

Problema del Bin Packing unidimensional

Miguel Ángel Álvarez Guzmán
Departamento Ingeniería Electrónica
Universidad de Antioquia
Medellín, Colombia
miguel.alvarezg@udea.edu.co

Daniel Felipe Meneses Rojas
Departamento Ingeniería Electrónica
Universidad de Antioquia
Medellín, Colombia
dfelipe.meneses@udea.edu.co

Abstract— Este trabajo aborda el Problema de Empaquetamiento en una Dimensión, utilizando Recocido Simulado. El problema consiste en minimizar el número de contenedores necesarios para empaquetar un conjunto dado de objetos con pesos variados, respetando un límite máximo de peso por contenedor. La estrategia propuesta inicia con una solución aleatoria y explora iterativamente el espacio de búsqueda, empleando una aceptación probabilística de soluciones subóptimas para evitar mínimos locales. Los resultados experimentales demuestran que el reconocido simulado equilibra eficazmente la calidad de la solución y la eficiencia computacional, ofreciendo resultados competitivos en diversas instancias del problema.

Palabras claves— Empaquetado, unidimensional, recocido simulado, pesos, contenedores.

I. Introducción

Este informe se explica la implementación de recocido simulado para resolver el problema de empaque unidimensional, evaluando el desempeño en diferentes instancias y analizando su capacidad para generar soluciones eficientes y competitivas.

El problema de empaquetamiento unidimensional se centra en determinar la mejor manera de empaquetar un conjunto de objetos de diferentes pesos en el menor número de contenedores, respetando una capacidad máxima por cada uno. Debido a la naturaleza NP-Completa del problema, es complicado resolverlo por métodos exactos.

El recocido simulado está inspirado en el proceso de recocido de materiales, en el que la materia se enfría gradualmente hasta alcanzar su estado energético más bajo. En el contexto de la optimización, este método nos permite explorar soluciones subóptimas de forma controlada para evitar mínimos locales y mejorar la calidad de las soluciones obtenidas.

II. Definición del problema

El problema de Empaquetamiento unidimensional busca de manera eficiente la asignación de objetos de diferentes tamaños o pesos en un número mínimo de contenedores, respetando ciertas restricciones de capacidad.

Supongamos que se tiene un conjunto de N objetos, cada uno con un peso W_i y un número ilimitado de contenedores, cada uno con una capacidad máxima C . El objetivo es agrupar los objetos en el menor número posible de contenedores, de manera que la

suma de los pesos de los objetos asignados a cada contenedor no exceda su capacidad C y todos los objetos sean asignados a algún contenedor.

III. Modelo de optimización del problema

Matemáticamente, podemos formular el problema de la siguiente manera:

Parámetros:

- W_i : Peso del objeto i .
- C : Capacidad máxima de cada contenedor.
- B : Numero de contenedores

Variables:

- X_{ij} : Variable binaria que indica si el objeto i está asignado al contenedor j .
- X_j : Variable binaria que indica si el contenedor j es utilizado.

Restricciones:

- Cada objeto debe ser asignado a un único contenedor:

$$\sum_{j=1}^B x_{ij} = 1, \quad \forall i \in \{1, 2, \dots, n\}. \quad (1)$$

- La suma de los pesos de los objetos en un contenedor no puede exceder su capacidad:

$$\sum_{j=1}^B w_i \cdot x_{ij} \leq C \cdot y_j, \quad \forall j \in \{1, 2, \dots, B\}. \quad (2)$$

- Las variables son binarias:

$$x_{ij}, y_j \in \{0, 1\}. \quad (3)$$

Función objetivo:

$$\text{Minimizar} \quad \sum_{j=1}^B y_j \quad (4)$$

IV. Recocido simulado

El método recocido simulado es un algoritmo de búsqueda metaheurística para problemas de optimización global; el objetivo general de este tipo de algoritmos es encontrar una buena aproximación al valor óptimo de una función en un espacio de búsqueda grande. Dicho "óptimo global" corresponde a la solución del problema de interés para el que no existe un mejor valor. En el caso de que tal problema sea de minimización, el óptimo global será aquél para el cual la función objetivo tenga el más pequeño posible de todos los de su espacio de búsqueda.

