Extremal optimization aplicado al problema de mochila

Juan Baeza Baeza y Fernando Cabezas Herrera

***Resumen*— En este articulo e implementará uno de los problemas clásicos en los problemas denominados COP (Problemas de Optimización Combinatoria) llamado Problema de la Mochila (Knaspack problem) a través del novedoso método de Extremal Optimisation que propuesto por Boettcher y Percus en 1999.**

I. INTRODUCCIÓN

EL clásico problema de la mochila (KP, del inglés Knapsack Problem), conocido como un Problema de Optimización Combinatoria (COP, del inglés combinatorial optimization problems) de tipo NP-hard [1], el cual, tiene como objetivo determinar qué partes se deben agregar a la mochila para obtener el máximo beneficio, asegurando que no se exceda el volumen o la capacidad de la mochila. [2]. Si bien es cierto que ya existen soluciones al problema utilizando diferentes métodos como: Colonia de Hormigas [3], Algoritmo evolutivo [4], etc. en este trabajo, para estudiar este comportamiento en este tipo de problemas se aplica un nuevo algoritmo llamado optimización extrema (EO, del inglés Extremal Optimization). El reciente método algorítmico metaheurístico llamado optimización extrema fue propuesto por el físico Stefan Boettcher y el matemático Allon Percus en el año 1999 [5].
Los datos utilizados fueron recuperados de la plataforma EV@ de la Universidad Católica de la Santísima Concepción por el profesor Pedro Gómez con el nombre de *knapPI_11_20_1000*

II. ESTADO DEL ARTE

Julio C. Ponce, Felipe Padil, Alejandro Padilla y Miguel A. Meza en el 2006 [3] implementaron un algoritmo de optimización con colonias de hormigas para resolver el problema de la mochila dando a la solución óptima, y llegando a la conclusión de que para dar solución a este clásico, el cual es un problema NP-hard al incrementar el número de objetos, el algoritmo requiere una realización heurística para obtener una solución en tiempo polinomial.

Gladys B. Enríquez, Diana S. Partida, Santiago C. Morales en el 2017 [4] implementaron un Algoritmo Genético (GA, del inglés Genetic algorithm) para resolver el problema de la mochila dentro de una empresa comercializadora en México llegando a una solución factible de manera libre con un error de 3.23 % a un problema con amplio número de productos (250) con dos restricciones de capacidad. Finalmente, se llego a la conclusión de que GA es una herramienta factible para que empresas con recursos limitados lo puedan utilizar para la optimización de sus inventarios.

Irene Moser y Tim Hendtlass en el 2007 [6] implementaron un algoritmo de optimización extrema para resolver de problemas dinámicos de aterrizaje de aeronaves en una sola pista. Los resultados de la prueba muestran que el algoritmo EO resuelve el problema muy rápidamente y con un muy buen resultado. Sin embargo, hay muy poca variación en los resultados para la mayoría de los problemas. Esto puede indicar que la solución se reduce a encontrar cierto mínimo local, o corroborar la teoría de que el problema no es muy exigente.

III. OBJETIVOS

1. Objetivo general
   - El objetivo general de este artículo es aplicar el método de optimización extrema al problema de mochila.
2. Objetivos específicos
   - Estudiar el problema de la mochila y optimización extrema.
   - Modelar el problema a través de matemática y algorítmicamente.
   - Implementar EO al problema propuesto.
   - Analizar los resultados obtenidos.

