PEV: Práctica 1

Jorge Vieira Luna

José Miguel Tajuelo Garrigós

19-3-2017

Breve descripción de operadores implementados

Cruce

Cruce monopunto clásico (*"Monopunto"*): Actúa con la probabilidad determinada sobre 2 individuos. De determinarse para cruzar, toma un entero aleatorio, en el rango de cortes posibles dentro de los cromosomas de los individuos. El corte puede caer entre genes o a mitad de un gen. La información genética es intercambiada entre los dos individuos a partir del punto de cruce.
Ejemplo con dos genes binarios (siendo '.' cada punto de cruce posible):
1.1.1.1.1 y 0.0.0.0.0, individuos de partida. Se elige un entero al azar 0 < n <= 3, sale 1.
Intercambian su información genética a partir del punto de cruce 1, resultan:

1.1▪0.0.0 y 0.0▪1.1.1

NOTA: En el caso de cromosomas reales, el punto de cruce se encuentra siempre entre genes. Esto se debe a que un gen real tiene una única base, no pudiéndose producir cruces dentro de los genes.

Cruce aritmético (*"Aritmético"*): Usado para números con representación real, partiendo de dos individuos, para el primero: Se da una relevancia "alfa" al valor del primero (entre 0 y 1), y una relevancia opuesta (1 - alfa) al valor del segundo. Combinando ambos se determina el nuevo valor real del primer individuo. Para el segundo, se le da una importancia 1 - alfa al primero y alfa al segundo, y de nuevo se combinan.
Ejemplo con dos genes reales (alfa = 0.7):
60 y 80 individuos de partida.
El primero pasa a ser: 0.7 * 60 + 0.3 * 80 = 66
El segundo pasa a ser: 0.3 * 60 + 0.7 * 80 = 74

Mutación

Mutación base a base (*"Base a base"*): En los cromosomas binarios cada dígito binario tiene la probabilidad especificada desde la GUI de cambiar su valor (0 -> 1, 1 -> 0).
En los cromosomas representados por números reales, hemos definido una constante alfa *hardcoded* con valor 0.2. Cuando una base real ha de mutar, se toma su valor real y se aumenta o decrece en un % alfa, manteniendo el valor de la variable dentro de los límites admisibles del problema.

Selección

Ruleta (*“Ruleta”*): Primero damos a toda la población un fitness adaptado proporcional a su fitness, pero que tiene siempre valor positivo e incremental. Una vez hecho esto damos a cada individuo una probabilidad de ser seleccionado, dividiendo su fitness adaptado entre el total de la suma de fitness adaptados. Obtenemos un vector de 0 a 1 en el que cada individuo tiene un espacio proporcional a su fitness adaptado, ordenados de menor a mayor. Una vez tenemos esto ruleta saca un número aleatorio entre 0 y 1 y selecciona el individuo a la izquierda del punto indicado por dicho número en el vector, utilizando búsqueda binaria.

Estocástico Universal (*“Estocástico Universal”*): Usa el mismo vector definido en ruleta, pero genera un único número aleatorio al principio. A partir de una marca definida por dicho valor, se seleccionan un elemento tras otro desplazando la marca en una distancia 1/N.

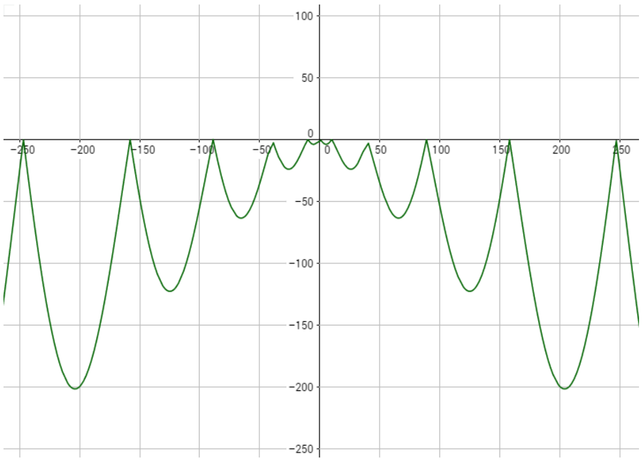
Torneo Determinista (*“Torneo DET(2)”, “Torneo DET(3)”*): Este método de selección consiste en la obtención de N individuos aleatoriamente de la población, dando el mismo peso a todos ellos. A continuación, se considera seleccionado el que posea un fitness mejor.