Y para un problema de maximización, el óptimo global es aquél con el valor más alto posible.

Conforme la temperatura se reduce, el algoritmo es más selectivo, concentrándose en optimizar las siguientes soluciones. Un buen método se basa en un equilibrio apropiado entre la exploración y la explotación, que se basa en la velocidad de enfriamiento, la cantidad de iteraciones y la aceptación de soluciones. Este enfoque para problemas complejos con varios locales, tales como el diseño de circuitos, la organización de rutas o la optimización combinatoria.

V. Resultados

Para realizar el análisis de los resultados se ha realizado una investigación basada en iteración de 4 parámetros para la obtención de las relaciones entre los óptimos obtenidos por Gurobi y los óptimos obtenidos por el algoritmo del recocido simulado. Los parámetros alterados son explicados a continuación:

- Capacidad (50-100): indica la capacidad máxima de los contenedores que fue indicada para cada una de las simulaciones
- Temperatura (500-2000): indica la temperatura inicial máxima a la cual se inicia el algoritmo y a la que se restablece al realizar recalentamiento.
- Factor de enfriamiento (0.8-0.95): indica el factor al cual se le va reduciendo la temperatura en cada iteración.
- Número de iteraciones (100-10000): indica la cantidad de iteraciones que hará el algoritmo por cada instancia.
- Brecha (%): indica el porcentaje de diferencia que hay entre el valor obtenido por el algoritmo Gurobi con respecto al algoritmo del recocido simulado.

Para realizar las simulaciones se realizaron 10000 iteraciones de los algoritmos en las cuales se variaban aleatoriamente los 4 parámetros y se encontraba la brecha de cada iteración, luego se realizaron las comparaciones mediante graficas que muestran la variabilidad de la brecha al cambiar los parámetros, una brecha menor significa una obtención de un óptimo más cercano al del algoritmo Gurobi. Finalmente se encontraron los valores individuales con los que se encontraban la brecha más grande y se mostraron en la tabla.

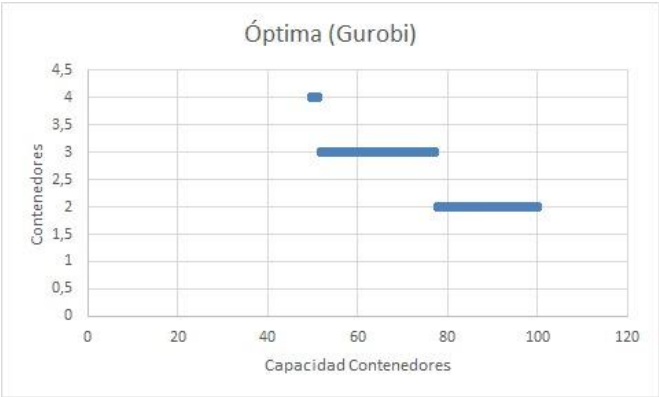


Figura 1. Relación de la cantidad de los contenedores usados basados en su capacidad.

Capacidad	T ₀	α	Iteraciones	Óptima (Gurobi)	Óptima (Recocido)	Brecha (%)
88	1000	0,9	100	2	9	350
94	500	0,95	100	2	9	350
98	1500	0,95	100	2	9	350
78	1500	0,95	100	2	9	350
85	2000	0,95	100	2	9	350
96	500	0,95	100	2	9	350
93	500	0,95	100	2	9	350
95	2000	0,95	100	2	9	350
96	2000	0,95	100	2	9	350
98	1500	0,95	100	2	9	350

Tabla 1. Valores con los que se encontraron mayor porcentaje de brecha entre el Gurobi y el recocido simulado

