Optimización estructural por medio del algoritmo genético

Conceptos teóricos y aplicaciones a funciones matemáticas simples (Primera parte)

Giovanni Martínez Martínez

Autor

GIOVANNI MARTÍNEZ MARTÍNEZ
Ingeniero Civil. Especialista en Análisis y Diseño Estructural, Candidato a Magíster en Ingeniería Sismorresistente, Universidad EAFIT, Diplomado en Didáctica y Pedagogía, Docente de tiempo completo y Coordinador del Semillero GRIDIC - Grupo de Investigación de Ingeniería Civil, línea Estructuras y Concretos del Politécnico Colombiano Jaime Isaza Cadavid, Correo electrónico: gjmartinez@elpoli.edu.co

Resumen

Los algoritmos genéticos tienen su base en la teoría de supervivencia de las especies de Darwin. Estos algoritmos han sido usados satisfactoriamente en genética y recientemente en una variedad de problemas de optimización en el campo de la ingeniería estructural. Procesos estocásticos generan una población inicial de individuos, entonces se aplican los principios de selección natural del más apto para mejorar los diseños, con base en restricciones dadas. Se tiene en cuenta las interfaces de análisis y diseño, y las operaciones básicas de selección, cruzamiento, mutación y parámetros de escalado.

Palabras claves

Cruzamiento - escalado - genético - mutación - optimización - selección

Abstract

The genetic algorithms have their base in the theory of survival of Darwin's species. These algorithms have been used satisfactorily in genetic and recently in a variety of problems of optimization in the field of the structural engineering. Stochastic processes generate an initial population of individuals, then the principles of natural selection of the most capable are applied to improve the designs, with base in given restrictions. One keeps in mind the analysis interfaces and design, and the basics operations of selection, crossing, mutation and parameters of having climbed.

Key words

Crossing - climbed - genetic - mutation - optimization - selection.

Optimización estructural por medio del algoritmo genético

Conceptos teóricos y aplicaciones a funciones matemáticas simples (Primera parte)

Giovanni Martínez Martínez

|||| POLITÉCNICA No. 1 | Medellín, junio - octubre de 2005, p.p. 7-22

1. Introducción

La Mecánica Natural Darviniana de supervivencia de los seres mejor adaptados, puede ser aplicada a la resolución de diferentes problemas en Ingeniería y otros campos. Los organismos vivos poseen destreza consumada en la resolución de problemas y manifiestan una versatilidad capaz de avergonzar a los programas más refinados. Los investigadores más pragmáticos consideran que lo que hay que hacer no es envidiar la eficacia de la evolución, sino emularla.

La selección natural elimina uno de los mayores obstáculos que entorpecen el diseño de programas: la especificación por adelantado de todas las características y peculiaridades de un problema y de las acciones requeridas para atenderlas. Si se consiguiera incorporar mecanismos evolutivos a los programas, podrían "criarse" programas aptos para la resolución de diversos problemas.

Y a decir verdad, algoritmos de este tipo, comúnmente denominados ALGORITMOS GENÉTICOS (A.G.), han demostrado ya su capacidad para abrir nuevas brechas en el diseño de sistemas tan complejos como los motores de reacción, sistemas de refrigeración, análisis y diseño estructural, sistemas de ventilación, redes de tráfico, hidráulicas y telefónicas, etc. Por lo tanto, un Algoritmo Genético es un programa que intenta resolver un problema específico mediante un procedimiento aleatorio de selección natural. Con el paso de las generaciones, los descendientes de los más dotados (que son parecidos pero no iguales a sus antecesores) deben luchar entre sí para seguir con vida, haciendo que de esta forma se desarrollen cada vez más las condiciones impuestas. En resumen, los Algoritmos Genéticos intentan parecerse a la forma en que la naturaleza escoge entre un conjunto inicial de seres a los más dotados para sobrevivir, y va desechando a los menos dotados.

Los Algoritmos Genéticos permiten la exploración de un abanico mucho más amplio de posibles soluciones que los programas tradicionales. La impresionante capacidad de cálculo de los computadores actuales y su continuo crecimiento, permiten realizar en un período razonable de tiempo la optimización de problemas ingenieriles.

La forma en que funciona un Algoritmo Genético básicamente es:

Se crea una "población", es decir, un conjunto de soluciones posibles (seleccionándolas por ejemplo de un archivo de datos de áreas de secciones transversales, diámetros de tuberías, etc.), que se pueden escoger con cierto criterio, o se pueden dejar simplemente al azar. Luego, la población es ordenada de acuerdo con ciertas evaluaciones matemáticas.

A continuación, cada miembro de la población se juega la vida en una especie de "ruleta rusa", donde la probabilidad de sobrevivir es una función de la posición que haya ocupado ese ser en el ordenamiento de la población. En este juego, cierta cantidad de habitantes muere, dejando sus casillas disponibles para los hijos de los sobrevivientes. Estas casillas son entonces asignadas a copias (hijos) de los mejores miembros de la población.

■ Cada miembro de la población se juega la vida en una especie de "ruleta rusa", donde la probabilidad de sobrevivir es una función de la posición que haya ocupado ese ser en el ordenamiento de la población.