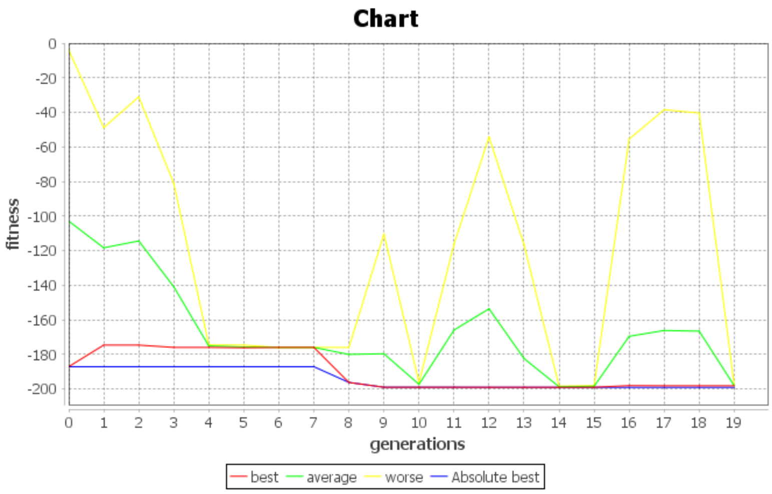
Función 1

$$f(x) = -\left|-x * sen(\sqrt{|x|})\right|$$

Rangos

x ∈ [-250, 250]





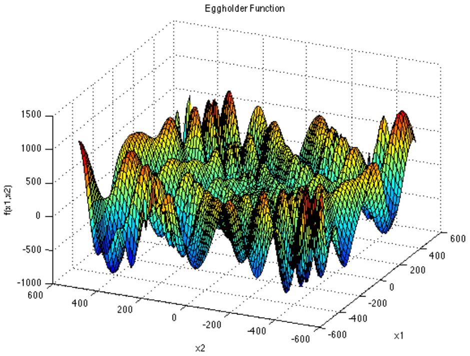
Población: 5
Generaciones: 20
Selección: Ruleta
Crossover: Monopunto 40%
Mutación: Base a base 2%
Elite: 0%

Análisis:

Se trata de un problema de minimización muy sencillo, al contar con pocos mínimos locales (en comparación con el resto de funciones de la práctica) y una sola dimensión. Esto hace que se encuentren muy buenos resultados muy rápidamente y con poblaciones muy pequeñas.
Al ser un problema tan sencillo se puede apreciar el efecto destructivo de la mutación base a base en algunos individuos en las generaciones 8, 10, 11 y 15, que a su vez se ve compensado por la selección con ruleta, al seleccionar más veces para las siguientes generaciones los individuos que mantienen un buen fitness.
También se aprecia que al no usar élite en la generación #1 el mejor individuo de la población empeora su fitness, manteniéndose peor que el mejor absoluto alcanzado hasta la generación #8.