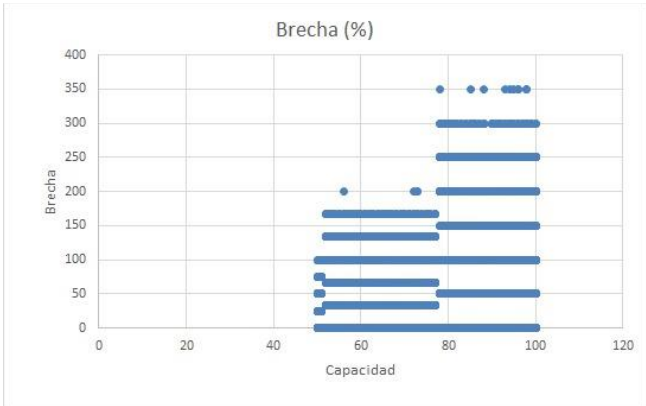


Figura 2. Relación entre capacidad y brecha.

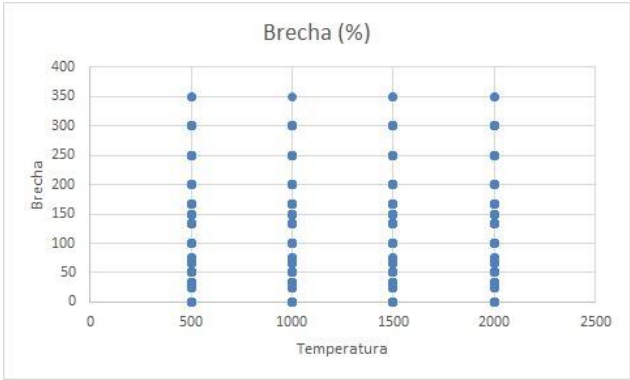


Figura 3. Relación entre temperatura y brecha.



Figura 4. Relación entre el número de iteraciones y la brecha (logarítmica)

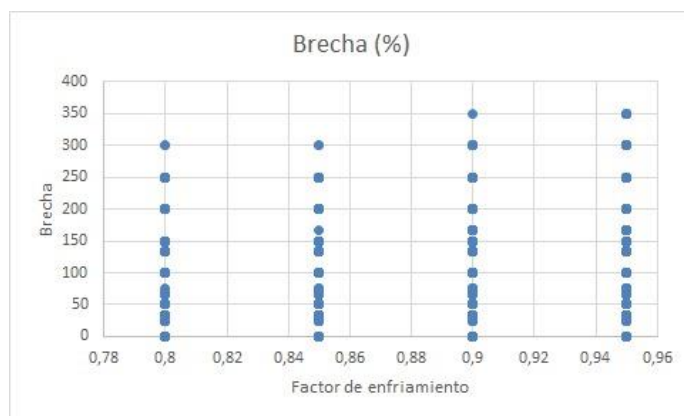


Figura 5. Relación entre el factor de enfriamiento y la brecha

VI. Análisis de resultados.

Los análisis de resultados se dan en función de las gráficas y que se puede apreciar de cada una de ellas.

En la figura 1 se realiza una comparación entre la capacidad de los contenedores usados y los contenedores que fueron requeridos según el algoritmo de Gurobi. La grafica muestra un resultado lógico según la realidad, a mayor capacidad de los contenedores, menor es el número de contenedores necesarios.

En la figura 2 se muestra la relación existente entre la capacidad de los contenedores y la brecha existente entre los algoritmos, los resultados muestran una conjetura interesante entre esta relación, ya que al aumentar la capacidad de los contenedores también se incrementa la brecha en algunos casos específicos, aunque la mayoría tiende a estar estables en un valor de brecha inferior al 200%. Este fenómeno es posible explicarlo dado a que a mayor capacidad:

- Aumenta la complejidad del espacio de búsqueda (más soluciones válidas).
- Reduce la presión de las restricciones, permitiendo soluciones subóptimas.
- Resalta las limitaciones del recocido simulado frente a métodos exactos como Gurobi.

En la figura 3 se muestra la relación entre el valor de la temperatura inicial y la brecha entre las soluciones, los resultados son claros según la gráfica, muestran que no existe una relación apreciable entre la temperatura inicial y el porcentaje de brecha, es posible que la brecha se vea más influenciada por otros parámetros, es posible indicar que La temperatura inicial no tiene un efecto determinante por sí sola sobre la brecha. Es probable que las configuraciones subóptimas de otros parámetros sean las principales responsables de las brechas grandes.

