para $w = 0,9$.



(a) Posiciones iniciales durante 3 iteraciones (b) Exploración después de 8 iteraciones (c) Agrupamiento después de 25 iteraciones (d) Posiciones finales al transcurrir 60 iteraciones

El número de iteraciones como indicador para evaluar qué tan rápido converge el algoritmo es otro factor que se ve influido a causa del valor que se le asigne al peso de

inercia, debido a que el peso de inercia conlleva una mayor o menor dinámica en la actualización de velocidad y posición para cada partícula. Esto ocasiona que el enjambre converja con mayor rapidez para los casos en donde el peso de inercia es menor a 0,5) debido a que es más fácil que las partículas se acumulen en un punto dado que su avance es moderado. El caso contrario ocurre para un valor en el peso de inercia mayor a 0,5 donde la velocidad de convergencia disminuye a causa de que toma más tiempo esperar a que las partículas se acumulen, dado que el avance de cada partícula es brusco (pasos largos). Un ejemplo que permite verificar lo expuesto anteriormente se presenta en las tablas 3, 4 y 6 donde se puede evidenciar el número de iteraciones gastado en la búsqueda del mínimo global de la función (Ackley) para distintos valores de w .

Tabla 3: Resultados para la función Ackley con $w = 0, 1$.

Parámetros	k	pg_x	pg_y	$f(x, y)$
$np = 30$	23	0	0	-2,7183
$c1 = 1,9$	18	0	0	-2,7182
$c2 = 1,8$	17	0,0006	-0,0010	-2,7150
$w = 0,1$	17	-0,0158	0,0174	-2,6371
$\epsilon_1 = 0,1$	18	0	0	-2,7183
$\epsilon_2 = 0,1$	17	0	0	-2,7183
$dt = 0,5$	16	0	0	-2,7183
$pg_x = 0,05$	17	0	0	-2,7182
$pg_y = 0,05$	19	0	0	-2,7183
$f(x, y)_{teorico} = -2,385$	18	0	0	-2,7183
<i>Promedio</i>	18			

Tabla 4: Resultados para la función Ackley con $w = 0, 3$.

Parámetros	k	pg_x	pg_y	$f(x, y)$
$np = 30$	17	0	0,0001	-2,7175
$c1 = 1,9$	10	0,0002	0	-2,7182
$c2 = 1,8$	19	0	0	-2,7182
$w = 0,3$	21	0	0	-2,7182
$\epsilon_1 = 0,1$	18	0	0	-2,7182
$\epsilon_2 = 0,1$	18	0	0,0001	-2,7179
$dt = 0,5$	23	0	0	-2,7183
$pg_x = 0,05$	17	0	0	-2,7180
$pg_y = 0,05$	18	0	0	-2,7182
$f(x, y)_{teorico} = -2,385$	21	0	0	-2,7183
<i>Promedio</i>	19,2			

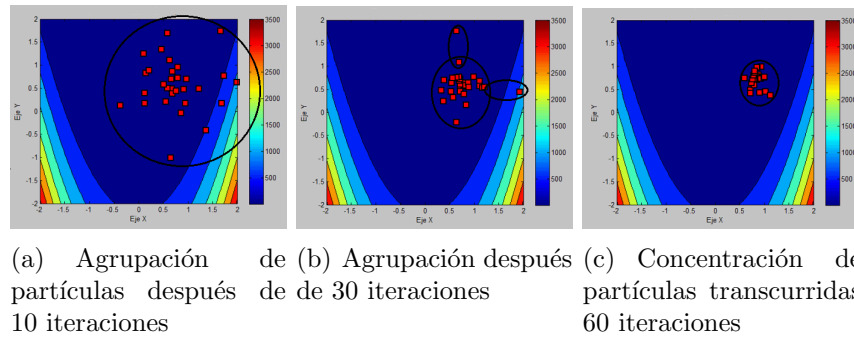
Variación en los valores para las constantes exploratorias Otro factor que influye en el comportamiento del enjambre, corresponde a la asignación de valores para

Tabla 5: Resultados para la función Ackley con $w = 0,5$.

Parámetros	k	pg_x	pg_y	$f(x, y)$
$np = 30$	25	0	0	-2,7181
$c1 = 1,9$	28	0	0	-2,7182
$c2 = 1,8$	24	0,0006	0	-2,7182
$w = 0,5$	26	0	0	-2,7182
$\epsilon1 = 0,1$	27	0	0	-2,7181
$\epsilon2 = 0,1$	27	0	0	-2,7180
$dt = 0,5$	29	0	0	-2,7183
$pg_x = 0,05$	26	0	0	-2,7179
$pg_y = 0,05$	25	0	0	-2,7180
$f(x, y)_{teorico} = -2,385$	30	0	0	-2,7182
Promedio	26,7			

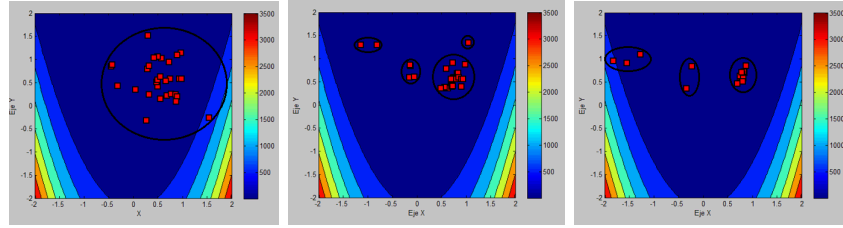
las constantes de exploración local y global. Un mayor peso a la constante de exploración local ($c1$) con respecto a la exploración global ($c2$), ocasiona que las partículas se concentren en el proceso de exploración local. Un mayor peso a la constante de exploración global ($c2$) con respecto a la constante de exploración local ($c1$), permite que las partículas exploren un área mayor, favoreciendo de esta manera la búsqueda global. Un ejemplo que permite observar lo expuesto anteriormente, se puede evidenciar en las figuras 32, 33 y 34 donde se presenta la relación que tienen los distintos valores de las componentes de exploración local y global con el tipo de exploración llevado a cabo por el enjambre.

Figura 32: Agrupación del enjambre para $c1 = 2,1$ y $c2 = 1,0$.



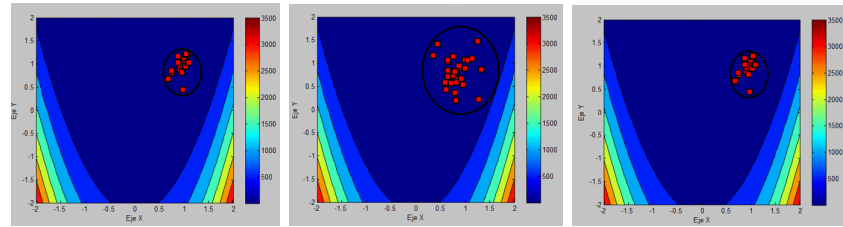
El número de iteraciones también es otro factor que afecta el desempeño del algoritmo a causa de la variación de los valores en las constantes de exploración local y global. Asignar mayor valor a la constante de exploración local tiene como consecuencia un gasto mínimo de 16 iteraciones. En caso de asignar un mayor valor a la constante de exploración global, el número de iteraciones aumenta hasta 23 iteraciones. Estas observaciones se obtuvieron teniendo en cuenta que el radio de avance asignado corresponde a un valor de $dt = 0,5$. Un ejemplo que relaciona lo expuesto con anterioridad, se

Figura 33: Agrupación del enjambre para $c1 = 1,0$ y $c2 = 2,1$.



(a) Concentración de partículas después de 12 iteraciones (b) Concentración de partículas después de 35 iteraciones (c) Concentración de partículas al finalizar 55 iteraciones

Figura 34: Agrupación del enjambre para $c1 = 1,8$ y $c2 = 2,1$.



(a) Exploración de partículas después de 7 iteraciones (b) Exploración alcanzada tras 17 iteraciones (c) Concentración de partículas al finalizar 36 iteraciones

presenta en las tablas 6 y 7 donde se evidencia el cambio en el número de iteraciones en relación con el peso asignado para las constantes de exploración.

Tabla 6: Resultados para la función Ackley con $c1 = 1,9$ y $c2 = 1,8$.

Parámetros	k	pg_x	pg_y	$f(x, y)$
$np = 30$	23	0	0	-2,7183
$c1 = 1,9$	18	0	0	-2,7182
$c2 = 1,8$	17	0,0006	-0,0010	-2,7150
$w = 0,1$	17	-0,0158	0,0174	-2,6371
$\epsilon1 = 0,1$	18	0	0	-2,7183
$\epsilon2 = 0,1$	17	0	0	-2,7183
$dt = 0,5$	16	0	0	-2,7183
$pg_x = 0,05$	17	0	0	-2,7182
$pg_y = 0,05$	19	0	0	-2,7183
$f(x, y)_{teorico} = -2,385$	18	0	0	-2,7183
<i>Promedio</i>	18			

De manera general, se realizaron distintas pruebas para cada una de las funciones que consistieron en variar un parámetro a la vez dejando los demás fijos según las indicaciones de valores estándar para cada uno de ellos. Estos valores recomendados o

Tabla 7: Resultados para la función Ackleys con $c1 = 1,8$ y $c2 = 1,9$.

Parámetros	k	pg_x	pg_y	$f(x, y)$
$np = 30$	15	0	0	-2,7182
$c1 = 1,8$	16	0	0	-2,7183
$c2 = 1,9$	16	0	0	-2,7182
$w = 0,1$	16	0	0	-2,7182
$\epsilon_1 = 0,1$	18	0	0	-2,7182
$\epsilon_2 = 0,1$	15	0	0	-2,7183
$dt = 0,5$	17	0	0	-2,7183
$pg_x = 0,05$	16	0	0	-2,7183
$pg_y = 0,05$	16	0	0	-2,7182
$f(x, y)_{teorico} = -2,385$	17		0	-2,7183
<i>Promedio</i>	16,2			

estándar, corresponden a un número de partículas de 30 ($ns = 30$), constantes de actualización local y global de 1,9 y 1,8 respectivamente ($c1 = 1,8$ y $c2 = 1,9$) y el valor de ϵ_1 y ϵ_2 de 0,1 para ambos ($\epsilon_1 = 0,1$ y $\epsilon_2 = 0,1$). El peso de inercia se varió iniciando en un valor de 0,1 hasta 0,9 con pasos de 0,2, después esto se eligió el mejor valor de w para fijarlo e iniciar las siguientes pruebas en las que se varió las constantes de actualización global y local en el rango de 1 a 3 en pasos de 0,3. El contenido de estas pruebas se puede observar con detalle en la carpeta de Tablas en el CD de anexo.

A partir de las observaciones con respecto al comportamiento del algoritmo de optimización mediante enjambre de partículas por medio de las implementaciones y el análisis de sus resultados, se sugiere que el número de partículas sea de 30 puesto que ubicar más partículas dentro del espacio mejora la búsqueda del mínimo global. De igual manera, el número de iteraciones no debe ser un límite que interrumpa el avance de las partículas, en ese sentido se propone que el valor de este parámetro sea mayor a 20 iteraciones teniendo en cuenta que se desconoce el momento en que el enjambre convergerá al mínimo global. Por esta razón se hizo necesario el uso de los criterios de restitución de partículas y de parada que garantizaran que la convergencia se facilitara. En cuanto al valor del peso de inercia y las constantes exploratorias, se presentaron distintos valores para cada función de prueba, sin embargo los valores más destacados correspondieron a un peso de inercia de 0,5 ($w=0,5$) y constantes globales y locales en el rango de 0,1 a 2,6. La variación de estos parámetros influye en el comportamiento del enjambre ante distintas funciones de prueba permitiendo observar la conformación de entornos geográficos y sociales, así como su influencia en la exploración local y global. Reconociendo que estos deben ser diferentes según la complejidad de la función objetivo.

El valor en el delta de tiempo que arrojó mejores resultados para todas las funciones

en pruebas preliminares fue de 0,5 ($dt = 0,5$). Por esta razón, se dejó como un valor fijo en la realización de las pruebas. Teniendo en cuenta lo anterior se decidió fijar de igual manera los ϵ de los criterios de parada con el valor de 0,1 manteniendo el mismo orden.

Los resultados obtenidos se almacenaron en tablas que se pueden observar en el la carpeta Tablas del CD de anexo. En la tabla 8 se presentan los mejores resultados obtenidos para la implementación del algoritmo PSO, recordando que en las diferentes pruebas aplicadas al algoritmo se variaron parámetros como el número de partículas (np), el peso de inercia (w), y las constantes de exploración local y global $c1$ y $c2$ respectivamente.

Tabla 8: Parámetros bajo los cuales se obtuvieron los mejores resultados en la implementación de PSO.

Función	w	$c1$	$c2$
Esfera	0,2	1,9	1,8
Ackley	0,1	1,8	2,4
Goldstein Price	0,3	1,9	1,8
Holder Table	0,7	1,0	1,8
Schwefel	0,5	2,2	2,2
Step	0,1	1,8	2,1
Stochastic	0,1	2,6	2,6
Básica	0,5	1,0	1,8
Beale	0,5	1,8	1,9
Eggholder	0,7	1,8	2,1
Easom	0,1	1,0	1,8
Rastrigin	0,7	1,0	1,8
Rosenbrock	0,5	1,8	2,1

3.1.9. Ventajas del algoritmo

- Facilidad de interpretación y comprensión del algoritmo en vista de que es bioinspirado.
- Facilidad en la implementación dado que se manejan pocos parámetros y tan solo existen dos ecuaciones de actualización, que corresponden a la velocidad y posición de cada partícula.
- Permite el ajuste directo en la característica de la búsqueda que se quiere realizar, por medio de la asignación de valores en las constantes $c1$ y $c2$.
- Por ser un algoritmo de enjambre, facilita la visualización del comportamiento del mismo frente a los retos que una función le puede ofrecer.

- Favorece la búsqueda global en vista de que maneja un gran número de partículas comunicadas entre sí y que a su vez brindan información por separado.

3.1.10. Desventajas del algoritmo

- En vista de que el comportamiento del enjambre es sensible a cambios en los valores de los parámetros, el mal manejo de estos valores puede ocasionar una convergencia prematura.
- En el caso de funciones multivariable, se hace necesario conocer el rango del espacio de solución de modo que el radio de avance que experimenta cada partícula esté dentro de ese espacio. De lo contrario, la búsqueda no se encaminará dentro de las fronteras sino fuera de ellas.

Los detalles de implementación del algoritmo en Matlab se pueden ver completamente en las memorias del seminario así como las características y resultados del mismo aplicado a todas las funciones de prueba. La realización de todas las pruebas permitieron discutir los distintos aspectos que se presentan a continuación.

3.1.11. Discusión Durante el desarrollo del seminario de investigación, se estudió el desempeño del algoritmo de optimización por enjambre de partículas. Para cumplir con este objetivo se propuso un esquema de pruebas en que se modificó un parámetro cada vez para comprender cómo influye cada uno en el desempeño del algoritmo sobre funciones multivariable. A partir de estas pruebas se establece que para los efectos de este seminario, cada uno de los parámetros tiene un intervalo de operación adecuado para las funciones multivariable. En este contexto el rango de valores adecuado para el peso de inercia w oscila entre 0,1 y 0,7. Los valores adecuados para las constantes de exploración global y local $c1$ y $c2$ corresponden al intervalo comprendido entre 1,0 y 2,6, sin embargo, es importante aclarar que a partir de las pruebas realizadas, los mejores valores para $c1$ y $c2$, corresponden a 1,8 y 1,9 respectivamente. Por otra parte el radio de avance oscila entre 0,3 y 0,7. Finalmente, los valores de ϵ_1 y ϵ_2 se mantuvieron iguales en un valor de 0,1.

Los valores encontrados para estos parámetros durante el desarrollo del seminario coinciden con los valores registrados en los trabajos de investigación realizados en la Universidad Industrial de Santander para el algoritmo PSO [2], [1], [9]. Sin embargo, la amplitud del rango encontrado en el seminario de investigación es mayor en vista de que se realizaron pruebas diferentes a las encontradas en la literatura expuesta en [2], [1], [9]. Así mismo, se consideraron más funciones con diferentes retos sobre las cuales se implementó el algoritmo por lo que algunos parámetros presentaron diferencias en relación con los obtenidos en los trabajos ya mencionados.

A partir de los resultados registrados para el esquema de pruebas propuesto, se evidencia que el peso de inercia es un factor que influye en la capacidad de exploración de las partículas, en la cantidad de iteraciones necesarias para la obtención del mínimo global y en la velocidad con que las partículas se mueven dentro del espacio solución. En algunas de las pruebas realizadas se comprobó que tal y como se menciona en [2], [1], [9], el aumento en el peso de inercia conlleva un aumento en el número de iteraciones. Otro parámetro que afecta la velocidad con que las partículas se desplazan en el espacio solución es el radio de avance. De acuerdo con los resultados obtenidos de las ejecuciones del algoritmo, se recomienda usar un valor de $dt = 0,5$ que permita a las partículas realizar una actualización equilibrada con respecto al área explorada del espacio solución.

En cuanto a las constantes $c1$ y $c2$, los resultados de las pruebas efectuadas indican que éstas constantes afectan la comunicación entre partículas y por ende, se afecta la orientación de las mismas hacia la obtención del mínimo global. En ese sentido, La conclusión a partir de los resultados obtenidos apunta a que es mejor asignar un mayor peso a la búsqueda global con respecto de la búsqueda local, con el objeto de reconocer el espacio de solución y permitir una búsqueda de largo alcance.

3.2. OPTIMIZACIÓN MEDIANTE ALGORITMOS GENÉTICOS (GA)

Es una técnica metaheurística que utiliza un conjunto de individuos representados mediante un esquema de codificación. Este esquema permite agrupar una lista de características de cada individuo, en una cadena que representa mediante parámetros estas características. Los individuos son seleccionados de acuerdo a su mayor nivel de aptitud mediante el uso de una función de adaptación. Esta función determina qué modelos presentan las mejores características según el requerimiento del problema bajo consideración. Una vez se han escogido los pares de mejores individuos se aplican las operaciones que facilitan el surgimiento de nuevos individuos cuyos valores de aptitud, sean mejores que los obtenidos por sus antecesores. Aquellos individuos cuyos valores de aptitud se consideren bajos, son descartados definitivamente, de acuerdo con lo consultado en [8].

3.2.1. Analogía con la naturaleza Todos los seres vivos están conformados en su estructura mínima por células dentro de las cuales se encuentra la unidad básica de transferencia de información genética, denominada cromosoma. Un cromosoma internamente está conformado por cadenas de ácido desoxirribonucleico (DNA, por sus siglas en inglés), las cuales contienen la información necesaria que determina el conjunto de características diferenciadoras entre los seres vivos. Los seres vivos en este método de optimización se denominan individuos y constituyen un conjunto de posibles soluciones al proceso bajo consideración. Cada individuo se identifica mediante el uso de

una cadena de caracteres binarios denominada cromosoma donde las combinaciones de dígitos binarios determinan las características de cada uno y por ende su adaptación al medio conocido como espacio de solución [8].

Las iteraciones del algoritmo son denominadas generaciones, durante cada una de estas se lanza un grupo de individuos que son evaluados por la función de adaptación. Los mejores calificados son seleccionados para mejorar la población en generaciones futuras, mediante operadores de cruce. Los de calificación más baja, sufren un proceso de cambio donde se modifican parcialmente sus cromosomas. Finalmente, los individuos inadaptados son remplazados en la siguiente generación por otros con mejores características, garantizando de esta manera una evolución progresiva de acuerdo a la función objetivo.

Al igual que en PSO, el algoritmo toma cada individuo como un punto ubicado en un espacio de D-dimensiones donde cada punto es una posible solución de la función objetivo. Como parámetros del algoritmo, se destaca el número de individuos que conforman la población en cada generación, la tasa de cruce, la tasa de mutación y el número de bits que conforman la cadena binaria de cada individuo.

Cantidad de individuos: Es el tamaño de la población, conformada por individuos, asignada para una generación.

Tasa de cruce: Es la probabilidad de cruce asignada para que se cumpla el proceso de creación de nuevos individuos, a partir de combinaciones de las cadenas binarias de los mejores individuos en cada generación.

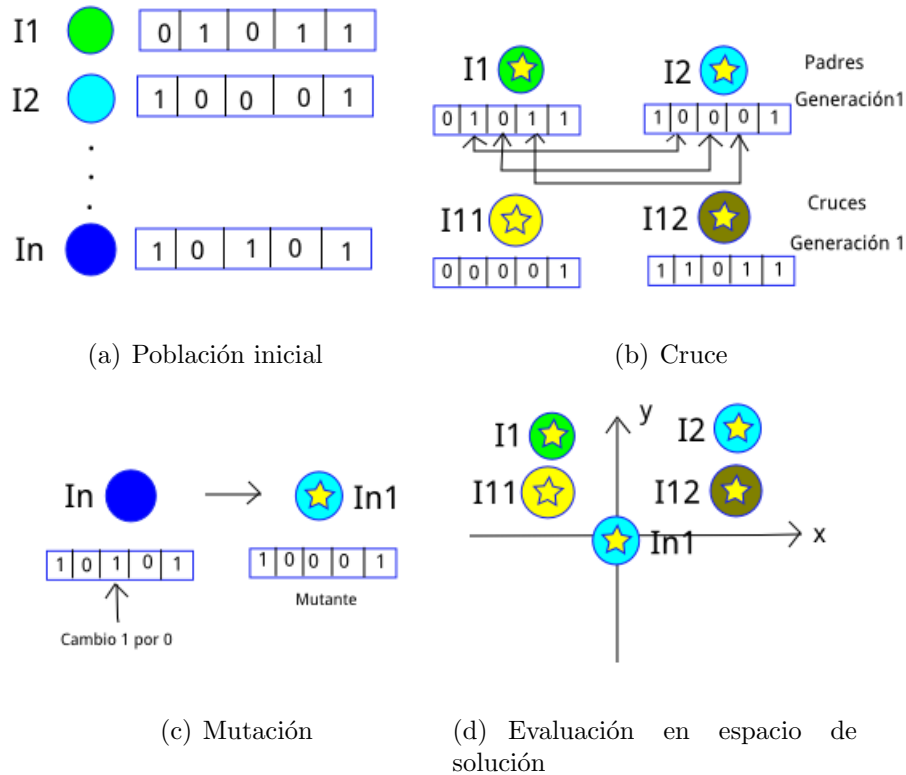
Tasa de mutación: Es la probabilidad de mutación asignada para que se cumpla el proceso de creación de nuevos individuos, a partir del cambio aleatorio de bits que conforman la cadena binaria de cada individuo.

Número de bits: Corresponde al tamaño que representara cada dimensión de la función objetivo. El número de bits multiplicado por las dimensiones conforma la cadena binaria de cada individuo.

Para comprender el comportamiento de los individuos considere el esquema que se muestra en la figura. En la figura 35(a), se muestra el listado de individuos que conforman la población inicial, los individuos identificados con la estrella amarilla presentan mejor aptitud. En la figura 35(b), se muestra el proceso de cruce que permite obtener mejores individuos. El proceso de cruce que transforma un individuo no apto en uno mejor adaptado se muestra en la figura 35(c). Finalmente, los mejores individuos se ubicarán en zonas próximas al mínimo global, mientras los menos aptos son descartados, como

lo muestra la figura 35(d).

Figura 35: Comportamiento de los individuos.



3.2.2. Pseudocódigo GA Describe los pasos necesarios para la implementación de un algoritmo genético en funciones multivariable. En su estructura, en el algoritmo inicialmente se define el criterio de parada y el espacio de solución delimitado según el tipo de función. Una vez definidos los parámetros requeridos, se realiza el procedimiento conforme se muestra en Algorithm 2.

3.2.3. Implementación del algoritmo La implementación realizada mediante el software Matlab, se basa en la estructura mostrada en el algoritmo de GA. Adicionalmente se define e implementa el criterio de parada que detiene el algoritmo cuando la posición del mejor individuo se repite en un promedio de 15 veces. El código implementado, las funciones que se requieren para su funcionamiento y el criterio de parada, se encuentran en la sección 6.

3.2.4. Evolución de individuos La función de adaptación evalúa a cada individuo de la generación anterior, determinando su nivel de aptitud para seleccionar a los dos

```

Data: ni, Tc, Tm, nbits
Result: g
1 Inicio;
  /* Definición límite repetición del mejor individuo */
2 lmejor;
  /* Delimitación fronteras espacio de solución */
3 Se definen las fronteras de la región solución;
  /* Función objetivo */
4  $F(x) = \sum_{i=1}^n X_i^2$ ;
  /* Creación y codificación de la población inicial */
5 Las características de cada individuo se codifican en formato binario
  aleatoriamente;
  /* Criterio de Parada */
6 while limit<=lmejor do
  /* Evaluación de individuos en la función objetivo */
7   for j=1:ni do
8     | Evaluar cada individuo en la función objetivo
9   end
10  Seleccionar el mejor individuo de la generación;
11  if mejor individuo se repite then
12    | limit=limit+1;
13  else
14    | limit=0
15  end
  /* Creación de nuevos individuos */
16  Seleccionar los mejores individuos de acuerdo a la función de adaptación;
17  for i=1:ni/2 do
18    | Operación de cruce de las características de los pares de mejores
    | individuos según la tasa de cruce;
19  end
20  Operación de mutación de las características de los pares de mejores
    individuos según la tasa de mutación;
21  Entrega de la nueva población;
22 end
23 Mínimo global = g

```

Algorithm 2: Algoritmo Genético.

Tabla 9: Resultados para la función Beale.

Parámetros	k	pg_x	pg_y	$f(x, y)$
$ni = 10$	22	-4,284	1,3073	7,7453
$Tc = 1,0$	22	-0,5586	1,7891	2,1639
$Tm = 0,01$	22	3,221	0,3159	1,1612
$nbits = 16$	34	2,022	-0,0762	0,8769
$pg_x = 3$	23	-1,8425	1,5357	5,1921
$pg_y = 0,5$	22	1,4083	-0,8863	5,1873
$f(x, y)_{teorico}$	22	-0,7635	1,7586	1,857
	99	4,4614	0,3842	6,4748
	22	-0,8157	1,3493	6,0386
	27	3,7134	0,4298	1,6155
Promedio	31,5			

mejores individuos. Una vez escogidos los individuos caracterizados por su mejor aptitud, se someten a los procesos de cruce y mutación. Estos procesos producen una nueva generación de individuos que exploran nuevos puntos dentro del espacio de solución. La nueva generación de individuos repite el ciclo de evaluación de aptitud, selección, cruce y mutación mientras que aquellos menos aptos de la generación anterior, se descartan.

3.2.5. Implementación sobre funciones multivariable Los resultados de la implementación de algoritmos genéticos en funciones multivariable, se registraron en una hoja de cálculo de Excel. En la parte lateral izquierda de éstas hojas de cálculo se muestran los parámetros modificados en cada experimento y en la parte lateral derecha se muestra el número de iteraciones (k) y las coordenadas del mínimo global encontrado en cada una de las 10 ejecuciones del algoritmo para cada experimento. Un ejemplo donde se muestra el registro de datos para cada experimento se observa en la tabla 9.

La implementación de algoritmos genéticos en funciones multivariable se hizo de manera similar a como se implementó el algoritmo PSO, conservando la metodología de ir variando un parámetro a la vez mientras que los demás se mantenían constantes en valores inicialmente recomendados. Estos valores recomendados en parámetros como: cantidad de individuos en una generación, tasa de cruce y tasa de mutación, corresponden a 20, 1, 0,01 ($ni = 20, Tc = 1,0, Tm = 0,01$) respectivamente. El número de bits para definir cada dimensión de la función objetivo fue de 16 ($nbits = 16$).

La razón por la cual los valores recomendados corresponden a esos valores, se debe a que en algoritmos genéticos es necesario contar con un tamaño de población grande que permita obtener una mayor información del espacio de solución. Con respecto a la tasa de cruce, es común observar en la naturaleza cómo las nuevas generaciones conservan rasgos característicos de generaciones pasadas permitiendo de esta manera un proceso evolutivo en varios rasgos de estas generaciones, por esta razón la tasa de cruce es alta.

Para la tasa de mutación, se considera que es necesaria tenerla en cuenta pero como un caso aislado, esto se debe a que en la naturaleza el fenómeno de mutación es poco común. A pesar de esto, siempre ocurre y en ocasiones puede ser punto clave para permitir una evolución que quizás no se dio en los demás individuos de la población, por esta razón la tasa de mutación es baja.

La implementación del algoritmo GA sobre funciones multivariable, se realiza en solo 8 funciones, dado que hay algunas de estas que se pueden obviar debido a la gran similitud con otras funciones en aspectos de forma y retos por ofrecer al algoritmo. La implementación dio lugar a una serie de pruebas donde ocurrieron variaciones de parámetros de la siguiente manera: La cantidad de población (ni) se varió para tamaños de 10, 20 y 30 individuos, La tasa de cruce (Tc) se varió en el rango de 0,3 a 1,0 en pasos de 0,2 y la tasa de mutación (Tm) dentro del rango $[0,01, 0,7]$ con valores intermedios de 0,1 y 0,5. Para obtener mayor información de las características y retos acerca de las funciones de prueba sobre las cuales se implementó el algoritmo, se recomienda remitirse a la sección 3.1.6 donde se explica con detalle cada uno de estos aspectos.

Los aspectos de interés presentados en cada una de las pruebas realizadas, se presentan con detalle en la memoria 8 6 y de igual forma en la recopilación de los datos tabulados ubicados en la carpeta Tablas del CD de anexo. Sin embargo, el verdadero interés encaminado a la información que se necesita adquirir por parte de los algoritmos genéticos con respecto a la implementación de funciones de prueba, se presenta a continuación. En primera medida con la presentación del análisis de resultados de las distintas pruebas hechas y en segunda medida, con la presentación de las discusiones que surgieron a partir de la observación de los resultados obtenidos.

3.2.6. Análisis de resultados Tal y como se explicó en la evolución de los individuos, el análisis de la variación de parámetros como el número de individuos y las tasas de cruce y mutación, no representan un impacto en el comportamiento que presentan los individuos a lo largo de las generaciones. Por tal razón, el análisis de la variación de estos parámetros se enfoca en la precisión del mínimo global de la función objetivo encontrado y en las iteraciones gastadas para encontrar dicho punto.

El computador utilizado para realizar las pruebas concernientes a la implementación del algoritmo de optimización global GA en funciones multivariable, contiene las siguientes especificaciones:

- Procesador Intel(R) Core i3-2310M CPU @ 2,10GHz
- Memoria RAM DDR3 de 2 GB

- Sistema operativo Ubuntu 12,04 de 32 bits

A continuación se presenta el análisis de los resultados obtenidos a partir de la implementación sobre las funciones multivariadas, junto con la descripción de los factores influyentes en la evolución de los individuos a lo largo de sus generaciones.

Tamaño de la población La asignación de un valor adecuado para el tamaño de la población, se verificó por medio de pruebas donde se evaluaba el impacto que tiene el aumento del tamaño de la población para las distintas funciones en las que se implementó el algoritmo. Los tamaños asignados en las pruebas fueron de 10, 20 y 30 individuos, con el fin de poderlos comparar con el tamaño asignado para el enjambre de partículas en PSO.

La manera en como se eligió qué tamaño de la población es mejor para el algoritmo, se hizo teniendo en cuenta el promedio en el número de iteraciones, tomando a consideración las veces en las que el algoritmo acertó la búsqueda del mínimo global. Con base en los resultados observados, se tiene que un tamaño de 20 a 30 individuos por generación es propicio para la obtención de buenos resultados. En este punto cabe aclarar que el tamaño de la población es constante en cada generación; es necesario hacer esta aclaración, debido a que es posible considerar una población creciente en vista de que es un algoritmo basado en la evolución de las especies generación tras generación. En las tablas [10 - 15] se presentan ejemplos del impacto que tiene el tamaño de la población en el gasto de iteraciones y en la precisión del mínimo global encontrado.

En las tablas [10 - 12], se puede observar una leve disminución en el número de iteraciones para los casos donde el tamaño de la población aumenta, tal y como se encontró en [7]. El número de iteraciones gastado para encontrar el mínimo global en la función Ackley, disminuye de 37,6 a 29,8 al presentar un aumento en la población de 10 a 30 individuos. Este comportamiento es análogo al presentado en PSO, debido a que una mayor población permite contar con nuevos individuos que aportan en el reconocimiento del espacio de solución permitiendo ahorrar zonas de exploración en generaciones futuras y de igual manera aumentar la probabilidad de que un individuo más apto se cree.

En las tablas [13 - 15] se puede observar como en cierto modo mejora la precisión del mínimo global encontrado, a medida que el tamaño de la población va aumentando. En este punto cabe aclarar y recordar que la elección de los mejores parámetros no se fijo por medio del promedio de los mismos en comparación con los esperados, sino que se tubo en cuenta la cantidad de aciertos y cercanía con los valores de referencia descartando los eventos erróneos.

Tabla 10: Resultados para la función Ackley con $ni=10$.

Parámetros	k	pg_x	pg_y	$f(x, y)$
$ni = 10$	30	-0,2642	-0,2222	-0,09
$Tc = 1,0$	47	-0,3561	-0,9562	1,4996
$Tm = 0,01$	41	-0,6518	0,0419	0,5512
$nbits = 16$	22	0,7925	0,0723	0,3405
$pg_x = 0,05$	54	0,9742	1,1765	1,8393
$pg_y = 0,05$	27	0,7169	-0,9133	1,6452
$f(x, y)_{teorico} = -2,385$	22	0,0285	-0,6753	0,5187
	49	-1,0067	0,0904	0,1521
	41	-0,8042	-0,0556	0,2644
	43	1,0302	0,2071	0,8954
Promedio	37,6			

Tabla 11: Resultados para la función Ackley con $ni=20$.

Parámetros	k	pg_x	pg_y	$f(x, y)$
$ni = 20$	88	-0,1675	-0,9489	0,4887
$Tc = 1,0$	22	0,9588	-0,8489	1,1435
$Tm = 0,01$	22	0,2831	-0,0321	-0,6819
$nbits = 16$	26	0,5004	-0,1168	0,5227
$pg_x = 0,05$	38	0,2168	-0,501	0,8132
$pg_y = 0,05$	49	-0,8767	0,2692	1,087
$f(x, y)_{teorico} = -2,385$	23	0,5453	1,1828	2,6049
	33	-0,805	-0,0446	0,24
	23	0,1092	0,1258	-1,6275
	22	0,2381	0,9312	0,9075
Promedio	34,6			

Tasa de cruce De acuerdo con las recomendaciones pertinentes para la tasa de cruce, se sabe que a este parámetro se le debe asignar una alta probabilidad que permita garantizar por lo menos que la mitad de la nueva población esté constituida a partir del cruce genético de los cromosomas de los mejores individuos en la población anterior. Teniendo en cuenta esto, se corrobora que la asignación de la tasa de cruce para las funciones donde se implementó el algoritmo apunta a valores altos $[0,7-1,0]$ con la excepción para la función Rastriging donde la mejor tasa de cruce fue de 0,5. La tabla 16 presenta el mejor conjunto de parámetros para las distintas funciones multivariantes sobre las que se implementó el algoritmo.

La tasa de cruce influye en el número de iteraciones gastadas para la obtención del mínimo global [8]. En ese sentido, se entiende que esto ocurre debido a que si se toma una tasa de cruce baja, se estaría restringiendo en la creación de nuevos individuos que hereden algunas características de los padres a cambio de la conservación de individuos que nunca se adapten. Un ejemplo que permite ver el aumento de las iteraciones a

Tabla 12: Resultados para la función Ackley con $ni=30$.

Parámetros	k	pg_x	pg_y	$zf(x, y)$
$ni = 30$	35	0,1324	0,0387	-1,889
$Tc = 1,0$	26	-0,1266	-0,3143	-0,2304
$Tm = 0,01$	24	-1,0836	0,0446	0,352
$nbits = 16$	37	0,2494	-0,2186	-0,1887
$pg_x = 0,05$	23	0,3928	-1,0562	1,8648
$pg_y = 0,05$	26	0,227	0,2302	-0,2498
$f(x, y)_{teorico} = -2,385$	22	0,4771	0,1247	0,4785
	42	0,0378	-0,2105	-1,2424
	32	-0,8761	0,024	-0,0089
	31	0,3136	-0,3083	0,5179
Promedio	29,8			

Tabla 13: Resultados para la función Básica con $ni=10$.

Parámetros	k	pg_x	pg_y	$f(x, y)$
$ni = 10$	101	0	1,5217	-3,2020541
$Tc = 1,0$	128	0	1,5195	-3,2021025
$Tm = 0,01$	68	0	1,5708	-3,1617816
$nbits = 16$	22	1,2496	0,0389	-0,4955543
$pg_x = 0$	25	0,2367	1,342	-2,2488211
$pg_y = 1,5199$	128	0	1,5708	-3,1617816
$f(x, y)_{teorico} = -3,202$	121	0	1,524	-3,2018394
	141	1,5712	3,1415	-0,3158817
	86	0,0008	1,5708	-3,1596759
	225	0	1,5724	-3,1593411
Promedio	104,5			

Tabla 14: Resultados para la función Básica con $ni=20$.

Parámetros	k	pg_x	pg_y	$f(x, y)$
$ni = 20$	68	0,0031	1,5217	-3,1934
$Tc = 1,0$	42	0,0062	1,5491	-3,1714
$Tm = 0,01$	84	0	1,5199	-3,2021
$nbits = 16$	117	0	1,52	-3,2021
$pg_x = 0$	23	0,8853	1,3179	-1,4761
$pg_y = 1,5199$	29	0,0732	1,9341	-1,1061
$f(x, y)_{teorico} = -3, 2021$	80	0,0004	1,5708	-3,1607
	86	0	1,5708	-3,1618
	103	0	1,5708	-3,1618
	22	1,1521	0,794	-1,0019
Promedio	65,4			

Tabla 15: Resultados para la función Básica con $ni=30$.

Parámetros	k	pg_x	pg_y	$f(x, y)$
$ni = 30$	65	0	1,5219	-3,202
$Tc = 1,0$	114	0	1,52	-3,2021
$Tm = 0,01$	105	0	1,5708	-3,1618
$nbits = 16$	122	0	1,5199	-3,2021
$pg_x = 3$	73	0	3,5339	-0,6058
$pg_y = 0,5$	72	0,0001	1,5218	-3,2018
$f(x, y)_{teorico} = 0$	43	0,1313	1,4397	-2,7638
	40	0,0058	1,4399	-3,0822
	70	0,0499	1,5663	-3,0333
	61	0,0081	1,4555	-3,112
Promedio	76,5			

Tabla 16: Parámetros bajo los cuales se obtuvieron los mejores resultados en la implementación de GA.

Función	No.individuos	Tc	Tm
Esfera	20	1,0	0,5
Ackley	20	1,0	0,5
Holder Table	30	1,0	0,5
Schwefel	30	0,7	0,1
Básica	30	0,7	0,1
Beale	30	0,7	0,5
Eggholder	20	0,7	0,7
Rastrigin	30	0,5	0,1

medida que disminuye la tasa de cruce, se presenta en la tablas [17 - 20].

Tabla 17: Resultados para la función Eggholder con $Tc=1$.

Parámetros	k	pg_x	pg_y	$f(x, y)$
$ni = 30$	131	500,0552	400,2553	-900,6586
$Tc = 1,0$	22	-400,0119	-400,2533	-600,3831
$Tm = 0,01$	56	-300,7609	300,2923	-700,183
$nbits = 16$	85	500,078	400,1525	-900,6928
$pg_x = 512$	46	-400,9873	0,0562	-500,5103
$pg_y = 404,2319$	22	400,7986	-300,9883	-600,779
$f(x, y)_{teorico} = -959,6407$	33	400,2701	400,6325	-900,2711
	26	400,5494	400,4714	-900,4181
	109	-500,0741	400,1164	-900,5876
	35	500,04	300,7375	-800,0718
Promedio	56,5			

Tabla 18: Resultados para la función Eggholder con $Tc=0,7$.

Parámetros	k	pg_x	pg_y	$f(x, y)$
$ni = 30$	22	-300,1328	-400,6666	-500,1146
$Tc = 0,7$	56	500,0002	400,2	-900,6553
$Tm = 0,01$	22	400,1625	400,3928	-700,8677
$nbits = 16$	24	400,4412	400,1137	-800,1482
$pg_x = 512$	25	400,5794	-400,0048	-700,042
$pg_y = 404,2319$	71	300,1209	-400,9675	-700,5738
$f(x, y)_{teorico} = -959,6407$	79	-300,8183	300,5501	-700,533
	22	300,4917	500,1127	-800,6446
	136	-400,4669	300,7183	-800,6045
	42	-400,1678	300,6401	-800,2579
Promedio	49,9			

Tabla 19: Resultados para la función Eggholder con $Tc=0,5$.

Parámetros	k	pg_x	pg_y	$f(x, y)$
$ni = 30$	22	300,7073	400,7675	-800,1744
$Tc = 0,5$	27	-200,9392	400,9311	-700,4139
$Tm = 0,01$	77	-300,4003	-400,483	-700,1833
$nbits = 16$	42	300,7376	400,8541	-900,0258
$pg_x = 512$	58	400,94	-300,2109	-700,3531
$pg_y = 404,2319$	22	-400,2072	-300,9442	-700,0985
$f(x, y)_{teorico} = -959,6407$	115	500,1112	400,06	-900,6249
	118	300,8128	400,678	-800,9078
	61	-300,0651	500,12	-800,4416
	62	300,9653	400,8303	-900,0536
Promedio	60,4			

Tabla 20: Resultados para la función Eggholder con $Tc=0,3$.

Parámetros	k	pg_x	pg_y	$f(x, y)$
$ni = 30$	145	-300,6976	-400,3403	-700,3294
$Tc = 0,3$	65	300,8101	400,8	-900,0437
$Tm = 0,01$	65	-500,0361	400,0287	-900,5256
$nbits = 16$	50	-200,3792	400,8706	-700,1254
$pg_x = 512$	30	-400,3573	300,5801	-700,6962
$pg_y = 404,2319$	49	400,7331	400,2789	-900,4799
$f(x, y)_{teorico} = -959,6407$	56	-400,3219	-300,9908	-700,7109
	22	300,639	400,7455	-700,0725
	38	300,3308	400,8875	-800,5755
	128	500,0144	400,1517	-900,636
Promedio	64,8			

Tasa de mutación De acuerdo con las recomendaciones pertinentes para la tasa de mutación, se sabe que a este parámetro debe asignarse una baja probabilidad que permita garantizar por lo menos un caso en el que se cree un nuevo individuo ajeno a las características que le puedan heredar los padres [8]. A pesar de estas consideraciones, como se pudo observar en la tabla 16. Se tiene que para los distintos tipos de funciones multivariadas, fue necesario contar con una alta tasa de mutación que permitiera crear nuevos individuos ajenos a las características heredadas de sus padres de modo que brindaran una nueva alternativa para encontrar el mínimo global buscado. El rango adecuado para la tasa de mutación de acuerdo a los resultados obtenidos en la implementación sobre funciones multivariadas es de 0,1 a 0,5.

3.2.7. Ventajas del algoritmo

- Facilidad de interpretación y comprensión del algoritmo en vista de que es un algoritmo bioinspirado. En este sentido, es posible relacionar el funcionamiento del algoritmo genético con los procesos de cruce y mutación existentes en el proceso evolutivo que presentan las especies en la naturaleza.
- Facilidad en la implementación en vista de que el algoritmo cuenta con pocos parámetros. De esta manera, es posible realizar un análisis de la influencia que presenta en el algoritmo la variación del valor de los parámetros con pruebas que no demanden un alto costo computacional, como ocurriría en el caso de contar con varios parámetros.
- Permite el ajuste directo en la característica de la búsqueda que se quiere realizar (búsqueda global o local), por medio de la asignación de valores en las tasas de cruce Tc y tasas de mutación Tm .
- La cantidad de alternativas que se pueden utilizar para realizar el proceso de selección natural. Básicamente las diversas formas que se presentan en la naturaleza, pueden ser plasmadas en el algoritmo como por ejemplo: la selección de los mejores individuos por torneo, elitismo, nivel de adaptabilidad, entre otros.

3.2.8. Desventajas del algoritmo

- La pérdida de precisión en los datos de entrada cuando la conversión de números decimales a binarios no es exacta.
- La tendencia a converger prematuramente. A pesar de implementar criterios de parada que garantizaron la convergencia en un punto luego de que el algoritmo repitiera veinte veces el mejor individuo, se evidencia que los cambios entre el mejor individuo a través de las generaciones es mínimo por lo cual el algoritmo converge con rapidez.

- La sensibilidad que presenta la estructura del algoritmo ante los cambios que se puedan hacer en la implementación del proceso de selección natural. Las diferentes alternativas que se pueden implementar en este proceso de selección, podrían ocasionar una variación en la dinámica del algoritmo GA.

3.2.9. Discusión El aumento de la población permite una mayor información del espacio de solución y consigo, garantiza una mejor precisión con respecto al punto encontrado. Por otra parte, una mayor cantidad de individuos permite la disminución en el número de iteraciones, debido a que no es necesario tener que esperar varias generaciones para obtener un individuo en particular, tal y como ocurre en poblaciones pequeñas.

El número de iteraciones es un indicador por el que se pueden considerar un conjunto de valores en los parámetros, mejores que otros. A pesar de esto, no siempre la búsqueda de la disminución de estas iteraciones presenta buenos resultados. Un ejemplo ocurre para el caso donde se sabe que el aumento de la tasa de mutación disminuye el número de iteraciones. A partir de esto, hay que tener en cuenta que la elección de una tasa de mutación (Tm) alta, por ejemplo de 0,9, podría causar que la mayoría de los individuos muten todo el tiempo y se presente el caso de que nunca converjan en alguna solución.

La tasa de mutación tiene gran relación con el tipo de búsqueda que se le quiere dar al algoritmo. Un valor alto (mayor a 0,1) de esta permite que los individuos se dispersen de mejor manera en el espacio de solución, garantizando una mayor exploración. Un valor bajo (menor a 0,1), permite que la búsqueda sea más compacta pero a la vez más detallada (explotación). Se considera que no siempre se debe contar con una tasa de mutación baja (menor a 0,1), puesto que una tasa de mutación alta (mayor a 0,1) permite una búsqueda más general del espacio de solución. En ocasiones esto es necesario, para los casos donde los individuos demoran más de 100 generaciones evolucionando sin lograr llegar al mínimo de la función objetivo. La limitante en el número de iteraciones o generaciones, que para el tipo de pruebas realizadas en este seminario de investigación fue de 100 iteraciones, depende del tipo de prueba sobre la que se este utilizando el algoritmo.