Inmediatamente después, la población es sometida a una ligera mutación, con lo cual los nuevos habitantes serán parecidos a sus padres, pero no necesariamente iguales. Esta mutación permite la aparición de nuevo material genético, en otras palabras, individuos con nuevas características ocultas en el proceso de selección. En este momento, se completa lo que llamaremos una generación, y el ciclo se repite durante tantas generaciones como se desee.

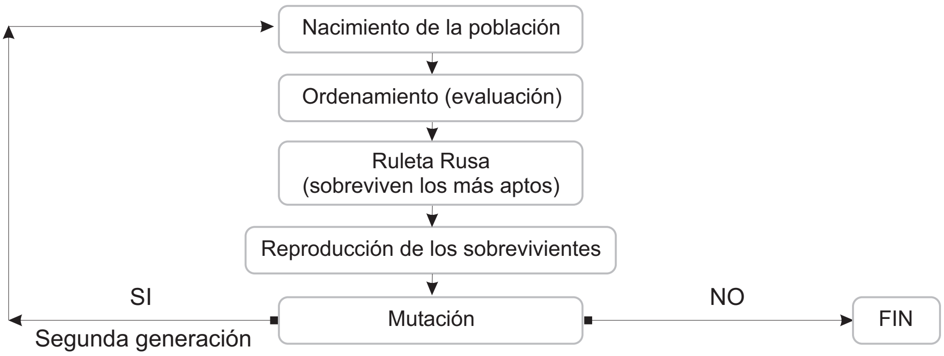
La selección constituye un proceso sencillo: cuando un organismo falla en alguna prueba de idoneidad, perece. Si por ejemplo un programa tiene la función de escoger las barras de una armadura con diámetro mínimo D, o con desplazamientos máximos permitidos Δ, las que estén por debajo de ese diámetro, o por encima del desplazamiento que restringe, simplemente son rechazadas.

El método del Algoritmo Genético, como la propia naturaleza, es discreto y encaja directamente en la vida real, donde para optimizar, por ejemplo, en el campo del diseño de obras civiles, se dispone de unos determinados parámetros o condiciones para llegar a un diseño óptimo (esfuerzos, diámetros, desplazamientos, presiones, caudales, etc.)

En el campo del diseño estructural, un diseño estructural "posible", es una combinación de las variables que satisfacen las restricciones del diseño. El conjunto completo de diseños posibles, usualmente un número grande, conforma el "espacio de diseños posibles" y el progreso hacia el diseño óptimo implicará algún tipo de búsqueda de este espacio.

La búsqueda puede ser de carácter Determinística donde se emplean métodos algorítmicos, como por ejemplo, métodos del gradiente (en general los métodos de programación matemática), o puede ser Estocástica, donde se introduce una componente aleatoria.

Si la búsqueda es Determinística y Estocástica, usualmente es posible mejorar la confiabilidad del resultado, donde confiabilidad significa proximidad al óptimo. El Algoritmo Genético, combina muy bien estos aspectos, pues el trabajar con variables de tipo discreto y no continuas como la vida misma, hace que sus procedimientos de trabajo y sus mismos resultados sean bastante ajustados a la vida real.



2. Representación binaria de la población

La forma de obtener por evolución un conjunto de reglas clasificadoras capaces de resolver un problema dado, debe comenzar, logrando la representación cromosomal de la población.

Se parte de una población de Ristras o Cadenas Binarias (unos y ceros).

Cadena 1	1 0 1 0 1 1 1 1 0 1 1 0 0 1 0 1	representa un valor X
Cadena 2	0 0 0 1 0 1 1 0 1 0 1 1 1 0 0 0	representa un valor Y

Cada cadena representa un dato cualquiera, que puede ser por ejemplo el área de una sección o de una barra de refuerzo, el diámetro de una tubería, un esfuerzo admisible, algún dato o restricción de tipo geométrico, etc.

Se procede al apareamiento aleatorio (sitio arbitrario de corte), dando como resultado una nueva pareja de datos:

1 0 1 0 1 1 1 1 0 1	1 0 0 1 0 1	
0 0 0 1 0 1 1 0 1 0	1 1 1 0 0 0	
1 0 1 0 1 1 1 1 0 1	1 0 0 1 0 1	CORTE GENÉTICO
0 0 0 1 0 1 1 0 1 0	1 1 1 0 0 0	
1 0 1 0 1 1 1 1 0 1	1 1 1 0 0 0	CRUCE GENÉTICO
0 0 0 1 0 1 1 0 1 0	1 0 0 1 0 1	

1 0 1 0 1 1 1 1 0 1 1 1 1 0 0 0	Cadena nueva - nuevo valor
0 0 0 1 0 1 1 0 1 0 1 0 0 1 0 1	Cadena nueva - nuevo valor

3. Conceptos básicos

El cruzamiento constituye el mecanismo fundamental de la reorganización genética. Los cromosomas se adosan, para intercambiar luego las porciones de sus códigos genéticos situados a partir del punto de cruce.

