Universidad Tecnológica de La Habana
José Antonio Echeverría

Ingeniería Mecánica
2020;23(2):e605. ISSN 1815-5944

Artículo de investigación científica y tecnológica

Influencia de distribuciones estadísticas en la complejidad de instancias del problema de empaquetamiento con tamaño y costo variable

Influence of statistical distributions in the hardness of instances of the variable size and cost bin packing problem

Jorge Herrera-Franklin[I], Alejandro Rosete-Suárez[II, *], Milton García-Borroto[II], Suitberto Cabrera-García[III]

I. Centro de Investigación y Manejo Ambiental del Transporte, CIMAB. La Habana, Cuba.
II. Universidad Tecnológica de La Habana José Antonio Echeverría. Facultad de Ingeniería Informática. La Habana, Cuba
III. Universidad Politécnica de Valencia. Dpto. de Estadística, Investigación Operativa Aplicadas y Calidad. Valencia, España
*Autor de correspondencia: rosete@ceis.cujae.edu.cu

Este documento posee una licencia Creative Commons Reconocimiento-No Comercial 4.0 internacional

Recibido: 14 de febrero de 2020 Aceptado: 30 de marzo de 2020

Resumen

El Problema de Empaquetamiento con Tamaño y Costo Variable consiste en minimizar el costo de todos los contenedores utilizados para empacar un conjunto de elementos. En los trabajos previos encontrados se han presentado y estudiado varios algoritmos para resolver diferentes instancias de este problema donde los pesos de los elementos a empaquetar siguen Distribución Uniforme. El objetivo del presente trabajo fue estudiar la influencia de la distribución estadística que siguen los componentes de las instancias del problema en su complejidad. Para ello, se diseñaron tres conjuntos de instancias de referencia que siguen diferentes distribuciones estadísticas y se utilizó el solver SCIP para resolver las mismas. De estos experimentos se obtuvo que las instancias donde los pesos de los elementos siguen Distribución Normal y las capacidades de los contenedores siguen la Distribución Weibull son más difíciles de resolver que las instancias con características similares a las estudiadas en los artículos previos.

Palabras claves: optimización combinatoria; problema de empaquetamiento con tamaño y costo variable; instancias de referencia; análisis estadístico.

Abstract

The Variable Size and Cost Bin Packing Problem consists of minimizing the cost of all the containers used to pack a set of items. In the previous works, several algorithms have been presented and studied, to solve different instances of this problem where the weights of the elements to be packed follow Uniform Distribution. The objective of this work is to study the influence of the statistical distribution that the components of the instances of the problem follow in their complexity. For this, three sets of reference instances were designed that follow different statistical distributions and the SCIP solver was used to solve them. From these experiments, it was obtained that the instances where the weights of the elements follow the Normal Distribution and the capacities of the containers follow the Weibull Distribution are more difficult to solve than the instances with characteristics similar to those studied in the previous articles.

Key words: combinatorial optimization; variable size and cost bin packing problem; benchmark instances; statistical analysis.

Cómo citar este artículo:
Herrera Franklin J, Rosete Suárez A, García Borroto A, et al. Influencia de distribuciones estadísticas en la complejidad de instancias del problema de empaquetamiento con tamaño y costo variable. Ingeniería Mecánica. 2020;23(2):e605. ISSN 1815-5944.

Introducción

En Investigación de Operaciones existen muchos modelos matemáticos correspondientes a situaciones de la vida real que se agrupan en familias de problemas. Una de estas familias es el Problema de Empaquetamiento (*Bin Packing Problem*, BPP) [1] el cual cuenta con numerosas variantes, algunas muy recientes [2]. Una de los problemas de esta familia es el Problema de Empaquetamiento con Tamaño y Costo Variable (*Variable Size and Cost Bin Packing Problem*, *VSCBPP*) que se aborda en el presente trabajo. Este problema tiene importantes aplicaciones en la organización de sistemas de transporte, tanto marítimos como terrestres. Este problema

consiste en empacar un conjunto de ítems en un conjunto de contenedores de varios tipos (tamaños) y con una cantidad infinita de cada tamaño, minimizando el costo de todos los contenedores usados [3].