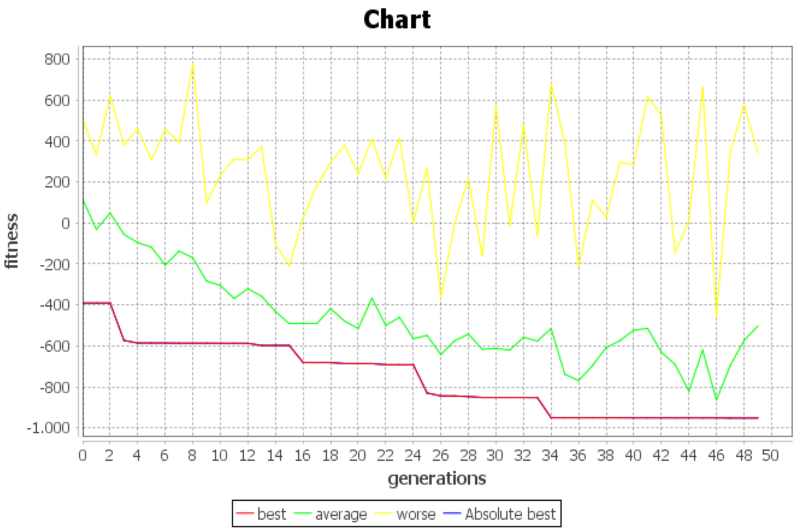
Función 2

$$f(\mathbf{x}) = -(x_2 + 47)\sin\left(\sqrt{\left|x_2 + \frac{x_1}{2} + 47\right|}\right) - x_1 \sin\left(\sqrt{\left|x_1 - (x_2 + 47)\right|}\right)$$



Rangos

$x_1 \in [-512, 512]$
$x_2 \in [-512, 512]$



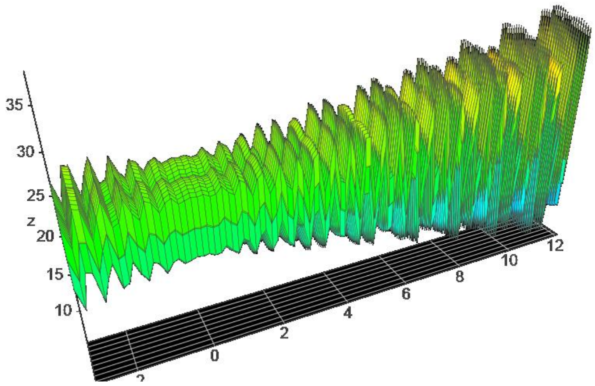
Población: 20
Generaciones: 50
Selección: Ruleta
Crossover: Monopunto 40%
Mutación: Base a base 2%
Elite: 5%

Análisis:

De nuevo se trata de un problema de minimización, con una complejidad algo mayor. En este hemos elegido usar élite en el ejemplo para observar como el mejor absoluto y el mejor de cada generación coinciden, ya que la élite garantiza su paso a la siguiente generación, sin cruzar ni mutar, aunque sus individuos pueden ser utilizados para producir la siguiente generación.
Al ser un poco más complejo, en este es más visible cierto progreso en la evolución de la población, alcanzándose mejores fitness conforme avanzan las generaciones. Esto es fácilmente observable en el fitness medio de la población.
Se mantiene una distancia considerable entre los mejores de cada generación y el fitness medio de la población, a la vez que los peores son bastante malos en relación a la media. Esto es muestra de que existe una diversidad adecuada a lo largo de toda la evolución, evitando convergencias prematuras y estancamientos en mínimos locales.

Función 3

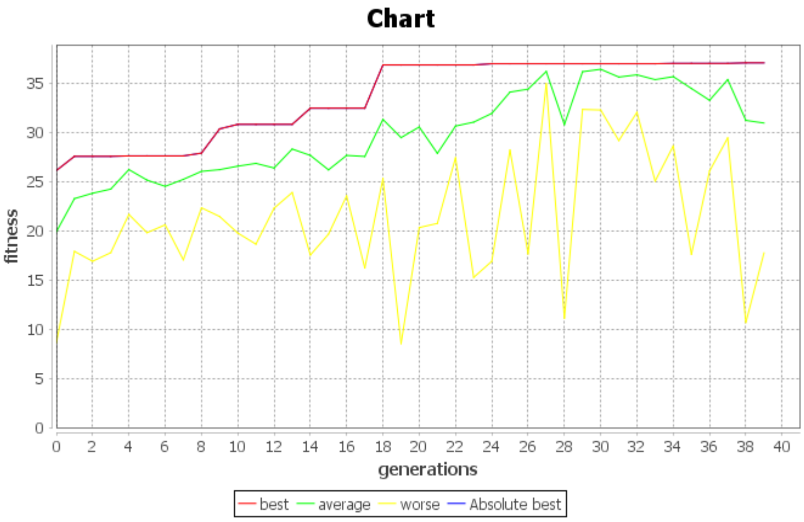
$f(x, y) = 21.5 + x * sen(4\pi x) + y * sen(20\pi y)$