La mutación es otra fase sencilla del proceso. Se hacen pequeñas alteraciones al código de cada individuo, en forma aleatoria. Según el problema que se trate, las cadenas de mayor calidad o aptitud, y de acuerdo a unas reglas de selección (probabilidad), se aparean; las inferiores, sencillamente, perecen.

Con el transcurso de las generaciones irán predominando las cadenas asociadas a soluciones cada vez más perfectas. Además, el proceso de apareamiento combina sin cesar estas ristras de nuevas formas, generando soluciones más y más refinadas.

El Algoritmo Genético explora las regiones de más alto rendimiento del espacio de soluciones porque las sucesivas generaciones de reproducción, cruzamiento y mutación, generan un número creciente de ristras pertenecientes a ellas. El Algoritmo favorece que las ristras más aptas asuman roles progenitores y, por esta razón, las cadenas superiores al promedio tendrán mayor descendencia en la generación siguiente. De hecho, el número de ristras de una región dada aumenta a ritmo proporcional a la estimación estadística de la idoneidad de esa región.

Un estadístico tendría que evaluar docenas de muestras tomadas de millares o millones de regiones para determinar la adecuación o idoneidad media de cada región. El Algoritmo Genético logra alcanzar el mismo resultado con muchísimas menos cadenas y prácticamente con muy pocos cálculos. Por lo tanto, un Algoritmo Genético que manipule una población de unos cuantos millares de cadenas, está realmente tomando muestras de un número de regiones enormemente mayor. Tal paralelismo implícito, proporciona al Algoritmo Genético su ventaja sobre otros procesos de resolución de problemas.

El Algoritmo Genético basado en la selección natural con una aleatoriedad condicionada en sus procesos, converge rápidamente hacia la estructura óptima habiendo examinado sólo una pequeñísima fracción de individuos del espacio de búsqueda.

Como un ejemplo demostrativo, supongamos individuos o cromosomas de 40 bits de longitud. En general se consigue la convergencia con una población de 100 individuos y analizando unas 50 generaciones. Esto supone examinar $100 \times 50 = 5000$ individuos de una población total de $240 = 1.00995 \times 1012$, lo que equivale a escoger un individuo en una población de 220 millones y resultar que éste es el mejor. Nuestro sentido común parece decirnos que esto es imposible, pero la naturaleza funciona así.

Aunque remeden los efectos de la selección natural, los Algoritmos Genéticos han operado hasta ahora a escala mucho menor que la evolución biológica. Conforme vayan siendo más corrientes los computadores de funcionamiento en paralelo a gran escala (múltiples CPU conectados en red), resultará factible hacer evolucionar poblaciones informáticas cuyo tamaño se aproxime más a los de las especies naturales.

De hecho, el Algoritmo Genético se adapta muy bien a tales máquinas. Cada procesador puede dedicarse a una única ristra, ya que las operaciones del algoritmo se concentran en ristras individuales o, a lo sumo, en parejas durante el cruzamiento. En consecuencia, puede procesarse en paralelo la población entera.

4. Filosofía del método

Para resolver un problema con un Algoritmo Genético, se deben concretar los siguientes pasos:

1. La representación Cromosomal de la población.
2. La creación de una población inicial aleatoria.
3. Una función de evaluación, que interprete la acción del "medio ambiente" sobre cada uno de los individuos de la población (evaluar todas las restricciones y decidir cuáles cadenas se cruzan o no).
4. Los operadores genéticos (cruce, mutación, etc.) que alteren la composición de los hijos.
5. La evaluación de los parámetros de control (tamaño de la población, probabilidad de reproducción, probabilidad de mutación, etc.).

5. Ejemplo de aplicación - minimización de una función matemática simple

Minimizar la siguiente ecuación:

$F(x) = 0.7 X_1^3 + 1.5 X_2^2 + 10 X_3$

F(x) = Función objetivo (objetivo a minimizar)

sujeta a las siguientes restricciones:

$X_1 > 0$

$X_2 > 0$

$X_3 > 0$

$X_1 + X_2 + X_3 \geq 15$

Número de parámetros (NP) = 3 (# de incógnitas)

5.1 Representación cromosomal

Se representa en forma de cadenas de bits (cromosomas), la población total (Generación 0). Se elige una longitud de cadena de 5. Por lo tanto, se pueden representar 32 valores posibles (# 0 al # 31).

$2^{LC} = 2^5 = 32$ donde: LC = Longitud de la cadena

Los 32 valores se representarían así:

NÚMERO	CADENA
0	00000
1	00001
2	00010
3	00011
4	00100
5	00101
.	
29	11101
30	11110
31	11111

La representación binaria del número 13 sería:

$$\begin{matrix} (0 & 1 & 1 & 0 & 1) \\ 2^4 & 2^3 & 2^2 & 2^1 & 2^0 \end{matrix}$$

Pero, donde hay un cero (0), el valor 2 x = 0; es decir, 24 =21=0, y los únicos valores a evaluar son:

$2^3 + 2^2 + 2^0 = 8 + 4 + 1 = 13$

Si tuviéramos una cadena de 7 bits, el máximo número representado es:

$2^7 = 128$ (# 0 al # 127)

El número 84 sería:

(1 0 1 0 1 0 0)

2^6 2^5 2^4 2^3 2^2 2^1 2^0

a evaluar serían

$= 2^6 + 2^4 + 2^2 = 64 + 16 + 4 = 84$

5.2 Cálculo de la aptitud

El proceso básico busca incrementar la aptitud (fitness), minimizando la función objetivo (proceso inverso):

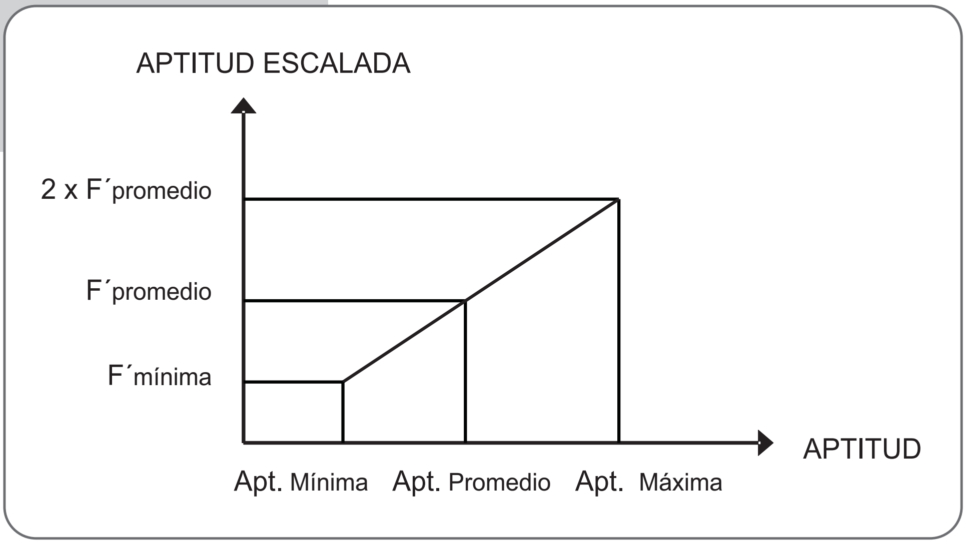
Aptitud = C – F(x)

donde C es una constante suficientemente grande, para excluir valores negativos de la Aptitud. A mayor aptitud, significa que F(x) es mínima, que es lo que se trata de conseguir en el algoritmo (tabla 1 generación 0).

5.3 Cálculo de la aptitud escalada y número de copias

El modelo de Escalado Lineal de Goldberg (2), que se aplica en este método, sirve para mejorar las ratas de producción de los mejores individuos y desalienta a los individuos más débiles. Es decir, favorece las mejores aptitudes y desfavorece las peores. Así se calcula lo que se llama, la Aptitud Escalada (figura 1).

Figura 1. Escalamiento Lineal de la Aptitud



Con la Aptitud Escalada, se calcula el Número de Copias (o número de veces que un individuo puede ser padre), de la siguiente manera:

Sea F'i la Aptitud Escalada del individuo **i y $\sum F'_i$** la sumatoria de las aptitudes escaladas de la generación actual, entonces la probabilidad de selección de un individuo i es:

PROBABILIDAD $_{\text{SELECCIÓN}}$ i (# COPIAS $_{\text{INDIVIDUO } i}$) = F'i / $\sum$ F'i

Si se tiene una población de 20 individuos (20 tripletas de números, ya que son 3 las incógnitas, para el ejemplo), el número de copias del individuo i será:

COPIAS $_{\text{INDIVIDUO } i}$ $= 20 \times F'_i / \sum F'_i$

El número de copias del individuo 12 (ver tabla 1), será:

COPIAS $_{\text{INDIVIDUO 12}}$ $= 20 \times 25071.86 / 289345.75 = 2$

O sea, el individuo 12 va a ser padre 2 veces en la siguiente generación; se debe cumplir que:

COPIAS de la Generación = # de individuos de la población

COPIAS generación i = 20 (para nuestro ejemplo)

Un valor de CERO (=0) en la Aptitud, significa que se ha violado alguna de las restricciones:

Tabla 1 - Procesamiento generación 0

GENERACIÓN 0

Ind.	X1 Bin.	X1 dec	X2 Bin.	X2 dec	X3 Bin.	X3 dec	Aptitud	Aptitud Escalada	Número copias
1	00100	4	11010	26	11001	25	18691.20	24478.00	2
2	00100	4	00100	4	00001	1	0.00*	0.00*	0
3	00011	3	01100	12	10110	22	19545.10	26217.02	2
4	00111	7	10100	20	11111	31	18849.90	24801.20	2
5	01100	12	10101	21	10110	22	17908.90	22884.80	2
6	11100	28	01000	8	10110	22	4317.60	-4794.72	0
7	00100	4	11010	26	11011	27	18671.20	24437.27	2
8	11001	25	00111	7	10010	18	8809.00	4352.29	0
9	01000	8	00010	2	00001	1	0.00 *	0.00 *	0
10	01111	15	11111	31	11110	30	15896.00	18785.40	1
11	11010	26	10011	19	10100	20	6955.30	577.12	0
12	00110	6	10110	22	01110	14	18982.80	25071.86	2
13	00111	7	10001	17	11100	28	19046.40	25201.39	2
14	01111	15	11011	27	01100	12	16424.00	19860.71	1
15	11001	25	00011	3	10101	21	8839.00	4413.39	0
16	01110	14	11011	27	01010	10	16885.70	20800.99	1
17	11000	24	00010	2	01110	14	10177.20	7138.72	0
18	01110	15	01001	9	00110	6	17456.00	21962.44	2
19	11000	24	10001	17	01101	13	9759.70	6288.45	0
20	10011	19	00111	7	10001	17	14955.20	16869.41	1
SUMATORIA							**262170.19**	**289345.75**	**20**