En [4, 5] se mostraron soluciones del VSCBPP empleando la Búsqueda de Vecindad Variable usando algunos operadores previamente definidos en [6]. Estos artículos se basan en un conjunto de aspectos propuestos en [7] donde se estudian tres tipos de función de costo, pero no se estudia la influencia de estas ni de la capacidad de los contenedores en la complejidad de las instancias. Por otro lado, en [8] se estudió el comportamiento del algoritmo *Harmony Search* empleando un conjunto de datos muy similar al anterior, con variaciones en el tamaño de las instancias y la función de costo de los tipos de contenedor. Otras investigaciones de corte teórico [9, 10, 11], sobre algoritmos o variantes de este problema tampoco estudian la influencia de las distribuciones estadística de los datos de las instancias en la complejidad de las mismas. En [12] también utilizaron un conjunto de datos muy similar al presentado en [7]. Parte de estos casos están disponibles en la biblioteca OR [13], pero hay otros, generados para su estudio que no se publican y no es trivial replicarlos. Existen otros trabajos más [14, 15] en que no declararon las instancias de referencia debido a su enfoque hacia adaptaciones específicas de este problema. Solo en [16] se incluyó parcialmente un conjunto publicado donde se utilizan instancias pequeñas en una propuesta difusa del VSCBPP.

En los trabajos encontrados y descritos anteriormente, la función lineal de costo de los contenedores está basada en una relación costo-capacidad basado en una igualdad, sin incluir parámetros como pendiente o intersección. Esto influye en la forma de seleccionar los contenedores donde empacar ítems, incluso si hay una lista ordenada, como es el caso de las heurísticas *First Fit Decreening* (FFD) y *Best Fit Decreasing* (BFD) [1].

En cuanto a la distribución estadística de los pesos de los ítems y las capacidades de los contenedores, se puede considerar que los resultados publicados son incompletos pues en todos se sigue una única distribución estadística: la Distribución Uniforme. Esto tiene una implicación directa en la demanda que debe empacarse en los contenedores. Sin embargo, concentrar los ítems cerca de un valor dentro del rango de los tamaños de los contenedores (como ocurre en otras distribuciones) puede afectar la complejidad de la elección del contenedor más barato para empacar ciertos ítems.

Con todos estos elementos, se estableció el objetivo principal del presente trabajo: demostrar la influencia de las distribuciones estadísticas en la complejidad del problema VSCBPP. Los resultados que se presentan demuestran que varias distribuciones estadísticas no utilizadas en los trabajos previos (particularmente la Distribución Normal y la Distribución Weibull) implican problemas de mayor complejidad.

Métodos

Para los experimentos fueron generados tres conjuntos de datos mediante una herramienta disponible en http://www.cimab.transnet.cu/files/asocfile-5528.zip . Cada uno contiene tres grupos de tamaños crecientes y en cada uno se utilizaron las tres funciones de costos diferentes. Para cada caso, el rango de los pesos de los ítems se estableció entre 20 y 120 condicionándose un empaquetado heterogéneo y más complejo. El límite superior seleccionado utiliza el número mínimo de elementos por tipo de contenedor y garantiza que todos los ítems que caben en cada tipo puedan ser empaquetados solo en dicho tipo, manteniéndose el axioma de que exista una lista infinita de contenedores de cada tipo.

Las instancias se resolvieron con un *solver* de Programación de Entera Mixta (del inglés *Mixed Integer Programming*), en este caso el SCIP [17], utilizándose en las instancias el formato ZIMPL [18]. Las ejecuciones se realizaron en un procesador Intel Core i5 con 2,4 GHz de velocidad de reloj, 8 Gb de RAM, estableciendo un número máximo de 4 subprocesos paralelos. Para cada instancia, el tiempo máximo de ejecución se estableció en 1 hora, y las instancias más grandes fueron las que tenían 1000 elementos y 6 tipos de contenedor. Se consideraron dos formas de medir la complejidad de las instancias: el tiempo para resolverlas y el gap (diferencia estimada entre la calidad de la solución aproximada que se encontró respecto a la solución óptima, cuando esta no es posible garantizar la optimalidad en el tiempo prefijado).

