

01149 31



UNIVERSIDAD NACIONAL
AUTÓNOMA DE
MÉXICO

UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

POSGRADO EN CIENCIA E INGENIERÍA DE LA COMPUTACIÓN

**APLICACIÓN DE LA COMPUTACIÓN
EVOLUTIVA EN LA ESTIMACIÓN DE
COTAS TOPOGRÁFICAS**

T E S I S

QUE PARA OBTENER EL GRADO DE:

MAESTRO (A) EN INGENIERÍA

P R E S E N T A:

ROSALVA MENDOZA RAMÍREZ

DIRECTORA DE LA TESIS: DRA. KATYA RODRÍGUEZ V.

MÉXICO, D.F.

2002.



Universidad Nacional
Autónoma de México



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

ESTA TESIS NO SALE
DE LA BIBLIOTECA

Autorizo a la Dirección General de Bibliotecas de la
UNAM a difundir en formato electrónico e impreso el
contenido de mi trabajo recepcional.

NOMBRE: ROSALVA

MENDOZA ROMÍEZ

FECHA: 09/11/2002

FIRMA: *Rosalva*

DEDICATORIA

A mis padres:

Ildefonso y Elvia.

Por darme la oportunidad de vivir y brindarme siempre lo mejor de ustedes.

A mis hermanos:

Gerardo, Alejandro, Adrián y Omar.

Por lo bello que ha sido vivir teniéndolos como hermanos.

A mi hija:

Carla Samara.

Por ser el regalo más hermoso y grande que Dios y la vida me han hecho.

A mi esposo:

Salvador Ulises.

Por haber unido tu camino al mío y quererme con todos mis defectos.

A mis cuñadas:

María Guadalupe y María de los Angeles.

Gracias por querer y cuidar a mis hermanos.

A mis sobrinos:

Jonathan Saúl, Yeraldín, Bianca y Yarazeth Deyanira.

Por llenar de luz la vida de todos nosotros y en especial la de sus abuelitos.

A mi tía Alis:

Por contar siempre con tu cariño y apoyo.

A mis abuelitos:

María y Francisco.

Por darme unos recuerdos memorables de mi infancia.

A mi amigo:

Héctor Gutiérrez Puente.

Por ser uno de mis ángeles aquí en la tierra.

AGRADECIMIENTOS

A la Universidad Nacional Autónoma de México (UNAM):

Por haberme brindado la oportunidad de ser orgullosamente parte de su personal y permitirme continuar mis estudios.

Al Instituto de Ingeniería (II):

Por apoyar mi trabajo y estudios.

Al Instituto de Investigaciones en Matemáticas Aplicadas y en Sistemas (IIMAS):

Por abrirme sus puertas y confiar en mí.

Al Dr. Boris Escalante Ramírez:

Por toda su comprensión y apoyo.

A la Dra. Katya Rodríguez Vázquez:

Por toda su ayuda y tiempo dedicados a la elaboración de este trabajo.

A mi jurado:

Dr. Luis Alvarez Icaza, Dr. Ricardo Arámbula C., Dr. Carlos Cruickshank V. Y Dr. Rafael Pérez Y Pérez.

Por el tiempo dedicado a la revisión de la tesis y por sus valiosas sugerencias.

A mis jefes:

M. en I. Víctor Franco, Dr. Carlos Cruickshank V y Dr. Ramón Domínguez M.

Por contar siempre con su apoyo.

A todos mis maestros, especialmente a:

M. en C. Mario Peña y M. en I. Roberto Tovar.

Gracias a la oportunidad brindada por ustedes pude continuar mis estudios.

A mi compañera de trabajo:

Lic. Margarita López Herranz.

Por todas sus valiosas correcciones.

A mis compañeros:

Dora Carmen Gálvez Cruz y Ricardo Duarte Pérez.

Por todas las experiencias compartidas.

INDICE

RESUMEN	
1. INTRODUCCIÓN	1
1.1 Antecedentes	1
1.2 Objetivos	2
1.3 Organización de la tesis	3
2. ALGORITMOS GENÉTICOS	5
2.1 Historia de los Algoritmos Genéticos	5
2.1.1 <i>Antecedentes Biológicos</i>	5
2.1.2 <i>Evolución y Genética Artificial</i>	9
2.2 Forma de operación de los Algoritmos Genéticos	12
3. ESTIMACIÓN DE COTAS TOPOGRÁFICAS: DEFINICIÓN DEL PROBLEMA	18
3.1 Antecedentes	18
3.2 Descripción del Algoritmo Genético	21
4. ANÁLISIS DE DESEMPEÑO: ALGORITMO GENÉTICO SIMPLE	25
4.1 Algoritmo Genético Binario	26
4.1.1 Análisis de comportamiento de los coeficientes	27
4.2 Algoritmo Genético con representación de punto flotante (real)	34
5. PROGRAMACIÓN GENÉTICA	40
5.1 Antecedentes	40
5.2 Conceptos básicos de la Programación Genética	42
5.2.1 <i>Operadores Genéticos</i>	45
5.2.2 <i>Métodos de selección</i>	51
5.3 Aplicación de la Programación Genética en la estimación de cotas topográficas	52
5.3.1 <i>Optimización regional</i>	54
5.3.2 <i>Optimización global</i>	61
6. CONCLUSIONES	69
7. REFERENCIAS	72
ANEXO A: OPTIMIZACIÓN DE PARÁMETROS	
ANEXO B: ALGORITMO GENÉTICO SIMPLE: CÓDIGO FUENTE	
ANEXO C: CONJUNTO GENERAL DE DATOS	
ANEXO D: ARCHIVOS DE DATOS PARA LA APLICACIÓN DE LA PROGRAMACIÓN GENÉTICA	

RESUMEN

La importancia de tener una expresión, obtenida en forma teórica o experimental, para representar un fenómeno, radica en que ésta permite evaluar la variable de interés en puntos en los que se carece de información, así como conocer otras variables involucradas en el problema. Por ejemplo, si se tiene una expresión analítica para calcular en cualquier punto de una región la función de corriente, los componentes de la velocidad tanto en dirección vertical (y) como horizontal (x) pueden ser obtenidos al derivar la función.

En el Instituto de Ingeniería de la UNAM, se desarrolló un modelo matemático en coordenadas curvilíneas ajustadas a las fronteras en flujos a superficie libre (Mendoza *et al*, 1996). Ese modelo requiere la estimación de cotas del terreno, así como de sus gradientes en las direcciones principales, en cualquier punto del dominio. El problema se resolvió al representar la elevación (cota topográfica) como una función dependiente de dos variables y hacer el desarrollo de ésta, hasta los términos de tercer orden, usando una serie de Taylor. En la serie propuesta, las derivadas se sustituyeron por coeficientes calculados minimizando la función, con un criterio de mínimos cuadrados, y resolviendo el sistema de ecuaciones con un método de optimización desarrollado por Davidon, Fletcher y Powell (Walsh, 1960), conocido como DFP.

Los resultados que se obtuvieron no fueron del todo satisfactorios (Mendoza *et al*, 1996); por ello, en el presente estudio se propone utilizar la computación evolutiva (CE), que consiste en técnicas relativamente nuevas que han probado su eficiencia en diversos campos de aplicación, pero sobre todo, han reportado excelentes logros en el campo de la optimización (Rodríguez, 2000). Los resultados obtenidos con la CE se compararán contra los obtenidos en el estudio previo arriba mencionado (Mendoza *et al*, 1996), y se darán las conclusiones derivadas del presente trabajo.

1. INTRODUCCIÓN

1.1 Antecedentes

La computación evolutiva (CE), también llamada algoritmos evolutivos (AE), tiene su fundamento en las ideas postuladas por el naturalista inglés Charles R. Darwin (1809 – 1882) (Encarta, 2000), las cuales dieron origen a la moderna teoría de la evolución de las especies, y en los trabajos realizados por el genetista austriaco Gregor J. Mendel (1822 – 1884) (Encarta, 2000) sobre los mecanismos de transmisión de la herencia genética.

La CE o AE se divide en tres ramas: los algoritmos genéticos (AG), las estrategias evolutivas (EE) y la programación evolutiva (PE); sólo la primera de ellas tiene una subdivisión que se denomina programación genética (PG).

Las tres ramas arriba mencionadas son muy parecidas y comparten muchos aspectos; cada una de ellas se puede resumir, a grandes rasgos, de la siguiente manera:

- Los AG son probablemente los representantes más conocidos de la CE, y aquéllos cuyo uso está más extendido. Fueron concebidos originalmente por John Holland y descritos en el texto “Adaptation in Natural and Artificial Systems” (Holland, 1975), que es considerado un clásico dentro de la literatura dedicada al tema. La característica fundamental de los AG es el uso de un operador de recombinación o cruce, como mecanismo principal de búsqueda, y de un algoritmo de reproducción proporcional al desempeño.
- Las EE son técnicas que empezaron a desarrollarse en Alemania. Su objetivo inicial era servir de herramienta para la optimización de parámetros en problemas de ingeniería. Se caracterizan por manejar vectores de números codificados en punto flotante y basan su funcionamiento en el empleo de un operador de reproducción asexual (mutación).

- La PE tiene su origen en los trabajos desarrollados en 1966 por Fogel (Rodríguez, 2000). Pone un especial énfasis en la adaptación de los individuos más que en la evolución del material genético de éstos. Ello implica una visión más abstracta del proceso, en la cual se modifica directamente el comportamiento de los individuos en lugar de trabajar sobre sus genes. Dicho comportamiento se modela mediante estructuras de datos relativamente complejas como pueden ser autómatas finitos. Tradicionalmente, estas técnicas emplean mecanismos de reproducción asexual y técnicas de selección mediante competencia directa entre los individuos.

Todas las ramas de la CE se han empleado satisfactoriamente en la resolución de problemas en diversos campos de la ciencia: ingeniería, econometría, investigación de operaciones, entre otros, y es por esto que en el presente trabajo se emplea una de ellas, concretamente los AG y su subdivisión, la PG, para resolver un problema de optimización de parámetros, que anteriormente se solucionó usando un método de búsqueda tradicional, con el cual no se obtuvieron resultados del todo satisfactorios (Mendoza *et al*, 1996).

1.2 Objetivos

Los objetivos del presente trabajo se resumen en dos puntos centrales:

- Evaluar el desempeño de los AG para optimizar nueve coeficientes de una función de tipo polinomial dependiente de dos variables, que se empleará para estimar el valor de la cota topográfica en un conjunto de datos topográficos digitalizados de una zona de un río de la República Mexicana y comparar los resultados obtenidos con los reportados en un estudio previo (Mendoza *et al*, 1996), en el cual se usó para la optimización el método desarrollado por Davidon, Fletcher y Powell (Walsh, 1960), conocido como DFP y clasificado como un método cuasi-Newton dentro de las técnicas de optimización tradicionales.

- Utilizar la PG para obtener una función y comparar los resultados logrados con ella en la estimación de cotas topográficas contra los arrojados por la función anteriormente propuesta (un desarrollo en serie de Taylor, hasta los términos de tercer orden, para una función de dos variables).

1.3 Organización de la tesis

La organización del trabajo es la siguiente:

- En el capítulo dos se realiza una breve introducción de los aspectos más generales de los AG.
- En el capítulo tres se describe el problema de optimización que se estudia y cómo se resolverá usando los AG.
- En el capítulo cuatro se presentan los resultados obtenidos usando los AG y se evalúa el desempeño de éstos comparándolo con el DFP.
- En el capítulo cinco se efectúa una introducción de los aspectos más generales de la PG y se describe el procedimiento para aplicar esta técnica en la obtención de la o las nuevas funciones que sirvan para la estimación de cotas topográficas. También se analiza el desempeño de la o las nuevas funciones y se compara con el logrado por la serie de Taylor propuesta.
- Finalmente, en el último capítulo se dan las conclusiones y recomendaciones derivadas del presente trabajo.

Se incluyen cuatro apéndices: el primero describe el método de optimización DFP, que es contra el cual se hace la comparación del AG empleado; en el segundo, se incluye el código fuente del programa de cómputo elaborado para la resolución del problema planteado usando AG en su representación binaria; en el tercero, se proporciona el conjunto de

datos general con los que se trabajó; por último, en el cuarto, se presentan los archivos de datos utilizados para la aplicación de la PG.

2. ALGORITMOS GENÉTICOS

2.1 Historia de los Algoritmos Genéticos

2.1.1 Antecedentes Biológicos

La teoría de la evolución fue descrita por Charles Darwin 20 años después de su viaje por las islas Galápagos en el Beagle, en el libro "Sobre el Origen de las Especies por medio de la Selección Natural" (Encarta, 2000).

La hipótesis de Darwin, presentada junto con el también naturalista británico Alfred R. Wallace (1823 - 1913) (Encarta, 2000), que llegó a las mismas conclusiones en forma independientemente, es que pequeños cambios heredables en los seres vivos y la selección son los dos hechos que provocan el cambio en la Naturaleza y la generación de nuevas especies.

Darwin desconocía cuál es la base de la herencia, pensaba que los rasgos de un ser vivo eran como un fluido, y que los "fluidos" de los dos padres se mezclaban en la descendencia. Esta hipótesis tenía el problema de que, al cabo de cierto tiempo, una población tendría los mismos rasgos intermedios.

Fue Mendel (Enciclopedia didáctica de ciencias naturales, 1998) quien descubrió que los caracteres se heredaban de forma discreta, y que se tomaban del padre o de la madre, dependiendo de su carácter dominante o recesivo. A estos caracteres que podían tomar diferentes valores se les llamaron genes, y a los valores que podían tomar, alelos. En realidad, las teorías de Mendel, que trabajó en total aislamiento, se olvidaron y no se volvieron a redescubrir hasta principios del siglo XX.

En 1879 el biólogo alemán Walther Flemming (1843 - 1905) (Encarta, 2000) describió el proceso de división celular, al cual llamó mitosis, y describió ciertos elementos activos de ella, como filamentos en los que se agregaba la cromatina del núcleo celular durante la división; poco más adelante, en 1888, el médico alemán H. W. G. Waldeyer (1837 - 1921) (Encarta, 2000) llamó a estos filamentos cromosomas.

El siglo XX trajo consigo los descubrimientos más importantes. Fue en los años 50, cuando el biofísico y premio Nobel estadounidense J. D. Watson (1928 -) (Encarta, 2000) y el biofísico inglés F. H. C. Crick (1916 -) (Encarta, 2000), basados en los trabajos realizados por el biofísico británico, de origen Neozelandés, M. H. F. Wilkins (1916 -) (Encarta, 2000), desentrañaron la estructura en doble hélice de la molécula del ácido desoxirribonucleico (ADN). Esta sustancia es la que transmite las características genéticas de una generación a la siguiente. Los cromosomas están compuestos de ADN, y por tanto los genes están en los cromosomas.

La macromolécula de ADN está compuesta por bases púricas y pirimidínicas: la adenina, citosina, guanina y timina. La combinación y la secuencia de estas bases forma el código genético, único para cada ser vivo. Grupos de 3 bases forman un codón, y cada codón codifica un aminoácido (el que exprese ese aminoácido o no depende de otros factores); el código genético codifica todas las proteínas que forman parte de un ser vivo. Mientras que al código genético se le llama genotipo, al cuerpo que construyen esas proteínas, modificado por la presión ambiental, la historia vital, y otros mecanismos dentro del cromosoma, se llama fenotipo.

No toda la cadena de ADN codifica proteínas, es decir, no todos son genes; las zonas que codifican proteínas se llaman intrones, las zonas que no lo hacen, exones. La cantidad de ADN basura aumenta desde los seres vivos más simples, como las bacterias, donde no hay nada, hasta los seres humanos, donde gran cantidad del ADN no codifica.

Todos estos hechos forman hoy en día la teoría del neodarwinismo, que afirma que la historia de la mayoría de la vida está causada por una serie de procesos que actúan en y dentro de las poblaciones: reproducción, mutación, competición y selección. La evolución se puede definir entonces como cambios en el conjunto genético de una población.

Un tema polémico, con opiniones variadas dependiendo de si se trata de informáticos evolutivos, biólogos o genetistas, es si la evolución optimiza o no. Según los informáticos evolutivos, sí lo hace puesto que va creando seres cada vez más perfectos, y su obra perfecta es el hombre; además, indicios de esta optimización se encuentran en el organismo de los animales, desde el tamaño y tasa de ramificación de las arterias, diseñada para maximizar flujo, hasta el metabolismo, que optimiza la cantidad de energía extraída de los alimentos.

Sin embargo, los genetistas y biólogos evolutivos afirman que la evolución no optimiza, sino que adapta y optimiza localmente en el espacio y el tiempo; evolución no significa progreso. Un organismo más evolucionado puede estar en desventaja competitiva con uno de sus antepasados, si se colocan en el ambiente del último.

Estos mecanismos de cambio serán necesarios para entender los algoritmos evolutivos, pues se trata de imitarlos para resolver problemas de ingeniería; por eso es importante conocerlos más a profundidad. Los mecanismos de cambio alteran la proporción de alelos de un tipo determinado en una población, y se dividen en dos tipos: los que disminuyen la variabilidad, y los que la aumentan

Los principales mecanismos que disminuyen la variabilidad son los siguientes:

- ◆ Selección natural: los individuos que tengan algún rasgo que los haga menos válidos para realizar su tarea de seres vivos, no llegarán a reproducirse, y por lo tanto, su patrimonio genético desaparecerá del conjunto; algunos no llegarán ni siquiera a nacer. Esta selección sucede a muchos niveles: competición entre miembros de la especie (intraespecífica), competición entre diferentes especies, y competición predador-presa, por ejemplo. También es importante la selección sexual, en la cual las hembras eligen el mejor individuo de su especie disponible para reproducirse.

- ◆ Deriva génica: el simple hecho de que un alelo sea más común en la población que otro, causará que la proporción de alelos de esa población vaya

aumentando en una población aislada, lo cual a veces da lugar a fenómenos de especiación, por ejemplo, por el denominado efecto fundador.

Otros mecanismos aumentan la diversidad, y suceden generalmente en el ámbito molecular. Los más importantes son:

- ◆ **Mutación:** la mutación es una alteración del código genético, que puede suceder por múltiples razones. En muchos casos, las mutaciones eliminan la ADN-polimerasa, que igual duplica, corrige, o desinvierte un segmento genético mal colocado. En muchos otros casos, las mutaciones, que cambian un nucleótido por otro, son letales, y los individuos ni siquiera llegan a desarrollarse, pero a veces se da lugar a la producción de una proteína que aumenta la supervivencia del individuo, y es pasada a la descendencia. Las mutaciones son totalmente aleatorias, y son el mecanismo básico de generación de variedad genética. A pesar de lo que se piensa habitualmente, la mayoría de las mutaciones ocurren de forma natural, aunque existen sustancias mutagénicas que aumentan su frecuencia.

- ◆ **Poliploidía:** mientras que las células normales poseen dos copias de cada cromosoma (diploides), y las células reproductoras una (haploides), puede suceder por accidente que alguna célula reproductora tenga más de dos grupos de cromosomas; si se logra combinar con otra célula diploide o haploide dará lugar a un ser vivo con varias copias de cada cromosoma. La mayoría de las veces, la poliploidía da lugar a individuos con algún defecto (por ejemplo, el tener 3 copias del cromosoma 21 da lugar al mongolismo), pero en algunos casos se crean individuos viables. Un caso conocido de mutación fue el que sufrió el mosquito *Culex pipiens*, en el cual se duplicó un gen que generaba una enzima que rompía los organofosfatos, componentes habituales de los insecticidas.

- ◆ **Recombinación:** cuando las dos células sexuales, o gametos, una masculina y otra femenina se combinan, los cromosomas de cada una también lo hacen, intercambiándose genes que a partir de ese momento pertenecerán a un cromosoma diferente. A veces también se produce traslocación dentro de un

cromosoma (una secuencia de código se elimina de un sitio y aparece en otro lugar, o en otro cromosoma).

- ♦ Flujo genético: o intercambio de material genético entre seres vivos de diferentes especies. Normalmente se produce a través de un vector (o agente), que suelen ser virus o bacterias; éstas incorporan a su material genético genes procedentes de una especie a la que han infectado, y cuando infectan a un individuo de otra especie pueden transmitirle esos genes a los tejidos generativos de gametos.

2.1.2 Evolución y Genética artificial

Una vez que se han descrito los mecanismos de la evolución, y una amplia gama de problemas que pueden o no tener relación entre sí, se pueden ahora abordar el cómo evolucionó la idea de simular o imitar la evolución con el objeto de resolver problemas humanos.

Las primeras ideas, incluso antes del descubrimiento del ADN, vinieron de Von Neumann (1903 – 1957) (Encarta, 2000), uno de los mayores científicos de este siglo. Von Neumann afirmó que la vida debía de estar apoyada por un código que a la vez describiera cómo se puede construir un ser vivo y que ese ser creado fuera capaz de autoreproducirse. Por tanto, un autómata tendría que ser capaz, aparte de contener las instrucciones para hacerlo, de copiar tales instrucciones a su descendencia.

Sin embargo, no fue hasta mediados de los años cincuenta, tiempo en el que el rompecabezas de la evolución se había prácticamente completado, cuando Box (Box *et al*, 1959) comenzó a pensar en imitarla para, en su caso, mejorar procesos industriales. La técnica de Box, denominada EVOP (Evolutionary Operation), consistía en elegir una serie de variables que regían un proceso industrial.

Sobre esas variables se creaban pequeñas variaciones que formaban un hipercubo; se probaba entonces con cada una de las esquinas del hipercubo

durante un tiempo, y al final del periodo de pruebas, un comité humano decidía sobre la calidad del resultado. Es decir, se estaba aplicando mutación y selección a los valores de los parámetros, con el objeto de mejorar la calidad del proceso. Este procedimiento se aplicó con éxito a algunas industrias químicas.

Un poco más adelante, en 1958, Friedberg y sus colaboradores (Friedberg *et al*, 1958) pensaron en mejorar, usando técnicas evolutivas, la operación de un programa. Para ello diseñaron un código máquina de 14 bits (2 para el código de operación, 6 para los datos y 6 para las instrucciones); cada programa, tenía 64 instrucciones. Un programa llamado Herman, ejecutaba los códigos creados, y otro, el Teacher o profesor, le mandaba a Herman ejecutarlos y ver si cada uno de ellos había realizado su tarea, la cual consistía en leer unas entradas, situadas en una posición de memoria, y depositar el resultado en otra, que era examinada al terminar de ejecutar la última instrucción.

Para hacer evolucionar los programas, Friedberg hizo que en cada posición de memoria hubiera dos alternativas. Para cambiar un código, alternaba las dos instrucciones (que eran una especie de alelos), o bien reemplazaba una de las dos instrucciones con una totalmente aleatoria.

En realidad, lo que estaba haciendo era usar mutación para generar nuevos códigos. Al parecer, no tuvo más éxito que si hubiera buscado aleatoriamente un programa que hiciera la misma tarea. El problema es que la mutación sola, sin ayuda de la selección, hace que la búsqueda sea prácticamente al azar.

Simultáneamente, Bremmerman (Bremmerman, 1962) trató de usar la evolución para "entender los procesos de pensamiento creativo y aprendizaje", y empezó a considerar la evolución como un proceso de aprendizaje. Para resolver un problema, codificaba las variables en una cadena binaria de 0s y 1s, y le aplicaba una mutación, cambiando un bit cada vez. De esta forma, estableció que la tasa ideal de mutación debía de ser tal que se cambiara un bit cada vez. En todo caso, se llegaba a un punto, la "trampa de Bremmerman", en el cual la solución no mejoraba. En intentos sucesivos trató de añadir cruzamiento entre soluciones,

pero tampoco obtuvo buenos resultados. Esto demostró que el simple uso de operadores que creen diversidad no es suficiente para dirigir la búsqueda genética hacia la solución correcta.

El primer uso de procedimientos evolutivos en inteligencia artificial se debe a Reed y sus colaboradores (Reed *et al*, 1967), que trataron de hacer evolucionar un tahúr que jugaba a un juego de cartas simplificado. Las estrategias de juego consistían en una serie de 4 probabilidades de apuesta alta o baja con una mano alta o baja, con cuatro parámetros de mutación asociados. Se mantenía una población de 50 individuos, y aparte de la mutación, había intercambio de probabilidades entre dos padres. Es de suponer que los perdedores se eliminaban de la población. Aparte de, probablemente, crear buenas estrategias, llegaron a la conclusión de que el cruzamiento no aportaba mucho a la búsqueda.

Los intentos posteriores corresponden a los algoritmos evolutivos modernos, y se han seguido investigando hasta nuestros días. Algunos de ellos son simultáneos a los algoritmos genéticos, pero se desarrollaron sin conocimiento unos de otros. Por ejemplo, la programación evolutiva creada por Fogel, Owens y Walsh (Fogel *et al*, 1965), se inició como un intento de usar la evolución para crear máquinas inteligentes, que pudieran prever su entorno y reaccionar adecuadamente a él.

Para simular una máquina pensante, se utilizó un autómata celular. Un autómata celular es un conjunto de estados y reglas de transición entre ellos, de forma que, al recibir una entrada, cambia o no de estado y produce una salida. Fogel trataba de hacer aprender a estos autómatas a encontrar regularidades en los símbolos que se le iban enviando. Como método de aprendizaje usó un algoritmo evolutivo: una población de diferentes autómatas competía para hallar la mejor solución; es decir, predecir cuál iba a ser el siguiente símbolo de la secuencia con un mínimo de errores. Los peores 50% eran eliminados cada generación y sustituidos por otros autómatas resultantes de una mutación de los existentes.

De esta forma, se lograron hacer evolucionar autómatas que predecían algunos números primos (por ejemplo, uno, cuando se le daban los números más altos, respondía siempre que no era primo; la mayoría de los números mayores de 100 son no primos). En cualquier caso, estos primeros experimentos demostraron el potencial de la evolución como método de búsqueda de soluciones novedosas.

2.2 Forma de operación de los Algoritmos Genéticos

En ingeniería, econometría, investigación de operaciones, etc. Se tiene una gran cantidad de problemas en los que se desea optimizar el valor de alguna función de una o varias variables. Existen métodos “tradicionales” para manejar este tipo de problemas, entre ellos, la Programación Matemática, la programación lineal o método Simplex, la programación dinámica, la cuadrática, la convexa, etc. En estas técnicas se debe resolver un conjunto de ecuaciones para encontrar una sucesión de soluciones básicas factibles, cada una mejor que la anterior, hasta que se llega a la solución óptima.

En general, estos métodos son adecuados cuando la función a optimizar es lineal, y las variables que controlan el problema son reducidas. Sin embargo, si el caso es no lineal, y/o si las variables que intervienen son múltiples, se complica enormemente. Para este tipo de problemas, se requiere de gran capacidad de cómputo y, aún así, muy frecuentemente, las soluciones están muy lejanas de ser las óptimas.

