COMPUTACIÓN INTELIGENTE: UN ESTUDIO COMPARATIVO DE METAHEURÍSTICAS

Ricardo González Tello, Ing. *Eduardo Alberto Chan Alejandre, Ing.*

José Félix Serrano Talamantes Dr. *Mauricio Olguín Carbajal Dr.*

Instituto Politécnico Nacional

CIDETEC

rgonzalezt0700@alumno.ipn.mx eduardo.chan.alejandre@hotmail.com

jfserranotal@gmail.com molguinc@ipn.mx

Resumen

En este articulo se realizó un comparativa entre las metaheurísticas; acenso de colinas con mutación aleatoria, recocido simulado, algoritmos genéticos, optimización por cumulo de partículas y optimización por arrecifes de coral. Se utilizaron las 14 funciones básicas para el CEC 2015 (Bent Cigar, Discus, Weierstrass, Katsuura, HappyCat, HGBat, Rosenbrock, Griewank, Rastrigin, Ackley, etc.). Principalmente se explican los algoritmos ya mencionados con los parámetros utilizados y finalmente se muestran los resultados obtenidos evidenciando la valides del teorema "no free lunch".

I. Introducción

El término Computación inteligente fue usado por primera vez en 1990, sin embargo, fue hasta 1994 que James C. Bezdek propuso y uso el término bajo la siguiente definición:

"Un sistema se llama computacionalmente inteligente si trata con datos de bajo nivel como datos numéricos, tiene un componente de reconocimiento de patrones y no usa conocimiento en el sentido IA, adicionalmente cuando comienza a exhibir adaptabilidad computacional, tolerancia a fallas, velocidad de respuesta y tasas de error que se aproximan al rendimiento humano."(Siddique & Hojjat, 2013).

En computación inteligente se busca resolver problemas de complejidad incremental con modelos de sistemas inteligentes(Engelbrecht, 2007) como los que se abordaran en este articulo; RMHC (Acenso de colinas con mutación aleatoria), AS (Recocido simulado), GA (Algoritmos genéticos), PSO (Optimización por cumulo de partículas) y CRO (Optimización por arrecifes de coral). A modo de ejemplo de los problemas que se busca resolver con estos algoritmos se tienen: knapsack problem (problema de la mochila) y traveling salesman problem (problema del vendedor viajero), para este caso de estudio se trabajó con las 14 funciones básicas para el CEC 2015(Chen, Liu, Zhang, & Liang, 2015) que son usadas para pruebas de rendimiento de algoritmos, entre las que se puede encontrar la función de Griewank, de la cual se puede destacar que el número de mínimos locales aumenta exponencialmente con el número de dimensiones(Locatelli, 2003), lo cual genera un problema para los algoritmos de optimización utilizados ya que, al ser ejecutados con el objetivo de encontrar el valor mínimo de la función, pueden encontrar el valor de un mínimo local y considerarlo el mínimo global.

II. Descripción de los algoritmos

RMHC

Es un algoritmo basado en el comportamiento de un alpinista al escalar, este buscará cual es el mejor sitio para apoyarse y seguir subiendo, el algoritmo sigue el mismo principio, pero al generar el mejor sitio de apoyo se genera un numero aleatorio dentro del rango reemplazando un elemento del vector actual.

AS

"La técnica del templado simulado tiene su origen en el proceso físico de templado del acero, el cristal o la cerámica. La técnica del templado consiste en elevar mucho la temperatura del material y luego enfriarlo lentamente para alterar sus propiedades físicas y hacerlos más resistentes."(Serrano, 1900).

El algoritmo fue aplicado con los siguientes parámetros:

1) Temperatura Máxima = 100
2) Temperatura Mínima = 0.1
3) Factor de enfriamiento = 10%
4) Numero de mutaciones por valor de temperatura = 6

GA

Los algoritmos genéticos están inspirados en la teoría de la evolución de Charles Darwin, por lo que este algoritmo tiene muchas variantes y configuraciones. Para este artículo se utilizó un modelo elitista, en cual, de una población de 20 individuos se seleccionaron, a través de un torneo, y cruzaron, con cruzamiento aritmético, a los más aptos para que estos les hereden a sus hijos las características que los hacen ser mejores que los demás, aun así, estos hijos tienen una probabilidad del 1% de sufrir una mutación, la cual los puede afectar de forma positiva o negativa. Los nuevos individuos al intentar integrarse a la población reemplazarán a los menos aptos, si son más aptos que estos.

