

Resumen

El objetivo de esta tarea es implementar una heurística constructiva y una búsqueda local por escalada estocástica que permita evaluación incremental.

Métodos

Heurística constructiva

Una heurística es un método de aproximación hecho a medida para un problema concreto. Existen dos tipos de heurísticas en optimización: de trayectoria y constructivas. Las heurísticas constructivas son procedimientos iterativos que en cada paso, añaden elementos hasta completar una solución. Por ejemplo, en el problema del viajero, puede tenerse la solución con un crecimiento por un lado (Nearest Neighbour Heuristic) o un doble crecimiento (Double-Ended Nearest Neighbor Heuristic). Usualmente estas heurísticas son deterministas y están basadas en seleccionar en cada iteración, el elemento con la mejor evaluación [1].

Para el problema de Bin Packing (BP) existen muchas heurísticas constructivas: First-fit-decreasing (FFD) que va empacando en orden decreciente los elementos en alguno de los contenedores donde haya espacio; Best-Fit-Decreasing (BFD) que empaqueta los elementos faltantes en los contenedores donde haya espacio; Bin-Oriented-Heuristic (BOH), MTP, entre otras [2].

En este caso, se trabajó con la heurística de BFD, donde se busca insertar los elementos en los espacios vacíos. En la Fig.1 se muestra un ejemplo de cómo funciona esta heurística, en este caso el tamaño de cada contenedor es de máximo 10 unidades, y se ingresan los elementos = { 2, 5, 4, 7, 1, 3, 8 }

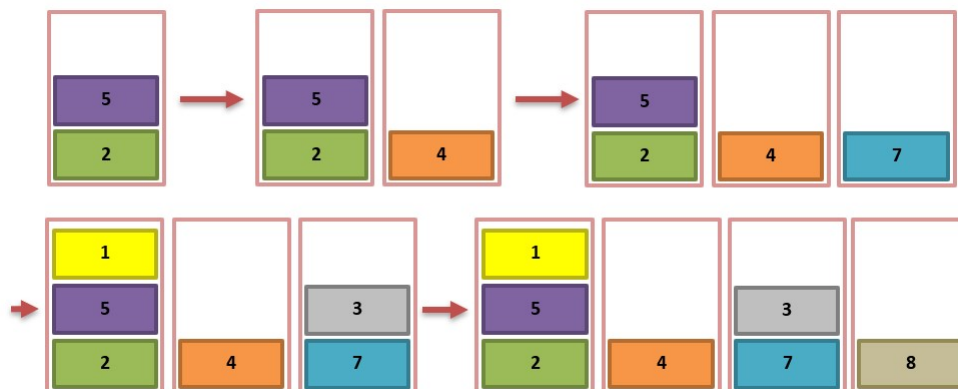


Figura 1: Procedimiento de BFD.

Una de las observaciones que se debe de tener para esta heurística, es el tiempo de búsqueda. Según la literatura, esta se puede implementar utilizando un árbol de búsqueda con actualizaciones, y eliminando nodos cada que estén llenos. Sin embargo, también se pueden utilizar otras herramientas para que el tiempo de búsqueda de espacio en algún contenedor no sea lineal. En este caso, se utilizó la función *lower_bound* de C++, que regresa un iterador al elemento más pequeño pero más grande que el elemento, o igual al elemento, y lo hace en $O(\log N)$.

Búsqueda local por escalada estocástica

Este procedimiento, toma cualquier permutación arbitraria de los elementos que no están dentro de un contenedor (para el caso de BP) y después inserta de izquierda a derecha en el contenedor que tenga el mínimo espacio requerido, y que no supere el tamaño del contenedor. Si no hay un contenedor posible, se abre otro y se inserta en ese lugar. De esta forma, se espera que haya menos tendencia a cierto tipo de soluciones, y que se mueva las combinaciones a las primeras que mejoren.

Se desarrolló el método mencionado arriba - que es muy parecido al método de la heurística constructiva, pero este comienza con una permutación aleatoria - y para hacer la evaluación incremental, en cada iteración se revisaba si había un contenedor completamente lleno, es decir, que el espacio vacío fuera nulo, y se guardaban estos elementos, y no se utilizaban para la siguiente solución. Por ejemplo, en la Fig. 2 se muestra que con la primer permutación que se hizo, se llenó el primer contenedor con los elementos 3 y 7, después estos se quedan como una buena solución y ya no entran a la siguiente permutación, así, se espera que llenando los contenedores, disminuya la cantidad de estos. Además, en la Fig. 2 disminuyó de 4 contenedores a 3.