En la figura 4 se presenta la gráfica que compara el número de iteraciones con la brecha entre los algoritmos. Los resultados muestran un resultado muy apreciable y distinguible, a mayor número de iteraciones menor es la brecha entre los algoritmos, este también muestra que la necesidad de numero de iteraciones al transcurrir el tiempo se va volviendo logarítmico, es decir que llega un punto donde elegir un mayor número de iteraciones no simboliza una menor brecha entre las soluciones, este se puede considerar como el parámetro que más afecta al momento de encontrar un resultado mejor o más aproximado al optimo.

En la gráfica 5 se muestra la gráfica que relaciona el factor de enfriamiento con la brecha. Esta muestra muy sutilmente que a mayor factor de enfriamiento mayor es el valor de la brecha, por lo que es preferible que para la solución de recocido el parámetro del factor de enfriamiento tienda a ser bajo.

En la tabla 1 se muestran los valores con los que se logró un mayor valor de brecha entre las soluciones, ósea que las diferencias entre la solución optima de Gurobi y la del recocido fueron máximas. En la tabla predomina lo siguiente:

- La cantidad de iteraciones en todos los casos fue la mínima posible
- El valor de la temperatura inicial fue despreciable ya que no hay un patrón concreto
- El valor del factor de enfriamiento tiende a ser el máximo, ósea de 0.95 únicamente variando 0.05 puntos porcentuales en un caso
- La capacidad de los contenedores tiende a estar en valores altos en el rango de 80-100.

Estos resultados confirman las gráficas anteriores, ya que muestran que en cada caso se cumplen los argumentos expuestos en los distintos ítems de trabajo.

VII. Conclusiones

- Es evidente que la cantidad de iteraciones tiene un impacto importante en la disminución de la brecha entre las soluciones de Gurobi y recocido simulado, mientras que el parámetro de la temperatura inicial tiene un menos significativo.
- El recocido simulado demostró ser una técnica efectiva para aproximarse a soluciones óptimas del problema, ofreciendo un equilibrio razonable entre calidad de las

soluciones y tiempo computacional en comparación con métodos exactos como Gurobi.

- Aunque Gurobi nos proporciona soluciones óptimas, el recocido simulado es capaz de generar soluciones satisfactorias en menos tiempo, siendo útil en aplicaciones donde se prioriza la rapidez sobre la exactitud absoluta.
- De manera general se pudo demostrar la viabilidad y eficiencia del recocido simulado como una estrategia de optimización para abordar el problema de empaquetado unidimensional, proporcionando soluciones aproximadas en un tiempo razonable y adaptándose a las características de cada instancia del problema. Aunque no tiene la misma precisión del método exacto Gurobi, su capacidad para explorar grandes espacios de búsqueda y evitar mínimos locales lo convierte en una alternativa práctica para problemas complejos y de gran escala. Es importancia de una adecuada parametrización del algoritmo y la necesidad de ajustar parámetros clave, como el número de iteraciones y el factor de enfriamiento, para maximizar su desempeño. Finalmente, el recocido simulado se posiciona como una herramienta versátil y eficiente para la solución de problemas de optimización combinatoria en ingeniería.

VIII. Referencias

- [1] Repositorio UTP, "Contenido del documento," *Universidad Tecnológica de Pereira*, [En línea]. Disponible: <https://repositorio.utp.edu.co/server/api/core/bitstreams/642e7991-8e5d-46d3-be19-26ac85269ce5/content>.
- [2] MQL5, "Introducción a la programación en MQL5 para MetaTrader 5," MQL5, [En línea]. Disponible: <https://www.mql5.com/es/articles/138511>.
- [3] JBC Games, "Trabajo Final Optimización," *GitHub Repository*, [En línea]. Disponible: <https://github.com/jbcgames/Trabajo-Final-Optimizacion>.