Para todas las implementaciones del algoritmo sobre funciones multivariable, se tuvo la idea de que una tasa alta de cruce siempre es necesaria para garantizar el proceso evolutivo conservando las características de los mejores individuos y permitiendo orientar las generaciones hacia una mejor solución. A pesar de esto, se tiene la hipótesis que para funciones con un nivel de complejidad alto, es necesario que la tasa de cruce no sea tan cercana a 1, es decir, por lo menos en un rango de 0,7 a 0,9.

Es necesario tener en cuenta que la interpretación del algoritmo GA es muy diferente a PSO. teniendo en cuenta esto, es erróneo esperar un comportamiento en los individuos generación tras generación puesto que no se trata de un enjambre que se mueve de acuerdo al tipo de función sino a una serie de procesos de cruce y mutación donde los nuevos individuos a pesar de conservar las características de los padres, son nuevos.

4. INTEGRACIÓN DE ALGORITMOS DE OPTIMIZACIÓN EN EL PROCESO DE INVERSIÓN SÍSMICA

Hasta el momento el enfoque que se ha desarrollado con los algoritmos de optimización ha sido netamente de carácter estructural y funcional, con respecto a distintos tipos de funciones multivariable. En este capítulo se hace énfasis en la manera como se integró el proceso de optimización desarrollado mediante los algoritmos PSO y GA, para la solución del problema de inversión sísmica. Por tal razón es necesario explicar cómo se interpretó dicha problemática para afrontarla desde el punto de vista de optimización.

En este capítulo se explican de manera consecutiva los desarrollos realizados en el seminario de investigación, en cuanto al reconocimiento del software de procesamiento de datos sísmicos; el uso de las herramientas necesarias para realizar el empalme entre Matlab y las implementaciones de los algoritmos de optimización; el estudio de las métricas implementadas para la comparación de trazas sísmicas; y el análisis de los resultados obtenidos.

El computador utilizado para realizar las pruebas concernientes a la integración de los dos algoritmos de optimización global PSO y GA en el proceso de inversión sísmica, corresponde al minicomputador OSAKA, propiedad del grupo CPS de la Universidad Industrial de Santander. La razón por la cual se cambió de computador en este tipo de pruebas, se debe a que este equipo tiene un mejor rendimiento con respecto al computador utilizado para las pruebas sobre funciones multivariables. Las especificaciones de este equipo son:

- Procesador intel(R) Xeon(R) CPU ES-2609 @2,40 GHz
- Memoria RAM DDR3 de 8 GB
- Sistema operativo debian 8 de 64 bits

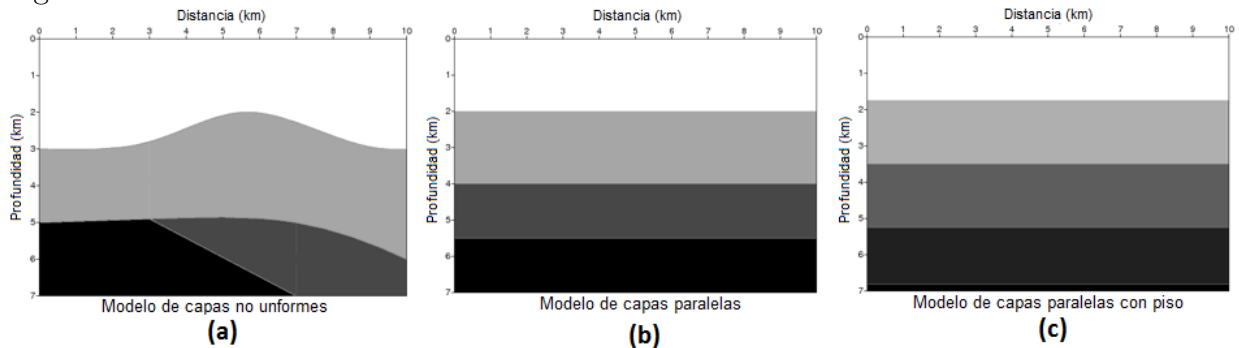
4.1. SEISMIC UNIX (SU)

SU es un paquete de aplicaciones para el tratamiento de señales sísmicas, apoyado por el centro de fenómenos ondulatorios de la Escuela de Minas de la Universidad de Colorado, en Estados Unidos [26]. Este software fue necesario en el desarrollo del seminario de investigación debido a que permite realizar, de manera simulada, los procesos que tienen lugar en el campo para la extracción de información acerca del terreno. El uso que se le dio a dicha herramienta fue la de crear un modelo del subsuelo de cuatro capas manteniendo una geometría fija, para así realizar la adquisición sísmica sobre dicho modelo, para luego obtener y visualizar las trazas sísmicas.

La explicación detallada del software SU, las herramientas que utiliza, los comandos para acceder a estas herramientas y las características de los archivos que se pueden generar, se encuentran con más detalle en la memoria 6 6.

Con el software SU se crearon los modelos del subsuelo, denominados modelos de referencia. Cabe aclarar que a lo largo del desarrollo del seminario de investigación se crearon 3 modelos de subsuelo. El primero de ellos con 4 capas no uniformes, el segundo con cuatro capas paralelas y el tercero con 5 capas paralelas, donde la última capa se tomaba como capa de piso. La razón por la cual se crearon distintos tipos de modelos de subsuelo, se hizo en primer lugar con el fin de facilitar la búsqueda que realizaban los algoritmos de optimización para obtener el valor del *sloth* en la capas del modelo de subsuelo. Por este motivo, se creó el modelo de capas paralelas. La razón por la cual se añadió una quinta capa haciendo la función de piso, se hizo con el fin de permitir la reflexión de los frentes de onda con la capa de piso permitiendo obtener información acerca de la cuarta capa del modelo de subsuelo, que para el caso de los modelos anteriores, no se podía obtener. La visualización de los modelos creados, se muestran en la figura 36.

Figura 36: Modelos creados.

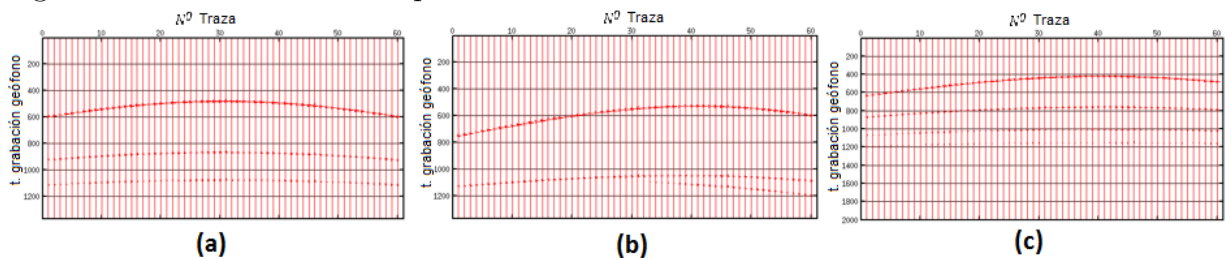


A partir de la creación de los modelos de subsuelo, se utilizó el software SU para simular el proceso de adquisición sísmica sobre los modelos creados. Para realizar dicho proceso se especificaron parámetros como: número de geófonos, espaciado entre geófonos, profundidad de los geófonos, rango de *offset* de los geófonos, número de disparos de la fuente, profundidad de la fuente, rango que emana la fuente y tiempo de muestreo. Todo esto a modo de que las trazas resultantes se asemejaran a un proceso, lo más real posible y que contenga la información necesaria para poder realizar la inversión sísmica que permita encontrar el modelo de referencia sobre el cual se hizo la adquisición. Cabe aclarar que la herramienta utilizada para realizar dicha adquisición fue la herramienta *triseis* que simula el proceso de adquisición de trazado de rayos. Otra técnica por la que también se podría realizar el proceso de adquisición por medio de SU, es a partir de la ecuación de onda. Sin embargo se decidió trabajar únicamente con trazado de

rayos, en vista de que es una herramienta más sencilla de trabajar.

Después de realizado el proceso de adquisición sísmica, se observaron las trazas sísmicas obtenidas por los geófonos virtuales como sismogramas. A partir de esto, se tiene la capacidad de comparar distintos conjuntos de trazas sísmicas dando lugar a las comparaciones que se realizan en la implementación de los algoritmos de optimización utilizando una figura de mérito conocida como métrica de comparación. Siendo posible la realización de procesos como la creación de los modelos, la adquisición sísmica de dichos modelos y la observación de las trazas sísmicas; además de contar con la posibilidad de manipular la información de trazas y realizar operaciones matemáticas como filtrado, escalamiento, sumatorias, y demás funciones que los comandos de SU permiten. Se tiene la limitante de solo contar con el proceso sísmico y las herramientas que SU permite utilizar, necesitando la manera de poder relacionarlos con el proceso de optimización a partir de los algoritmos de optimización y las herramientas que Matlab permite utilizar. Por esta razón fue necesario crear un puente de comunicación que enlazara la parte de optimización y la de sísmica, teniendo en cuenta también el uso de las métricas de comparación. En la figura 37 se presentan las trazas sísmicas de referencia, correspondientes a la adquisición de los modelos creados.

Figura 37: Trazas sísmicas adquiridas.



4.2. COMUNICACIÓN ENTRE SU Y MATLAB

En el desarrollo del seminario de investigación se presentaron dos ocasiones donde fue necesario modificar la función, que en un principio se creó para lograr una comunicación entre SU y Matlab. Dicha función se denominó como *inversión automática* y se explica en detalle en las memorias 6 y 9 6. A continuación se presenta una breve descripción de los pasos que se siguieron para crear la función y la estructura que ésta maneja.

El primer puente de comunicación entre SU y Matlab, se realizó por medio de la creación de una función que se encarga de realizar automáticamente las líneas de código en la terminal de Ubuntu para ejecutar los procesos de creación de modelos de subsuelo, adquisición sísmica y comparación de trazas en SU. Cabe aclarar que este tipo de

función se creó en Matlab y es un complemento que el código principal de los algoritmos de optimización llaman después de crear una nueva población. La comunicación existente entre SU y Matlab es de carácter unidireccional en vista de que la función controla el inicio de los procesos sísmicos de SU desde Matlab. La manera en como se vuelve bidireccional, se hace por medio del uso de los comandos de operaciones matemáticas que ofrece SU como lo son filtrado, máxima diferencia, autocorrelación, entre otros. A partir de los resultados de estas operaciones, se procede a convertir a decimal la cadena de caracteres que entrega SU, de modo que Matlab pueda leer el indicador de comparación que en primera instancia se hacía a partir de las operaciones matemáticas de SU.

La segunda modificación entre la comunicación de SU y Matlab se hace para lograr la manipulación de las trazas sísmicas desde Matlab, en vista de que estas están limitadas por los comandos que ofrece el software SU. Para lograr esto, se optó por manipular las trazas sísmicas por medio de un cambio de formato de los archivos de *.SU* a *.SEGY*, permitiendo importar a modo de estructura desde Matlab, toda la información contenida en las trazas con la posibilidad de implementar una métrica de comparación que manipule esta información y arroje así los datos más dicentes a los algoritmos de optimización.

Finalmente la función *inversión automática* tiene la capacidad de hacer de manera automática las siguientes funciones:

- Actualiza el modelo sintético para cada partícula, de acuerdo a las coordenadas (*sloths*) iniciales por cada una de ellas.
- Genera el modelo con las capas actualizadas.
- Simula la adquisición para obtener las trazas sintéticas.
- Realiza la conversión del formato *.SU* a *.SEGY*, y viceversa.
- Importa la traza sintética y la real a Matlab en forma de estructura.
- Calcula los coeficientes de la correlación entre las trazas sintéticas y reales.

4.3. PSO EN INVERSIÓN SÍSMICA

En el transcurso del seminario de investigación surgieron una serie de modificaciones progresivas, que dieron lugar a una implementación adecuada del algoritmo en el problema de inversión sísmica. La explicación paso a paso de las modificaciones que tuvieron lugar en el algoritmo de optimización global PSO, se presentan en las memorias 9-16 6. Una síntesis de los procesos por los cuales el algoritmo pasó, se explican a continuación:

En primer lugar se realizó una implementación del algoritmo PSO para buscar un conjunto de parámetros, por medio de las coordenadas que la mejor partícula presentara, de modo que coincidieran con los mismos valores o con valores aproximados a los *sloths* del modelo de referencia. Para lograr esto, en primer lugar se optó por hacer una comparación sencilla por medio del comando *maxdiff* de SU, el cual permite calcular la máxima diferencia entre las trazas de referencia y las trazas sintéticas, provenientes del mejor modelo encontrado por PSO.

En segundo lugar, las modificaciones hechas llevaron a la implementación de una métrica de comparación más adecuada con el problema que se tenía (comparación de trazas sísmicas). Esta métrica fue la de correlación punto a punto. La manera en como se obtenía la información de comparación de las trazas sintéticas y las de referencia, se hizo por medio de un vector de coeficientes de correlación que la métrica arrojaba. A partir de esto, se determinó iteración tras iteración si la partícula mejoraba. Esto se hizo de manera repetitiva, hasta llegar al punto donde se obtenía la mejor partícula con el mejor conjunto de *sloths* que generó un modelo de subsuelo y un conjunto de trazas sintéticas lo más semejantes posibles a los datos de referencia.

La tercer modificación consistió en la implementación de una nueva métrica alternativa, conocida como correlación cruzada. La manera de cómo se implementó dicha técnica difiere en que el vector que entrega la métrica, se conforma por los coeficientes máximos de correlación presentados en la comparación dinámica de las trazas. La metodología empleada para este tipo de métrica consistió en la misma que para la métrica de correlación punto a punto.

La cuarta modificación fue la de utilizar una tercer métrica de comparación que consistía en el calculo del error absoluto a partir de la diferencia de las trazas sintéticas con las reales. Esta métrica fue nombrada como diferencia de trazas. El uso de esta métrica de comparación requirió realizar modificaciones en el código principal de PSO, en vista de que ya no se contaba con la representación por medio de coeficientes de correlación sino con un vector con los valores de la diferencia directa entre las trazas; es por esto que la manera como se determinó si la partícula mejoraba, consistía en encontrar el mínimo del vector resultante de la diferencia, lo cual representaba la mejor partícula del enjambre .

Finalmente se mantuvo las últimas modificaciones del algoritmo para las tres métricas de comparación, y se procedió a hacer pruebas con los mejores valores extraídos del análisis en las funciones multivariables. A partir de la variación de estos parámetros, se seleccionó el mejor conjunto de parámetros para el problema de inversión sísmica.

Este conjunto de parámetros permitió encontrar los *sloths*, correspondientes a un modelo sintético de subsuelo, con valores aproximados a los del modelo de subsuelo usado como referencia.

4.3.1. Implementación Para la implementación del algoritmo PSO en inversión sísmica es necesario tener claro que el número de dimensiones va a pasar de 2 a 4 debido a que el modelo del subsuelo utilizado posee 4 capas. Por otra parte, fue necesario cambiar la manera como se implementó el algoritmo en funciones multivariable ya que no se cuenta con la evaluación de partículas sobre una función objetivo, sino con un indicador constituido por vectores de datos o coeficientes de correlación.

En primer lugar, la función sobre la que se evaluaron las partículas para determinar los datos de su posición, correspondió al valor que la métrica entregaba. Con base a esto, la manera en como se identificó si la partícula mejoraba, se hizo por medio de operaciones matemáticas sobre los vectores de coeficientes de la correlación o de la diferencia de trazas, según la métrica implementada. En la figura 38 se ilustran las operaciones matemáticas realizadas por las métricas de comparación, para determinar si la partícula mejora en cada iteración. El proceso que se lleva a cabo para las métricas de correlación y correlación cruzada, consiste en el cálculo del error medio cuadrático a partir del error calculado con la diferencia de los vectores de coeficientes ideales y actuales, según la iteración que el algoritmo este ejecutando. Teniendo el vector con los errores medios cuadráticos, se encuentra el mínimo error correspondiente a la mejor partícula la cual encontró el mejor conjunto de parámetros que más se aproximó a los datos observados. La determinación de si la partícula mejora, se hace a partir de la comparación del mínimo error en la iteración actual con la anterior, donde se espera que en la iteración actual el error sea menor. El proceso que se realiza para a la métrica de diferencia de trazas consiste básicamente en encontrar el mínimo error absoluto, correspondiente a la mejor partícula, y a partir de éste, compararlo con el de la iteración anterior, de igual manera a como se realiza en las métricas de comparación.

Figura 38: Operaciones que se realizan para las métricas de correlación y correlación cruzada .

$$\begin{array}{ccc}
 \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} & - & \begin{bmatrix} \# \\ \# \\ \# \end{bmatrix} = \begin{bmatrix} \# \\ \# \\ \# \end{bmatrix} \\
 \text{Coef. ideales} & \text{Coef. actuales} & \text{Error}
 \end{array}
 \quad
 \text{Error medio cuadrático} = \sqrt{\frac{\sum [\text{Error}]^2}{\text{No. coef}}}$$

min (Error medio cuadrático)

En el transcurso de las modificaciones que tuvo el algoritmo PSO en el proceso de inversión sísmica, se implementaron los criterios de restitución de las partículas y de

parada utilizados para las funciones multivariable. El criterio de restitución de las partículas, suministrado por el grupo de investigación CEMOS de la Universidad Industrial de Santander, inicialmente ofreció la posibilidad de reubicar las partículas (de manera aleatoria en el espacio de búsqueda o en las fronteras) para los casos donde éstas salieran del espacio de solución a causa del cálculo de su nueva posición. Posteriormente, este criterio se modificó de modo que la reubicación de partículas se hiciera a la última posición aceptada dentro del espacio de solución. La razón por la que se hizo de esta manera, era que se necesitaba conservar la trayectoria que llevaban las partículas, y en caso de tomar un paso erróneo que la ubicara fuera del espacio de solución, volvieran a la última posición aceptable, y así tomaran un nuevo rumbo.

El criterio de parada implementado a lo largo del seminario de investigación, autoría del grupo de investigación CEMOS de la Universidad Industrial de Santander, en primer lugar contó solo con la restricción de que llegará al número de iteraciones permitidas por el usuario. Posteriormente, este criterio se permitió detener el algoritmo si, por un lado se registraba en 5 iteraciones consecutivas un mismo mínimo global, o si por el contrario, se llegaba a un máximo número de iteraciones permitidas. Finalmente, se modificó y se perfeccionó este criterio a criterio propio, logrando detener el algoritmo por tres razones distintas: la primera de ellas ocurría si se llegaba al máximo número de iteraciones, la segunda ocurría garantizando dos cosas: que el enjambre convergiera en un punto y que el mínimo global encontrado se mantuviera constante en 5 iteraciones consecutivas.

4.3.2. Análisis de resultados Para la interpretación del análisis de resultados que se obtuvieron a lo largo del seminario de optimización, se debe tener en cuenta que se hizo bajo la metodología utilizada para la realización de las pruebas de implementación del algoritmo PSO en funciones multivariable y en inversión sísmica. Para el caso de inversión sísmica, en primer lugar, es de tener claro que las primeras implementaciones se hicieron para el modelo de capas no uniformes y con una geometría fija. Los resultados obtenidos para estos casos en un principio solo encontraron el primer valor del *sloth* del modelo de referencia para los tres tipos de métricas utilizadas, los demás valores de *sloths* fueron aleatorios y no presentaron ninguna cercanía o relación con los valores esperados. A partir de esto, se optó por crear una estrategia que permitiera encontrar los valores de *sloths* de las capas 2, 3 y 4, a partir de los resultados obtenidos en la ejecuciones anteriores. Cabe aclarar que en un principio se tenía que el algoritmo únicamente encontraba el valor de *sloth* de la primer capa del modelo, debido a que el arribo de las trazas sísmicas contenía una mayor energía en la primer capa. A raíz de esto, surgieron ideas con respecto a la posibilidad de implementar un métrica que tuviera en cuenta la energía de las demás capas. Estas ideas se utilizaron como recomendaciones para trabajos futuros en vista de que el estudio de una nueva métrica conlleva una demanda de tiempo que entraría en conflicto con el cronograma asignado para la realización del seminario de investigación. El detalle de las ideas que surgieron para dar solución a esta problemática se pueden encontrar en las memorias 14, 15 y 16

6.

Finalmente la estrategia implementada para el proceso de inversión sísmica, fue la de guiar la búsqueda de *sloths* que realiza el algoritmo conforme se van encontrando el valor de los mismos, capa por capa, según la ejecución del algoritmo. Dicho de otra manera, la idea consiste en que en el primer lanzamiento, se encuentra por lo menos la primera capa del modelo de referencia. Teniendo esto, en la segunda ejecución consecutiva del algoritmo, todo el enjambre inicia la búsqueda con el valor de la primera capa ya encontrada excepto una partícula que posee la ubicación del mínimo global encontrado en la primera ejecución. Terminada la segunda ejecución, se vuelve a realizar el mismo procedimiento hasta llegar a la cuarta ejecución donde se espera que se encuentre el valor de los *sloths* asociados a las cuatro capas del modelo de subsuelo de referencia. Sin embargo, los mejores resultados obtenidos bajo esta estrategia encontraron 3 de las capas del modelo de referencia, con las tres métricas implementadas. Una ilustración de la estrategia implementada se muestra en la figura 39 donde se muestran los valores de *sloths* que el algoritmo va encontrando a medida que realiza las ejecuciones. El símbolo # indica que este valor es aleatorio según el conjunto de parámetros que encuentra el algoritmo en esa ejecución.

Figura 39: Ilustración de la estrategia implementada.

<i>Sloth 1</i>	<i>Sloth 2</i>	<i>Sloth 3</i>	<i>Sloth 4</i>
↓ 0,77	#	#	#
↓ 0,77	↓ 0,50	#	#
↓ 0,77	↓ 0,50	↓ 0,26	#
↓ 0,77	↓ 0,50	↓ 0,26	↓ 0,12

La estrategia implementada para el modelo de capas no uniformes con una geometría fija, fue la misma para los modelos de capas paralelas con y sin la capa de piso, donde se logró obtener buenos resultados en vista de que se encontraron el valor de los *sloths* de las cuatro capas correspondientes al modelo de referencia (modelo de 4 capas paralelas con piso). Cabe resaltar que en un principio, los resultados no fueron mejores a los ya obtenidos para el modelo de capas no uniformes donde se encontraron 3 *sloths* de las cuatro capas del modelo. Esto se debía a que la información de las trazas sísmicas obtenía únicamente tres arribos (información de tres capas) a causa de que el último no se estaba obteniendo a falta de la reflexión de los frentes de onda con la capa de piso, o por el corte de la información de la traza a causa del poco tiempo de grabación de los geófonos virtuales. Teniendo en cuenta estas consideraciones, se aumentó el tiempo de grabación de los geófonos permitiendo obtener un comportamiento distinto a los presentados para los casos anteriores. Bajo esta modificación se obtuvieron los mejores resultados mostrados en la figura 21.

A pesar de haber obtenido el resultado que un principio se quería obtener (encontrar para algún algoritmo los 4 *sloths* de referencia), se optó por realizar unas últimas pruebas para el algoritmo PSO sobre los modelos de cuatro capas no uniformes y paralelas, con la característica de que la población inicial de todos los experimentos fuese la misma (para todos los modelos con cada una de las métricas de comparación). Cabe recordar que la ubicación de esta población, se hace con una distribución uniforme dentro del espacio de búsqueda. Los resultados obtenidos bajo estas condiciones, no presentaron la cercanía que se había logrado con las pruebas anteriores. La hipótesis planteada para este caso presentado, se debe a que la población inicial que se le dio al algoritmo no permitía las mismas condiciones que se tuvieron a la hora de hacer las pruebas por separado, es decir que el punto de partida de otros experimentos pudo ser más factible que el que se entregó para estas pruebas.

A partir de los mejores resultados obtenidos, donde se encontraron 3 *sloths* para el modelo de capas no uniformes y 4 *sloths* para el modelo de capas paralelas con piso. El análisis que se hizo en la última sesión del seminario de investigación permitió concluir que el no poseer la información del frente de onda que se refleja en la última capa del modelo de subsuelo, es un caso que en campo también puede ocurrir puesto que la energía de las ondas que se propagan desde la fuente en una adquisición sísmica, va disminuyendo su intensidad a medida que recorre terreno en el subsuelo; perdiendo en algún momento la capacidad de ser captada por los geófonos que perciben la reflexión de dichas ondas al pasar de una capa a otra.

En la tabla 21, se muestran los mejores resultados obtenidos por el algoritmo PSO aplicado a inversión sísmica. Se puede evidenciar que en cada ejecución del algoritmo, el enjambre se fue acercando cada vez más a los parámetros (*sloths*) del modelo de referencia. En la tabla se muestra el número de iteraciones que demoró el enjambre en cada lanzamiento. Los parámetros utilizados en este experimento fueron 30 partículas de búsqueda, constantes de exploración local y global de 1,8 y 1,9 respectivamente, peso de inercia y radio de avance de 0,5, un contador de saturación de 100 y ϵ_1 y ϵ_2 de 10^{-3} . Esta prueba se aplicó al modelo de 4 capas paralelas con una capa adicional de piso, y utilizando la métrica de correlación cruzada. Los detalles de la prueba como la trayectoria de los *sloths* y coeficientes se pueden ver en las estructuras ubicadas en la carpeta Estructuras del CD anexo. El tiempo de ejecución de este experimento fue de 9 horas aproximadamente.

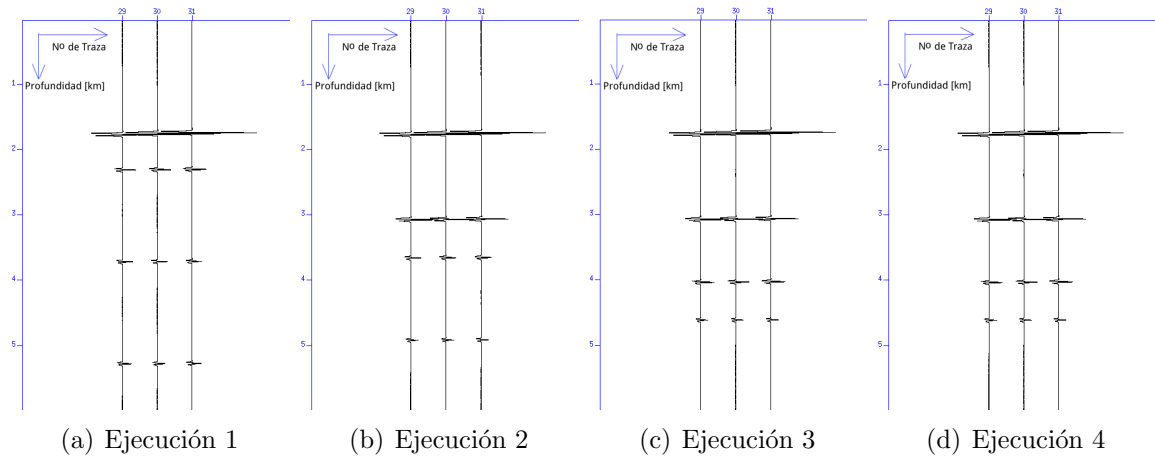
Una representación gráfica de una sección de las trazas sísmicas obtenidas por la mejor partícula en cada ejecución de la prueba que arrojó mejores resultados para PSO, se muestra en la figura 40. En esta figura se tiene que la última sección de las trazas sísmicas obtenidas es igual a la sección de trazas tomadas como referencia.

La tendencia del error medio cuadrático que presentó la mejor partícula en cada eje-

Tabla 21: *Sloths* encontrados por el algoritmo PSO en inversión sísmica.

Lanzamiento	Iteraciones	<i>Sloth</i> 1	<i>Sloth</i> 2	<i>Sloth</i> 3	<i>Sloth</i> 4
1	53	0,7700	0,1000	0,5450	0,8516
2	100	0,7700	0,5000	0,1000	0,5547
3	100	0,7700	0,5000	0,2600	0,1200
4	39	0,7700	0,5000	0,2600	0,1200

Figura 40: Trazas generadas en las tres primeras ejecuciones del algoritmo PSO.



cución, para los mejores resultados obtenidos en PSO, se muestra en la figura 41. Esta tendencia presenta una disminución a medida que van transcurriendo las iteraciones.

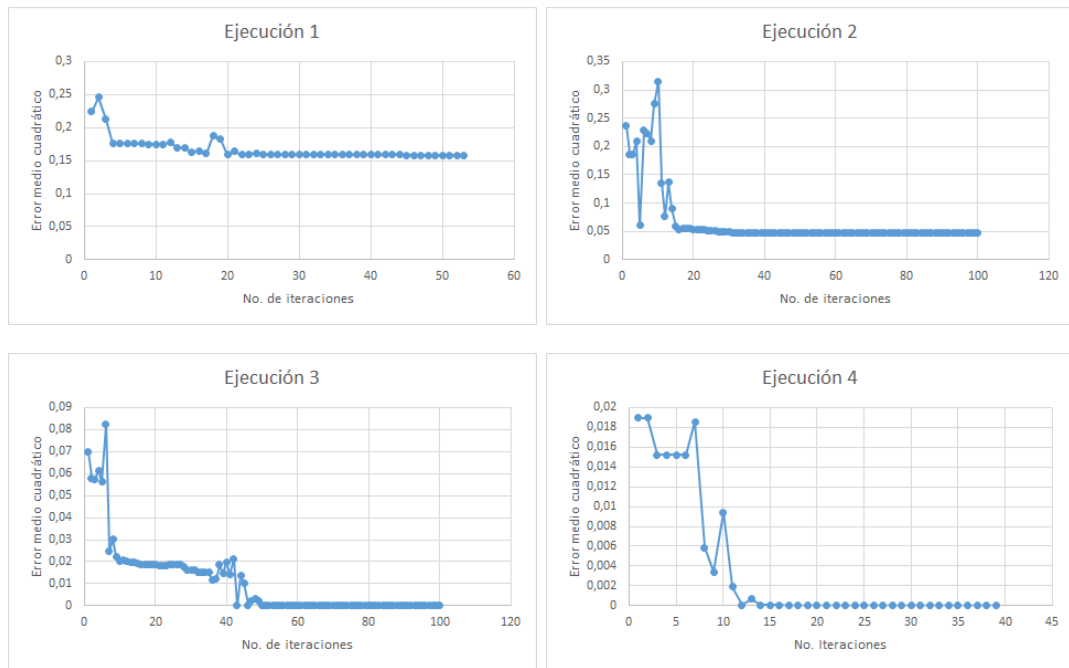
4.4. GA EN INVERSIÓN SÍSMICA

4.4.1. Implementación La implementación de algoritmos genéticos se hizo de manera similar a PSO, donde se tuvieron en cuenta las mismas consideraciones del aumento de dimensiones del problema, las métricas de comparación utilizadas y los modelos de subsuelo usados como referencia.

debido a que el número de dimensiones igual que en el otro algoritmo aumentó de 2 a 4. El cambio de la evaluación de la función objetivo por la implementación de una determinada métrica de comparación de trazas sísmicas es el mismo y las tres métricas explicadas anteriormente también fueron implementadas, los detalles de la implementación del algoritmo en Matlab se encuentran en la memoria 10 6.

El algoritmo tuvo modificaciones particulares en la manera como se implementó el cruce de individuos. En un principio, este se había realizado con un punto de cruce aleatorio

Figura 41: Tendencia del error medio cuadrático para las cuatro ejecuciones del algoritmo PSO.



en un vector de 64 bits que conformaba cada individuo. Este punto se asignaba sin respetar en dicho vector la conformación de las 4 dimensiones existentes en el proceso de inversión sísmica; es decir que era posible que cuando se hiciera el cruce para crear un nuevo individuo, se generara una mutación debido a que algunas dimensiones del nuevo individuo podrían quedar conformadas por tramos de bits de dos dimensiones. Para solucionar este problema se reprogramó el algoritmo haciendo que el punto de cruce solo pudiese tomar valores que dividían la cadena de bits de los individuos padres en las 4 dimensiones. La necesidad de respetar cada dimensión, permitiría al algoritmo crear nuevos individuos a partir del cruce de dimensiones de los mejores padres, garantizando de esta manera, la evolución de las mejores características en las generaciones futuras.

Los errores cometidos en las primeras implementaciones y pruebas realizadas para el algoritmo PSO no se repitieron en algoritmos genéticos, debido a esto, la explicación de las implementaciones en el algoritmo GA no tienen la misma cantidad que se tuvo para PSO. Como ejemplo de estos errores, se tiene el momento en que se implementó una métrica de comparación sencilla, que calculaba la máxima diferencia entre dos vectores de trazas correspondientes a las trazas sintéticas y de las referencia; este método no se utilizó con algoritmos genéticos debido a que fue una primer prueba para la comprobación del funcionamiento del canal de comunicación entre SU y Matlab, memoria 6 6. Teniendo en cuenta lo anterior, se realizaron menos pruebas para algoritmos genéticos pero aun así en las últimas sesiones se implementaron los dos algoritmos con el fin de

poder hacer una comparación entre éstos.

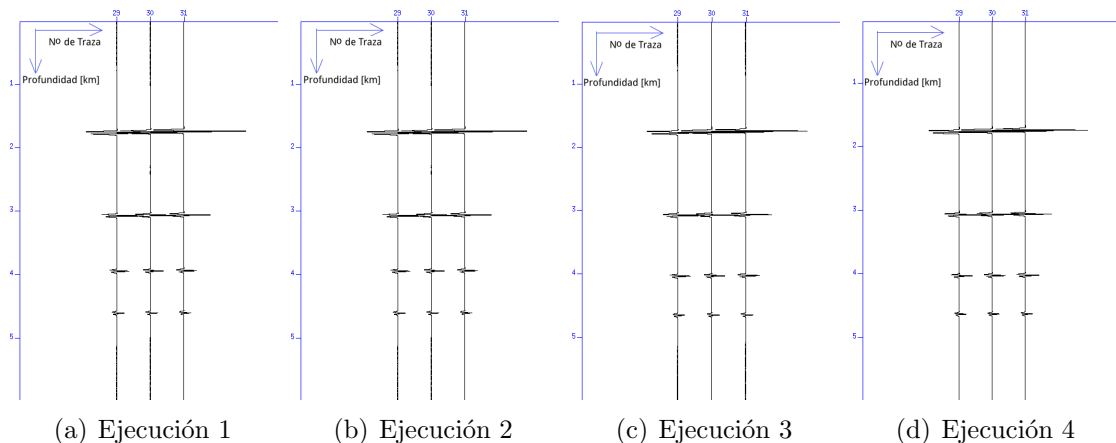
4.4.2. Análisis de resultados En la tabla 22 se muestran los mejores resultados obtenidos con algoritmos genéticos aplicados a inversión sísmica. Esta prueba se aplicó al modelo de 4 capas paralelas, modificado con una capa adicional y utilizando la métrica de correlación. Los parámetros empleados en este experimento fueron: 30 individuos de búsqueda, tasa de cruce y mutación de 1,0 y 0,1, respectivamente, un contador de saturaciones de 100 iteraciones y número de bits de 16. Los detalles de la prueba como la trayectoria de los *sloths* y los coeficientes se pueden ver en las estructuras de la carpeta Estructuras del CD anexo. El tiempo de ejecución de este experimento fue de 1 hora aproximadamente.

Tabla 22: *Sloths* encontrados por el algoritmo GA en inversión sísmica.

Lanzamiento	Iteraciones	<i>Sloth</i> 1	<i>Sloth</i> 2	<i>Sloth</i> 3	<i>Sloth</i> 4
1	100	0,7697	0,5006	0,2167	0,1561
2	100	0,7697	0,5006	0,2167	0,1561
3	100	0,7697	0,5004	0,2589	0,1362
4	39	0,7697	0,5004	0,2600	0,1353

La representación gráfica de una sección de las trazas sísmicas obtenidas por el mejor individuo en cada ejecución de la prueba que arrojó mejores resultados para GA, se muestra en la figura 42. Es de recordar que el conjunto de parámetros encontrados no fue idéntico a los de referencia, por tal razón, la sección de trazas sísmicas de esta prueba se aproximan a las encontradas en PSO.

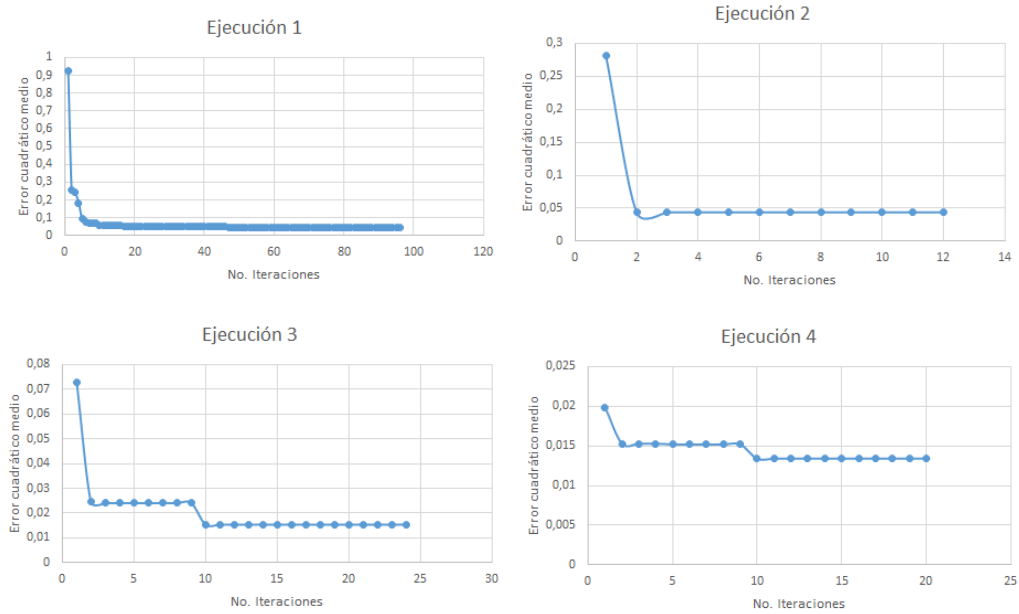
Figura 42: Trazas generadas en cada ejecución del algoritmo GA.



La tendencia del error medio cuadrático que presentó el mejor individuo en cada ejecución, para los mejores resultados obtenidos en GA, se muestra en la figura 43. En

esta figura, de igual manera a como se presenta en PSO, se observa una disminución en el error medio cuadrático a medida que avanzan las iteraciones.

Figura 43: Tendencia del error medio cuadrático para las cuatro ejecuciones del algoritmo GA.



4.5. DISCUSIÓN

En vista de que los resultados obtenidos fueron cercanos a los datos de referencia (*sloths* del modelo de placas paralelas), se considera que este tipo de modelos facilitan la comparación entre las trazas y la búsqueda de los parámetros de un modelo base con geometría sencilla. Hay que tener en cuenta que el mantener una geometría fija para los modelos de subsuelo, ayuda en la complejidad del problema a resolver. Los problemas que en el campo se presentan, son modelos de subsuelo donde no existe ninguna geometría de las capas y los *sloths* de las mismas varían sin ningún patrón.

Ya que los algoritmos de optimización utilizados son robustos, pudiendo utilizarlos en distintos problemas de optimización. Se considera que el problema de inversión sísmica constituye uno de estos problemas donde el desempeño de las técnicas metaheurísticas como PSO y GA, funciona.

Para realizar el proceso de inversión sísmica basada en el modelo utilizando algoritmos metaheurísticos, es necesario que las trazas sísmicas proporcionen la información suficiente de todas las capas del subsuelo. De lo contrario el algoritmo encontrará solo un

determinado número de capas de acuerdo a la información suministrada. Una de las herramientas que permiten verificar la información de las trazas sísmicas y la evolución que van teniendo a medida que pasan las generaciones, es la visualización de las trazas sísmicas en todo el proceso evolutivo.

La combinación adecuada de los parámetros como el tamaño de la población, la tasa de cruce, la tasa de mutación, el peso de inercia, el radio de avance y las constantes de exploración para los algoritmos GA y PSO; es un factor que influye en el funcionamiento de los algoritmos y en la obtención de buenos resultados. La elección del mejor conjunto de valores de los parámetros para cada algoritmo, se hizo a partir de la variación de los mismos sobre la implementación de funciones multivariadas. Los mejores valores extraídos de estas implementaciones, no funcionaron de la misma manera en el proceso de inversión sísmica. Esto se debe a que el espacio de solución del problema de inversión sísmica es menor con respecto a los presentados por los distintos tipos de funciones multivariadas.

Guiar la búsqueda de los algoritmos en problemas de inversión sísmica, si bien sesga el problema que se está atacando, significó una muy buena estrategia que conllevó a la obtención del valor de los *sloths* del modelo del subsuelo de referencia.

Fijar el punto de partida de las partículas de búsqueda, si bien podría permitir la comparación de los resultados en las diferentes pruebas de los dos algoritmos meta-heurísticos, ocasiona un comportamiento distinto y erróneo al caso donde funcionan por separado.

A pesar de considerar distintas estrategias para no sesgar el problema desde la población inicial, se sabe que de una u otra forma la ubicación inicial de las partículas o individuos va influir en la manera como el algoritmo encuentra una solución.

5. CONCLUSIONES

1. Un tamaño de 30 agentes de búsqueda en las técnicas metaheurísticas PSO y GA en este estudio, permitió obtener mayor información del espacio de solución y garantizar mejores precisiones con respecto al punto encontrado. De igual manera, representa una reducción en el número de iteraciones realizadas debido que al tener mayor información del espacio de solución, se evita tener que esperar varias iteraciones para obtener información de algunas zonas que para el caso de poblaciones grandes ya están exploradas. En problemas de inversión sísmica se tiene que el incremento de la población aumenta la probabilidad de encontrar un conjunto de *sloths* que se acerquen a los datos observados. Sin embargo, se debe considerar que éste aumento en la población demanda tiempo de computo que será un factor de interés según la necesidad y el problema que se esta atacando.
2. El análisis de la implementación sobre funciones multivariable, permitió identificar un rango de valores adecuado para los cuales el mínimo global fue encontrado en pocas iteraciones y con valores aproximados a los de referencia. Sin embargo, el uso de estos valores en inversión sísmica no produjo resultados adecuados. A pesar de esto, el análisis sirvió para la elección de los nuevos valores con los cuales se logró llegar a encontrar por completo el conjunto de parámetros buscados.
3. En ocasiones el número de iteraciones representa un indicador por el cual se pueden considerar valores en los parámetros mejores que otros. No siempre la búsqueda de la disminución del número de iteraciones presenta buenos resultados y un número elevado de iteraciones no garantiza buenos resultados. En la inversión sísmica ocurrió en la mayoría de las pruebas que el número de iteraciones fue elevado (mayor a 100) y aún así el algoritmo no logró encontrar buenos resultados. Por tal razón fue necesario el uso de estrategias que garantizaran la convergencia del algoritmo evitando iteraciones innecesarias y el hallazgo de mejores resultados. Dichas estrategias se pueden resumir como la búsqueda de *sloths* de los algoritmos PSO y GA en inversión sísmica, de manera directa (búsqueda estándar) y de manera guiada (búsqueda guiada conservando el valor de las capas que en cada ejecución se encuentra).
4. Se comprobó que un peso de inercia intermedio entre 0 y 1 facilita la búsqueda del punto de interés, guardando un equilibrio en los factores sobre los cuales tiene influencia. El valor asignado para este parámetro en el problema de inversión sísmica fue de 0,5. La elección de este valor surgió a partir del análisis hecho para la implementación del algoritmo en funciones multivariable.
5. El radio de avance para el algoritmo PSO es otro parámetro que determina de manera directa qué tan dinámica se quiere hacer la búsqueda en cada actualización. Comúnmente este parámetro es considerado como 1 en las ecuaciones de

actualización de PSO, sin embargo es oportuno mencionar que en el seminario de investigación se tuvo en cuenta un valor distinto con el que se obtuvieron buenos resultados en funciones multivariantes. Se recomienda usar un valor intermedio entre 0 y 1 que permita a las partículas realizar una actualización equilibrada y acorde con el espacio de solución que se define de acuerdo con las fronteras de la función. El valor asignado para este parámetro, teniendo en cuenta el análisis hecho en la implementación de funciones multivariable, fue de 0,5.

6. La dinámica en la actualización de las partículas de PSO, se puede compensar con los parámetros w y dt . Se recomienda un valor intermedio entre 0 y 1 para ambos parámetros a modo que se guarde un equilibrio entre ellos. Los valores recomendados por el seminario de investigación con respecto a la implementación de los algoritmos en problemas de inversión sísmica corresponden a $w = 0,5$ y $dt = 0,5$.
7. Los valores de las constantes $c1$ y $c2$ para el algoritmo PSO, influyen de manera directa en la forma como se comunican las partículas y se agrupan en su movimiento progresivo. Por tanto es mejor darle un peso mayor a la búsqueda global con respecto a la local; por un lado, porque se está trabajando con algoritmos de optimización global, pero por el otro, porque es necesario explorar muy bien el espacio de solución y no las pequeñas zonas donde, tentativamente, se encuentre el mínimo global. Es oportuno aclarar que es necesario contar con ambas componentes puesto que no hacerlo cambiaría la estructura de los algoritmos. El valor asignado para cada constante a partir del análisis en funciones multivariable y en trabajos de PSO [2], [1], [9], fue de $c1 = 1,8$ y $c2 = 1,9$ para el problema de inversión sísmica.
8. La tasa de mutación en GA tiene gran relación con respecto al tipo de búsqueda que se le quiere dar al algoritmo. Un valor alto para dicho parámetro (mayor a 0,1) permite que los individuos se dispersen de mejor manera en el espacio de solución, permitiendo una búsqueda más global. Un valor bajo (menor a 0,1) permite que la búsqueda sea más compacta, pero a la vez, más detallada. El valor recomendado por el seminario de investigación es $Tm = 0,5$, puesto que para los tipos de funciones multivariable es oportuno contar con la capacidad de tener saltos evolutivos que permitan llegar más rápidamente a la solución. A pesar de obtener buenos resultados en las funciones multivariable, para el problema de inversión fue necesario disminuir la tasa de mutación a un valor de 0,1, en vista de que el espacio de solución, para este caso, es inferior al que las funciones presentaban.
9. Una tasa alta de cruce (cercana a 1) en GA es oportuna debido a que permite conservar de manera evolutiva las características de los mejores individuos en cada

generación. El valor recomendado por el seminario de investigación oscila en el rango de 0,7 a 1,0. El análisis extraído, a partir de la implementación sobre funciones multivariable, permitió realizar más pruebas en el problema de inversión; a pesar de esto, el valor que presentó mejores resultados, bajo la metodología de la variación en los parámetros del algoritmo desarrollada en el seminario, fue para una tasa de cruce de 1.