PSO

Este algoritmo también tiene su origen en el comportamiento social de los animales, como parvadas de pájaros o bancos de peces(Kennedy & Eberhart, 1995) por lo que se empieza con un número fijo de individuos que se van a mover en el espacio en busca de la solución. Para lograr esto, cada individuo tiene conocimiento de cuál fue el sitio donde obtuvo mejor resultado, cuál es el resultado del sitio actual y cuál es el mejor sitio de los visitados por la población. Los individuos, al igual que los pájaros buscando alimento, se moverán en dirección al mejor sitio encontrado por la población. A continuación, se describe la configuración utilizada:

1) Tamaño de la población (N) = 60 individuos
2) Inercia (w) = 0.5
3) Peso del conocimiento personal ($c1$) = 1.5
4) Peso del conocimiento social ($c2$) = 1.5
5) Velocidad inicial del individuo (v) = [-2 < v > -0.2] o [0.2 < v > 2]

CRO

Este algoritmo, como su nombre lo indica está inspirado en los arrecifes de coral y su comportamiento es similar al de los GA. Se empieza con una población de individuos (corales) que habitan en un área limitada (el arrecife) representada por una matriz de NXM, estos corales compiten constantemente por el espacio, los recursos del arrecife para intentar sobrevivir ante los depredadores, la reproducción de éstos es de dos formas: sexual (broadcast spawning y brooding) y asexual (budding) (Salcedo-Sanz, Ser, Landa-Torres, Gil-López, & Portilla-Figueras, 2014).

La reproducción por broadcast spawning(Society MarineBio Conservation, 2015) los corales macho y hembra expulsan masivamente gametos para que se fertilicen y las corrientes marinas los lleven a un sitio donde se puedan desarrollar. De forma análoga se seleccionaron al azar dos individuos que fungirán como padres y usando cruzamiento aritmético de genera un individuo hijo, el cual intentara apropiarse de algún espacio del arrecife, solo un porcentaje previamente definido puede reproducirse por este medio.

En la reproducción por brooding(Society MarineBio Conservation, 2015) los corales machos son los que sueltan sus gametos a las corrientes marinas para que con suerte sean transportados a un coral hembra para fertilizarlo. Para el algoritmo, se usaron los corales restantes que no se pudieron reproducir por broadcast spawning, estos generan hijos los cuales son una mutación de sí mismos y al igual que los anteriores buscaran lugar en el arrecife para intentar desarrollarse.

La reproducción por budding(secore international, 2012) es también conocida como fragmentación, ya que cuando una parte del coral es desprendido este puede formar otra colonia si las condiciones son favorables. En el algoritmo, se genera una copia de los corales con mejor resultado arrojado, y también, buscara un sitio en el arrecife, pero se define un límite de corales idénticos.

Los parámetros usados son los siguientes:

1) Tamaño del arrecife: 10X10
2) Tamaño de la población inicial: 50
3) Cantidad de corales a Fragmentar: 5
4) Cantidad de corales idénticos: 2
5) Porcentaje de corales para reproducción por broadcast spawning: 50 %
6) Numero de intentos para establecer la larva en el arrecife: 2
7) Probabilidad de depredación: 10%
8) Numero de posibles presas: 10

III. Pruebas y resultados

Para realizar esta comparativa de algoritmos, se ejecutaron los algoritmos con el objetivo de localizar el mínimo global de las 14 funciones básicas del CEC 2015(Chen et al., 2015), bajo las siguientes condiciones:

- 20 ejecuciones independientes
- 500 evaluación de la función por ejecución
- Un vector de prueba de 10 dimensiones, donde cada valor aleatorio está comprendido en el rango de [-100,100]

Obteniendo un puntaje total con la siguiente ecuación:

$$P_{Total} = \sum_{i=1}^{14} media(F_i) + \sum_{i=1}^{14} mediana(F_i)$$

En las siguientes tablas se muestra el mejor resultado obtenido por cada algoritmo, al intentar encontrar el mínimo global de las funciones ya mencionadas. Se resalta en verde el algoritmo se muestra más estable, esto es, una desviación estándar menor.

Tabla 1 Mejor resultado de los algoritmos.

