

Optimización con Algoritmos Genéticos, mutando genes en forma adaptativa

Juan Pablo Salamanca Ramírez

Departamento de Ingeniería de Sistemas e Industrial, Universidad Nacional de
Colombia, Bogotá, Carrera 45 No 26-85, Colombia,
jpsalamarcara@unal.edu.co

Abstract. Optimization problems are present in the daily life of all human beings. The fact of asking How to make a good investment of money when you go grocery shopping? or What is the best way to go school using of different means of transport ?, is to consider an optimization problem. Although there are various optimization techniques, one of the most appreciated in the academic community and industry are the genetic algorithms. In this document a proposed genetic algorithm for optimization of functions in the search space of real numbers is presented, facing the problem of non-separability.

Keywords: evolutionary computation, evolutionary algorithm, real optimization, large scale, non-separability

1 Introducción

Los problemas de optimización están presentes en la vida diaria de todos los seres humanos, el hecho de preguntarse ¿Cómo hacer una buena inversión del dinero disponible? cuando vas de compras al supermercado o ¿Cuál es la mejor forma de llegar a la escuela? haciendo uso de los diferentes medios de transporte, es plantearse un problema de optimización. Aunque existen diversas técnicas de optimización, una de las más apreciadas en la comunidad académica y en la industria, son los algoritmos genéticos. En este documento se presenta una propuesta de algoritmo genético para optimización de funciones en el espacio de búsqueda de los números reales, haciendo frente al problema de la no separabilidad.

2 Estado del Arte

En la actualidad existen diversas técnicas de optimización para el dominio de los números reales. Algunas de estas son muy conocidas, sin embargo, tienen muchas condiciones particulares para poderse usar correctamente. Por ejemplo, para aplicar el gradiente descendente, la función a optimizar debe ser diferenciable y el manejo de las restricciones puede volverse todo un dolor de cabeza. Otras técnicas no dependientes de gradientes o derivadas. suelen tener resultados muy buenos, sin embargo, estas sufren la maldición de la dimensionalidad, es decir,

cuando las dimensiones del problema crecen, sus soluciones ya dejan de ser tan buenas. En este grupo se encuentran los algoritmos genéticos, los cuales pueden describirse como una búsqueda aleatoria inspirada en la teoría de la evolución. Muchos investigadores han intentado resolver problemas de optimización sobre el dominio de los reales, aplicando diferentes variantes de algoritmos genéticos. Por ejemplo, el algoritmo HaEA [1] hace que la selección de sus operadores de cruce y mutación sea aleatoria y auto adaptable, lo cual lo libera de muchos parámetros y hace que sea muy eficiente. Otras técnicas pertenecientes a la computación evolutiva, hacen frente a la optimización con estrategias distintas. Como es el caso de la optimización multi-objetivo, la cual toma base en el concepto de dominancia entre individuos de una población y divide el problema general en pequeños problemas de optimización manejables [4]. Sin embargo, el inconveniente de muchas de estas técnicas, radica en la naturaleza de la función a optimizar. En muchos casos, las funciones son separables, es decir, sus variables de entrada no presentan interdependencia, en este caso, podríamos dividir el conjunto de variables de entrada y hacer optimización multi-objetivo. O hacer una búsqueda aleatoria optimizando una a una las variables de entrada. Sin embargo, las funciones reales, no suelen tener este tipo de naturaleza aprovechable [5]. En los últimos años han surgido novedosas técnicas de acuerdo a la separabilidad o no separabilidad de una función a optimizar. Esto crea la molesta tarea de estudiar analíticamente la función objetivo para identificar su naturaleza separable o no separable.

3 Propuesta

Teniendo en cuenta las principales conclusiones de J. Gómez [1] en relación a la preservación de la diversidad como estrategia evolutiva y al análisis de los diferentes esquemas de selección realizado por T. Blicke & L. Thiele[3]; se plantea encontrar un método para seleccionar individuos de una población, en el cual se favorezca la exploración y explotación como principios básicos para disminuir la pérdida de diversidad y la probabilidad de estancamiento en un máximo (o mínimo) local. El resultado es un híbrido entre la selección por ruleta y la selección por ranking.