10. Se debe tener principal cuidado que a la hora de implementar estas técnicas meta-heurísticas, de modo que corresponda a la interpretación adecuada de la analogía, que éstas tienen con la naturaleza. En ese sentido, se debe corroborar que lo implementado se ajuste al comportamiento correcto del algoritmo, de lo contrario, se estaría incurriendo en una variante del algoritmo, cuya dinámica debe estudiarse por separado.
11. Considerar una resolución más fina para la geometría de las capas del modelo del subsuelo cambia por completo el planteamiento del problema que se estudió en este seminario. Esta consideración constituye un reto de mayor nivel, que solo hace parte de una pequeña porción del proceso de inversión sísmica real.
12. La estrategia implementada para guiar la búsqueda de los *sloths* a partir de los resultados obtenidos en cada ejecución, permitió evidenciar buenos resultados reflejados en la obtención de los cuatro *sloths* que se tenían como referencia. La carencia de esta estrategia, conlleva a la interpretación errónea de los resultados del algoritmo, donde se cree que el funcionamiento es erróneo (o incompleto) en vista de que solo se encuentra un valor aproximado al primer *sloth* de la capa del modelo de referencia. La razón por la cual el algoritmo tiende a encontrar solo el parámetro de la primer capa, se debe a que el primer arribo presentado en las trazas sísmicas contiene mayor energía que los demás.

6. RECOMENDACIONES

1. Para trabajos posteriores en la implementación de los algoritmos de optimización global PSO y GA, sería muy útil hacer un programa que haga una variación minuciosa de todos los parámetros que contienen los algoritmos, con la idea de automáticamente obtener resultados de las mejores combinaciones de parámetros. Hecho esto, es posible obtener una base de datos que contenga información importante de las pruebas donde se permita hacer un estudio detallado del comportamiento de los algoritmos con relación a los valores de los parámetros. Finalmente se espera que con el conjunto de mejores combinaciones de parámetros, se logren encontrar resultados precisos en el hallazgo del mínimo global en funciones de prueba de 2D y de igual manera el valor de *sloths* en problemas referentes a la inversión sísmica.
2. Programar los algoritmos PSO y GA para que sus parámetros cambien de manera dinámica durante su ejecución, teniendo en cuenta la evolución de los mismos.
3. Reducir las líneas de código de los algoritmos con la finalidad de minimizar el uso de los recursos computacionales. Por ejemplo, las líneas de código que hacen la actualización de las ecuaciones de posición y velocidad de las partículas en el algoritmo PSO, se podrían reducir al asignar una matriz que represente en sus columnas y filas los valores necesarios para actualizar estas ecuaciones, quitando un ciclo de repetición que recorre cada una de las dimensiones para la respectiva actualización.
4. Realizar la implementación de estos algoritmos de optimización global de modo que el computo se haga en paralelo. Como se mencionó en los detalles de la implementación, estas pruebas contienen un alto costo computacional que ocasionan que el resultado de las ejecuciones del algoritmo tarden horas. Paralelizar el computo realizado permitiría obtener los resultados en un menor tiempo, lo cual sería factible para los requerimientos que los procesos de inversión sísmica manejan.
5. Para trabajos futuros relacionados con la implementación de algoritmos de optimización en problemas de inversión sísmica, se podría ahondar en las temáticas requeridas para hacer que la combinación de *sloths* esperada no corresponda a un modelo con geometría fija, como se trabajo durante este seminario, sino que se agreguen las dimensiones necesarias para poner a prueba estos algoritmos en la búsqueda de la combinación de *sloths* y de la geometría del modelo de referencia.
6. Para trabajos futuros relacionados con la implementación de algoritmos de optimización en problemas de inversión sísmica, se podría ahondar en las temáticas requeridas para hacer que la combinación de *sloths* esperada, no corresponda a

un modelo con geometría fija como se trabajo durante este seminario. Por el contrario, se podría aumentar la resolución de la grilla que definen los *sloths* de las capas del modelo, de modo que se aumente la complejidad del problema de inversión y a partir de las variaciones en los *sloths* que se definen en cada grilla, inferir en el comportamiento que se tienen en las capas del modelo del subsuelo.

7. La implementación de métricas de comparación de las trazas sísmicas fueron enfocadas en la comparación de señales en toda su extensión. Estas fueron la correlación, la correlación cruzada y la diferencia de sismogramas. En este trabajo de investigación no se utilizó una métrica que tuviera en cuenta la energía de la señal, sin embargo se planteó la necesidad de considerar este factor, en vista de que el primer arribo de las trazas contiene una mayor energía que puede opacar el valor de las capas más profundas. Por esta razón, es recomendable buscar una métrica que tenga en cuenta la energía con el fin de permitir una nueva alternativa de comparación no contemplada en este trabajo.
8. Realizar pruebas donde se haga un ensamble de las métricas de comparación que permitan obtener un mejor indicador de similitud entre las señales a comparar.
9. Realizar el mismo proceso que se llevó a cabo, pero esta vez realizando una adquisición por medio del software SU a partir de la ecuación de onda y no de trazado de rayos como realizó. Para al final, contrastar lo resultados.

CITAS BIBLIOGRÁFICAS

- [1] WIKIPEDIA. *Geofísica*. [en línea]. Wikipedia, La enciclopedia libre. Fecha de consulta: 5 de Mayo de 2014. Disponible en <http://es.wikipedia.org/wiki/Geofisica>.
- [2] WIKIPEDIA. *Sismología*. [en línea]. Wikipedia, La enciclopedia libre. Fecha de consulta: 5 de Mayo de 2014. Disponible en <http://es.wikipedia.org/wiki/Sismologia>.
- [3] SCHLUMBERGER. *Traza sísmica*. [en línea]. Oilfield Glossary en Español. Fecha de consulta: 5 de Mayo de 2014. Disponible en http://www.glossary.oilfield.slb.com/es/Terms/s/seismic_trace.aspx.
- [4] SCHLUMBERGER. *Modelo del subsuelo*. [en línea]. Oilfield Glossary en Español. Fecha de consulta: 5 de Mayo de 2014. Disponible en <http://www.glossary.oilfield.slb.com/es/Terms/m/model.aspx>.
- [5] TRUJILLO TRUJILLO, MAURICIO. *Exploración Sísmica*. [en línea]. Mauricio Trujillo Trujillo. Fecha de consulta: 5 de Mayo de 2014. Disponible en <http://exploracionsismica.blogspot.com/>.
- [6] MRINAL, SEN y STOFFA, PAUL. *Global Optimization Methods in Geophysical Inversion*. Amsterdam, The Netherlands: Elsevier Science B.V, 1995. 293p.
- [7] GINOZKY. *Registro o Perfilaje de Pozos*. [en línea]. Fecha de consulta: 6 de Mayo de 2014. Disponible en <http://es.scribd.com/doc/26228291/Registro-o-Perfilaje-de-Pozos>.
- [8] WIKIPEDIA. *Optimización*. [en línea]. Wikipedia, La enciclopedia libre. Fecha de consulta: 7 de Mayo de 2014. Disponible en [http://es.wikipedia.org/wiki/Optimizacion_\(matematica\)](http://es.wikipedia.org/wiki/Optimizacion_(matematica)).
- [9] SUAREZ, ORLANDO DE ANTONIO. *Una aproximación a la heurística y meta-heurísticas*. [en línea]. Fecha de consulta: 7 de Mayo de 2014. Disponible en csifesvr.uan.edu.co/index.php/ingewan/article/download/198/170.
- [10] WIKIPEDIA. *Metaheurística*. [en línea]. Wikipedia, La enciclopedia libre. Fecha de consulta: 7 de Mayo de 2014. Disponible en <http://es.wikipedia.org/wiki/Metaheuristica>.

[//es.wikipedia.org/wiki/Metaheuristica](http://es.wikipedia.org/wiki/Metaheuristica).

- [11] GOMEZ, JUAN CARLOS. *Reconocimiento Automático de Voz basado en Técnicas de Comparación de Patrones*. [en línea]. Fecha de consulta: 8 de Mayo de 2014. Disponible en http://www.fceia.unr.edu.ar/prodivoz/RAV_Comparacion_Patrones_bw.pdf.

- [12] WIKIPEDIA. *Medidas de dispersión*. [en línea]. Wikipedia, La enciclopedia libre. Fecha de consulta: 8 de Mayo de 2014. Disponible en http://es.wikipedia.org/wiki/Medidas_de_dispersi%C3%B3n.

- [13] WIKIPEDIA. *Correlación*. [en línea]. Wikipedia, La enciclopedia libre. Fecha de consulta: 8 de Mayo de 2014. Disponible en <http://es.wikipedia.org/wiki/Correlaci%C3%B3n>.

- [14] WIKIPEDIA. *Coefficiente de correlación de Pearson*. [en línea]. Wikipedia, La enciclopedia libre. Fecha de consulta: 8 de Mayo de 2014. Disponible en http://es.wikipedia.org/wiki/Coefficiente_de_correlaci%C3%B3n_de_Pearson.

- [15] HINTON, OLIVER. *Describing Random Sequences*. [en línea]. EEE305 - Digital Signal Processing Resources Available. Fecha de consulta: 8 de Mayo de 2014. Disponible en <https://www.staff.ncl.ac.uk/oliver.hinton/eee305/Chapter6.pdf>.

- [16] WIKIPEDIA. *Correlación cruzada*. [en línea]. Wikipedia, La enciclopedia libre. Fecha de consulta: 8 de Mayo de 2014. Disponible en http://es.wikipedia.org/wiki/Correlaci%C3%B3n_cruzada.

- [17] WIKIPEDIA. *Error experimental*. [en línea]. Wikipedia, La enciclopedia libre. Fecha de consulta: 8 de Mayo de 2014. Disponible en http://es.wikipedia.org/wiki/Error_experimental.

- [18] CAICEDO, MARIO y MORA, PLÁCIDO. *Temas de Propagación de Ondas*. [en línea]. Universidad Simón Bolívar. Fecha de consulta: 8 de Mayo de 2014. Disponible en <http://www.fis.usb.ve/~mcaicedo/geophysics/ondas.pdf>.

- [19] BARRAGAN, KAROL y VANEGAS, DIEGO. *Real potencial del uso del método de análisis de intervalos para la optimización con restricciones frente al PSO de convergencia garantizada*. Bucaramanga, 2010, 102p. Trabajo de grado (Ingeniero

Electrónico). Universidad Industrial de Santander. Facultad de Ingenierías Físico-Mecánicas. Escuela de Ingenierías Eléctrica, Electrónica y de Telecomunicaciones.

- [20] ARIAS, CESAR y AGUILAR RICARDO. *Soluciones de sistemas de ecuaciones no lineales mediante el método metaheurístico PSO*. Bucaramanga, 2011, 50p. Trabajo de grado (Ingeniero Electrónico). Universidad Industrial de Santander. Facultad de Ingenierías Físico-Mecánicas. Escuela de Ingenierías Eléctrica, Electrónica y de Telecomunicaciones.

- [21] PLATA, JULIÁN y REYES SERGIO. *Determinación de raíces reales y complejas en sistemas de ecuaciones utilizando optimización por enjambre de partículas (PSO)*. Bucaramanga, 2011, 49p. Trabajo de grado (Ingeniero Electrónico). Universidad Industrial de Santander. Facultad de Ingenierías Físico-Mecánicas. Escuela de Ingenierías Eléctrica, Electrónica y de Telecomunicaciones.

- [22] EBERHART, RUSS C y KENNEDY, JAMES. *A new optimizer using particle swarm theory*. En Proceedings of the sixth international symposium on micro machine and human science. 1995. p. 39-43.

- [23] SHI, YUHUI y EBERHART, RUSSELL. *A modified particle swarm optimizer*. En Evolutionary Computation Proceedings, 1998. IEEE World Congress on Computational Intelligence. The 1998 IEEE International Conference on. IEEE, 1998. p. 69-73.

- [24] GARCÍA, JOSÉ . *Algoritmos Basados en Cúmulos de Partículas para la Resolución de Problemas Complejos*. [en línea]. Fecha de consulta: 12 de Mayo de 2014. Disponible en neo.lcc.uma.es/staff/jmgn/doc/Memoria_PFC_JMGN.pdf.

- [25] MOLINA, JORGE y PAEZ, EDUARDO. *Optimización del peso de cerchas en 3D mediante Algoritmos Genéticos (GA) y Optimización por Enjambre de Partículas (PSO)*. Bucaramanga, 2010, 186p. Trabajo de grado (Ingeniero Civil). Universidad Industrial de Santander. Facultad de Ingenierías Físico-Mecánicas. Escuela de Ingeniería Civil.

- [26] WIKIPEDIA. *Seismic Unix*. [en línea]. Wikipedia, La enciclopedia libre. Fecha de consulta: 14 de Julio de 2014. Disponible en http://en.wikipedia.org/wiki/Seismic_Unix.

- [27] CRUZ, JAVIER. *Ejemplo de algoritmo genético en Matlab código y funciones*. [en línea]. Fecha de consulta: 30 de Junio de 2014.

Disponible en <http://cienciafisicamatlab.blogspot.com/2011/11/ejemplo-de-algoritmo-genetico-en-matlab.html>.

- [28] FOREL, DAVID, BENZ, THOMAS y PENNINGTON, WAYNE. *Seismic Data Proccesing with Seismic Un*x*. United States of América: Series Editor, 2005. 277p.
- [29] GARCÍA, ALVARO. *Técnicas metaheurísticas*. [en línea]. Fecha de consulta: 7 de Mayo de 2014. Disponible en <http://www.iol.etsii.upm.es/arch/metaheuristicas.pdf>.

BIBLIOGRAFÍA

- [1] ARIAS, CESAR y AGUILAR RICARDO. *Soluciones de sistemas de ecuaciones no lineales mediante el método metaheurístico PSO*. Bucaramanga, 2011, 50p. Trabajo de grado (Ingeniero Electrónico). Universidad Industrial de Santander. Facultad de Ingenierías Físico-Mecánicas. Escuela de Ingenierías Eléctrica, Electrónica y de Telecomunicaciones.
- [2] BARRAGAN, KAROL y VANEGAS, DIEGO. *Real potencial del uso del método de análisis de intervalos para la optimización con restricciones frente al PSO de convergencia garantizada*. Bucaramanga, 2010, 102p. Trabajo de grado (Ingeniero Electrónico). Universidad Industrial de Santander. Facultad de Ingenierías Físico-Mecánicas. Escuela de Ingenierías Eléctrica, Electrónica y de Telecomunicaciones.
- [3] CAICEDO, MARIO y MORA, PLÁCIDO. *Temas de Propagación de Ondas*. [en línea]. Universidad Simón Bolívar. Fecha de consulta: 8 de Mayo de 2014. Disponible en <http://www.fis.usb.ve/~mcaicedo/geophysics/ondas.pdf>.
- [4] CRUZ, JAVIER. *Ejemplo de algoritmo genético en Matlab código y funciones*. [en línea]. Fecha de consulta: 30 de Junio de 2014. Disponible en <http://cienciafisicamatlab.blogspot.com/2011/11/ejemplo-de-algoritmo-genetico-en-matlab.html>.
- [5] EBERHART, RUSS C y KENNEDY, JAMES. *A new optimizer using particle swarm theory*. En Proceedings of the sixth international symposium on micro machine and human science. 1995. p. 39-43.
- [6] FOREL, DAVID, BENZ, THOMAS y PENNINGTON, WAYNE. *Seismic Data Proccesing with Seismic Un*x*. United States of América: Series Editor, 2005. 277p.
- [7] MOLINA, JORGE y PAEZ, EDUARDO. *Optimización del peso de cerchas en 3D mediante Algoritmos Genéticos (GA) y Optimización por Enjambre de Partículas (PSO)*. Bucaramanga, 2010, 186p. Trabajo de grado (Ingeniero Civil). Universidad Industrial de Santander. Facultad de Ingenierías Físico-Mecánicas. Escuela de Ingeniería Civil.
- [8] MRINAL, SEN y STOFFA, PAUL. *Global Optimization Methods in Geophysical Inversion*. Amsterdam, The Netherlands: Elsevier Science B.V, 1995. 293p.

- [9] PLATA, JULIÁN y REYES SERGIO. *Determinación de raíces reales y complejas en sistemas de ecuaciones utilizando optimización por enjambre de partículas (PSO)*. Bucaramanga, 2011, 49p. Trabajo de grado (Ingeniero Electrónico). Universidad Industrial de Santander. Facultad de Ingenierías Físico-Mecánicas. Escuela de Ingenierías Eléctrica, Electrónica y de Telecomunicaciones.
- [10] SHI, YUHUI y EBERHART, RUSSELL. *A modified particle swarm optimizer*. En *Evolutionary Computation Proceedings, 1998*. IEEE World Congress on Computational Intelligence. The 1998 IEEE International Conference on. IEEE, 1998. p. 69-73.
- [11] SUAREZ, ORLANDO DE ANTONIO. *Una aproximación a la heurística y meta-heurísticas*. [en línea]. Fecha de consulta: 7 de Mayo de 2014. Disponible en csifesvr.uan.edu.co/index.php/ingean/article/download/198/170.
- [12] TRUJILLO TRUJILLO, MAURICIO. *Exploración Sísmica*. [en línea]. Mauricio Trujillo Trujillo. Fecha de consulta: 5 de Mayo de 2014. Disponible en <http://exploracionsismica.blogspot.com/>.

ANEXOS

El capítulo de anexos corresponde a una extensión del documento principal, en la cual se muestra: los códigos de las implementaciones de los algoritmos de optimización en funciones multivariable e inversión sísmica, y las memorias semanales que describen detalladamente lo que se hizo semana tras semana en el seminario de investigación.

DESCRIPCIÓN DEL CÓDIGO PSO

Es importante mencionar que la realización del código de PSO utilizado en este seminario, se hizo a partir de las asesorías en PSO brindadas por el grupo CEMOS de la Universidad Industrial de Santander. De igual manera, es oportuno aclarar que la función empleada para el criterio de parada de este algoritmo, es una modificación propia de la función creada por el grupo CEMOS, nombrada `Chk_out.m`.

Código Principal De acuerdo con el pseudocódigo, el primer paso corresponde a la declaración de las variables que corresponden a: número de partículas (ns), constantes de exploración local y global ($c1$ y $c2$), peso de inercia (w), radio de avance (dt), constantes necesarias para el criterio de parada ($\epsilon1$, $\epsilon2$, $Cont$, $flag$), número máximo de iteraciones permitidas ($contsat$) y definición de las fronteras del espacio de solución, que en este caso corresponden a los de la función de prueba básica.

Listing 6.1: Declaración de variables

```
1 clear all,close all,clc
2 format long
3 %% Definicion de valores
4 ns= 30; % Numero de particulas
5 c1=1.9; % Constante 1
6 c2=1.8; % Constante 2
7 w= 0.5; % Peso de la inercia
8 dt=0.5; % Delta de tiempo Define el radio de avance de la particula
9 e1=0.1; % Primer criterio de Parada
10 e2=0.1; % Segundo criterio de Parada
11 contsat=2000; % Contador de saturacion de iteraciones
12 Xmin=0; % Limite inferior en X
13 Xmax=2*pi; % Limite superior en X
14 Ymin=0; % Limite inferior en X
15 Ymax=2*pi; % Limite superior en X
16 Zmin=-4;
17 Zmax=4;
18 Cont=0; % Inicializacion del contador del criterio de parada
19 flag=0; % Bandera de criterio de Parada
20
21 % Definicion fronteras
22 bnd=[Xmin Xmax
23      Ymin Ymax ];
```

El segundo paso corresponde a la ubicación de las posiciones iniciales del enjambre de manera uniformemente aleatoria y a la inicialización del vector de velocidad en cero. Hay que recordar que las funciones de prueba son de dos dimensiones cada una.

Listing 6.2: Ubicación e inicialización del vector velocidad

```

1  % Ciclo para crear las posiciones iniciales
2  for i=1:ns
3      % Define las posiciones iniciales de manera aleatoria
4      X(i,:) = bnd(:,1) + rand(size(bnd,1),1) .* (bnd(:,2) - bnd(:,1));
5  end
6
7  V=zeros(ns,2); % Vector velocidad inicializado en cero

```

Como tercer paso se procede a la evaluación de la población inicial sobre la función de prueba (función básica), y se define la mejor posición local y global que para este instante son valores temporales.

Listing 6.3: Evaluación de la población inicial y definición de mínimo local temporal

```

1  %% Ciclo para evaluar la funcion
2  % Funcion prueba
3  f= @(x)10*sin(3*sqrt(x(1).^2+x(2).^2)).*exp(-sqrt((x(1)+1).^2+(x(2)-1).^2));
4  % Evalua la funcion en las posiciones iniciales
5  for j=1:ns
6      fe(j)=f(X(j,:));
7  end
8  % La mejor posicion actual es el vector de las posiciones iniciales
9  Pi=X;
10 % Se obtiene el minimo de la funcion evaluada y su posicion
11 [fpg,ipg]=min(fe);
12 % Se obtiene el minimo global inicial
13 PG=X(ipg,:);
14 FE=fe; % necesaria en la primera iteracion

```

Como cuarto paso se realiza la actualización de velocidad y posición para cada partícula, la evaluación sobre la función objetivo, la utilización del criterio de restitución de partículas y la actualización de la mejor posición local conforme a las posiciones actualizadas.

Listing 6.4: Actualización - evaluación y restitución de partículas

```

1  for k=2:contsat; % K representa el numero de iteraciones
2  Xlast=X; % Contiene la posicion anterior a la actualizacion
3  for i=1:ns
4      for j=1:2
5          % Ecuaciones de actualizacion de velocidad y posicion

```

```

6   V(i,j)=w*V(i,j) + c1*rand*(Pi(i,j)- X(i,j))+ c2*rand*(PG(1,j)- X(i,j));
7   X(i,j)=X(i,j)+ V(i,j)*dt;
8   end
9   % Criterio de restitution de particulas
10  X1=X(i,:);
11  if X1(1)<bnd(1,1)|| X1(2)<bnd(2,1)|| X1(1)>bnd(1,2)|| X1(2)>bnd(2,2)
12  X(i,:)= Xlast(i,:); % Retorna al valor anterior
13  end
14  % Valor de la funcion evaluada antes de la actualizacion
15  FElast(i)=FE(i);
16  % Actualizacion de la funcion evaluada con la nueva particula
17  FE(i)=f(X(i,:));
18  % Actualizacion local de la particula
19  if FE(i)<fe(i)
20  Pi(i,:)=X(i,:);
21  fe(i)=FE(i);
22  end
23  end

```

Finalmente, se realiza la actualización de la mejor partícula a nivel global y se procede a verificar por medio del criterio de parada si el algoritmo debe detenerse o continuar. Para el caso de no cumplirse el criterio de parada, el ciclo repetitivo del cuarto paso sigue su curso hasta que el criterio se cumpla.

Listing 6.5: Actualización del mínimo global y criterio de parada

```

1  % Actualizacion Global
2  [FPG,ipg]=min(fe);
3  if FPG<fpg
4  PG=X(ipg,:); % Se obtiene el minimo global
5  fpg=FPG;
6  end
7  % Guardo los mejores valores de la funcion evaluada
8  F_PG(k)=FPG;
9  %F_E(:,k)=FE;
10 B=FE;
11 C=FElast;
12 D=abs(C-B);
13 maximo=max(D);
14 % Criterio de Parada
15 % verifica si el algoritmo debe detenerse
16 [Cont,flag]=Chk_Out_Camilo(FPG,F_PG(k-1),e1,e2,Cont,maximo);
17 if flag ~= 0,
18 break;
19 end
20 end
21 % Esta es la mejor particula
22 PG
23 Cont
24 k
25 fpg

```

Criterio de restitución de partículas Este criterio hace referencia a la decisión que debe tomar el algoritmo para los casos donde la partícula se sale del espacio de solución a causa de la actualización de posición y velocidad de la misma. Dicha solución consiste en preguntar si la ubicación actual de la partícula sobrepasa cualquier límite demarcado por las fronteras de la función. En caso de que ocurra, lo que hace es reubicar dicha partícula a la última ubicación permitida dentro del espacio de solución. Si al realizarse la actualización la partícula aún sigue dentro del espacio de solución, el criterio no modifica las componentes de velocidad y posición de la misma.

Listing 6.6: Criterio de restitución de partículas

```

1 % Criterio de restitucion de particulas
2 X1=X(i, :);
3 if X1(1)<bnd(1, 1) || X1(2)<bnd(2, 1) || X1(1)>bnd(1, 2) || X1(2)>bnd(2, 2)
4 X(i, :)= Xlast(i, :); % Retorna al valor anterior
5 end

```

Criterio de parada Este criterio hace referencia a la verificación que debe hacer el algoritmo para decidir si se detiene o si sigue iterando. Dicho criterio requiere dos cosas: primero garantizar que el enjambre halla convergido. Para esto se debe cumplir que la máxima diferencia de las partículas en la iteración actual con respecto a la anterior, este por debajo de un umbral denominado ϵ_2 . Cumplida esta condición se pregunta si la diferencia entre el mínimo global actual y el anterior esta por debajo de un segundo umbral denominado ϵ_1 . En caso de ser correcto, se hace un conteo que de ser cinco veces seguidas, el criterio tomará la decisión de detener el algoritmo.

De manera más resumida, el criterio garantiza que el enjambre halla convergido en un punto, y que la obtención del mínimo global se mantenga constante o con una variación mínima, en 5 iteraciones consecutivas.

Listing 6.7: Criterio de parada

```

1 function [Cont, ExFlag]=Chk_Out_Camilo (FPG, FPGLast, e, ...
2 e2, Cont, max)
3 ExFlag = 0; % Seguir iterando
4 if max < e2
5     if abs(FPG-FPGLast) <= e %abs(finall) == abs(final2)
6         Cont=Cont+1;
7     else
8         Cont=0;
9     end
10    if Cont==5;
11        ExFlag = 1;

```

```
12     end
13 end
```

Cabe resaltar que el paso a paso de la obtención del código principal y de los criterios de restitución y de parada para la implementación sobre funciones de prueba, se presenta de manera detallada en el capítulo de apéndices y de igual manera en la memoria 3 6 donde se detalla su explicación. Teniendo en cuenta que la descripción anterior hizo referencia a la versión final del código PSO.

DESCRIPCIÓN DEL CÓDIGO GA

Antes de hacer la descripción del código de algoritmos genéticos, es oportuno mencionar que el utilizado en el seminario de investigación, se hizo a partir de modificaciones del código encontrado en [4]. A continuación se hace la descripción del código de algoritmos genéticos utilizado en este seminario.

El primer paso consiste en la creación de variables junto con la definición de la función objetivo que en este caso corresponde a la función de prueba básica.

Listing 6.8: Creación de variables y definición de la función objetivo

```
1  clc, clear all, close all
2  format long
3  syms x y
4
5  % Main
6  popuSize=2;      % Poblacion inicial
7  xover_rate=0.7; % Tasa de cruce
8  mutaterate=0.01; % Tasa de mutacion
9  bit_n=16;       % Numero de bits para cada dimension
10 limit=0;       % Contador del criterio de parada
11
12 % Fonteras de la funcion objetivo
13 Xmin=0;        % Limite inferior en X
14 Xmax=2*pi;    % Limite superior en X
15 Ymin=0;       % Limite inferior en Y
16 Ymax=2*pi;    % Limite superior en Y
17
18 var_n=2;      % Numero de dimensiones de entrada
19 range=[Xmin,Xmax;Ymin,Ymax]; % Rango de la entradas
20
21 % Funcion objetivo
22 z=10*sin(3*sqrt(x.^2+y.^2)).*exp(-sqrt((x+1).^2+(y-1).^2));
```

El segundo paso consiste en la creación de la población inicial de manera aleatoria, de modo que se defina la ubicación inicial para cada individuo dentro del espacio de solución. Cabe recordar que la ubicación de cada individuo se define por medio de la cadena binaria a decimal.

Listing 6.9: Creación de la población inicial

```

1 % Creacion de la poblacion inicial
2 popu=rand(popuSize,bit_n*var_n)>0.5;

```

El tercer paso consiste en la evaluación de los nuevos individuos sobre la función objetivo. Para realizar esta evaluación, es necesario realizar una conversión de binario a decimal, de la cadena binaria, que permita representar las coordenadas de cada individuo dentro del espacio de solución.

Listing 6.10: Evaluación de los nuevos individuos

```

1 fcn.value=evalpopu(popu,bit_n,range,z);
2
3 %% Funcion de evaluacion de los individuos
4     function fitness=evalpopu(popu,bit_n,range,obj_fcn)
5     global count      % Contador global de individuos
6     pop_n=size(popu,1);
7     fitness=zeros(pop_n,1);
8     % Evaluacion individuo por individuo
9     for count=1:pop_n
10        fitness(count)=evaleach(popu(count,:),bit_n,range,obj_fcn);
11    end
12
13 %% Funcion de evaluacion de cada individuo
14     function out=evaleach(string,bit_n,range,obj_fcn)
15
16     % string: cadena de bits que representa un individuo
17     var_n=length(string)/bit_n;
18     input=zeros(1,var_n);
19     % Ciclo para convertir cada individuo de bits a decimal
20     for i=1:var_n
21        input(i)=bit2num(string((i-1)*bit_n+1:i*bit_n),range(i,:));
22    end
23     out=feval(z,input); % Regresa el valor de salida
24
25 %Funcion para convertir la cadena de bits a numero decimal
26     function num=bit2num(bit,range)
27     integer=polyval(bit,2);
28     num=integer*(range(2)-range(1))/(2^length(bit)-1)+range(1);

```

El cuarto paso consiste en la creación de la nueva población por medio de las estrategias de selección, cruce y mutación. De igual manera se presenta el criterio de parada utilizado para detener el algoritmo cuando se considere necesario.

Listing 6.11: Creación de la nueva población

```

1  popu=nextpopu (popu, fcn_value, xover_rate, mutata_rate, k);
2
3  function new_popu=nextpopu (popu, fitness, xover_rate, mut_rate, k)
4      new_popu=popu; % Copia de la poblacion actual
5      popu_s=size (popu, 1);
6      string_leng=size (popu, 2);
7
8  %% Seleccion por Elitismo: Se conservan los mejores 2 individuos
9      tmp_fitness=fitness;
10     [a, index1]=min (tmp_fitness); % Encuentra el mejor
11     tmp_fitness (index1)=max (tmp_fitness);
12     [a, index2]=min (tmp_fitness); % Encuentra el segundo mejor
13     new_popu ([1 2], :)=popu ([index1 index2], :);
14
15     total=sum (fitness);
16     cum_prob=cumsum (fitness); % Probabilidad acumulada
17
18 % Seleccion y Cruzamiento
19 for i=2:popu_s/2; % conserva los individuos padres
20
21 % Se seleccionan dos padres de acuerdo con su nivel de ajuste
22     tmp=find (cum_prob-rand>0);
23     parent1=popu (tmp (1), :);
24     tmp=find (cum_prob-rand>0);
25     parent2=popu (tmp (1), :);
26
27 % Se determina si se cruza o no
28     if rand<xover_rate
29         % Operacion de cruce
30         xover_point=ceil (rand*(string_leng-1));
31         new_popu (i*2-1, :)=...
32             [parent1 (1:xover_point) parent2 (xover_point+1:string_leng)];
33         new_popu (i*2, :)=...
34             [parent2 (1:xover_point) parent1 (xover_point+1:string_leng)];
35     end
36 end
37
38 % Mutacion (Los elites no se mutan)
39 mask=rand (popu_s, string_leng)<mut_rate;
40 % Se restauran los miembros elite despues de la mutacion
41 new_popu ([1 2], :)=popu ([index1 index2], :);

```

Como se indica en la descripción del pseudocódigo del algoritmo, este proceso se repite desde el tercer paso donde se evalúa la nueva población en la función objetivo. El criterio de parada que tiene lugar en el ciclo repetitivo que encierra los pasos 3 y 4, consiste en tomar la decisión de detener el algoritmo, únicamente para el caso donde el mejor individuo se repita 20 generaciones o iteraciones seguidas.

Listing 6.12: Criterio de parada

```

1 % Ciclo de repeticion
2 i=0; % contador del ciclo
3 % Cuando el ind. mas apto no mejora despues de 20 generaciones el GA
4 % se detiene
5 while(limit<=20)
6     i=i+1; % conteo
7     k=i;   % generacion
8
9 % se Llena el vector con los mejores individuos
10    upper(i)=min(fcn.value);
11
12    % Registro de si el ind. mas apto mejora o no
13    if (i>=2)
14        if(upper(i)==upper(i-1))
15            limit=limit+1;
16        else
17            limit=0;
18        end
19    end
20
21 end
22 [best,index]=min(fcn.value);

```

MEMORIAS

Memorial

Uso de técnicas de Optimización Global para resolver problemas de Inversión Sísmica

Edgar Téllez Cortés, Johao Coronel Silva, Juan Arias Pinto

Universidad Industrial de Santander
Escuela de Ingenierías Eléctrica, Electrónica y Telecomunicaciones

Mayo 13 de 2014

Sesión 1: Marco conceptual

1. Introducción

El objetivo general del Seminario de Investigación es estudiar y aplicar técnicas metaheurísticas en problemas de inversión sísmica. Con la aplicación de técnicas metaheurísticas se busca explorar un conjunto de posibles soluciones al problema de inversión sísmica sin utilizar punto de partida alguno. Para entender el problema que aborda este trabajo final de curso de pregrado, este documento expone la revisión de un conjunto de conceptos en temas como geofísica, exploración sísmica, problema directo, problema inverso e inversión sísmica.

2. Conceptos Básicos

Estos conceptos básicos en geofísica, problema directo, problema inverso e inversión sísmica se utilizan para entender el problema que se plantea desarrollar en el seminario de investigación. Así mismo, entender estos conceptos permite interpretar los resultados obtenidos durante cada una de las etapas que componen este trabajo final de curso de pregrado.

2.1. Geofísica

La geofísica es una ciencia que estudia la estructura, composición y propiedades de la Tierra [1]. Es una ciencia que experimenta e interpreta conceptos y métodos relacionados con áreas de la física como acústica, electricidad, fenómenos ondulatorios, magnetismo y termodinámica. Se divide en dos áreas de conocimiento dependiendo de si los estudios se realizan al interior de la Tierra mediante el uso de pozos (geofísica interna) o si los estudios se realizan en la superficie terrestre (geofísica externa).[2]

2.2. Sismología

Es la ciencia que estudia la propagación de ondas elásticas al interior de la Tierra. Hace parte de la geofísica interna y estudia los mecanismos de propagación de las ondas elásticas generadas naturalmente debido a fenómenos como los terremotos, cambios en la presión atmosférica y aquellas que se generan utilizando cargas explosivas de bajo poder a poca profundidad, como las actividades mineras, lanzamiento de cohetes y el uso continuo de vías ferreas, como lo explica [3]. Estas ondas elásticas afectan la estructura de los materiales bajo la corteza terrestre induciendo deformidades en los mismos. La intensidad con que se deforman los materiales que componen esta corteza se mide con un sismógrafo [4]. La estructura básica de este instrumento consta de un sistema conformado por una masa sujeta por un resorte, el cual se acopla a una base. Cuando un evento sísmico se presenta, la base transmite parte de este movimiento al sistema de masa y resorte, el cual oscila según la intensidad y este movimiento se registra mediante un marcador que imprime sobre un tambor de papel. en la figura 1, se observa un esquema para este instrumento.

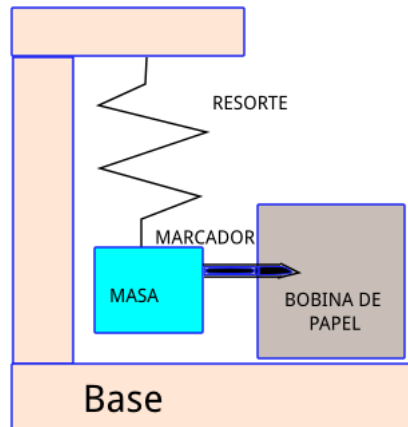


Figura 1: Sismógrafo básico

La información suministrada por los sismógrafos se utiliza para entender cómo está conformada la corteza terrestre. Existen procedimientos que mediante la generación de sismos de forma artificial facilitan la interpretación de esta información. Uno

de estos procedimientos se denomina exploración sísmica.

3. Exploración Sísmica

Es el proceso que permite identificar la estructura bajo la corteza terrestre mediante el uso de fuentes generadoras de ondas elásticas y sensores denominados geófonos. El objetivo de la exploración sísmica es entender cómo se conforma una porción de la corteza terrestre de muestra [5]. La información se recoge a través de los geófonos y corresponde a trazas sísmicas. Las trazas sísmicas corresponden a una representación gráfica del recorrido que hace una onda elástica ó mecánica que se propaga dentro de los materiales que componen esta corteza denominada también como subsuelo. Un ejemplo de esta representación se observa a continuación en la figura. 2. El eje horizontal representa el número de traza a observar y el eje vertical representa la profundidad alcanzada por la onda sísmica generada.

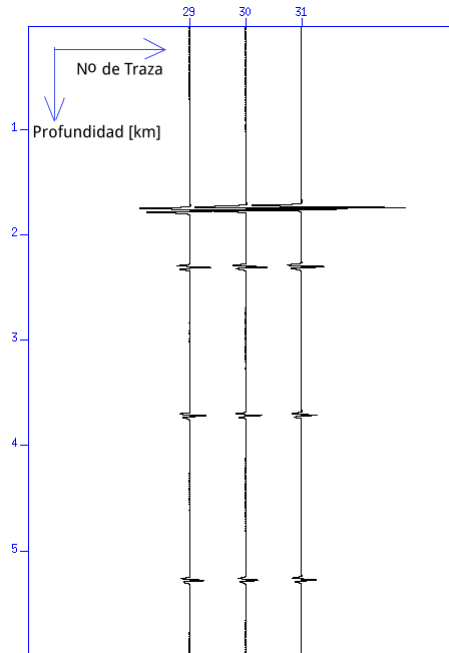


Figura 2: Traza sísmica

3.1. Exploración mediante trazador de rayos

Hasta el momento se ha mencionado que el proceso de exploración sísmica implica la detección de ondas ó frentes de onda producto del uso de generadores de sismos. Otra forma de registrar la información generada a partir de éste proceso consiste en la

detección de una parte de los frentes de onda conocidos como rayos de onda los cuales son generados por un trazador de rayos [6]. Una de las ventajas de utilizar un trazador de rayos es que se puede simular por computador con un costo computacional bajo respecto del uso de los frentes de onda. Para comprender cómo funciona el trazado de rayos observe la figura 3.

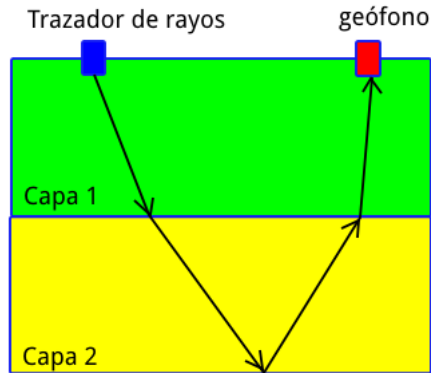


Figura 3: Trazado de rayos

La fuente de color azul, genera un rayo de color negro que se propaga a través de las capas 1 y 2 de subsuelo, éste rayo se devuelve hacia la primera capa donde se registra su llegada a través del geófono de color rojo. Matemáticamente se representa este fenómeno de trazado de rayos mediante la expresión matemática conocida como la ecuación eikonal [6]. El principio matemático de propagación de rayos a través de medios representado por ésta ecuación, es el método que se utiliza para el desarrollo del seminario de investigación.

$$(\partial_x \Theta)^2 + (\partial_y \Theta)^2 + (\partial_z \Theta)^2 = \frac{1}{u^2}, \quad (1)$$

donde:

Θ : Representa los frentes de onda (rayos generados por el trazador).

u : Representa la velocidad de propagación de los frentes (rayos) de onda en cada capa.

4. Problema Directo y Problema Inverso

Un problema directo consiste en la descripción del comportamiento de un fenómeno físico particular, a partir de un modelo matemático cuyos parámetros son conocidos. En este sentido, el modelo matemático está completo y para describir el comportamiento

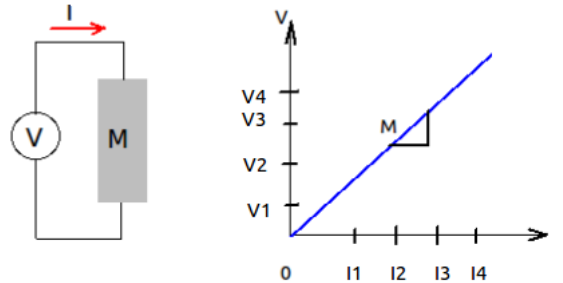
de dicho modelo, basta con replicar mediante experimentos sencillos el modelo. Los experimentos permiten generar un conjunto de datos denominados datos modelados, los cuales se utilizan para representar mediante una gráfica, el comportamiento observado de cada experimento. Un ejemplo de problema directo corresponde a la relación de la tensión eléctrica presente entre los terminales de un trozo de alambre conductor de electricidad y la corriente eléctrica que lo atraviesa. El sistema de este problema se muestra en la figura 4 (a). La relación anterior se puede expresar matemáticamente mediante

$$V = I \times M, \quad (2)$$

donde:

- V : Es el nivel de tensión presente entre los terminales de un trozo de alambre conductor de electricidad.
- I : Es la corriente eléctrica que fluye por el trozo de alambre conductor.
- M : Representa un parámetro denominado resistencia eléctrica, constante para el trozo de alambre utilizado.

A partir de la ecuación (2) se puede conocer la relación entre la tensión y la corriente eléctrica, en vista de que el valor del parámetro M es conocido. Por tal razón, es posible observar el comportamiento del fenómeno físico bajo consideración asignando valores tabulados para la variable independiente I . La figura 4 (b) muestra la relación entre la tensión y la corriente sobre el alambre conductor.

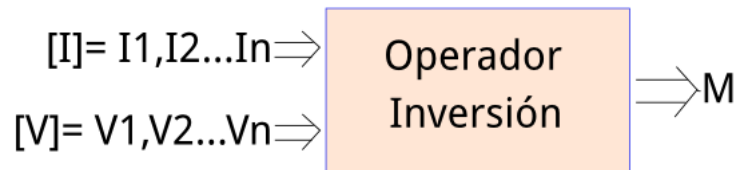


(a) Sistema del problema directo. (b) Relación entre la tensión y la corriente para un alambre conductor M conocido.

Figura 4: Esquema modelado problema directo.

Un problema inverso consiste en la búsqueda de los parámetros de un modelo matemático dado, que mejor se ajusten a los datos observados para más entradas definidas [7]. De esta manera, es posible encontrar el conjunto de parámetros que mejor ajustan

el modelo al comportamiento real del fenómeno físico. Un ejemplo de problema inverso corresponde a la búsqueda del valor del parámetro M , a partir del registro de datos experimentales de la corriente eléctrica I que circula por un trozo de alambre, y la tensión entre los extremos del mismo. La figura 5 muestra una representación gráfica del proceso realizado para la obtención del parámetro M a partir de los datos observados.



(a) Esquema problema inverso

Figura 5: Esquema para la obtención de parámetros M , a partir de datos observados $[V]$ e $[I]$.

5. Inversión Sísmica

Es un problema inverso que permite determinar los parámetros de los modelos matemáticos que representan las propiedades del subsuelo a partir de un conjunto de datos observados mediante el desarrollo de experimentos sísmicos. Se trata de identificar las propiedades elásticas de las capas de roca a partir de mediciones sobre las amplitudes de ondas acústicas en función del tiempo y de la distancia recorrida. Existen otros métodos geofísicos que complementan los estudios realizados utilizando métodos sísmicos. Se hace referencia de acuerdo con lo planteado por [7] a la medición de resistividad aparente en ubicaciones previamente establecidas, lo que se conoce como **magnetotelúrica** que identifica las propiedades eléctricas de los materiales que conforman el subsuelo. Existen dos métodos de inversión sísmica, según si el punto de partida para la reconstrucción de datos se basa en los datos adquiridos o si se utiliza un modelo previamente elaborado. En el caso de los métodos de inversión directa, se diseña un operador matemático que se aplica directamente sobre los datos observados para reconstruir un modelo matemático. En el caso de los métodos de inversión basada en el modelo, no se intenta reconstruir directamente un modelo con base en los datos obtenidos. Se parte de un modelo matemático asumido para el cual se replica un experimento físico que genera un conjunto de datos modelados que se comparan con los datos observados, hasta que la diferencia entre ellos sea mínima.

6. Análisis y Discusión

Los avances en la implementación de nuevos métodos geofísicos aplicados en áreas como el diseño, construcción de obras de infraestructura, detección y desarrollo de actividades de extracción de minerales, constituye una de las áreas que ha impulsado el desarrollo de la industria desde hace una década. En este escenario, áreas de la ingeniería como el procesamiento de señales juegan un papel importante en el momento de tomar decisiones que impulsen el avance de la investigación en la geofísica, la reducción de riesgos y costos, así como el incremento en la producción de bienes y servicios.

El desarrollo del seminario de investigación denominado *Uso de Técnicas de Optimización Global para resolver problemas de Inversión Sísmica*, se enfoca en esta línea de investigación. Dentro de los objetivos propuestos durante el desarrollo de esta modalidad de Trabajo de Grado, se encuentra el estudio, implementación y análisis de métodos de optimización global, que permitan reconstruir modelos del subsuelo a partir del uso de métodos geofísicos que simplifican la adquisición de datos, utilizando experimentos físicos basados en la Sísmica de Reflexión.