Los AG resuelven esta clase de problemas de una manera totalmente diferente a las empleadas hasta ahora. El AG es un proceso de cómputo que emula la forma de actuar de la evolución biológica. Opera sobre una población de individuos que representan las soluciones potenciales a un determinado caso y busca producir individuos más aptos al combinar a los mejores existentes. Al emplear el principio de la “supervivencia del más apto” descarta aquellos individuos deficientes y trata de producir más de los idóneos. Con esto no sólo se producen más soluciones sino que éstas son también mejores debido a que combina las mejores cualidades de los “padres” para producir mejores “hijos”.

El elemento más importante en todo AG es la selección, la cual determina qué elementos de la población aportarán su “material genético” a la siguiente generación de individuos.

El objetivo del método de selección es proporcionar un número creciente en forma exponencial de oportunidad de sobrevivir a los individuos más aptos. El segundo atributo crítico de todo AG es que contiene alguna forma de reproducir a los individuos. Es aquí cuando dos individuos seleccionados se les permite emparentarse para reproducirse.

El emparentamiento se verifica mediante el operador genético denominado cruza. La cruza es el procedimiento mediante el cual las cadenas de bits de los padres se combinan para producir dos individuos hijos.

Cuando se usa un sólo punto de cruza, éste se escoge de forma aleatoria sobre la longitud de la cadena que representa el cromosoma, y a partir de él se realiza el intercambio de material de los dos individuos, tal como se muestra en la Figura 1.

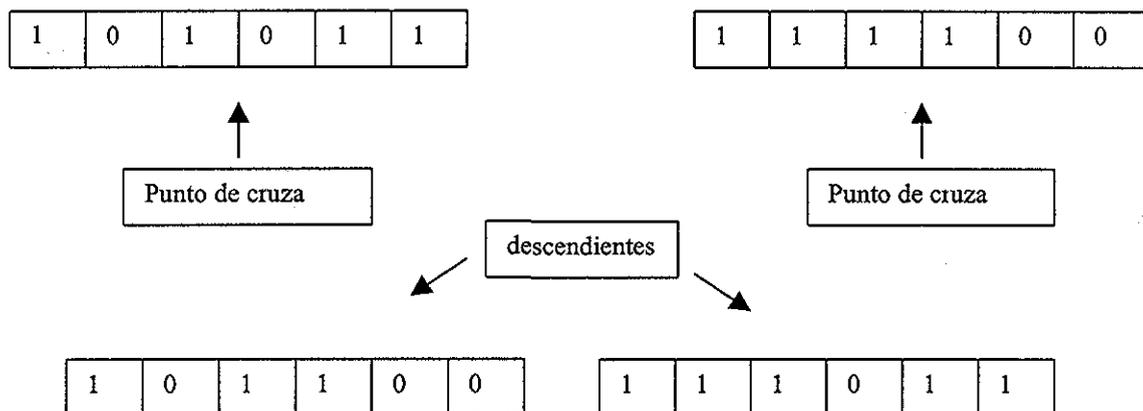


Fig 1. Cruzamiento en un sólo punto.

Cuando se usan dos puntos de cruza, se procede de manera similar, sólo que en este caso el intercambio se realiza como se ilustra en la Figura 2.

Normalmente la cruce se maneja dentro de la implementación del AG como un porcentaje que indica con qué frecuencia se efectuará. Esto significa que no todas las parejas de individuos se cruzarán, sino que habrá algunas que pasarán intactas a la siguiente generación.

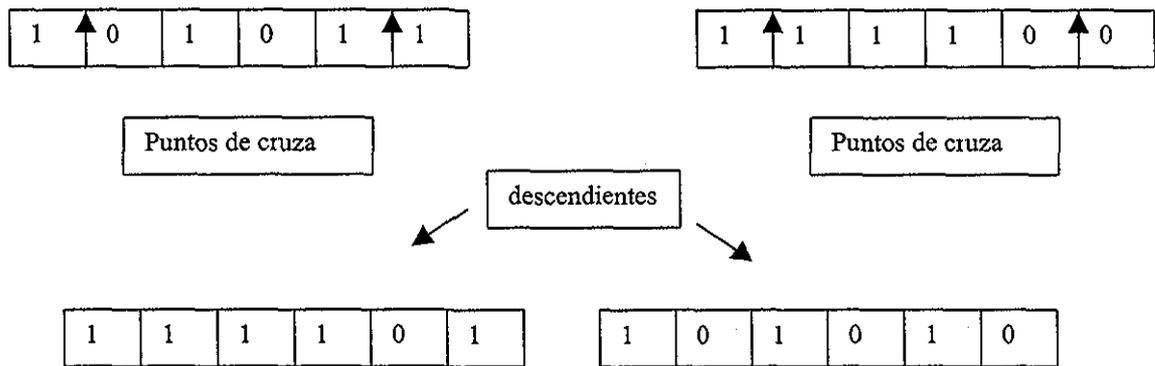


Fig 2. Cruzamiento en dos puntos.

De hecho existe una técnica desarrollada hace algunos años en la que el individuo más apto a lo largo de las distintas generaciones no se cruza con nadie, y se mantiene intacto hasta que surge otro individuo mejor que él, que lo desplazará. Dicha técnica es llamada elitismo.

Otro operador importante en los AG es la mutación. Mucha gente tiene la creencia errónea de que la mutación juega un papel predominante en la evolución natural. Esto simplemente no es cierto. La razón es que la mutación puede más fácilmente producir cambios dañinos que benéficos. Un ambiente muy propenso a la mutación mataría rápidamente a la mayoría de los individuos, o podría en el peor de los casos acabar con todos ellos. En AG, mucha mutación hace que el algoritmo degenera en búsqueda aleatoria.

La mutación realiza un cambio a uno de los genes de un individuo elegido aleatoriamente. Cuando se usa una representación binaria, el gene seleccionado se sustituye por su complemento. Esto es, un cero cambia a uno y viceversa. El operador de mutación permite la introducción de nuevo material cromosómico en la población, tal y como sucede con sus equivalentes biológicos.

Al igual que la cruce, la mutación se maneja como un porcentaje que indica con qué frecuencia se efectuará, aunque se distingue de la cruce por ocurrir mucho más esporádicamente. El porcentaje de cruce normalmente es de más del 60%, mientras que el de mutación normalmente nunca supera al 5%.

La operación de un AG simple puede ilustrarse con el siguiente segmento de pseudo-código:

- $t = 0$
- generar población inicial, $P(t)$;
- evaluar la población generada $P(t)$;
- repetir
 - $t = t+1$;
 - generar $P(t)$ usando $P(t-1)$;
 - evaluar $P(t)$;
- hasta encontrar una solución;

Esto es, primero se genera aleatoriamente la población inicial, que estará constituida por un conjunto de cromosomas o cadenas de caracteres, los cuales representan las soluciones posibles del problema. A cada uno de los cromosomas (también llamados individuos) de esta población se le aplica una función de aptitud para saber qué tan buena es la solución que se está codificando. Sabiendo la aptitud de cada individuo, se procede a la selección de los individuos que se cruzarán en la siguiente generación (teóricamente se escoge a los mejores).

Existen muchos métodos de selección (Goldberg, 1989), entre ellos se puede mencionar el de la ruleta, el torneo, la selección estocástica universal, etc., pero los dos más comunes son la ruleta y el torneo. A continuación se describe brevemente estos dos métodos, resaltando que el método de selección de la ruleta fue el que se eligió para resolver el problema que se plantea en el presente trabajo.

- La ruleta: descrito por Goldberg en (Goldberg, 1989). Este método es muy simple, consiste en crear una ruleta en la que cada individuo tiene asignada una fracción proporcional a su aptitud.

Por ejemplo, supongamos que se tiene una población de 5 individuos cuyas aptitudes están dadas por los valores mostrados en la Tabla 1.

Tabla 1. Método de la ruleta (Goldberg, 1989)

Poblador	Cadena	Aptitud	% del total
1	11010110	254	24.5
2	10100111	47	4.5
3	00110110	457	44.1
4	01110010	194	18.7
5	11110010	85	8.2
Total		1037	100.0

Con los porcentajes mostrados en la cuarta columna de la Tabla 1 se elabora la ruleta mostrada en la Figura 3. Esta ruleta se gira cinco veces para determinar los N individuos a seleccionar. Debido a que a los individuos más aptos se les asignó un área mayor de la ruleta, se espera que sean seleccionados más veces que los menos aptos.

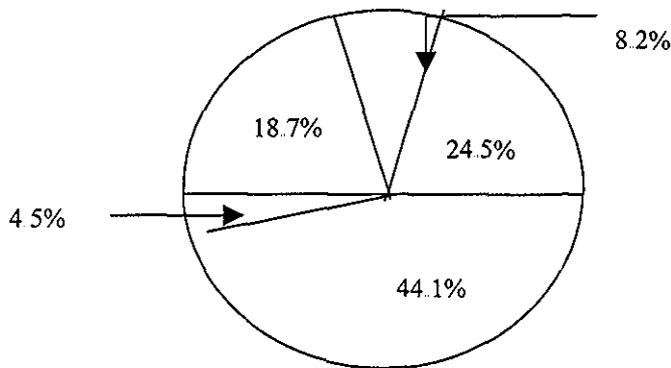


Fig 3. Ruleta que representa los valores de aptitud de los individuos de la Tabla 1.

- El torneo: la idea de este método es barajar la población y después hacer competir a los individuos que la integran en grupos de tamaño predefinido (normalmente compiten en parejas) en un torneo en el que resultarán ganadores aquéllos que tengan valores de aptitud más altos. Si se efectúa un torneo binario (i.e. competencia por parejas), entonces

la población se debe barajar dos veces. La técnica garantiza la obtención de múltiples copias del mejor individuo entre los progenitores de la siguiente generación (si se efectúa un torneo binario, el mejor individuo será seleccionado dos veces).

Una vez sentadas las bases teóricas que se necesitan conocer de los AG, se describe, en el siguiente capítulo a detalle, el problema a resolver con ellos.

3. ESTIMACIÓN DE COTAS TOPOGRÁFICAS: DEFINICIÓN DEL PROBLEMA.

3.1 Antecedentes

Un problema clásico que se presenta en hidráulica fluvial y marítima es el conocimiento adecuado del campo de velocidades y cotas de la superficie libre del agua en ríos muy anchos, llanuras de inundación, lagunas someras y marinas. La determinación de estos patrones de flujo se conoce como el cálculo de la hidrodinámica.

El cálculo de la hidrodinámica es indispensable para resolver problemas como:

- a) Determinación de corrientes en ríos muy anchos y marinas
- b) Diseño de bordos en planicies y zonas de inundación
- c) Estudio del efecto de algunas obras construidas sobre el cauce de un río
- d) Cálculo de la dispersión de contaminantes descargados a un río
- e) Predicción de la demanda bioquímica de oxígeno y oxígeno disuelto en almacenamiento
- f) Estimación de la erosión y el azolve en tramos de cauces
- g) Análisis de tendencias a la erosión marginal y al meandro, etc

Algunos de estos problemas se resuelven de manera aceptable con modelos matemáticos unidimensionales; sin embargo, en muchos otros esto no es suficiente (Jiménez *et al*, 1997). Tradicionalmente, los modelos numéricos se basan en resolver las ecuaciones del movimiento del flujo con esquemas de diferencias finitas; la aplicación de esta técnica requiere hacer una discretización de la topografía y batimetría de la zona a estudiar. Generalmente, la zona en estudio donde escurre el flujo tiene forma irregular; esto dificulta enormemente la aplicación de los modelos matemáticos tradicionales al tratar de ajustar un sistema de coordenadas rectangulares a las fronteras irregulares que definen la superficie libre del agua. Para salvar este problema se han usado métodos de elemento finito, sin embargo, se ha visto que aunque esta técnica mejora notablemente la definición de las fronteras irregulares, los métodos de solución de las ecuaciones se complican a tal grado que dejan de ser atractivos los métodos de elemento finito. Otra alternativa que se ha propuesto para calcular

en forma adecuada las características de flujos en fronteras de forma irregular, es el uso de sistemas de coordenadas curvilíneas ajustadas a la frontera; esta técnica combina las ventajas de los métodos de elemento finito y diferencias finitas (Jiménez *et al*, 1997)

En el Instituto de Ingeniería (II) de la Universidad Nacional Autónoma de México (UNAM), se ha trabajado en el desarrollo de modelos matemáticos en coordenadas curvilíneas, ajustadas a las fronteras en flujos a superficie libre. En 1996 se elaboró un modelo que incluye la determinación del volumen de control ocupado por el agua, que podría llamarse un modelo en malla adaptable al volumen de control. El modelo requiere la estimación de cotas del terreno así como de sus gradientes en las direcciones principales en cualquier punto del dominio. Para resolver esto último, se propuso representar la elevación como una función dependiente de dos variables y hacer su desarrollo usando una serie de Taylor, hasta los términos de tercer orden (Mendoza *et al*, 1996).

Al hacer el desarrollo en serie de Taylor de una función de dos variables hasta los términos de tercer orden, se tienen derivadas de grado k , $k-1$ y $k-2$, siendo k el orden del desarrollo (Arfken, 1980; Piskunov, 1978). Para obtener las derivadas, éstas se igualaron a coeficientes a_0, a_1, \dots, a_n , siendo $n = 8$ en este caso, que se calcularon minimizando la función $z = f(x, y)$ con un criterio de mínimos cuadrados y el sistema se resolvió usando un método de optimización de los llamados de gradiente conocido como DFP¹.

Se fijó un radio de búsqueda, similar al procedimiento seguido por el software comercial disponible para elaborar planos topográficos, como el SURFER por ejemplo (Golden Software Inc, 1999), para sólo tomar en cuenta un determinado número de datos alrededor del punto en el cual se quiere hacer la estimación; se recomienda hacer esto para evitar tomar en cuenta información que sólo provocaría errores en la estimación. Por ejemplo, si se tuviera que hacer la estimación del valor de la cota topográfica en un punto que está situado en una zona plana y sus vecinos al norte, al sur y al oeste también están en la misma

¹ Para la descripción del método DFP consulte el apéndice A del presente trabajo.

zona pero los vecinos del lado este están en un valle. La estimación del valor se verá afectada por todos los datos y muy probablemente se tendrán errores importantes. Una forma de evitarlo sería asignando un peso o costo a los valores dependiendo de su distancia al punto en el que se quiere hacer la estimación, así, los valores más alejados pesarían menos y tendrían menor influencia en el valor calculado para la cota topográfica.

Aquí sin embargo, se optó por definir un radio de búsqueda y hacer diversas pruebas hasta encontrar el número óptimo de vecinos que se deberían tomar en cuenta para el cálculo. Esto se ejemplifica en la Figura 4. En ella sólo se tomarán en cuenta los datos que están dentro del círculo para hacer la estimación del punto marcado en negro (al centro del círculo) del cual se conocen las coordenadas x-y. Los puntos que se toman en consideración son aquéllos que cumplen la restricción de que su distancia al punto de estimación es menor que una cierta cantidad.

El desempeño de la función propuesta se probó con un conjunto de 6000 puntos en los que se conoce el valor de las coordenadas (x_i, y_i) y el correspondiente valor de z_i . Estos son datos digitalizados de una zona del Río Colorado, ubicado en la vertiente hidrográfica del Pacífico, en la República Mexicana.

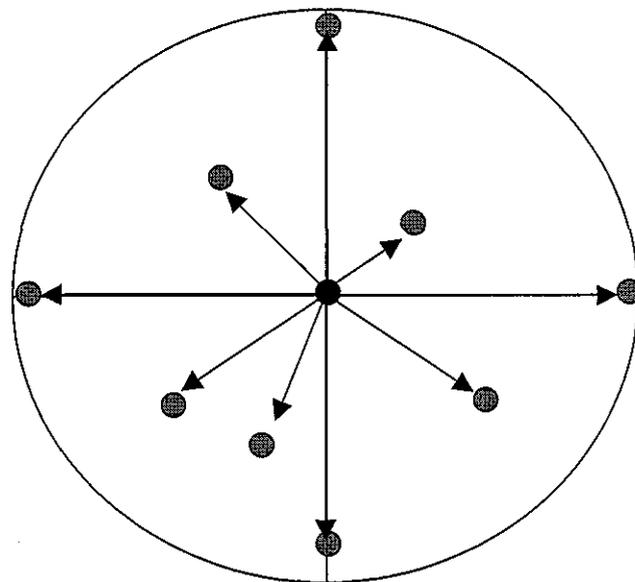


Fig 4. Radio de búsqueda

Se trabajó con un subconjunto (1000 puntos) del conjunto original de datos. Los resultados obtenidos se reportan en un informe interno del Instituto de Ingeniería (Mendoza *et al*, 1996) y serán contra los que se compare el desempeño del AG que se empleará en el presente estudio.

3.2 Descripción del Algoritmo Genético

Para desarrollar el trabajo se seleccionó como lenguaje de programación al C para codificar la operación de los AG. La secuencia que se siguió para aplicar los principios de los AG en la optimización de los nueve parámetros de la serie de Taylor se describe a continuación:

- El valor de la cota topográfica, z , se estimará como:

$$z(k) = a_0x + a_1y + \frac{1}{2}[a_2x^2 + a_3xy + a_4y^2] + \frac{1}{6}[a_5x^3 + a_6x^2y + a_7xy^2 + a_8y^3] \quad (1)$$

Siendo

k el punto de estimación

$a_0, a_1, a_2, \dots, a_8$ los coeficientes que optimizan el valor de la función z

(x, y) las coordenadas del punto de estimación

- Límite máximo de generaciones: 500, se hicieron varias pruebas y con este número se obtuvieron buenos resultados iniciales.
- Máximo número de puntos a evaluar: 100, para evitar que tanto el intérprete de matlab como el mismo programa hecho en lenguaje C tardaran demasiado en la ejecución.
- Se tomó una población de cien individuos ($i = 0, 1, 2, \dots, 99$), valor que se consideró adecuado como espacio de búsqueda.

- Longitud de la cadena para evaluar cada coeficiente: 8 bits, se decidió que era suficiente en términos de resolución.
- Probabilidad de cruzamiento: 0.95, valor recomendado en la literatura (Goldberg, 1989)
- Probabilidad de mutación: 0.09, valor tomado de la literatura (Goldberg, 1989)
- Probabilidad de reproducción: 0.05, valor sugerido en la literatura (Goldberg, 1989)
- Tolerancia: 3.0×10^{-5} , error que en un modelo topográfico se considera adecuado (Mendoza *et al*, 1996).
- La función que mide el desempeño de cada individuo i (función objetivo) es:

$$\text{función objetivo}(i) = \left[\sum_{j=1}^{NP} |z_{real}(j) - z_{calc}(j)| \right]$$

Siendo:

z_{real} el valor de la cota topográfica del punto que se evalúa

z_{calc} el valor calculado de la cota topográfica usando la ec. (1)

NP número total de puntos de evaluación

- Cada individuo consta de dos índices: el primero dice el número del individuo y el segundo representa la longitud total de una cadena de 72 bits con valores de unos y ceros que se generan de manera aleatoria. Cada subconjunto de 8 bits de esta cadena representan uno de los coeficientes de la ec. (1). Por ejemplo:

$P[i][j]$; $i = 1, \dots, N$ siendo N el número de pobladores definido; $j = 1, \dots, 72$; los primeros 8 valores de j (que deben ser ceros y unos) representan el valor del

coeficiente a_0 ; del 9 al 16 serán los correspondientes al coeficiente a_1 y así sucesivamente toman sus valores los restantes coeficientes.

- Se conserva siempre el 5% de la población; los individuos que pasarán sin alteración se eligen aplicando elitismo. Este valor también se tomó siguiendo las recomendaciones de la literatura consultada (Goldberg, 1989)
- El cruzamiento (en un sólo punto) de individuos se efectúa entre el individuo i y el $i+1$.
- La selección de los individuos se hizo con el método de la ruleta, por ser uno de los métodos más usados (Goldberg, 1989) y más fácilmente programables.
- A cada coeficiente de la ec. (1) se le asignó un rango mínimo y uno máximo, tratando de que éste abarcara de manera eficiente los valores que pudiera tener cada coeficiente. Para la asignación de valores de cada rango se tomó como base los resultados promedio presentados por los 9 coeficientes en el estudio del 96 (Mendoza *et al*, 1996)
- Se definió un radio de búsqueda tomando éste como la diferencia absoluta entre el valor de la coordenada (x_i, y_i) y la (x_{i+1}, y_{i+1}) .
- Si la distancia del punto que se está considerando es menor o igual que 3 veces el radio de búsqueda entonces el punto se toma en cuenta para hacer la estimación, en caso contrario se desecha. Esto tuvo como base también los resultados del estudio del 96, los cuales indicaron que si se tomaba un radio de búsqueda mayor no se obtenían buenas estimaciones (Mendoza *et al*, 1996)
- Si el número de puntos que caen dentro del radio de búsqueda excede a 15, entonces sólo se toman los primeros quince puntos encontrados. Esto viene del estudio del 96. Los autores (Mendoza *et al*, 1996) mostraron que sólo los

puntos cercanos al punto de interés aportan información buena para la estimación.

- Los valores digitalizados originales tienen valores para la coordenada x (en metros) que van desde 10000 a 13000; para y (en metros) desde 1900 a 2500 y para z (en msnm) de 36 a 17. Para manejar valores de error pequeños se dividió cada una de las coordenadas x-y-z por 10000. En el caso de AG, esto facilitó también un poco la definición de los rangos mínimo y máximo para cada coeficiente.

Los resultados obtenidos con AG se describen en el capítulo siguiente y se compara su desempeño con el método DFP.

4. ANÁLISIS DE DESEMPEÑO: ALGORITMO GENÉTICO SIMPLE

La zona que se digitalizó del río Colorado se muestra en su totalidad en forma tridimensional en la Figura 5. Como se mencionó anteriormente, del conjunto inicial de datos se eligió un subconjunto de mil puntos distribuidos en toda la zona de tal forma que se tuviera información tanto en zonas planas como en zonas con crestas y con valles. Tanto el conjunto original de datos como los elegidos se presentan en el apéndice C del presente trabajo.

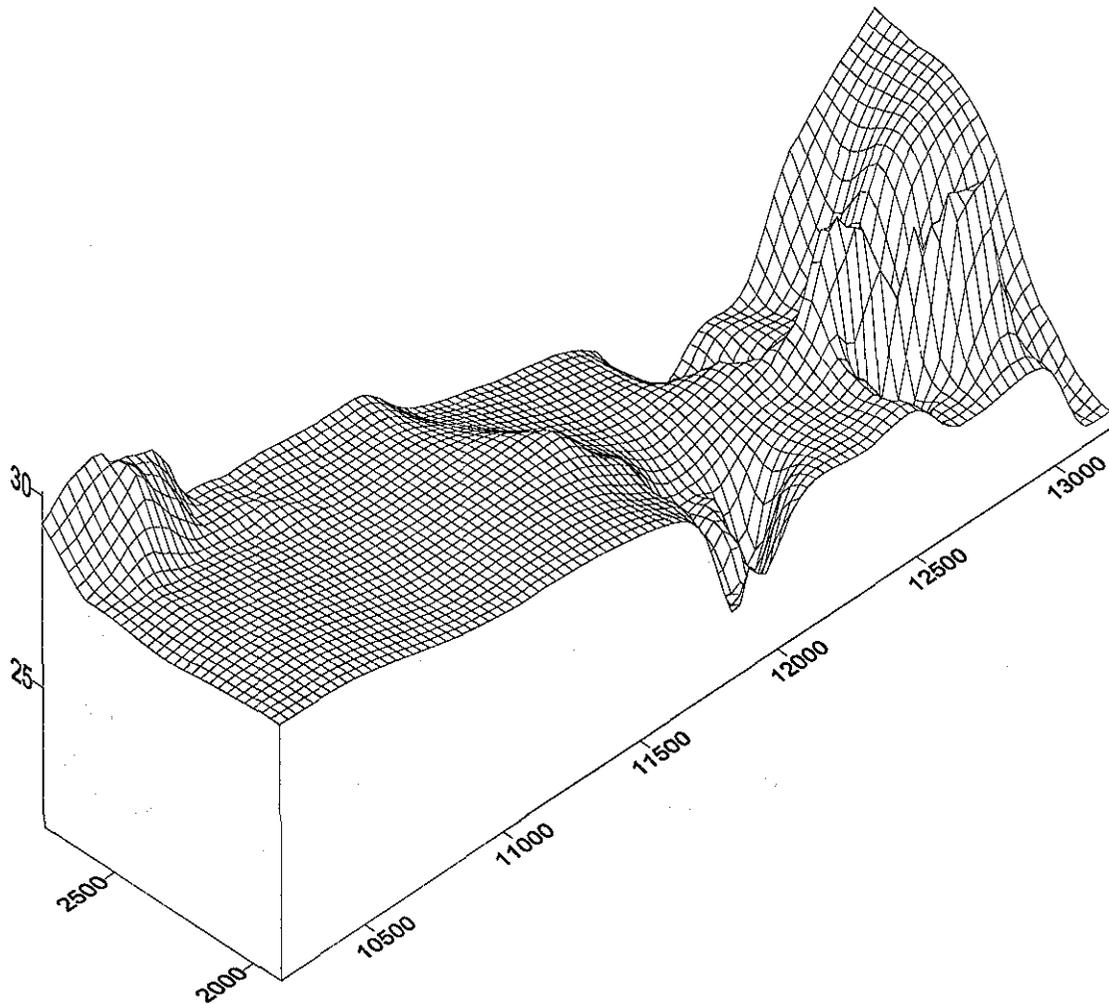


Fig 5. Datos digitalizados del río Colorado.

TESIS CON
FALLA DE ORIGEN

NOO 2222
UNIVERSIDAD DE ALABAMA

4.1 Algoritmo Genético Binario

Los dos programas de cómputo, el de los AG y el del estudio del 96 (llamado estima y elaborado y compilado con C ++), se corrieron usando la información topográfica almacenada en un archivo principal de datos y los resultados de los errores que se obtuvieron se muestran en la Figura 6. Al analizar los errores obtenidos, usando el archivo de datos principal que se presenta en el apéndice C del presente trabajo y contiene las cotas topográficas medidas. Se pudo apreciar que tanto el método DFP como el de AG tienen errores importantes en las zonas en las cuales el punto a estimar es un valle o una cresta y los puntos circundantes no auxilian a los métodos para poder obtener una buena aproximación al valor buscado.