3.1 Estrategia de Selección: Ruleta-Ranking

El procedimiento de selección propuesto, llamado inicialmente Ruleta-Ranking, se describe paso a paso a continuación:

1. Se normalizan las medidas de aptitud de cada individuo de la población usando escala decimal.
2. Se normalizan los datos del paso anterior en el intervalo $[0, 1]$
3. Se ordenan de forma descendente los individuos de acuerdo su medida de aptitud normalizada, de manera que el mejor individuo sea el primero de la lista y el peor el último.

4. Generar un número aleatorio usando una distribución normal con $\mu = 0$ y $\sigma = \frac{1}{3}$, y aplicarle la función valor absoluto.
5. Seleccionar el individuo con la medida de aptitud normalizada inmediatamente mayor al número aleatorio generado en el paso anterior.
6. Retirar al individuo seleccionado de la lista y ajustar las probabilidades de los individuos restantes.

En esta estrategia, los mejores individuos tienen mayor probabilidad de ser seleccionados inicialmente, sin embargo, con su salida de la lista de espera, se redistribuye la probabilidad en los individuos restantes, ofreciéndole prioridad a los nuevos mejores. Esta técnica permite exploración ya que por la forma de la distribución normal, existe la probabilidad de seleccionar individuos no tan buenos. Y permite explotación, porque se seleccionan los mejores en mayor proporción. Para controlar la complejidad del algoritmo, se mantiene la sumatoria total de las medidas de aptitud normalizadas de los individuos en espera y cada vez que alguno es elegido, su medida de aptitud es restada del total y se aplica otra normalización respecto a este nuevo total solamente cuando se lanza nuevamente la ruleta.

Comentario. Al cambiar la forma de la distribución del número generado en el paso 4, por una con $\mu = 1$, se obtiene que la ejecución del algoritmo genético minimiza la función de aptitud.

3.2 Estrategia de Generación y Reemplazo

Para la generación de nuevos individuos se propone como estrategia el matrimonio, generando de a dos hijos por pareja. Sin embargo, el criterio de selección de pareja se dividió en dos: por mayor similitud y por mejor medida de aptitud, teniendo en cuenta que el mejor individuo tiene $n - 1$ oportunidades de elegir, el segundo mejor individuo tiene $n - 2$ y así sucesivamente hasta que el penúltimo individuo solo puede elegir al peor (n es el tamaño total de los individuos elegibles).

Como estrategia de reemplazo se propone reemplazo de estado estable ó generacional, en combinación con alguna de las estrategias de generación anteriormente descritas.

3.3 Función de Aptitud

La función de aptitud es una función de las quince funciones disponibles en [2], las cuales fueron usadas para la competencia de optimización a gran escala organizada por la IEEE en 2013. Estas funciones cuentan con 1000 variables de entrada y presentan diversas características de separabilidad.

3.4 Inicialización

Para la inicialización se generan cadenas de números aleatorios, bajo una distribución normal con $\sigma = 1$ y $\mu = 0$

3.5 Operadores

En cuanto a los operadores se tienen dos, mutación y cruce.

Mutación Para la mutación, se propone un operador auto adaptable y selectivo. Para lograr esto, se define un radio de búsqueda con centro en el valor de la j -ésima variable de cada individuo, este radio está definido por una distribución normal con σ = tasa de aprendizaje del individuo, valor que a su vez, fue asignado inicialmente siguiendo otra distribución de probabilidad normal. Esta tasa de aprendizaje, será ajustada de acuerdo a la regla de $\frac{1}{5}$. Sin embargo, todavía queda la incógnita de cómo hacer frente a la no separabilidad. Tomando inspiración en la auto adaptabilidad de HaEA [1] para seleccionar sus operadores, se propuso detectar la sensibilidad de cada una de las variables de la función objetivo al ser modificadas, teniendo en cuenta, que posiblemente durante todo el proceso de optimización esta sensibilidad podría disminuir o aumentar de acuerdo al estado de las demás variables. La solución fue premiar o castigar las variables que después de aplicarles una mutación aumenten o disminuyan la medida de aptitud del individuo. Sin embargo, si por cada variable mutada se midiera el cambio en la medida de aptitud, aumentaría la complejidad del algoritmo, entonces, la solución fue usar el mecanismo ya conocido para premiar o castigar la tasa de aprendizaje (regla de $\frac{1}{5}$), para adicionalmente premiar la sensibilidad de cada variable mutada y no abusar de la función de aptitud. Cabe aclarar, que de acuerdo a la sensibilidad de cada variable, aumenta o disminuye la probabilidad de mutar esa variable. Por otra parte, queda la posibilidad de que una o más variables sean lo suficientemente castigadas como para que la población se quede en un mínimo (o máximo) local. Para hacer frente a este problema se propuso el mecanismo de insertar ruido en la sensibilidad acumulada de las variables. Para que nuevamente se empiece a adaptar dicha sensibilidad y sea cambiante en el transcurso de la optimización. Este proceso de inserción de ruido es aleatorio y está definido por una probabilidad p como parámetro de entrada.