Es importante resaltar además que dentro del marco del respeto al medio ambiente y con el objetivo de reducir los costos del desarrollo de las actividades previstas durante el seminario, se ha escogido el uso de herramientas tecnológicas instaladas sobre ordenadores de bajo costo, utilizando licencia para sistemas operativos UNIX. Se realiza la simulación consistente en la adquisición de datos y reconstrucción de un modelo de referencia sintético, el cual se utilizará para compararlo con un modelo generado a partir de datos teóricos, el cual se ajustará utilizando dos métodos de optimización global, definidos por el grupo de exploración.

7. Conclusiones

El desarrollo del seminario de investigación comienza con la introducción de un conjunto de conceptos básicos en geofísica, que permiten la interpretación de cada uno de los aspectos relacionados con la reconstrucción de modelos del subsuelo, que constituye el tema central a tratar. El método que se utiliza para determinar la estructura y composición de los materiales subyacentes se denomina inversión sísmica basada en el modelo. Es un método que a partir de un conjunto de datos adquiridos mediante la implementación de experimentos sísmicos, genera un modelo de referencia que contiene las propiedades observadas en las rocas que existen bajo la corteza terrestre. Otras características presentes se incluyen mediante la comparación de dicho referente, con un modelo teórico basado en la medición de parámetros eléctricos y magnéticos obtenidos normalmente con el método de registro de pozos, la comparación repetitiva de estos modelos completa el modelo teórico con los datos observados utilizando sísmica de re-

flexión. Es en este punto donde intervienen las técnicas de optimización global, junto con las métricas que facilitan la comparación entre modelos. Se pretende obtener la menor diferencia posible entre ellos asegurando de este modo, un modelo de la corteza subyacente que incluya la mayor cantidad de fenómenos físicos existentes en el mundo real.

El uso de herramientas tecnológicas para la recolección de datos y el modelado, permite no solo adquirir datos de alta calidad, además permite la repetición de los experimentos que incluyen el uso de figuras de mérito o de comparación de acuerdo con la forma como se genere la distribución de los datos teóricos utilizados en la construcción de nuevos y mejores modelos ajustados al referente obtenido con base en los datos observados.

Referencias

- [1] ESTRADALuis. Apuntes de sismología. <http://www.herrera.unt.edu.ar/geofisica/Sismologia%20para%20Ingenieros.pdf>, 2012. Revisado: Febrero 2 de 2014.
- [2] Wikipedia. Geofísica. <http://es.wikipedia.org/wiki/Geofisica>.
- [3] Thorne Lay and Terry C Wallace. *Modern global seismology*, volume 58. Academic press, 1995.
- [4] consorcio IRIS. Cómo trabaja un sismógrafo? https://www.iris.edu/hq/files/publications/brochures_onepagers/doc/SP_1pager_7.pdf, 2008. Revisado: Febrero 2 de 2014.
- [5] Mauricio TRUJILLO. Exploración sísmica. <http://exploracionsismica.blogspot.com/>.
- [6] Mario CAICEDO and MORA Placido. Temas de propagación de ondas - universidad simón bolívar. <http://www.fis.usb.ve/~mcaicedo/geophysics/ondas.pdf>.
- [7] Sen MRINAL and STOFFA Paul. *Global Optimization Methods in Geophysical Inversion*. Elsevier, 1995.

Memoria2

Uso de técnicas de Optimización Global para resolver problemas de Inversión Sísmica

Edgar Téllez Cortés, Johao Coronel Silva, Juan Arias Pinto

Universidad Industrial de Santander
Escuela de Ingenierías Eléctrica, Electrónica y Telecomunicaciones

Mayo 21 de 2014

Sesión 2: Técnicas de Optimización

1. Introducción

Las técnicas de optimización se utilizan cuando no es posible utilizar métodos analíticos para el tratamiento de problemas complejos. Existe una gran variedad de técnicas de optimización entre las que se encuentran las técnicas directas, las técnicas heurísticas y las más utilizadas actualmente, las metaheurísticas. La mayoría de las técnicas metaheurísticas se basan en procesos naturales, como el comportamiento de las bandadas de aves, otras se basan en procesos evolutivos y algunas, recurren a comportamientos aprendidos por grupos particulares de seres vivos. En este documento se definen previamente conceptos necesarios para entender en qué consiste la optimización, los tipos de técnicas de optimización que se conocen y posteriormente se expone la primera técnica de optimización que utilizará el seminario de investigación para tratar el problema de inversión sísmica: Optimización por enjambre de partículas.

2. Conceptos Previos

Las definiciones que se exponen a continuación permiten entender cómo la optimización utiliza diferentes técnicas y estrategias para tratar problemas de diversos niveles de complejidad. Cada una de los tipos de técnicas definidas a continuación, se aplica en casos específicos. Dentro de los criterios para la selección de una u otra técnica se encuentra al nivel de complejidad del proceso a tratar, la rapidez con que desea encon-

trarse una respuesta y si el proceso que se trata puede ser tratado analíticamente.

2.1. Optimización

Es el área de las matemáticas que mediante el diseño e implementación de algoritmos, selecciona el mejor elemento contenido dentro de un conjunto de posibles candidatos de acuerdo con un criterio definido [1]. La selección de estos mejores elementos se realiza con el objeto de mejorar un proceso o resolver problemas como la maximización o minimización de funciones matemáticas.

2.2. Técnicas de Optimización

Son conjuntos de pasos ordenados que incorporan herramientas de análisis y búsqueda para mejorar el desempeño de un proceso. Algunas de las técnicas de optimización de uso frecuente, recurren a la comparación repetitiva entre varias soluciones alternativas, las cuales se complementan entre sí para obtener una mejor solución. Dependiendo del tipo de proceso a mejorar, se recurre al uso de herramientas y operaciones matemáticas, adaptación e implementación de procesos presentes en fenómenos naturales y procesos de búsqueda dirigida. De acuerdo con estos criterios, las técnicas de optimización se clasifican como: técnicas directas, técnicas heurísticas y técnicas metaheurísticas.

2.2.1. Técnicas Directas

Son conjuntos de pasos ordenados que utilizan operadores matemáticos para encontrar una respuesta. Estas técnicas no exploran el espacio de posibles soluciones, en su lugar, utilizan uno o varios operadores matemáticos que se aplican sobre la función objetivo ó proceso a mejorar. La respuesta que se obtiene corresponde al resultado de la correcta aplicación de estos operadores y relaciones matemáticas.

2.2.2. Técnicas Heurísticas

Es una metodología o conjunto de pasos ordenados, basados en el uso de reglas empíricas para la búsqueda de una respuesta aproximada a un problema en particular. Las técnicas heurísticas, a pesar de proveer un alto grado de confianza en su solución, no garantizan como óptimo su resultado. Por tal razón es fundamental evaluar la calidad de la solución por medio del análisis del peor de los casos, o contrastando con otros métodos de optimización o cotas que indiquen la solución del problema. De igual manera es importante conocer los casos en que son aplicables y los límites de su uso. El uso de estas técnicas permite dar solución a problemas donde los métodos exactos

tienen un alto costo computacional. [2]

2.2.3. Técnicas Metaheurísticas

Al igual que las técnicas heurísticas, éstas consisten en un conjunto de pasos ordenados, inspirados basan en la experiencia, para encontrar una respuesta que si bien no constituye una solución definitiva, se considera como una alternativa o indicio hacia la solución. La diferencia con las heurísticas radica en que éstas se diseñan con el propósito de mejorar el rendimiento de las mismas, ampliando su aplicación a los problemas que implican el uso separado de varias heurísticas. Son técnicas de propósito general creadas a partir de la integración de áreas como: genética, biología, matemática, física, inteligencia artificial, redes naturales, entre otras. Entre las técnicas metaheurísticas más conocidas, se destacan las técnicas inspiradas en la naturaleza, basadas en una población, basadas en la trayectoria, técnicas de búsqueda local y con memoria [?], [3].

2.3. Algunas técnicas metaheurísticas conocidas

Dentro de esta categoría se encuentran métodos de optimización como los algoritmos genéticos, recocido simulado, búsqueda tabú, supernova (conformación de estrellas y galaxias en el universo), colonia de hormigas, gotas de agua inteligente. A continuación, se explica el método de Optimización basado en *Enjambres de Partículas (Particle Swarm Optimization)*, que constituye uno de los métodos de interés para el desarrollo del seminario de investigación.

3. Optimización mediante Enjambres de Partículas

Es un método inspirado en el comportamiento de bandadas de aves, bancos de peces y grupos de abejas, cuando buscan alimento. En analogía con la naturaleza, el algoritmo inicia con un grupo de posibles soluciones denominado enjambre de partículas, las cuales mediante variaciones de posición y velocidad se desplazan dentro del espacio solución en búsqueda del área donde se ubica el valor deseado. Para el desarrollo del seminario, el valor deseado corresponde a la obtención de un mínimo global [4].

3.1. Relación con la Naturaleza

Un grupo de abejas que sale en busca del área donde se localiza la mayor cantidad de alimento, constituye un ejemplo natural de optimización mediante enjambres de partículas. Cada abeja se desplaza dentro del área donde se encuentra el alimento, establece que tanta cantidad de alimento ha encontrado e informa a sus compañeras

de su hallazgo. Además de controlar su rumbo individual, permanece en alerta a los rumbos tomados por sus compañeras, a fin de conocer si han encontrado mayor o menor cantidad de alimento en relación con la cantidad encontrada por ella misma [4].

Del mismo modo, una nube de partículas que constituyen las posibles soluciones a un problema particular, se desplazan dentro del espacio solución del mismo. En su recorrido, cada partícula es evaluada por una función de adaptación que determina para la posición actual de cada partícula, el nivel de adaptación al entorno; en otras palabras, determina qué tan lejos se encuentra de la respuesta esperada. En función del nivel de adaptación, las partículas se desplazan a mayor o menor velocidad y cuando una partícula ha alcanzado la región donde se ubica el valor deseado, el resto del enjambre orienta su posición hacia dicha región. Las partículas no están exentas de caer en valores locales; sin embargo, esta situación contribuye a la exploración de la mayor área posible, a fin de descartarlos. Se busca en estas circunstancias, que el algoritmo mantenga un equilibrio entre la exploración y el reconocimiento del área solución.

3.2. Parámetros

Los parámetros utilizados en la implementación se utilizan para ajustar la posición y la velocidad de las partículas, así como para definir cómo se comporta el algoritmo; es decir, el ajuste de estos parámetros, determina tanto el nivel de exploración de todo el espacio de búsqueda, como la velocidad con que se puede encontrar la respuesta. A continuación, se describen estos factores y su influencia.

3.2.1. Peso de Inercia

La masa de inercia w se emplea para controlar el impacto de la historia previa de las velocidades en la velocidad de la corriente del enjambre, lo que influye en el equilibrio entre la exploración global y local.

3.2.2. Radio de Avance

Se define como el tamaño del paso dt que utilizan las partículas para avanzar dentro del espacio de búsqueda. Un radio de avance muy alto puede permitir la salida inesperada de las partículas del espacio de búsqueda o espacio solución.

3.2.3. Constantes de Exploración Local y Global

Son los valores que definen el nivel de profundidad en la exploración tanto de las regiones donde se puede ubicar la respuesta deseada, como la capacidad de exploración

de todo el espacio solución. De acuerdo con la bibliografía consultada [5], se debe establecer una relación de equilibrio entre estos dos factores, para evitar problemas de convergencia prematura y sobreexploración.

3.3. Configuraciones Adicionales

El algoritmo basado en Enjambres de Partículas en su estructura básica, contempla únicamente el uso de los parámetros descritos anteriormente; sin embargo, cuando se requiere la implementación sobre problemas o procesos caracterizados por su elevada complejidad, se dispone de algunas configuraciones adicionales que facilitan el proceso de exploración de las partículas y el alcance rápido de una respuesta. Dentro de las opciones de configuración disponibles, se destacan los entornos y las topologías.

3.3.1. Entornos

Dentro de los aspectos a considerar en los algoritmos basados en acumulación de partículas, se destaca la interacción que presenta una partícula frente a las demás que se ubican cerca de un vecindario en particular. De acuerdo con lo enunciado en [5], las topologías definen el entorno de interacción de una partícula con su vecindario, estableciéndose una relación de dependencia entre la partícula y el entorno donde se encuentra. De acuerdo con lo explicado anteriormente, el entorno se clasifica en dos tipos.

Entorno Geográfico : Consiste en calcular la distancia entre la partícula a evaluar y las demás. Se toman posteriormente las partículas cuya distancia es más corta, para componer el entorno de la partícula a evaluar.

Entorno Social : Consiste en definir inicialmente una lista de partículas cercanas a la que se requiere evaluar, independientemente de su ubicación en el espacio.

Un ejemplo donde se observa el concepto de entornos, se encuentra en la figura 1 donde las partículas que se encuentran cercanas son encerradas dentro de un entorno geográfico y las partículas que se encuentran alejadas entre sí, conforman un grupo demarcado como entorno social, ya que si bien es cierto que están alejadas, existe un medio de comunicación entre ellas.

3.3.2. Criterios para la selección de un entorno

La exploración global se puede mejorar, si se elige además de un valor adecuado para el peso de inercia, un adecuado entorno. La selección de un entorno adecuado, depende del tipo de problema bajo consideración, algunos criterios pueden contribuir en este sentido.

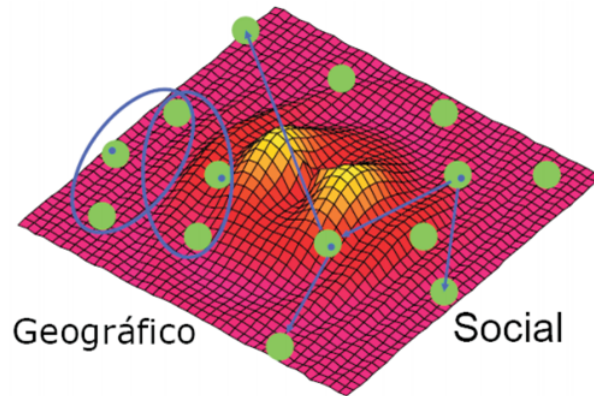


Figura 1: Entornos Geográfico y Social. Fuente: [5]

- La selección del entorno viene acompañada del tamaño del mismo. Si se elige como tamaño del entorno a todo el enjambre, se considera un entorno PSO global, en cuyo caso se define un entorno global como geográfico y social al mismo tiempo.
- Es posible además definir varios entornos dentro de un mismo enjambre dependiendo del tamaño, normalmente se elige un valor comprendido entre 3 y 5, en cuyos casos el comportamiento es bueno.

3.3.3. Topologías

El algoritmo PSO puede incurrir con facilidad en situaciones de convergencia prematura, dependiendo de la asignación inicial del punto de mejor posición denominado PG. Sin embargo se han visto variantes en la manera como se maneja la topología del enjambre, existen varios tipos de topologías utilizadas junto a los entornos para cubrir todo el espacio solución.

Sociometría Gbest : Se establece un vecindario global en el que cada una de las partículas es vecina de la totalidad del enjambre [5], la ventaja de esta topología es que facilita la explotación del espacio de soluciones ya que abarca toda la región bajo análisis.

Sociometría Lbest : Cada partícula se conecta a su vecina inmediata formando un anillo de partículas, la ventaja que presenta esta topología es que se pueden establecer subcúmulos que realicen la búsqueda en diferentes regiones del espacio problema, favoreciendo la exploración.

Topología de Von Neumann : Es un arreglo que consiste de una matriz rectangular en la que cada partícula se conecta con las partículas ubicadas en las regiones superior, frontal, lateral izquierda y lateral derecha, solapando los bordes de manera toroidal.

En la figura 2 se observa la estructura interconectada propia de la sociometría Gbest, cada una de las partículas establece comunicación con sus vecinas dentro del total de partículas lanzadas en el espacio de búsqueda.

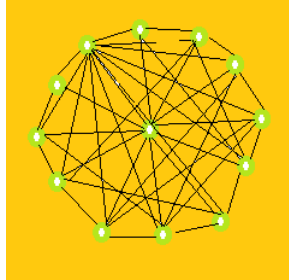


Figura 2: Topología Gbest. Fuente: Autores.

La topología lbest representa una estructura en forma de anillo en la cual, las partículas vecinas se unen una a lado de la otra. Este comportamiento se observa en la figura 3.

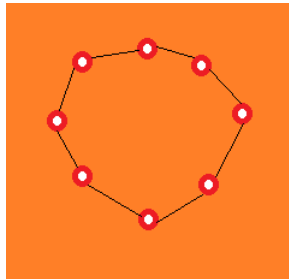


Figura 3: Topología Lbest. Fuente: Autores.

Una propuesta de tipo matricial es el desarrollo de la topología de Von Neumann también conocida como topología cuadrada, en la cual se disponen arreglos de partículas por filas y columnas; al mismo tiempo, los arreglos se conectan entre sí por sus bordes superior, inferior y laterales. Este arreglo se observa en la figura 4.

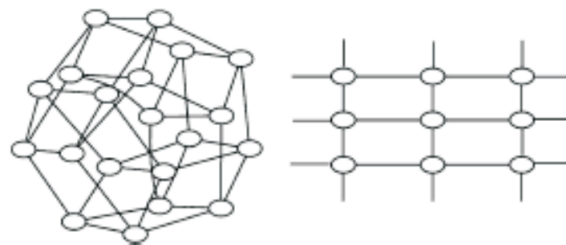


Figura 4: Topología de Von Neumann. Fuente: [5]

3.4. Pseudocódigo PSO

El pseudocódigo del algoritmo de optimización basado en Enjambres de Partículas se compone de las siguientes partes.

I) Declaración de las Variables Requeridas.

- X_i = Posición actual de la partícula i .
- V_i = Velocidad actual de la partícula i .
- P_i = Mejor posición local conocida de la partícula i .
- G = Mejor posición global conocida.
- Criterio de finalización.

II) Desarrollo del algoritmo.

Aplicable a cada partícula.

- Inicializar la posición mediante una distribución de probabilidad uniforme.
- Inicializar la posición local más conocida de la partícula a la posición inicial de la partícula.

$$P_i = X_i \tag{1}$$

- Si $F(P_i) < F(G)$, actualizar la posición más conocida de la partícula.

$$G = P_i \tag{2}$$

- Inicializar la velocidad de las partículas.
- Hasta que se cumpla el criterio de terminación (especificado por el usuario), se repite el siguiente procedimiento para cada partícula.

1. Se eligen números al azar, denominados R_p y R_g , con valores asignados entre cero y uno.

2. Para cada dimensión, actualizar la velocidad, mediante las expresiones:

$$\vec{V}_i = (w * \vec{V}_i) + c_1 * rand * (\vec{P}_i - \vec{X}_i) + c_2 * rand * (\vec{G} - \vec{X}_i), \quad (3)$$

donde:

w : Peso de inercia que controla el desplazamiento de las partículas.

c_1 y c_2 : Hace referencia a los pesos que se incluyen en la ecuación de actualización de velocidad para darle mayor importancia a la búsqueda local o global del enjambre. Se definen mediante las variables c_1 para búsqueda local y c_2 para búsqueda global.

\vec{P}_i : Mejor posición individual.

\vec{G} : Mejor posición de grupo.

La ecuación de actualización de posición para cada partícula esta definida por

$$\vec{X}_i = \vec{X}_i + \vec{V}_i * dt, \quad (4)$$

donde:

dt : Radio de avance, entendido como el paso que utilizan las partículas para avanzar dentro del espacio de búsqueda. Un valor adecuado permite mantener las partículas dentro del espacio de búsqueda predeterminado.

\vec{V}_i : Vector velocidad actual de la i ésima partícula.

\vec{X}_i : Vector posición actual de la i ésima partícula.

3. Si se cumple que:

$$F(X_i) < F(P_i) \quad (5)$$

4. Actualizar la posición más conocida de la partícula mediante la expresión.

$$P_i = X_i \quad (6)$$

5. Si se cumple ahora la condición:

$$F(P_i) < F(G) \quad (7)$$

6. Actualizar la posición más conocida del enjambre con la expresión.

$$G = P_i \quad (8)$$

7. Ahora G tiene la mejor solución encontrada.

3.5. Expresiones Matemáticas

3.5.1. Influencia del peso de Inercia en la Velocidad de las partículas

$$v_{k+1}^i = w_k * v_k^i + C_1 * r_1 * (P_k^i - x_k^i) + C_2 * r_2 * (P_k^g - x_k^i) \quad (9)$$

3.5.2. Actualización de Posición sin cambios

$$x_{k+1}^i = x_k^i + v_{k+1}^i * (1) \quad (10)$$

3.5.3. Reducción Lineal del Peso de Inercia

Se efectúa la reducción lineal del peso de inercia en la expresión para la velocidad de las partículas.

$$v_{k+1}^i = w_k * v_k^i + C_1 * r_1 * (P_k^i - x_k^i) + C_2 * r_2 * (P_k^g - x_k^i) \quad (11)$$

Donde $w_{k+1} = \alpha * w_k$
y $0 < \alpha < 1$

3.5.4. Factor de Constricción

Consiste en el escalamiento de la expresión matemática para la velocidad de cada partícula en un factor de escala (K).

$$v_{k+1}^i = K * [w_k * v_k^i + C_1 * r_1 * (P_k^i - x_k^i) + C_2 * r_2 * (P_k^g - x_k^i)] \quad (12)$$

$$K = \frac{2}{|2 - \phi - \sqrt{\phi^2 - 4 * \phi}|} \quad (13)$$

Donde:

$$\phi = C_1 + C_2$$

$$\phi > 4$$

3.6. Ventajas del Algoritmo

- Implementación sencilla.
- Pocos parámetros del algoritmo.
- Eficiencia en búsqueda global.

3.7. Desventajas del Algoritmo.

- Convergencia prematura si el manejo de las variables es inadecuado.
- La forma más rápida de encontrar el punto solución se da cuando dicho valor se ubica en uno de los ejes del espacio de búsqueda, lo cual implica un error por sesgo.
- Débil capacidad de búsqueda local.

3.8. Aplicaciones

Debido a la analogía que el algoritmo presenta con el comportamiento de las aves en la naturaleza, la estructura básica del algoritmo basado en Enjambres de Partículas, puede aplicarse en la optimización de diversos procesos, así como en el desarrollo de nuevos métodos de optimización global, enfocados a la solución de nuevos problemas.

- Formación de redes neuronales.
- Identificación de la enfermedad de Parkinson.
- Extracción de las normas de las redes difusas.
- Reconocimiento de imagen.
- Optimización de la distribución de energía de redes eléctricas.
- Optimización estructural.
- Forma óptima y un diseño de tamaño.
- Optimización de topología.
- Proceso de la bioquímica.
- Identificación de sistemas biomecánicos.

4. Análisis y Discusión

Las técnicas metaheurísticas constituyen un avance en la investigación respecto de las heurísticas debido a su capacidad de reestructuración, lo que permite la aplicación de diversas estrategias enfocadas a la solución de problemas de diversas áreas del conocimiento y concretamente para la reconstrucción completa de modelos del subsuelo, los cuales involucran una cantidad enorme de variables asociadas a fenómenos físicos. Estas capacidades han permitido incluir estos métodos dentro de la categoría de los

más adecuados para tratar volúmenes elevados de información, utilizando pocos recursos computacionales.

Es importante aclarar que el verdadero sentido de implementar estas técnicas de optimización, no se halla en el simple ingreso de un conjunto de datos mínimos por parte del usuario y esperar el resultado final. Más allá de si el resultado obtenido es correcto o no, el sentido que representa utilizar estos algoritmos de optimización, es interpretar los resultados que este entrega en la construcción de nuevo conocimiento a partir de los ensayos realizados de las diferentes estrategias puestas a prueba. El uso inteligente de los algoritmos y la interpretación de los resultados obtenidos, permite encontrar nuevas soluciones a los problemas planteados.

5. Conclusiones

El método de optimización basado en Enjambres de Partículas, constituye una de las técnicas metaheurísticas de mayor aplicación en problemas de inversión sísmica basada en el modelo. Su relación con la naturaleza, su capacidad de exploración global y local en la obtención de un valor cercano al deseado, hace que este método facilite el desarrollo de un modelo teórico ajustado mediante la comparación con un modelo de referencia. El uso masivo de estas técnicas metaheurísticas caracterizadas por su capacidad de exploración, adaptabilidad a diversos procesos, flexibilidad en el establecimiento de criterios de terminación, ha permitido la resolución y mejoramiento de muchos procesos que anteriormente no habían podido resolverse mediante el uso de técnicas directas. El hecho de utilizar estrategias muy comunes en la naturaleza, ha simplificado el cálculo y diseño de productos, servicios y han contribuido al desarrollo de la industria, reduciendo costos y minimizando riesgos. En su estructura más sencilla, este algoritmo distribuye dentro de un espacio delimitado, un conjunto de posibles soluciones las cuales son evaluadas mediante la implementación de una función de adaptabilidad que determina el mayor o menor grado de adaptación que tienen las partículas respecto del punto solución esperado, lo que motiva su desplazamiento en todo el espacio de búsqueda aceptando en algunas oportunidades respuestas inadecuadas, con el objeto de explorar lo suficiente hasta encontrar esa solución esperada; o como sucede en la mayoría de los casos, una región cercana a la solución. Un ajuste adecuado de los parámetros utilizados, como son el peso de inercia, el radio de avance, y las constantes de exploración local y global, denominadas $C1$ y $C2$ respectivamente, define el funcionamiento adecuado de este método. La implementación del algoritmo para resolver problemas de elevada complejidad, requiere del uso de estrategias que habiliten la comunicación entre todas las partículas, tal es el caso de los entornos y de las topologías las cuales se aplican cuando los procesos a optimizar requieren de una capacidad de exploración muy superior. En las siguientes sesiones, se abordarán temas como la implementación de este algoritmo sobre funciones de prueba, los criterios de terminación, el ajuste de parámetros y el enfoque propuesto para la construcción de un

modelo ajustado a los datos adquiridos y a los fenómenos descritos teóricamente.

Referencias

- [1] Wikipedia. Optimización. [http://es.wikipedia.org/wiki/Optimizacion_\(matematica\)](http://es.wikipedia.org/wiki/Optimizacion_(matematica)). Revisado: Septiembre de 2014.
- [2] Wikipedia. Heurística. <http://es.wikipedia.org/wiki/Heuristica>.
- [3] Wikipedia. Metaheurística. <http://es.wikipedia.org/wiki/Metaheuristica>.
- [4] Wikipedia. Optimización por enjambre de partículas. http://es.wikipedia.org/wiki/Optimizacion_por_enjambre_de_particulas.
- [5] José Manuel García Nieto. Algoritmos basados en cúmulos de partículas para la resolución de problemas complejos. neo.lcc.uma.es/staff/jmgn/doc/Memoria_PFC_JMGN.pdf.

Memoria3

Uso de técnicas de Optimización Global para resolver problemas de Inversión Sísmica

Edgar Téllez Cortés, Johao Coronel Silva, Juan Arias Pinto

Universidad Industrial de Santander
Escuela de Ingenierías Eléctrica, Electrónica y Telecomunicaciones

Mayo 28 de 2014

Sesión 3: Optimización por Enjambre de Partículas

1. Introducción

En este reporte se presenta la revisión conceptual de la técnica metaheurística de optimización mediante Enjambre de Partículas, que incluye aspectos como la analogía con la naturaleza y la definición de estrategias de implementación como los entornos y las topologías. Para comprender como funciona la técnica, se describe el algoritmo utilizado así como las variables que controlan el funcionamiento del mismo. Las primeras pruebas, se realizan sobre funciones matemáticas multivariable, denominadas funciones de prueba, para las cuales se presenta su gráfica y sus características. El objetivo de éstas primeras pruebas consiste en estudiar la influencia de las variables utilizadas por la técnica, en su desempeño. Se registraron los parámetros que ajustan el desempeño del algoritmo en que se basa la técnica, mediante el uso de tablas. Éstos parámetros, son elegidos para la posterior aplicación de la técnica en el proceso de inversión sísmica.

2. Implementación del algoritmo Enjambre de Partículas *Particle Swarm Optimization* en funciones de Prueba

Para el desarrollo de las primeras pruebas del algoritmo se eligió una función matemática multivariable, caracterizada por la presencia de un sólo mínimo global

y varios mínimos locales cuya gráfica se observa en la figura 1(a). Otras características destacadas son la presencia de una planicie y algunas elevaciones que contienen puntos máximos y zonas de estancamiento dentro de las fronteras establecidas para dicha función, la planicie puede observarse en la figura 1(b) como un plano de color verde claro que ocupa buena parte del área total de exploración.

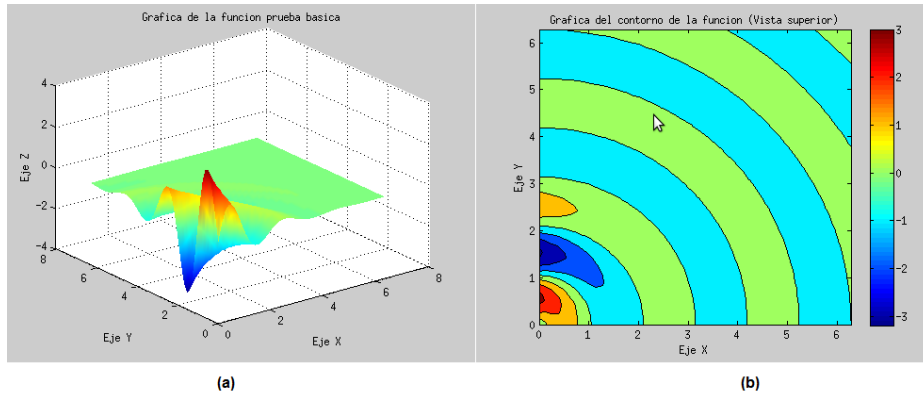


Figura 1: Función de Prueba Básica (fuente: Autores)

El nivel de complejidad para dicha función es bajo por sus características morfológicas, el desarrollo de las pruebas busca demostrar que por las características de la función se espera que al menos una partícula ubicada cerca del mínimo se encargue de comunicar su mejor posición al enjambre y dirigirlo rápidamente al punto de interés; asegurando la convergencia del enjambre de partículas.

Para el desarrollo de las pruebas se eligió un conjunto de parámetros para estudiar su influencia en el funcionamiento del algoritmo. Entre los parámetros elegidos se destaca el número de partículas (np), el número de iteraciones (ni) y el peso de inercia (W), permaneciendo constantes las variables $C1$ y $C2$ con los valores recomendados de 1,9 y 1,8 respectivamente por la bibliografía consultada [1] y [2].

2.1. Implementación en Matlab

La implementación del algoritmo de enjambre de partículas (PSO) se desarrolla en la herramienta computacional Matlab, el código completo se encuentra en la sección 5.3.1 del capítulo apéndices. El primer paso para el desarrollo de un código para este algoritmo, consiste en el borrado de variables, cierre de ventanas adicionales y limpieza de la ventana principal, conocida también como *command window*, de modo que se asegure que las variables a revisar y las animaciones por ver, sean las correspondientes a la ejecución del programa. Después de esto, se inicializa la variable **ns** que simboliza el número de partículas que van a conformar el enjambre, luego se crea una matriz que contiene los límites inferior y superior de los dos ejes coordenados correspondientes a las variables independientes de la función, de modo que se establezca el espacio solución dentro de dichas fronteras. Este código se observa en detalle en la lista denominada *listing 1*.

Listing 1: Ingreso de parametros

```

1 clear all,close all,clc % Borrar variables y limpiar pantalla
2 ns = 10; % Numero de particulas
3 bnd=[0 2*pi 0 2*pi]; % Delimitacion fronteras en ejes x e y

```

Una vez se han asignado las variables iniciales, se crea un ciclo de repetición en cual se asigna la posición inicial de las partículas de manera aleatoria con una distribución uniforme; esto se hace con ayuda de la función Seeder creada especialmente para esta tarea y que es de autoría del grupo de investigación CEMOS de la Universidad Industrial de Santander, quién nos facilitó la función debido al desarrollo de trabajos anteriores relacionados con la implementación de algoritmos de optimización por parte de este grupo. La función Seeder que se encuentra en la sección 5.3.2 del capítulo apéndices de la Tesis debe estar guardada en la misma carpeta del código del algoritmo implementado, facilitando así el llamado de la función en el código principal. Luego se inicializa la velocidad de las partículas con un valor de cero (Considerando que el enjambre inicialmente esta quieto), después se crean vectores de las variables independientes y se escribe la ecuación de la función sobre la cual se va a trabajar, esto se observa en el código mostrado en la lista llamada *listing 2*.

Listing 2: Posiciones iniciales y Funcion Objetivo

```

1 %Ciclo para crear las posiciones iniciales
2 for i=1:ns
3     X(i,:)=Seeder(bnd); % Define las posiciones iniciales
4     de manera aleatoria
5 end
6 V=zeros(ns,2); % Vector velocidad inicializado en cero
7
8 % Grafica de la funcion objetivo
9 % Genera vectores de X e Y de 0 a 2*pi con 50 muestras
10 equiespaciadas
11 [x1,y1]= meshgrid( linspace(0,2*pi,50) );
12
13 % Expresion matematica de la funcion a optimizar
14 z1= 10*sin(3*sqrt(x1.^2+y1.^2)).*exp(-sqrt((x1+1).^2+(y1-1).^2));
15 %

```

El siguiente paso consiste en graficar en tres dimensiones la función objetivo, después se evalúa la función y se crea un vector con los valores de evaluación de las partículas utilizando un ciclo de repetición. Se grafica ahora la posición de las partículas dentro de la función en una sola ventana, para visualizar la posición inicial de las mismas y saber qué tan cerca están del mínimo global y qué tan dispersas se

encuentran en el espacio solución. Luego tal y como lo indica el pseudocódigo del algoritmo PSO, expuesto en la sesión anterior, se establece el vector de mejores posiciones de cada partícula y el valor del mínimo global que corresponde a la partícula cuyo valor evaluado en la función sea menor. Seguidamente se crean y asignan los valores de las variables de actualización del algoritmo, que en este caso corresponden a las constantes de exploración asignadas a la capacidad de búsqueda local y global, identificadas como $C1$ y $C2$, respectivamente y W que indica el peso de inercia. Este código se observa en la lista *listing 3*

Listing 3: Evaluacion de particulas

```

1  for j=1:ns
2    fe(j)=f(X(j,:)) ; % Crea el vector fe con los valores de la
      funcion evaluada en las posiciones aleatorias iniciales
3  end
4
5  for u=1:ns
6    % Grafica de las particulas en la funcion objetivo
7      plot3(X(u,1),X(u,2),fe(u),'--rs','LineWidth',2,...
8          'MarkerEdgeColor','k'
9          'MarkerFaceColor','r'
10         'MarkerSize',10);
11      hold on
12  end
13
14  Pi=X ; % La mejor posicion actual es el vector de las
      posiciones iniciales
15  [fpg,ipg]=min(fe); % Se obtiene el minimo de los valores de la
      funcion evaluada y su posicion
16  PG=X(ipg,:); % Se obtiene el minimo global "Posicion en 2D"
17
18  c1=1.9; % Valor recomendado (Peso de la busqueda local)
19  c2=1.8; % Valor recomendado (Peso de la busqueda global)
20  w= 0.7; % Valor recomendado (Peso de la inercia)
21
22  FE=fe; % Se hace una copia de los valores iniciales

```

En la primera línea del código representado en la lista *listing 4*, se puede ver el comienzo de un ciclo de repetición que corresponde al número de iteraciones permitidas para la ejecución del algoritmo; dentro de dicho ciclo, existen otros dos ciclos anidados donde el primero recorre el número de partículas y el otro (interno al ciclo de partículas) recorre las dimensiones de la función implementada, con el objetivo de hacer la actualización de posiciones y velocidades de las partículas dimensión por dimensión. Adicional a esto, es necesario el uso de un criterio que restituya las partículas en los casos donde estas se salgan del espacio solución luego de una actua-

lización, por tal razón se utiliza una función creada para hacer dicha reubicación, la función se llama `Chk_Feas`. Luego de esto, se vuelve a hacer la gráfica de la función y las partículas ya actualizadas en cada repetición y con ayuda del comando `pause` de Matlab, se permite la visualización de una animación en la cual las partículas se van moviendo y acercando a la solución.

Dentro del ciclo que recorre el número de partículas se hace la actualización local de las mismas utilizando un condicional y dentro del ciclo del número de repeticiones se hace la actualización global. Finalmente cuando se termine el número de repeticiones asignadas el algoritmo imprime en pantalla el valor del mínimo global encontrado.

Listing 4: Actualización de posición y velocidad

```

1  for k=2:20; %K representa el numero de iteraciones
2
3  %for i=1:ns
4
5      for j=1:2
6          % Ecuaciones de actualización de velocidad y posición
7          V(i,j)=w*V(i,j) + c1*rand*(Pi(i,j)- X(i,j))+ c2*rand
            *(PG(1,j)- X(i,j));
8          X(i,j)=X(i,j)+ V(i,j);
9      end
10
11     % Grafica del cambio de la posición de las partículas
12     hold off %Borrar la grafica anterior
13     surf(x1,y1,z1), shading interp
14     title ('Grafica del la funcion')
15     xlabel ('Eje X')
16     ylabel ('Eje Y')
17     hold on
18
19     X(i,:) = Chk_Feas(X(i,:),bnd,1); % Verifica si las
            posiciones están dentro de los límites
20
21     FE(i)=f(X(i,:)); %FE contiene a la función
            evaluada en la posición actualizada
22
23     for u=1:ns
24
25         % Grafica de las partículas en la función objetivo
26         plot3(X(u,1),X(u,2),FE(u), '--rs', 'LineWidth', 2, ...
27             'MarkerEdgeColor', 'k'
28             , ...
                'MarkerFaceColor', 'r'
                , ...

```

```

29                                     'MarkerSize',10);
30         hold on
31         end
32
33         pause(0.001)
34
35         if FE(i)<fe(i) %Si la posicion actualizada es menor a la
36             anterior
37             Pi(i,:)=X(i,:);%Se establece como la mejor
38                 posicion actual
39             fe(i)=FE(i);
40         end
41     end
42 [FPG, ipg]=min(FE); %Se obtiene el minimo de los valores de la
43     funcion evaluada y su posicion
44     if FPG<fpg
45         PG=X(ipg,:); %Se obtiene el minimo global
46         fpg=FPG;
47     end
48
49 PG %Es el minimo global encontrado "posicion en 2D"

```

2.1.1. Función Seeder

La función Seeder es la encargada de realizar la distribución de las partículas entre el espacio delimitado por la función a implementar, dicha función presenta dos alternativas: la primera solo realiza una distribución uniformemente aleatoria entre los límites demarcados y la segunda se presenta en caso de que no se cuente con limitaciones fijas sino que se cuente con fronteras conformadas por otro tipo de funciones.

El uso que se le dio fue la de tipo estándar, donde simplemente se ingresaban los límites de la función que se quería implementar y la función Seeder se encarga de distribuir el enjambre en el correspondiente espacio solución ya delimitado.

2.1.2. Función Chk_Feas

La función Chk_Feas que se encuentra en la sección 5.3.3 de los apéndices de la Tesis, es la encargada de tomar la decisión cuando una de las partículas luego de actualizarse se sale del espacio solución. Dicha función proporciona 2 posibles alternativas para la restitución de las partículas. La primer alternativa se encarga de reubicar la partícula de manera aleatoria pero dentro del espacio solución, la segunda alternativa se encarga de poner la partícula en el límite de la frontera de modo que siga su recorrido dentro del espacio delimitado.

3. Análisis de Resultados

La implementación del algoritmo se hizo sin favorecer una determinada topología en la ubicación inicial de las partículas, las cuales se distribuyeron al inicio de manera uniformemente aleatoria con el fin de evitar sesgar el punto de partida y garantizando el cubrimiento uniforme del espacio solución. En este orden de ideas es más fácil y rápido para el algoritmo encontrar el mínimo global si inicialmente se cuenta con suerte y una de las partículas toma un valor cercano al punto de interés que se esta buscando. A partir de esto se ve la importancia de ser imparciales en la ubicación inicial de las partículas y la manera en como se distribuyen las mismas en el espacio solución, sin dejar a un lado la influencia que toma la cantidad de dichas partículas en el enjambre.

Un ejemplo que permite ver lo expuesto anteriormente se puede ver en las figuras 2 y 3, donde un mismo experimento bajo los mismos parámetros representa dos casos distintos en la convergencia del enjambre hasta la iteración final, debido a la ubicación inicial de las partículas. En la figura 2 se puede observar que la convergencia de las partículas es más compacta debido a que la posición inicial de una de estas fue cercana al mínimo global, favoreciendo la atracción del enjambre hacia la mejor posición, caso contrario se evidencia en el experimento cuyos resultados se muestra en la figura 3 en el que todas las partículas en su posición inicial quedaron lejos de la mejor solución y la convergencia final de las mismas no fue tan compacta.

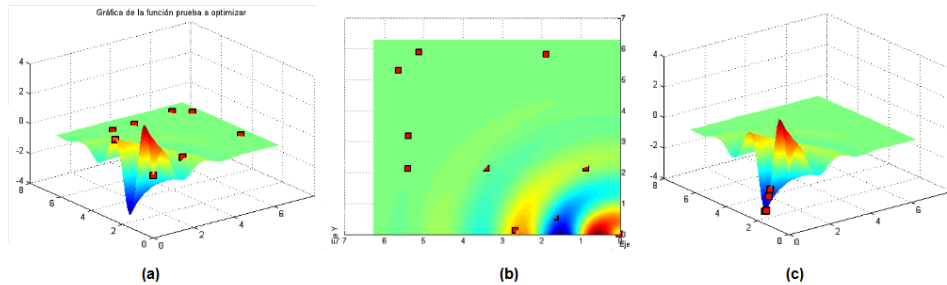


Figura 2: Posiciones iniciales (a y b) y finales (c), del primer experimento. (fuente: Autores)

3.0.3. Influencia del número de partículas en la convergencia

Se hicieron pruebas en las que se incrementó paulatinamente el número de partículas con el fin de establecer la influencia de este parámetro en la convergencia del algoritmo. A partir de esto, se observó que si el número de partículas en el enjambre es muy bajo (10 partículas) la información del espacio solución y reconocimiento del mismo carece de diversidad; caso contrario a enjambres más grandes (15 a 20 partículas) donde la mayor cantidad de información permite examinar un mayor número de zonas del espacio solución y a partir de ello dirigir el enjambre hacia zonas óptimas a medida que el enjambre va haciendo un reconocimiento más detallado

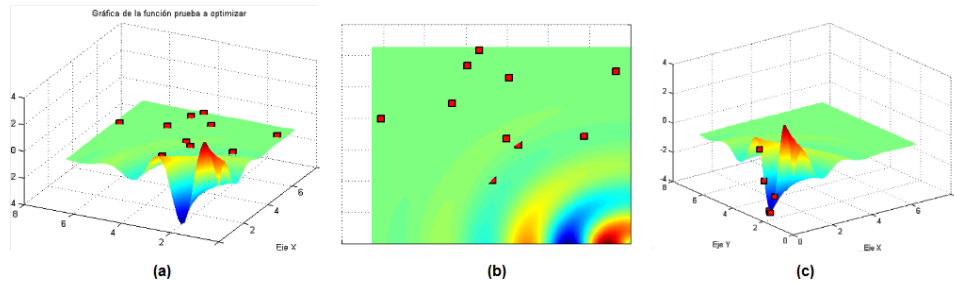


Figura 3: Posiciones iniciales (a y b) y finales (c), del segundo experimento. (fuente: Autores)

del espacio. Esto se puede observar en las figuras 4, 5 y 6 donde se puede observar la convergencia final para una cantidad de 10, 15 y 20 partículas respectivamente.

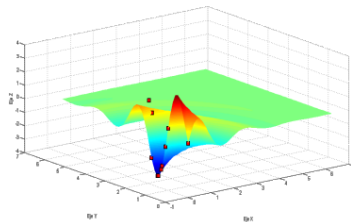


Figura 4: Posiciones finales, de un experimento para 10 partículas. (fuente: Autores)

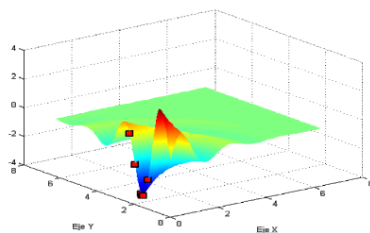


Figura 5: Posiciones finales, de un experimento para 15 partículas. (fuente: Autores)

La cantidad de partículas en el enjambre también influye en la calidad o precisión de los resultados encontrados, puesto que un número de partículas bajo no ofrece la misma información que un número alto, donde dicha información es necesaria para la depuración del área que la técnica metaheurística permite localizar. En la tabla 7 se observa que la posición donde se ubica el mínimo global oscila entre valores muy cercanos, lo cual se asocia al elevado número de partículas lanzadas que en este caso fue de 20 partículas.

La reducción a la mitad del número de partículas lanzadas junto con un número de iteraciones alto y constante, demuestra obtener resultados similares a los

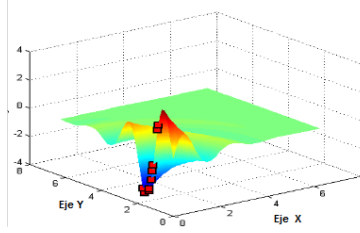


Figura 6: Posiciones finales, de un experimento para 20 partículas. (fuente: Autores)

	ni= 30	np=20
W=0,7	x	y
	0	1,4979
	0	1,5192
	0	1,5205
	0	1,5142
	0	1,5174
	0	1,52
	0	1,5212
	0	1,5193
	0	1,5187
	0	1,5183
	0	1,5167

Figura 7: Posición de la mejor partícula para ni=30, np=20, w=0.7

presentados en la tabla anterior. La posición del mínimo global, como se evidencia en la tabla 8 donde la posición del mínimo varió con respecto al experimento previo en 10 milésimas, pasando de una posición en el eje vertical de 1,5167 a 1,5177.

	ni= 30	np=10
W=0,7	x	y
	0	1,5216
	0	1,4877
	0	1,5215
	0	1,5222
	0	1,5339
	0	1,518
	0	1,5209
	0	1,5192
	0	1,5197
	0	1,5124
	0	1,5177

Figura 8: Posición de la mejor partícula para ni=30, np=10, w=0.7

El lanzamiento de pocas partículas, de acuerdo con los datos obtenidos en la tabla 9, evidencia un distanciamiento notorio del mínimo frente a los experimentos anteriores, lo cual indica que este algoritmo requiere de una cantidad lo suficientemente amplia de partículas para obtener un mejor desempeño. Otro ajuste que se puede implementar es la reducción del área de exploración, en caso de utilizar una cantidad baja de partículas, según lo consultado en [1].

	ni= 30	np=5
W=0,7	x	y
	0	1,4476
	0	5,6585
	0	1,3154
	0	1,7644
	0	1,7327
	0	1,5051
	0	1,5388
	0	5,6627
	0	1,4559
	0	1,4851
	0	2,3566

Figura 9: Posición de la mejor partícula para ni=30, np=5, w=0.7

Otro factor de interés corresponde al número de iteraciones realizadas por el algoritmo en una ejecución. El número de iteraciones normalmente no deben ser tan bajo ya que podría afectar la credibilidad del resultado y la exactitud de los mismos. La razón por la cual el número de iteraciones requeridas en la búsqueda de un mínimo global no debe ser pequeño y de hecho fijo, se debe a que es posible que se interrumpa el avance de las partículas en la búsqueda de dicho punto; de ahí surge la necesidad de buscar la manera de que el número de iteraciones no sea un factor que impida la evolución de la búsqueda del mínimo global por parte del enjambre.