IV. MÉTODO

IV-A. *Problema de la mochila*

El objetivo de un KP es aumentar la suma de valores de los elementos que se elegirán de un conjunto específico mediante la consideración de restricciones de recursos múltiples. Este problema ha sido ampliamente estudiado durante muchas décadas debido tanto a los intereses teóricos como a sus amplias aplicaciones en varios campos de la ingeniería, la investigación de operaciones y las ciencias de la computación y la administración [7]. Básicamente, el KP se puede formular de la siguiente manera:

$$\begin{aligned} &Maximizar \sum_{i=1}^{n} V_i X_i \\ &s.a.: \sum_{i=1}^{n} p_i X_i \leq C \\ &X_i \in \{0,1\}\, \forall i 1 \leq i \leq n \end{aligned} \tag{1}$$

donde:

Juan Baeza Baeza y Fernando Cabezas Herrera son estudiantes de la carrera de Ingeniería Civil Informática del Departamento de Ingeniería Informática, Facultad de Ingeniería, Universidad Católica de la Santísima Concepción, Concepción, Chile. Email: {jbaeza,fcabezas}@ing.ucsc.cl

Manuscrito recibido el 17 de Diciembre de 2022.

- V_i es el valor monetario del elemento i,
- X_i es la variable de decisión, 1 si el elemento está en la solución, 0 en otro caso,
- p_i es el peso del elemento i,
- C es la capacidad en peso de la mochila

IV-B. *Optimización Extrema*

Algoritmo 1 Pseudocódigo del algoritmo EO

```
Generar una solución inicial aleatoria X = (x_1, x_2, ..., x_n)
Fijar X_best = X
Generar el arreglo de probabilidades P de acuerdo a la ecuación (2)
for Un número de iteraciones do
    Evaluar y rankear el fitness τ_i for cada X_i de la peor a la mejor, 1 ≤ i ≤ n
    Seleccionar un componente j basado en la probabilidad de su ranking P_i usando RWS
    x_j = Generar un valor distinto a x_j
    Eva(X) = Evaluar la nueva solución
    if Eva(X) <Eva(X_best) then
        X_best = X
    end if
end for
return  X_best y Eva(X_best)
```

El algoritmo de optimización extrema es una metaheurística de búsqueda global, basada en un modelo de evolución natural [8], y especialmente diseñada para ser utilizada en problemas complejos de optimización. Tiene sus fundamentos en la teoría de la criticidad autoorganizada (SOC, del inglés self-organized criticality), que ha sido utilizada para explicar la firmas de leyes de potencia que emergen de muchos sistemas complejos. [9]

La peculiaridad de este método es que requiere solo un parámetro en el algoritmo τ y utiliza una estructura algorítmica simple porque solo funciona en uno. Una solución que evoluciona de acuerdo a una función de aptitud que estima la contribución de una especie o variable a la solución global. Esto asegura un uso mínimo Cantidad de memoria durante la ejecución del método [10]. Su funcionamiento trata de lo siguiente:

1. Cada componente de la solución representa a una especie.
2. Se asigna un fitness a cada especie.
3. P_i es la probabilidad de que la i^{th} especie sea escogida

$$P_i = i^{-\tau} \forall i 1 \leq i \leq n \tag{2}$$

4. Se rankean las especies de la peor a la mejor.
5. Se selecciona una especie utilizando el método de selección de la ruleta (RWS, del inglés Roulette Wheel Selection) usando la probabilidad P_i.
6. Se incorpora o elimina una especie de la solución.

IV-C. *Función de aptitud*

La función de aptitud fue calculada en base a la relación valor versus peso, por lo cual se dividen estos valores y son ordenados de menor a mayor resultado.

V. Resultados

Esta sección evalúa la mayor suma del valor de objetos que pueden ser asignados, sin que la suma de sus pesos supere una capacidad límite. Los valores se encuentran en el rango [38;1020] y los pesos pertenecen a [132;970].

Los parámetros utilizados para el algoritmo fueron 50 semillas y 1000 iteraciones, por otra parte, para el desarrollo del problema τ fue designado en valores pertenecientes al conjunto [1.4, 1.5, 1.6, 1.7, 1.8]. Como se puede observar en la figura 1, un ajuste apropiado para nuestros datos, es un τ de 1.5, ya que contiene una media alta y bajo nivel de dispersión de los datos, aún presentando mayor cantidad de datos atípicos. También podemos observar que a partir de un τ igual a 1.8 la media comienza a bajar, por lo cual no está encontrando un mejor resultado, sino al contrario. Por lo cual se concluye que a partir de este valor en adelante los resultados tienden a empeorar.