Resumen Generación cero (0):·
· Sumatoria Número de Copias = 20
· Sumatoria Aptitud = 262170.19 · Sumatoria Aptitud Escalada = 289345.75
· Promedio Aptitud = 13108.51 · Promedio Aptitud Escalada = 14467.29
· Aptitud Máxima = 19545.10 · Aptitud Escalada Máxima = 26217.02

* Restricción violada

5.4 Apareamiento y mutación

Una vez seleccionados los padres en función de las probabilidades de cruce PC, pasarán a formar parte de la nueva población y se cruzan como se indicó anteriormente:

Sean 2 cadenas de N bits:

Padre 1 = (a1 a2 a3 a4 a5 a6 a7 a8 a9 a10)

Padre 2 = (b1 b2 b3 b4 b5 b6 b7 b8 b9 b10)

Se elige aleatoriamente un número K en le intervalo (1, N-1), y se crean 2 hijos intercambiando las cadenas en el K elegido; si K=6 se tiene:

hijo 1 = (**a1 a2 a3 a4 a5 a6** b7 b8 b9 b10)

hijo 2 = (b1 b2 b3 b4 b5 b6 **a7 a8 a9 a10**)

Continuando con la exploración y adaptación de los principios de la genética natural, el último paso de las operaciones de transformación lo constituye el proceso de mutación, que consiste en elegir aleatoriamente un número K del intervalo (0, N-1), y si a la posición K del cromosoma le corresponde un 1 se coloca un 0 y viceversa.

La mutación aumenta las probabilidades de ampliar el espacio de búsqueda a zonas que de otra manera quedarían ocultas, es decir, estimulan el desarrollo de nuevo material genético.

Al igual que en la operación de cruce existe un parámetro que regula la probabilidad de mutación de los cromosomas; el proceso se controla, prescribiendo un porcentaje o probabilidad de mutación PM, y que generalmente es del orden de 0.005 (0.5%).

Es esencial conservar un equilibrio en la intensidad de la mutación: mutaciones demasiado pequeñas alargan los procesos y hacen que se necesiten más generaciones y en muchos casos los estancan, y por otro lado, mutaciones muy grandes hacen que se pierda la información de los mejores individuos. También, esta probabilidad de mutación depende del tipo de trabajo a desarrollar y de los logros que se deseen en el trabajo.

En la tabla 2, se puede apreciar cómo es el proceso de apareamiento y los nuevos números que se generan para la próxima generación (generación 1).

Tabla 2 – Resultados apareamiento generación 0

X1				X2				X3			
Padres	Cx	Cad.	N m.	Padres	Cx	Cad.	N m.	Padres	Cx	Cad.	N m.
20,7	2	10000	16	1,5	1	11011	27	12,16	5	01010	10
20,7	2	00111	7	1,5	1	10100	20	12,16	5	01110	14
10,3	3	01011	11	1,3	2	11000	24	4,1	2	11101	29
10,3	3	00111	7	1,3	2	01110	14	4,1	2	11011	28
7,4	5	00111	7	3,14	1	01101	13	13,5	5	10110	22
7,4	5	00100	4	3,14	1	11010	26	13,5	5	11100	28
4,3	1	00111	7	7,20	5	00111	7	18,14	3	00100	4
4,3	1	00011	3	7,20	5	11010	26	18,14	3	01110	14
14,16	3	01110	14	12,13	4	10001	17	3,18	3	10110	22
14,16	3	01111	15	12,13	4	10110	22	3,18	3	00110	6
5,18	1	01101	13	18,10	2	01011	11	20,10	3	10111	23
5,18	1	01110	14	18,10	2	11101	29	20,10	3	11001	25
1,12	2	00110	6	12,16	5	11011	27	13,1	2	11101	29
1,12	2	00100	4	12,16	5	10110	22	13,1	2	11000	24
18,5	4	01100	12	5,18	5	01001	9	4,7	3	11011	27
18,5	4	01111	15	5,18	5	10101	21	4,7	3	11111	31
12,13	2	00111	7	4,7	3	10010	18	3,7	5	11011	27
12,13	2	00110	6	4,7	3	11100	28	3,7	5	10110	22
1,13	3	00111	7	4,13	3	10001	17	15,5	2	01110	14
1,13	3	00110	6	4,13	3	10100	20	15,5	2	10110	22

Después del apareamiento y mutación, los nuevos números generados por la generación 0, comenzarán de nuevo el proceso (generación 1 en adelante), implicando los mismos pasos vistos hasta ahora:

- · Evaluación de la Aptitud
- · Escalado lineal de la Aptitud
- · Selección y Nº copias
- · Cruce y Apareamiento

En la tabla 3, se aprecian los resultados de las generaciones 10 y 14.