Dentro de las observaciones realizadas a los resultados de las diferentes pruebas aplicadas, resulta interesante comprender la relación existente el número de partículas lanzada y el rendimiento del algoritmo para un factor de peso de inercia alto. En las pruebas anteriormente mencionadas, el peso de inercia se mantuvo constante en un factor de 0,7 de acuerdo con la literatura consultada. En otras pruebas realizadas, se definió un peso de inercia constante de 5, restringiendo el número de partículas lanzadas en la búsqueda del mínimo en cada prueba. En la primera prueba, la posición alcanzada se alejó del mínimo global esperado; sin embargo, las partículas no oscilaron de acuerdo con los datos registrados en la tabla 10.

	ni= 30	np=30
W=5	x	y
	0	6,2832
	0	6,2832
	0	6,2832
	0	6,2832
	0	6,2832
	0	6,2832
	0	6,2832
	0	6,2832
	0	6,2832
	0	6,2832
	0	6,2832

Figura 10: Posición de la mejor partícula para ni=30, np=30, w=5

La reducción sensible del número de partículas (se lanzaron 10 partículas menos) no cambió la percepción de distanciamiento de la posición del mínimo por parte de

las partículas, este comportamiento fue registrado por la tabla 11.

	ni= 30	np=20
W=5	x	y
	0	6,2832
	0	6,2832
	0	6,2832
	0	6,2832
	0	6,2832
	0	6,2832
	0	6,2832
	0	6,2832
	0	6,2832
	0	6,2832
	0	6,2832

Figura 11: Posición de la mejor partícula para ni=30, np=20, w=5

Para la siguiente prueba, se utilizaron solamente 10 partículas y los resultados registrados en la tabla 12 evidencian una dispersión de las partículas, ya que nunca localizaron una posición única para el mínimo global.

	ni= 30	np=10
W=5	x	y
	0	1,9185
	0	6,3832
	0	6,2832
	0	6,2832
	6,2832	0,0000
	0,5916	3,1813
	2,8088	1,5500
	1,8662	3,1465
	0	6,2832
	0	6,2832
	1,15498	4,1312

Figura 12: Posición de la mejor partícula para ni=30, np=10, w=5

Nota: El último dato corresponde a la posición promedio.

4. Análisis y Discusión

Entre los temas tratados de mayor interés en esta sesión del seminario, se destaca la necesidad por observar cada partícula perteneciente al enjambre en todo su proceso de evolución teniendo en cuenta que el algoritmo no siempre va a llegar a la solución esperada, sino que va a brindar indicios que sirven para llegar a una posible solución. Bajo este principio se planteó que el código debería permitir el registro de toda la evolución para cada una de las partículas.

De igual manera se aclaró acerca de la imparcialidad que se debía tener en la posición inicial de las partículas evitando así el sesgo en la solución del problema,

por tal razón se decidió en conjunto, que el algoritmo debería seguir funcionando sin considerar ninguna topología puesto que el uso de ellas representaría la necesidad de caracterizar las ventajas y desventajas de las mismas con respecto al tipo de función que se quería implementar, y que para el objeto del seminario, dicha función es desconocida.

Además de las consideraciones anteriores, se estableció la necesidad de caracterizar y entender el comportamiento del enjambre conforme este iba actuando en la ejecución sobre la función de prueba implementada, de modo que se concluyera y se comprendiera la manera en cómo actúa el enjambre respecto a las distintas pruebas que una función puede presentar para el algoritmo; teniendo en cuenta a su vez la influencia a las que tiene lugar el valor de las distintas variables en las ecuaciones del algoritmo PSO.

Adicional a esto se discutió acerca de la manera correcta de manipular los resultados obtenidos en los distintos experimentos realizados, donde se cuestionaba si el promedio de estos era una medida adecuada para caracterizar y comparar los resultados obtenidos en las distintas pruebas. Finalmente se concluyó que es necesario recurrir a distintas medidas que resuman los resultados de varios experimentos que permitan observar aspectos que pueden pasar desapercibidos, cuando la interpretación de resultados se hace en función de una única medida como el cálculo de la posición promedio del mínimo global encontrado, tal y como se muestra en las tablas 10, 11 y 12 donde a simple vista el valor promedio encontrado en la tabla 12 se aleja notoriamente del valor esperado, cuando en las tablas 10 y 11, las partículas no cambiaron bruscamente la ubicación de un posible mínimo local.

Cabe aclarar que la implementación del código del algoritmo PSO en esta sesión, no contempla el uso de radio de avance distinto de 1 para cada partícula en su correspondiente actualización de posición y velocidad. Esto implica que se utiliza una expresión matemática cuya velocidad está multiplicada por 1, como se evidencia en la subsección de expresiones matemáticas para PSO, en la sesión 2. A modo de repaso, se observa la expresión utilizada para el cambio de posición de las partículas durante el desarrollo de las primeras pruebas.

$$X_{i,j} = X_{i,j} + V(i,j) * dt \quad (1)$$

Donde:

$X_{i,j}$ = Posición en dos dimensiones de cada partícula.
 $V(i,j)$ = Velocidad en dos dimensiones de cada partícula.
 dt = Radio de avance de cada partícula (por defecto 1).

5. Conclusiones

- Un bajo número de partículas limita enormemente la información acerca del espacio solución que se quiere depurar, por tal razón es recomendable usar un enjambre numeroso para garantizar parte del buen funcionamiento del algoritmo PSO.
- El número de iteraciones no debe ser una limitante que corte la búsqueda del algoritmo, sin embargo, el registro de estas puede llegar a ser usado como un indicativo de la rapidez con la que el algoritmo converge.
- Las medidas de dispersión y de resumen de resultados como el promedio, son indicativos aceptables que pueden ayudar a inferir en el análisis de los resultados ya tabulados, sin embargo, es de tener en cuenta que cada evento suministra una información en particular que podría darle un valor agregado a la interpretación de los resultados.
- El peso de la inercia constituye un parámetro de gran influencia en el movimiento que tiene el enjambre, tal y como se pudo evidenciar en los resultados contenidos en la carpeta Tablas - sesión 3. En este punto es difícil caracterizar y relacionar dicho parámetro con el comportamiento del enjambre, a causa de que tan solo se a implementado en una función prueba.

Referencias

- [1] José Manuel García Nieto. Algoritmos basados en cúmulos de partículas para la resolución de problemas complejos. neo.lcc.uma.es/staff/jmgn/doc/Memoria_PFC_JMGN.pdf.
- [2] Wikipedia. Optimización por enjambre de partículas. http://es.wikipedia.org/wiki/Optimizacion_por_enjambre_de_particulas.

Memoria4

Uso de técnicas de Optimización Global para resolver problemas de Inversión Sísmica

Edgar Téllez Cortés, Johao Coronel Silva, Juan Arias Pinto

Universidad Industrial de Santander
Escuela de Ingenierías Eléctrica, Electrónica y Telecomunicaciones

Junio 4 de 2014

Sesión 4: Implementación de PSO sobre funciones de prueba, segunda parte

1. Introducción

Una vez se ha comprobado el mecanismo de funcionamiento del algoritmo basado en Enjambre de Partículas sobre una función multidimensional sencilla, el siguiente paso es adaptar el algoritmo a cualquier función independiente del nivel de dificultad de la misma, para ello se eligió un conjunto de funciones de diversos niveles de dificultad. La realización de pruebas, se encamina al reconocimiento de patrones en los parámetros del algoritmo, en este documento, se describen las modificaciones realizadas al código presentado en la sección 5.3.1 del capítulo apéndices de la Tesis, las consideraciones que se tomaron a partir del espacio de discusión de la sesión 3 y el análisis de resultados obtenido a partir de la implementación del algoritmo sobre cada una de las funciones de prueba.

2. Implementación del Algoritmo de Enjambre de Partículas en otras Funciones de Prueba

En la sesión 5.4.1 del capítulo apéndices de la Tesis de Grado, se encuentra el código en Matlab que permite graficar cualquiera de estas funciones y observar su respectiva ecuación matemática, tan sólo hay que descomentar las líneas de código relacionadas con la función. El conjunto de funciones prueba seleccionadas para

analizar el comportamiento del algoritmo aplicando nuevas pruebas consta de 12 funciones distintas, adicionales a la función básica, en donde cada una de ellas ofrece un nivel de complejidad diferente y de igual manera distintos retos en cuanto al reconocimiento del espacio solución por parte del enjambre de partículas. Las funciones de prueba son conocidas como: Función Ackley's, Beale's, Goldstein-Price, Easom, Eggholder, Holder table, Esfera, Steps, Rastriging, Schewefels, Rosenbrock, Stochastic y la función de prueba básica ya explicada en la sesión anterior.

2.1. Función de Ackley

La función Ackleys se puede considerar como una función con un nivel de complejidad bajo puesto que el acceso a su mínimo global que es único, se puede realizar sin ninguna dificultad desde cualquier punto dentro del espacio solución dado que el mínimo se encuentra en el centro del espacio solución, tal como se observa en la figura 1. A pesar de esto, la función ofrece curvas descendientes hacia el centro, lo cual podría ser un reto para el enjambre dado que la función da la sensación de que se pueden encontrar distintos puntos mínimos con respecto a los puntos lejanos del centro.

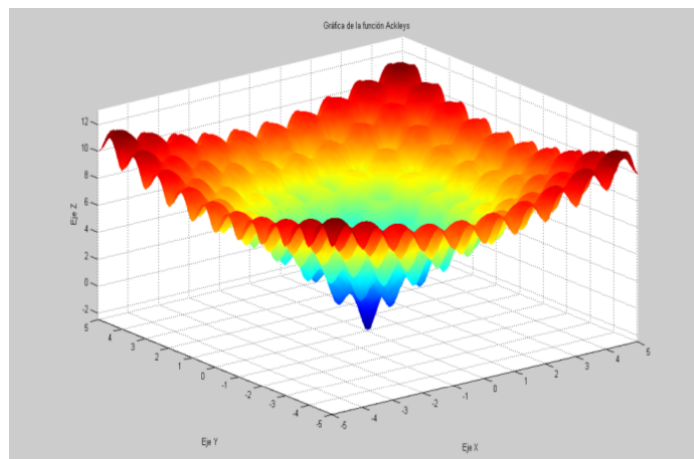


Figura 1: Función de Ackley. Fuente: Autores

2.2. Función Beales

La función Beales se puede considerar como una función con nivel de complejidad intermedio dado que presenta una gran planicie en el centro de la función, lo cual ofrece un gran reto para el enjambre debido a que los puntos por donde transitan las partículas no presentan una mejora notoria según se observa en la figura 2. De ahí que la capacidad de exploración global deberá permitir al enjambre encontrar una mínima diferencia en amplitud con respecto a los distintos puntos alrededor del mínimo global de interés

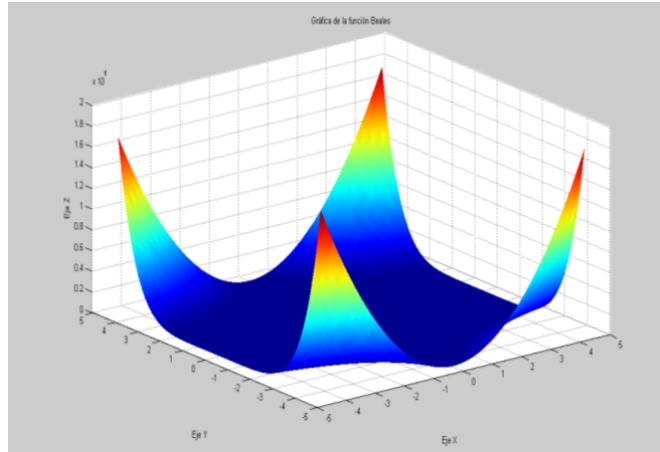


Figura 2: Función Beales. Fuente: Autores

2.3. Función Goldstein Price

La función Goldstein-Price se puede considerar como una función con nivel de complejidad intermedio puesto que es muy similar en cuanto a las características de interés a la función Beales, la única diferencia con respecto a la función anterior radica en que solo uno de los puntos externos presenta un punto máximo mientras que la función anterior presenta en sus cuatro extremos puntos máximos. Figura 3.

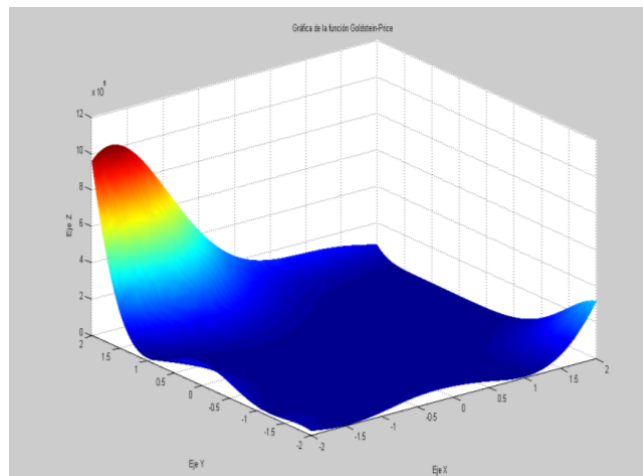


Figura 3: Función Goldstein Price. Fuente: Autores

2.4. Función Eason

La función Eason también puede ser considerada con un nivel de complejidad intermedio, dado que presenta un único mínimo global en medio de una gran planicie. El reto que ofrece dicha función se relaciona con el hecho de poder encontrar verdaderamente el mínimo global dado que la zona que presenta una menor amplitud

con respecto a los demás puntos, desciende rápidamente como lo muestra la figura 4. En caso de no poseer una gran cantidad de información y una alta resolución en el avance por parte de las partículas, es posible tomar un punto cercano al punto de interés como mínimo global o en el caso más desfavorable, caer en un óptimo local.

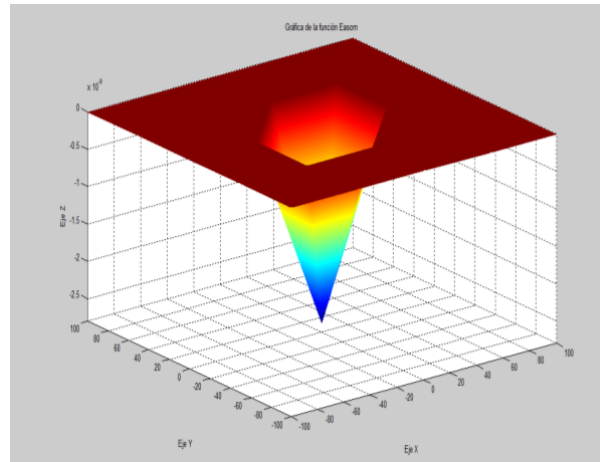


Figura 4: Función Easom. Fuente: Autores

2.5. Función Egg Holder

La función Eggholder se puede considerar como una función con un nivel de complejidad alto, dado que presenta varios mínimos globales y un solo mínimo global que a no ser de una gran reconocimiento del espacio solución, se puede confundir con uno de los muchos mínimos locales. Este tipo de función verdaderamente presenta un gran reto para el enjambre dado que presenta muchas zonas tentativas, como se observa en la figura 5 donde se evidencia un gran progreso en el paso de un máximo a un mínimo y adicional a esto un descenso diagonal hacia puntos de menor amplitud.

2.6. Función Holder Table

La función Holder Table se puede considerar como una función con un nivel de complejidad intermedio donde se puede encontrar un mezcla de los retos que ofrecen las funciones Easom y Eggholder en cuanto a la precisión en la búsqueda del mínimo global en un área de descenso rápido y la elección del mínimo global y no local, teniendo en cuenta el reconocimiento de todo el espacio solución. Además de esto la función presenta cuatro mínimos globales ubicados en los extremos de la función, como se puede detallar en la gráfica de la figura 6, sin embargo la optima búsqueda de dichos puntos depende del buen reconocimiento de la función en las zonas en donde tentativamente se pueden encontrar muchos mínimos.

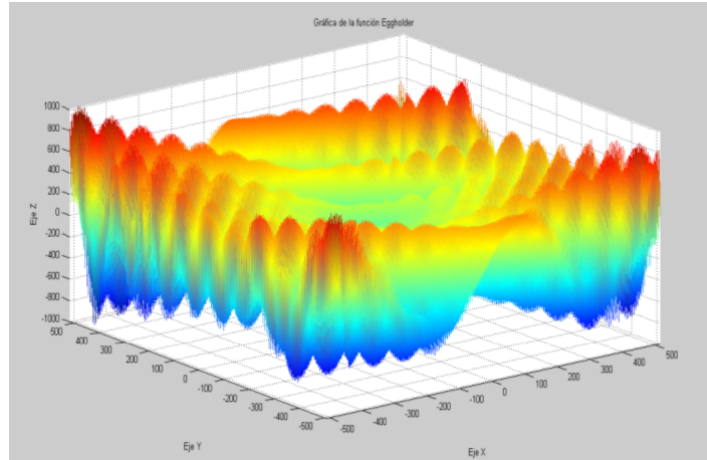


Figura 5: Función Egg Holder. Fuente: Autores

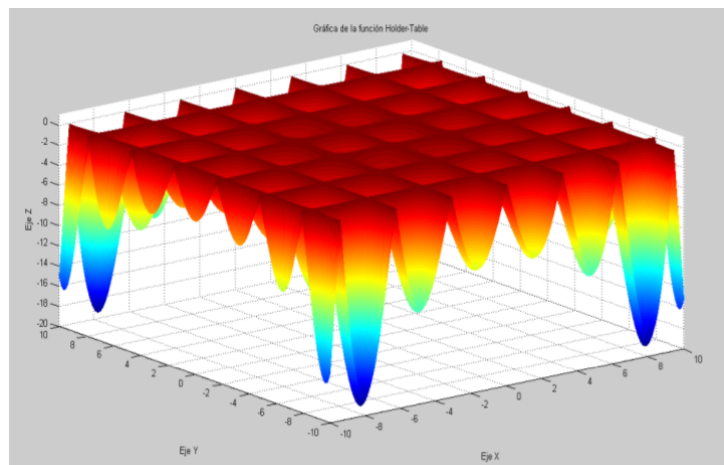


Figura 6: Función Holder Table. Fuente: Autores

2.7. Función Esfera

La función Esfera presenta un nivel de complejidad bajo, como lo evidencia su gráfica, la cual se muestra en la figura 7 dado que solo presenta un mínimo global sin la presencia de otros mínimos. El acceso al punto de interés no presenta ningún tipo de dificultad, sin embargo presenta un reto para el enjambre relacionado con la precisión del punto de interés dado que se encuentra en una zona donde cualquier punto alrededor puede ser tomado como mínimo global sin serlo estrictamente. Este tipo de función más que todo es usada para la verificación del correcto funcionamiento del algoritmo.

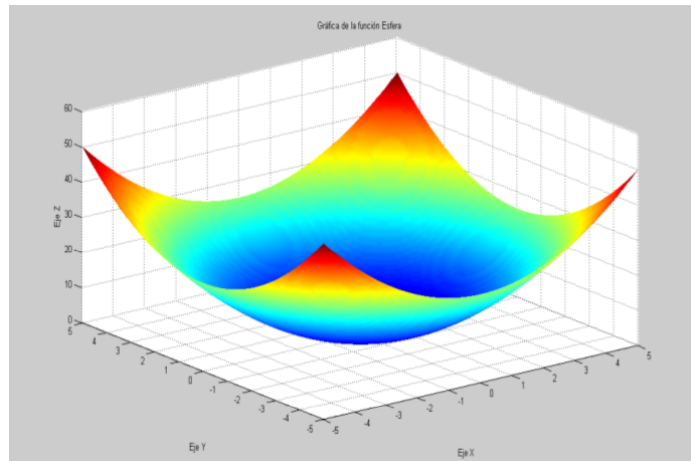


Figura 7: Función Esfera. Fuente: Autores

2.8. Función Step

La función Step es una representación discretizada de la función Esfera, por tal razón se considera con un nivel de complejidad bajo. El reto que quizás puede ofrecer se atribuye al hecho de que el enjambre debe reconocer muy bien el espacio solución y tener la capacidad de progresivamente ir transitando hacia las zonas con menor amplitud hasta llegar al punto de interés, la gráfica de esta función se observa en la figura 8.

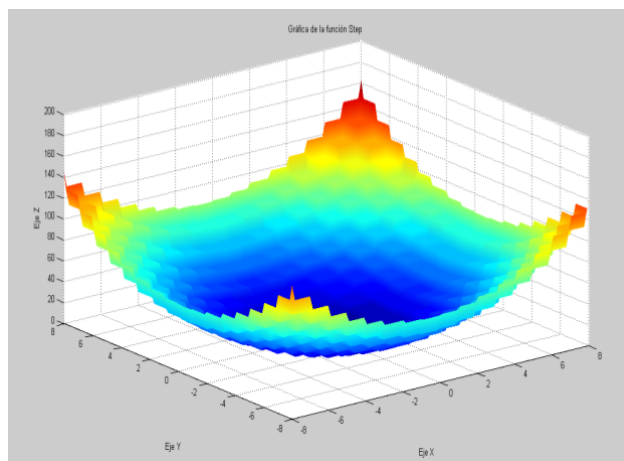


Figura 8: Función Step. Fuente: Autores

2.9. Función Rastrigin

La función Rastrigin se puede considerar con un nivel de complejidad alto como se puede observar en la figura 9, debido a que presenta una gran cantidad de mínimos locales y un único mínimo global que a no ser de un gran reconocimiento de todo

el espacio solución puede confundirse con los mínimos locales que se encuentran alrededor. El reto que presenta dicha función es similar al que ofrece la función Eggholder donde se evidencia constantemente un cambio entre puntos máximos y mínimos.

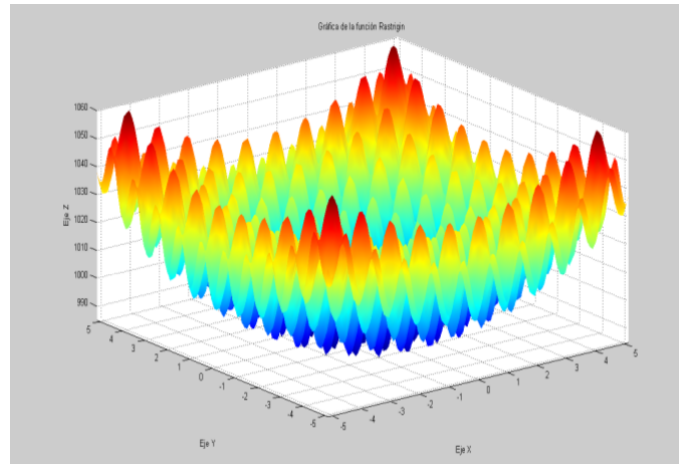


Figura 9: Función Rastrigin. Fuente: Autores

2.10. Función Schwefel

La función Schwefel también es considerada como una función con un nivel de complejidad alto, ofrece el mismo reto de la función Rastrigin con la única diferencia que el cambio entre los puntos máximos y mínimos es más lento y presentan una mayor área. Las características nombradas anteriormente también se presentan en esta función razón por la cual la hacen interesante al presentar un único mínimo global con una pequeña diferencia de amplitud con respecto a los mínimos locales ubicados en los extremos de la función. Este comportamiento se observa en la figura 10.

2.11. Función Valle de Rosenbrock

La función Rosenbrock se puede considerar con un nivel de complejidad intermedio. A diferencia de las funciones que presentan una gran planicie y en medio de ella el mínimo global, este tipo de función presenta un área con forma de parábola para los puntos más bajos en amplitud, como lo evidencia la gráfica de la figura 11. El reto que ofrece es la precisión que se debe tener para llegar al punto exacto de interés y no converger prematuramente a causa de no evidenciar un cambio notorio en amplitud en los puntos de la planicie y de la zona más baja.

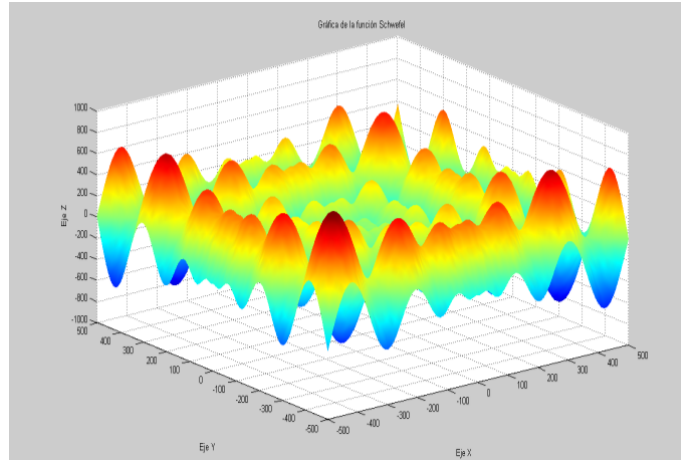


Figura 10: Función Schwefel. Fuente: Autores

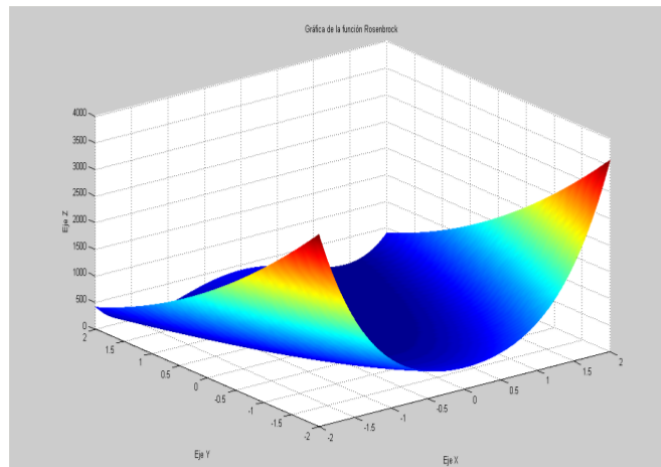


Figura 11: Función del Valle de Rosenbrock. Fuente: Autores

2.12. Función Stochastic

La función Stochastic se puede considerar con un nivel de complejidad alto, presenta un fácil acceso por cualquier punto externo hacia el mínimo global que es único y está centrado, como se muestra en la figura 12. La función presenta el reto de contener una gran variedad en las amplitudes de los puntos, evidenciando un drástico cambio entre los máximos y mínimos. El reto que representa para el algoritmo se relaciona con el hecho de encontrar verdaderamente el mínimo global y no los posibles mínimos locales situados alrededor del punto de interés o en cualquier otro punto de la función.

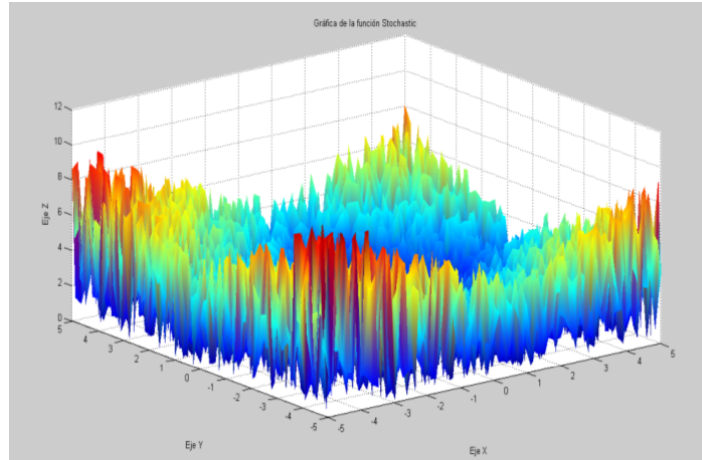


Figura 12: Función Stochastic. Fuente: Autores

3. Código Implementado sobre Matlab

La explicación necesaria para entender el algoritmo PSO se encuentra en la memoria de la sesión 3 donde se logra explicar línea por línea la interpretación del código. El contenido de esta sesión con respecto a la implementación del algoritmo se refiere a las modificaciones realizadas a partir de las discusiones hechas en la sesión anterior y las ideas obtenidas para la mejora en el rendimiento del algoritmo. Las modificaciones realizadas durante esta sesión se pueden encontrar detalladamente en la sección 5.4.2 del capítulo apéndices de la Tesis.

En la implementación pasada se programó una animación para poder ver la trayectoria de las partículas desde su posición inicial hasta que el algoritmo se detuviera, sin embargo se presentaba el inconveniente de que cada vez se imprimía en pantalla una nueva gráfica, esta se corría con respecto a la anterior; es por eso que fue necesario fijar los ejes de la función con ayuda de la herramienta axis de Matlab, esto se hace en el momento de realizar la gráfica de la función y sus respectivas partículas en la misma.

Listing 1: Nuevos ajustes

```

1 %% Grafica de la funcion objetivo
2 [x,y]= meshgrid(linspace(-5,5,100));
3 % Ecuacion de la funcion a optimizar
4 z1= x.^2 + y.^2;
5 % Grafica la funcion en tres dimensiones con colores interpolados
6 surf(x,y,z1), shading interp
7 axis([-5 5 -5 5])
8 title ('Grafica de la funcion Esfera')
9 xlabel ('Eje X')
10 ylabel ('Eje Y')
11 zlabel ('Eje Z')

```

Teniendo en cuenta que se quería ver la trayectoria de cada partícula fue necesario modificar el código asignando un color diferente a cada una. La estrategia empleada para que cada partícula tuviera un color distinto fue expandir el ciclo de repetición con el que se grafican las partículas en la función objetivo, de manera que el código quedo un poco extenso debido a que entre más partículas fueran lanzadas en la búsqueda del mínimo global, más líneas de código saldrían. Las líneas correspondientes a dicha modificación se encuentran entre las líneas [50,200] y [232,382] del código mostrado en la sección 5.4.1 del capítulo apéndices. También se agregó una nueva gráfica de la vista superior de la función con sus respectivas partículas ubicadas en su región de exploración de modo que la visualización se hiciera más fácil y permitiera interpretar mejor el comportamiento que tenía el enjambre con respecto al reto que la función presentaba.

Listing 2: Contornos de la función

```
1 figure
2 contourf(x,y,z1) % Muestra el contorno de la funcion
3 colorbar % Muestra la columna de colores para saber la altura de la
   funcion
4 title ('Grafica del contorno de la funcion (Vista superior)')
5 xlabel ('Eje X')
6 ylabel ('Eje Y')
7 hold on
```

En vista de que el código anterior se detenía por el número de iteraciones definidas por el usuario, se implemento un criterio de parada que se encargara de detener el algoritmo si se cumplían algunos requisitos asignados por el usuario. Dicho criterio se implementó a partir de la función Chk_Out con autoría del grupo CMOS de la Universidad Industrial de Santander y modificada por nosotros como función Chk_Out_camilo. Dicha función cumple la tarea de detener el algoritmo en caso de ser necesario, las alternativas que presenta son: por número máximo de iteraciones permitidas, por poca variabilidad entre el mínimo global encontrado en la iteración actual en comparación con el de la iteración anterior o por la mejora del mínimo global encontrado después de un número de iteraciones asignado.

Dentro de los parámetros que se deben asignar a la función del criterio de parada, está el máximo número de iteraciones permitidas por el usuario, un épsilon que se comparará con la diferencia entre el mínimo global actual y el de la iteración anterior para determinar si el cambio entre una iteración y otra es notorio, y un número máximo de ocasiones seguidas en donde el mínimo global no cambie. La función Chk_Out_camilo se encuentra en el la sección 5.4.5 del capítulo apéndices.

4. Análisis de Resultados

En el caso de la función básica empleada en la sesión pasada, se realizó la implementación del algoritmo con las modificaciones hechas hasta ahora, observando su

comportamiento en las figuras de la 1 a la 6 de la sección 5.4.6 del capítulo apéndices, donde se muestran las tomas de vista superior de las partículas en las primeras iteraciones. En estas figuras se puede observar la trayectoria de cada una de ellas evidenciando en general los entornos sociales y geográficos iteración tras iteración, como por ejemplo ocurre con la partícula resaltada por el color verde y contorno negro, en la cual la misma parte de un lugar lejano al punto de interés pero se va acercando poco a poco de acuerdo a la atracción que le brindan las demás partículas; este avance es progresivo sin dejar a un lado el reconocimiento del espacio solución que esta partícula también hace a medida que cambia su posición.

Por otro lado en las imágenes de la 7 a la 28 de la sección 5.4.6 del capítulo apéndices, se puede observar la convergencia de todo el enjambre al mínimo global. Los parámetros utilizados en esta prueba fueron $W = 0,7$, $C1 = 1,9$, $C2 = 1,8$, $np = 30$, $psilon = 0,001$, donde w es la masa de inercia, $c1$ y $c2$ son las constantes de exploración local y global respectivamente, np es el número de partículas y $psilon$ el valor que se debe asignar para el criterio de parada del algoritmo.

Para la implementación del algoritmo en las demás funciones de prueba, se realizó la variación de parámetros como la masa de inercia para tres valores distintos. Estos fueron 0,1, 0,7 y 1,0, dicha variación se hizo con el fin de analizar el impacto de la inercia en los distintos retos que cada función ofrece. Los resultados obtenidos para las distintas funciones se consignaron en las tablas mostradas en la carpeta Tablas - sesión 4, donde para cada valor de W se hicieron 10 lanzamientos del algoritmo recogiendo los resultados del mínimo global encontrado.

Las funciones que presentaron los mejores resultados con un W de 0,7 fueron las funciones Básica, Beales y Stochastic, por otro lado las funciones que presentaron los mejores resultados para un W de 0,1 fueron las funciones Easom y Esfera; al final con un W de 1 la función Schwefel fue la que presentó resultados favorables bajo este parámetro. Cabe resaltar que los resultados obtenidos bajo los distintos valores del parámetro W se parecían mucho unos a otros, la manera en la que se decidió tomar el mejor de ellos fue teniendo en cuenta la precisión del mínimo encontrado y la cantidad de aciertos en los lanzamientos. Por otra parte se logra evidenciar la relación que tiene el valor de W y el aspecto de la función; por ejemplo, un valor de $W = 0,7$ fue necesario para los tipos de funciones donde se debía tener una capacidad de exploración local que permitiera al enjambre distinguir el mínimos global a partir de las zonas donde se encontraban mínimos locales, por tal razón un peso de inercia no muy alto, favorece un mayor dinamismo en el paso de cada partículas permitiendo una mejor exploración global. Para el caso donde $W = 0,1$, las funciones favorecidas fueron aquellas que requerían una mayor precisión en el área donde se encontraba el mínimo global, tal fue el caso de las funciones Easom y Esfera, donde un paso pequeño en la actualización de las partículas permitía un reconocimiento más depurado de la zona de interés. La función Schwefel se vio favorecida para un valor de $W = 1$ debido a que una mayor inercia en el avance de las partículas, permitía un mayor dinamismo en su posición y consigo un mejor reconocimiento del área en total. Cabe recordar que el mínimo de esta función se encontraba en un

extremo razón por la cual se cree que la alta inercia permitió llegar a tal extremo de la función y no converger en otro extremo situado por un mínimo local.

Los códigos que permiten ver la evolución de las partículas para cada uno de los valores de las pruebas mencionadas con anterioridad implementadas en cada una de las funciones, se encuentra en la sección 5.4.2 del capítulo apéndices.

5. Análisis y Discusión

El espacio de discusión llevado a cabo durante la sesión No 4, estuvo conformado por una serie de aportes con respecto al radio de avance que puede ser considerado en la actualización de cada partícula. La manera en como se puede utilizar mejor el criterio de restitución de partículas y las pruebas que se necesitan realizar para observar de mejor manera la influencia de los parámetros W , $C1$, $C2$, np y dt a partir de los ajustes y modificaciones que se trataron durante la sesión.

Con respecto al delta de tiempo (dt), cabe aclarar que es un parámetro opcional puesto que en las ecuaciones de actualización de posición generalmente se toma como $dt = 1$. Sin embargo, el uso de este parámetro permite ajustar de mejor manera el avance que puede llegar a tener la partícula en los casos donde es necesario dependiendo el tipo de función sobre la cual se implementa el algoritmo. La ubicación donde tiene lugar este nuevo parámetro se presenta en la ecuación de actualización de posición.

$$x_{(i,j)} = x_{(i,j)} + v_{(i,j)} * dt \quad (1)$$

donde:

$X(i,j)$ = Posición en dos dimensiones de cada partícula.

$V(i,j)$ = Velocidad en dos dimensiones de cada partícula.

dt = Radio de avance de cada partícula.

El criterio de restitución de partículas fue otro tema a tratar, hasta el momento dicho criterio presenta la posibilidad de restituir la partícula de manera aleatoria o ubicándola en la frontera del espacio solución. Sin embargo, una nueva posibilidad podría ser considerada como la restitución en la última posición correcta dentro del espacio solución. Cabe aclarar que dicho criterio es utilizado en caso de que la partícula en su actualización se salga del espacio solución delimitado por la función. Teniendo en cuenta lo anterior, es necesario trabajar sobre este criterio brindándole la posibilidad al enjambre de restituir mejor la partícula. La razón por la cual surgió esta nueva idea se debe a que en el caso de la función básica, ocurría que la restitución de la partícula en la frontera favorecía la búsqueda del mínimo global que también se encontraba en ese lugar. Además de esto, las dos maneras que ofrece dicho criterio para restituir la partícula en cierto modo sesgaba la búsqueda puesto que la ubicación aleatoria o en la frontera, modificaba la trayectoria que se tenía en

las iteraciones anteriores, por tal razón es mejor no modificar dicha trayectoria sino retomar a ella en caso de que la trayectoria fallara en la última actualización, y a partir de dicho punto seguir un nuevo camino.

6. Conclusiones

- El peso de inercia es un factor que afecta de manera directa el dinamismo que tiene una partícula en actualizar su posición, teniendo en cuenta la velocidad que lleva el enjambre. En ese sentido, este parámetro influye en la manera como el enjambre puede tener la capacidad de reconocer el espacio solución en pasos largos o pasos cortos con el fin de dar una mayor información del área que se está recorriendo. Estos pasos también pueden ser conocidos como la capacidad que tiene el enjambre en una búsqueda global (pasos largos) o local (pasos cortos).
- Se pueden obtener diferentes valores de parámetros para obtener un mejor funcionamiento del enjambre, a partir de esto es necesario encontrar un conjunto de valores de los parámetros que permitan garantizar un buen funcionamiento considerando las todas características que se lograron extraer de las funciones de prueba.
- Teniendo en cuenta los aportes obtenidos en el espacio solución, otro factor que puede influir en el dinamismo que tiene la partícula al momento de hacer su actualización en velocidad y posición, es el radio de avance dt de la partícula el cual permite ajustar aún más el paso que se tiene al momento de actualizar todo el enjambre.
- Hasta el momento solo se han considerado la influencia que tienen los parámetros como cantidad de partículas, peso de inercia y conforme al espacio de discusión el radio de avance que debe tener la partícula. Cabe aclarar que las constantes $C1$ y $C2$ en las ecuaciones de actualización también permiten darle un peso a las componente de búsqueda local y global, que por supuesto también son necesarias para encontrar el mejor conjunto de valores de los parámetros que debe tener el algoritmo para un buen funcionamiento.

Memoria5

Uso de técnicas de Optimización Global para resolver problemas de Inversión Sísmica

Edgar Téllez Cortés, Johao Coronel Silva, Juan Arias Pinto

Universidad Industrial de Santander
Escuela de Ingenierías Eléctrica, Electrónica y Telecomunicaciones

Junio 10 de 2014

Sesión 5: Implementación de PSO sobre funciones de prueba, tercera parte

1. Introducción

En la sesión No. 5 se presentó nuevamente los resultados obtenidos tras la implementación del algoritmo modificado de acuerdo a las consideraciones tratadas en la sesión anterior. Cabe aclarar que las pruebas presentadas durante esta sesión se volvieron a ejecutar de acuerdo al último código modificado, es decir que parte de las pruebas realizadas en las sesiones anteriores se volvieron a correr.

En el desarrollo de la sesión No. 5 se hizo énfasis en las nuevas consideraciones que se tuvieron en cuenta como lo fue el criterio de restitución de partículas y las variaciones de las constantes $C1$ y $C2$ al igual que el radio de avance (dt). De igual manera se dedicó gran parte a la exposición del análisis de resultados para cada una de las funciones y los cuestionamientos y discusiones que surgieron a partir de ellos.

2. Implementación sobre funciones de Prueba

La implementación sobre las distintas funciones de prueba se hizo de una manera estándar para cada una de ellas, de modo que se pudieran comparar los resultados entre ellos y elegir para cada función el mejor conjunto de valores de parámetros. La metodología que se llevó a cabo en la ejecución de las pruebas fue variar un

parámetro a la vez mientras los demás permanecían constantes.

La primer ejecución del algoritmo se hizo con los valores recomendados para PSO según trabajos relacionados con el algoritmo, una vez hecha esta prueba se procedía a variar los parámetros y elegir el mejor conjunto de ellos de acuerdo a las práctica. Las primeras pruebas se hicieron variando el radio de avance dt y dejando fijos los parámetros np , W , $C1$ y $C2$ con los valores recomendados. Ya establecido el mejor valor de dt se procedió a variar la masa de inercia W en un rango de 0.1 a 0.9, acto seguido se establecía el mejor valor de W y se variaban los valores de las constantes $C1$ y $C2$ en un rango de 1 a 3. Cada uno de los resultados obtenidos en estas pruebas fueron tabulados en formatos de tablas que se pueden observar en la carpeta - Tablas - sesión 5.

3. Implementación del Algoritmo Basado en Enjambres de Partículas en Matlab

Para comprender completamente la implementación del algoritmo PSO en Matlab es necesario remitirse a la memoria número tres, en la cual se presenta una explicación detallada del mismo. Las modificaciones realizadas en el código se pueden evidenciar en las secciones 5.5.2 y 5.5.3 del capítulo apéndices, donde se muestra los correspondientes códigos para ver el comportamiento del enjambre paso a paso y la toma de datos para la construcción de tablas de resultados.

Según discusiones realizadas durante la presentación del seminario en la sesión 5, es conveniente que todos los parámetros que se empleen en el algoritmo se asignen en las primeras líneas de código tal y como se hizo en la presente versión del mismo. Por otra parte, se prescindió de la utilización de la función `Seeder` encargada de asignar las posiciones iniciales de las partículas y en su reemplazo se escribió directamente una línea de código que realiza la asignación de las mismas mediante una distribución uniforme. A su vez se omitió el uso de la función `Resti` encargada de hacer la restitución de partículas, en vez de esto se escribieron 4 líneas de código con la utilización de un condicional y operadores matemáticos de mayor y menor que, que garantizarán la restitución de la partícula en su ultima posición permitida en caso de que se saliera del espacio solución. También se modificó la función encargada de establecer los criterios de parada del algoritmo, quedando como alternativas el criterio de convergencia de partículas y la poca variación del mínimo global encontrado.

Con más detalle el nuevo criterio de parada garantiza dos cosas: que el enjambre halla convergido y que el mínimo global no cambie luego de un número determinado de iteraciones consecutivas. Para garantizar lo primero, la función se encarga de comparar la posición de las partículas en la iteración actual y la anterior garantizando que la máxima diferencia entre todas estas sea menor a un epsilon denominado dentro del código como `Epsilon 2`. Por otra parte para asegurar que el mejor resultado tenga poca variación como mínimo en 5 oportunidades consecutivas, la función

utiliza un condicional que compara la diferencia entre el mejor resultado actual con el anterior y un valor denominado Epsilon que es el que determina en que dicho mínimo global no varíe en un valor considerable. Adicional a esto, el número máximo de iteraciones permitido se programó por fuera de esta función en el inicio del ciclo de repetición de iteraciones. El código del criterio de parada se puede observar en la sección 5.5.3 del capítulo apéndices, en la función `Chk_Out_Camilo.m`.

Una de las modificaciones importantes que se hicieron para esta sesión fue la incorporación del parámetro radio de avance dt , el cual esta presente en la ecuación de actualización de posición del algoritmo. Este factor multiplica a la velocidad actualizada que se suma a la posición actual con la idea de hacer coherente la ecuación al observar que así en la misma se suman dos posiciones y no una velocidad y una posición.

4. Análisis de Resultados

A continuación se muestra el análisis de las pruebas realizadas para cada función objetivo a partir de las animaciones vistas por el movimiento de las partículas en el espacio de búsqueda y a las tablas mostradas en la carpeta - Tablas - sesión 5. Todas estas tablas registran el número de iteraciones realizadas y las tres componentes del mínimo global encontrado para cada prueba, además muestran los principales parámetros utilizados en cada prueba de cada función y el valor exacto de la ubicación del mínimo global.

En pruebas preliminares se observó que un radio de avance del orden de 10^{-2} y 10^{-3} ocasiona que las partículas se muevan muy lento produciendo un número de iteraciones muy grande y un gasto innecesario en memoria de las posiciones recorridas, por esta razón se buscó un buen valor de este parámetro debe estar en el orden de 10^{-1} .

Las pruebas realizadas consistieron en variar un parámetro a la vez dejando los demás fijos según las indicaciones de valores estándar, estos son un número de partículas de 30, constantes de actualización local y global de 1,9 y 1,8 respectivamente y el valor de epsilon 1 y 2 de 0,1 para ambos. En primer lugar se varió la masa de inercia iniciando en 0,1 hasta 0,9 en pasos de 0,2, después esto se eligió el mejor valor de W para fijarlo e iniciar las siguientes pruebas en las que se varió las constantes de actualización global y local. En este caso se dejó una de las dos constantes fija mientras la otra variaba de 1 a 3 en pasos de 0,3.