Algoritmo	Funciones	Máximo	Mínimo	Mediana	Promedio	Desviación Estándar	Tiempo Promedio
RMHC	**F9**	4.61E+00	3.99E+00	4.34E+00	4.31E+00	1.68E-01	1.18E-02
AS	**F14**	2.20E+01	2.12E+01	2.16E+01	2.16E+01	1.83E-01	5.67E-02
GA	**F11**	1.06E+00	6.77E-01	1.02E+00	9.70E-01	1.04E-01	3.70E-02
PSO	**F5**	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	3.17E+05
CRO	**F9**	4.63E+00	3.79E+00	4.42E+00	4.34E+00	2.06E-01	1.15E+00

A continuación, se muestra la tabla con los puntajes totales y se marca en verde el algoritmo cuyo puntaje fue mejor para este conjunto de funciones.

Tabla 2 Puntajes totales.

Algoritmo	Puntaje Total
RMHC	1.73E+15
SA	3.00E+15
GA	2.64E+08
PSO	7.00E+14
CRO	3.96E+13

IV. Conclusiones

Observando la tabla de resultados, cada algoritmo tiene un desempeño diferente y este varía dependiendo de la función. Se puede observar que el algoritmo PSO tiene una desviación estándar de 0 en la función 5; lo que indica que hay mayor

estabilidad en sus resultados porque es el límite de la exploración que proporciona el algoritmo para ese espacio de búsqueda con los parámetros proporcionados.

El algoritmo GA fue que obtuvo mejor puntuación total, esto es, porque obtuvo más resultados cercanos al mínimo global.

Finalmente podemos confirmar que cada algoritmo puede resolver algunos problemas mejor que otros; por lo que nuevamente se reconoce la validez del teorema "no free lunch"(Wolpert & Macready, 1997).

V. Referencias

Chen, Q., Liu, B., Zhang, Q., & Liang, J. (2015). Problem Definition and Evaluation Criteria for CEC 2015 Special Session and Competition on Bound Constrained Single-Objective Computationally Expensive Numerical Optimization. *Mysites.Ntu.Edu.Sg*. Retrieved from http://web.mysites.ntu.edu.sg/epnsugan/PublicSite/Shared Documents/CEC-2015/Expensive Problems/CEC15-single objective optimization competition - expensive.pdf

Engelbrecht, A. P. (2007). *Computational Intelligence: An Introduction*. Wiley. Retrieved from https://www.amazon.com/Computational-Intelligence-Introduction-Andries-Engelbrecht/dp/0470035617?SubscriptionId=0JYN1NVW651KCA56C102&tag=techkie-20&linkCode=xm2&camp=2025&creative=165953&creativeASIN=0470035617

Kennedy, J., & Eberhart, R. (1995). Particle swarm optimization. In *Neural Networks, 1995. Proceedings., IEEE International Conference on* (Vol. 4, pp. 1942–1948 vol.4). https://doi.org/10.1109/ICNN.1995.488968

Locatelli, M. (2003). A Note on the Griewank Test Function. *Journal of Global Optimization*, *25*(2), 169–174. https://doi.org/10.1023/A:1021956306041

Salcedo-Sanz, S., Ser, J. Del, Landa-Torres, I., Gil-López, S., & Portilla-Figueras, J. A. (2014). The Coral Reefs Optimization Algorithm: A Novel Metaheuristic for Efficiently Solving Optimization Problems. *The Scientific World Journal*, *2014*, 1–15. https://doi.org/10.1155/2014/739768

secore international. (2012). Giving coral reefs a future worldwide coral reef conservation through research, education, outreach, and restoration. Retrieved

February 12, 2018, from http://www.secore.org/site/corals/detail/coral-reproduction.15.html

Serrano, A. G. (1900). *Inteligencia artificial*. RC Libros. Retrieved from https://www.amazon.com/Inteligencia-artificial-Alberto-García-Serrano/dp/8493945021?SubscriptionId=0JYN1NVW651KCA56C102&tag=techkie-20&linkCode=xm2&camp=2025&creative=165953&creativeASIN=8493945021

Siddique, N., & Hojjat, A. (2013). *Computational Intelligence: Synergies of Fuzzy Logic, Neural Networks and Evolutionary Computing*. (J. W. & Sons, Ed.).

Society MarineBio Conservation. (2015). Coral Reefs - MarineBio.org. Retrieved February 12, 2018, from http://marinebio.org/oceans/coral-reefs/

Wolpert, D. H., & Macready, W. G. (1997). No free lunch theorems for optimization. No free lunch theorems for optimization. https://doi.org/10.1109/4235.585893