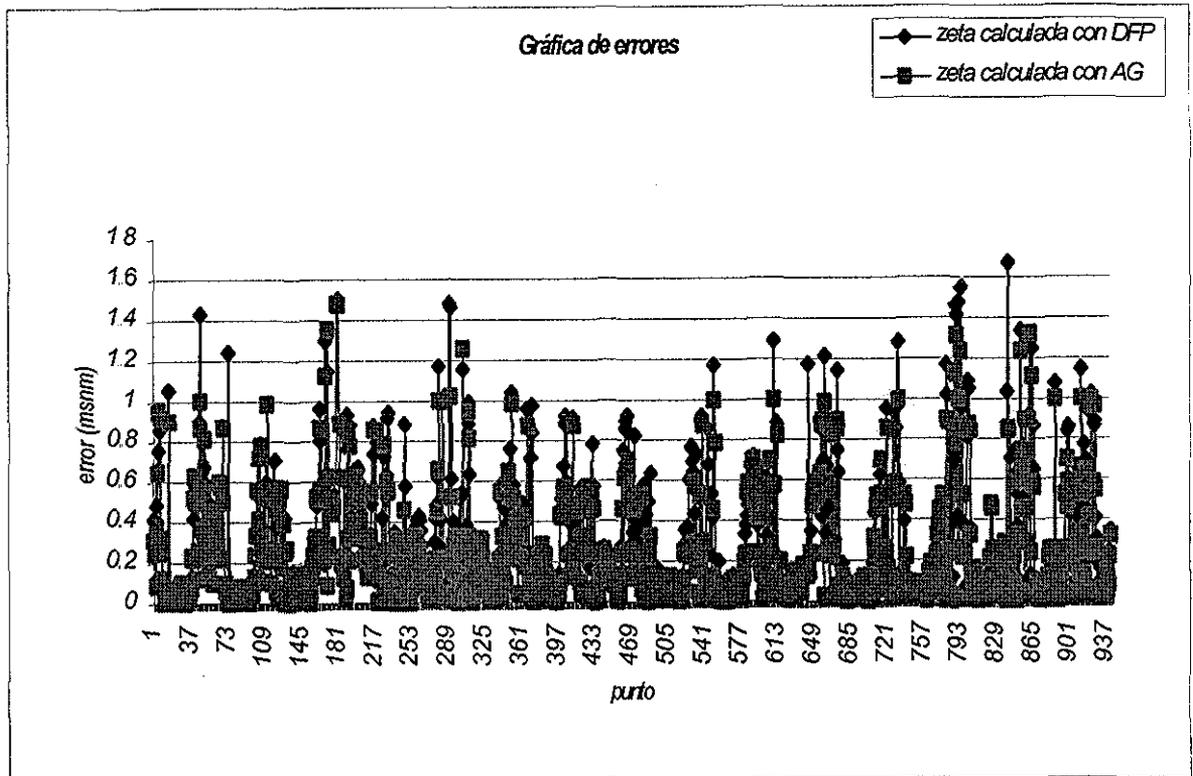


Fig 6. Errores obtenidos con AG y con DFP.

En ambos casos se probaron diversos valores para el radio de búsqueda pero no se logró obtener con ello una mejoría notable en los puntos difíciles de la zona.

TESIS CON
FALLA DE ORIGEN

Se puede decir, analizando la gráfica de la Figura 6, que el desempeño de los AG supera el del DFP pero no de manera significativa, ya que en general los errores obtenidos con los AG son menores. Para evidenciar esto último se seleccionaron cincuenta puntos del conjunto de mil (considerados los más difíciles para realizar la estimación) y se graficaron sus errores, los cuales se muestran en la Figura 7.

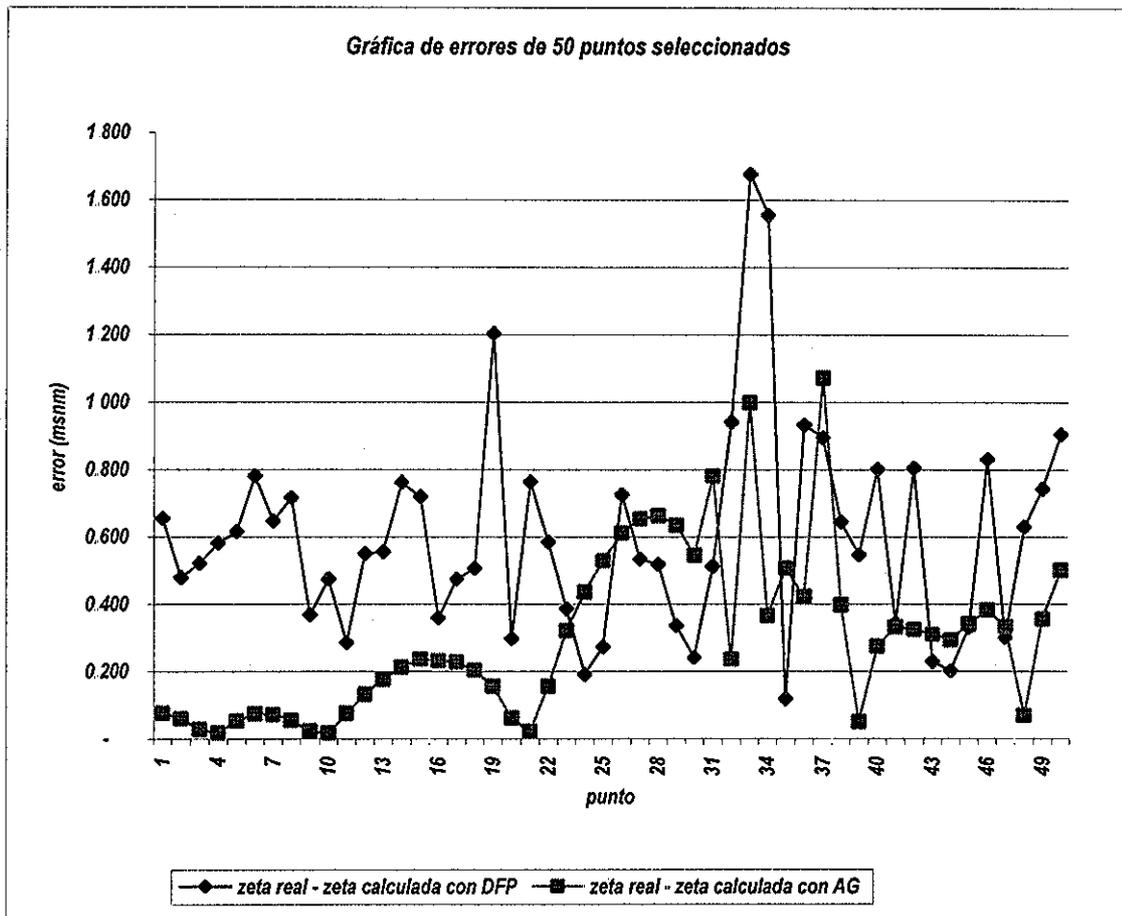
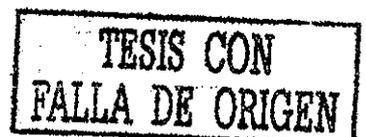


Fig 7. Gráfica de errores (crestas y valles).

4.1.1 Análisis de comportamiento de los coeficientes

Analizando los valores que se obtuvieron para los nueve coeficientes de la serie de Taylor, se puede decir lo siguiente:

- El coeficiente a_0 tuvo un valor promedio de 2.78, una desviación estándar de 1.57, un valor mínimo de -3.30 y un valor máximo de 7.32; se graficó el valor que el coeficiente presentó en cada uno de los puntos que se



evaluaron y el resultado se muestra en la Figura 8. Se puede observar que la mayor parte de los valores presentados se encuentran entre 1 y 4.

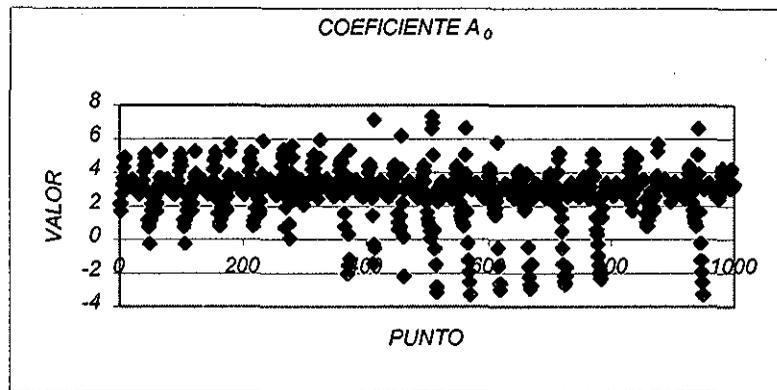


Fig 8. Valores del coeficiente a_0

- El coeficiente a_1 por su parte tuvo un valor promedio de 1.29, una desviación estándar de 0.59, un valor máximo de 2.04 y un mínimo de -2.62. Al igual que para el coeficiente a_0 , se graficó la evolución del comportamiento de su valor en todos los puntos. El resultado se muestra en la Figura 9. Tanto el coeficiente anterior como éste pertenecen a los términos lineales de la serie de Taylor. La mayor parte de los valores del parámetro se encuentran en el rango de 1 a 2.

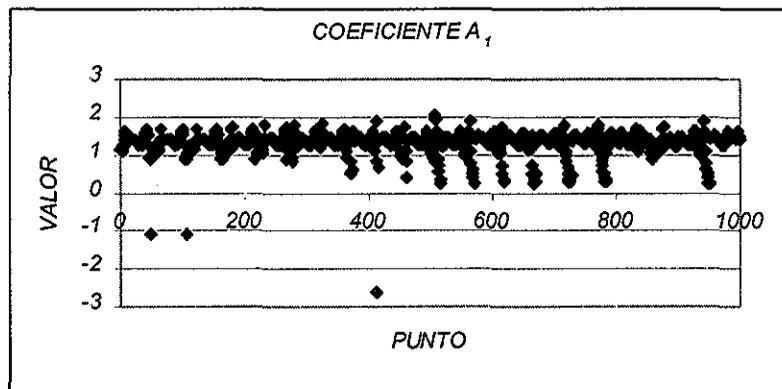


Fig 9. Valores del coeficiente a_1 .

- El coeficiente a_2 por su parte tuvo un valor promedio de 2.97, una desviación estándar de 2.08, un valor mínimo de -5.63 y un máximo de 9.82; y es el que presenta el máximo maximorum. Cabe destacar aquí que

TESIS CON
 FALLA DE ORIGEN

RECIBO
 DE LA
 BIBLIOTECA

el coeficiente a_2 multiplica al primero de los términos de orden dos de la serie de Taylor que se está evaluando. La gráfica de la Figura 10 muestra la evolución de los valores del coeficiente en los puntos de la zona en estudio. Las magnitudes del parámetro son en su mayoría positivas y casi equiparables en valor a las mostradas por el coeficiente a_0 .

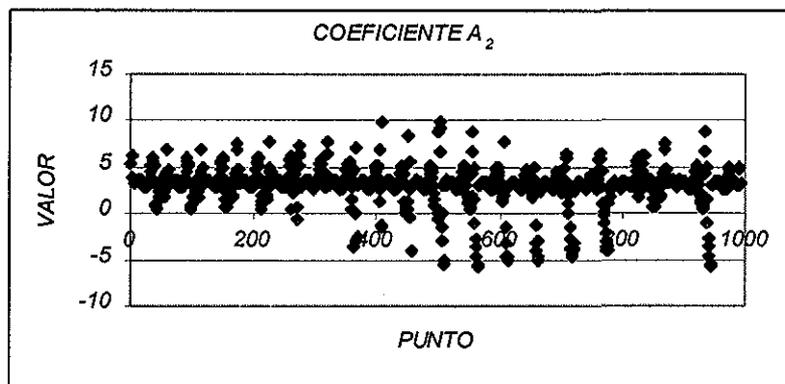


Fig 10. Valores del coeficiente a_2 .

- El coeficiente a_3 tuvo un valor promedio de 1.49, una desviación estándar de 1.40, su valor mínimo fue de -6.43 y su máximo de 3.69 ; la Figura 11 muestra el comportamiento de los valores presentados por el coeficiente, los cuales casi siempre se encuentran alrededor del valor 2.

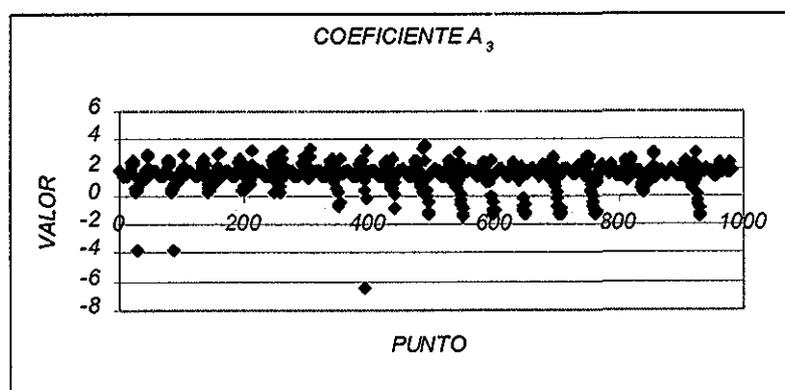


Fig 11. Valores del coeficiente a_3 .



- El coeficiente a_4 , perteneciente también a los términos de orden dos, mostró un valor promedio de 0.99, una desviación estándar de 0.89, un valor mínimo de -0.55 y un máximo de 1.60; la Figura 12 presenta el comportamiento del parámetro, observándose que sus valores casi siempre se encuentran muy cercanos a la unidad.

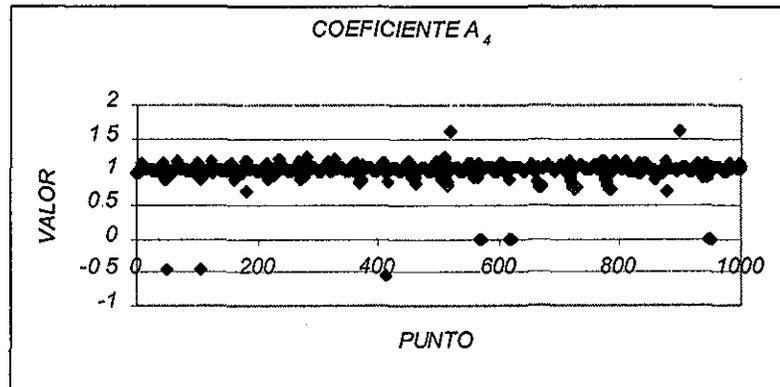


Fig 12. Valores del coeficiente a_4 .

- El coeficiente a_5 , primero de los términos de tercer orden, tuvo un valor promedio de -1.19, una desviación estándar de 1.16, un valor mínimo de -4.20 y un máximo de 2.69. Los valores obtenidos para el coeficiente se muestran en la Figura 13, en la cual se puede ver que una gran parte de ellos son valores negativos, concentrándose en el rango de -2 a -1.

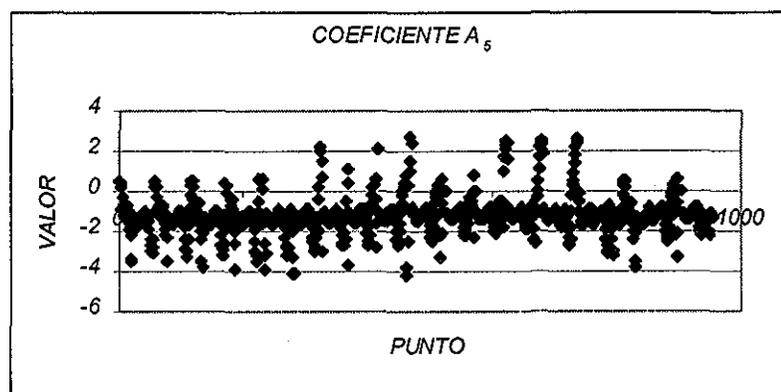


Fig 13. Valores del coeficiente a_5 .

TESIS CON
FALLA DE ORIGEN

- El coeficiente a_6 , involucra un término de orden dos y otro de orden uno, mostró un valor promedio de -0.80 una desviación estándar de 1.95 , un valor mínimo de -3.87 y un valor máximo de 1.96 . El comportamiento de los valores presentados se muestra en la Figura 14, apreciándose que la mayor parte de ellos se concentran en el rango de -2 a 0 .

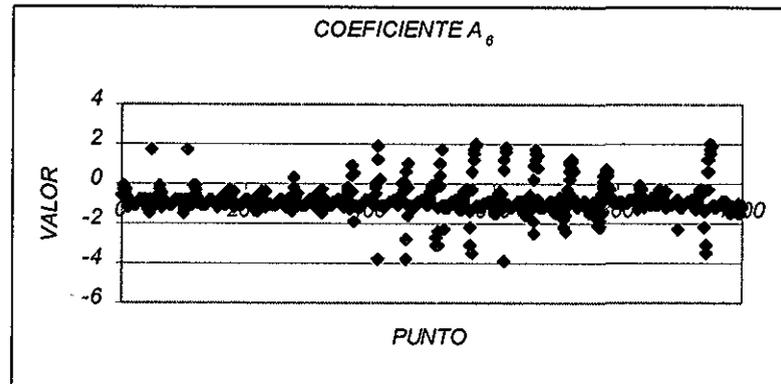


Fig 14. Valores para el coeficiente a_6 .

- El coeficiente a_7 , también involucra un término de orden dos y otro de orden lineal sólo que a diferencia del anterior, mostró un valor promedio de 0.37 , una desviación estándar de 3.82 , un valor mínimo de -2.45 y un valor máximo de 2.93 . La evolución de valores se muestra en la Figura 15. En ella se puede observar que este parámetro mantiene principalmente sus valores en el rango de 0 a 1 .

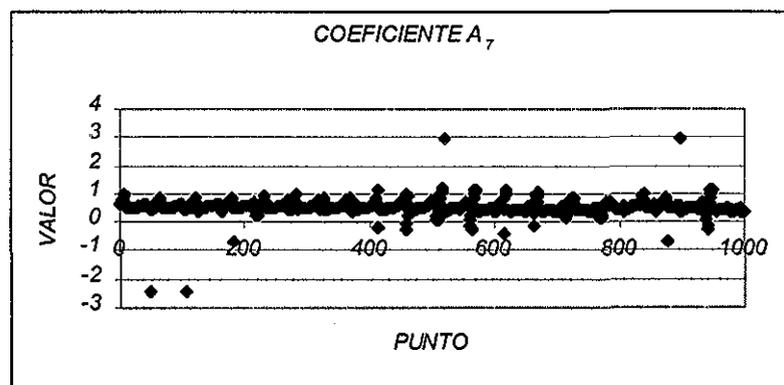


Fig 15. Valores para el coeficiente a_7 .

100 8221
VALORES EN ALABAMA

TESIS CON
FALLA DE ORIGEN

- El coeficiente a_8 , último de los términos de tercer orden, mostró un valor promedio de 0.93, una desviación estándar de 0.77, un valor mínimo de 0.42 y un máximo de 1.58; su comportamiento se muestra en la Figura 16. En ella se puede apreciar que es el que menor variabilidad presenta, casi siempre mantiene su valor cercano a la unidad.

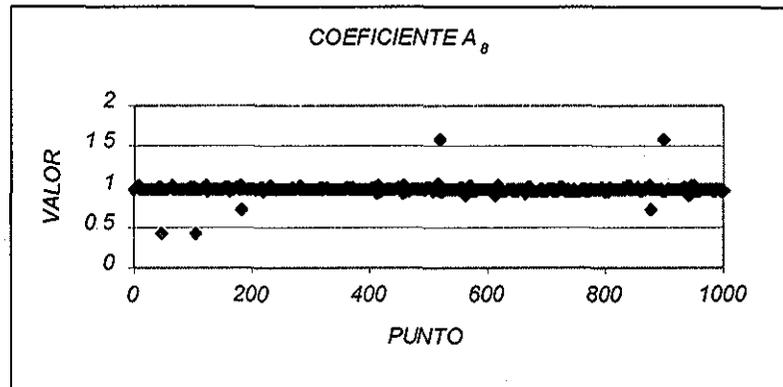


Fig 16. Valores para el coeficiente a_8 .

Dos cosas importantes se deben destacar del análisis hecho sobre el comportamiento de cada uno de los coeficientes de la serie:

- La primera es que hay coeficientes cuyo intervalo de valores es pequeño. Se probó a quitar estos coeficientes de la serie para evidenciar la importancia de cada uno de ellos. Hecho lo anterior sólo se puede decir que todos los coeficientes que están involucrados en la función son importantes y quitar alguno de ellos no llevó en ningún caso a resultados satisfactorios.
- La segunda es que las mayores desviaciones de los coeficientes corresponden siempre a los puntos difíciles de la topografía estudiada.

El estudio de Mendoza *et al* (1996), reporta sólo los valores promedio obtenidos con el método DFP para cada uno de los coeficientes de la serie de Taylor, siendo éstos muy parecidos a los obtenidos en el presente trabajo; ello se puede apreciar al analizar la comparación de valores con ambos métodos que se muestra en la Tabla 2.

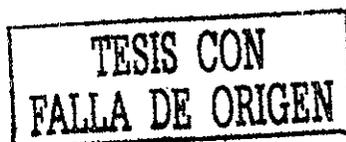
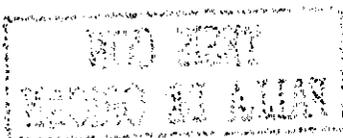


Tabla 2. Valores promedio de los coeficientes de la función.

<i>Valores promedio de los coeficientes de la serie de Taylor</i>	
Método DFP	AG(binario)
$\bar{a}_0 = 2.38$	$\bar{a}_0 = 2.78$
$\bar{a}_1 = 1.09$	$\bar{a}_1 = 1.29$
$\bar{a}_2 = 3.01$	$\bar{a}_2 = 2.97$
$\bar{a}_3 = 1.58$	$\bar{a}_3 = 1.49$
$\bar{a}_4 = 0.88$	$\bar{a}_4 = 0.99$
$\bar{a}_5 = -1.39$	$\bar{a}_5 = -1.19$
$\bar{a}_6 = -1.23$	$\bar{a}_6 = -0.80$
$\bar{a}_7 = 0.67$	$\bar{a}_7 = 0.37$
$\bar{a}_8 = 1.55$	$\bar{a}_8 = 0.93$

Si se analiza el valor promedio mostrado por los coeficientes de la serie se puede ver que los valores más altos corresponden a los primeros cuatro coeficientes. De ellos, los dos primeros y el cuarto corresponden a términos lineales, el tercero es el correspondiente al coeficiente a_2 , siendo el primer término de orden dos de la serie. Se podría pensar con ello que, proponer un polinomio que sólo contemple estos términos podría llevar a buenos resultados. Esto se hizo pero únicamente sirvió para poner en evidencia la importancia de los términos de tercer orden pues sin ellos los resultados que se obtuvieron se calificaron como pésimos.

De los coeficientes que pertenecen a los términos de tercer orden, los más relevantes son el que involucra el producto de un término cuadrático y uno lineal, coeficiente a_7 , y el a_8 que está relacionado con un término cúbico

4.2 Algoritmo Genético con representación de punto flotante (real)

Con el propósito de mejorar la estimación en los puntos en los cuales los AG con representación binaria no tiene un buen desempeño, se planteó la propuesta de definir a cada poblador con una representación real en vez de binaria, forma tradicional de los AG.

Se hicieron las modificaciones pertinentes al programa base, lo más importante a destacar es que ahora el cruzamiento se hace empleando una constante, cuyo rango de definición (Mühlenbein *et al*, 1993) se encuentra en el intervalo $[-0.25, 1.25]$ y se selecciona aleatoriamente.

Ejemplificando lo anterior se tiene que:

Sean p_1 y q_1 , los dos individuos que se recombinarán para obtener al nuevo individuo s_1 mediante:

$$s_1 = p_1 + \alpha(q_1 - p_1)$$

generalizando:

- Sea $p = (p_1, p_2, \dots, p_n)$, $q = (q_1, q_2, \dots, q_n)$ los individuos padre, los descendientes se obtendrán como:

$$s_i = p_i + \alpha(q_i - p_i) \quad i = 1, 2, \dots, n$$

De igual forma la mutación se hace sumando o restando al valor del poblador que se seleccionó para mutar, un valor definido entre cero y uno, llamado rango de mutación, con lo cual el nuevo valor es calculado de la siguiente forma:

- $s_i = p_i \pm \text{rango}_i \delta$

El signo positivo o negativo tienen igual probabilidad de selección (0.5) y la variable δ se calcula usando la siguiente serie:

- $\delta = \sum_{i=0}^{15} \alpha_i 2^{-i} \quad \alpha_i \in 0,1$

Basados en esta codificación de los parámetros a estimar, los resultados que se obtuvieron no mostraron una mejoría notable, en realidad acentúan un poco más tanto los aciertos como los errores importantes ya que al tener una representación real se mejoró la precisión de los valores de los coeficientes. Los resultados obtenidos con los AG en representación real mantuvieron un mejor desempeño en comparación a los logrados con el método DFP.

Se calculó el valor promedio de cada coeficiente, los cuales se muestran en la Tabla 3. Se observa que la tendencia mostrada anteriormente en la Tabla 2 se mantiene, es decir, los valores promedio calculados ahora con la representación real de los AG son muy parecidos a los obtenidos con los AG trabajados en su forma original (binaria), y ambos a su vez, son similares a los resultados mostrados por el método DFP.

Tabla 3. Valores promedio de los coeficientes de la función propuesta.

<i>Valores promedio de los coeficientes de la serie de Taylor</i>		
Método DFP	AG(binario)	AG(real)
$\bar{\alpha}_0 = 2.38$	$\bar{\alpha}_0 = 2.78$	$\bar{\alpha}_0 = 2.55$
$\bar{\alpha}_1 = 1.09$	$\bar{\alpha}_1 = 1.29$	$\bar{\alpha}_1 = 1.13$
$\bar{\alpha}_2 = 3.01$	$\bar{\alpha}_2 = 2.97$	$\bar{\alpha}_2 = 2.95$
$\bar{\alpha}_3 = 1.58$	$\bar{\alpha}_3 = 1.49$	$\bar{\alpha}_3 = 1.36$
$\bar{\alpha}_4 = 0.88$	$\bar{\alpha}_4 = 0.99$	$\bar{\alpha}_4 = 0.91$
$\bar{\alpha}_5 = -1.39$	$\bar{\alpha}_5 = -1.19$	$\bar{\alpha}_5 = -1.20$
$\bar{\alpha}_6 = -1.23$	$\bar{\alpha}_6 = -0.80$	$\bar{\alpha}_6 = -0.72$
$\bar{\alpha}_7 = 0.67$	$\bar{\alpha}_7 = 0.37$	$\bar{\alpha}_7 = 0.23$
$\bar{\alpha}_8 = 1.55$	$\bar{\alpha}_8 = 0.93$	$\bar{\alpha}_8 = 1.10$

Analizando el valor promedio de cada uno de los coeficientes de la serie de Taylor, mostrado para los AG tanto en representación binaria como en real y para el método DFP en la Tabla 3, se puede destacar lo siguiente:

- El coeficiente a_0 muestra el valor promedio más alto con los AG en representación binaria (2.78) y el más bajo (2.38) con el método DFP, siendo por consecuencia, la diferencia¹ entre estos dos valores la más alta. La diferencia menor es la que se obtiene entre el valor del AG en representación real y el método DFP (0.17). Para este coeficiente el método DFP y los AG en representación real obtienen un valor promedio muy similar.
- El comportamiento del coeficiente a_1 , es muy parecido al mostrado por el a_0 . La diferencia menor se obtiene entre el método DFP y el AG en representación real (0.04). Tanto el coeficiente a_0 como el a_1 pertenecen a los términos lineales de la serie analizada. Se realizó una prueba en la cual estos coeficientes no son considerados y los resultados que se obtuvieron pusieron en evidencia la importancia de estos dos términos. Se puede decir que son los de mayor peso en la serie.
- Para el coeficiente a_2 la diferencia mayor se encuentra entre el valor promedio obtenido con el DFP y el AG en representación real (0.06). Aquí, prácticamente no existe diferencia entre el valor calculado con el AG en representación binaria y el mostrado por el AG en representación real. La diferencia entre ambos es de 0.02; este coeficiente, al igual que los dos involucrados en los términos de primer orden, también es de los que muestran valores de peso en la serie.
- El método DFP y el AG en representación binaria son los métodos con los cuales se obtiene el valor promedio más parecido para el coeficiente a_3 . La diferencia entre ellos es de 0.09; la diferencia más grande se tiene entre el DFP y el AG en representación real (0.22).