Cruce El operador de cruce es sumamente sencillo, se selecciona un punto pivote en la cadena de variables de cada individuo y a partir de este son generados dos nuevos individuos, a los cuales se les asigna la tasa de aprendizaje y la sensibilidad de alguno de sus dos progenitores.

4 Experimentos y Resultados

Para determinar cuál de las estrategias evolutivas es mejor para complementar el algoritmo propuesto, se ejecutaron varias pruebas obteniendo como resultado que: Para una estrategia reemplazo de estado estable, el algoritmo encuentra mejores soluciones sobre las funciones $F5, F6, F9, F10$, las cuales presentan una naturaleza parcialmente separable. En el caso de las demás funciones, buenos resultados fueron hallados muy rápidamente por una estrategia de reemplazo generacional, aunque en el caso particular de la función $F15$, cuya naturaleza es

no separable, el algoritmo evolutivo después de algunas iteraciones comenzaba a oscilar . En cuanto a la estrategia de generación de nuevos individuos, los parejas realizadas por mayor similitud demostraron evolucionar más lentamente en comparación a las parejas armadas por mayor medida de aptitud.

4.1 Shifted Elliptic Function (F1) [2]

Nuevos experimentos fueron realizados usando cuatro posibles combinaciones de estrategias evolutivas, sobre la función $F1$, la cual tiene una naturaleza unimodal y separable :

1. Reemplazo generacional con parejas por mejor medida de aptitud. Fig 3.
2. Reemplazo generacional con parejas por similitud. Fig 5.
3. Reemplazo de estado estable con parejas por mejor medida de aptitud. Fig 4.
4. Reemplazo de estado estable con matrimonio por similitud. Fig 6.

Los parámetros fueron fijados de la siguiente manera: el tamaño de la población = 100, total de iteraciones = 1000 y $p = 0.01$.

4.2 Shifted Schwefel's Function (F15) [2]

Para observar el comportamiento del algoritmo sobre una función de naturaleza no separable, se realizaron experimentos con la función 15. Fig 1 y 2.

5 Trabajo Futuro

En el marco de tiempo tan corto para este estudio, se pudo apreciar experimentalmente la superioridad del algoritmo propuesto, comparado, en relación a si mismo sin hacer uso del operador de mutación adaptativo. Sin embargo, hace falta un estudio formal, en el cual se le compare con otros algoritmos desde una perspectiva teórico practica. En un futuro se espera probar las principales cualidades del algoritmo propuesto, en optimización de múltiples objetivos, como es el caso de SPEA2 [4]. Otra propuesta que queda para un estudio futuro, es aplicar cambios en la estrategia evolutiva en tiempo de ejecución, es decir, cambiar por ejemplo, la estrategia de reemplazo mientras el algoritmo esta ejecutándose, con el objetivo de hacer explotación en un periodo de tiempo y luego exploración en el siguiente periodo.