Tabla 3 – Resultados generaciones 10 y 14

Generación 10					**Generación 14**				
No	**X_1**	**X_2**	**X_3**	**Aptitud**	**No**	**X_1**	**X_2**	**X_3**	**Aptitud**
1	4	6	10	19801.2	1	4	4	10	19831.2
2	4	4	11	19821.2	2	4	5	10	19817.7
3	**4**	**4**	**8**	**19851.2**	3	4	4	8	19851.2
4	3	5	12	19823.6	4	4	5	8	19837.7
5	3	10	12	19711.1	5	4	4	14	19791.2
6	4	5	13	19787.7	6	3	5	8	19863.6
7	4	4	11	19821.2	7	2	5	9	19866.9
8	4	4	12	19811.2	**8**	**2**	**4**	**9**	**19880.4**
9	5	8	10	19716.5	9	4	4	8	19851.2
10	2	6	9	19850.4	10	4	5	11	19807.7
11	4	4	27	19661.2	11	4	4	10	19831.2
12	6	5	24	19571.3	12	4	4	11	19821.2
13	4	4	10	19831.2	13	3	5	11	19833.6
14	4	4	10	19831.2	14	3	4	8	19877.1
15	3	9	10	19759.6	15	4	4	9	19841.2
16	4	6	10	19801.2	16	4	4	10	19831.2
17	6	4	24	19548.8	17	3	4	9	19867.1
18	4	5	3	0.0	18	3	4	8	19877.1
19	4	5	10	19817.7	19	3	4	10	19857.1
20	4	4	9	19841.2	20	4	4	10	19831.2
$X_1 = 4$ $X_2 = 4$ $X_3 = 8$					**$X_1 = 2$ $X_2 = 4$ $X_3 = 9$**				

A partir de aquí el ejercicio empieza a converger, el valor 19880.4 en la aptitud se vuelve dominante, y siempre predominará sobre los demás valores (proceso de convergencia). Unas generaciones más adelante como se puede apreciar, a partir de la generación 50 (tabla 4), los valores de 2, 4 y 9 se vuelven constantes, y por más generaciones que se quiera correr, estos valores siempre se mantendrán (lo mismo que sus aptitudes).

Tabla 4 – Resultados generación 50

Generación 50				
No	**X_1**	**X_2**	**X_3**	**Aptitud**
1	2	4	9	19880.4
2	2	4	9	19880.4
3	2	4	9	19880.4
4	2	4	9	19880.4
5	2	4	9	19880.4
6	2	4	9	19880.4
7	2	4	9	19880.4
8	2	4	9	19880.4
9	2	4	9	19880.4
10	2	4	9	19880.4
11	2	4	9	19880.4
12	2	4	9	19880.4
13	2	4	9	19880.4
14	2	4	9	19880.4
15	2	4	9	19880.4
16	2	4	9	19880.4
17	2	4	9	19880.4
18	2	4	9	19880.4
19	2	4	9	19880.4
20	2	4	9	19880.4

Con X1=2, X2=4 y X3=9 se evalúa la ecuación para comprobar si en efecto es el mínimo posible:

$F(x) = 0.7 X_1^3 + 1.5 X_2^2 + 10 X_3$

$F(x) = 0.7*(2)^3 + 1.5*(4)^2 + 10*(9) = 119.6$

$X1 + X2 + X3 \geq 15$

$2 + 4 + 9 \geq 15$

Y surge la pregunta, ¿es el mínimo posible?

En general, como los procesos son en su mayoría aleatorios, un Algoritmo Genético producirá resultados diferentes cada vez que es ejecutado. Es preferible correr el programa varias veces con un número aceptable de generaciones que correrlo una sola vez con un número exagerado de generaciones.

En una segunda corrida (tabla 5), con una semilla de inicio diferente lo mismo que el tamaño de la población, se arrojaron los siguientes resultados:

X 1 = 2

X 2 = 3

X 3 = 10

$F(x) = 0.7*(2)^3 + 1.5*(4)^2 + 10*(9) = 119.1$

$X\,1 + X\,2 + X\,3 \geq 15$

$2 + 3 + 10 \geq 15$

$119.1 < 119.6$

En otras palabras, con los valores de X1 = 2, X2 = 3, X3 = 10 se obtiene la tripleta de números que minimizan la ecuación. Se necesitaron solamente 2 corridas, para obtener los números que minimizan la ecuación.

Datos de entrada segunda corrida:

NP = 3 (Número de parámetros)

LC = 5 (Longitud de Cadena)

TPE = 30 (Tamaño de Población)

NG = 50 (Número de generaciones)

PM = 0.5 % (0.005) (Porcentaje de Mutación)

CONSTANTE (C) = 10000

TIEMPO DE EJECUCIÓN = 7 Seg.