¹ La diferencia se calcula tomando el valor absoluto de los valores involucrados.

- En el coeficiente a_4 , la diferencia entre el DFP y el AG en representación binaria es de 0.10; entre el DFP y el AG en representación real es de 0.03 y entre las dos representaciones del AG es de 0.08; esto indica que la menor diferencia en valores se tiene entre el DFP y el AG en su representación real y la mayor entre el DFP y el AG en su representación tradicional.
- Para el coeficiente a_5 , el AG en sus dos representaciones obtiene un valor muy parecido, la diferencia entre ellos es de sólo 0.01; si se toma el valor absoluto de los valores promedio, se tiene que el método DFP es el que calcula el valor más alto para este coeficiente.
- Tanto el coeficiente a_6 como el a_8 , ambos involucrados en los términos de tercer orden de la serie, son los que presentan las diferencias más altas. Para el primero se tiene que la diferencia entre el método DFP y el AG en su representación tradicional es de 0.43 y de 0.51 con el AG en representación binaria. El segundo, muestra 0.62 de diferencia entre el DFP y el AG en binario y, de 0.45 entre el DFP y el AG usando representación con números reales.
- Para el coeficiente a_7 , el AG en sus dos representaciones obtiene un valor muy semejante. La diferencia entre ellos es de 0.17; la mayor discrepancia se tiene entre el valor calculado con el método DFP y el AG en representación real (0.44).

El comportamiento de los nueve coeficientes se muestra gráficamente en la Figura 17, en ella se puede apreciar lo analizado anteriormente para cada uno de ellos.



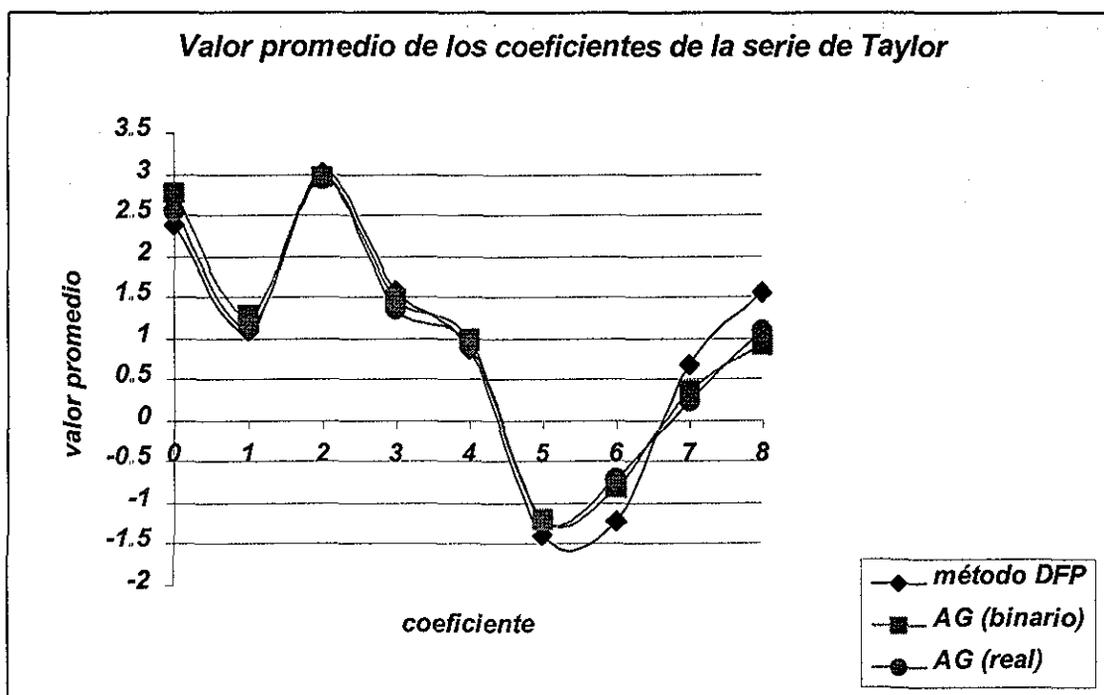


Fig 17. Valor promedio de los coeficientes.

Una vez hecho el análisis de los resultados que se han obtenido tanto con los AG en representación binaria como en real y su respectiva comparación con el método de optimización conocido como DFP, se tiene como conclusión principal que la función propuesta, un desarrollo en serie de Taylor para una función de dos variables hasta los términos de tercer orden, no es la más adecuada para hacer la estimación del valor de la variable z ya que en las gráficas de error mostradas se puede ver que hay errores por arriba del metro, magnitudes que un modelo del terreno aceptable no puede tener. El objetivo principal que se persigue es el de tener un método que logre estimaciones del valor de la cota topográfica con un error inferior a los 5 cm, error que sería aceptable en el modelo que se propuso en el estudio del 96 (Mendoza *et al*, 1996).

Al ser uno de los objetivos del presente trabajo el tener una forma confiable para realizar tanto estimaciones de valores conocidos como la interpolación o extrapolación de ellos, conviene entonces, explorar la subrama de los AG, la Programación Genética PG, para encontrar la forma de la función que mejor se adapte al comportamiento de los datos digitalizados y con ello, reducir la

incertidumbre en la predicción de valores. Esto último se analizará en el siguiente capítulo.

5. PROGRAMACIÓN GENÉTICA

5.1 Antecedentes

Como se mencionó brevemente al inicio del presente trabajo, la programación genética (PG) es una rama de los AG. El paradigma de la PG fue introducido por Koza a finales de los 80's y trata de encontrar el programa de computadora que resuelve un determinado problema efectuándose el proceso de búsqueda sobre el espacio de posibles programas. Esto último constituye la principal diferencia entre la PG y los AG, dicho en otras palabras, la distinción principal entre ellas radica en la representación de la solución.

En la PG, la estructura que se adapta es la formada por una población de programas estructurados en forma jerárquica. Dichos programas, que son de formas y tamaños diferentes, pueden ir cambiando dinámicamente a lo largo del proceso.

El esquema básico del algoritmo de la PG es el siguiente:

1. Se genera la población inicial usando de forma aleatoria el conjunto de funciones y terminales definido³.
2. Se ejecuta cada programa en la población y se asigna la función de evaluación (fitness value) para evaluar el desempeño.
3. Se crea una nueva población de programas considerando:
 - copiar los mejores programas existentes.
 - crear nuevos programas usando el operador mutación.
 - crear nuevos programas usando el operador cruzamiento.
4. El mejor programa que aparece en cualquier generación, que es la mejor solución posible, es designado como el resultado de la PG.

³ En el punto 5.2 del presente capítulo se definirán formalmente los conceptos básicos de la PG.

Los programas de computadora en la generación inicial tienen generalmente un valor muy pobre de *fitness*. Sin embargo, con el paso de las generaciones, algunos de los pobladores cambiarán este valor haciendo uso de los principios NeoDarwinianos de cruzamiento y supervivencia del más apto, creando con ello nuevas generaciones de individuos.

La forma de crear nuevos individuos, programas de computadora en este caso, es mediante el uso de los operadores de reproducción, cruzamiento y mutación. El primero de ellos requiere seleccionar, usando el *fitness*, un programa de computadora de la población actual y transmitirlo íntegramente a la siguiente generación. El segundo de los operadores, es usado para crear nuevos vástagos a partir de dos programas de cómputo padres. Generalmente, los programas padres son de diferente tamaño y forma. Los programas hijo están compuestos básicamente de sub-expresiones de sus padres: subárboles, subprogramas, subrutinas, bloques de construcción, etc , etc. Finalmente, el operador de mutación se usa para cambiar en forma aleatoria una o varias partes de un programa de computadora.

Después de que se realizan los cambios en la población usando los operadores genéticos, la nueva generación reemplaza a la población antigua. Cada individuo en la nueva población de programas de computadora es evaluado usando el *fitness* y este proceso se repetirá a través de las siguientes generaciones hasta encontrar el fin del ciclo.

Lo descrito anteriormente constituye básicamente la forma de operación de la programación genética y ha encontrado aplicación en diversos tipos de problemas. Se ha aplicado exitosamente en casos para los cuales no existe una solución única, por ejemplo: un programa que maneje un carro. También PG ha sido útil para encontrar soluciones donde las variables están en constante cambio, por ejemplo para el caso del carro, PG encontrará una solución si la variable que define el tipo de pavimento es concreto suave y una totalmente diferente si ahora se trata de un camino sin pavimento.

También se puede mencionar dentro de las aplicaciones de la PG el modelado e identificación de sistemas, procesamiento de señales e imágenes, diseño de circuitos electrónicos, control y robótica, entre otras (Rodríguez, 2000).

5.2 Conceptos básicos de la Programación Genética

Conviene definir algunos de los términos que se usan en el ambiente de la PG (Banzhaf *et al*, 1998):

- El conjunto de **funciones** y el conjunto de **terminales** son los elementos primitivos con los cuales se construye un programa en PG.
- El conjunto de funciones (*function set*): está compuesto por las expresiones, los operadores y las funciones disponibles en el ambiente de la PG. Algunos ejemplos de funciones elegibles son:
 - Funciones Booleanas: and, or, not, xor.
 - Funciones aritméticas: +, -, x, /.
 - Funciones trascendentales: trigonométricas, logarítmicas.
 - Subrutinas.
 - Ciclos: while... do, repeat... until, for, do.
 - Expresiones de transferencia de control: go to, call, jump.
 - Expresiones condicionales: if, then, else, case, switch.
- El conjunto de terminales (*terminal set*) son aquellos elementos con los cuales termina una rama del árbol, por ejemplo: variables que sean definidas como constantes para el proceso.
- La aridad (*arity*) de una función es el número de entradas o de argumentos que toma esa función, por ejemplo la *aridad* de la división es 2 y la del seno es 1.
- Notación infija y prefija: La forma más común de escribir una función con dos argumentos, por ejemplo la suma, es la notación infija. Esto es, los

dos argumentos están conectados o relacionados con el símbolo de la operación que se realiza entre ellos dos: $a+b$. Un método diferente es la llamada notación prefija o polaca. Aquí, el símbolo de la operación es escrito primero seguido de los argumentos requeridos: $+ a b$. Esta notación será la utilizada para evaluar la función que se obtenga como resultado de la aplicación de la PG en la estimación de cotas topográficas.

- Árboles y subárboles: En este contexto un árbol es una estructura jerárquica constituida por nodos y hojas (ver Figura 18). El árbol más simple consiste de una sola hoja. Los nodos son puntos internos y pueden tener más nodos o estar sólo conectados por hojas. El nodo más alto se llama raíz del árbol. Al ser los árboles estructuras dinámicas de información se adaptan perfectamente para el manejo de las poblaciones en la PG.

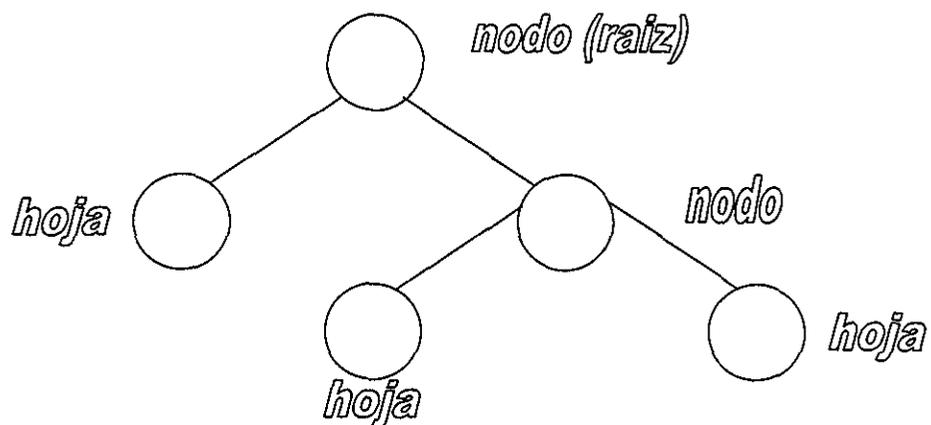


Fig 18. Ejemplo de una estructura jerárquica (árbol).

La estructura que se desarrolla en la interpretación de un programa de computadora se denomina *parse tree*. Las funciones se consideran como nodos y sus argumentos como hojas. Existe una relación muy directa entre estos árboles y las expresiones en notación prefija o infija. De hecho, los árboles son sólo otra forma de escribir las expresiones, dependiendo de la notación que se esté utilizando, por ejemplo:

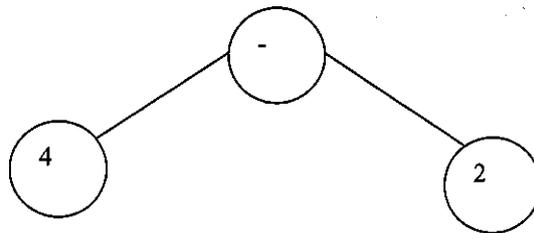
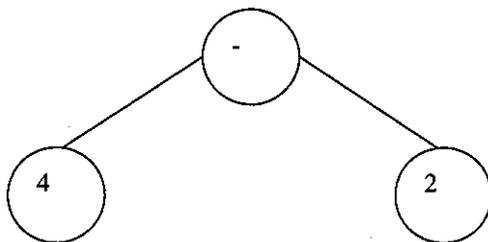


Fig 19. (4 - 2) Notación infija. (-4 2) Notación prefija.

Un subárbol (ver Figura 20) es una parte del árbol que está debajo de un nodo interno de ese árbol. Si el árbol se corta en ese punto el nodo interno se convierte en nodo raíz, por ejemplo:



Es un subárbol válido de:

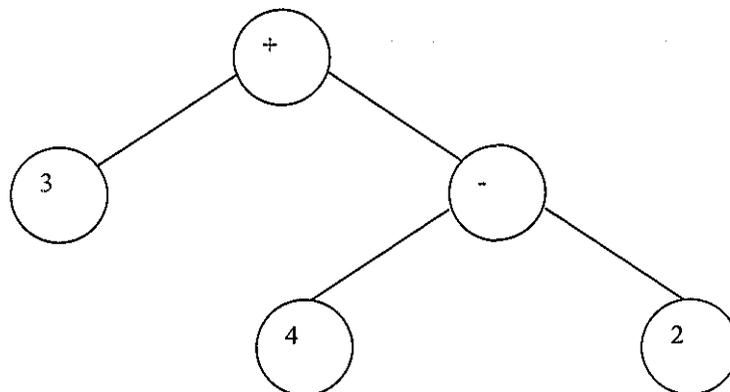


Fig 20. Ejemplo de un subárbol

Una forma de medir la complejidad de un árbol es mediante la profundidad, expresada ésta como el máximo número de nodos padre que una hoja puede

tener. Otra es usando el tamaño completo del árbol; esto es, la suma total de hojas y nodos que el árbol tiene.

5.2.1 Operadores Genéticos.

Anteriormente se mencionó que los operadores de los que hace uso la PG para la modificación de las estructuras de los pobladores son: el cruzamiento, la copia o reproducción y la mutación. Ejemplos de cómo opera cada uno de ellos se dan a continuación.

- Operador de Cruza

El operador de cruza trabaja de la siguiente manera:

- ❖ Dados dos árboles padres (Figura 21 a y 21 b), se eligen aleatoriamente un nodo en cada árbol padre (nodos de cruza). Los nodos así seleccionados constituirán la raíz del subárbol asociado. Posteriormente los subárboles son intercambiados entre los padres produciendo así dos nuevos individuos (figura 22 a y 22 b, respectivamente), los cuales son generalmente diferentes en tamaño y forma a los padres.

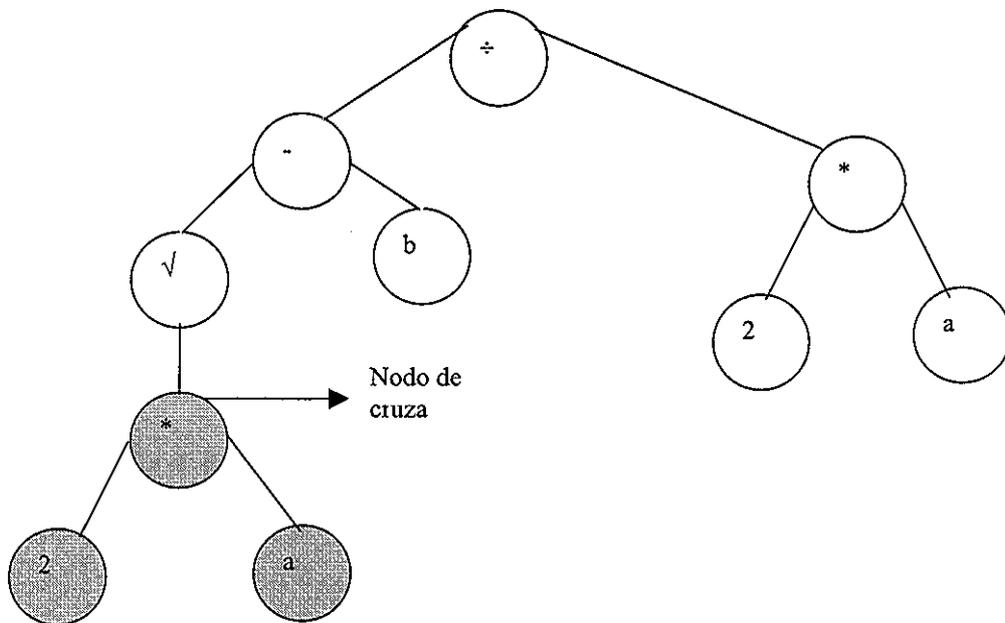


Fig 21 (a). Padre 1.

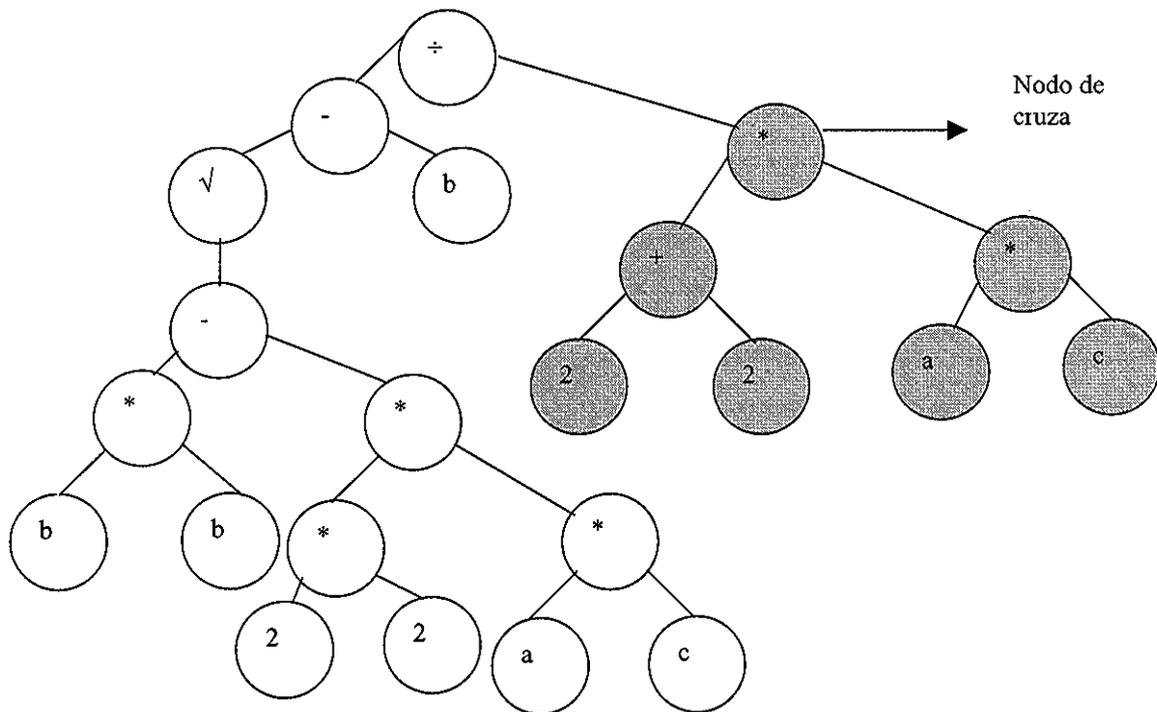


Fig 21 (b). Padre 2.

TESIS CON
FALLA DE ORIGEN

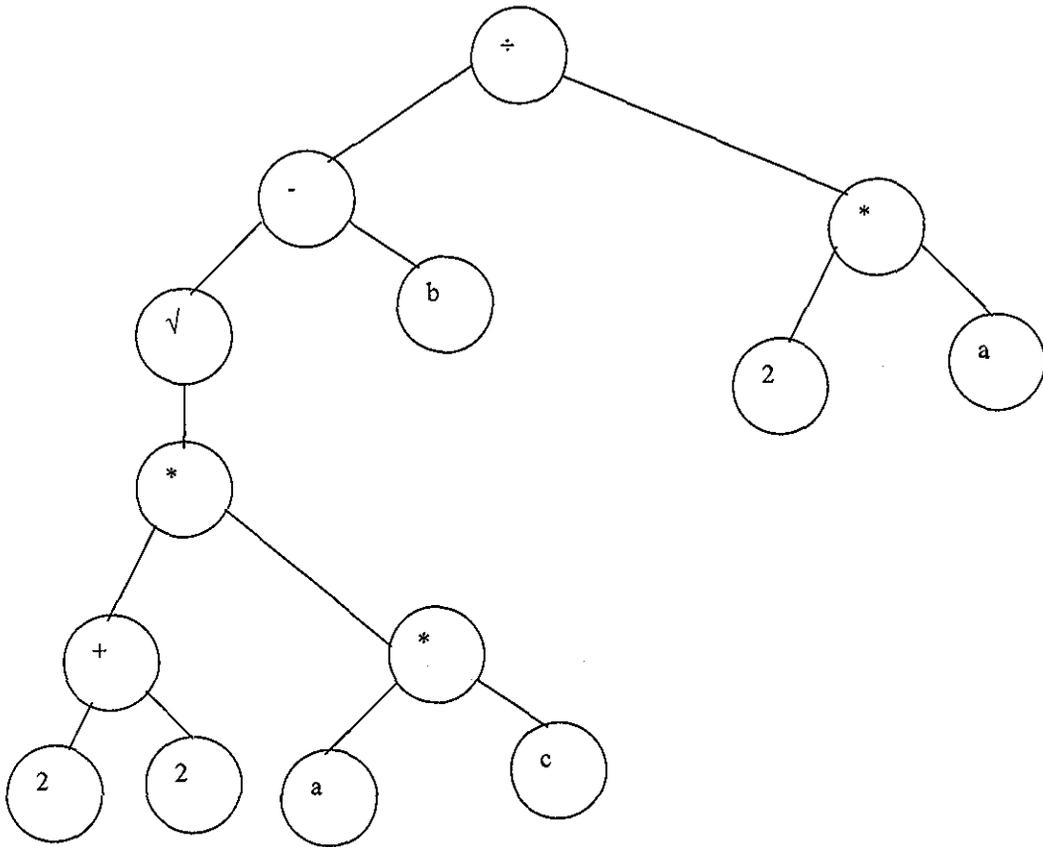
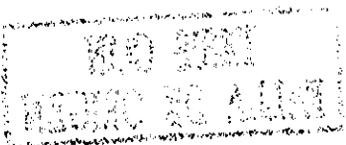


Fig 22 (a). Primer descendiente generado (operador de cruza).



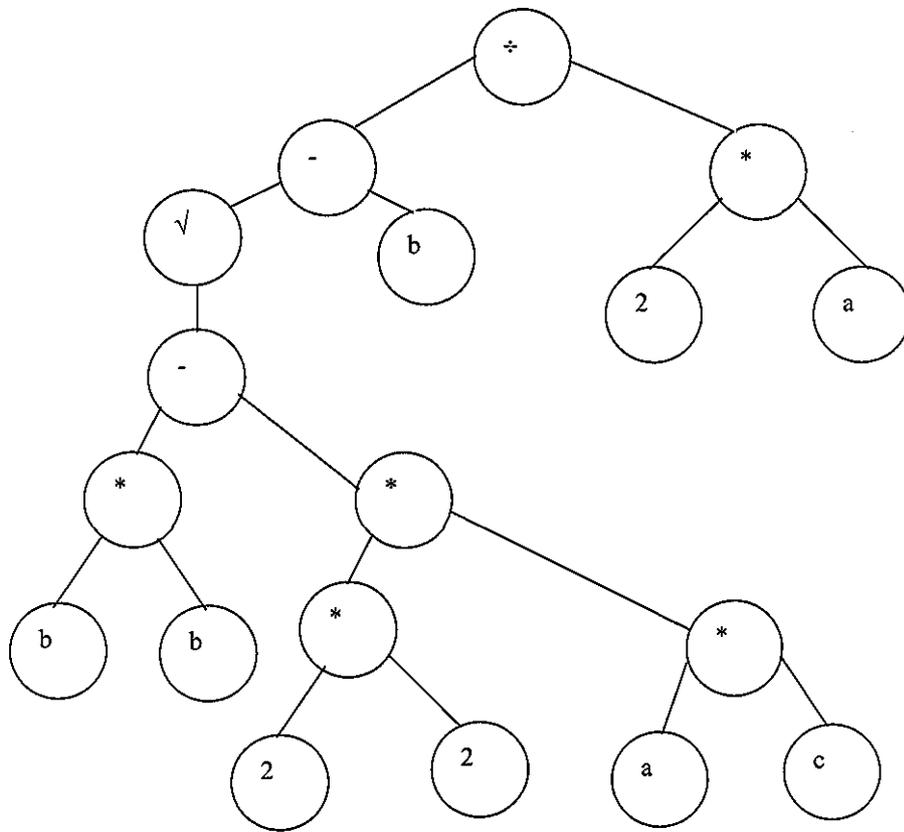
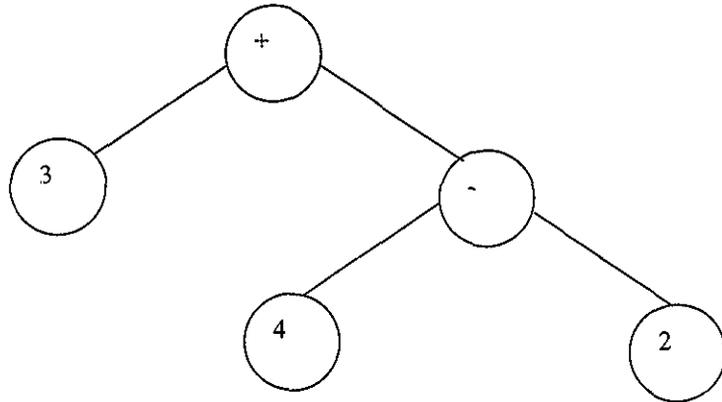


Fig 22 (b). Segundo descendiente generado (operador de cruza)

- Operador de Reproducción.