Por pruebas preliminares se llegó a la conclusión de que el valor de los epsilon correspondientes a los criterios de parada deben ser del mismo orden que el delta de tiempo que establece el radio de avance de las partículas, puesto que de no ser así, el criterio de parada podría estar deteniendo el algoritmo en tan solo unos cuantos pasos. También es de tener en cuenta que el radio de avance no debe ser asumido como uno ($dt = 1$) ya que esto ocasiona que el cambio en la posición de las partículas

sea tan grande que las partículas se salen del espacio de búsqueda y sea necesario restituir las partículas a su posición anterior aceptada. Al ser el $dt = 1$ ocurre que la salida de partículas es permanente ocasionando que las mismas no tengan movilidad en el algoritmo y queden estancadas. Para llegar a este tipo de conclusión fue necesario observar con detalle qué estaba sucediendo en cada actualización de partículas, la simple observación del enjambre por medio de la animación del código en Matlab permite dejar ver este fenómeno en algunas de las funciones de prueba.

En las pruebas consignadas en las tablas solo se varió la masa de inercia y las constantes de actualización local y global ya que los demás parámetros como el número de iteraciones, el radio de avance y los épsilon de los criterios de parada se mantuvieron constantes al hacer pruebas preliminares, esto con la idea de reducir el tiempo y el número de pruebas por realizar. El delta de tiempo de 0,5 arrojó buenos resultados en todas las funciones en pruebas preliminares, por esta razón se dejó como un valor fijo en la realización de las tablas, teniendo en cuenta lo anterior se decidió fijar de igual manera los epsilon de los criterios de parada con el valor de 0.1. Los criterios de restitución y de parada utilizados en las pruebas finales son el máximo número de iteraciones y el detenimiento por poca variabilidad de las mejores partículas junto con la restitución de partículas a su última posición aceptada.

4.1. Función Básica

Después de realizar las pruebas pertinentes con esta función se determinó que el radio de avance de las partículas no debe ser tan pequeño, por ejemplo un valor de 0,001 ya que esto puede ocasionar que las partículas tarden mucho en recorrer el espacio de búsqueda e incluso que nunca lleguen todas al mínimo global. De igual manera ocurre que las partículas se mueven tan lento que no encuentran mínimos globales ni locales y en una planicie cualquiera convergen de manera prematura. Un avance adecuado para esta función es de 0,5 con el cual todas las partículas convergen al mínimo global en un reducido número de iteraciones (45 iteraciones) dejando actuar el criterio de parada cumpliéndose los requisitos explicados con anterioridad.

Para los casos en que W es menor a 0,5 se puede observar que las partículas están muy unidas unas de otras, por esta razón todo el enjambre llega al mínimo global. Ocurren casos donde las partículas quedan atrapadas en un mínimo local pero esto ocurre en 1 de 10 situaciones. Para los casos donde el W es mayor a 0,5 se observa que las partículas están más distantes y que solo una porción del enjambre llega al mínimo, las demás partículas se dispersan en la función. El número de iteraciones aumenta en una pequeña porción al aumentar el peso de inercia W , esto se debe a que las partículas se mueven con mayor libertad lo cual toma tiempo en la convergencia de la misma. El mejor valor encontrado para este parámetro es de 0,5, ya que presenta un número pequeño de iteraciones y las partículas permanecen unidas en el mínimo.

La variación de las constantes $C1$ y $C2$ presentan una leve influencia en el número de iteraciones, esto se debe a que la asignación de pesos en las búsquedas locales

y globales influyen en la rapidez de la convergencia del enjambre. De acuerdo a las observaciones en la ejecución del algoritmo se tiene que a medida que $C2$ se va acercando a $C1$ las partículas cada vez permanecen más unidas y llegan con rapidez al mínimo global; para los casos donde el valor de $C2$ esta por arriba de $C1$, las partículas tienden a estar un poco más distantes entre ellas, sin embargo el movimiento del enjambre es rápido y llegan al mínimo en pocas iteraciones. Para los casos donde $C1$ aumenta por encima de $C2$ el enjambre se comporta muy similar para todos los casos, es decir llega con rapidez y las partículas permanecen muy unidas, por tal razón no es posible ver diferencia alguna.

Teniendo en cuenta lo anterior es mejor darle un mayor peso a la componente global $C2$ por encima de la componente global $C1$, teniendo en cuenta que dicho distanciamiento en valores no sea tan lejano.

4.2. Función de Ackley

En la implementación de esta función se llegó a la misma conclusión que la función básica en cuanto al radio de avance de las partículas, este no debe ser de un orden muy pequeño ya que el avance de las partículas es muy lento ocasionando que el número de iteraciones incremente de manera exorbitante llegando a 1000 iteraciones sin progreso de las partículas. Aún y cuando algunas de las partículas llegan al punto de interés, no todo el enjambre lo hace lo que produce poca confiabilidad en los resultados.

Aunque el número de iteraciones se ve más afectado por el del radio de avance, la masa de inercia también influye en esta condición (entre más alto sea W , será mayor el número de iteraciones). Además se observó que los dos parámetros tanto el W como el radio de avance se pueden complementar ya que si se decide utilizar un radio de avance del orden de 10^{-3} se puede compensar con un W de 1. En este caso las partículas llegan al mínimo, pero debido al peso de inercia alto con la cual se desplazan las partículas, vuelven a dispersarse creando un ciclo en donde se acercan y se alejan del mínimo objetivo sin llegar a la convergencia.

El aumento o disminución de las constantes $C1$ y $C2$ producen un cambio en el número de iteraciones, donde se tiene que para valores entre el rango de 1,5 a 2,5, el número de iteraciones es menor con valores cercanos a 17, a diferencia de valores por fuera del rango en donde el número de iteraciones aumenta. Las características correspondientes a los comportamientos del enjambre conforme se varían las constantes $C1$ y $C2$, son muy similares a las de la función básica por tal razón se obvia este análisis.

4.3. Función Beales

Para valores de masa de inercia entre 0,1 y 0,3 las partículas no convergen a un punto específico, algunas se dispersan y otras conforman una zona muy cerca del mínimo al cual no se acercan. El incremento para la masa de inercia entre 0,5 y 0,7 genera durante las primeras iteraciones una tendencia a agruparlas en la zona del mínimo, pero no todas llegan a dicho valor, varias presentan dispersión hacia una región contraria. Un valor para masa de inercia de 0,9 genera una mayor convergencia de las partículas hacia el mínimo global, sin embargo en algunas de las pruebas realizadas se produce un número elevado de iteraciones que desencadena un comportamiento inesperado, las partículas se desplazan al lado contrario de la ubicación del mínimo global.

En la variación de las constantes $C1$ y $C2$ se observa que a medida que aumenta $C2$ acercándose al valor de $C1$, las partículas llegan con más rapidez al punto donde converge el enjambre por esta razón el número de iteraciones va disminuyendo. Para el caso en que $C2$ es menor que $C1$ se evidencia que algunas partículas quedan a la deriva y pierden la referencia del enjambre, por tal razón las iteraciones son mayores debido a que toma tiempo el que todas las partículas lleguen al lugar del enjambre. En el caso en que $C2$ aumenta por encima de $C1$ los resultados son muy buenos puesto que el enjambre converge con rapidez y las partículas permanecen muy unidas. La diferencia entre una prueba y otra es el número de iteraciones que aumenta pero un factor mínimo. En ocasiones ocurre que el enjambre se parte en dos una apuntando hacia el mínimo global y el otra hacia un local, lo bueno de estas situaciones es que se evidencia como poco a poco las partículas van pasando de un grupo a otro dirigiéndose hacia el mínimo global.

Teniendo en cuenta lo anterior para este tipo de funciones donde el mínimo se encuentra en medio de una gran planicie, es oportuno contar con un W alto que garantice un mayor dinamismo en las actualizaciones de las partículas. De igual manera, es ventajoso asignarle un mayor peso a la componente $C2$ con respecto a la componente $C1$, sin embargo es recomendable que sus valores sean muy cercanos garantizando así que las partículas permanezcan más unidas en todo su recorrido.

4.4. Función Easom

Al hacer una variación de la masa de inercia desde 0,1 a 0,7 se puede observar que todas las partículas llegan al mínimo global con número de iteraciones bajo (100 iteraciones aproximadamente). Con una masa de inercia de 0,9 las partículas convergen en un número de iteraciones más alto en comparación con el experimento anterior (318 iteraciones aproximadamente), sin embargo todas las partículas convergen al mismo punto. Con una masa de inercia de 0,9 también se puede observar que en algunos experimentos, hay una parada rápida del algoritmo ya que las partículas quedan sin moverse mucho cada una en un lugar del espacio de búsqueda (durando 6 iteraciones); esto es un problema de convergencia prematura que hace que las partículas no lleguen al mínimo global sino que queden dispersas en el espacio de la

función objetivo a causa de no encontrar un mejor punto.

El mejor valor encontrado para W es de $W = 0,1$, la razón por la cual este valor es propicio para esta función se debe a que es ventajoso que las partículas se muevan todas juntas acercándose así al mínimo de manera uniforme y no desordenada como en otros casos. Esto hace que el número de iteraciones sea mínimo. Para este tipo de función se debe tener en cuenta la manera en cómo se interpretan los datos dado que en casi todos los experimentos el algoritmo converge, la rapidez con la que lo hace y la precisión con que encuentra el punto de interés constituyen el factor decisivo en la elección de los mejores parámetros. De igual manera, la razón por la cual muchas veces el enjambre converge prematuramente en un punto cualquiera, es un factor a tener en cuenta en la decisión.

De acuerdo con las variaciones de las constantes $C1$ y $C2$ se observó que para valores de $C2$ por debajo de 1,8 se observa que hay poca comunicación entre las partículas dado que cada partícula actualiza su posición y no se le ve un avance progresivo hacia el mínimo, también ocurre que poco a poco cada partícula por si sola encuentra una buena posición y se queda quieta. Para el caso en que $C2$ aumenta por encima de $C1$ se observa que las partículas convergen con mayor rapidez, lo cual genera un menor número de iteraciones teniendo en cuenta la aclaración de sesiones pasadas con respecto al análisis de resultados en el promedio de iteraciones.

A partir de lo anterior se concluye que para este tipo de función el peso de la inercia debe ser bajo respondiendo a la necesidad de ser precisos en la búsqueda del mínimo, de igual manera se ve la necesidad de darle un mayor peso a la componente global con respecto a la local, de modo que la búsqueda se haga generalizada pero con avances minuciosos.

4.5. Función Esfera

Si la masa de inercia es igual a 0,1, la cantidad de iteraciones oscila entre 28 y 33, además las partículas se desplazan hasta llegar al mínimo global. No es común observar partículas en la periferia del valor mínimo, todas se acumulan en este valor. El aumento del valor de la masa de inercia entre 0,3 y 0,7 en pasos de 0,2 genera una cantidad de iteraciones menor (entre 24 y 28 iteraciones), sin embargo no todas las partículas se ubican encima del valor puntual, por el contrario se genera una zona alrededor del mínimo, donde se concentran las partículas. Con el incremento de la masa de inercia a 0,9 la cantidad de iteraciones necesarias para alcanzar el mínimo global se duplicó, además se formó una zona alrededor de la cual se puede identificar el punto de ubicación del mismo.

Teniendo en cuenta lo anterior, para este tipo de función el valor adecuado para W es de 0,1 la razón por la cual debe ser un valor pequeño se debe a la morfología de la función, puesto que el mínimo es único y no tiene problemas por sesgo, la mejor manera de llegar a el es de manera uniforme, rápida y orientada. A medida

que aumenta W el número de iteraciones aumenta dado que el avance de las partículas es mayor, ocurre el caso en q las partículas están prontas a llegar pero en una actualización vuelven a salir un poco del área óptima en la que estaban. Se puede evidenciar que para un valor muy alto de W las partículas llegan al mínimo con rapidez y debido a la gran inercia que llevan no son capaces de frenar en el mínimo sino que debido a esta velocidad llegan y se alejan de manera constante. Esto lleva a que el número de iteraciones aumente significativamente.

En cuanto a la influencia de las constantes $C1$ y $C2$, los análisis pertinentes a este tipo de funciones con nivel de complejidad bajo son muy similares. Por tal razón se considera que el valor de estas variables es mejor que sea cercano y se se le de un leve peso mayor a la componente global.

4.6. Función Goldstein Price

En términos generales para esta función se evidencia una rápida convergencia de las partículas en pocas iteraciones, todos los experimentos llegaron al mínimo global de la función y no se observa dispersión entre los valores que toman las partículas al ser evaluadas en la función objetivo. Al asignar a la masa de inercia un valor de 0,3 el número de iteraciones promedio es bajo (alrededor de 25 iteraciones), la convergencia hacia el mínimo es rápida para un valor de 0,7 en la masa de inercia, el número de iteraciones aumenta, las partículas forman un cúmulo uniforme cerca de la región donde se ubica el mínimo, al final todas convergen. Con una masa de inercia de 0,9 incrementa el número de iteraciones, las partículas exploran un poco más antes de llegar al mínimo global, la convergencia de las partículas hacia un único valor es una característica muy marcada.

Teniendo en cuenta lo anterior, para este tipo de función es recomendable un peso de inercia bajo que garantice la rapidez de la convergencia a su vez la uniformidad de las partículas conforme van avanzando, en cuanto a la influencia que tiene las constantes $C1$ y $C2$, se conserva la idea de que los valores sean cercanos y se tenga un mayor peso en la componente global debido a que la función presenta una gran planicie. La tabla donde se encuentra el mejor conjunto de valores para los parámetros de acuerdo con los resultados se puede observar en la figura 1.

4.7. Función Rastrigin

La función presenta un nivel de dificultad considerablemente alto debido a que presentan varios mínimos y máximos locales, sin embargo se tiene la ventaja de que existe un solo mínimo global. Cabe resaltar que la función facilita la llegada al área del mínimo puesto que es único, por tal razón en la mayoría de las pruebas se llega al punto mínimo, su factor de relevancia lo determina el número de iteraciones. El radio de avance influye de manera directa en el número de iteraciones, debido a que representa qué tanto se mueven las partículas en el espacio de búsqueda, el mejor

valor de radio de avance fue de 0,5 puesto que las partículas se sitúan alrededor del mínimo y finalmente todas convergen al punto en pocas iteraciones. Por el contrario para valores inferiores a 0,5 se presentan casos donde algunas partículas quedan moviéndose alrededor de un máximo (aunque cercano al mínimo global).

Para valores de radio de avance superiores a 0,5 se tiene que las partículas se mueven con mayor distancia entre ellas ocasionando un mayor número de iteraciones, sin embargo al final las partículas convergen en su gran mayoría al punto mínimo, por su parte la masa de inercia determina cuanto se agrupan las partículas en el enjambre y en cierta manera también influye en la cantidad de iteraciones. Para valores superiores a 0,5 todo el enjambre converge al mínimo global, caso contrario para valores menores a 0,5 donde ocurre que el criterio de parada no se cumple mientras que el del límite de saturación por iteraciones máximas si, por tal razón las partículas quedan en medio de su trayectoria sin llegar al punto de interés.

Para el caso en donde W esta entre 0,5 y 0,7 las partículas llegan alrededor del mínimo y luego de un par de iteraciones todas las partículas convergen al punto de interés; para el caso donde W es 0.9 las iteraciones son mayores, sin embargo todo el enjambre converge al final. Finalmente el mejor valor de W para esta función correspondió a 0,7 donde el número de iteraciones fue menor.

La influencia en cuanto a los valores de las constantes $C1$ y $C2$ se evidencia de manera importante para los casos en que $C2$ aumenta por encima de $C1$, donde se observa que el enjambre tiende a explorar mejor toda el espacio de búsqueda, lo más importante que ocurrió, es que en ocasiones el enjambre convergía a un mínimo local pero unas pocas partículas seguían explorando; cuando estas partículas llegaban al mínimo global inmediatamente el enjambre llegaba a la posición de esta partícula. caso contrario para las pruebas anteriores en donde en ocasiones convergían muy cerca del global terminando en un local.

De acuerdo con lo anterior se concluye que el radio de avance para este tipo de funciones debe ser alto garantizando así una mayor búsqueda generalizada, sin embargo con el fin de intentar estandarizar este parámetro para todas las funciones, se establece que el valor intermedio de 0,5 es bueno. Además de esto se considera que el valor de la inercia de ser alto conservando la característica de una búsqueda global, de igual manera se establece que la componente global debe tener un mayor peso con respecto a la componente local ratificando así las características anteriormente mencionadas.

4.8. Función Valle de Rosenbrock

Igual que en la función anterior el parámetro de radio de avance influye de manera directa en el número de iteraciones, para variaciones de este parámetro se observa que el mejor valor encontrado es 0,5 debido a que el enjambre converge con rapidez. Para valores entre 0,1 y 0,5 el número de iteraciones aumenta debido a que el avance

de las partículas es pequeño ya que algunas partículas se quedan en la trayectoria del enjambre y convergen con mayor número de iteraciones con respecto al promedio.

El parámetro W presenta un influencia en el número de iteraciones además de influir en la manera cómo se dispersan o agrupan las partículas. Para valores menores a 0,7 se tiene que las partículas se mueven muy despacio y permanecen unidas todo el tiempo lo que les facilita encontrar un punto mínimo. Para valores mayores a 0,7 las partículas se van moviendo con un mayor desplazamiento permitiendo explorar gran parte de la planicie, sin embargo el enjambre converge luego de varias iteraciones. El mejor valor para W fue de 0,5 bajo el cual las partículas se agruparon por completo y el número de iteraciones en este proceso fue mínimo.

Respecto a la influencia de los parámetros $C1$ y $C2$, se observa que a medida que $C2$ va aumentando hasta llegar al valor de $C1$, las partículas se agrupan con mayor rapidez que cuando $C2$ es muy pequeño. En ocasiones ocurre que queda una partícula por fuera de la aglomeración del enjambre lo cual retarda la convergencia del algoritmo. Para el caso en que $C2$ es menor a $C1$ se observa que la exploración no es completa dado que se botan las primeras partículas y dependiendo su mejor posición esta arrastrará a las demás. Cuando $C2$ aumenta por encima de $C1$ se observa una más rápida agrupación de las partículas, lo cual genera que el enjambre se forme y llegue al punto de interés, resultando un menor número de iteraciones. Para los casos donde $C1$ aumenta por encima de $C2$ se puede observar que el enjambre se agrupa con rapidez pero siempre quedan algunas partículas distantes que evitan que el enjambre converja en su totalidad en el mínimo global. A partir de lo anterior se destaca la ventaja de poseer una mayor búsqueda global que local, que en este caso se observa más en la manera como se agrupan las partículas.

4.9. Función Schwefel

En este tipo de función se tienen 4 mínimos globales, por lo tanto se tomó cualquiera de ellos como un mismo punto puesto que la elección por parte de las partículas hacia ese punto lo determina las posiciones iniciales de las mismas.

Dada la complejidad de la función donde se presentan varios mínimos y máximos, se tiene que para un radio de avance del orden de 10^{-2} ó 10^{-3} el número de iteraciones es significativamente mayor (1000 iteraciones), además ocurren casos en que las partículas se quedan en un máximo, es decir no convergen al punto de interés. El mejor valor de radio de avance es de 0,5 puesto que permite que el avance de las partículas no sea ni muy grande ocasionando que estas se salgan de la frontera, ni muy pequeño ocasionando una convergencia prematura.

La influencia de W en esta función tiene que ver con el nivel de agrupación del enjambre una vez se ha cumplido el criterio de parada. Para valores mayores a 0,7 las partículas convergen al área (no al punto) de manera dispersa, por otro lado para valores en donde el W es menor a 0,7 se obtiene a todo el enjambre ubicado en el

mínimo global.

El valor más adecuado para la masa de inercia fue de 0,5. Para valores por debajo de 0,5 ocurren casos en que el número de iteraciones aumenta debido a que algunas partículas se quedan explorando en el espacio solución. Para valores por encima de 0,5 ocurren casos donde muchas partículas permanecen quietas debido a que se salen constantemente de las fronteras.

Con respecto a los mejores valores de $C1$ y $C2$, se encuentra gran similitud con respecto a la función Rastrigin, sin embargo los valores que presentaron un mejor resultado fueron de 2,2 para ambas constantes. De esta manera se le daba el mismo peso a la búsqueda global y local y conforme al valor de W , la búsqueda en este tipo de función se hacía de pasos largos pero un poco de manera uniforme.

4.10. Función Step

Para la implementación del algoritmo en esta función, entre más grande es la masa de inercia más dispersas quedan las partículas. Para una masa de inercia de 0,9 las partículas no encuentran el mínimo global aunque quedan cerca de este, en ocasiones ocurre que algunas partículas se quedan dando vueltas por el espacio de búsqueda ocasionando que el algoritmo no converja sino que se detenga por llegar al máximo de las iteraciones permitidas, llegado esta situación se evidencia una alta dispersión en las partículas. El mejor valor de W es un valor bajo puesto que debido a que la función es de bajo nivel de dificultad y el acceso al mínimo global es directo desde cualquier punto, es mejor garantizar un progreso compacto, uniforme y con velocidad intermedia.

En cuanto a los valores del radio de avance y las constantes $C1$ y $C2$, los requerimientos son muy similares a los de la función esfera por tal razón el mejor valor para los parámetros es de 0,5 para el radio de avance, y un mayor peso en la componente global con respecto a la componente local. La tabla donde se encuentra el mejor conjunto de valores para los parámetros de acuerdo con los resultados se puede observar en la figura 1.

4.11. Función Stochastic

La variación del radio de avance en esta función de prueba influye de manera directa en el número de iteraciones. Entre más pequeño sea este parámetro, es más probable que el número de iteraciones aumente debido a que el cambio de posición de una partícula implica un mayor tiempo para que esta converja con el enjambre.

Debido a la morfología de la función, se hicieron pruebas variando el radio de avance con el fin de determinar si para este tipo de función el valor del parámetro cambia con respecto a las demás funciones. Para un radio de avance de 0,001 se tiene

un número de iteraciones alto (alrededor de 600 iteraciones), bajo estas condiciones se tiene que el avance es muy lento por tanto las partículas recorren con lentitud la función y poco a poco se van agrupando en un mínimo. Para un radio de avance de 0,01 el número de iteraciones disminuye llegando a 90 iteraciones y por último para un radio de avance de 0,5 el número de iteraciones es de 38 iteraciones. Teniendo en cuenta lo anterior, una disminución de una década de este parámetro evidencia una reducción del 85% en el número de iteraciones, la razón se debe a que el avance de las partículas es mayor a medida que el radio de avance aumenta por lo tanto las partículas se van a mover más rápido hacia el punto que converge el enjambre.

Para variaciones del parámetro del radio de avance entre 0 y 1, se observa que entre más cercano se está a 0,5, las partículas se mueven con mayor velocidad y llegan al mínimo en un menor número de iteraciones al igual que con un mayor agrupamiento en el enjambre (Partículas más unidas).

En cuanto a la variación de la masa de inercia se confirma que al aumentar o disminuir este parámetro se obtiene un mayor énfasis en la exploración global y local. Además se observa la influencia del mismo en el número de iteraciones, según las pruebas realizadas a medida que aumenta el W el número de iteraciones también lo hace, y a medida que disminuye W las iteraciones disminuyen. Cuando el W es menor a 0,5 las partículas se mueven muy unidas unas de otras y su radio de avance es pequeño, por el contrario para el caso donde W es alto (0,9) se tienen partículas distantes entre ellas y su avance es mayor, por lo cual les permite explorar más área.

En cuanto a la influencia de las constantes $C1$ y $C2$, es muy difícil lograr ver el comportamiento de las partículas debido al alto costo computacional que requiere la función, sin embargo es de rescatar que en base a las observaciones de las demás funciones, el valor de estas constantes debe ser cercano y dándole una mayor prioridad a la componente global. Por otra parte, es de aclarar que en esta función el valor de W se esperaba alto, sin embargo los mejores resultados se obtuvieron para un W bajo debido a que el número de iteraciones era menor con respecto a los demás experimentos.

4.12. Función Holder Table

Debido a que la función presenta cuatro mínimos equidistantes, se va a considerar cualquiera de ellos como el mínimo global de la función. La manera como se distribuyen las partículas inicialmente determinan el mínimo al cual apunta el enjambre.

El valor que produjo mejores resultados en cuanto al número de iteraciones fue un radio de avance de 0,5, de igual manera que en las demás funciones es necesario tener un equilibrio en el radio de avance, que no se a muy rápido pero tan poco muy lento.

La variación del parámetro W no influye directamente en el número de iteracio-

nes, pero sí influye en la manera en qué tanto se agrupan las partículas al cumplirse el criterio de parada. Para un W alto, las partículas terminan alrededor del mínimo pero un poco distantes, por el contrario para un W pequeño las partículas convergen a un mismo punto y la distancia entre cada partícula es muy mínima. El valor para el parámetro W que presentó un menor número de iteraciones fue de 0,7.

Las consideraciones que se deben tener en cuenta para asignar un valor adecuado a las constantes $C1$ y $C2$, se pueden deducir a partir de las demás funciones con características similares a esta. La razón por la cual no se detalla en este tipo de función se debe a que la observación del movimiento del enjambre no se puede ver debido al alto costo computacional que requiere la función.

4.13. Función Egg Holder

En las pruebas realizadas para esta función se determinó que para una masa de inercia igual a 0,1, la función en general no tiene una convergencia de todas las partículas hacia un mismo punto. Por su parte, si la masa de inercia es igual a 0,3 se observa que todas las partículas llegan al mismo punto después de muchas iteraciones pero el mínimo alcanzado aparentemente es global.

Cuando $W = 0,5$ y $w = 0,7$, las partículas presentan convergencia al mismo punto, sin embargo si $W = 0,9$ el algoritmo no presenta convergencia al mismo punto, por lo tanto no encuentra el mínimo global aunque sí un valor cercano a un mínimo local. El mejor valor de masa de inercia es de 0,7 considerando la manera en como llegan las partículas y el número de iteraciones requerido para converger. La demora en la convergencia de las partículas a un mismo punto se presenta porque independientemente de que alguna está en el mínimo global las otras tratan de acceder a este punto, pero al estar ubicadas en niveles superiores de la función no convergen tan rápido en el mínimo global.

Las consideraciones necesarias para las constantes $C1$ y $C2$ son muy similares a las de las funciones anteriormente analizadas, donde básicamente se rescata la idea de que es mejor darle un mayor peso a la componente global por encima de la componente local.

5. Análisis y Discusión

Dentro de los temas tratados en esta sesión se comentó que es posible modificar los algoritmos para hacer que se obtengan mejores resultados, que sean más rápidos o incluso más precisos. Una propuesta que queda para trabajos posteriores es la idea de implementar parámetros variables, es decir que según vaya evolucionando el algoritmo en cada iteración, algunos de los parámetros requieran un cambio que se ajuste a los resultados que se van obteniendo y encamine la búsqueda a los que se quieren obtener; la idea es que este cambio se haga de manera automática sin

Mejores valores			
Función	w	c1	c2
Esfera	0,2	1,9	1,8
Ackleys	0,1	1,8	2,4
Goldstein_Price	0,3	1,9	1,8
Holder Table	0,7	1	1,8
Schwefel	0,5	2,2	2,2
Step	0,1	1,8	2,1
Stochastic	0,1	2,6	2,6
Básica	0,5	1	1,8
Beales	0,5	1,8	1,9
Eggholder	0,7	1,8	2,1
Easom	0,1	1	1,8
Rastrigin	0,7	1	1,8
Rosenbrock	0,5	1,8	2,1

Figura 1: Mejores Resultados. Fuente: Autores

necesidad de tener que reasignarlo, de modo que se permita guardar la evolución de las partículas y no perder la información obtenida hasta la reasignación del valor del parámetro. Estas modificaciones quedan propuestas para trabajos posteriores ya que la idea del trabajo de cada semana es seguir la línea del seminario sin desviarse en implementaciones que si bien pueden generar buenos resultados podrían interferir en los objetivos del seminario.

En cuanto a la implementación del algoritmo en Matlab algunas líneas de código se podrían optimizar, como es el caso de la actualización de posición y velocidad que se programaron dentro de un ciclo de repetición. Esta modificación se podría hacer utilizando las propiedades de matrices y reduciendo las líneas de código.

Otro tema a tratar fue la correcta metodología utilizada en la recopilación de pruebas en la implementación de las distintas funciones. En el grupo de trabajo se cometió el error en un principio de cambiar las constantes de exploración local y global al mismo tiempo, por lo cual estas pruebas no contenían los aspectos que se querían analizar, como lo es la influencia que se tiene al asignarle un mayor peso a una componente que otra, a partir de esta observación el grupo de trabajo procedió a volver a realizar las pruebas esta vez dejando una constante fija y la otra variable y viceversa. La recopilación y exposición de los resultados ya fue expuesta en esta memoria, sin embargo fue necesario nombrarlo puesto que el error podría cometerse nuevamente en trabajos de este tipo; por otra parte porque la evidencia de estos resultados también se encuentra en las tablas de la carpeta Tablas - sesión 5.

6. Conclusiones

- La variación de la masa de inercia es un factor que afecta de manera directa la convergencia de las partículas, junto con la manera en como se agrupan en su movimiento progresivo; de igual manera afecta el número de iteraciones sobre el cual converge el enjambre. Con base en los resultados obtenidos una masa de inercia intermedia, facilita la búsqueda del punto de interés, quedando un equilibrio en los factores a que este parámetro tiene influencia.
- El radio de avance es otro parámetro que determina de manera directa que tan dinámica se quiere hacer la búsqueda en cada actualización, de acuerdo con los resultados obtenidos, se recomienda usar un valor intermedio que permita a las partículas realizar una actualización equilibrada y acorde al espacio solución que se define de acuerdo a las fronteras de la función.
- El dinamismo en la actualización de las partículas se puede compensar con los parámetros W y dt , esto se corroboró en pruebas donde el comportamiento era similar para un dt bajo junto con W alto y viceversa. A pesar de esto se decidió elegir un valor intermedio para ambos parámetros teniendo en cuenta también los resultados obtenidos.
- Los valores de las constantes $C1$ y $C2$ influyen de manera directa en la manera como se comunican las partículas y también en la manera como se agrupan en su movimiento progresivo. La conclusión que se pudo obtener a partir de los resultados, apunta a que es mejor darle un peso mayor a la búsqueda global con respecto a la local, por un lado porque se esta trabajando con un algoritmo de optimización global, pero por el otro, porque es necesario intentar reconocer muy bien el espacio solución y no las pequeñas zonas donde tentativamente se encuentra el punto de interés.

Memoria6

Uso de técnicas de Optimización Global para resolver problemas de Inversión Sísmica

Edgar Téllez Cortés, Johao Coronel Silva, Juan Arias Pinto

Universidad Industrial de Santander
Escuela de Ingenierías Eléctrica, Electrónica y Telecomunicaciones

Junio 18 de 2014

Sesión 6: Inversión Sísmica

1. Introducción

En esta sesión del seminario de investigación se procedió a explicar la herramienta SEISMIC UNIX (SU), con la cual se generó un modelo de capas del subsuelo y de igual manera se simuló el proceso de adquisición de trazas sísmicas. Dicho modelo va a ser utilizado como modelo de referencia a lo largo del seminario, por tal razón se explicó la manera como se decidió crear el modelo, los valores de *sloths* del mismo y la geometría especificada para el modelo. También se mostró la forma en la que se empalmó esta herramienta con las implementaciones hechas en Matlab del algoritmo PSO.

La idea principal que se quiso exponer en esta sesión, fue la de realizar una inversión sísmica basada en el modelo, donde se creaba un modelo de 4 capas del subsuelo llamado modelo de referencia y poner a prueba el algoritmo PSO como herramienta para dar solución al reto de encontrar el valor de los *sloths* de cada capa del modelo de referencia a partir de la información proveniente de las trazas sísmicas adquiridas en un proceso de adquisición sintético del modelo de referencia con ayuda de la herramienta SU. Cabe resaltar que la geometría utilizada para el modelo de referencia creado fue una geometría fija.

2. Seismic Unix

Seismic Unix (SU por sus siglas en inglés), es un paquete de aplicaciones para el tratamiento de señales sísmicas apoyado por el Centro de Fenómenos Ondulatorios

de la Escuela de Minas de la Universidad de Colorado, en Estados Unidos [1].

2.1. Estructura de Seismic Unix

Todas las rutinas de Seismic Unix, se pueden ejecutar desde una terminal de Unix, bajo sistemas operativos Linux o MAC OS-X. Se evidencia máxima eficiencia bajo esta configuración si se utiliza junto a las rutinas, las técnicas asociadas al núcleo Bourne, conocido como Bourne Shell (sh) y el núcleo extendido Bourne Again Shell (bash).

La filosofía básica de SU, al igual que Linux y sus distribuciones asociadas tal como Free BSD Unix, entre otras, se basa en los siguientes principios [2].

- Sencillo es más bello.
- Un programa para cada tarea.
- Para una tarea nueva, escribir un programa nuevo.
- Portabilidad mejor que eficiencia.
- Haz partes simples conectadas mediante interfaces limpias y bien definidas
- Pensar en paralelo.

2.2. Comandos Básicos

2.2.1. Rutina Simple para visualizar sismogramas como trazas sísmicas

```
$ suxwigb <seismogram.su
```

Este comando permite llamar un archivo denominado seismogram.su y representar mediante trazas sísmicas (wiggles, en inglés), un sismograma [1].

2.2.2. Rutina para representar mediante imagen un modelo de capas de un sismograma

```
$ suximage <seismogram.su
```

Mediante el uso de este comando, es posible partir de un archivo, representar un modelo de capas por medio de una imagen.

2.3. Datos SEG-Y

El formato de archivo SEG-Y es desarrollado y mantenido por la Sociedad de geofísicos de exploración para el almacenamiento de datos sísmicos. SEG-Y es un formato abierto y se utiliza sobre todo para almacenar una sola línea de datos digitales sísmicos en cintas magnéticas [3].

2.4. Otros comandos útiles [4]

2.4.1. Compresión y Empaquetado de Datos

supack1 : Función que permite comprimir los datos de trazas tipo seg-y en caracteres.

suunpack1 : Permite descomprimir los datos de trazas tipo seg-y que se comprimieron como caracteres, a la forma de punto flotante.

supack2 : Comprime datos de trazas tipo seg-y a formato de 2 bytes de baja precisión.

suunpack2 : Descomprime datos tipo seg-y del formato de baja precisión, a formato de punto flotante.

2.4.2. Transformadas

wpc1comp2 : Comprime una sección de sísmica 2D traza por traza utilizando los paquetes Wavelet.

wpc1uncomp2 : Descomprime una sección 2D que ha sido previamente comprimida utilizando los paquetes Wavelet.

wpcuncompress : Descomprime una sección 2D.

wptcomp : Compresión mediante paquetes de compresión Wavelet.

wptuncomp : Descomprimir datos basados en Paquetes Wavelet.

wtcomp : Compresión mediante Transformada Wavelet.

2.4.3. Filtrado

sudipfilt : Establecer la mejor pendiente de un filtro en el dominio de f.

sufxdecon : Atenuación del ruido aleatorio mediante deconvolución.

sumedian : Mediana de un filtro especificada sobre una curva poligonal definida por el usuario con la distancia a lo largo de la curva especificada por la palabra clave del cabezal.

sukfilter : Filtro poligonal radialmente simétrico en función de la función Sen^2 .

2.4.4. Otros comandos de uso común

maxdiff : Determina la máxima diferencia absoluta entre dos datos seg-y.

segyhdrmod : Reemplaza el encabezado de un archivo segy.

suaddnoise : Agrega ruido al conjunto de trazas.

sucmp : Compara un conjunto de dos trazas sísmicas y retorna un valor cero si son iguales ó un valor de 1 si son diferentes.

sudiff, susum, suprod, suquo : Realiza la diferencia, suma, producto y cociente de dos trazas en SU, utilizando la palabra *suop*.

suacor : Establece la autocorrelación existente entre dos trazas sísmicas.

suop2 : Realiza una operación entre trazas en binario.

suressamp : Realiza nuevamente muestreo en el dominio del tiempo.

suxcor : Establece la correlación entre una traza y un filtro proporcionado por el usuario.

2.5. Aplicación de Seismic Unix

Para realizar las simulaciones de una adquisición sísmica en primer lugar es necesario crear un modelo de capas del subsuelo sobre el cual se hará un trazado de rayos que represente las ondas que cierto número de geófonos detectarán a partir del disparo de una fuente sintética sobre dicho modelo. Cabe aclarar que el modelo que se creó fue uno con una geometría fija y con cuatro capas únicamente.

La creación del modelo de capas del subsuelo, se realizó con la ayuda del segundo grupo del seminario de investigación, quién realizó un aporte en la familiarización con la herramienta SU y su manejo. El primer paso para crear el modelo de referencia corresponde a la creación un archivo de texto con extensión *.sh*, en el cual se especifica el tamaño de la región en dos dimensiones del subsuelo, así como la distribución y el número de capas que se van asignar al modelo. De igual manera se establece la geometría del mismo y se asignan los valores de los *sloths* de cada capa. Cabe recordar que el *sloth* hace referencia al inverso de la velocidad al cuadrado de la capa del subsuelo.

$$Sloth = \frac{1}{v^2} \quad (1)$$

Después de tener este archivo de texto con todas las especificaciones necesarias para su adecuado funcionamiento y habiendo instalado previamente el software SU, se abre una terminal en el sistema operativo Ubuntu para generar dos archivos con extensiones *.eps* y *.dat*, el primero de ellos para poder visualizar el modelo en 2D y el otro para servir de entrada a otro documento de texto que generará la adquisición y entregará como resultado las trazas sísmicas.

La visualización del modelo de capas del subsuelo a partir del archivo con extensión *.eps* da una tonalidad diferente a cada capa dependiendo de los *sloths* asignados en programación y permite hacer una diferenciación de colores que permite comparar qué tan parecidos son dos modelos sin necesidad de ver los valores de *sloths* correspondientes.

La adquisición de las trazas sísmicas del modelo ya creado, se hace a partir de un nuevo documento de texto con extensión `.sh`, en el que se especifican parámetros como: número de geófonos, espaciado entre geófonos, profundidad de los geófonos, rango de offset de los geófonos, número de disparos, profundidad de la fuente, rango que emana la fuente, muestreo y tiempo de muestreo. Cabe aclarar que la herramienta utilizada para realizar dicha adquisición fue la herramienta *triseis* que simula el proceso de adquisición de trazado de rayos. Otra técnica por la cual también se podría realizar el proceso de adquisición por medio de SU es a partir de la ecuación de onda, sin embargo se decidió trabajar únicamente con trazado de rayos en vista de que es una herramienta un poco más sencilla.

Finalmente se ejecuta el archivo de la adquisición con extensión `.sh` por medio de la terminal de Ubuntu, hecho esto se creará un nuevo archivo con extensión `.su` el cual contendrá las trazas sísmicas; luego de esto es posible observar el contenido del archivo con ayuda de los comandos *suxwigb* o *suximage* en caso de querer observar las trazas sísmicas o el sísmograma respectivamente. A partir de estas trazas sintéticas y que por ahora van a ser la información de referencia, es posible empezar a comparar las distintas trazas sintéticas generadas a partir del modelo sintético creado con las capas que el algoritmo PSO arroje como resultado. Los archivos `.sh` necesarios para la creación del modelo y generar su respectiva adquisición se encuentran en la sección 5.6.2 y 5.6.3 del capítulo apéndices.

2.6. Operación conjunta de Seismic Unix con Matlab

Hasta el momento solo se cuenta con la capacidad de crear un modelo, realizar su respectiva adquisición y observar sus correspondientes trazas sintéticas. El paso a seguir corresponde a la manera en como se puede acceder a la información de las trazas por medio de Matlab, de modo que sea posible comparar por medio de un indicador qué tan semejantes son las trazas sintéticas con las trazas de referencia. Teniendo en cuenta lo anterior, después de tener el código en Matlab con la implementación del algoritmo PSO y un modelo de capas del subsuelo con los *sloths* asignados a cada una de estas capas, fue necesario buscar un método para incorporar la herramienta SEISMIC UNIX al algoritmo PSO ya programado, teniendo en cuenta que SU es una herramienta que opera con línea de comandos desde la terminal de ubuntu. La idea principal era que el paso que se hacía en una línea de código en Matlab encargado de evaluar las partículas de búsqueda en la función objetivo, ahora debía generar un nuevo conjunto de valores como resultado, para a partir de ellos asignar los valores de los *sloths* que constituirían las capas del nuevo modelo sintético. Cabe aclarar que hasta el momento la evaluación de las partículas se hacía con respecto a la función de prueba, en este caso dicha función pasa a ser el valor que indica qué tan parecidas son las trazas sintéticas con las trazas de referencia.

Luego de obtener el valor de los *sloths* de las capas y consigo el modelo sintético, se debía proceder a simular una adquisición sísmica del modelo sintético y obtener las trazas sintéticas correspondientes; logrado esto se debían comparar las trazas

con las de referencia y a partir de una métrica adecuada establecer un indicador de semejanza. Finalmente dicho indicador sería entregado al algoritmo PSO de modo que brindara un indicio para saber si este debía detenerse o seguir iterando, es decir seguir creando modelos sintéticos y adquirir sus respectivas trazas a tal punto que se llegara a el valor exacto de los *sloths* del modelo de referencia.

Es así como se implementó una función en Matlab capaz de hacer todo lo mencionado en el párrafo anterior de manera automática, la cual debía ser llamada por el código principal que tenía la implementación del algoritmo PSO. Esta función recibió el nombre de **Inversión Automática**.

2.6.1. Inversión Automática

Es una función creada en Matlab que se encarga de ejecutar desde software líneas en la terminal de Ubuntu, por tal motivo sirve como una integración entre Matlab y SEISMIC UNIX. Dicha función cumple la tarea de generar, adquirir y comparar el modelo sintético de cada partícula actualizada con el modelo real de referencia.

La función tiene como parámetros de entrada los cuatro *sloths* del modelo de capas del subsuelo y una variable que indica la partícula del algoritmo que se va a evaluar; como parámetro de salida tiene un número que indica la máxima diferencia entre los vectores que representan las trazas de referencia y la sintética generada en su momento por el algoritmo.

Listing 1: Inversión Automática

```
1 function [z]=inversion_automatica(cap1,capa2,capa3,capa4,i)
```

Donde:

Z= Es el parámetro de salida.

capa= Parámetro de entrada correspondiente a cada sloth.

i= Parámetro de entrada que indica que partícula se esta evaluando.

Cabe aclarar que el parámetro de salida corresponde a la máxima diferencia entre las trazas sintéticas y las de referencia. Esta métrica se usó simplemente como un indicador de uso provisional con el cual se trabaja debido a que permite obtener una medida de manera sencilla, sin embargo la figura de mérito que indicará la semejanza entre trazas, deberá ajustarse a las características que se buscan en el desarrollo del seminario de investigación.

Para generar el modelo y simular la adquisición sísmica que da como resultado las trazas sísmicas, es necesario tener el modelo y las trazas sísmicas de referencia en una carpeta. Según esto, se decidió asignar a cada partícula de búsqueda una carpeta con los archivos necesarios para el proceso ya mencionado, dentro de cada

carpeta se deberá crear el modelo actualizado junto con las trazas actualizadas para la comparación; además cada carpeta contendrá un archivo llamado Clean.sh que borrará en cada iteración los archivos innecesarios. En este punto cabe aclarar que cada partícula sera asociada a la creación de un modelo sintético, por tal razón en cada carpeta se irá creando de manera automática una serie de modelos hasta lograr conseguir el modelo sintético que se parezca al de referencia, claro esta que por medio de la información de las trazas sísmicas no directamente por el modelo.

Al hacer una analogía con la implementación del algoritmo en funciones prueba, es como si en este caso se implementara en una función de 4 dimensiones en la que el valor del *Sloth* para las cuatro capas será conocido como las coordenadas de una partícula. La diferencia entre las trazas actualizadas y las de referencia representa la evaluación de la función en la partícula actual.

La función inversión automática controla los comandos de SEISMIC UNIX

comando sh modelo_actualizado.sh Para generar el modelo actual correspondiente a la partícula actual.

comando sh adquisicion_actualizada.sh Para simular la adquisición y obtener la traza actualizada correspondiente a la partícula actual.

comandos suxwigg o suximage Si se quiere ver la adquisición sísmica como trazas o como un sismograma respectivamente.

comando maxdiff Que realiza la máxima diferencia entre los vector formados por la traza actual y la traza de referencia.

comando str2num Que se encarga de convertir a decimal el número en formato string entregado por SU al hacer maxdiff.

Teniendo en cuenta que cada partícula es evaluada una y otra vez dependiendo el número de iteraciones del algoritmo en una misma carpeta, el documento de texto utilizado para generar el modelo y trazas actuales en cada iteración debe modificarse, pero solo es necesario cambiar las líneas de correspondientes a los *sloths* del modelo los cuales entrega el algoritmo PSO, esto se puede ver con más detalle en el numeral 6 de la función inversion_automatizada.m mostrada en la sección 5.6.1 del capítulo apéndices.

2.6.2. Estructura de la función Inversión Automática

- 1. De manera automática se crea un documento de texto llamado capas.sh donde se irán cambiando las líneas correspondientes a las capas del modelo, de acuerdo a las coordenadas de cada partícula.
- 2. Se borra el documento modelo_actualizado.sh para volver a crearlo más adelante.

- 3,4,5. Corresponde a las líneas necesarias para copiar tal cual las líneas del modelo de referencia para su creación, pero con la modificación de la definición del nuevo valor de las capas que corresponden a las coordenadas de cada partícula. Para lograr esto se usaron los comandos *cat* y *sed* que permiten copiar y pegar de un documento a otro en líneas específicas. (Para este paso fue necesario tener cuidado con los espacios para poder conservar el formato .sh).
- 6. Se genera el modelo actual, es decir con el valor de las capas de acuerdo a las coordenadas de la partícula actual.
- 7. Se simula la adquisición y se obtienen las trazas correspondientes al modelo sintético.
- 8. Se observan las trazas sintéticas o el sismograma correspondiente.
- 9. Se calcula la máxima diferencia entre las trazas sintéticas y las de referencia.