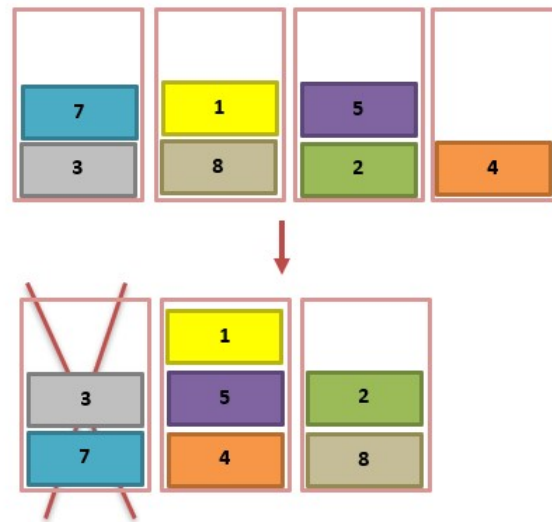


Figura 2: Procedimiento de búsqueda local estocástica.

Resultados

Se escogieron tres instancias: Falkenauer 120-09, Falkenauer 250-10, y Falkenauer 500-07, donde el término al lado del nombre indica cuántos elementos hay, y el último indica una instancia de veinte que había en total por tamaño (se escogieron de forma aleatoria).

Heurística constructiva

En la Tabla. 1 se muestran los resultados al aplicar el algoritmo BFD, en esta se muestra que conforme aumentaba la cantidad de elementos por acomodar, había más diferencia respecto a los mejores resultados conocidos. Además, en la Fig.3 se observar cómo en la instancia de tamaño más grande (Instancia U500-07), hay más elementos en el cuartil.

Instancia	Min	Media	Máximo	Solución óptima	Diferencia
Falkenauer 120 - 09	49	50	52	46	3
Falkenauer 250 - 10	113	115	117	105	8
Falkenauer 500 - 07	219	221	225	204	15

Cuadro 1: Cantidad de contenedores utilizando la heurística constructiva.

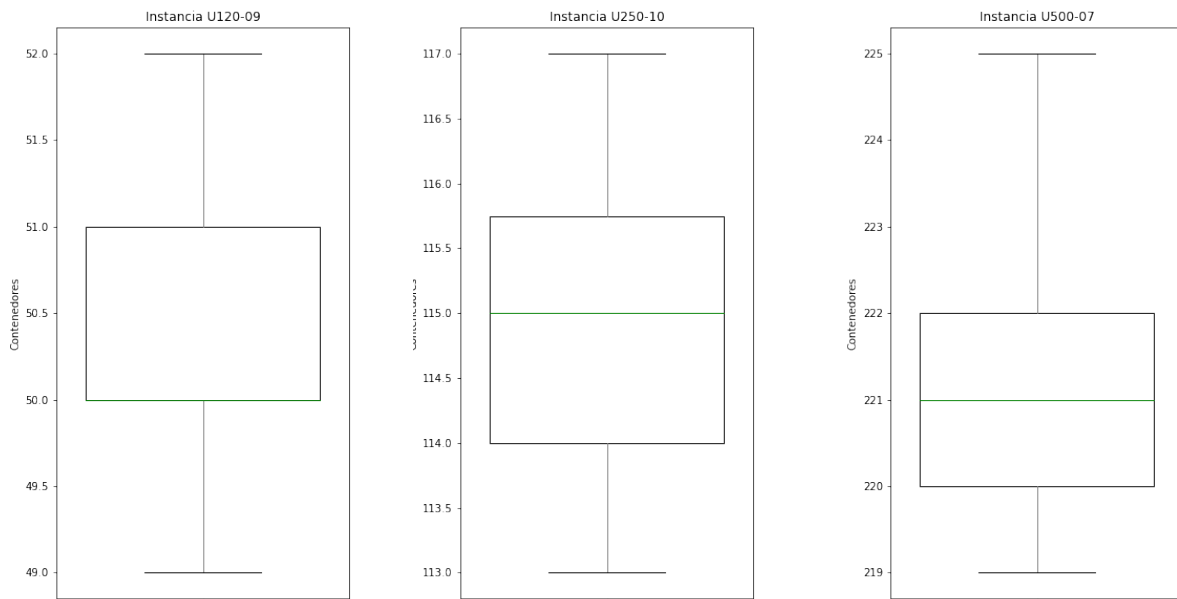


Figura 3: Boxplot del BFD

En la Tabla. 2, se muestran los tiempos de ejecución al implementar en BFD, y también se observa el aumento de tiempo en relación con el tamaño de la instancia.

Instancia	Min	Media	Máximo
Falkenauer 120 - 09	0.000089	0.000106	0.000743
Falkenauer 250 - 10	0.000200	0.000282	0.003581
Falkenauer 500 - 07	0.000413	0.000678	0.005597

Cuadro 2: Tiempos de ejecución de BFD en segundos.