Este operador selecciona un árbol padre y lo pasa sin alteración a la siguiente generación (Figura 23).

Generación 1:



Generación 2:

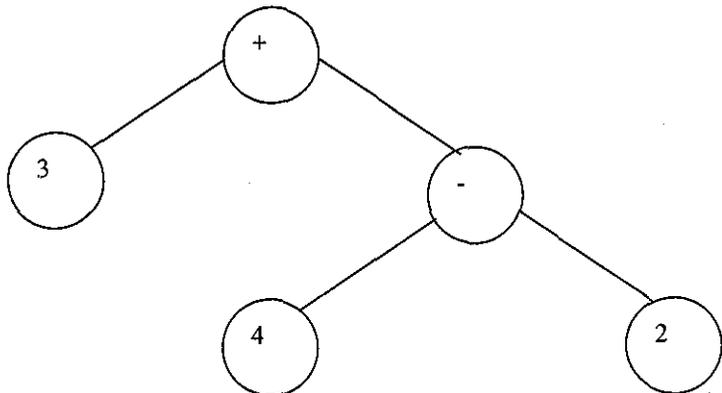
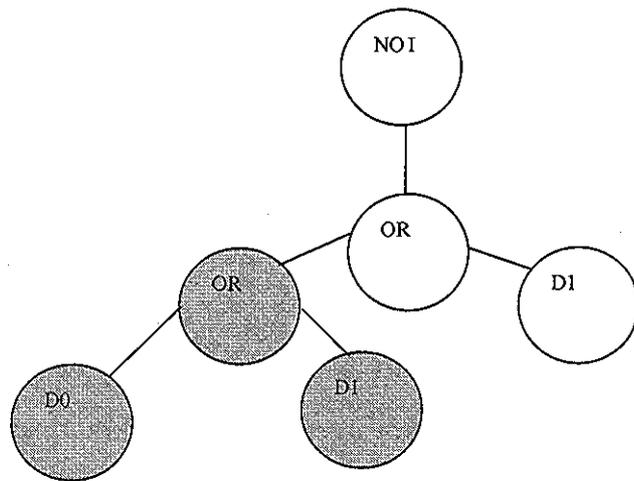


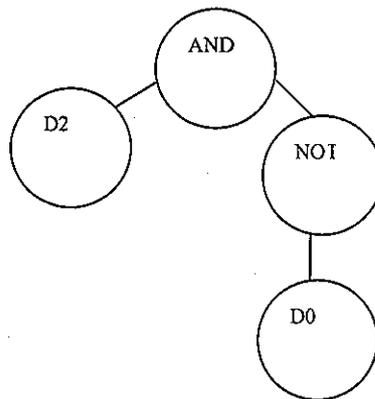
Fig 23. Ejemplo del funcionamiento del operador de reproducción.

- Operador de Mutación

La mutación es un cambio aleatorio en el material genético de un individuo. Esto se hace borrando un subárbol aleatoriamente seleccionado y reemplazándolo con uno nuevo que se crea. Este proceso se ejemplifica en la Figura 24.



Subárbol nuevo creado:



EL individuo mutado será:

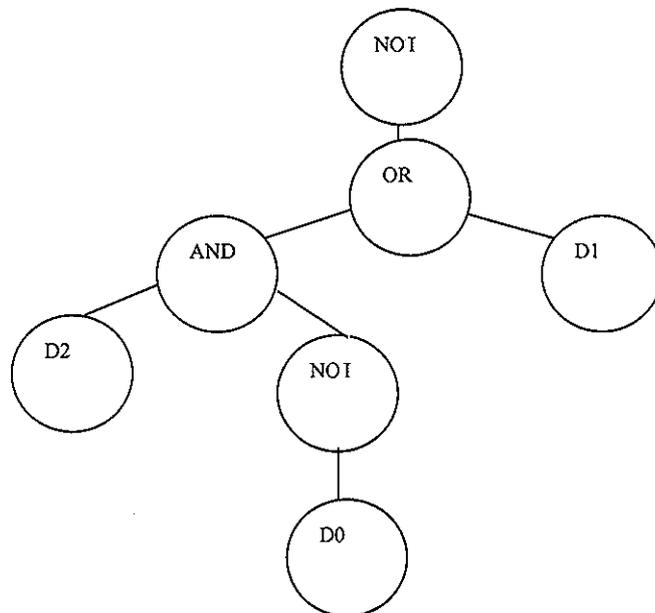


Fig 24. Ejemplo del funcionamiento del operador de mutación.

5.2.2 Métodos de selección

Los métodos de selección de los individuos que pasan a la siguiente generación son los mismos que los utilizados por los AG. En el caso de aplicación de los AG se usó el método de la ruleta combinado con elitismo. Para la PG se optó por elegir el método de la ruleta combinado con un reajuste de *fitness* llamado ranking (Banzhaf *et al*, 1998), que es un escalador que ajusta los valores de *fitness* de los individuos. La combinación de estos métodos de selección se hizo con el objetivo de garantizar que siempre los individuos más aptos sean los que pasen de una generación a otra.

La selección usando el ranking está basada en el ordenamiento del *fitness* de la población (*fitness order*), dentro del cual los individuos podrán ser clasificados. La probabilidad de selección se asigna a los pobladores como una función de su orden en la población. Principalmente se usan órdenes lineales y exponenciales.

Para un orden lineal la probabilidad es una función lineal del rango (orden):

$$p_i = \frac{1}{N} \left[p^- + (p^* - p^-) \frac{i-1}{N-1} \right]$$

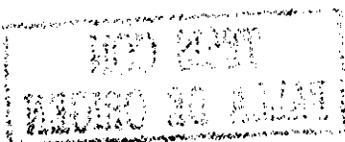
donde $\frac{p^-}{N}$ es la probabilidad de que sea seleccionado el peor individuo y

$\frac{p^*}{N}$ representa la probabilidad de que sea seleccionado el mejor individuo.

Si el orden clasificación es exponencial, entonces, la probabilidad puede ser calculada usando un sesgo de selección constante c :

$$p_i = \frac{c-1}{c^{N-1}} c^N - i \text{ siendo } 0 < c < 1$$

La selección es la responsable de la velocidad de evolución y es a menudo señalada como la culpable en casos en los cuales se tiene una convergencia prematura que conduce a un estancamiento del algoritmo evolutivo.

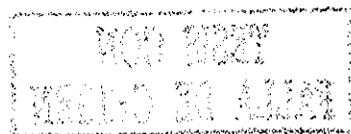


5.3 Aplicación de la Programación Genética en la estimación de cotas topográficas

Se usaron como base para la aplicación de la programación genética las rutinas de la caja de herramientas de genéticos del intérprete matlab (The MathWorks, 1999) y se hicieron en ellas los cambios pertinentes para el análisis y resolución del problema en estudio. Los cambios principales se refieren al conjunto de funciones y al de terminales con los cuales la PG puede jugar para la combinación de los términos en los que quedará expresada la función que se busca.

Se graficó, en dos (Figura 25) y en tres dimensiones (Figura 5, mostrada al inicio del capítulo anterior), el conjunto de datos con el fin de seleccionar así algunas de las zonas en las que se pudiera evaluar la bondad de la función que PG arroje

Se presenta, en la Figura 25, las curvas topográficas de la zona en estudio; en ella se puede apreciar que la topografía es compleja y que tratar de encontrar una expresión universal que pueda servir para estimar la variable z en cualquier punto dadas las coordenadas x - y será difícil. Se optó por seleccionar un subconjunto de mil puntos (los mismos con los que se trabajó en el método DFP y en los AG, capítulo 4 del presente trabajo), y de ellos tomar dos regiones para buscar una función que considere para la estimación sólo los 15 puntos más cercanos a ella. Esto es, al igual que en AG y con el método DFP se definió un radio de búsqueda poniendo como condición que la distancia existente entre los puntos de la región en el cual se quiera hacer la estimación y los puntos aledaños sea menor que una distancia dada (50 metros, en este caso) y si el número de puntos que cumplen con la condición es mayor a 15 sólo se toman los primeros 15 y el resto se elimina.



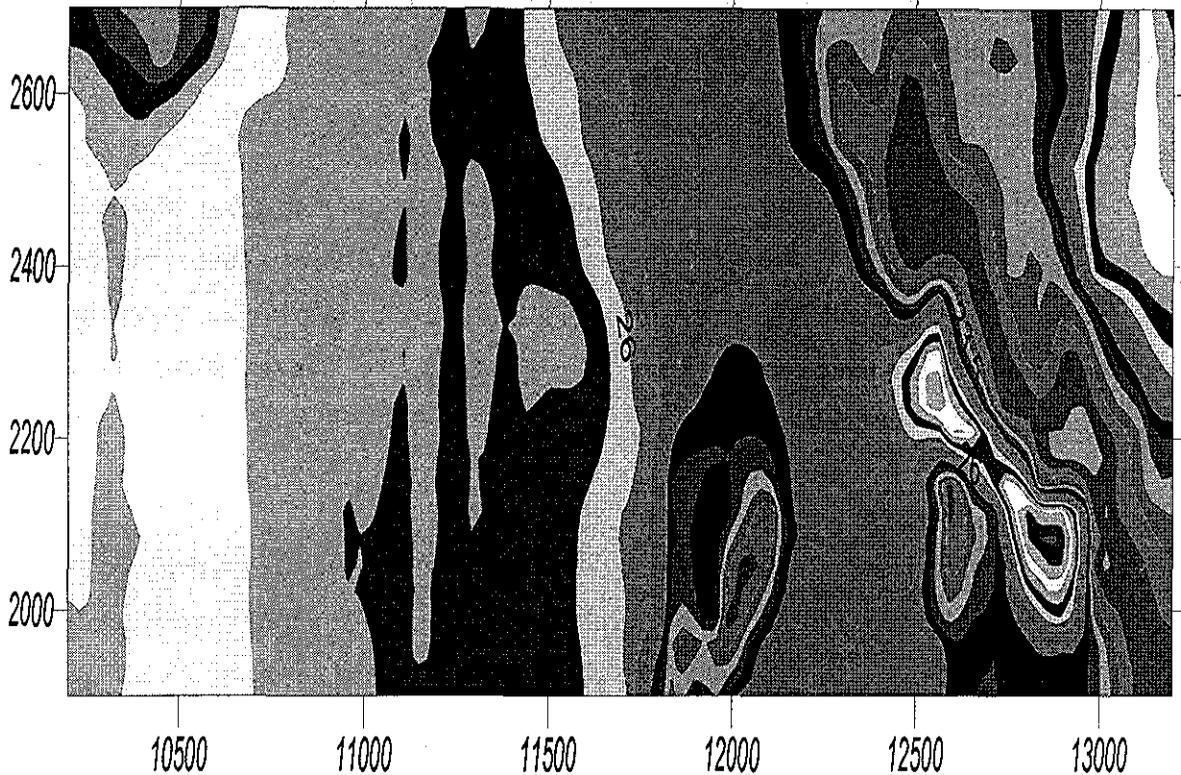


Fig 25. Topografía de la zona en estudio.

Hecho lo anterior, se pensó que como primer intento se probaran funciones de tipo aritmético y trigonométricas. Para las primeras se usaron los cuatro operadores básicos: suma, resta, multiplicación y división; para las trigonométricas sólo se definieron el seno y el coseno. El conjunto de terminales incluyó un rango amplio de números reales para poder así encontrar las constantes que las expresiones requieran.

Se definieron los siguientes parámetros en el cuerpo principal del programa de cómputo elaborado:

$N_{ind} = 100$, número de individuos de la población. Se definieron 100 pobladores para evitar que al tener una población muy grande la ejecución del programa en el intérprete matlab fuera muy lenta.

$MaxGen = 2000$, máximo número de generaciones. Definido con una serie de pruebas efectuadas.

**TESIS CON
FALLA DE ORIGEN**

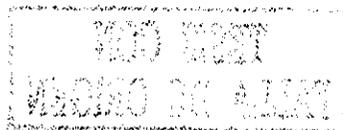
MaxNumNodes = 128, máximo número de nodos. Complejidad máxima de la solución buscada, definida también con diversas pruebas hechas.

Pcross = 0.90, probabilidad de cruzamiento, valor tomado de la literatura consultada (Goldberg, 1989).

Pmut = 0.05, probabilidad de mutación, valor tomado de la literatura consultada (Goldberg, 1989).

5.3.1 Optimización regional

Del subconjunto de mil puntos analizado se seleccionaron 2 regiones dentro de la topografía que incluyeran partes planas y partes con crestas y/o valles. Se hicieron varias corridas y se vió que con 1000 generaciones y 16 o 32 nodos se obtenían buenas estimaciones. La primera región seleccionada se muestra en la Figura 26; el rango de valores de la coordenada x va de los 13000 m a los 13800 m, la coordenada y se mantiene en los 2700 m y los 2750 m. La variable z tiene un valle en esta zona de aproximadamente 19 msnm y el resto de los valores se mantiene entre los 29 y 30 msnm.



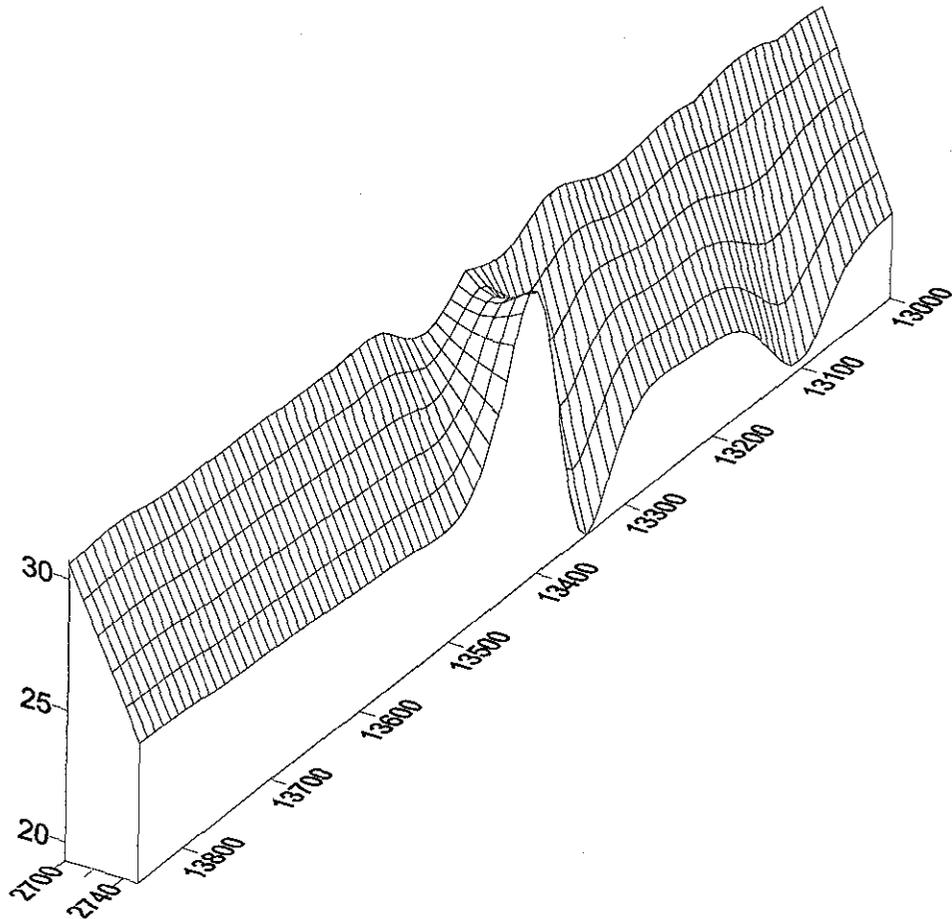


Fig 26. Zona 1.

Los resultados que se encontraron para esta zona se muestran en la Figura 27, en ella se puede ver que a partir de la generación 250, el error del mejor individuo se mantiene prácticamente constante y que dicho error es pequeño. La función que PG encuentra para estimar la variable z es una expresión simple que combina las funciones y terminales definidos, para lograr que en cada punto de coordenadas x - y se evalúe la variable z con excelentes resultados.

**TESIS CON
FALLA DE ORIGEN**

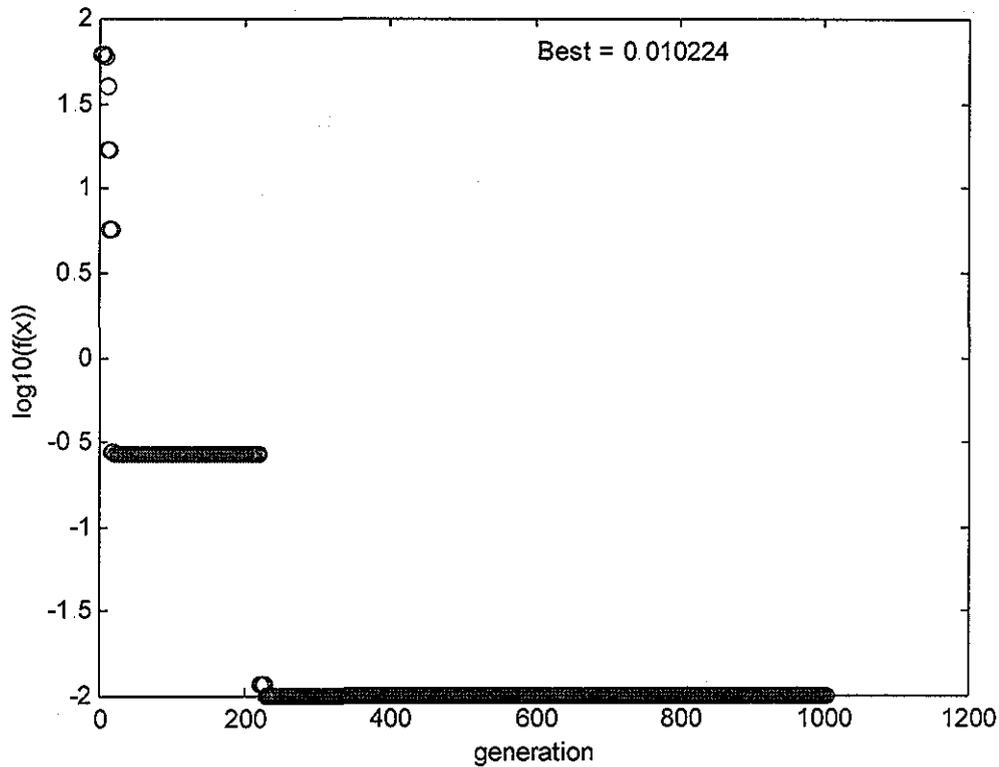


Fig 27. Optimización regional, zona 1.

$$z(i) = cte[y(i) + [(x(i) + 0.6)(\cos(y(i) + 0.6))]] i = 1, 15; cte = 00514287$$

con ella se calculó el valor de la variable z para los puntos de coordenadas x-y mostradas en el apéndice D, sección D.1.1 del presente trabajo. Los resultados se muestran en la Tabla 4.

TESIS CON
FALLA DE ORIGEN

Tabla 4. Resultados obtenidos en la región 1.

Punto	Z real	Z calculada con PG	z real - z calc
1	19.59	19.608440	0.01844
2	29.55	29.547390	0.00261
3	29.50	29.59763	0.09763
4	30.15	30.149790	0.00210
5	30.20	30.210030	0.01003
6	30.38	30.390750	0.01075
7	30.44	30.450990	0.01099
8	29.79	29.788350	0.00135
9	30.01	30.029310	0.01931
10	19.70	19.712750	0.01275
11	29.73	29.728110	0.00189
12	30.01	30.027310	0.01731
13	30.56	30.571470	0.01147
14	30.26	30.270270	0.01027
15	30.32	30.330510	0.01051

Para la zona dos, mostrada en la Figura 28, el rango de valores de la coordenada x se encuentra en el rango de los 10200 m a los 10900 m, mientras que para la coordenada y se tomaron cuatro valores en los 1900 m, seis en los 2000 m, uno en los 2100, uno en los 2200, uno en los 2300, uno en los 2500 y uno en los 2800 m Esta segunda zona muestra una topografía casi plana, los valores de la variable z están dentro del rango de los 27 y 29 msnm. El archivo de datos completo se presenta en el apéndice D, sección D 1.2.



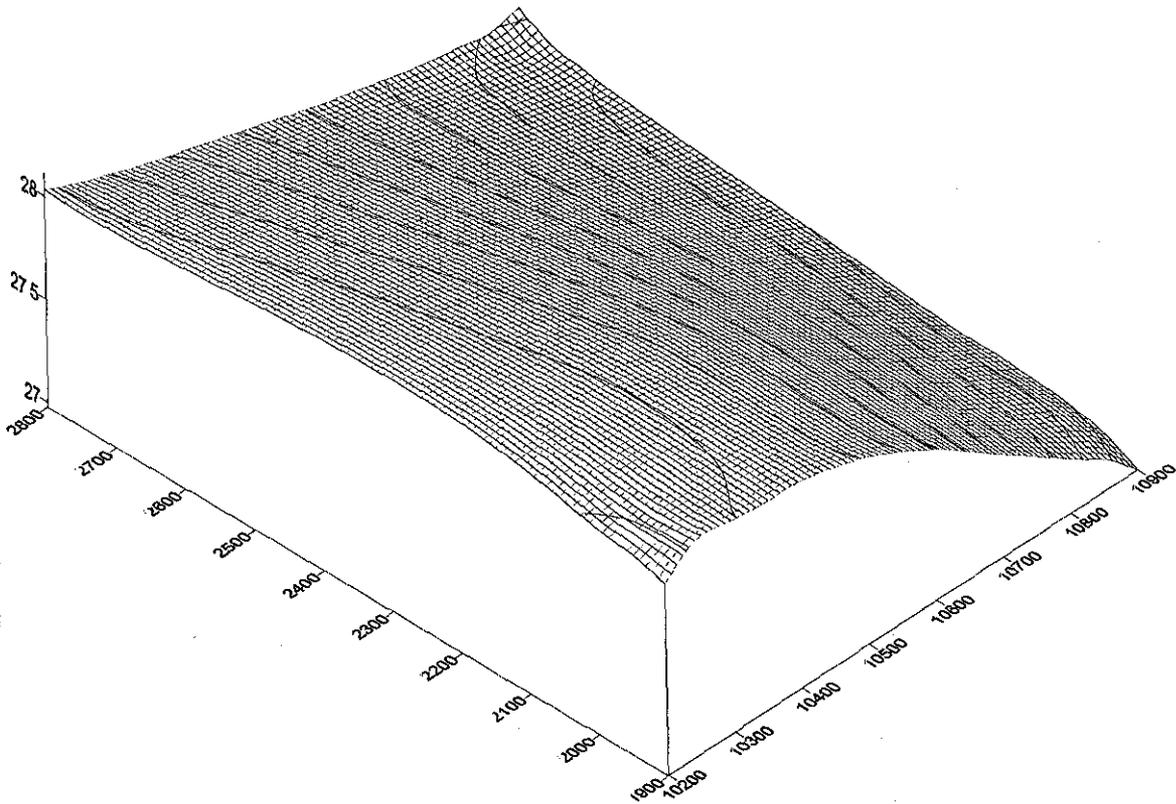


Fig 28. Zona 2.

Aquí se probó inicialmente con 16 nodos pero los resultados que se obtuvieron no fueron satisfactorios pues el mejor individuo a lo largo de 500 generaciones mostró un error promedio de 1.6 m; se optó entonces por aumentar la complejidad del árbol y se definieron 32 nodos, con ello, los resultados mejoraron notablemente, lo cual se puede apreciar en la Figura 29 en la que se muestra el comportamiento del mejor individuo. La función encontrada se muestra decodificada a continuación de la gráfica y se muestra también, en la Tabla 5, el valor de la variable z y el valor calculado con PG, así como el valor absoluto del error.

NO SE
 DEBE
 REPRODUCIR
 SIN
 PERMISO DE
 ALIANZA

TESIS CON
 FALLA DE ORIGEN

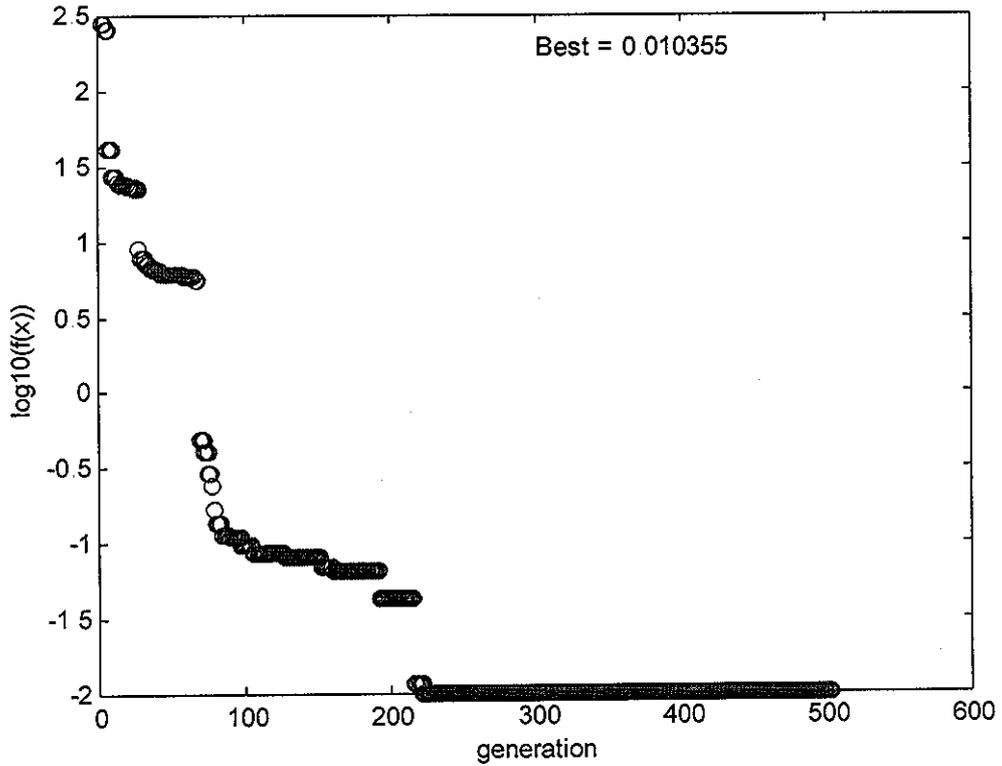


Fig 29. Optimización regional, zona 2.

$$z(i) = \frac{y(i)}{(y(i) + \cos(y(cte)) * 0.4 - \sin(x(i)))} + \cos\left(-0.6 * y(i) + \frac{x(i)}{y(i)} - y(i)\right) \quad i = 1, 15; \quad cte = 1001$$

TESIS CON
FALLA DE ORIGEN

WUO 2021
REPOSICION DE ALIAS

Tabla 5. Resultados obtenidos en la región 2.