2.6.3. Modificaciones al Código desarrollado en Matlab

Para implementar el algoritmo PSO enfocado a la inversión sísmica se hicieron ciertas modificaciones del código implementado que se tenía para funciones prueba, teniendo en cuenta que el número de dimensiones ahora era de cuatro. Estas dimensiones representaban los *sloths* de las cuatro capas del modelo del subsuelo. la idea es que ahora el resultado de la evaluación de cada partícula en la función era la diferencia entre las trazas actualizadas y las de referencia, de este modo el algoritmo se podía guiar para saber si una determinada combinación de *sloths* correspondía con un mínimo global.

El código del algoritmo se modificó para que almacenara en cada iteración los valores de los *sloths* encontrados, de este modo sería posible ver la trayectoria de cada partícula comparándola con la combinación de *sloths* real para determinar que tanto se acercaba al mejor resultado.

Finalmente se escribieron unas líneas de código encargadas de graficar el mejor modelo encontrado para hacer una comparación visual con el modelo de referencia y observar qué tan cerca quedó el algoritmo de este, también se imprimió la trayectoria de todas las partículas.

3. Análisis de Resultados

En las pruebas correspondientes a la implementación del algoritmo PSO en inversión sísmica, se partió de los valores recomendados por nuestro grupo de investigación donde :

$$W = 0.5$$

$$dt = 0.5$$

epsilon 1 y 2= 0.1

De acuerdo con las pruebas realizadas, se observa que sin importar el número de partículas y el número de iteraciones, los modelos encontrados son aleatorios; es decir, que no cumplen con la característica de PSO visto en las funciones de prueba puesto que en este caso las partículas no presentan un cambio progresivo hacia el mínimo.

Por tal razón se hace necesario verificar por medio de pruebas, qué valores de parámetros son adecuados para introducirle la heurística al modelo. Teniendo en cuenta eso, se realizaron pruebas donde se logró observar una serie de comportamiento poco comunes.

- Que para valores muy pequeños en el radio de avance ocurre que el criterio de parada se cumple pero las variaciones de las partículas son nulas, es decir que evidencian un cambio progresivo, por tal razón el algoritmo converge con rapidez.
- A medida que aumenta W las coordenadas de la mejor partícula tienden a acercarse al mínimo global, sin embargo las pruebas se hicieron para pocas partículas y en ocasiones el cambio presentado para la mejor partícula es nulo; es decir que hubo una partícula mejor ubicada con respecto a las demás, sin embargo esa partícula se quedó estática iteración tras iteración.
- Para pruebas hechas con 30 partículas y los parámetros recomendados por nosotros, se observa que un valor de W alto presenta un mejor acercamiento al mínimo global, sin embargo se da el caso en que muy pocas veces el algoritmo se detiene por el criterio de parada sino que se detiene por número máximo de iteraciones; es decir que el enjambre nunca convergió en las pruebas realizadas.
- Una prueba que se hizo fue implementar PSO con los mismos parámetros definidos para una función de prueba en particular. Los resultados obtenidos mostraron que este tipo de pruebas no sirven debido a que el modelo encontrado presenta unas coordenadas alejadas del mínimo global (modelo real), por tal razón se piensa que las pruebas hechas con funciones de prueba no sirven de manera directa en las implementaciones que tienen que ver con inversión sísmica.

4. Análisis y Discusión

Dentro del análisis hecho en la presentación de la sesión número 6 se planteó que una de las posibles causas del mal funcionamiento del algoritmo en inversión sísmica es la métrica utilizada para la comparación de trazas, sin embargo se entendió que lo que se buscaba presentar en esta sesión era una implementación del algoritmo en inversión aclarando las consideraciones necesarias para el empalme con el problema sísmico y su correcta interpretación.

A partir de lo ocurrido, se planteó la necesidad de encontrar una figura de mérito que realmente lograra comparar las trazas sintéticas de las de referencia por medio de datos más dicentes puesto que con un solo valor como se dio en el caso de la máxima diferencia, no se lograba tener con detalle esta comparación. La figura de mérito que tentativamente se pensó durante el espacio de discusión fue la métrica de correlación, sin embargo la búsqueda de la métrica que logre encajar con los requisitos que se tienen hasta el momento, se deja abierta en ambos grupos de investigación.

Adicional a los temas tratados con respecto a la presentación de la sesión No 6, se habló acerca de la necesidad de definir cual iba a ser el otro algoritmo de optimización global a implementar. Por tal razón, se definió como próximo compromiso, la definición de la técnica, su correspondiente explicación e implementación por lo menos en una función de prueba. Esto pensando en que es necesario ir avanzando en ambas técnicas y de ser posible complementando los avances que se tengan de acuerdo a las mejoras por realizar en las siguientes sesiones.

5. Conclusiones

Como tal no se lograron obtener en el desarrollo de esta sesión, más bien se logró entender con claridad cómo interpretar el problema sísmico y cómo poder tratarlo de acuerdo a lo que se venía trabajando en sesiones pasadas. Por otra parte se llegó al punto donde se requiere integrar conocimientos de señales dado que la comparación que se necesita entre dos trazas distintas, corresponde a un problema que de acuerdo a la teoría de señales se puede dar solución.

Adicional a esto, los resultados obtenidos en la primer implementación del algoritmo PSO en inversión sísmica no permitieron lograr extraer conclusiones puesto que los resultados no fueron los esperados. A pesar de esto se rescatan las ideas que se tuvieron entorno a estas complicaciones y se da vía libre para soluciones que se puedan ocurrir y solucionar la problemática encontrada.

Referencias

- [1] Wikipedia. Seismic unix. http://en.wikipedia.org/wiki/Seismic_Unix.
- [2] Henry Barrientos. Filosofía unix. <http://neobrr.wordpress.com/2009/02/04/filosofia-unix/>.
- [3] http://www.ehowenespanol.com/convertir-segd-segy-como_231398/.
- [4] David FOREL, BENZ Thomas, and PENNIGTON Wayne. *Seismic Data Processing with Seismic Unix*. Lawrence Gochioco, Series Editor, 2005.

Memoria7

Uso de técnicas de Optimización Global para resolver problemas de Inversión Sísmica

Edgar Téllez Cortés, Johao Coronel Silva, Juan Arias Pinto

Universidad Industrial de Santander
Escuela de Ingenierías Eléctrica, Electrónica y Telecomunicaciones

Julio 7 de 2014

Sesión 7: Optimización mediante Algoritmos Genéticos

1. Introducción

La segunda técnica metaheurística utilizada por el primer grupo de exploración abordada para dar solución al problema de inversión sísmica se conoce como Algoritmos Genéticos (GA). Aspectos como parámetros, analogía con la naturaleza, ventajas y desventajas, recomendaciones al momento de implementar, y la aplicabilidad que tiene en problemas de optimización, se describen en esta sesión. Adicionalmente se expone una discusión acerca de su implementación en el contexto de la problemática planteada al inicio del seminario de investigación.

2. Algoritmos Genéticos

Es una técnica metaheurística de optimización basada en procesos de adaptación y evolución de los seres vivos en relación con el entorno donde se encuentran. Utiliza un conjunto de individuos representados mediante un esquema de codificación. Este esquema permite agrupar una lista de características de cada individuo, en una cadena que representa mediante parámetros estas características. Los individuos son seleccionados de acuerdo a su mayor nivel de aptitud mediante el uso de una función de adaptación. Esta función determina qué modelos presentan las mejores características según el requerimiento del problema bajo consideración. Una vez se han escogido los pares de mejores individuos se aplican las operaciones que facilitan el surgimiento de nuevos individuos cuyos valores de aptitud, sean mejores que los obtenidos por sus antecesores. Aquellos individuos cuyos valores de aptitud

se consideren bajos, son descartados definitivamente, de acuerdo con lo consultado en [1].

2.1. Nomenclatura

Los seres vivos se conforman por unidades denominadas células dentro de las cuales se encuentra una unidad denominada cromosoma que contiene información sobre las características de cada individuo. La información contenida dentro de un cromosoma define el grado de aptitud de los seres vivos en el entorno en que se encuentran. Una función de adaptación, semejante a las condiciones particulares del entorno evalúa a cada individuo. Si el grado de aptitud de un individuo o ser vivo es favorable al entorno, el individuo es elegido para someterse a procesos de cruce con otros de aptitud semejante para conservar las mejores características dentro de la población. Si por el contrario, la aptitud de un individuo no favorece al entorno, se descarta [1].

2.2. Analogía Metaheurística

Inicialmente se define una población conformada por individuos (posibles soluciones al problema propuesto). Cada individuo se identifica mediante un esquema de codificación binaria denominado cromosoma. Este cromosoma define un conjunto de características específicas para cada individuo. La función objetivo o problema que se pretende tratar actúa como el entorno bajo el cual se hallan sometidos los individuos a una función de adaptación. Individuos con niveles de aptitud favorable son sometidos a cruce y mutación con el objeto de mantener niveles de aptitud acordes con la respuesta buscada. Individuos con niveles de aptitud desfavorable son eliminados para dar paso a los mejor adaptados [1].

2.3. Implementación

Para la implementación de un algoritmo genético, se requiere de 4 pasos [1].

- Codificación.
- Selección.
- Cruce.
- Mutación.

2.3.1. Codificación

Uno de los procesos que se implementa dentro de un algoritmo genético, consiste en la discretización de parámetros aplicando un esquema de codificación binaria lo cual genera como resultado, una cadena de caracteres denominada cromosoma

dentro de la cual cada bit representa un gen que puede tomar valores de 0 ó 1, denominado alelo. Es importante aclarar que normalmente se utiliza un solo cromosoma, sin embargo es posible implementar codificaciones de orden superior o implementar múltiples cromosomas, los cuales se aplican en caso de problemas de orden complejo .

2.3.2. Selección

Una vez que se determina la aptitud de cada individuo modelo de la población, se emplea el proceso genético de selección; se seleccionan pares de individuos modelo para su reproducción en función de sus valores de aptitud. Los modelos con mayor valores de aptitud tienen más probabilidades de ser seleccionados que los modelos con valores de aptitud menor, este es el análogo a la herencia ya que los individuos más aptos serán seleccionados para la reproducción con mayor frecuencia.

La probabilidad de selección puede estar directamente relacionada con el valor de la aptitud o conveniencia a partir de la evaluación de los individuos en la función objetivo o alguna función de aptitud preestablecida según [2]. También es posible eliminar de manera arbitraria algunos modelos que estén por debajo o por encima de los umbrales de aptitud especificados.

2.3.3. Métodos de Selección

Existen al menos tres métodos de selección que utilizan los GA, denominados selección de aptitud proporcional, selección mediante rango y selección mediante torneo.

Selección Proporcional : Se define como el cociente entre la evaluación de cada individuo en la función de adaptación y la suma de las evaluaciones de cada uno de los individuos que conforman la población. [1].

Selección mediante rango : En el método de selección de rango según [1] y [2], los valores de adaptabilidad de los individuos se evalúan y luego se ordenan de modo que a cada individuo se le asigna un rango. Que puede ir desde 0 para el caso del mejor individuo hasta n-1 en caso del peor individuo. La selección se realiza de tal forma que el mejor modelo en la población contribuye con un número entero predeterminado de copias que el peor individuo recibe.

Selección mediante Torneo : Según se encuentra en [1] .*Este método simula la competencia que existe en la naturaleza entre los individuos por el derecho a aparearse. En la versión más simple del método, los pares son seleccionados al azar de una población de n individuos y se calculan sus valores de adaptación; estos resultados de adaptación se comparan y uno de los dos individuos modelo se acepta en base a una probabilidad P_s , llamado el número de selección torneo que es un parámetro ajustable. Para $P_s > 0,5$, se favorecen mejores modelos.*"

Tomado de [1]. *Este procedimiento se repite hasta que hay n modelos en la población descendencia, por lo tanto, la selección de torneo es esencialmente una versión probabilística de selección de rango. En una versión más general del método, la selección puede hacerse a partir de un grupo más grande de dos como se hace en programación evolutiva (EP). En un algoritmo genético básico, si la población de un principio contenía 100 individuos modelo, cincuenta pares son seleccionados en base a sus valores de adaptación. Cada par de individuos modelo ahora producirá dos crías usando los operadores genéticos de cruce y mutación, esto dará lugar a una nueva población de individuos. Alternativamente podemos rechazar un porcentaje de los modelos que presenten menor adaptación y los modelos emparejados pueden producir una sola cría, que requiere más sucesos de apareamiento para obtener modelos aceptables".*

2.3.4. Cruce

Una vez escogidos los mejores pares de individuos se utiliza el operador genético de cruce para compartir información genética entre individuos que presentan características de aptitud comunes. Este proceso hace que el intercambio de información entre individuos con características comunes de aptitud óptima se comparta generando nuevos pares de individuos. Estos nuevos individuos mejoran la obtención de una mejor solución al problema o proceso bajo análisis. [1].

2.3.5. Métodos de Cruce

Tal y como lo menciona [1], *"Se puede realizar en dos modalidades: individuales y multi-punto. En un único punto de cruce, la posición de un bit en la cadena binaria se selecciona en azar de una distribución uniforme; Todos los bits a la derecha de este bit ahora se intercambian entre los dos modelos que generan dos nuevos modelos. En multi-punto de cruce, esta operación se lleva a cabo de forma independiente para cada modelo parámetro en la cadena, es decir que para el primer parámetro de modelo una posición de bit es seleccionada al azar y los bits a la derecha; dicho de otra manera, los bits de orden inferior son intercambiados entre los modelos emparejados. Para el segundo parámetro de modelo, otra posición de bit se selecciona al azar y los bits a la derecha son de nuevo intercambiados".*

2.3.6. Mutación

La mutación es la alteración aleatoria de un bit en la cadena que representa a un individuo. Este proceso cambia el valor de un bit dentro de la cadena con el objeto de generar características que mejoran la aptitud. Se define una tasa de mutación por medio de una probabilidad especificada por el analista. Si se tiene una probabilidad baja de mutación la capacidad de exploración del espacio solución disminuye y la convergencia del algoritmo se produce muy rápido. Una alta probabilidad de

mutación aumenta la diversidad de individuos lo que se traduce en una mayor capacidad exploratoria. En contraprestación la convergencia del algoritmo tarda mucho más tiempo [1].

2.4. Comparación con otros métodos

Para el caso de los algoritmos genéticos se sabe que es un algoritmo robusto que aplica para todo tipo de problemas, es fácil de entender y manipular y es inmune a los mínimos locales en la mayoría de casos en que se aplica.

Tomado de [3]:

"Los GA se caracterizan por explorar simultáneamente todo el espacio solución primero antes de tomar una decisión, mientras que el método del Gradiente sólo analiza un punto y una pequeña área alrededor del mismo".

"Para el caso en donde la función tiene mesetas o escalones, los GA son favorecidos gracias a su búsqueda simultánea de otros puntos, mientras que en el caso del método del gradiente, el gradiente en un escalón es cero por lo tanto no se podría saber qué dirección tomar en ese punto, para avanzar hacia el mínimo buscado".

Tomado de [4]:

"Los algoritmos genéticos a comparación de los sistemas expertos presentan la ventaja de ser más versátiles en cuanto a la aplicación en distintas áreas y el poco conocimiento requerido en el área trabajar".

"Para el caso de los sistemas expertos se observa una gran dependencia del conocimiento del área a trabajar y aun más de la dependencia de las bases de datos que contengan la información necesaria para tomar una decisión".

Una desventaja de los sistemas expertos es que en caso de querer modificar alguna variable en el programa, se requiere de volver a entrenar el sistema, caso contrario a los GA que se adaptan directamente al cambio".

Tomado de [5]:

"Los GA a comparación de otros métodos de optimización no es sensible a la no linealidad, ni tampoco al efecto de las informaciones adicionales acerca del problema a solucionar. Los GA son como una caja negra en donde su único valor de entrada es la función fitness, gracias a la poca información de las características del problema a solucionar, los GA son poco propensos a caer en mínimos locales".

3. Implementación del Método de Algoritmos Genéticos en Matlab

En primer lugar se realizó una codificación poco convencional debido a que se contó con un número suficiente de bits, en este caso de 16 bits. De esta manera se podían representar $2^{16} - 1$ números binarios en el rango de búsqueda que determinara cada función objetivo, la codificación consistió en dividir el rango entre el número de posibles representaciones binarias de tal manera que a cada número decimal determinado correspondería una cadena de bits conservando 4 cifras decimales intactas.

La implementación de algoritmos genéticos se llevó a cabo con ayuda de varias subfunciones que simplifican el entendimiento del mismo, el respectivo código se puede ver en el anexo 5 así como las funciones de apoyo que utiliza para poderse ejecutar. Cabe aclarar que el código de GA no fue creado por nosotros, el código sobre el que se empezó a trabajar fue sacado del libro Inteligencia Artificial con aplicaciones en Ingeniería [4], sin embargo fue necesario realizar una serie de modificaciones que hiciera más entendible el código y que se ajustara a las necesidades que fueran surgiendo poco a poco.

3.1. Código principal

La estructura del código empieza en primer lugar borrando las variables que se puedan tener almacenadas por defecto, luego se procede asignando las variables necesarias tales como el tamaño de la población, la tasa de cruce, la tasa de mutación, el número de bits con el que se codificó, el rango del espacio de búsqueda y la cantidad de variables a manipular, es decir cuantas dimensiones tiene la función implementada.

Listing 1: Código Principal Algoritmos Genéticos

```
1 clc, clear all, close all
2 format long
3
4 popuSize=2;      % Poblacion inicial
5 xover_rate=0.7; % Tasa de cruzamiento
6 mutate_rate=0.1; % Tasa de mutacion
7 bit_n=16;       % Numero de bits para cada variable
8
9 Xmin=0;         % Limite inferior en X
10 Xmax=2*pi;     % Limite superior en X
11 Ymin=0;        % Limite inferior en Y
12 Ymax=2*pi;     % Limite superior en Y
13 var_n=2;       % Numero de variables de entrada
14 range=[Xmin,Xmax;Ymin,Ymax]; % Rango de la entradas
```

Después de esto se realiza la gráfica de contorno de la función objetivo a minimizar y se crea la población inicial dependiendo del número de bits asignado

anteriormente, esta población es creada en formato binario directamente.

Listing 2: Funcion objetivo y Poblacion inicial

```
1 %Grafica de la funcion
2 syms x;
3 syms y;
4 %Ecuacion de la funcion objetivo
5 z=10*sin(3*sqrt(x.^2+y.^2)).*exp(-sqrt((x+1).^2+(y-1).^2));
6 ezcontour(z,[Xmin,Xmax,Ymin,Ymax]);
7 xlabel('eje x');ylabel('eje y');title('eje z');
8
9 % Creacion de la poblacion inicial
10 popu=rand(popuSize,bit_n*var_n)>0.5;
```

Seguido de esto se inicia un ciclo de repetición en el cual como primer paso se evalúa la población en la función objetivo, se llena un vector de mejores posiciones que inicialmente es cero y luego se revisa el criterio por el cual el algoritmo podría detenerse. En esta primer versión del código, este criterio consiste en mirar si el mejor individuo no mejora después de 20 iteraciones se detiene. Finalmente se hace un llamado a la función encargada de crear la nueva población mediante las estrategias de selección, cruce y mutación.

Listing 3: Lazo Principal

```
1 upper=zeros(100,1); % Creacion de la matriz de mejor individuo
2 % Loop Principal
3 i=0;
4
5 while(limit<=20)
6     i=i+1;
7     k=i;
8     % Reseteo de variables
9     delete(findobj(0,'tag','member'));
10    delete(findobj(0,'tag','count'));
11    % Evaluacion de la funcion para cada miembro de la poblacion
12    fcn_value=evalpopu(popu,bit_n,range,obj_fcn);
13    if (i==1)
14        fprintf('Poblacion inicial\n');
15        for j=1:popuSize
16            fprintf('f( %f, %f)= %f\n',...
17                bit2num(popu(j,1:bit_n),range(1,:)),...
18                bit2num(popu(j,bit_n+1:2*bit_n),range(2,:)),...
19                fcn_value(j));
20        end
21    end
22
23    % Llenar la matriz con los mejores individuos
24    upper(i)=min(fcn_value);
25
26    % Registro de si el individuo mas apto mejora o no
```

```

27     if (i>=2)
28         if(upper(i)==upper(i-1))
29             limit=limit+1;
30         else
31             limit=0;
32         end
33     end
34
35     % Creacion de la siguiente poblacion con seleccion, cruce y mutacion
36     popu=nextpopu(popu, fcn_value, xover_rate, mutata_rate, k);
37
38 end

```

3.1.1. Función Next popu

Esta es una función encargada de crear la nueva población en cada iteración del código principal. En primer lugar se escoge a los dos mejores individuos de la primer población, los cuales serán conservados después de hacer el cruce y mutación en las distintas generaciones de acuerdo a las tasas definidas por el usuario inicialmente. Cabe resaltar que la técnica que el algoritmo está usando para la creación de una nueva población corresponde a la técnica de elitismo donde solo dos individuos son los que se van a conservar en cada generación.

Para hacer el proceso de cruce de individuos, en primer lugar se deben definir cuales van a ser los padres de esta generación, es decir los dos mejores individuos seleccionados anteriormente. Este proceso de cruce solo se aplica a la mitad de la población, la otra mitad estará expuesta al proceso de mutación o conservación de individuos.

En el cruce se define un punto de corte aleatorio el cual definirá en que punto se rompe la cadena de bits de cada individuo y a partir de esto hacer el cruce, la nueva población se crea a partir de la unión de los dos padres de la siguiente manera: Para un nuevo individuo la cadena de bits estará formada por parte de la cadena de bits del padre del inicio hasta el punto de corte, y la cadena de bits de la madre del punto de corte hasta el punto final; para el siguiente individuo el proceso es inverso. Este procedimiento se realiza para los demás individuos recordando que el punto de cruce es aleatorio para cada individuo. La tasa de cruce determina en cada iteración la probabilidad de que el cruce ocurra o no.

En cuanto a la mutación también se aplica a la mitad de la población y su tasa asignada al inicio del algoritmo determinará en que medida se alteran los bits de la población correspondientes a cada una de las cadenas binarias de los individuos.

3.1.2. Función Bit2num

Esta función es la encargada de convertir números binarios en decimales, la utilidad de esta función se da en el momento en que se requiere convertir la cadena

binaria de cada individuo para poder evaluarla en la función objetivo. La razón por la cual se encuentra como una función aparte del código principal, se da porque su uso se requiere con frecuencia y es mejor llamarla que no escribirla varias veces en un mismo código de fuente.

3.1.3. Función `evaleach` – `evalpopu` - `GP_function2`

Estas funciones simplemente se encargan de organizar correctamente la cadena de binaria de cada individuo, de evaluar cada individuo en la función objetivo y de entregar un valor de salida conforme a la evaluación que será leído en el código principal para la interpretación de su resultado.

Referencias

- [1] STOFFA. Paul MRINAL. Sen. *Global Optimization Methods in Geophysical Inversion*. Elsevier, 1995.
- [2] <http://www.iol.etsii.upm.es/arch/metaheuristicas.pdf>.
- [3] <http://infofich.unl.edu.ar/upload/ae1aaa6027cde52154b4925609e808a44e42d8ea.pdf>.
- [4] <http://nando1-utb.blogspot.com/p/algoritmos-geneticos.html>.
- [5] <http://www.fing.edu.uy/~sergion/Tesis.pdf>.

Memoria8

Uso de técnicas de Optimización Global para resolver problemas de Inversión Sísmica

Edgar Téllez Cortés, Johao Coronel Silva, Juan Arias Pinto

Universidad Industrial de Santander
Escuela de Ingenierías Eléctrica, Electrónica y Telecomunicaciones

Julio 15 de 2014

Sesión 8: Implementación de Algoritmos Genéticos sobre funciones multivariable

1. Introducción

En esta sesión del seminario de investigación se presentó la implementación de algoritmos genéticos para 8 funciones de prueba, variando la tasa de cruce, la tasa de mutación y el número de individuos. La metodología empleada fue la misma que se realizó para las pruebas del algoritmo PSO, donde se decidió variar un solo parámetro a la vez mientras los demás permanecían iguales con valores estándar y recomendados; estos valores corresponden a una tasa de cruce de 1 ($Tc = 1$), una tasa de mutación de 0,01 ($Tm = 0,01$), un número de individuos de 20 ($ni = 20$) y un número de bits para la codificación de 16 ($Nbits = 16$). Cabe aclarar que a pesar de nombrar con anterioridad que el conjunto de funciones de prueba era de 13 funciones, se consideró únicamente implementar el algoritmo GA en solo 8 funciones dado que hay algunas que se pueden obviar debido a la similitud con otras funciones.

2. Implementación del Método de Algoritmos Genéticos sobre Funciones de Prueba

Para entender de manera más detallada la importancia de implementar el algoritmo GA en un conjunto de funciones de prueba, es necesario remitirse a la sesión No 4 donde se explica con más detalle la razón de la implementación y las características y retos que ofrecen las distintas funciones de prueba. Para observar con

más detalle los resultados de la implementación, se pueden encontrar los datos por medio de tablas ubicadas en la carpeta Tablas - sesión 8.

La variación de los parámetros se hizo de la siguiente manera: La tasa de cruce se varió en el rango de 0,1 a 1, de igual manera para la tasa de mutación. En cuanto a la cantidad de individuos en la población, se hicieron pruebas para los valores de 10, 20 y 30 individuos respectivamente.

2.1. Función Básica

Se evidenció que el aumento de la población genera mejores resultados en cuanto a la llegada de los individuos al mínimo global, la mejor tasa de cruce encontrada fue de 0,7 y la mejor tasa de mutación fue de 0,1.

2.2. Función Esfera

Este tipo de función también tiene la tendencia marcada de que el aumento de la población entrega mejores resultados, sin embargo para este caso los mejores se obtuvieron con 20 individuos. La mejor tasa de cruce fue de 1 y la mejor tasa de mutación fue de 0,5.

2.3. Función Rastrigin

Para esta función también los mejores resultados se obtuvieron con un número alto de individuos (para este caso de 30). La mejor tasa de cruce fue de 0,5 y la mejor tasa de mutación de 0,1.

2.4. Función Schwefel

Del mismo modo que las demás funciones prueba el aumento de la población genera mejores resultados. La mejor tasa de cruce es de 0,7 y la mejor tasa de mutación es de 0,1.

2.5. Función Beales

El aumento de la cantidad de individuos presenta un leve mejora en los resultados obtenidos, sin embargo en este tipo de función donde se evidencia una gran planicie, se ve la necesidad de manejar un tipo de tasa de mutación más alta de modo que los individuos exploren globalmente la zona. Los mejores resultados se obtuvieron para una cantidad de individuos de 30, tasa de cruce de 0,7 y tasa de mutación de 0,5.

Una tasa de mutación alta en algunos casos puede significar una mejora en el dinamismo que se le da a los individuos generación tras generación, sin embargo no es aconsejable asignarle toda la probabilidad a esta tasa puesto que el resultado terminaría en la no convergencia en ningún punto.

2.6. Función Egg Holder

Este tipo de función es una función compleja puesto que no contiene un solo mínimo o mínimos sino que presenta franjas con mayor profundidad que otras, por esta razón lo que se quiso observar es la manera en como el algoritmo va reconociendo estas franjas y al final decide cual es la que posee el mínimo global.

Con el aumento de la población se evidencia un mejora en los resultados obtenidos, lógicamente a causa de que se logra poseer más información del espacio solución. Además de esto se evidencia que la disminución de la tasa de cruce conlleva un mayor número de iteraciones, caso contrario en si se tomara una alta tasa de cruce. Por otra parte, una alta tasa de mutación permite que las iteraciones disminuyan y a la vez los resultados son óptimos. Los mejores valores para los parámetros correspondieron a: cantidad de individuos de 20, tasa de cruce de 0,7 y tasa de mutación de 0,7.

2.7. Función Ackley

Para este tipo de función el aumento en el número de individuos favorece la disminución del número de iteraciones, de igual manera ocurre en los casos donde se obtiene una tasa de mutación alta. La razón por la cual ocurre esto se debe a que con una tasa de mutación alta los individuos se mueven con mayor rapidez y dinamismo logrando que algunos de ellos queden en el mínimo o muy cerca a él. El conjunto de valores que presentó mejores resultados corresponden a una cantidad de individuos de 20, tasa de cruce de 1 y tasa de mutación de 0,5.

2.8. Función Holder Table

Para este tipo de función se presencian muy buenos resultados al igual que el buen comportamiento de toda la población puesto que permanece un poco más unida a comparación de otras funciones. Cabe resaltar que en este tipo de función, para todos los casos se encuentra el mínimo global.

A partir de las pruebas realizadas se puede observar que al aumentar la población, el número de iteraciones va disminuyendo poco a poco. Por otra parte, la disminución de la tasa de cruce causa un aumento en el número de iteraciones, caso contrario a la tasa de mutación donde el aumento de esta disminuye el número de iteraciones; sin embargo para todos los casos en los que se corrió el algoritmo, se obtuvieron buenos resultados.

El conjunto de parámetros que presentó mejores resultados correspondió a una cantidad de individuos de 30, una tasa de cruce de 1 y una tasa de mutación de 0.5.

3. Análisis y Discusiones

Durante el espacio que se dio en la sesión para hablar acerca del avance presentado en la semana con respecto a la segunda técnica metaheurística, se destacan los aportes relacionados con la cantidad de individuos usados en las pruebas y la necesidad de almacenar cada una de las generaciones no solo para este algoritmo sino también para PSO.

Con respecto al tema de la cantidad de individuos, se aclaró la idea de que como el algoritmo está programado, se descarta la idea de que la población fuese creciente; es decir, que en vista de que es un algoritmo bioinspirado y evolutivo, se podría implementar la forma en la que la población fuese creciente generación tras generación. El problema con este hecho apunta a que sería necesario un mayor costo computacional del que ya se tiene, por tal razón la manera en como opera el algoritmo y se decidió trabajar de aquí en adelante, fue la de mantener una población constante en las generaciones que sean necesarias para encontrar el punto de interés.

Respecto a la necesidad de almacenar de manera permanente y manipulable los resultados y datos que se obtienen en las distintas implementaciones de ambos algoritmos, se acordó buscar una manera estándar para guardar las pruebas que hasta el momento se han hecho y las que se harán; teniendo claro que esto representaría una mejor manera para la interpretación de los datos, puesto que son datos limpios y completos. Adicional a esto, el almacenamiento de dichos datos permitiría de igual manera ser de gran utilidad para posibles estudios posteriores.

4. Conclusiones

La tasa de mutación tiene una gran relación con respecto al tipo de búsqueda que se le quiere dar al algoritmo, un valor alto para dicho parámetro permite que los individuos se dispersen de mejor manera en el espacio solución garantizando una búsqueda más global.

El aumento de la población permite una mayor información del espacio solución y consigo garantiza una mejor precisión con respecto al punto encontrado. Por otra parte, una mayor cantidad de individuos permite la disminución del número de iteraciones debido a que no es necesario tener que esperar varias generaciones para obtener un individuo en particular como ocurre en poblaciones pequeñas.

El número de iteraciones representa un indicador por el cual se pueden considerar un conjunto de valores en los parámetros mejores que otros. A pesar de esto,

no siempre la búsqueda de la disminución de estas iteraciones presenta resultados óptimos, como ocurre en el caso donde se sabe que el aumento de la tasa de mutación disminuye directamente el número de iteraciones, puesto que una elección de este parámetro (Tm) muy alto, podría causar que los individuos se vuelvan tan dinámicos que nunca convergan.

Con base en los resultados obtenidos en las distintas pruebas, se considera la idea de que no siempre se debe contar con una tasa de mutación baja; teniendo en cuenta que este fenómeno muta los individuos que progresivamente van evolucionando, puesto que una tasa de mutación relativamente alta permite una búsqueda más general que en ocasiones es necesaria para los casos donde los individuos que evolucionan no llegan al punto de interés.

Para todo los casos se tuvo la idea de que una tasa alta de cruce siempre es necesaria puesto que el manejar población tras población de manera evolutiva y conservando las características de los mejores individuos, permite orientar las generaciones hacia una mejor solución. A pesar de esto, se tiene la hipótesis de que para funciones con un nivel de complejidad alto es necesario que la tasa de cruce no sea tan cercana a 1.

Es necesario tener en cuenta a la hora de interpretar el comportamiento del algoritmo, que esta técnica de optimización es muy diferente a la técnica metaheurística PSO. Teniendo en cuenta esto, es erróneo esperar un comportamiento en particular en la manera como se comportan los individuos generación tras generación, puesto que no se trata de un enjambre que se mueve de acuerdo al tipo de función sino que por el contrario, es una serie de procesos de cruce y mutación donde los nuevos individuos a pesar de conservar las características de los padres, son nuevos.

Memoria9

Uso de técnicas de Optimización Global para resolver problemas de Inversión Sísmica

Edgar Téllez Cortés, Johao Coronel Silva, Juan Arias Pinto

Universidad Industrial de Santander
Escuela de Ingenierías Eléctrica, Electrónica y Telecomunicaciones

Julio 23 de 2014

Sesión 9: Implementación métrica de correlación en algoritmo de optimización global PSO aplicado en inversión sísmica

1. Introducción

El desarrollo de la sesión No 9, retoma el avance que se tuvo para el algoritmo PSO. Por tal razón se centra en el uso de una figura de mérito que permita comparar las trazas sintéticas y las de referencia.

La métrica de correlación de señales fue expuesta como dicha figura de mérito para la utilización del algoritmo PSO en inversión sísmica, adicional a esto se explicó la estrategia que se llevó a cabo para poder realizar dicha comparación donde fue necesario emplear un paquete de procesamiento de datos de Matlab utilizado como puente de unión entre este software y SU ya que las trazas sísmicas obtenidas con SU no podían manipularse directamente por Matlab sin antes hacer una conversión de las mismas a formato seg-y. El paquete de herramientas utilizado para esta conversión es denominado *seismicLab*.

Durante la sesión No 9 se realizó una breve explicación acerca de la métrica utilizada, las variantes que existen y la interpretación necesaria para aplicarla al problema de inversión sísmica. Adicional a esto, se trató acerca de las modificaciones que tuvieron lugar en el algoritmo PSO, la función de inversión automática y los nuevos criterios implementados correspondientes a los resultados de las métricas.

2. Correlación

Es una medida de comparación muy usada en probabilidad y estadística, donde se caracteriza la relación que tiene una variable con otra. Esta operación puede ser representada mediante una línea de ajuste que indique el grado de similitud entre variables. La línea de ajuste consta de componentes como fuerza, sentido y forma, donde se clasifica la relación de fase, la relación proporcional o inversa y el ajuste lineal o curvo entre variables [1]. El tipo de comparación que realiza esta medida se hace componente a componente (punto a punto), donde el grado de similitud de las señales comparadas, se indica por medio un coeficiente de correlación (ρ).

2.1. Coeficientes de Correlación

Existe variedad en los coeficientes de correlación que se ajustan de mejor manera de acuerdo a la naturaleza de los datos.

- Coeficiente de correlación de Pearson.
- Coeficiente de correlación de Spearman.
- Correlación canónica.
- Correlación múltiple.

2.1.1. Coeficiente de correlación de Pearson

La expresión matemática para determinar el coeficiente de correlación de Pearson, consta de los siguientes términos [2].

$$\rho_{(X,Y)} = \frac{\sigma_{(X,Y)}}{\sigma_X * \sigma_Y} = \frac{E[(X - \mu_X) * (Y - \mu_Y)]}{\sigma_X * \sigma_Y} \quad (1)$$

Donde : σ_{xy} = Covarianza de (X,Y) σ_x = Desviación típica de (X) σ_y = Desviación típica de (Y)

Interpretación según [1] y [2]:

$\rho = 1$; indica una correlación positiva perfecta, con relación lineal y proporcional.

$0 < \rho < 1$; existe correlación positiva.

$\rho = 0$; no existe relación lineal, pero podría existir una relación curva.

$-1 < \rho < 0$; existe correlación negativa.

$\rho = -1$; indica una correlación negativa perfecta, con relación inversa.

2.1.2. Coeficiente de correlación de Spearman

El cálculo de los coeficientes de la correlación de Spearman, se basa en el desarrollo de dos expresiones matemáticas [3].

$$\rho = 1 - \frac{6\Sigma D^2}{N(N^2 - 1)} \quad (2)$$

$$t = \frac{\rho}{\sqrt{(1 - \rho^2)(N - 2)}} \quad (3)$$

Donde :

D es la diferencia es la diferencia entre X y Y.

N es el número de parejas.

Para observaciones mayores a 20 se usa la ecuación 3, con ayuda de la distribución de probabilidad t student. Este tipo de correlación es menos sensible en comparación con la correlación de Pearson, para valores muy lejanos la interpretación es igual.

2.2. Análisis de correlación de Canónica

- Es un método de análisis multivariable [4].
- Busca las relaciones que pueda haber entre dos grupos de variables y la validez de las mismas.
- Predice múltiples variables dependientes a partir de múltiples independientes.
- La manera de interpretarlas se basa en la observación de la varianza de la variable con respecto al valor teórico.

2.3. Análisis de correlación múltiple

Predice una variable dependiente a partir de múltiples variables independientes, según lo explicado en [4].

3. Datos Seg-y

El formato de archivo SEG-Y es desarrollado y mantenido por la Sociedad de geofísicos de exploración para el almacenamiento de datos sísmicos. SEG-Y es un formato abierto y se utiliza sobre todo para almacenar una sola línea de datos digitales sísmicos en cintas magnéticas [5].

4. Algunos comandos de Seismic Unix [6], [7]

maxdiff :Determina la máxima diferencia absoluta entre dos datos seg-y.

segymod : Remplaza el encabezado de un archivo seg-y.

suaddnoise : Agrega ruido al conjunto de trazas.

sucmp : Compara un conjunto de dos trazas sísmicas y retorna un valor cero si son iguales ó un valor de 1 si son diferentes.

sudiff, susum, suprod, suquo : Realiza la diferencia, suma, producto y cociente de dos trazas en SU, utilizando la palabra suop.

suacor : Establece la autocorrelación existente entre la traza sísmica.

suop2 : Realiza una operación entre trazas en binario.

suresamp : Realiza nuevamente muestreo en el dominio del tiempo.

suxcor : Establece la correlación entre una traza y un filtro proporcionado por el usuario.

5. Modificación de la Función Inversión Automática

La razón por la cual fue necesario modificar la función inversión automática, se hace de acuerdo a que de ahora en adelante es indispensable poder importar las trazas sísmicas de SU a Matlab para poder manipular su contenido. Los ajustes hechos a esta función fueron pocos, se agregaron ciertas líneas de código necesarias para cambiar el formato de las trazas de SU a seg-y, luego otras líneas encargadas de importar estas trazas a Matlab en forma de estructura. La razón por la cual es ventajoso contar con esta estructura, es el hecho en que los campos de la misma poseen toda la información correspondiente a las trazas. Finalmente se programó dentro de esta misma función el cálculo del coeficiente de correlación entre la traza actual y la proveniente del modelo de referencia.

Es importante resaltar que la función inversión automática realiza el cálculo del coeficiente de correlación de cada una de las 60 trazas que conforman el sismograma proveniente de la adquisición sísmica, entregando como salida no un valor como ocurría en la sesión anterior, sino un vector de coeficientes de correlación que será manipulado por el código principal del algoritmo enjambre de partículas.

A continuación un breve resumen de la estructura manejada por la función inversión automática:

- Se actualiza el modelo sintético para cada partícula de acuerdo a los *sloths* iniciales por cada una de ellas.
- Se genera el modelo con las capas actualizadas.
- Se simula la adquisición para obtener las trazas sintéticas.
- Conversión del formato .SU a .SEGY y viceversa.
- Se importa la traza sintética y real a Matlab en forma de estructura.
- Se calculan los coeficientes de correlación entre las trazas sintéticas y reales.

6. Modificación del Algoritmo PSO en Matlab

El principal cambio realizado en la implementación del algoritmo PSO giró en torno a la manipulación de los coeficientes de correlación entregados por la función inversión automática. En base a esto, se decidió crear dos criterios que permitieran manipular de la mejor manera posible dichos coeficientes de correlación y poder elegir correctamente las partículas pertenecientes a los mejores resultados.

6.1. Criterio No 1

Teniendo en cuenta que cada partícula entrega un vector de 60 coeficientes de correlación por las razones antes mencionadas y que el valor ideal del coeficiente de correlación es de 1, se procedió a hacer la resta entre un vector de coeficientes ideales (unos), del mismo tamaño del arrojado por la función, y el obtenido para cada partícula. Luego de esto, se calcula la magnitud del vector resultante para finalmente compararla con los vectores resultantes de las demás partículas de modo que se eligiera la partícula que presentó el vector con los coeficientes de correlación más cercanos al ideal. Garantizando esto, se cumple con el requerimiento del algoritmo respecto a la minimización entorno a la semejanza presentada entre las trazas sintéticas y las de referencia.

A continuación se presenta de manera más estructurada, la secuencia programada para el criterio No 1.

- En vista de que se van a analizar los coeficientes de correlación que están en el rango de -1 a 1.
- Se tiene un vector columna con los coeficientes ideales de correlación (vector columna de unos).
- Se tienen vectores columna con los coeficientes de correlación entre las trazas de cada partícula y las trazas del modelo real.
- Se hace la diferencia entre el vector de coeficientes ideales con el vector de coeficientes actuales de cada partícula en cada iteración.
- Se hace la suma componente a componente de cada vector resultante.
- Se elige el vector con la menor diferencia que corresponderá a la partícula que generó la traza más aproximada a la traza real.

6.2. Criterio No 2

El criterio No 2 se creó con el objetivo de definir de la mejor manera posible, si una partícula era mejor a la de la iteración anterior Teniendo en cuenta su respectivo vector de coeficientes de correlación. En base a esto se decidió comparar los vectores de ambas partículas (anterior y actual), de modo que se hiciera un listado de los coeficientes iguales, menores y mayores en dicha comparación. A partir de este

listado se tomaba la decisión de si la partícula era mejor o no, para ello se procedía a hacer un conteo de los coeficientes mayores y los menores, y en caso de que el listado diera un mayor número de coeficientes mayores a los menores, se escogía esa partícula como óptimo local.

La estructura un poco más simplificada para el criterio antes descrito, se presenta de la siguiente manera.

- En vista de que se quiere determinar si la partícula actual genera una traza más aproximada a la real en relación con la partícula anterior.
- Se comparan dos vectores de coeficientes de correlación correspondientes a una partícula antes de después de la actualización.
- Primero se hace una lista de los coeficientes mayores, menores e iguales entre los dos los vectores (el vector actual con respecto al anterior).
- Si la lista de coeficientes mayores es superior a la de coeficientes menores, se actualiza la mejor posición local.
- Si lista de coeficientes mayores es inferior a la de coeficientes menores, se deja la mejor posición local como estaba.

7. Estructura del Algoritmo de Enjambre de Partículas

La estructura del Algoritmo basado en Enjambre de Partículas, se ha modificado para facilitar la incorporación de nuevos criterios.

- Se inicializan los parámetros (Fronteras, No. de partículas, No. de iteraciones, constantes, radio de avance).
- Se crean los *Sloths* iniciales con una distribución uniforme.
- Se utiliza la función `inversion_automatca.m` con los *sloths* iniciales, para obtener el vector de coeficientes de correlación.
- Se elige el mínimo local.
- Se elige la partícula con los mejores coeficientes de correlación. (Criterio N°1)
- Se elige el mínimo global.
- Ciclo de iteraciones Ciclo de partículas.
Actualización de la posición y la velocidad.
Criterio de restitución de partículas.
Uso de la función `inversion_automatca.m`.
Actualización local de la partícula(Criterio N°2).

- Finalización ciclo.
- Actualización global (Criterio N°1).
- Finalización del Algoritmo.

8. Análisis y Discusión

Durante el desarrollo del espacio de discusión en la sesión No 9, se habló acerca de los criterios implementados en el algoritmo PSO. Se dijo que a pesar de tener una justificación y estructura valida, no correspondan a medidas reconocidas matemáticamente; por tal razón se recomendó que estos criterios fueran nuevamente planteados y lograran obtener un criterio de medida ya existente. A pesar de esto se rescata la idea de realizar la diferencia entre vectores de coeficientes de correlación ideales y los actuales, puesto que la operación matemática correspondiente a este ejercicio, es conocida como cálculo del error.

Referencias

- [1] Wikipedia. Correlación. <http://es.wikipedia.org/wiki/Correlacion>.
- [2] Wikipedia. Coeficiente de correlación de pearson. http://es.wikipedia.org/wiki/Coeficiente_de_correlacion_de_Pearson.
- [3] Wikipedia. Coeficiente de correlación de spearman. http://es.wikipedia.org/wiki/Coeficiente_de_correlacion_de_Spearman.
- [4] Wikipedia. Correlación canónica. http://es.wikipedia.org/wiki/Correlacion_canonica.
- [5] INTERGEO. Seg-y. <http://inter-geo.org/Study/Seismic/Register-data/SEG-Y.php?lang=es>.
- [6] Wikipedia. Seismic unix. http://en.wikipedia.org/wiki/Seismic_Unix.
- [7] David FOREL, BENZ Thomas, and PENNIGTON Wayne. *Seismic Data Processing with Seismic Unix*. Lawrence Gochioco, Series Editor, 2005.

Memoria10

Uso de técnicas de Optimización Global para resolver problemas de Inversión Sísmica

Edgar Téllez Cortés, Johao Coronel Silva, Juan Arias Pinto

Universidad Industrial de Santander
Escuela de Ingenierías Eléctrica, Electrónica y Telecomunicaciones

Julio 29 de 2014

Sesión 10: Implementación de Algoritmos Genéticos para la obtención de un modelo geológico

1. Introducción

En el desarrollo de la sesión 10 se explicó la manera como se implementó algoritmos genéticos al problema de inversión sísmica empleando la métrica de correlación de trazas sísmicas. De igual manera, se volvió a explicar con más detalle el código del algoritmo y las funciones que se necesitan para su funcionamiento. Adicional a esto se habló acerca de los inconvenientes que se tuvieron con respecto a la implementación generando un espacio de discusión propicio para hablar acerca de los errores cometidos y las soluciones correspondientes a dichos errores.

2. Estructura del Método de Algoritmos Genéticos

Para la implementación de las métricas de comparación entre modelos geológicos, se realizaron modificaciones al método de Algoritmos Genéticos.