Rangos

x ∈ [-3, 12.1]

y ∈ [4.1, 5.8]



Población: 10

Generaciones: 40

Selección: Ruleta

Crossover: Monopunto 40%

Mutación: Base a base 2%

Elite: 10%

Análisis:

En esta ocasión el problema es de maximización. Para garantizar el correcto funcionamiento del algoritmo genético independientemente de si el problema pretende minimizar o maximizar el fitness, empleamos en todos los problemas, en lugar del fitness habitual, un fitness adaptado para cada elemento, el cual es proporcional según el rango al fitness original, pero siempre con números positivos, y siempre asignando a mejores individuos un valor mayor.

Por lo demás el desempeño del algoritmo y la complejidad de la función es muy similar al problema anterior.

Función 4

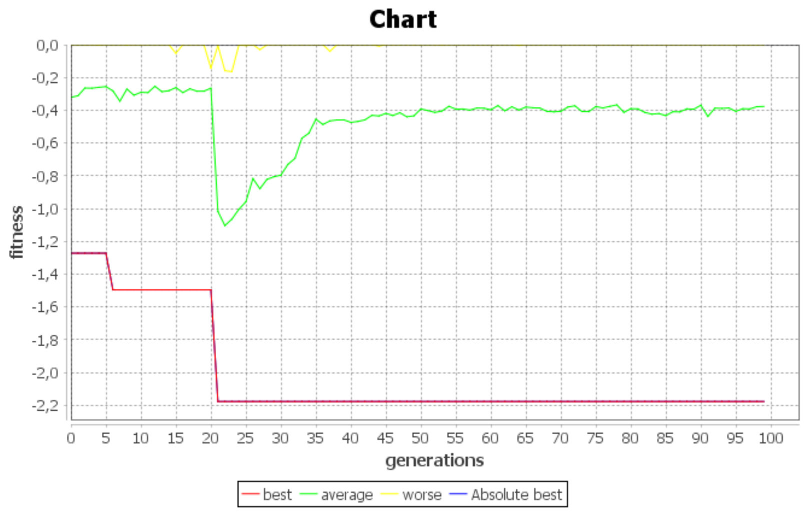
$$f(x_i|i = i..n) = -\sum_{i=1}^{n} sen(x_i) * sen^{20}(\frac{(i+1) * x_i^2}{\pi})$$

Rangos

$x_i \in [0, \pi]$

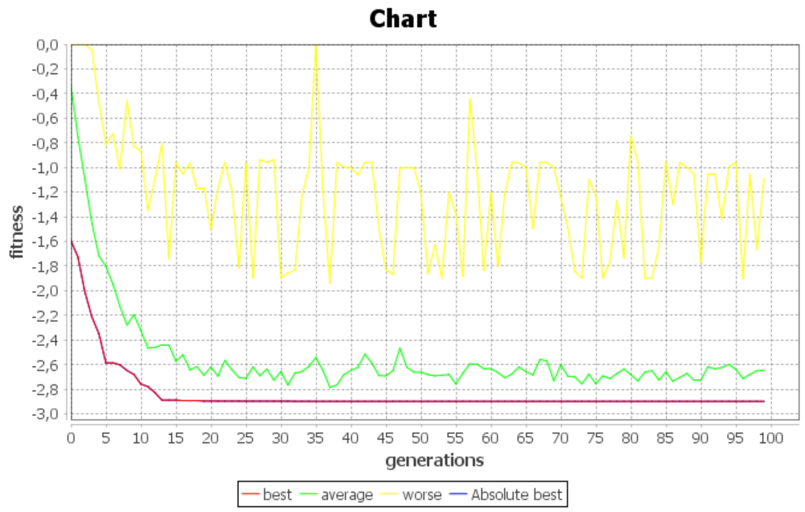
Análisis A, binario:

Gráfica a)



Población: 50
Generaciones: 100
Selección: Ruleta
Crossover: Monopunto 40%
Mutación: Base a base 2%
Elite: 10%
N: 3

Gráfica b)



Población: 50
Generaciones: 100
Selección: Torneo determinista (3 individuos)
Crossover: Monopunto 40%
Mutación: Base a base 2%
Elite: 10%
N: 3

Hemos incluido dos gráficas para el análisis del problema 4, de minimización (versión binaria).

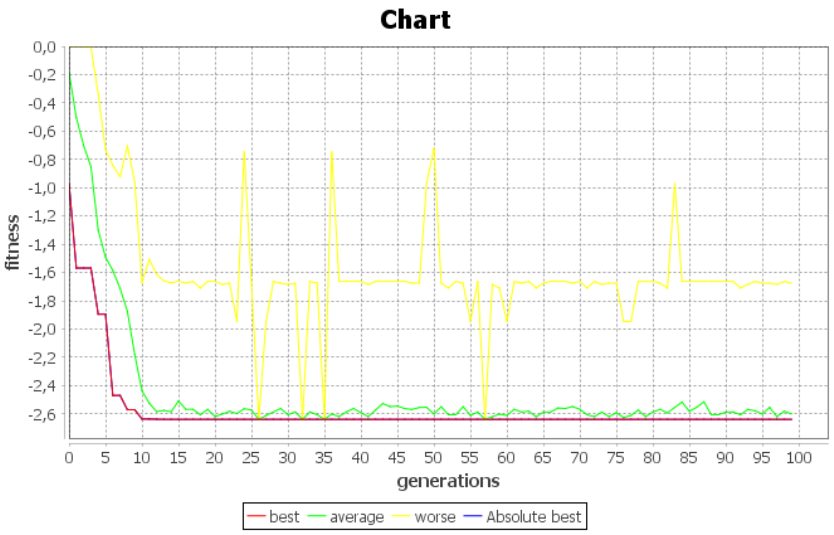
En la primera hemos usado ruleta como método de selección, y se aprecian varias cosas interesantes. La principal es que se estanca en un mínimo local desde la generación 21 a la 100 (-2.19) del que no logra salir, muy lejos del objetivo (-2.89). Aparte, en esa ejecución concreta, encontramos algo interesante: En la generación #21 se encuentra un individuo mucho mejor que el resto. Este individuo es seleccionado masivamente por el método de ruleta dado que es mucho mejor, saturando con sus copias y cruces la mayoría de la población. Esto es apreciable en la mejora del fitness promedio, que se dispara a -1.1. En las generaciones siguientes el fitness promedio decrece lentamente; esto es porque la diferencia entre los individuos ya no es tanta tras la convergencia de la población al mejor individuo, y los peores se ven seleccionados con más frecuencia para pasar a la siguiente generación, hasta que vuelve a estabilizarse en torno a -0.4.

En la segunda gráfica hemos usado torneo determinista (3 individuos) como método de selección, obteniendo resultados mucho mejores. Se puede apreciar como el mejor individuo mejora en cada una de las generaciones 0 - 13, poniéndose de manifiesto que torneo es mucho mejor evitando mínimos locales que ruleta (ruleta produce líneas horizontales en el mejor individuo de generaciones consecutivas, lo cual indica estancamiento). Esto le permite alcanzar el mínimo objetivo de -2.89, y además hacerlo en únicamente 13 generaciones.