Punto	Z real	Z calculada con PG	z real - z calc
1	29.0241	29.034	0.0099
2	28.1165	28.110	0.0065
3	27.6051	27.675	0.0699
4	28.3995	28.378	0.0215
5	29.0787	29.063	0.0157
6	27.5385	27.583	0.0445
7	29.4757	29.471	0.0047
8	29.5974	29.501	0.0964
9	27.5069	27.501	0.0059
10	29.4113	29.499	0.0877
11	29.2703	29.278	0.0077
12	29.4924	29.491	0.0014
13	27.5181	27.510	0.0081
14	29.4095	29.479	0.0695
15	29.2531	29.254	0.0009

Con los resultados anteriores se puede hacer ver que la PG mejora considerablemente los valores obtenidos tanto con los AG como con el método DFP. Para ello, basta recordar que los mejores errores obtenidos, tanto con el método DFP como con los AG generalmente eran del orden de décimas; esto es: si el valor real de la cota topográfica era de 27.40 la mejor estimación que se tenía con DF o con AG podía ser 27.50 o 27.30. Para destacar más lo dicho, la tabla 6 muestra las estimaciones con DFP, con AG y con PG para 5 puntos de las regiones estudiadas.

La función que generalmente encuentra PG tiene siempre presente a las funciones senoidal y/o cosenoidal. La decodificación de la función no presenta mayor problema pues se observó que con 16 nodos era suficiente para obtener excelentes resultados.

Tabla 6 Comparación de resultados para algunos puntos.

Punto No.	Valor real de la cota topográfica	Método de estimación		
		DFP	AG (binario)	PG
1	28.1165	27.954	27.840	28.110
2	30.440	28.005	29.543	30.450
3	30.32	28.116	29.023	30.330
4	19.60	18.250	18.100	19.712
5	27.53	25.998	26.567	27.58

5.3.2 Optimización global

Con tan buenos resultados obtenidos se probó ahora el encontrar una función que se ajustara de manera global a un pequeño conjunto de datos. Se seleccionaron 100 de los puntos digitalizados. La Figura 30 proporciona una vista tridimensional de estos 100 puntos y en el apéndice D, sección D 2, se presenta el archivo ASCII utilizado.

Se hicieron varias pruebas jugando principalmente con los valores del número de pobladores, máximo número de generaciones y número de nodos del árbol. En este caso, la corrida que mejor desempeño tuvo manejó 200 pobladores, 1000 generaciones y 48 nodos. La función encontrada con PG es un poco más compleja en su estructura que las mostradas en todos los casos anteriores, pero sigue siendo una expresión manejable en cuanto a complejidad se refiere.

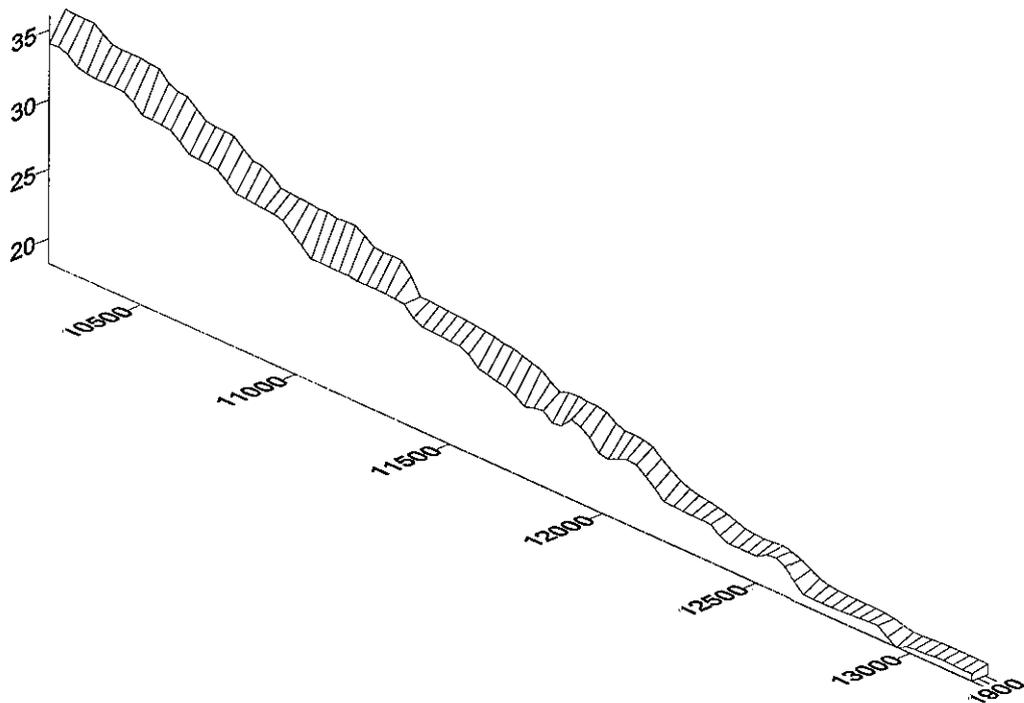


Fig 30. Región seleccionada para optimización global.

La Figura 31 muestra la evolución del mejor individuo para este caso de optimización global. A continuación de ella se presenta la expresión encontrada con PG para la evaluación de los 100 puntos de la zona y se muestra también, en la Tabla 7, los resultados obtenidos para cada punto así como el valor absoluto de la diferencia entre el valor real de la cota topográfica y el calculado con PG.

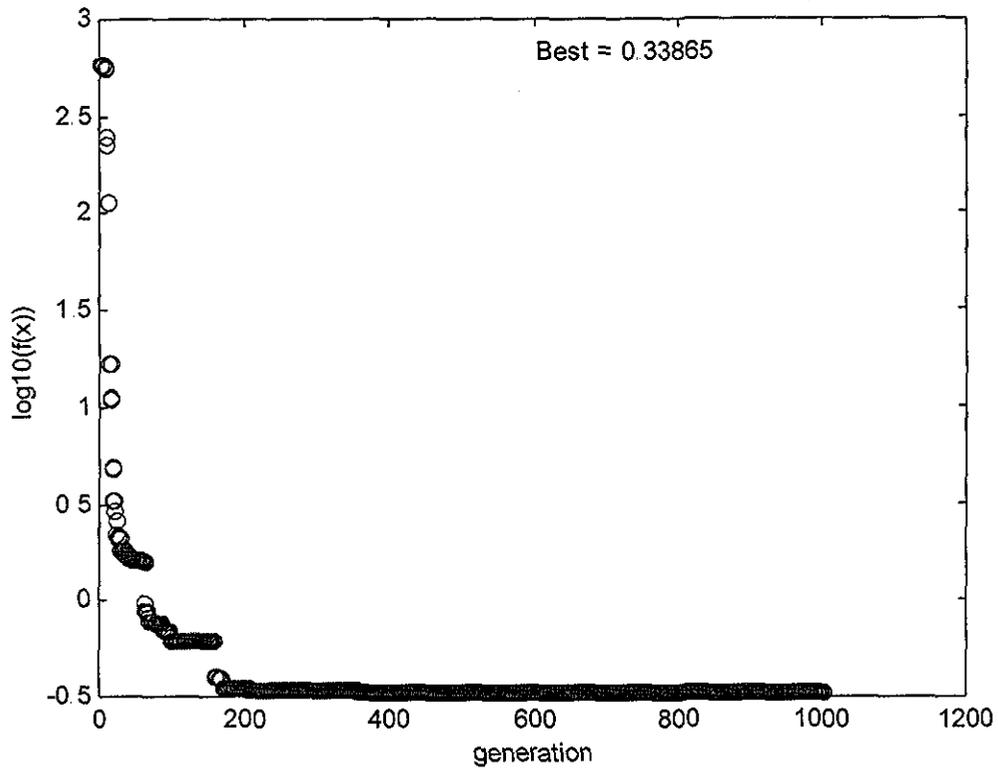


Fig 31. Optimización global.

Función encontrada con PG:

$$zcal(i) = \frac{y(i) - \cos B}{\left[\frac{y(i) + x(i)}{x(i)p} \right] + \cos x(i) + \frac{0.5x(i)}{y(i)} \left[\frac{0.5px(i) - 0.5 \cos A - y(i)p}{0.5p^2} \right]} \quad i = 1, 100$$

donde :

$$B = 1001;$$

$$p = 1.0e-15$$

$$A = \frac{y(i) - 0.5}{p}$$

Tabla 7. Resultados obtenidos de optimización global

z real	z calc con PG	z real - z calc
34.245700	34.031400	2.142982E-01
33.827380	34.016500	1.891136E-01
33.418840	33.005100	4.137383E-01
33.018750	32.999500	1.924896E-02
32.625840	32.978700	3.528557E-01
32.238930	32.938500	6.995659E-01
31.857010	31.875700	1.869392E-02
31.479260	31.797400	3.181438E-01
31.105120	31.706900	6.017818E-01
30.734310	30.611300	1.230068E-01
30.366770	30.500300	1.335335E-01
30.002700	30.392400	3.896980E-01
29.642500	29.298100	3.444023E-01
29.286720	29.209500	7.721901E-02
28.936000	29.131000	1.950016E-01
28.591020	29.077100	4.860764E-01
28.252460	28.020100	2.323589E-01
27.920890	26.983900	9.369869E-01
27.596780	26.962200	6.345806E-01
27.280450	26.981200	2.992535E-01
26.972040	26.962300	9.737015E-03
26.671490	26.942500	2.710075E-01
26.378570	26.917400	5.388298E-01
26.092890	26.890800	7.979088E-01
25.813910	25.828900	1.499367E-02
25.540970	25.768000	2.270260E-01
25.273380	25.702500	4.291172E-01
25.010400	25.620400	6.099987E-01
24.751320	24.505900	2.454205E-01
24.495490	24.329400	1.660938E-01
24.242370	24.063500	1.788692E-01
23.991530	23.293600	6.979275E-01
23.742690	23.860100	1.174145E-01
23.495720	22.794500	7.012177E-01
23.250640	24.036100	7.854614E-01
23.007610	23.461200	4.535904E-01
22.766890	22.101300	6.655903E-01
22.528840	22.672300	1.434650E-01
22.293850	22.998500	7.046509E-01
22.062350	22.123100	6.075096E-02
21.834730	21.100500	7.342339E-01
21.611350	21.035700	5.756531E-01
21.392490	21.000800	3.916874E-01
21.178310	21.013800	1.645069E-01

... continuación de la Tabla 7

z real	z calc con PG	z real - z calc
20.968890	20.127700	8.411846E-01
20.764180	20.204100	5.600758E-01
20.564020	20.166000	3.980236E-01
20.368180	20.836700	4.685173E-01
20.176310	20.297500	1.211929E-01
19.988000	19.019500	9.685040E-01
19.802840	19.004900	7.979412E-01
19.620370	19.026000	5.943756E-01
19.440180	19.086800	3.533840E-01
19.261890	19.171400	9.048653E-02
19.085170	19.231200	1.460342E-01
18.909790	18.162400	7.473869E-01
18.735610	18.725300	1.030540E-02
18.562570	18.708600	1.460323E-01
18.390710	18.852100	4.613876E-01
18.220160	18.773200	5.530357E-01
18.051110	18.781400	7.302933E-01
36.641830	36.042900	5.989342E-01
36.190400	36.025800	1.645966E-01
35.749650	36.013000	2.633514E-01
35.318130	35.002300	3.158302E-01
34.894430	34.984700	9.027100E-02
34.477230	34.943300	4.660721E-01
34.065400	34.878200	8.128052E-01
33.658040	33.797300	1.392555E-01
33.254540	33.703700	4.491577E-01
32.854580	32.600100	2.544746E-01
32.458090	32.479900	2.180862E-02
32.065320	32.378100	3.127861E-01
31.676690	31.279300	3.973942E-01
31.292840	31.187800	1.050453E-01
30.914480	30.108000	8.064823E-01
30.542370	30.053400	4.889698E-01
30.177250	30.000400	1.768475E-01
29.819770	29.975200	1.554317E-01
29.470460	29.963800	4.933414E-01
29.129660	29.991100	8.614426E-01
28.797520	28.979500	1.819782E-01
28.474000	28.965700	4.917011E-01
28.158830	28.939200	7.803669E-01
27.851570	26.907300	9.442692E-01
27.551630	26.860300	6.913261E-01
27.258280	26.790400	4.678783E-01
26.970740	26.710900	2.598419E-01
26.688210	26.604500	8.370590E-02
26.409890	26.462000	5.211449E-02

... continuación de la Tabla 7

z real	z calc con PG	z real - z calc
26.135070	26.252000	1.169357E-01
25.863140	25.961600	9.846306E-02
25.593630	25.598700	5.065918E-03
25.326250	24.588300	7.379475E-01
25.060840	25.011600	4.923630E-02
24.797420	24.795500	1.924515E-03
24.536190	24.809700	2.735119E-01
24.277420	24.060700	2.167244E-01
24.021520	23.965400	5.612183E-02
23.768930	23.802100	3.316879E-02

Con las coordenadas x-y de los 100 puntos analizados se graficó la superficie generada con la función que PG encontró para el cálculo de la cota topográfica en esos cien puntos. El resultado se muestra en la Figura 32.

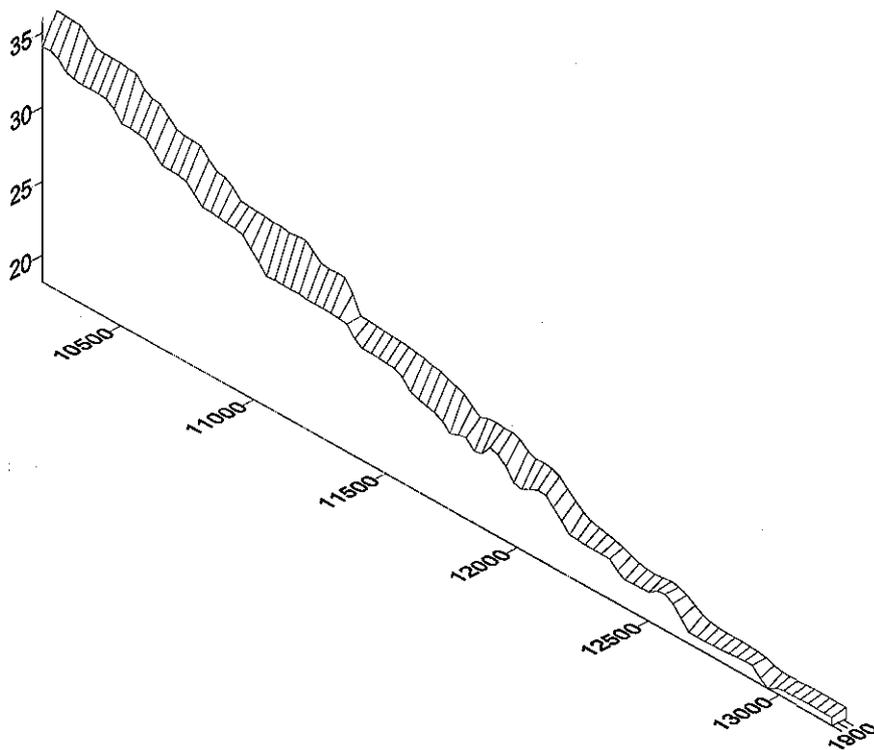


Fig 32. Superficie generada con PG para el caso de optimización global.

En la Figura 33 se presenta el valor absoluto de la diferencia entre el valor real de la cota topográfica y el calculado (tercera columna de la Tabla 7). Se puede apreciar en la gráfica que si bien hay errores muy cercanos al metro la gran mayoría se encuentra en valores bajos y por lo tanto los resultados obtenidos se pueden considerar buenos. Una forma de mejorar esto sería definir más funciones con las cuales PG pueda encontrar una nueva función que mejore los resultados que se lograron con el conjunto de funciones que se trabajó en el presente estudio.

De los casos analizados con la PG el que no obtiene errores muy pequeños es el de optimización global, esto se puede deber a que en realidad encontrar una expresión que reproduzca el comportamiento de una serie de datos de la topografía del río Colorado necesita un espacio de búsqueda más amplio que el definido en el presente estudio. Sin embargo, al aumentar el espacio de búsqueda se incrementa la complejidad de los árboles por lo cual, la solución a explorar en nuevos estudios sería incrementar el número de funciones pero incluir un operador que cada determinado número de generaciones revise y controle el crecimiento de los árboles para evitar redundancias innecesarias.

No obstante lo mencionado anteriormente, en los casos de optimización analizados los resultados se pueden considerar buenos y superan notablemente tanto los obtenidos por los AG como los del método DFP (Tabla 6 del capítulo 5)

Estos resultados y análisis previos hechos a lo largo de los capítulos anteriores constituyen la base sobre las que se sentarán las conclusiones del presente trabajo, las cuales se detallan en el capítulo siguiente.

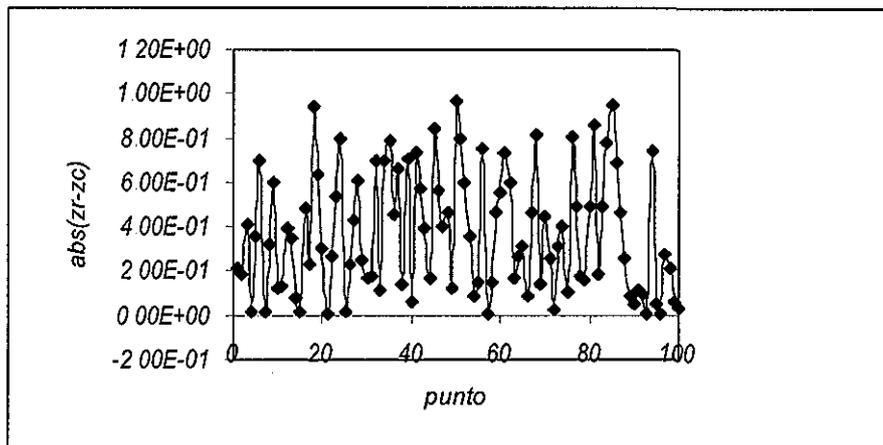


Fig 33. Gráfica de errores para el caso de optimización global.

TESIS CON
FALLA DE ORIGEN

6. CONCLUSIONES

Los Algoritmos Genéticos (AG), tanto en su representación binaria como en la real, mostraron un mejor desempeño que el método de optimización desarrollado por Davidon, Fletcher y Powell (DFP). Las figuras 6 y 7 de capítulo 4 muestran los errores entre el DFP y los AG en su representación binaria solamente, pero conviene recordar que los resultados obtenidos con los AG en su representación binaria fueron muy parecidos con los AG en notación real. Por lo que se puede decir que las conclusiones aplicables a los AG binarios son válidas para los AG en representación real.

El valor promedio de los coeficientes involucrados en la serie de Taylor analizada, es muy parecido tanto con los AG en representación binaria y real como con el método DFP. El comportamiento de los valores de los coeficientes es semejante en ambos casos y esto se puede apreciar en la gráfica mostrada en la Figura 17, en el capítulo cuatro.

A pesar de que los resultados obtenidos con los AG mejoran los mostrados en el estudio del 96 (Mendoza *et al*, 1996) fallan de manera importante cuando el punto por estimar es una cresta o un valle y la información de los puntos aledaños no ayuda.

Al realizarse las corridas para la obtención de resultados, los AG necesitan, en promedio un menor número de iteraciones. Esto se traduce en un menor tiempo de ejecución del programa de cómputo.

Una desventaja que se puede mencionar de los AG es que el rango de búsqueda para los nueve coeficientes de la serie requirió más pruebas y esfuerzo que la definición de los valores iniciales que necesita el método DFP. Según los resultados del estudio realizado en el Instituto de Ingeniería (Mendoza *et al*, 1996, 1998), el método DFP siempre partió de valores iniciales iguales a cero para todos los coeficientes, algo muy valioso cuando no se sabe por dónde orientar la solución del problema, como es el caso que se analizó en el presente estudio.

La conclusión que se puede obtener con los AG, tanto en representación binaria como en real, es que la función elegida (un desarrollo en serie de Taylor de dos variables hasta los términos de tercer orden) para hacer la estimación de la variable zeta dadas las coordenadas x-y del punto no resulta la más adecuada, pues tiene errores importantes en ambos métodos. Y se debe recordar que el objetivo primordial del trabajo es disponer de un método confiable y seguro para hacer estimaciones de la cota topográfica en la zona estudiada, que permitan diseñar obras de protección futuras, así como obtener otras variables de interés en puntos carentes de esta información.

La subrama de los AG, la Programación Genética (PG), resultó ser una herramienta más poderosa para realizar las predicciones de los valores de la cota topográfica. Es un poco más compleja de entender que los AG, pues los individuos aquí son estructuras del tipo árbol y los operadores de cruzamiento, mutación y reproducción, que son los usados por la PG para transformar a los individuos de la población, se deben tratar con mayor cuidado que en los AG.

Fijando un radio de búsqueda, la PG obtuvo excelentes resultados (tablas 6 y 7 del capítulo 5) mejorando considerablemente los obtenidos por los AG, tanto en representación binaria como en real, y el método DFP. Las expresiones que la PG mostró como resultado son una combinación de constantes y funciones aritméticas, senoidales y cosenoidales. En todos los casos analizados (optimización regional y global) la PG proporcionó funciones que se consideran, desde el punto de vista matemático, como funciones elementales (Piskunov, 1978).

Al investigar el desempeño de la PG usando las funciones definidas para tratar de ajustar una función universal a un subconjunto de 100 puntos seleccionados se observó que con 1000 generaciones el resultado es aceptable, pues el mejor individuo de la población presenta un error promedio bajo en todos los puntos. Se hizo una prueba aumentando sólo el número de generaciones a 1000 para ver si el error seguía disminuyendo, pero no sucedió así. De hecho el valor del error se mantiene casi constante a partir de la generación 1000. Se probó entonces a

aumentar la complejidad de la función esperada (del árbol de resultados); se pusieron primero 64 nodos y después se aumentó a 128, con este último valor se observó una pequeña mejoría y la expresión final, sin embargo, se complica en exceso, lo cual nos lleva a la conclusión de que sería mejor probar a aumentar las funciones con las cuales puede jugar la PG y agregar el operador de edición para controlar el crecimiento de los árboles para lograr expresiones más sencillas que conduzcan también a buenos resultados.

La conclusión final del presente trabajo es que la Computación Evolutiva (CE) proporcionó herramientas buenas para encontrar un modelo que permite la estimación de cotas topográficas con un rango de confiabilidad aceptable. Para resaltar lo dicho basta analizar los resultados presentados en la tabla 6 del capítulo 5, para el caso de optimización regional y, los de la tabla 7 para optimización global. Con esto se logra el objetivo que se planteó para el desarrollo del presente trabajo: encontrar un modelo que permitiera realizar estimaciones con un alto grado de confiabilidad para un conjunto de datos digitalizados de una zona del río Colorado.

7. REFERENCIAS

- [1] Arfken, G., Métodos matemáticos para físicos, Diana, 1980, 400 p.
- [2] Banzhaf, W., Nordin, P., Keller, R.E. y Francone, F.D., Genetic Programming An Introduction, Morgan Kaufmann Publishers, Inc., 1998, 470 p.
- [3] Box, G.E., y Hunter, J.S., Condensed Calculations for Evolutionary Operation Programs, Technometrics, 1959, Vol. 1, pp 77-95.
- [4] Bremmerman, H. J. Optimization through Evolution and Recombination, Spartan Books, 1962, 93 p.
- [5] Buckles, B.P. y Petry, F.E., Genetic Algorithms, IEEE Computer Society Press, 1992, 109 p.
- [6] Encarta, Enciclopedia 2000, Microsoft Corporation, 1993-1999, 1500 p.
- [7] Enciclopedia didáctica de ciencias naturales, Editorial Grupo Océano, 1998, 500 p.
- [8] Fogel, L., Owens, A. y Walsh, M., Artificial Intelligence Through a Simulation of Evolution, John Wiley & Sons, N. Y., 1965, 230 p.
- [9] Friedberg, R., Dunham, B., y North, T., A Learning Machine, Part II. IBM Journal of Research and Development, 1959, Vol. (3) 3, pp 282-287.
- [10] Goldberg, D.E., Genetic Algorithms in Search, Optimization, and Machine Learning, Addison-Wesley Publishing Company, 1989, 412 p.
- [11] Golden Software Inc., Surfer, Versión 7, Surface Mapping System, Golden, Colorado, U. S. A., 1999.

- [12] Jiménez, A., Berezowsky, M., Soto, G. Y Rivera, F., Cálculo del campo de velocidades en cuerpos de agua con modelo matemático bidimensional en coordenadas curvilíneas adaptables, Informe final, Vol. 3, Proyecto CONACYT Núm. 0641P-A9506, 1996, 104 p.
- [13] Koza, J R., Genetic Programming. On the Programming of Computer by Means of Natural Selection, The MIT Press, 1992, 819 p.
- [14] Mendoza, R., Alarcón, P. y Berezowsky, M., Cálculo del campo de velocidades en cuerpos de agua con modelo matemático bidimensional en coordenadas curvilíneas adaptables, Informe final, Vol. 2, Proyecto CONACYT Núm. 0641P-A9506, 1996, 30 p.
- [15] Mendoza, R. y Berezowsky, M., Interpolación de valores con un desarrollo en serie de Taylor de tercer orden, Memorias del XVII Congreso Latinoamericano de Hidráulica, Guayaquil, Ecuador, 1998, Vol. 1 pp 281-291
- [16] Mühlenbein, H y Schlierkamp-Voosen, D., Predictive Models for the Breeder Genetic Algorithm. Continuous Parameter Optimization, Evolutionary Computation, The MIT Press, 1993, 1(1), pp 25-49.
- [17] Piskunov, N., Cálculo diferencial e integral, Tomo I, MIR, 1978, 375 p.
- [18] Reed, R., Tombs, R., y Baricelli, N.A., Simulation of Biological Evolution and Machine Learning, Journal of Theoretical Biology, 1967, pp 319-342.
- [19] Rodríguez, K., Introducción a la programación genética, seminario de computación, Instituto de Investigaciones en Matemáticas Aplicadas y en Sistemas (IIMAS), 2000, México, D.F.
- [20] The MathWorks, Matlab Reference Guide, 1999, The MathWorks Inc.
- [21] Walsh, R., Optimization, Mac-Graw Hill, 1960, 200 p.

Apéndice A: Optimización de parámetros: Método DFP.

A.1 Introducción

Los métodos de optimización tienen como objetivo encontrar el valor óptimo de una función $f(\alpha_1, \alpha_2, \dots, \alpha_n)$ que depende de n parámetros $\alpha_1, \alpha_2, \dots, \alpha_n$. Se clasifican a grandes rasgos en métodos de búsqueda directa y métodos de gradiente o indirectos. Un método de búsqueda directa es aquél que, para alcanzar el punto óptimo x^* , se basa en evaluar $f(x)$ en una secuencia de puntos x_1, x_2, \dots y comparar sus valores. Se usan comúnmente en las siguientes circunstancias:

- (a) la función $f(x)$ no es diferenciable
- (b) las derivadas son discontinuas, o su evaluación es más difícil que la evaluación de la función $f(x)$ misma
- (c) si se está trabajando en procesos que involucren la dimensión temporal y se requiere una solución aproximada para cualquier instante durante el curso de los cálculos

Los métodos de gradiente son en general más efectivos que los métodos de búsqueda directa y están basados en el hecho de que $f(x)$ crece (o decrece) en una dirección d dependiendo de que la derivada direccional $[\Delta f(x)]'d$ sea positiva o negativa. Requieren la evaluación tanto de la función misma como de sus derivadas.