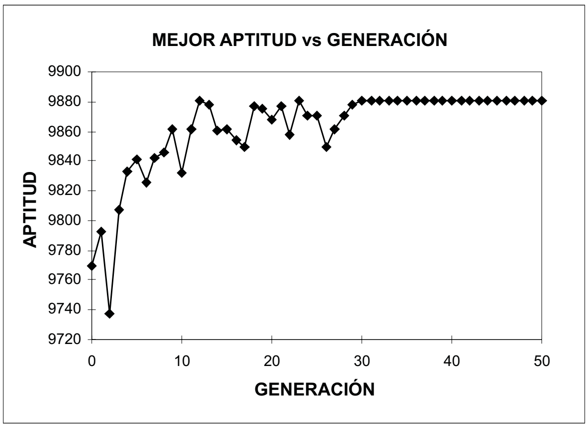
Tabla 5 – Resultados segunda corrida

GENERACIÓN	APTITUD	X1	X2	X3	GENERACIÓN	APTITUD	X1	X2	X3
0	9769.3	1	10	8	26	9849.2	4	2	10
1	9792.9	2	9	8	27	9861.7	4	3	8
2	9737.8	1	11	8	28	9870.4	2	4	10
3	9807.7	4	5	11	29	9878.4	2	2	11
4	9833.3	1	2	16	30	9880.9	2	3	10
5	9840.9	2	7	8	31	9880.9	2	3	10
6	9825.8	1	3	16	32	9880.9	2	3	10
7	9841.7	4	3	10	33	9880.9	2	3	10
8	9845.8	1	3	14	34	9880.9	2	3	10
9	9861.7	4	3	8	35	9880.9	2	3	10
10	9831.7	4	3	11	36	9880.9	2	3	10
11	9861.7	4	3	8	37	9880.9	2	3	10
12	9880.9	2	3	10	38	9880.9	2	3	10
13	9878.4	2	2	11	39	9880.9	2	3	10
14	9860.9	2	3	12	40	9880.9	2	3	10
15	9861.7	4	3	8	41	9880.9	2	3	10
16	9853.7	4	1	10	42	9880.9	2	3	10
17	9849.2	4	2	10	43	9880.9	2	3	10
18	9876.9	2	5	8	44	9880.9	2	3	10
19	9875.1	3	2	10	45	9880.9	2	3	10
20	9867.6	3	3	10	46	9880.9	2	3	10
21	9876.9	2	5	8	47	9880.9	2	3	10
22	9857.6	3	3	11	48	9880.9	2	3	10
23	9880.9	2	3	10	49	9880.9	2	3	10
24	9870.4	2	4	10	50	9880.9	2	3	10
25	9870.4	2	4	10					

6. Conclusiones

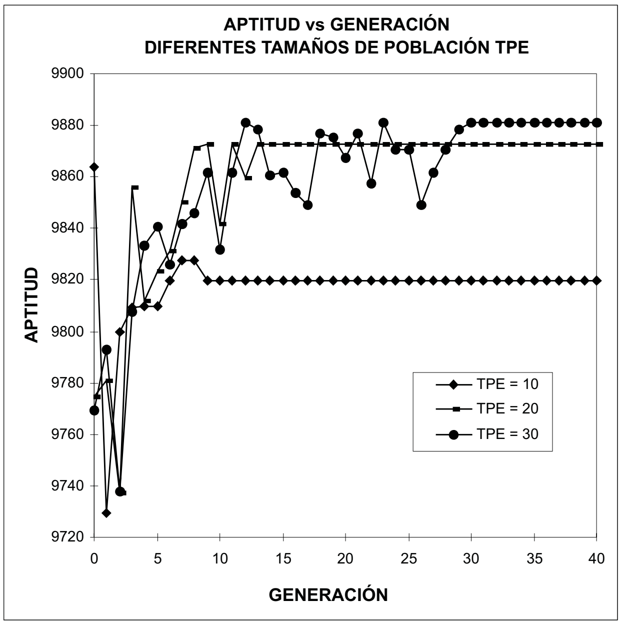
En la tabla 5 y en la gráfica Mejor Aptitud vs. Generación (figura 2), se puede observar que la aptitud máxima de 9880.9, correspondiente a X1=2, X2=3 y X3=10, se obtiene aproximadamente en la generación 30, y después se mantiene constante, demostrándose que el proceso converge hacia un valor óptimo. Aunque esto no siempre se consigue, por lo general se converge hacia un valor óptimo o un valor cercano al óptimo. Por eso, es aconsejable correr el mismo ejercicio varias veces, y sacar conclusiones.

Figura 2 – Mejor Aptitud vs. Generación



En la gráfica Comparación Diferentes Poblaciones (figura 3), se aprecia que con una población muy pequeña no se generan los mejores resultados. Cabe decir que esto se cumple en la mayoría de los casos, pero no en todos, por la misma aleatoriedad del proceso. Aumentando un poco la población, pero no demasiado, se obtienen unos resultados mejores. En cambio, una población demasiado grande haría que el Algoritmo Genético no tuviera sentido, ya que la importancia del algoritmo es que con una población pequeña, en comparación con todas las posibilidades posibles, se obtengan los resultados óptimos.