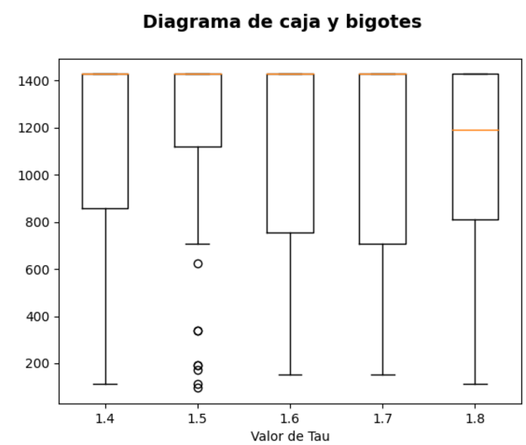

Figura 1. Valores de la mochila a cierto valor de Tau

VI. Conclusiones

En conclusión de lo anterior, EO nos permite realizar un problema NP-hard como es KP, el cual es un problema combinatorio. Con el algoritmo propuesto se ratificó llegar a la solución planteada en un inicio, ya que la media del diagrama de caja y bigotes fue visto en su mayoría de valores, cerca de 1400. Sin embargo, ver que a valores superiores de τ no fue posible llegar constantemente a la solución hace creer que para ellos es necesario ampliar el tiempo de reproducción para estas pruebas, o en su defecto aplicar otro algoritmo metaheurístico.

Referencias

[1] A. Fuentes-Penna, D. Vélez-Díaz, S. Moreno-Gutiérrez, M. Martínez-Cervantes, O. Sánchez-Muñoz *et al.*, "Problema de la mochila (knapsack problem)," *XIKUA Boletín Científico de la Escuela Superior de Tlahuelilpan*, vol. 3, no. 6, 2015.

[2] C. A. C. Flórez, R. A. B. Ocampo, and A. M. Cabrera, "Algoritmo multiobjetivo nsga ii aplicado al problema de la mochila." *Scientia et technica*, vol. 2, no. 39, pp. 206–211, 2008.

[3] J. C. Ponce, F. Padilla, A. Padilla, M. A. Meza, and C. A. Ochoa, "Achpm: Algoritmo de optimización con colonia de hormigas para el problema de la mochila," *Departamento de Sistemas Electrónicos, Universidad Autónoma de Aguascalientes*, 2006.

[4] D. A. Rossit, M. Méndez González, M. Frutos, and B. González Landín, "Algoritmo evolutivo híbrido basado en la división del espacio de los objetivos para el problema de la mochila bi-objetivo," 2020.

[5] Y.-Z. Lu, Y.-W. Chen, M.-R. Chen, P. Chen, and G. Zeng, "Extremal optimization: Fundamentals, algorithms, and applications," *Journal of Process Control*, vol. 16, pp. 795–808, 2016.

[6] I. Moser and T. Hendtlass, "Solving dynamic single-runway aircraft landing problems with extremal optimisation," in *2007 IEEE Symposium on Computational Intelligence in Scheduling*. IEEE, 2007, pp. 206–211.

[7] F. Djannaty and S. Doostdar, "A hybrid genetic algorithm for the multi-dimensional knapsack problem," *International Journal of Contemporary Mathematical Sciences*, vol. 3, no. 9, pp. 443–456, 2008.

[8] P. Bak and K. Sneppen, "Punctuated equilibrium and criticality in a simple model of evolution," *Phys. Rev. Lett.*, vol. 71, pp. 4083–4086, Dec 1993. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevLett.71.4083

[9] P. Bak, *how nature works*. na, 2002.

[10] M. Randall and A. Lewis, "Population extremal optimisation for discrete multi-objective optimisation problems," *Information Sciences*, vol. 367, pp. 390–402, 2016.