6 Conclusiones

- La combinación entre el esquema de generación de nuevos individuos por mejor medida de aptitud y reemplazo de tipo generacional, manifiesta en el algoritmo propuesto un carácter voraz. Esta particularidad se muestra muy

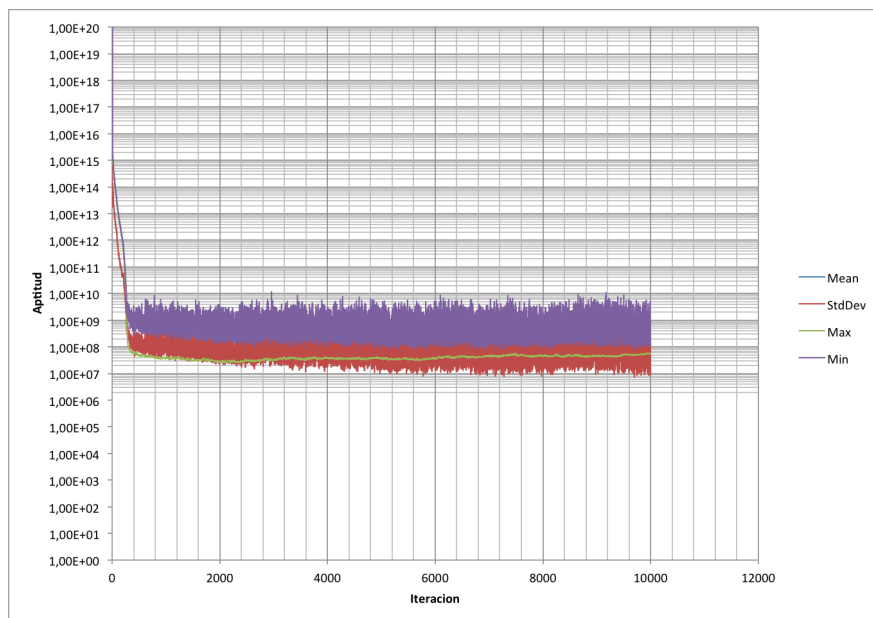


Fig. 1. Parejas por mejor aptitud y reemplazo generacional en $F15$

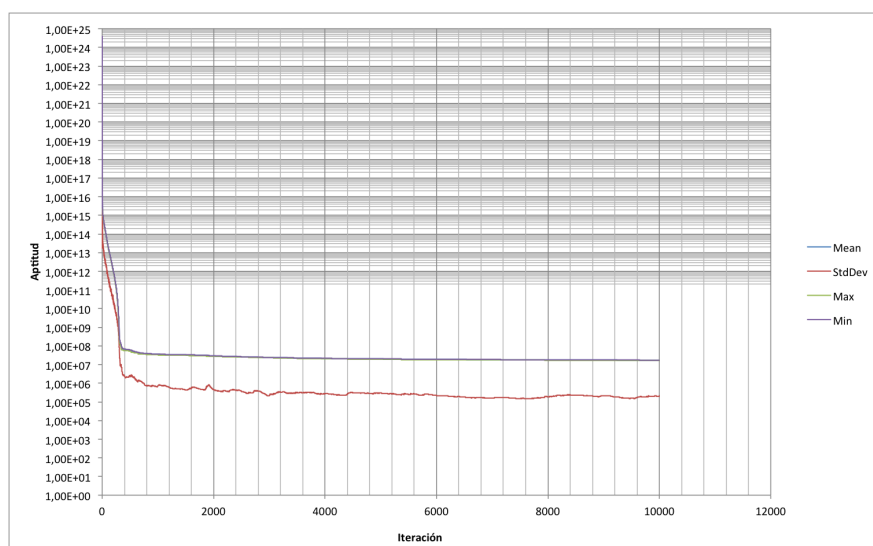


Fig. 2. Parejas por mejor aptitud y reemplazo de estado estable en $F15$

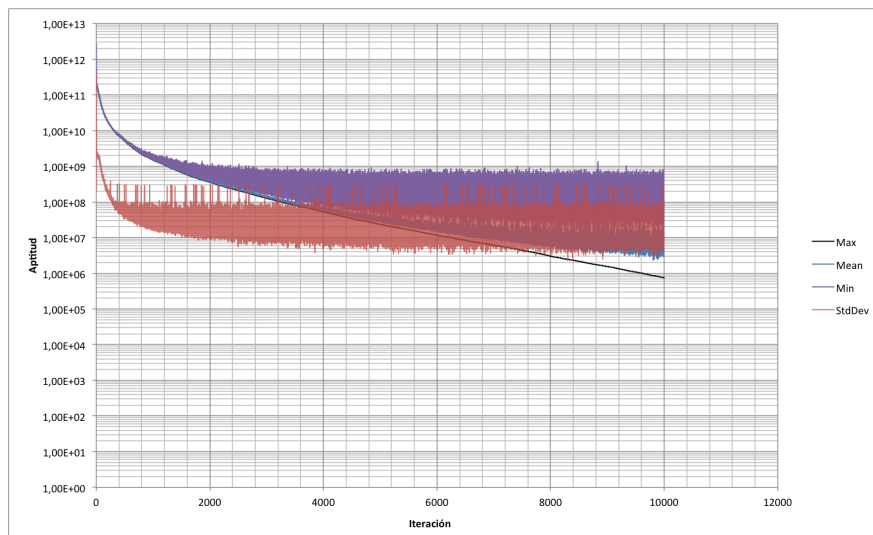


Fig. 3. Parejas por mejor aptitud y reemplazo generacional en $F1$

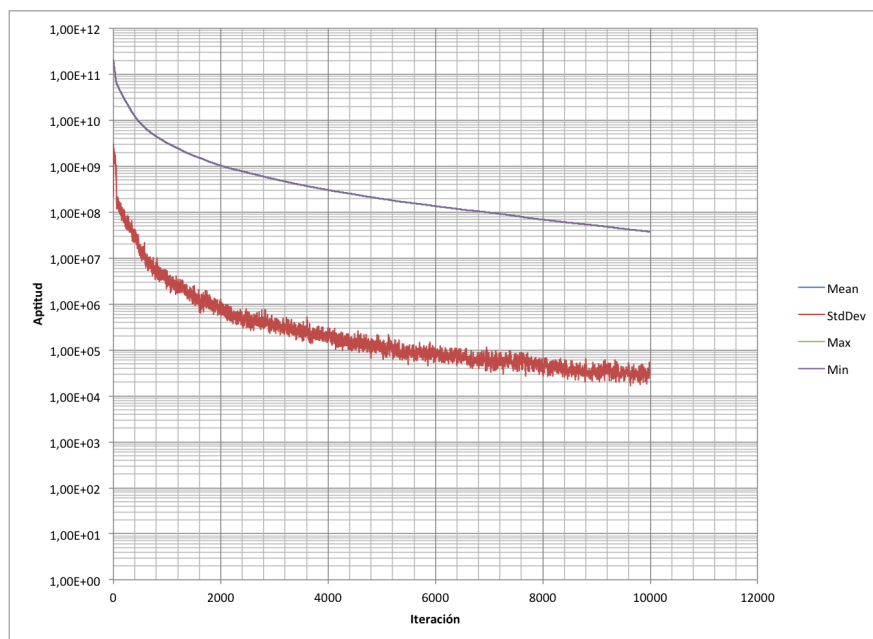


Fig. 4. parejas por mejor aptitud y reemplazo de estado estable en $F1$

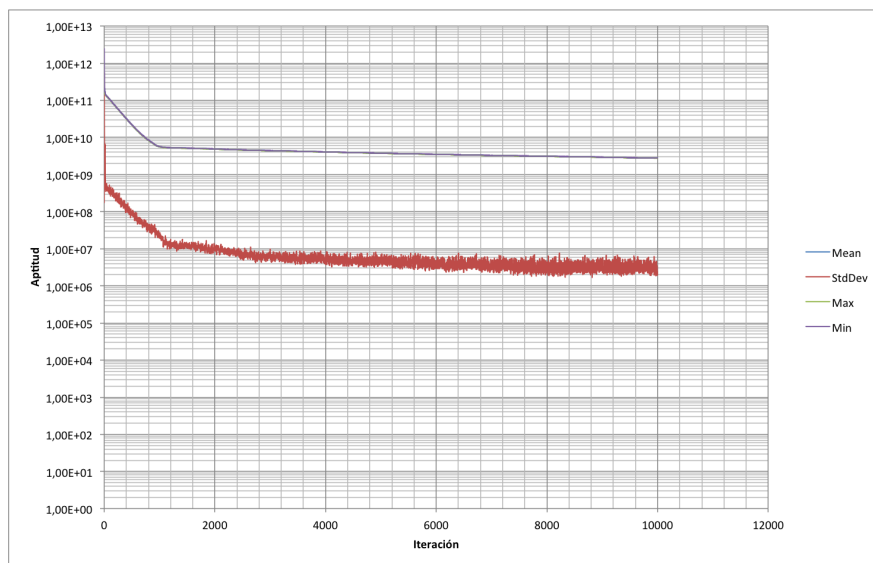


Fig. 5. Parejas por mayor similitud y reemplazo generacional en $F1$