Búsqueda local por escalada estocástica

En la Tabla. 3 se muestra los resultados al implementar la búsqueda local, se observa que hay menos diferencia entre la solución óptima conocida y que sus resultados mejoraron respecto al primer algoritmo que no tenía componente aleatorio.

Instancia	Min	Media	Máximo	Solución óptima	Diferencia
Falkenauer 120 - 09	48	51	54	46	2
Falkenauer 250 - 10	109	112	114	105	4
Falkenauer 500 - 07	209	213	218	204	5

Cuadro 3: Resultados con búsqueda local por escalada estocástica.

Sin embargo, los tiempos aumentaron considerablemente, esto posiblemente fue porque se añadió una función lineal para buscar los contenedores que ya estaban llenos y quedarse con el índice del último encontrado (este índice contenía los elementos que lo hacían cero). Asimismo, en la Fig.4 se tienen cuartiles más pequeños y muestra datos atípicos, que coinciden con los máximos y los mínimos presentados en la Tabla. 3.

Instancia	Min	Media	Máximo
Falkenauer 120 - 09	0.001950	0.003320	0.009017
Falkenauer 250 - 10	0.009425	0.01119	0.035912
Falkenauer 500 - 07	0.039609	0.053084	0.110738

Cuadro 4: Tiempos de ejecución búsqueda local por escalada estocástica en segundos.

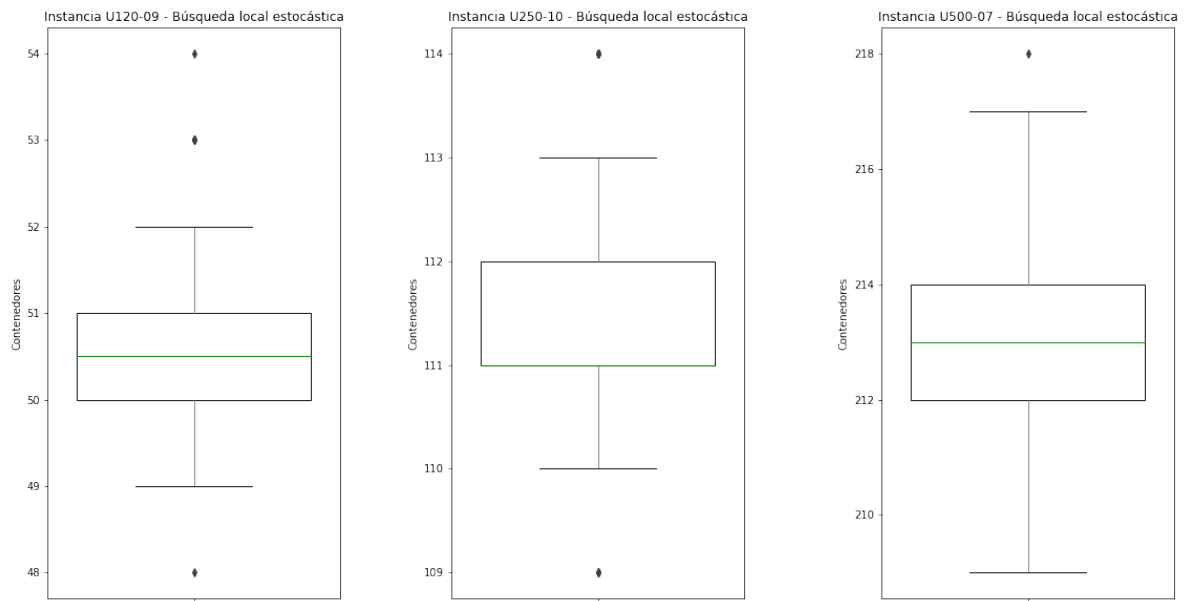


Figura 4: Boxplot de la búsqueda local.

Conclusiones

No se hizo la comparación con la tarea anterior, porque se cambiaron dos instancias, debido a que las tres instancias antiguas tenían el mismo tamaño. Respecto a la instancia que no se cambió (Faulkenauer 120-09), el resultado previo devolvió mínimo 56 contenedores, y con estos nuevos métodos, el máximo no superaba ese número. Sin embargo, el plantear cómo iba a mejorar la solución por evaluación incremental, fue un proceso complicado. Finalmente, debido a que en un método los tiempos son mejores, pero los resultados no; se utilizará la que tiene mejores tiempos por la implementación desarrollada.

Referencias

- [1] Yadira Rosabal Alfonso, Cynthia Porras Nodarse, and Jenny Fajardo Calderín. Heurísticas de construcción voraces para el problema de máxima cobertura dinámico. *Revista Cubana de Ciencias Informáticas*, 13(4):81–97, 2019.
- [2] Krzysztof Fleszar and Christoforos Charalambous. Average-weight-controlled bin-oriented heuristics for the one-dimensional bin-packing problem. *European Journal of Operational Research*, 210(2):176–184, 2011.