Figura 3 – Comparación Diferentes Poblaciones



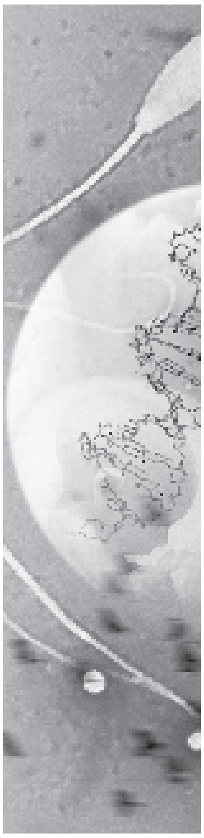
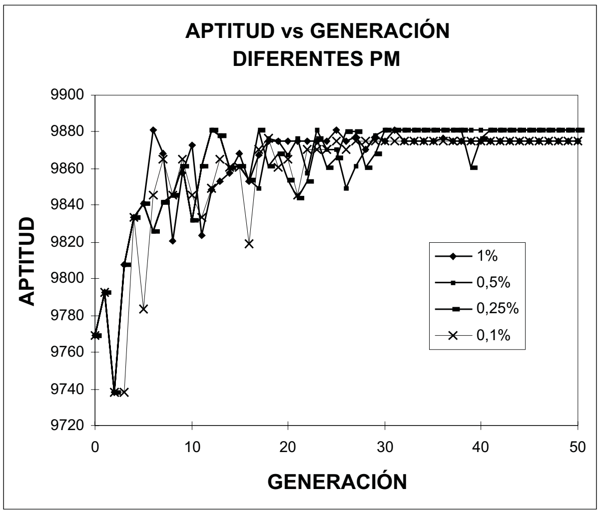


Figura 4 – Comparación Diferentes PM



3. En la gráfica Comparación Diferentes PM (figura 4), el cambiar el porcentaje o probabilidad de mutación no es tan importante o significativo como con el cambio de la población. Es más, si se aprecia con detenimiento, los resultados son muy parecidos y difieren muy poco uno del otro. Pero no obstante lo anterior, es preferible trabajar con una PM no muy grande (se aconseja 0.005), ya que con una PM grande hay una excesiva cantidad de mutaciones, haciendo que el proceso hacia el óptimo se desvíe en la mayoría de las veces, y se necesite de muchas iteraciones para lograr una convergencia en los resultados. En el otro extremo, una PM muy pequeña hace que la búsqueda de nuevo material genético que se encontraba oculto no sea posible, teniendo como consecuencia, que no se logren los mejores resultados

Para realizar los análisis de optimización de cualquier función matemática por medio del Algoritmo Genético, se realizó un programa en lenguaje Fortran (Microsoft Power Station versión 3.0). Con base en este programa se obtuvieron los anteriores resultados. En la segunda parte de este artículo se expondrá la aplicación del método del Algoritmo Genético a la optimización de estructuras tipo armaduras.

Bibliografía

1. GALANTE, Miguel, *Un algoritmo genético simple para la optimización de estructuras planas articuladas*. Revista internacional de métodos numéricos para cálculo y diseño en Ingeniería. Universidad Politécnica de Cataluña, Vol. 9, no. 2, diciembre 1992, p. 179-199.
2. GOLDBERG, David E. *Genetic algorithms in search, optimization and machine learning.* Addison-Wesley. Massachusetts, 412 p., 1989.
3. GRIERSON, D.E., Pak, W.H. *Optimal sizing, geometrical and topological design using a genetic algorithm*. Structural Optimization, No. 6, 1993, p. 151-159.
4. HOLLAND, John H., *Algoritmos Genéticos.* Revista Investigación y Ciencia, España, septiembre 1992, p. 38-45.
5. JENKINS, W.M., *Towards structural optimization via the genetic algorithm.* Computers & Structures, vol. 40, no. 5, 1991, p. 1321-1327.
6. JENKINS, W.M., *Improving structural design by genetic search,* Computer-Aided Civil Engineering, vol. 13, 1998, p. 5-11.
7. KOUMOUSIS, Vlasis K., GEORGIOU, Panos. *Genetic algorithm in discrete optimization of steel truss roofs.* Journal of Computing in Civil Engineering, vol. 8, no. 3, julio 1994, p. 309-325.
8. PELÁEZ A., Alejandro, *Ingeniería de Software: Introducción a los algoritmos genéticos,* Revista Silicio, no. 5, año 4, p. 21-29.
9. RAJEEV, S., KRISHNAMOORTY, C. S. *Discrete optimization of structures using genetic algorithm.* Journal of Structural Engineering, vol. 118, no. 5, mayo 1992, p. 1233-1249.
10. RAJEEV, S., KRISHNAMOORTY, C. S. *Genetic algorithm-based methodology for design optimization of reinforced concrete frames*. Computer-Aided Civil and infrastructure Engineering, 1998.