- Se inicializan las variables como son (No. de individuos, tasa de cruce, tasa de mutación, No. de bits, dimensiones, fronteras y No. de iteraciones). Cabe aclarar que actualmente el algoritmo se está parando por número de iteraciones.
- Se crea la población inicial de manera aleatoria.

- bucle iterativo Generaciones
 - Reinicio de variables
 - Evaluación de cada individuo en la función objetivo
 - Se registra si el individuo mejora o no
 - Se crea la siguiente población

- Fin bucle iterativo.

2.1. Funciones Creadas para el Método de Algoritmos Genéticos

2.1.1. Bit2num

Se encarga de convertir una cadena de bits a su correspondiente número en decimal. Es utilizada para poder evaluar la función objetivo de acuerdo a las coordenadas de cada individuo que inicialmente están descritas como una cadena binaria.

2.1.2. Num2bit

Se encarga de convertir un número a cadena de bits. se utiliza para ingresar unas coordenadas específicas a la población inicial, esto con el fin de probar el algoritmo ingresándole la solución a la población inicial.

2.1.3. Nextpopu

Genera la nueva población. Utiliza la técnica de ELITÍSMO donde se encogen los dos mejores individuos. El cruce se realiza únicamente con estos dos individuos, teniendo en cuenta la tasa de cruce especificada.

- El punto de cruce es aleatorio.
- La mutación se genera también de manera aleatoria pero teniendo en cuenta la tasa de mutación especificada.

2.1.4. inversion_automática

Se encarga de generar el modelo con las nuevas coordenadas del individuo, simula la adquisición del modelo sintético y genera las trazas sintéticas correspondientes, realiza la correlación entre las trazas sintéticas y las trazas reales, entrega los coeficientes de correlación por cada traza y para cada individuo.

3. Modificaciones Implementadas sobre Algoritmos Genéticos

Las modificaciones realizadas a la implementación de algoritmos genéticos en Matlab es muy similar a la realizada para el algoritmo PSO en la aplicación en inversión sísmica. En primer lugar el número de dimensiones aumenta, ahora es considerado como 4, y cada una de las dimensiones fueron el valor de los 4 *sloths* que dan como resultado un modelo diferente en cada combinación.

En el código principal la población inicial de individuos se creaba en formato binario y se requería una función encargada de transformar los valores de *sloths* de binario a decimal. Sin embargo, para esta implementación con la idea de guiar un poco la población de individuos al asignar valores de *sloths* cercanos a la solución esperada, fue necesario programar una función que convirtiera los números de decimal a binario puesto que la población se hacia en formato binario y para nuestra prueba, dicha población debía ser acorde con valores en decimal conocidos y cercanos a los reales. Cabe recordar que la población se crea en formato binario con el fin de realizar lo procesos de cruce y mutación conforme la genética lo indica.

La función inversión automática no requirió ningún cambio y la implementación de la métrica se hizo de la misma manera que en PSO, usando los coeficientes de correlación y a su vez guardando la trayectoria de los individuos a través de las iteraciones.

Cabe aclarar que en vista de que se discutió en la sesión pasada acerca de la manera en como se plantearon los criterios de selección de la mejor partícula y el reconocimiento de si mejora o no en cada actualización para el algoritmo PSO, se tuvo en cuenta las recomendaciones y se procedió a modificar el criterio de selección del mejor individuo para el algoritmo GA, donde básicamente se modificó la parte final conociendola como el cálculo del error medio cuadrático. A continuación se presenta un breve estructura que explica el planteamiento del criterio utilizado.

3.1. Criterio de selección del mejor individuo

En vista de que se van a analizar los coeficientes de correlación que están en el rango de -1 a 1, se tiene un vector columna con los coeficientes ideales de correlación (vector columna de unos), se tienen vectores columna con los coeficientes de correlación entre las trazas de cada individuo y las trazas del modelo real, se hace la diferencia entre el vector de coeficientes ideales con el valor absoluto del vector de coeficientes actuales de cada individuo en cada iteración, se hace la suma componente a componente elevada al cuadrado de cada vector resultante dividiendola en el número de elementos del vector y finalmente se le saca la raíz. Calculado el error cuadrático medio, se elige el vector con el menor error cuadrático que corresponderá al individuo que generó la traza más aproximada a la traza real.

4. Análisis de Resultados

En cuanto a la implementación de GA se puede decir que no se recogieron muchos resultados ya que el proceso de generación del modelo de capas del subsuelo y la adquisición sísmica es demorado y requiere de un alto costo computacional; es por eso que se hicieron pruebas con un bajo número de individuos con lo cual se tomó el pantallazo del modelo resultante y del mínimo global encontrado por el algoritmo, estos resultados se pueden observar en la sección 5.8.1 del capítulo apéndices.

En estas pruebas preliminares se puede concluir que una tasa de cruce que arrojó resultados aceptables en esta primera implementación fue de 1, en la cual se encontraron de manera aproximada el valor del *sloth* de la primera y segunda capa con una población muy pequeña (4 individuos) en 50 iteraciones.

5. Discusión

Se presentaron discusiones y aportes por parte de los profesores director y co-director del proyecto acerca de la posible pérdida de información de los *sloths* del modelo de velocidades, debido al método de conversión de decimal a binario y viceversa. Sin embargo se hicieron pequeñas pruebas en el salón de exposiciones donde quedó claro que al pasar un número cualquiera con el método implementado quedaban intactos 4 decimales de los números sometidos a prueba. También se sugirió la utilización de la función *corr* de Matlab en vez de *corrcoef* ya que la primera solo calcula el coeficiente de correlación mientras que la otra calcula demás valores innecesarios para la aplicación requerida. Otro aspecto sobre el cual también se habló corresponde a la comparación tan estricta que hace la correlación, esto en vista de que al ingresar valores de *sloths* muy cercanos a los del modelo real, los coeficientes de correlación correspondientes a la comparación, evidenciaban que eran totalmente distintos. Teniendo en cuenta lo anterior, se decidió dar vía libre a la implementación de una métrica que ojala pudiera comparar estas trazas de una manera más detallada y diciente.

Memoria1

Uso de técnicas de Optimización Global para resolver problemas de Inversión Sísmica

Edgar Téllez Cortés, Johao Coronel Silva, Juan Arias Pinto

Universidad Industrial de Santander
Escuela de Ingenierías Eléctrica, Electrónica y Telecomunicaciones

Agosto 6 de 2014

Sesión 11: Implementación de la métrica correlación cruzada en algoritmos genéticos y PSO

1. Introducción

Teniendo en cuenta las recomendaciones hechas en sesiones anteriores, en esta sesión se probaron otras métricas de comparación de trazas. De igual manera, se realizó una observación de las trazas y del efecto que tienen los diferentes *sloths* en las mismas. La métrica de comparación utilizada fue la correlación cruzada debido a que esta tiene la ventaja de comparar señales y determinar qué tan parecidas son, incluso aunque una de estas este desfasada con respecto a la otra.

2. Correlación Cruzada

Es una medida de similitud que comúnmente se aplica para la búsqueda de características similares de una señal desconocida a partir de una de referencia. Es una operación muy similar a la convolución, solo que difiere en que ninguna de las señales se invierte. La correlación cruzada entrega un vector de datos correspondientes a la sumatoria del producto punto entre cada componente de ambas señales, para cada posición de la trayectoria de operación. En vista de que el indicador no es un valor sino un vector, se deben normalizar sus componentes y luego seleccionar los valores máximos correspondientes a cada trayectoria de operación, de modo que se pueda ver esta medida de similitud como los coeficientes de correlación cruzada. A diferencia de la correlación, esta medida compara las señales en todo su conjunto,

considerando los desplazamientos en tiempo de las mismas [1], [2].

Es posible obtener las normalizaciones del vector de correlación cruzada conocidas como.

2.1. Normalización Biased (parcial)

$$R_{xy,baised}(m) = \frac{1}{N} R_{xy}(m) \quad (1)$$

2.2. Normalización Unbiased (imparcial)

$$R_{xy,unbiased}(m) = \frac{1}{N - |m|} R_{xy}(m) \quad (2)$$

2.3. Coeficiente de Correlación Cruzada

Es posible conocer los coeficientes de correlación cruzada en un rango de -1 a 1 que se pueden interpretar del mismo modo que los coeficientes de correlación punto a punto. Lo que se hace es una normalización con la potencia de cada señal suponiendo un retardo de cero [2].

$$l_{xy}(m) = \frac{\phi_{xy}(m)}{\sqrt{[\phi_{xx}[0]\phi_{yy}[0]]}} \quad (3)$$

3. Influencia de los *Sloths* en las Trazas Sísmicas

Realizando una observación detallada de la influencia de los *sloths* del modelo de capas del subsuelo, se pudo determinar que la distancia entre un arribo y otro presente en una traza sísmica depende directamente de la velocidad de cada capa. Por otro lado, la amplitud de estos arribos depende de la diferencia de velocidad entre dos capas del modelo. Teniendo en cuenta esto, una misma amplitud de arribo podría ser generada por dos combinaciones de velocidades diferentes corroborando el concepto de la no unicidad de las trazas sísmicas, es decir que una misma traza sísmica podría ser generada por diferentes modelos de velocidades entorpeciendo el proceso de comparación de trazas sísmicas vital para el buen funcionamiento del algoritmo, en base a esto se tiene que la única diferencia entre las trazas idénticas generadas por diferentes modelos de velocidades es que estarían desplazadas unas de otras. Teniendo en cuenta lo anterior, se es consciente de que en vista que la métrica utilizada (correlación cruzada) contempla la idea de un desfase, se podría correr el riesgo de encontrar un nuevo juego de *sloths* que se ajuste al tipo de traza que se tiene como referencia; a pesar de esto, el interés por realizar dichas pruebas se da en búsqueda de resultados positivos y de corroborar que la métrica trabaja de manera correcta y que es un indicador efectivo en la comparación de trazas.

4. Implementación de la Métrica

En cuanto a la implementación de la métrica de correlación cruzada, la diferencia frente a la correlación normal es que en la comparación de dos trazas sísmicas el vector de coeficientes de correlación esta compuesto por coeficientes máximos que corresponden a la comparación entre las trazas recordando que una de ellas es dinámica al momento de compararlas. Finalmente se conforma un vector con los 60 coeficientes máximos y el procedimiento a seguir es igual al de la métrica anterior.

En este punto cabe aclarar que la métrica anterior llamada correlación, se define de mejor manera como correlación punto a punto, es decir que el vector de coeficientes de correlación que arroja la métrica corresponden a la comparación directa entre ambas trazas, estando estas estáticas y comparando su morfología punto a punto. Para el caso de la correlación cruzada, es necesario tener en cuenta que esta métrica realiza el mismo proceso de convolución con la única diferencia de que ninguna de las trazas a comparar se deben invertir, en base a esto se entiende que se van tener muchos vectores con coeficientes de correlación. Sin embargo el vector de coeficientes general estará conformado por el máximo de los coeficientes de cada vector, los cuales indican el momento exacto donde dichas trazas sísmicas se asemejaron más.

5. Discusión

Durante esta sesión se presentó el caso donde el espacio de discusión permitió llegar a conclusiones donde se dieron lugar posibles soluciones a los problemas encontrados. Dentro de las observaciones que se expusieron en la sesión acerca de las trazas sísmicas y la influencia de los *sloths*, se pudo ver que cuando se asignaban valores altos a los mismos los últimos arribos de las trazas sísmicas no se podían evidenciar en el sismograma, ocasionando que algunas de estas trazas sísmicas fueran similares a las esperadas por el algoritmo debido a que la pérdida de información de la traza sintética representaba una mayor similitud a la traza de referencia a como realmente debía ser. El problema que estaba sucediendo en estos casos era que el tiempo de grabación de los geófonos programado en el software SU, no era suficiente para captar todos los arribos, por tanto este tipo de parámetro de interés para la adquisición sísmica se debió incrementar, dando solución al problema encontrado.

Referencias

- [1] Oliver Hinton. Describing random sequences. <http://www.staff.ncl.ac.uk/oliver.hinton/eee305/Chapter6.pdf>.
- [2] Wikipedia. Correlación cruzada. http://es.wikipedia.org/wiki/Correlacion_cruzada.

Memoria12

Uso de técnicas de Optimización Global para resolver problemas de Inversión Sísmica

Edgar Téllez Cortés, Johao Coronel Silva, Juan Arias Pinto

Universidad Industrial de Santander
Escuela de Ingenierías Eléctrica, Electrónica y Telecomunicaciones

Agosto 12 de 2014

Sesión 12: Implementación de los criterios de parada y recolección de datos experimentales para el algoritmo PSO

1. Introducción

En esta sesión del seminario de investigación se implementó el algoritmo PSO aplicado a inversión sísmica basada en el modelo, añadiendo los criterios de parada empleados en las funciones prueba y un mecanismo de recolección de datos en forma de estructuras utilizando el software Matlab. Cabe recordar que el desarrollo de esta sesión se realizó teniendo en cuenta las recomendaciones hechas en sesiones pasadas, y que al punto donde se lleva la investigación y el desarrollo del seminario, eran necesarias realizarlas.

2. Implementación de PSO

Teniendo en cuenta las sugerencias acerca del almacenamiento de los resultados para su posterior análisis, se implementó un mecanismo de recolección de datos en forma de estructuras las cuales son posible manejar en Matlab. Dichas estructuras organizan la información importante de las diferentes pruebas realizadas de manera ordenada y por campos, la manera en como se decidió distribuir la información almacenada fue la siguiente:

Fecha de la prueba.
Nombre del algoritmo.
Parámetros:
Número de partículas.
c1.
c2.
Masa de inercia.
Radio de avance.
Contador de saturación.
psilon1.
psilon2.
Número de iteraciones.
Velocidades.
Coeficientes.
Modelo encontrado.
Mejor partícula.

3. Observaciones de la prueba

Además de la explicación de la creación de las estructuras de almacenamiento de datos, se implementaron los criterios de parada utilizados en las funciones prueba. Estos criterios corresponden a la poca variabilidad del error de la mejor partícula, la convergencia del enjambre y el contador por saturaciones, teniendo en cuenta que el orden en que fueron nombrados indica la prioridad sobre la cual se maneja en programación.

A continuación se presenta un recuento de la manera en como funcionan dichos criterios.

3.1. Contador de Saturación

Es el número máximo de iteraciones permitidas en caso de que el algoritmo no converja por los otros criterios de parada.

3.2. Convergencia de las Partículas

Se calcula el error medio cuadrático de las partículas antes y después de cada actualización, después se comparan los dos vectores de estos errores encontrando su máxima diferencia que debe ser menor a un *psilon2* para cumplir con este criterio.

La interpretación de dicho criterio corresponde a la manera en como se decidió garantizar que el enjambre halla convergido en algún punto, es por eso que en cada iteración se calcula la máxima diferencia entre la posición actual y la pasada, la

cual indica si alguna de las partículas esta realizando un cambio notorio o si por el contrario el cambio es muy mínimo lo cual se relaciona y se interpreta como si el enjambre convergió.

3.3. Poca Variabilidad

En este caso se observa la mejor partícula de todo el enjambre, se hace la diferencia del error medio cuadrático en la iteración actual y la anterior. Si esta diferencia es menor que la variable *psilon1* cinco veces seguidas, el algoritmo se detiene.

La interpretación correspondiente a este criterio tiene que ver con la manera en como se decidió garantizar que el mínimo global encontrado no cambiara de una iteración a otra, es por eso que el criterio garantiza que el mínimo global encontrado, se mantenga constante aun así cuando las demás partículas sigan explorando. Cumplido esto, se procede a revisar si el enjambre convergió en ese punto que tentativamente corresponde al mínimo global, de ser así, el algoritmo se detiene por criterio de parada.

4. Análisis de Resultados

En las pruebas realizadas se utilizaron 30 partículas, un valor de constantes $c1$ y $c2$ de 1,8 y 1,9 respectivamente, una masa de inercia de 0,5, un radio de avance de 0,05, un contador de saturaciones de 1000, un valor de *psilon* 1 y 2 de 0,005. Con estos valores el algoritmo presentó una buena convergencia y el enjambre en su totalidad llegó a un mismo punto del cual no salió, sin embargo después de encontrar el primer *Sloths*, el enjambre se quedó estancado perdiendo su búsqueda exploratoria; en cuanto a la trayectoria de los coeficientes de correlación cruzada mejoraron un poco en cada iteración. Los resultados de las pruebas se pueden ver en la carpeta estructuras correspondiente a esta sesión. Se hace la aclaración de que se hicieron pocas pruebas debido al costo computacional y a la gran cantidad de tiempo que empleaba la prueba, aproximadamente una hora y media.

Cabe aclarar que en vista de que el conjunto de parámetros difiere al conjunto de los mismos para las funciones de prueba, es conveniente explicar la razón por la cual está usando estos valores. En primer lugar el espacio de solución que se tiene para el problema de inversión sísmica, se encuentra en el rango de 0,1 a 1; un espacio muy reducido en cuanto a los que se tenían con las funciones de prueba. Por esta razón, se hizo necesario ajustar el radio de avance de cada partícula de modo que se garantizara la libertad de sus movimientos dentro del nuevo y reducido espacio solución, de igual manera el valor de los *epsilon* necesarios para el criterio de parada fueron ajustados conforme a la nueva resolución que se tenía en la trayectoria del enjambre.

5. Discusiones finales

Dentro de las discusiones presentadas en la sesión se llegó a que un posible criterio de aceptación es descartar un determinada combinación de valores de la partícula si los *sloths* de las capas superiores son menores a los de las capas inferiores, ya que esto es imposible en el ámbito realista teniendo en cuenta que entre más profundidad del subsuelo la velocidad de las capas incrementa. Este criterio modificaría la parte del código que hace la restitución de partículas.

También se discutió la posibilidad de que el algoritmo solo estaba encontrando el *sloth* de la primera capa debido a que los arribos provenientes por la transición de la onda entre la primer capa y la siguiente tienen más energía que los provenientes por capas inferiores, haciendo más difícil la comparación de trazas con técnicas como la correlación y la correlación cruzada donde se le da una mayor prioridad a la parte que contiene mayor energía de la señal. Por tal razón se planteó la hipótesis de que el algoritmo podría encontrar el valor de *sloths* de capas inferiores si se pudiera recortar o filtrar las trazas quitando los primeros arribos justo después de que el algoritmo encuentre el valor de una capa superior.

Otra posible hipótesis planteada fue la idea de hacer varios lanzamientos del algoritmo, en donde cada vez que este encontrara el valor de *sloth* de una capa superior, en el siguiente lanzamiento se asignara éste valor a las partículas de búsqueda, con la idea de que el algoritmo se concentrara en las siguientes capas inferiores encontrando paulatinamente el valor de las siguientes capas.

Memoria13

Uso de técnicas de Optimización Global para resolver problemas de Inversión Sísmica

Edgar Téllez Cortés, Johao Coronel Silva, Juan Arias Pinto

Universidad Industrial de Santander
Escuela de Ingenierías Eléctrica, Electrónica y Telecomunicaciones

Agosto 20 de 2014

Sesión 13: Implementación de una nueva métrica basada en la diferencia de sísmogramas para el algoritmo de optimización global PSO

1. Introducción

En esta sesión del seminario de investigación se implementó una nueva métrica de comparación de trazas sísmicas utilizando el algoritmo PSO aplicado a inversión sísmica, se utilizó un minicomputador denominado OSAKA, propiedad del grupo de investigación CPS de la UIS, para realizar pruebas que acarrear un elevado costo computacional y que demandan una mayor cantidad de tiempo. También se decidió trabajar con un modelo de capas diferente al utilizado en pruebas anteriores, este nuevo modelo cuenta con una geometría más sencilla en vista de que considera únicamente capas paralelas. La cantidad de capas se mantuvo igual, es decir de 4 capas. El modelo sobre el cual se trabajó se muestra en la figura 1. Adicional a esto se habló acerca de las modificaciones que tuvieron lugar en la estructura encargada de almacenar todos los datos de las pruebas.

2. Implementación

Los cambios pertinentes en cuanto a la implementación correspondieron a la aplicación de la nueva métrica de comparación la cual se programó como las demás dentro de la función inversión automática, ya que esta es la función encargada de

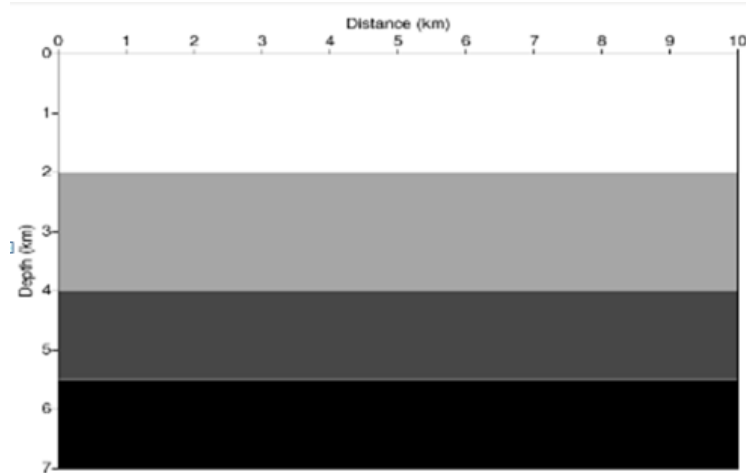


Figura 1: Primer Modelo Sísmico. Fuente: Autores

entregar al algoritmo PSO en el código principal, el resultado de la comparación de trazas de referencia y actualizada.

2.1. Métrica de Diferencia de Trazas

Hasta el momento se ha contado con dos tipos de métricas conocidas como Correlación punto a punto y Correlación cruzada, ahora se incluye una nueva métrica que no considera coeficientes de correlación sino que involucra directamente la diferencia de los vectores de las trazas que poseen todos los datos concernientes a la amplitud, tiempo de grabado y forma. Esta nueva métrica de comparación de trazas se basa en la diferencia de los vectores correspondientes a las mismas, donde se eleva al cuadrado cada una de las componentes del vector resultante y se suman para obtener un único valor. Este valor se evalúa para determinar el grado de semejanza entre las dos trazas, si es pequeño indica una elevada semejanza entre las dos señales, si es grande indica que la traza es muy distinta con respecto a la de referencia. Teniendo en cuenta que la comparación se hace entre dos sismogramas, el resultado de este métrica de comparación será un vector de números que indicarán la diferencia entre trazas, no un vector de coeficientes de correlación como hasta el momento se había trabajado.

Cabe resaltar que en vista de que el vector entregado por esta nueva métrica de comparación, es diferente a los que se venían trabajando, es necesario modificar el código principal de PSO donde no es necesario realizar la diferencia con el vector de coeficientes ideales (unos), sino que simplemente basta con calcular el error medio cuadrático de estos vectores y el que arroje un menor valor, corresponde al vector que indica que partícula generó un conjunto de *sloths* más cercanos a los del modelo de referencia.

Adicional a la presentación de los resultados obtenidos, se explicaron las modi-

ficaciones que se hicieron en la estructura de almacenamiento de datos, donde fue necesario agregar tres nuevos campos que brindarán información acerca del tiempo gastado en la ejecución de la prueba, el tipo de métrica implementada y la arquitectura computacional sobre la cual se trabajó.

3. Análisis de Resultados

Se realizó una prueba en la cual se utilizaron 30 partículas de búsqueda, constantes $c1$ y $c2$ de 1,8 y 1,9 respectivamente, masa de inercia de 0,5, radio de avance de 0,05, contador de saturaciones de 100 y $psilon$ 1 y 2 de 0,001. En este tipo de prueba se hicieron dos lanzamientos consecutivos del algoritmo, donde en la segunda ocasión se fijó la segunda partícula con el valor del mejor resultado encontrado en el primer lanzamiento. En el primero de ellos se logró encontrar el valor del *sloth* de la primera capa del subsuelo, sin embargo en el segundo lanzamiento el mejor resultado fue igual al primero puesto que este valor se mantuvo constante. Finalmente el algoritmo convergió por máximo número de iteraciones (100 iteraciones), este resultado se guardó en una estructura y se puede observar con más detalle en la carpeta estructuras correspondiente a la sesión 13.

Se puede decir que es positivo que el algoritmo siempre encuentre el valor de *sloth* de la primera capa, sin embargo se debe buscar una forma de que encuentren las demás capas. Hasta el momento la prueba que se hizo como se explicó con anterioridad, se hizo pensando en que si una partícula inicia con el mejor valor de la ejecución anterior permitiría guiar más rápidamente a las demás partículas en vista de que su ubicación desde el principio, fue diciente y cercana al punto de interés.

4. Espacio de Discusión

Se planteó que muy seguramente el problema se atribuía a la métrica de comparación de trazas ya que el algoritmo se había implementado en funciones prueba exitosamente y el único cambio en cuanto a la implementación del mismo tiene que ver con la forma en la que el recibe la información de la comparación de trazas para tomar la decisión de seguir iterando, buscar una mejor solución o aceptar la que ya está.

Se discutieron temas tratados con la realizaron de las pruebas en Matlab de manera remota en el minicomputador OSAKA, en vista de que no era posible realizar pruebas dejándolas corriendo para finalmente recoger los resultados, puesto que apagar el computador o terminar la conexión a Internet ocasionaba que el proceso que se estaba haciendo de manera remota se terminará de manera instantánea.

También se trató el tema de una posible combinación de las métricas de comparación hasta ahora implementadas, esto con el objetivo de intentar mejorar el

rendimiento del algoritmo. Por otra parte también se recomendó la necesidad de revisar más detalladamente la métrica usada, puesto que en ocasiones ocurre que el algoritmo acepta una peor solución con respecto a lo que se sabe, se acerca más al valor esperado. Adicional a esto se recomienda revisar a ojo las trazas sintéticas que se van generando buscando quizás un indicio que permita entender en que esta fallando la métrica.

Otro aspecto sobre el cual se discutió y de hecho se dio una recomendación, fue que en vista de que con el algoritmo PSO se estaba encontrando casi siempre la primer capa. Esta fuese constante para las demás partículas, de modo que ahora la atención de búsqueda en el enjambre se centrara encontrar las demás capas faltantes.

Memoria14

Uso de técnicas de Optimización Global para resolver problemas de Inversión Sísmica

Edgar Téllez Cortés, Johao Coronel Silva, Juan Arias Pinto

Universidad Industrial de Santander
Escuela de Ingenierías Eléctrica, Electrónica y Telecomunicaciones

Agosto 26 de 2014

Sesión 14: Implementación de métricas sobre Algoritmos Genéticos y PSO

1. Introducción

En esta sesión del seminario de investigación se implementaron los algoritmos PSO y GA aplicando las tres métricas de comparación de trazas hasta ahora estudiadas (correlación punto a punto, correlación cruzada y diferencia de sismogramas) utilizando el minicomputador OSAKA. Adicional a la presentación de los resultados, se buscó la manera en como comparar los resultados de ambos algoritmos y de igual manera, inferir acerca de qué métrica era la que presentaba un mejor resultado en comparación con las demás.

2. Implementación

La implementación de los algoritmos siguió con la metodología planteada en sesiones anteriores, en la cual se pretendía hacer el lanzamiento de los mismos varias veces consecutivas con la idea de que paulatinamente se fuera encontrando el valor de las capas inferiores y siguientes a las que los algoritmos fueran encontrando. Para lograr esto, se adicionaron algunas líneas de código las cuales se encargaban de preguntar en qué iteración y lanzamiento estaba el algoritmo actualmente, y fijar en cada uno de estos el mejor óptimo encontrado, de modo que este guiaran de mejor manera al enjambre o a la población dependiendo del algoritmo. Cabe destacar y

aclarar que la recomendación hecha en la sesión pasada se refería a que quizás manteniendo fija en el segundo lanzamiento del algoritmo la primer capa, ya encontrada en el primer lanzamiento, se iban a obtener mejores resultados; sin embargo, un análisis de dicha propuesta llevó a la conclusión de que hacer esto implicaría sesgar la búsqueda del enjambre. A pesar de esto, surgió una nueva idea que en parte incluía la recomendación hecha, dicha idea consistía que en vez de mantener constante el valor de esta capa, ya encontrada, para toda la población o enjambre, era mejor poner este valor como punto de partida para todas las partículas donde todas tuvieran en común solo la primer capa y una sola de ellas conservara el resultado completo de la iteración anterior. Esta idea se estableció como metodología para ambos algoritmos lo cual conllevó a una mejora rotunda en los resultados encontrados.

Simplificando la explicación de la nueva metodología empleada, se lanza la primer prueba obteniendo por lo menos la primer capa, en el segundo lanzamiento todas las partículas inician con la primer capa fija, las demás son aleatorias, y solo a una de las partículas se le asignará la solución del lanzamiento anterior. Se espera que se logre encontrar la segunda capa y se ratifique el valor de la primera dado que no se obliga al enjambre a mantenerlo durante la ejecución. Terminado esto, el tercer lanzamiento se hará iniciando en todo el enjambre las dos capas ya encontradas de modo que esto siga ocurriendo al punto de encontrar las cuatro capas.

3. Análisis de Resultados

Con los cambios realizados en el algoritmo PSO, en todas las pruebas se alcanzó el valor de *sloth* correspondiente a la primer capa del modelo de subsuelo de referencia, partiendo de valores aleatorios. Una vez se mantuvo fijo el valor del primer *sloth* encontrado, el algoritmo encontró el siguiente valor, manteniendo fijos los dos primeros valores de *sloth* ya encontrados; el algoritmo PSO logró llegar al tercer valor de *sloth*. Finalmente establecidos estos tres valores se esperaba que este encontrara el último valor para llegar a que el modelo sintético fuera igual al de referencia, sin embargo, el algoritmo no alcanzó dicho valor sino que tomó un valor arbitrario y terminó en una convergencia prematura. Con base en estos resultados se ve un gran avance, pues se pasó a encontrar no solo la primer capa sino tres de las cuatro que presenta el modelo de referencia. Además de esto, este óptimo comportamiento se dio no solo para una métrica sino para las tres implementadas. En cuanto a algoritmos genéticos aplicando la misma estrategia de implementación, solo se obtuvieron los valores de la primer capa del modelo del subsuelo de referencia en un valor aproximado en todas las pruebas realizadas. Estos resultados se hicieron para las tres métricas implementadas.

Los resultados anteriormente nombrados se pueden ver con mayor detalle en las estructuras guardadas en la carpeta Estructuras - sesión 14, donde se puede ver el detalle de las pruebas así como la trayectoria de los *sloths* de las partículas e individuos respectivamente, teniendo en cuenta que los campos que permite ver la estructura son: Fecha de realización de la prueba, tipo de algoritmo implementado,

parámetros, trayectoria de los *sloths*, trayectoria de coeficientes, trayectoria de las magnitudes, modelo encontrado, mejor individuo, tiempo de ejecución de la prueba, arquitectura computacional utilizada. Cabe aclarar que en las estructuras se encuentran subcarpetas con f=1, f=2, f=3 y f=4 para ambos algoritmos, este parámetro indica las veces que se ejecutó el algoritmo bajo la metodología de guiar el enjambre o la población. Para el algoritmo PSO debido a que presentó mejores resultados, se hicieron pruebas con f=3 y f=4, para el algoritmo GA únicamente se hicieron pruebas hasta f=2 dado que los buenos resultados no llegaban a encontrar la segunda capa del modelo de referencia.

4. Análisis y Discusión

Dentro de los temas tratados en el espacio de discusión, se lanzó la pregunta de por qué algoritmos genéticos no funcionaba con el mismo éxito que se logró obtener con PSO, A partir de esto, se pudo analizar que una posible fuente de error en el algoritmo se da en el cruce en vista de que no respeta las dimensiones de cada individuo. Es decir, cada individuo está conformado por 4 dimensiones y cada dimensión ocupa 16 bits, al final se tiene un vector de 64 bits por cada individuo el cual esta representado por esta cadena binaria; el punto de corte que tiene lugar en el proceso de cruce de individuos se esta haciendo de manera aleatoria sin importar los 3 puntos de cruce que separaban las dimensiones. Teniendo en cuenta esto, es posible que en vez de cruce se este realizando una forma particular de mutación. La propuesta para la nueva implementación se basa en que el punto de cruce debería ser aleatorio pero que solo pueda tomar uno de los tres valores que dividen el vector en las 4 dimensiones, de este modo, la creación de nuevos individuos se hará con valores de *sloths* reales solo con el intercambio en dimensiones de los padres escogidos en el proceso de elitismo.

Memoria15

Uso de técnicas de Optimización Global para resolver problemas de Inversión Sísmica

Edgar Téllez Cortés, Johao Coronel Silva, Juan Arias Pinto

Universidad Industrial de Santander
Escuela de Ingenierías Eléctrica, Electrónica y Telecomunicaciones

Septiembre 3 de 2014

Sesión 15: Implementación de métricas en Algoritmos Genéticos y PSO, segunda parte

1. Introducción

En esta sesión del seminario de investigación se realizó la segunda implementación de las tres métricas de comparación de trazas sísmicas, en la utilización de algoritmos genéticos aplicados a inversión sísmica. Se aplicaron las tres métricas de comparación a dos modelos distintos: El modelo inicial, donde las capas no son uniformes, y uno con capas paralelas con una última capa agregada con el fin de recibir información del *sloth* de la última capa. Este se explica en la sección de implementación. Adicional a esto, se hizo énfasis en las modificaciones que tuvieron lugar en GA y se explicó la manera como estas modificaciones tienen lugar en la interpretación que se tiene con respecto a la analogía con la naturaleza.

2. Implementación

Teniendo en cuenta las recomendaciones recibidas en la sesión pasada, se modificó el cruce de Algoritmos Genéticos haciendo que este fuera por dimensiones estableciendo tres puntos de cruce aleatorios cada 16 bits en la cadena de los individuos. Esta modificación tiene lugar en la naturaleza, en la manera como generación tras generación van surgiendo nuevos tipos de individuos a causa de los fenómenos de cruce y mutación evidenciados. La importancia radica en que a pesar de que estos

individuos son nuevos, algunas características genéticas del gen se conservan, es decir que un nuevo individuo debe guardar ciertos rasgos de los padres, para permitir la evolución de dichas características. Este aspecto era el que se estaba obviando en la implementación del algoritmo, puesto que al no respetar el gen en cada individuo, es decir dimensión tras dimensión, se estaban creando nuevos individuos sin ninguna de las características de los padres.

Se observó en detalle el comportamiento de las trazas sísmicas y las métricas de comparación de las mismas, viendo que sin importar el valor de *sloth* de la última capa (la más profunda del modelo del subsuelo), las métricas de comparación para estas trazas eran iguales, por lo tanto había un problema en la información presente en las mismas. Observando en detalle lo ocurrido, se logró determinar que las trazas solo tenían tres arribos, osea que las ondas generadas por las fuentes explosivas solo se estaban reflejando en tres discontinuidades que correspondían a la base de las tres primeras capas, pero al no haber una discontinuidad entre la última capa y el límite inferior del modelo, no ocurría una reflexión y refracción del frente de onda, por lo tanto las trazas no mostraban un arribo que evidenciara información por parte de esta última capa. Debido a esto, era lógico que cuando el algoritmo encontrara el valor de los tres primeros *sloths* convergiera prematuramente ya que no tenía información para seguir iterando.

Teniendo en cuenta lo anterior, se llegó a que si el problema era el modelo de capas del subsuelo y no la implementación de los algoritmos y las métricas utilizadas, la solución radicaba en el modelo, por tal razón se debía modificar y buscar una forma de obtener información de la última capa, es por eso que se decidió agregar una discontinuidad al modelo para que los frentes de ondas generadas por la fuente pudieran reflejarse y ser captados por los geófonos; esta última capa se agregó en la base del modelo haciendo la función de piso. Por otra parte fue necesario expandir el tiempo de grabación de los geófonos de modo que se lograra captar en las trazas sísmicas, los arribos de la última capa utilizada como piso.

3. Análisis de Resultados

Como se mencionó anteriormente, se hicieron pruebas para dos modelos distintos del subsuelo, los resultados para cada uno de estos se muestra a continuación.

3.1. Modelo inicial de capas no uniformes

Teniendo en cuenta las modificaciones de implementación de cruce, se observó una mejoría en la precisión del hallazgo del primer *sloth* para las tres métricas implementadas. Además de esto el algoritmo regido bajo la métrica de correlación, logró encontrar los tres primeros *sloths* del modelo, representando así una mejoría notable en los resultados que hasta el momento se habían tenido en GA. La estructura correspondiente a estas pruebas se encuentra en la carpeta Estructuras - sesión 15 en

la carpeta Modelo de capas no paralelas. También se hizo una prueba en la cual se variaron las tasas de cruce y mutación únicamente para la métrica de correlación cruzada, con el fin de quizás poder encontrar un mejor conjunto de tasas que arrojara un mejor resultado. En esta prueba se destacan los valores de:

- $Tc = 0,4$ y $Tm = 0,4$
- $Tc = 0,7$ y $Tm = 0,7$
- $Tc = 1,0$ y $Tm = 0,7$

Donde se encontraron en valores aproximados, dos de las cuatro capas esperadas. Por otra parte el mejor resultado se obtuvo para un conjunto de tasas de: $Tc = 1$ y $Tm = 0,4$ donde se alcanzó a encontrar 3 capas del modelo de referencia. La estructura correspondiente a estas pruebas se puede encontrar en la carpeta Estructuras - sesión 15 en la carpeta Modelo de capas no paralelas – GA variando tasas. En vista de que este nuevo conjunto de tasas de cruce y mutación arrojaron buenos resultados, se hizo otra prueba para los tres tipos de métricas bajo los valores de $Tc = 1$, $Tm = 0,1$, $Tc = 1$ y $Tm = 0,4$. Para el primer conjunto de tasas se obtuvo únicamente el valor de la primer capa del modelo de referencia para los tres tipos de métrica, para el segundo conjunto de tasas los resultados obtenidos fueron encontrar 2 capas también para el caso de las tres métricas. En este punto es necesario aclarar que en las pruebas pasadas se alcanzó a encontrar 3 capas para la métrica de correlación cruzada, sin embargo, en esta última prueba solo se alcanzaron a encontrar 2 capas. La razón por la cual ocurre esto, es debido a que el punto de partida es distinto entre estas pruebas por tal motivo puede que una prueba se halla favorecido más que la otra. La estructura correspondiente a estas pruebas se puede encontrar en la carpeta estructuras - sesión 15 en la carpeta Modelo de capas no paralelas – Ajuste de tasas.

3.2. Modelo de capas paralelas con discontinuidad

En la implementación realizada, los mejores resultados obtenidos ocurrieron para las métricas de diferencia de sismogramas y correlación cruzada, donde se logró encontrar 2 capas. Para el caso de la métrica de correlación punto a punto se logró obtener únicamente el valor de la primer capa. La estructura correspondiente a estas pruebas se puede encontrar en la carpeta Estructuras - sesión 15 en la carpeta Modelo de capas paralelas con piso.

3.3. Modelo de capas paralelas sin discontinuidad

Las pruebas implementadas para este modelo arrojaron el reconocimiento de la primer capa para las métricas de correlación cruzada y diferencia de trazas. Por otra parte, se contó con que la métrica de correlación punto a punto logró encontrar 3 capas del modelo de referencia con una precisión alta, el valor de la última capa estuvo muy cercano al esperado. La estructura correspondiente a estas pruebas se

puede encontrar la carpeta Estructuras - sesión 15 en la carpeta Modelo de capas paralelas sin piso.

4. Análisis y Discusión

Debido a que en las dos últimas sesiones del seminario se realizaron cambios como el número y conformación de las capas del modelo del subsuelo y el tiempo de grabación de los geófonos durante la adquisición sísmica, se planteó que estos cambios realizados podrían ser contraproducentes en cuanto a la comparación de resultados ya que esto no se podría hacer si se trata experimentos totalmente diferentes. Por esta razón quedó planteada la recomendación de ser organizados en las próximas implementaciones para poder obtener resultados más dicentes y comparativos.

Cabe destacar que en este punto el interés radica en hacer muchas pruebas, pero hacerlas bien, de modo que lo que se haga y se obtenga realmente sirva como herramientas para concluir respecto al comportamiento de los algoritmos, la calidad de las métricas utilizadas y la veracidad de los resultados. Adicional a esto, se rescatan las ideas de mantener un tiempo constante de grabación de los geófonos para todas pruebas, de igual manera, que las pruebas se hagan para los tres distintos modelos del subsuelo y también para las tres métricas utilizadas.

Otro tema sobre el cual se discutió fue acerca de la manera como se hace el proceso de mutación, donde surge la duda de si se esta haciendo de manera correcta conforme se debe hacer en algoritmos genéticos teniendo relación con lo que ocurre en la naturaleza, o si es necesaria alguna modificación. Por tal razón se recomendó revisar la manera en como un individuo muta y si es necesario que se haga por genes o en toda la cadena binaria.

Memoria16

Uso de técnicas de Optimización Global para resolver problemas de Inversión Sísmica

Edgar Téllez Cortés, Johao Coronel Silva, Juan Arias Pinto

Universidad Industrial de Santander

Escuela de Ingenierías Eléctrica, Electrónica y Telecomunicaciones

Septiembre 9 de 2014

Sesión 16: Implementación de métricas en Algoritmos Genéticos y PSO, tercera parte

1. Introducción

En esta sesión del seminario de investigación se realizó la tercera implementación de las tres métricas de comparación de trazas sísmicas, en la utilización de los algoritmos PSO y Genéticos aplicados al proceso de inversión sísmica. Las tres métricas de comparación se aplicaron a los dos modelos de subsuelo, explicados en la sesión anterior, con la diferencia de que en algunas pruebas se asignó un mismo punto de partida en la población inicial de los algoritmos, con la idea de hacer un poco más comparables las pruebas. Como ya se había planteado en sesiones anteriores, los resultados quedaron consignados en estructuras que se pueden observar desde cualquier computador que tenga instalado Matlab.

2. Análisis de Resultados

Se realizaron pruebas para los dos modelos del subsuelo, con la particularidad de que estos presentaban una discontinuidad adicional en la base de la capa más profunda por las razones mencionadas en la sesión anterior. A continuación se muestran los resultados para cada uno:

2.1. Modelo de capas no uniformes

Se empleó una estrategia para la implementación de Algoritmos Genéticos en la que se inició la búsqueda empleando una tasa de mutación alta y terminó con una baja, sin embargo, con esta metodología ninguna capa es encontrada, esto se puede observar en la carpeta Estructuras - sesión 16 en la carpeta Modelo de capas no uniforme – GA – Tazas de mutación dinámica. En cuanto a la implementación normal del algoritmo GA, tampoco se obtuvieron resultados satisfactorios en ninguna de las tres métricas implementadas. La estructura correspondiente a estas pruebas se encuentra en la misma ruta de las pruebas anteriores en la carpeta Métricas.

En cuando a la implementación de PSO bajo las métricas de comparación de correlación normal y cruzada, los resultados fueron aceptables debido a que en el caso de la correlación, se logró encontrar el valor de tres capas; y en el caso de la correlación cruzada, se encontraron dos. En cuanto a la métrica de diferencia de sismogramas no se encontró el valor de ninguna capa. La estructura correspondiente a estas pruebas se puede encontrar en la carpeta Estructuras - sesión 16 en la carpeta Modelo de capas no uniforme – PSO.

2.2. Modelo de capas paralelas con discontinuidad

En algoritmos genéticos solo se obtuvieron resultados aceptables en la implementación de las métricas de correlación normal y cruzada, logrando encontrar los dos primeros *sloths* del modelo de referencia. La métrica de diferencia de sismogramas en este caso no se implementó. La estructura correspondiente a estas pruebas se puede encontrar en la carpeta Estructuras - sesión 16 en la carpeta Modelo de capas paralelas con piso – GA.

En cuanto a PSO los resultados fueron muy buenos debido a que para las tres métricas aplicadas, el algoritmo encontró el valor de los *sloths* de todas las capas del modelo del subsuelo con la particularidad de que con la

métrica dos (diferencia de sismogramas), el valor de la última capa se desvió del esperado en la segunda décima; aún así el comportamiento del algoritmo fue el deseado. Estos resultados se pueden ver en la carpeta Estructuras - sesión 16 en la carpeta Modelo de capas paralelas con piso – PSO.

3. Espacio de Discusión

La discusión entorno a la modificación que se le hizo al modelo de capas del subsuelo con la nueva capa de piso, se encamino en que a pesar de ser una muy buena idea y de hecho solución a la problemática que se estaba presentando; constituye una modificación radical a la analogía que se tiene de estos temas en campo, puesto que la razón por la cual no era posible acceder a la información de la capa, se refería a que la energía del frente de onda no es suficiente para llegar a los geófonos a causa de la pérdida de la misma a medida que el frente de onda recorre terreno. Teniendo en cuenta esto, es comprensible la razón por la cual no se lograba en algunos casos encontrar el modelo de referencia en su totalidad, dado que no había forma de obtener esta información sin la capa de piso, es por eso, que tal y como estaban actuando los algoritmos en los casos donde se encontraron 3 capas de modelo de referencia, es una clara razón de su correcto funcionamiento.

Por otra parte se discutió acerca de la efectividad que se tenía con respecto a las métricas implementadas, tratando el caso de que dichas métricas de comparación no estaban del todo ajustadas para que dado un par de sismogramas, dieran una medida muy acertada de qué tan parecidos son. Todo esto tiene lugar debido a que el hecho de representar con un número o vectores de coeficientes la comparación de una gran cantidad de señales, tarde que temprano se quedará corto con respecto a la precisión requerida. La razón por la cual se dice que estas métricas no estaban bien ajustadas aparte de lo anteriormente comentado, se refiere a que se hicieron varias pruebas donde se realizaba una comparación entre unas trazas sintéticas provenientes de modelos con *sloths* lejanos y cercanos a los de referencia y las trazas de referencia ya conocidas. Los resultados obtenidos mostraban que en muchas ocasiones el valor de comparación indicaba que modelos más lejanos al de referencia generaban un indicador más alto que inclusive modelos casi que iguales que al de referencia, debido a esto fue que se generó la discusión y el cuestionamiento de si las métricas utilizadas a lo largo del seminario estaban bien ajustadas y si su aplicabilidad iba de la mano al problema que se quería dar solución. Sin embargo, los resultados obtenidos al final y el recuento

del avance investigativo y práctico a lo largo del seminario, apuntan a una satisfacción por el hecho de que los algoritmos trabajaron bien y que la interpretación de los mismos permitió atacar el problema de inversión sísmica de una manera óptima y productiva.