Asociamos la increíble mejora de torneo frente a ruleta en este problema debido a la complejidad del problema. A mayor complejidad del problema (ubicación y cantidad de mínimos locales) es más fácil estancarse. Ruleta es proclive a ello y fracasa, mientras que torneo lo evita en cierta medida. Para más inri, hemos comprobado que en este problema una elevada mutación mejora mucho los resultados obtenidos, mientras que con una baja mutación incluso torneo se estanca, especialmente en la versión B (real) del problema, ya que la mutación que utilizamos es menos agresiva.

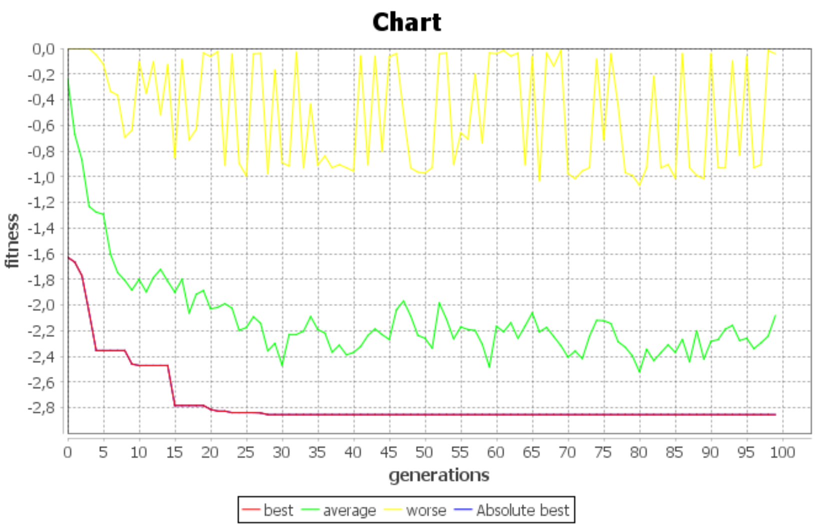
Análisis B, real:

Gráfica a)



Población: 50
Generaciones: 100
Selección: Torneo determinista (3 individuos)
Crossover: Aritmético 40%
Mutación: Base a base 2%
Elite: 10%
N: 3

Gráfica b)



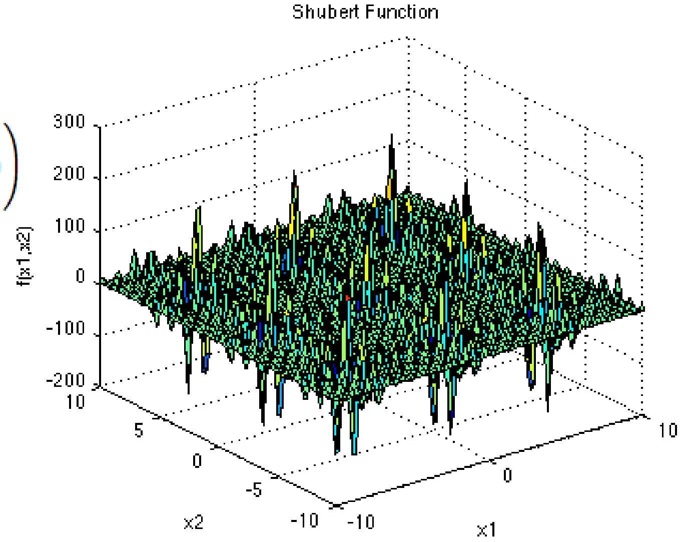
La ejecución de problema 4 empleando reales requiere el uso de operadores de cruce y selección enfocados a cromosomas reales, por lo que hemos optado por utilizar una modificación de la mutación bit a bit (que hemos denominado base a base) y el operador de cruce aritmético.

Gracias a estos dos últimos operadores garantizamos la generación de valores reales nuevos a partir de otros. Esto resulta en una ejecución con resultados ligeramente peores que con cromosomas binarios debido al empleo de una mutación mucho más discreta.