A.2 Serie de Taylor

En forma general, el desarrollo en Serie de Taylor para una función de dos variables hasta los términos de tercer orden se expresa como (Arfken, 1980; Piskunov, 1978):

$$\begin{aligned}
f(x, y) = & f(a, b) + (x - a) \frac{\partial f}{\partial x} \Big|_{(a, b)} + (y - b) \frac{\partial f}{\partial y} \Big|_{(a, b)} + \\
& \frac{1}{2!} \left[(x - a)^2 \frac{\partial^2 f}{\partial x^2} \Big|_{(a, b)} + 2(x - a)(y - b) \frac{\partial^2 f}{\partial x \partial y} \Big|_{(a, b)} + (y - b)^2 \frac{\partial^2 f}{\partial y^2} \Big|_{(a, b)} \right] + \\
& \frac{1}{3!} \left[(x - a)^3 \frac{\partial^3 f}{\partial x^3} \Big|_{(a, b)} + 3(x - a)^2 (y - b) \frac{\partial^3 f}{\partial x^2 \partial y} \Big|_{(a, b)} \right] + \\
& \frac{1}{3!} \left[3(x - a)(y - b)^2 \frac{\partial^3 f}{\partial x \partial y^2} \Big|_{(a, b)} + (y - b)^3 \frac{\partial^3 f}{\partial y^3} \Big|_{(a, b)} \right]
\end{aligned} \tag{A. 1}$$

donde a y b son los puntos de evaluación de la serie.

Es posible formar un sistema de ecuaciones si en la ecuación A. 1 se hace $a=b=0$ y las derivadas se sustituyen por coeficientes $a_0, a_1, a_2, \dots, a_8$. Para la función de la ecuación A. 1 es relativamente fácil obtener, de forma analítica, las derivadas que requieren los métodos de gradiente. Por ello, se optó aquí por seleccionar, de la gran variedad de algoritmos existentes un método, de los más sencillos de trabajar, que es conocido como DFP.

El algoritmo requiere evaluar el llamado parámetro de longitud de paso, λ , que sirve para acelerar la convergencia del método, para lo cual se empleó un método de búsqueda directa, seleccionándose el algoritmo de la búsqueda de la sección de oro.

A.3 El método DFP (Davidon-Fletcher-Powell)

Una gran parte de los métodos de gradiente intentan encontrar una dirección de búsqueda de la forma $H_k g_k$, donde H es una matriz simétrica positiva definida, la cual iteración tras iteración converge a G^{-1} . Estos métodos son conocidos colectivamente como métodos cuasi-Newton. Uno de los algoritmos más poderosos en esta clase, es el desarrollado por Davidon (Walsh, 1975) y más tarde perfeccionado por Fletcher y Powell. La fórmula iterativa del método es:

$$x_{k+1} = x_k - \lambda_k^* H_k g_k \quad (k = 1, 2, \dots) \tag{A. 2}$$

siendo H_k la k-ésima aproximación a la matriz inversa de segundas derivadas; si $k=1$ entonces $H_k = H_1 = I$, siendo I la matriz identidad. Este método, cerca del punto inicial se parece al de paso descendente (Walsh, 1975) en su rápida reducción inicial del valor de la función; cerca del punto óptimo se parece al de Newton-Raphson teniendo, por lo tanto, rápida convergencia

Si se supone que el punto inicial es x_1 y el punto al iniciar la k-ésima iteración es x_k , el algoritmo del método es el siguiente:

- 1) se calcula la dirección de búsqueda del punto actual x_k

$$d_k = -H_k g_k$$

- 2) se realiza una búsqueda lineal para encontrar $\lambda_k^* (> 0)$, siendo λ_k^* el valor de λ_k que minimiza $f(x_k + \lambda_k d_k)$

- 3) se recalcula la dirección de búsqueda

$$\sigma_k = \lambda_k^* d_k$$

- 4) se calcula el nuevo punto

$$x_{k+1} = x_k + \sigma_k$$

- 5) se evalúa el vector gradiente g en el punto actual x_{k+1}

- 6) se calcula la diferencia del vector gradiente

$$v_k = g_{k+1} - g_k$$

- 7) se construye la matriz H como $H_{k+1} = H_k + A_k + B_k$ siendo:

$$A_k = \frac{\sigma_k \sigma_k^T}{\sigma_k^T \nu_k}$$

$$B_k = \frac{H_k \nu_k \nu_k^T H_k}{\nu_k^T H_k \nu_k}$$

- 8) se evalúa cada componente del vector $|d_k|$ y si son menores que una cantidad prescrita (tolerancia), se detiene el proceso iterativo; en caso contrario se hace $k = k + 1$ y se regresa al paso 1.

Para realizar la búsqueda lineal del parámetro λ , se usó el método de la búsqueda de la sección de oro (Press et al, 1987); en el cuál es necesario definir una tripleta de puntos $\beta_1, \beta_2, \beta_3$ que satisfagan la relación $\beta_1 < \beta_2 < \beta_3$ y suponer que al ser unimodal la función, el punto óptimo de la misma se encuentra dentro del intervalo $[\beta_1, \beta_3]$ Está basado en la relación:

$$r = \lim_{n \rightarrow \infty} \frac{F_n}{F_{n+1}} = \frac{\sqrt{5} - 1}{2} \approx 0.618034$$

siendo F_n y F_{n+1} términos sucesivos de la Serie de Fibonacci, definida como:

$$F_0 = F_1 = 1, F_n = F_{n-1} + F_{n-2} \quad (n \geq 2)$$

En este algoritmo, el intervalo de búsqueda es reducido, en cada iteración, por el factor constante r , por lo que, la búsqueda de la sección de oro es una forma restrictiva del método de Fibonacci

Apéndice B: Algoritmo Genético Simple: Código Fuente.

```
/* estima cpp:
objetivo. El programa tiene codificado el esquema básico de un AG en su forma tradicional (binario) de
operación. Debe leer un archivo general que contiene datos de topografía de una zona del Río Colorado y
para cadauno de los puntos en que se desee estimar la coordenada z, deberá dar los valores de los
coeficientes de la función propuesta para estimar ésta (serie de Taylor de 2 funciones truncada hasta los
términos de tercer orden), así como el valor estimado de la cota topográfica, la Cota topográfica real y la
diferencia entre ambos valores.
versión para la tesis: julio del 2001 */

#define maxgen 500
#define maxpun 100
#define maxpop 100
#define maxlong 72
#define ncoef 9
#define base 2
#define pc 0.95
#define pm 0.09
#define pr 0.05
#define tolerancia 3.0e-05

#include<stdlib.h>
#include<stdio.h>
#include<math.h>
#include<time.h>
#include<dos.h>

void main() {
    /* -----
       prototipo de funciones
       -----
       */
    void generacion(int a[maxpop][maxlong]);
    void evalua(int individuo[maxpop][maxlong],float psel[maxpop],int indin[maxpop],
               double err_acum[],int gene,double rmin[],double rmax[],
               float x[],float y[],float z[],int nup,int npi);
    void mutacion(int a[maxpop][maxlong],int numeco);
    void seleccion(int a[maxpop][maxlong],int nume,float psel[maxpop],int indin[maxpop],int
nuevoindi[maxpop][maxlong]);
    void cruzamiento(int a[maxpop][maxlong],int numeco);
    void reasin(int a[maxpop][maxlong],int b[maxpop][maxlong]);
    /* -----
       declaración de variables
       -----
       */

    int i, individuo[maxpop][maxlong];
    int nuevoindi[maxpop][maxlong];
```

```

int nup;
float psel[maxpop];
int indin[maxpop];
double err_acum[maxpop];

float dife1,dife2,distt;
int numeco;
int ig, nupu;
int npi,ik;
float x[maxpun],y[maxpun],z[maxpun];
float xpe[10],ype[10];
float ncx[30],ncy[30],ncz[30];
float xp, yp;
float radio;
double rmin[ncoef],rmax[ncoef];
FILE *archda,*archra,*archpe;
double veri,aux,

/* lectura de los archivos de datos:
  1 de las coordenadas x-y-z de toda la región
  2 de los rangos
  3 de las coordenadas del punto en donde se hará la estimación*/
archda = fopen("datos top", "r");
archra = fopen("datos1 par", "r");
archpe = fopen("datos pes", "r");

fscanf(archda, "%d\n", &nupu);
for (i=0,i<nupu,i++){
  fscanf(archda, "%f %f %f\n", &x[i], &y[i], &z[i]);
  x[i]/=10000 0;
  y[i]/=10000 0;
  z[i]/=10000 0;
}

/* lectura de los rangos de búsqueda para los coeficientes */
for (i=0,i<ncoef,i++)
{
  fscanf(archra, "%f %f\n", &rmin[i], &rmax[i]);
}

/* lectura del punto (o puntos) donde se estimará */
fscanf(archpe, "%d\n", &npi);
for (i=0,i<npi,i++)
{
  fscanf(archpe, "%f %f\n", &xpe[i], &ype[i]);
  xpe[i]/=10000 0;
  ype[i]/=10000 0;
}

/* verifica que el número por conservar de individuos sea un entero positivo*/
aux = pr*maxpop;
veri = modf(aux, &veri);
if(veri !=0 0){

```

```

    printf("error: número por conservar de individuos no entero\n");
    exit(1);}
else
    numeco = pr*maxpop;

/* define radio de búsqueda */
    radio = x[1]-x[0];
/* iniciar proceso de cálculo para todos los xy*/
for(ik=0,ik<npi,ik++)
{
    xp=xpe[ik];
    yp=ype[ik];
/* contabiliza los puntos de influencia*/
    nup=-1;
    for (i=0,i<nupu,i++)
    {
        dife1 = pow((x[i]-xp),2.0);
        dife2 = pow((y[i]-yp),2.0);
        distt = sqrt(dife1+dife2);
        if(distt<=(3.*radio))
        {
            nup+=1;
            ncx[nup]=x[i];
            ncy[nup]=y[i];
            ncz[nup]=z[i];
        }
    }
}

if(nup<1){printf("No hay puntos de influencia\n");
    exit(1);}

/* reducción del número de puntos de influencia */
if(nup>15 ){ printf("NÚMERO DE PUNTOS DE INFLUENCIA = %d\n",nup);
    printf("MODIFICAR, EL NÚMERO DE PUNTOS A 15\n");
    nup = 15;
    printf("nuevo número %d\n",nup);
}

ig=-1,
/* genera la primera población con un máximo de individuos definido por maxpop */
    generacion(individuo);
do { ig+=1;
    printf("Generación: %d\n",ig);
    evalua(individuo,psel,indin,err_acum,ig,rmin,rmax,ncx,ncy,ncz,nup,npi);
    seleccion(individuo,numeco,psel,indin,nuevoindi);
    cruzamiento(nuevoindi,numeco);
    mutacion(nuevoindi,numeco);
    reasin(nuevoindi,individuo);
} while(ig<maxgen);

```

```

} /* fin del for de los puntos a estimar */
} /* fin del principal */

```

```

/* *****
definición de funciones
***** */

```

```

void generacion (int individuo[maxpop][maxlong]){

```

```

    int i, j,
    randomize();
    for (i=0;i<maxpop,i++){
        for (j=0;j<maxlong;j++){
            individuo[i][j]=rand() %2,
        }
    }
} /* fin de la función de generación */

```

```

void evalua(int individuo[maxpop][maxlong],float psel[maxpop],int indin[maxpop],
double err_acum[maxpop],int ige, double rmin[],double rmax[],float ncx[],float ncy[],
float ncz[],int nup,int npi){

```

```

/* @@@@ prototipo de funciones locales
@@@@ */

```

```

void decodifica(int individuo[maxpop][maxlong],
double a0[],double a1[],double a2[],
double a3[],double a4[], double a5[],
double a6[],double a7[],double a8[],double rmin[],
double rmax[]);

```

```

void funcion_objetivo(double a0[], double a1[],double a2[],
double a3[], double a4[], double a5[],
double a6[], double a7[], double a8[],
float fit[maxpop],int indin[maxpop],
float ncx[],float ncy[],
float ncz[],int npu,int npi,double err_acum[maxpop],
int ige);

```

```

double a0[maxpop],a1[maxpop],a2[maxpop],a3[maxpop],a4[maxpop],a5[maxpop],
double a6[maxpop], a7[maxpop],a8[maxpop];

```

```

decodifica(individuo,a0,a1,a2,a3,a4,a5,a6,a7,a8,rmin,rmax);
funcion_objetivo(a0,a1,a2,a3,a4,a5,a6,a7,a8,psel,indin,ncx,ncy,ncz,nup,npi,err_acum,ige),

```

```

} /* fin de la función de evaluación */

```

```

void decodifica(int individuo[maxpop][maxlong], double a0[], double a1[],
double a2[], double a3[],double a4[], double a5[], double a6[],

```

```

    double a7[], double a8[], double rmin[],double rmax[]){
/* prototipo de funciones locales */
double valor_real(int coef,int pobla,int poten, int inicio[],int fin[],
    int individuo[][maxlength],double rmin[],double rmax[]),
int i, poten,ll;
int inicio[ncoef],fin[ncoef],
double coef,
    poten = maxlength/ncoef,
    for (i=0;i<maxpop,i++) {
        for (ll=0,ll<ncoef,ll++){
            if(ll==0) {
                inicio[ll]=0,
                fin[ll]=poten,}
            else {
                inicio[ll]=fin[ll-1],
                fin[ll]=inicio[ll]+poten,}
        coef=valor_real(ll,i,poten,inicio,fin,individuo,rmin,rmax),
        switch(ll) {
            case 0:
                a0[i] =coef,
                break,
            case 1: a1[i] = coef,
                break,
            case 2: a2[i] = coef,
                break,
            case 3: a3[i] = coef,
                break,
            case 4: a4[i] = coef,
                break,
            case 5: a5[i] = coef,
                break,
            case 6: a6[i] = coef,
                break,
            case 7: a7[i] = coef,
                break,
            case 8: a8[i] = coef,
                break,
        }
    }
}
}
} /* fin de la función decodifica */
double valor_real(int nc,int pobla,int poten, int inicio[],int fin[],
    int individuo[][maxlength],double rmin[],double rmax[]){
int coefe, dosalan,j;
double rt,coef,
    coefe =0,
    dosalan = pow(base,poten);
    for(j=inicio[nc],j<fin[nc],j++){
        poten-=1;
        coefe+=individuo[pobla][j]*pow(2 0,poten);
    }
}
}
}

```

```

    }
    rt = rmax[nc]-rmin[nc];
    coef = rmin[nc]+(coefe*rt)/(dosalan-1);
    return(coef);

} /* fin de la función de valor_real */

void funcion_objetivo(double a0[],double a1[],double a2[], double a3[],
    double a4[], double a5[], double a6[], double a7[],
    double a8[],float psel[maxpop],int indin[maxpop],float ncx[],float ncy[],
    float ncz[], int nup,int npi,double err_acum[],int ige){
/* la función a evaluar es una serie de taylor de tercer orden */
    int k,punto,i;
    float errortot=0.0;
    double fzeta[maxpop];
    double z1[maxpop],z2[maxpop];

    float d1;
    float eac1=0.0;
    FILE *fp;

    for(k = 0,k<maxpop,k++){
        fzeta[k]=0.0;
        err_acum[k]=0.0;
        for (punto=0,punto<nup,punto++){
            fzeta[k]=a0[k]*ncx[punto]+a1[k]*ncy[punto]+(1./2.)*a2[k]*pow(ncx[punto],2.0)+
                (1./2.)*a3[k]*ncx[punto]*ncy[punto]+(1./2.)*a4[k]*pow(ncy[punto],2.0)+
                (1./6.)*a5[k]*pow(ncx[punto],3.0)+(1./6.)*a6[k]*pow(ncx[punto],2.0)*ncy[punto]+
                (1./6.)*a7[k]*ncx[punto]*pow(ncy[punto],2.0)+(1./6.)*a8[k]*pow(ncy[punto],3.0);
            err_acum[k]+=fabs(fzeta[k]-ncz[punto]);
        } /* fin del do de los puntos */
    } /* err_acum[k]=nup; */
    printf("%e %d\n",err_acum[k]/nup,k),
    errortot+=err_acum[k];
    if((err_acum[k]/nup)<=tolerancia) {
        fp = fopen("resul sal", "w");
        fprintf(fp,"los coeficientes universales son\n");
        fprintf(fp,"a0=%e a1=%e a2=%e a3=%e\n",a0[k],a1[k],a2[k],a3[k]);
        fprintf(fp,"a4=%e a5=%e a6=%e a7=%e\n",a4[k],a5[k],a6[k],a7[k]);
        fprintf(fp,"a8=%e\n",a8[k]);
        for(i=0,i<npi,i++){
            z1[i]=fzeta[i];
            fprintf(fp,"zc = %e ",z1[i]);
        }
    }
    else{ if(ige == maxgen-1) {
        k = maxpop-1;
        printf("los coeficientes universales son\n");
        printf("a0=%e a1=%e a2=%e a3=%e\n",a0[k],a1[k],a2[k],a3[k]);
        printf("a4=%e a5=%e a6=%e a7=%e\n",a4[k],a5[k],a6[k],a7[k]);
        printf("a8=%e\n",a8[k]);
    }
}

```

```

        for(i=0,i<npi,i++){
            z1[i]=fzeta[i];
            printf("zc = %e",z1[i]);
            exit(1);
        }
    }
}

} /* fin del do de la población */

/* calcula la probabilidad de ser seleccionado */
for(k=0;k<maxpop,k++){
    psel[k]=(1.0-(err_acum[k]/errortot))/(maxpop-1),
    indin[k]=k;
}

} /* fin de la función funcion_objetivo */

void seleccion(int individuo[maxpop][maxlong],int numeco,float psel[maxpop],
               int indin[maxpop],int nuevoindi[maxpop][maxlong]) {

/* la selección se hace usando el método de la ruleta */
/* prototipo de función local */

void sort(float a[maxpop], float b[maxpop],int indin[maxpop]),

    int i, j,numeg,k;
    float intini[maxpop],intfin[maxpop],numec;
    int indicon[maxpop];
    int indisel[maxpop];
    float prob[maxpop];
    time_t t=101;

for (i=0,i<maxpop,i++){
    indicon[i]=0, intini[i]=0.0, intfin[i]=0.0,
}
/* selección elitista de los individuos que se conservarán
ordenamos primero las probabilidades de selección en
orden decreciente */
    sort(psel,prob,indin);

    for (i=0,i<numeco,i++){
        indisel[i]=indin[i];
/* printf("indi conservado [%d] %d\n",i,indisel[i]);
sleep(3);*/
        for (j=0,j<maxlong,j++){
            nuevoindi[i][j]=individuo[indisel[i]][j];
        }
    }

for(k=0;k<maxpop,k++){
    if(k==0) {

```

```

        intini[0]=0.0;
        intfin[0]=psel[0];
    }
    else {
        intini[k]=intfin[k-1];
        intfin[k]=intini[k]+psel[k];
    }
}

randomize();
for(i=numeco,i<maxpop,i++){
    for(k=0,k<1;k++){
        numeg=rand() % 101;
        numec = numeg/100.0;
        for (j=0;j<maxpop;j++){
            if(numec>=intini[j] && numec<=intfin[j]){
                indisel[i]=j;
                for (k =0;k<maxlong;k++){
                    nuevoindi[i][k]=individuo[indisel[i]][k];
                }
            }
        }
    }
}

```

} /* fin de la función seleccion */

```

void sort(float psel[maxpop],float prob[maxpop],int indin[maxpop]){
    int i, j,temp1;
    float temp;
    for(i=0,i<maxpop-1,i++){
        for (j=i+1,j<maxpop,j++){
            if(psel[j]<psel[i]){
                temp = psel[i];
                psel[i]=psel[j];
                psel[j]= temp;
            }
            /* asignamos los indices */
            temp1 = indin[i];
            indin[i]=indin[j];
            indin[j]= temp1;
        }
        for (i=0,i<maxpop,i++)
            prob[i]=psel[i];
    }
}

```

} /* fin de la función local sort */

```

void cruzamiento(int nuevoindi[maxpop][maxlong],int numeco){

```

```

    int ik,k,l,padre0[maxlong],padre1[maxlong],pcruz;
    int nge,i,j,jj;
    float npcruza;
    time_t t=101;

```

```

randomize();
for (k=0;k<maxpop;k+=2) {
    for(jj=0;jj<1;jj++){
        nge = rand() % 101;}
        npcruza=nge/100.0;
/*      printf("numero aleatorio %d , porcentaje %f\n",nge,npcruza);*/
        if(npcruza<=pc) {
/*      printf("padre 1\n"),
        printf("k %d\n",k);*/
        for(ik=0,ik<maxlong,ik++){
            padre0[ik] =nuevoindi[k][ik];
/*      printf(" %d ",padre0[ik]);*/
        }
/*      printf("\n");
        printf("padre 1\n"),
        printf("k+1 %d\n",k+1);*/
        for(ik=0,ik<maxlong,ik++){
            padre1[ik] =nuevoindi[k+1][ik];
/*      printf(" %d ",padre1[ik]);*/
        }

/*      randomize();*/
        for(jj=0;jj<1;jj++){
            pcruz =rand() % maxlong;}
/*      printf("punto de cruzamiento %d\n",pcruz);*/
            for(l=0,l<maxlong,l++){
                if(l<pcruz){
                    nuevoindi[k][l]=padre0[l];
                }
                else{
                    nuevoindi[k][l]=padre1[l];
                }
            }
            for(l=0,l<maxlong,l++){
                if(l<pcruz){
                    nuevoindi[k+1][l]=padre1[l];
                }
                else{
                    nuevoindi[k+1][l]=padre0[l];
                }
            }
        }
    }
}
} /* fin de la función cruzamiento*/

```

```

void mutacion(int nuevoindi[maxpop][maxlong],int numeco){

```

```

    int i,puntomuta,k;

```

```

time_t t=101;
int nume;
float numegen;
randomize();
for (i=numeco;i<maxpop,i++){
for(k=0;k<1;k++){
nume=rand()%101;}
numegen=nume/100.0;
/* printf("numero generado en mutación %d poblador %d\n",nume,i);*/

if(numegen<=pm ) {
for(k=0;k<1;k++){
puntomuta=rand()%maxlong;}
if(nuevoindi[i][puntomuta]==1)
nuevoindi[i][puntomuta]=0;
else
nuevoindi[i][puntomuta]=1;
}
}

} /* fin de la función de mutación */

void reasin(int nuevoindi[maxpop][maxlong],int individuo[maxpop][maxlong]){
int i,j;

for(i=0,i<maxpop,i++){
for(j=0,j<maxlong,j++){
individuo[i][j]=nuevoindi[i][j];
}
}
} /* fin de la función de reasignación */

```

□

Apéndice C: Conjunto General de Datos.

x (m)	y (m)	z (msnm)
10200	1900	28.0314
10250	1900	28.0165
10300	1900	28.0051
10350	1900	27.9995
10400	1900	27.9787
10450	1900	27.9385
10500	1900	27.8757
10550	1900	27.7974
10600	1900	27.7069
10650	1900	27.6113
10700	1900	27.5003
10750	1900	27.3924
10800	1900	27.2981
10850	1900	27.2095
10900	1900	27.131
10950	1900	27.0771
11000	1900	27.0201
11050	1900	26.9839
11100	1900	26.9622
11150	1900	26.9812
11200	1900	26.9623
11250	1900	26.9425
11300	1900	26.9174
11350	1900	26.8908
11400	1900	26.8289
11450	1900	26.768
11500	1900	26.7025
11550	1900	26.6204
11600	1900	26.5059
11650	1900	26.3294
11700	1900	26.0635
11750	1900	25.2936
11800	1900	23.8601
11850	1900	22.7945
11900	1900	24.0361
11950	1900	23.4612
12000	1900	24.1013
12050	1900	24.6723
12100	1900	24.9985
12150	1900	25.1231
12200	1900	25.1005
12250	1900	25.0357
12300	1900	25.0008
12350	1900	25.0138
12400	1900	25.1277
12450	1900	25.2041
12500	1900	25.166
12550	1900	24.8367
12600	1900	24.2975

12650	1900	24.0195
12700	1900	24.0049
12750	1900	24.026
12800	1900	24.0868
12850	1900	24.1714
12900	1900	24.2312
12950	1900	24.1624
13000	1900	23.7253
13050	1900	22.7086
13100	1900	21.8521
13150	1900	21.7732
13200	1900	21.7814
10200	1950	28.0429
10250	1950	28.0258
10300	1950	28.013
10350	1950	28.0023
10400	1950	27.9847
10450	1950	27.9433
10500	1950	27.8782
10550	1950	27.7973
10600	1950	27.7037
10650	1950	27.6001
10700	1950	27.4799
10750	1950	27.3781
10800	1950	27.2793
10850	1950	27.1878
10900	1950	27.108
10950	1950	27.0534
11000	1950	27.0004
11050	1950	26.9752
11100	1950	26.9638
11150	1950	26.9911
11200	1950	26.9795
11250	1950	26.9657
11300	1950	26.9392
11350	1950	26.9073
11400	1950	26.8603
11450	1950	26.7904
11500	1950	26.7109
11550	1950	26.6045
11600	1950	26.462
11650	1950	26.252
11700	1950	25.9616
11750	1950	25.5987
11800	1950	24.5883
11850	1950	23.0116
11900	1950	22.7955
11950	1950	22.8097
12000	1950	22.0607
12050	1950	23.9654
12100	1950	24.8021
12150	1950	25.0175

12200	1950	25	.0645
12250	1950	25	.0379
12300	1950	25	.0198
12350	1950	25	.0169
12400	1950	25	.1223
12450	1950	25	.2344
12500	1950	25	.2176
12550	1950	24	.7322
12600	1950	24	.139
12650	1950	24	.0512
12700	1950	24	.0236
12750	1950	24	.1101
12800	1950	24	.3856
12850	1950	24	.6342
12900	1950	24	.6028
12950	1950	24	.361
13000	1950	23	.7671
13050	1950	22	.3736
13100	1950	21	.7505
13150	1950	21	.7588
13200	1950	21	.8734
10200	2000	28	.0267
10250	2000	28	.0227
10300	2000	28	.0126
10350	2000	28	.0041
10400	2000	27	.9979
10450	2000	27	.9556
10500	2000	27	.8847
10550	2000	27	.7997
10600	2000	27	.706
10650	2000	27	.5876
10700	2000	27	.4814
10750	2000	27	.3644
10800	2000	27	.262
10850	2000	27	.1654
10900	2000	27	.0821
10950	2000	27	.0278
11000	2000	26	.9923
11050	2000	26	.9734
11100	2000	26	.9697
11150	2000	26	.9964
11200	2000	26	.9909
11250	2000	26	.9793
11300	2000	26	.9646
11350	2000	26	.931
11400	2000	26	.8897
11450	2000	26	.8038
11500	2000	26	.7089
11550	2000	26	.589
11600	2000	26	.4217
11650	2000	26	.2136
11700	2000	26	.0129