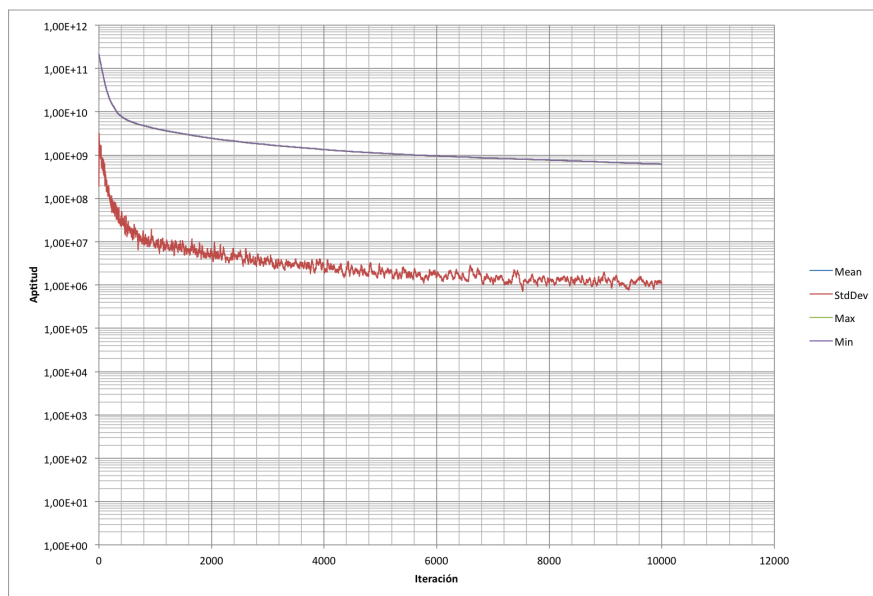


Fig. 6. Parejas por mayor similitud y reemplazo de estado estable en $F1$

efectiva en funciones separables, sin embargo, en funciones parcialmente separables, este comportamiento parece ser contraproducente. Ya que experimentalmente se ve claramente que se obtienen mejores soluciones con un esquema de reemplazo de estado estable.

- Al hacer uso del esquema de reemplazo de estado estable, se observa que aunque este tiene buenos resultados, su avance se estabiliza en algún punto del tiempo. lo que lleva a concluir que es ideal para hacer explotación.

References

1. J. Gómez: Self Adaptation of Operator Rates in Evolutionary Algorithms Universidad Nacional de Colombia and The University of Memphis, (2004)
2. X. Li, K. Tang, M. Omidvar, Z. Yang and K. Qin, "Benchmark Functions for the CEC'2013 Special Session and Competition on Large Scale Global Optimization," Technical Report, Evolutionary Computation and Machine Learning Group, RMIT University, Australia, 2013.
3. Tobias Blickle & Lothar Thiele: A Comparison of Selection Schemes used in Genetic Algorithms. Computer Engineering and Communication Networks Lab. Swiss Federal Institute of Technology. (1995)
4. Eckart Zitzler, Marco Laumanns, and Lothar Thiele, SPEA2: Improving the Strength Pareto Evolutionary Algorithm Computer Engineering and Networks Laboratory (TIK), Department of Electrical Engineering, Swiss Federal Institute of Technology (ETH) Zurich (2001)
5. Pilar Caamaño*, Francisco Bellas, Jose A. Becerra, Richard J. Duro, Evolutionary algorithm characterization in real parameter optimization problems, Applied Soft Computing 13 (2013) 1902–1921