Puede ser parcialmente solventado aumentando el porcentaje de mutación, como refleja la gráfica B, en la que en vez de 2% hemos usado 20% manteniendo el resto de parámetros sin alterar.

Caso de interés: En el problema 4 empleando reales, al ejecutar sin mutación y con cruce monopunto (intercambio de genes), la mejora se produce únicamente al formar cromosomas intercambiando genes completos (al tener un gen real una sola base) de elementos de la población inicial. De esta forma, el algoritmo al cruzar lo que hace es formar nuevos individuos probando variaciones posibles entre las diversas alternativas que la población total tiene para cada variable (El número de variables es N, y el número de alternativas es el tamaño de la población, pues cada individuo tiene un valor para cada una de las variables).
A partir de esto llegamos a la conclusión de que el máximo alcanzable está predeterminado en la población inicial, al ser este la mejor combinación de genes inalterados de todos los individuos. En este caso se puede comprobar que el óptimo alcanzado depende exclusivamente del tamaño de la población, al ser el número de combinaciones posibles para formar cromosomas $tampob^N$.

Función 5

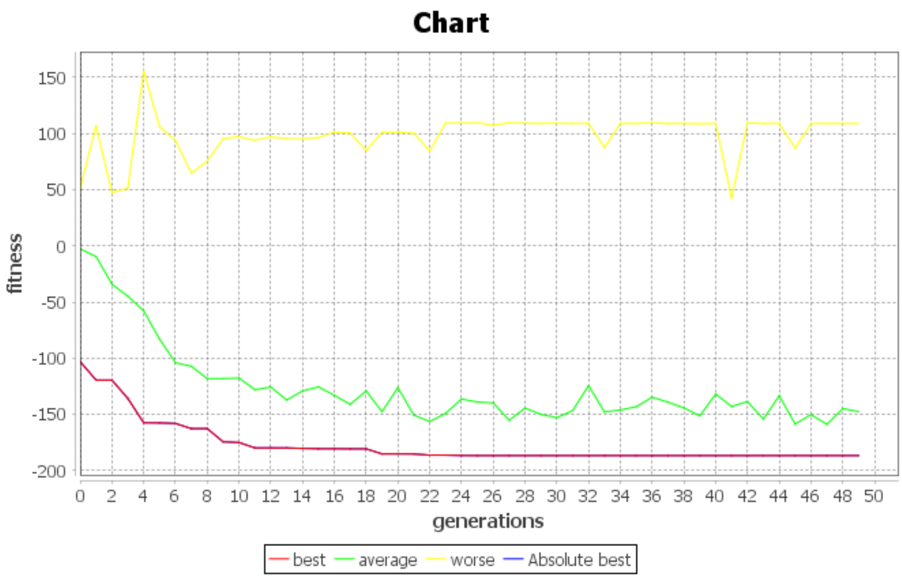


$$f(\mathbf{x}) = \left(\sum_{i=1}^{5} i\cos((i+1)x_1 + i)\right)\left(\sum_{i=1}^{5} i\cos((i+1)x_2 + i)\right)$$

Rangos

$x_1 \in [-10, 10]$

$x_2 \in [-10, 10]$



Población: 100
Generaciones: 50
Selección: Ruleta
Crossover: Monopunto 40%
Mutación: Base a base 2%
Elite: 5%

Análisis:

Es una función de minimización tal vez algo más compleja que las tres primeras. Utilizando la configuración descrita, se obtiene aproximadamente 2 veces de cada 3 uno de los mínimos (-186.73), y 1 de cada 3 (-186.00 ~ -186.70) como es el caso de la gráfica mostrada, poniendo de manifiesto que es fácil estancarse en mínimos locales en esta función concreta. Aumentando las generaciones a 100, cambiando el cruce a torneo de 3 individuos y la mutación a 5%, llega al óptimo prácticamente siempre.