11750	2000	25	.7504
11800	2000	25	.2217
11850	2000	23	.5879
11900	2000	23	.534
11950	2000	24	.2923
12000	2000	21	.8874
12050	2000	22	.7695
12100	2000	24	.2489
12150	2000	24	.7676
12200	2000	24	.9463
12250	2000	24	.9942
12300	2000	24	.9904
12350	2000	24	.9886
12400	2000	25	.0566
12450	2000	25	.16
12500	2000	24	.9802
12550	2000	23	.6949
12600	2000	23	.2768
12650	2000	23	.9975
12700	2000	24	.0359
12750	2000	24	.1981
12800	2000	25	.0182
12850	2000	25	.719
12900	2000	26	.524
12950	2000	24	.7897
13000	2000	22	.9875
13050	2000	22	.093
13100	2000	21	.7754
13150	2000	21	.8661
13200	2000	22	.0936
10200	2050	27	.9937
10250	2050	28	.0118
10300	2050	28	.011
10350	2050	28	.0036
10400	2050	28	.0121
10450	2050	27	.966
10500	2050	27	.8857
10550	2050	27	.7971
10600	2050	27	.6959
10650	2050	27	.5822
10700	2050	27	.4763
10750	2050	27	.356
10800	2050	27	.2484
10850	2050	27	.1463
10900	2050	27	.0531
10950	2050	27	.0022
11000	2050	26	.9814
11050	2050	26	.9688
11100	2050	26	.9719
11150	2050	26	.9938
11200	2050	26	.9918
11250	2050	26	.9874

11300	2050	26.9799
11350	2050	26.9584
11400	2050	26.9182
11450	2050	26.8287
11500	2050	26.7305
11550	2050	26.5937
11600	2050	26.4054
11650	2050	26.1499
11700	2050	25.9651
11750	2050	25.7366
11800	2050	25.0881
11850	2050	23.8407
11900	2050	24.0629
11950	2050	24.3025
12000	2050	22.6024
12050	2050	21.8125
12100	2050	22.7602
12150	2050	24.2056
12200	2050	24.7638
12250	2050	24.8975
12300	2050	24.9342
12350	2050	24.9413
12400	2050	24.9369
12450	2050	25.036
12500	2050	25.0968
12550	2050	23.1301
12600	2050	21.9134
12650	2050	23.3685
12700	2050	24.0086
12750	2050	24.4056
12800	2050	27.635
12850	2050	28.0504
12900	2050	28.5247
12950	2050	25.2839
13000	2050	21.7954
13050	2050	21.8125
13100	2050	21.9222
13150	2050	22.0514
13200	2050	22.4597
10200	2100	27.9629
10250	2100	28.0047
10300	2100	28.009
10350	2100	28.0032
10400	2100	28.0004
10450	2100	27.9546
10500	2100	27.8791
10550	2100	27.7878
10600	2100	27.6821
10650	2100	27.5785
10700	2100	27.4721
10750	2100	27.3515
10800	2100	27.2462

10850	2100	27	.1415
10900	2100	27	.0453
10950	2100	26	.9942
11000	2100	26	.9864
11050	2100	26	.9816
11100	2100	26	.9812
11150	2100	26	.997
11200	2100	26	.9936
11250	2100	26	.9887
11300	2100	26	.9866
11350	2100	26	.9712
11400	2100	26	.9381
11450	2100	26	.8674
11500	2100	26	.7775
11550	2100	26	.6421
11600	2100	26	.4475
11650	2100	26	.1598
11700	2100	25	.9601
11750	2100	25	.7038
11800	2100	25	.0529
11850	2100	23	.9942
11900	2100	23	.9991
11950	2100	24	.2634
12000	2100	24	.2874
12050	2100	22	.7899
12100	2100	22	.1006
12150	2100	23	.8779
12200	2100	24	.6205
12250	2100	24	.8373
12300	2100	24	.9086
12350	2100	24	.9194
12400	2100	24	.9445
12450	2100	25	.0996
12500	2100	25	.25
12550	2100	23	.5513
12600	2100	21	.8455
12650	2100	23	.1434
12700	2100	23	.8902
12750	2100	26	.2633
12800	2100	28	.6727
12850	2100	28	.0473
12900	2100	28	.0562
12950	2100	25	.1331
13000	2100	21	.8056
13050	2100	22	.0149
13100	2100	22	.3919
13150	2100	22	.7276
13200	2100	23	.0351
10200	2150	27	.9329
10250	2150	28	.007
10300	2150	28	.003
10350	2150	28	.0008

10400	2150	27.9837
10450	2150	27.9357
10500	2150	27.8673
10550	2150	27.7821
10600	2150	27.6742
10650	2150	27.5761
10700	2150	27.4552
10750	2150	27.3519
10800	2150	27.2522
10850	2150	27.1537
10900	2150	27.0741
10950	2150	27.0237
11000	2150	27.0048
11050	2150	26.9995
11100	2150	26.9894
11150	2150	26.998
11200	2150	26.9972
11250	2150	26.9938
11300	2150	26.9899
11350	2150	26.9825
11400	2150	26.9627
11450	2150	26.9171
11500	2150	26.8525
11550	2150	26.7405
11600	2150	26.5702
11650	2150	26.3506
11700	2150	26.1482
11750	2150	25.8339
11800	2150	25.053
11850	2150	23.9096
11900	2150	23.9441
11950	2150	24.0857
12000	2150	24.0789
12050	2150	23.2799
12100	2150	23.4812
12150	2150	24.2566
12200	2150	24.6288
12250	2150	24.8131
12300	2150	24.89
12350	2150	24.9294
12400	2150	25.0723
12450	2150	25.3675
12500	2150	25.3621
12550	2150	22.74
12600	2150	21.7237
12650	2150	24.4082
12700	2150	26.5987
12750	2150	27.8907
12800	2150	26.3551
12850	2150	22.8823
12900	2150	23.1175
12950	2150	22.2372

13000	2150	21	.5551
13050	2150	22	.084
13100	2150	22	.8099
13150	2150	23	.3399
13200	2150	23	.3792
10200	2200	27	.8866
10250	2200	27	.9922
10300	2200	27	.9957
10350	2200	28	.0005
10400	2200	27	.9643
10450	2200	27	.9159
10500	2200	27	.8554
10550	2200	27	.7613
10600	2200	27	.6668
10650	2200	27	.5716
10700	2200	27	.4542
10750	2200	27	.3577
10800	2200	27	.2633
10850	2200	27	.1715
10900	2200	27	.1021
10950	2200	27	.0526
11000	2200	27	.0311
11050	2200	27	.0128
11100	2200	27	.0015
11150	2200	27	.0022
11200	2200	27	.0004
11250	2200	26	.9972
11300	2200	26	.9938
11350	2200	26	.9893
11400	2200	26	.985
11450	2200	26	.9666
11500	2200	26	.9282
11550	2200	26	.8644
11600	2200	26	.7336
11650	2200	26	.5081
11700	2200	26	.2596
11750	2200	26	.0479
11800	2200	25	.3945
11850	2200	24	.6019
11900	2200	24	.1396
11950	2200	24	.0254
12000	2200	24	.0409
12050	2200	24	.0204
12100	2200	24	.2288
12150	2200	24	.4713
12200	2200	24	.6847
12250	2200	24	.8393
12300	2200	24	.89
12350	2200	24	.949
12400	2200	25	.2489
12450	2200	25	.6134
12500	2200	26	.2666

12550	2200	26.4122
12600	2200	27.9452
12650	2200	27.8785
12700	2200	25.3893
12750	2200	23.5735
12800	2200	22.2548
12850	2200	21.6406
12900	2200	21.4776
12950	2200	21.3149
13000	2200	21.5057
13050	2200	22.1773
13100	2200	23.0226
13150	2200	23.8203
13200	2200	24.0492
10200	2250	27.8246
10250	2250	27.932
10300	2250	27.9773
10350	2250	27.9985
10400	2250	27.9408
10450	2250	27.8977
10500	2250	27.8349
10550	2250	27.7483
10600	2250	27.6586
10650	2250	27.5676
10700	2250	27.4542
10750	2250	27.3636
10800	2250	27.2752
10850	2250	27.1971
10900	2250	27.124
10950	2250	27.0821
11000	2250	27.055
11050	2250	27.0302
11100	2250	27.009
11150	2250	27.0075
11200	2250	27.0052
11250	2250	27.0039
11300	2250	27.0035
11350	2250	27.003
11400	2250	27.0058
11450	2250	26.9983
11500	2250	26.9848
11550	2250	26.9802
11600	2250	26.9207
11650	2250	26.6042
11700	2250	26.1555
11750	2250	25.9006
11800	2250	25.5076
11850	2250	25.0393
11900	2250	24.5329
11950	2250	24.2718
12000	2250	24.2269
12050	2250	24.2816

12100	2250	24	.4652
12150	2250	24	.6366
12200	2250	24	.7744
12250	2250	24	.9085
12300	2250	24	.983
12350	2250	25	.1215
12400	2250	25	.5443
12450	2250	25	.9153
12500	2250	27	.4
12550	2250	28	.5013
12600	2250	27	.6991
12650	2250	25	.7519
12700	2250	22	.6622
12750	2250	21	.6073
12800	2250	21	.7572
12850	2250	21	.8977
12900	2250	21	.7454
12950	2250	21	.505
13000	2250	21	.9684
13050	2250	22	.9654
13100	2250	23	.6597
13150	2250	24	.485
13200	2250	25	.0862
10200	2300	27	.8041
10250	2300	27	.9031
10300	2300	27	.9617
10350	2300	27	.9957
10400	2300	27	.9194
10450	2300	27	.8765
10500	2300	27	.802
10550	2300	27	.7267
10600	2300	27	.6517
10650	2300	27	.5659
10700	2300	27	.4539
10750	2300	27	.3693
10800	2300	27	.2884
10850	2300	27	.2143
10900	2300	27	.148
10950	2300	27	.1014
11000	2300	27	.0729
11050	2300	27	.0431
11100	2300	27	.013
11150	2300	27	.0088
11200	2300	27	.0048
11250	2300	27	.0031
11300	2300	27	.0039
11350	2300	27	.0063
11400	2300	27	.0142
11450	2300	27	.0193
11500	2300	27	.0273
11550	2300	27	.0657
11600	2300	27	.064

11650	2300	26.6148
11700	2300	26.0535
11750	2300	25.7411
11800	2300	25.446
11850	2300	25.1044
11900	2300	24.774
11950	2300	24.5451
12000	2300	24.4583
12050	2300	24.4771
12100	2300	24.6304
12150	2300	24.7794
12200	2300	24.8846
12250	2300	24.9982
12300	2300	25.1437
12350	2300	25.2754
12400	2300	25.6315
12450	2300	25.8171
12500	2300	26.5378
12550	2300	27.7478
12600	2300	25.3304
12650	2300	22.945
12700	2300	21.8507
12750	2300	21.85
12800	2300	22.8437
12850	2300	22.9995
12900	2300	22.0129
12950	2300	22.1606
13000	2300	22.9563
13050	2300	24.2967
13100	2300	25.1851
13150	2300	25.8767
13200	2300	26.4267
10200	2350	27.8203
10250	2350	27.9063
10300	2350	27.9611
10350	2350	27.9942
10400	2350	27.9018
10450	2350	27.8643
10500	2350	27.7841
10550	2350	27.7128
10600	2350	27.6452
10650	2350	27.5655
10700	2350	27.4533
10750	2350	27.3741
10800	2350	27.2978
10850	2350	27.2194
10900	2350	27.1548
10950	2350	27.1111
11000	2350	27.0756
11050	2350	27.0227
11100	2350	27.015
11150	2350	27.0086

11200 2350 27.0022
11250 2350 26.9996
11300 2350 27.001
11350 2350 27.0011
11400 2350 27.0026
11450 2350 27.0219
11500 2350 27.0421
11550 2350 27.0022
11600 2350 26.8337
11650 2350 26.3543
11700 2350 25.9168
11750 2350 25.6451
11800 2350 25.3905
11850 2350 25.1177
11900 2350 24.8995
11950 2350 24.7067
12000 2350 24.611
12050 2350 24.6938
12100 2350 24.7354
12150 2350 24.8527
12200 2350 24.9515
12250 2350 25.0641
12300 2350 25.1976
12350 2350 25.0446
12400 2350 24.5589
12450 2350 23.5593
12500 2350 23.2331
12550 2350 25.0513
12600 2350 23.7047
12650 2350 21.9362
12700 2350 21.919
12750 2350 22.4158
12800 2350 22.8896
12850 2350 23.0115
12900 2350 22.7342
12950 2350 23.7048
13000 2350 24.6402
13050 2350 25.7846
13100 2350 26.5958
13150 2350 26.9143
13200 2350 27.1049
10200 2400 27.8693
10250 2400 27.9256
10300 2400 27.9738
10350 2400 28.0018
10400 2400 27.8953
10450 2400 27.8541
10500 2400 27.769
10550 2400 27.701
10600 2400 27.6298
10650 2400 27.5612
10700 2400 27.4523

10750	2400	27.3776
10800	2400	27.2902
10850	2400	27.2233
10900	2400	27.1591
10950	2400	27.1174
11000	2400	27.0763
11050	2400	27.0174
11100	2400	27.011
11150	2400	27.007
11200	2400	27.0028
11250	2400	27.0005
11300	2400	27.0029
11350	2400	26.999
11400	2400	26.9845
11450	2400	26.9648
11500	2400	26.9318
11550	2400	26.7858
11600	2400	26.4995
11650	2400	26.0819
11700	2400	25.7789
11750	2400	25.5685
11800	2400	25.3427
11850	2400	25.1288
11900	2400	24.9727
11950	2400	24.8245
12000	2400	24.7899
12050	2400	24.7661
12100	2400	24.8373
12150	2400	24.8764
12200	2400	24.9314
12250	2400	24.987
12300	2400	25.0744
12350	2400	24.2758
12400	2400	23.1543
12450	2400	22.4554
12500	2400	22.0377
12550	2400	22.5407
12600	2400	22.467
12650	2400	21.8392
12700	2400	22.3965
12750	2400	23.0276
12800	2400	22.9666
12850	2400	22.967
12900	2400	23.3273
12950	2400	25.353
13000	2400	26.4818
13050	2400	26.8695
13100	2400	27.1936
13150	2400	27.4197
13200	2400	27.5563
10200	2450	27.932
10250	2450	27.9607

10300 2450 27.9869
10350 2450 28.017
10400 2450 27.8926
10450 2450 27.8194
10500 2450 27.756
10550 2450 27.6909
10600 2450 27.6226
10650 2450 27.5406
10700 2450 27.4306
10750 2450 27.3598
10800 2450 27.289
10850 2450 27.2232
10900 2450 27.1671
10950 2450 27.1238
11000 2450 27.0405
11050 2450 27.0212
11100 2450 27.0089
11150 2450 27.0031
11200 2450 27.0023
11250 2450 27.0024
11300 2450 27.0068
11350 2450 27.002
11400 2450 26.9392
11450 2450 26.8549
11500 2450 26.7371
11550 2450 26.5422
11600 2450 26.2591
11650 2450 25.9537
11700 2450 25.7071
11750 2450 25.5152
11800 2450 25.3232
11850 2450 25.1204
11900 2450 24.9682
11950 2450 24.9038
12000 2450 24.8395
12050 2450 24.8118
12100 2450 24.852
12150 2450 24.819
12200 2450 24.7938
12250 2450 24.7213
12300 2450 24.5403
12350 2450 23.5466
12400 2450 22.1808
12450 2450 21.8715
12500 2450 21.679
12550 2450 21.7413
12600 2450 21.8005
12650 2450 21.8506
12700 2450 22.6689
12750 2450 23.12
12800 2450 23.0436
12850 2450 23.0324

12900	2450	24.2904
12950	2450	26.2285
13000	2450	27
13050	2450	27.3273
13100	2450	27.5426
13150	2450	27.7239
13200	2450	27.8435
10200	2500	27.9892
10250	2500	27.9796
10300	2500	27.9763
10350	2500	28.0367
10400	2500	27.8652
10450	2500	27.8116
10500	2500	27.747
10550	2500	27.6592
10600	2500	27.5902
10650	2500	27.5265
10700	2500	27.4183
10750	2500	27.3506
10800	2500	27.2851
10850	2500	27.2819
10900	2500	27.2386
10950	2500	27.1767
11000	2500	27.0532
11050	2500	27.0264
11100	2500	27.0029
11150	2500	26.998
11200	2500	26.9993
11250	2500	26.9975
11300	2500	26.9923
11350	2500	26.9586
11400	2500	26.8642
11450	2500	26.7334
11500	2500	26.5646
11550	2500	26.3469
11600	2500	26.0953
11650	2500	25.8508
11700	2500	25.6455
11750	2500	25.4719
11800	2500	25.2873
11850	2500	25.1219
11900	2500	24.9932
11950	2500	24.9134
12000	2500	24.8747
12050	2500	24.835
12100	2500	24.8076
12150	2500	24.6965
12200	2500	24.4892
12250	2500	24.2492
12300	2500	23.7062
12350	2500	22.7289
12400	2500	22.0151

12450	2500	21.7331
12500	2500	21.7131
12550	2500	21.9352
12600	2500	22.2187
12650	2500	22.5296
12700	2500	23.0786
12750	2500	23.1748
12800	2500	23.2595
12850	2500	23.8666
12900	2500	25.4983
12950	2500	26.5079
13000	2500	27.0312
13050	2500	27.4139
13100	2500	27.6405
13150	2500	27.8939
13200	2500	28.0415
10200	2550	28.0003
10250	2550	28.0396
10300	2550	28.1161
10350	2550	28.2825
10400	2550	28.3284
10450	2550	28.0002
10500	2550	27.7849
10550	2550	27.6383
10600	2550	27.5735
10650	2550	27.5108
10700	2550	27.4067
10750	2550	27.3409
10800	2550	27.3769
10850	2550	27.3699
10900	2550	27.2905
10950	2550	27.1301
11000	2550	27.0821
11050	2550	27.0412
11100	2550	27.0038
11150	2550	26.9945
11200	2550	26.9934
11250	2550	26.9889
11300	2550	26.9664
11350	2550	26.9049
11400	2550	26.7843
11450	2550	26.6158
11500	2550	26.4181
11550	2550	26.1934
11600	2550	25.971
11650	2550	25.7785
11700	2550	25.5918
11750	2550	25.4404
11800	2550	25.2648
11850	2550	25.1187
11900	2550	25.0013
11950	2550	24.9352

12000	2550	24.8799
12050	2550	24.8014
12100	2550	24.7432
12150	2550	24.5457
12200	2550	24.1738
12250	2550	23.5275
12300	2550	22.5965
12350	2550	22.0112
12400	2550	22.1049
12450	2550	21.893
12500	2550	21.8529
12550	2550	22.2173
12600	2550	22.8429
12650	2550	23.0782
12700	2550	23.1179
12750	2550	23.1567
12800	2550	23.3253
12850	2550	23.8618
12900	2550	25.0432
12950	2550	26.0824
13000	2550	26.7886
13050	2550	27.2762
13100	2550	27.5988
13150	2550	27.8846
13200	2550	28.0812
10200	2600	27.9968
10250	2600	28.1637
10300	2600	28.3126
10350	2600	28.7116
10400	2600	28.9466
10450	2600	28.842
10500	2600	28.56
10550	2600	28.0104
10600	2600	27.6799
10650	2600	27.6252
10700	2600	27.5594
10750	2600	27.5109
10800	2600	27.4761
10850	2600	27.4375
10900	2600	27.2442
10950	2600	27.1881
11000	2600	27.1093
11050	2600	27.0674
11100	2600	27.0179
11150	2600	26.9933
11200	2600	26.9885
11250	2600	26.9845
11300	2600	26.9597
11350	2600	26.8816
11400	2600	26.7249
11450	2600	26.5114
11500	2600	26.2902

11550	2600	26.0944
11600	2600	25.8894
11650	2600	25.7078
11700	2600	25.5384
11750	2600	25.4038
11800	2600	25.2503
11850	2600	25.118
11900	2600	25.0221
11950	2600	24.9416
12000	2600	24.8599
12050	2600	24.7794
12100	2600	24.693
12150	2600	24.4708
12200	2600	24.1119
12250	2600	23.0407
12300	2600	22.3571
12350	2600	22.6892
12400	2600	22.9137
12450	2600	22.0859
12500	2600	21.9355
12550	2600	22.3863
12600	2600	22.9427
12650	2600	23.0991
12700	2600	23.0672
12750	2600	23.0462
12800	2600	23.2186
12850	2600	23.6203
12900	2600	24.177
12950	2600	25.4804
13000	2600	26.5098
13050	2600	27.1518
13100	2600	27.5849
13150	2600	27.9427
13200	2600	28.0372
10200	2650	28.3892
10250	2650	28.4878
10300	2650	28.7545
10350	2650	28.9616
10400	2650	29.3214
10450	2650	29.8326
10500	2650	29.1404
10550	2650	28.6696
10600	2650	28.3514
10650	2650	27.6095
10700	2650	27.583
10750	2650	27.5472
10800	2650	27.4927
10850	2650	27.321
10900	2650	27.2981
10950	2650	27.2337
11000	2650	27.1641
11050	2650	27.0959

11100	2650	27.0442
11150	2650	26.9956
11200	2650	26.9907
11250	2650	26.9878
11300	2650	26.984
11350	2650	26.9102
11400	2650	26.6999
11450	2650	26.4165
11500	2650	26.14
11550	2650	25.9826
11600	2650	25.8051
11650	2650	25.6394
11700	2650	25.502
11750	2650	25.3715
11800	2650	25.228
11850	2650	25.1092
11900	2650	25.0194
11950	2650	24.9403
12000	2650	24.8617
12050	2650	24.7724
12100	2650	24.6607
12150	2650	24.4176
12200	2650	23.9417
12250	2650	22.8543
12300	2650	22.7257
12350	2650	23.0095
12400	2650	22.9598
12450	2650	22.2809
12500	2650	22.1271
12550	2650	22.7451
12600	2650	23.058
12650	2650	23.0714
12700	2650	23.009
12750	2650	22.9524
12800	2650	23.2286
12850	2650	23.808
12900	2650	24.4427
12950	2650	25.4288
13000	2650	26.3891
13050	2650	27.0745
13100	2650	27.5515
13150	2650	27.8566
13200	2650	28.1604
10200	2700	28.8645
10250	2700	28.9732
10300	2700	29.2271
10350	2700	29.5702
10400	2700	29.7311
10450	2700	29.9829
10500	2700	29.4035
10550	2700	29.2908
10600	2700	28.6768

10650	2700	28.0563
10700	2700	27.6087
10750	2700	27.5567
10800	2700	27.4854
10850	2700	27.3145
10900	2700	27.3136
10950	2700	27.2898
11000	2700	27.2286
11050	2700	27.1668
11100	2700	27.098
11150	2700	27.0538
11200	2700	27.0244
11250	2700	26.9957
11300	2700	27.0331
11350	2700	26.9717;

Apéndice D: Archivo de datos para la aplicación de la Programación Genética.

D.1 optimización Regional.

D.1.1 Región 1.

ARCHIVO: CXYZR1.DAT

x(m) y(m) z(msnm)

13000 2700 29 54739

13050 2700 29.60763

13150 2700 29.72811

13200 2700 29.78835

13400 2700 30.02931

13500 2700 30 14979

13550 2700 30 21003

13600 2700 30 27027

13650 2700 30.33051

13700 2700 30.39075

13750 2700 30.45099

13850 2700 30 57147

13100 2750 19 60844

13350 2750 19 71275

13400 2750 30.02931

D.1.2 Región 2.

ARCHIVO: CXYZR2.DAT

x(m) y(m) z(msnm)

10250 1900 28 110100

10300 1900 27 675630

10400 1900 29 063000

10450 1900 27 583300

10550 2000 29 501850

10555 2000 27 588898

10650 2000 29 499240

10750 2000 29 491530

10800 2000 27 510610

10850 2000 29 479180

10900 2100 29 254950

10500 2200 29 471090

10350 2300 28 378460

10700 2500 29 278490

10200 2800 29 034790

D.2 Optimización Global.

ARCHIVO: CXYZ100.DAT

x(m) y(m) z(msnm)

10200 1900 34 0314

10250 1900 34 0165

10300 1900 33 0051

10350 1900 32 9995

10400 1900 32 9787

10450 1900 32 9385

10500 1900 31 8757
10550 1900 31.7974
10600 1900 31.7069
10650 1900 30 6113
10700 1900 30 5003
10750 1900 30.3924
10800 1900 29.2981
10850 1900 29 2095
10900 1900 29.131
10950 1900 29.0771
11000 1900 28 0201
11050 1900 26.9839
11100 1900 26.9622
11150 1900 26 9812
11200 1900 26 9623
11250 1900 26 9425
11300 1900 26 9174
11350 1900 26 8908
11400 1900 25 8289
11450 1900 25.768
11500 1900 25.7025
11550 1900 25.6204
11600 1900 24 5059
11650 1900 24.3294
11700 1900 24 0635
11750 1900 23 2936
11800 1900 23 8601
11850 1900 22.7945
11900 1900 24 0361
11950 1900 23 4612
12000 1900 22 1013
12050 1900 22 6723
12100 1900 22 9985
12150 1900 22 1231
12200 1900 21 1005
12250 1900 21.0357
12300 1900 21 0008
12350 1900 21 0138
12400 1900 20 1277
12450 1900 20 2041
12500 1900 20.166
12550 1900 20 8367
12600 1900 20 2975
12650 1900 19 0195
12700 1900 19.0049
12750 1900 19 026
12800 1900 19.0868
12850 1900 19 1714
12900 1900 19 2312
12950 1900 18 1624
13000 1900 18 7253

13050 1900 18 7086
13100 1900 18 8521
13150 1900 18 7732
13200 1900 18 7814
10200 1950 36 0429
10250 1950 36 0258
10300 1950 36 013
10350 1950 35 0023
10400 1950 34 9847
10450 1950 34 9433
10500 1950 34 8782
10550 1950 33 7973
10600 1950 33 7037
10650 1950 32 6001
10700 1950 32 4799
10750 1950 32 3781
10800 1950 31 2793
10850 1950 31 1878
10900 1950 30 108
10950 1950 30 0534
11000 1950 30 0004
11050 1950 29 9752
11100 1950 29 9638
11150 1950 29 9911
11200 1950 28 9795
11250 1950 28 9657
11300 1950 28 9392
11350 1950 26 9073
11400 1950 26 8603
11450 1950 26 7904
11500 1950 26 7109
11550 1950 26 6045
11600 1950 26 462
11650 1950 26 252
11700 1950 25 9616
11750 1950 25 5987
11800 1950 24 5883
11850 1950 25 0116
11900 1950 24 7955
11950 1950 24 8097
12000 1950 24 0607
12050 1950 23 9654
12100 1950 23 8021