Conjuntos de datos

Considerando I el número de conjuntos de elementos, n el número de ítems en cada conjunto, m el número de tamaños del conjunto de contenedores, W los tamaños de los contenedores en el conjunto, los conjuntos tienen las siguientes características:

- Conjunto 1: tiene como objetivo medir el efecto del tamaño de la instancia y la función de costo del tipo de contenedor en la complejidad del problema. Los pesos de los ítems se generaron aleatoriamente siguiendo una Distribución Uniforme en el rango especificado.
    - $I = 3$, n ∈ {25,100}, $m = 3$, $W \in \{50,100,150\}$, 6 instancias.
    - $I = 3$, $n \in \{200,500,1000\}$, $m = 6$, W ∈ {150,75,100,125,150,175}, 9 instancias.
    - $I = 1$, n ∈ {2000}, $m = 12$, $W \in \{25,50,75,100,125,150,175,200,225,250,275,300\}$, 3 instancias.

- Conjunto 2: contiene variaciones en la distribución de los pesos de los elementos debido al uso de las distribuciones Uniforme, Gamma, Normal, Weibull y Ruleta. Los parámetros para cada distribución son: *U* [20,120], *N* (60,15), para Gamma *Γ* (20,4) y para Weibull, *W* (32,90) En el caso de la Ruleta, la probabilidad para los más pesados se estableció en 45 %, los medios en 15 % y los más pequeños tienen 40 %. La partición utilizada es exactamente la tercera parte del rango especificado.
    - *I* = 1, *n* ∈ {50}, *m* = 3, W ∈ {50,100,150}, 12 instancias.
    - *I* = 1, *n* ∈ {500}, *m* = 6, ∈ {50,75,100,125,150,175}, 12 instancias.
    - *I* = 1, *n* ∈ {2000}, *m* = 12, *W* ∈ {25,50,75,100,125,150,175,200,225,250,275,300}, 12 instancias.
- Conjunto 3: contiene variaciones en la distribución de las capacidades del contenedor, dentro de las distribuciones Uniforme, Normal y Weibull. Para cada grupo de este conjunto, los parámetros de las distribuciones fueron: *j* ∈ *U* [90,150], *j* ∈ *N* (110,25), *j* ∈ *W* (70,130).
    - I = 1, n ∈ {50}, m = 3, 9 instancias.
    - I = 1, n ∈ {500}, m = 6, 9 instancias.
    - I = 1, n ∈ {2000}, m = 12, 9 instancias.

Se crearon 81 instancias en total, considerando funciones de correlación entre costo y capacidad de los contenedores lineal y cóncava [7,3,6]. La función lineal se modificó adicionándosele un coeficiente (pendiente) y un costo fijo (desplazamiento) a la capacidad, para lo cual se utilizaron números aleatorios, en este caso 15 y 32 respectivamente. También se incluyó otra forma de correlación lineal más simple que se denominará en lo adelante Proporcional:

a. Lineal: $nx + m$
b. Cóncava: $10\sqrt{x}$
c. Proporcional: $0.1x$

De esta forma, se obtuvo un conjunto de instancias que podría ser útil para muchos propósitos relacionados con los estudios de VSCBPP que podrían comenzar a partir de los resultados expuestos en la siguiente sección. El conjunto de datos completo con todos los formatos está disponible en www.cimab.transnet.cu/files/asocfile-2862.zip.

Modelo matemático

Se empleó el modelo matemático definido en [4] siendo: I = {1...,i,...,n} el conjunto de ítems, J = {1...,j,...,m} el conjunto de contenedores, w_i el peso de cada ítem, W_j la capacidad (tamaño) de cada contenedor y C_j el costo de cada contenedor. Para mejorar el rendimiento de los métodos propuestos en [4] y [6], se ordenó descendentemente la lista de elementos $w_1 \geq w_i \geq w_n$ y la lista de tipos de contenedores $W_1 \geq W_j \geq W_m$. La variable binaria x_{ij} indica si el elemento *i* está empaquetado en el contenedor *j* y la variable binaria y_j indica si el contenedor *j* se usa o no.

El VSCBPP se formula empleando la función objetivo (expresión 1) que minimiza el costo de los contenedores utilizados, la restricción (expresión 2) que asegura que cada ítem i esté empaquetado en un solo contenedor y la restricción (expresión 3) que establece que para cada contenedor usado j, la suma de los pesos de los ítems empacados no puede exceder la capacidad del mismo.

$$Min \rightarrow \sum_{j \in J} C_j y_j \tag{1}$$

sujeto a:

$$\sum_{j \in J} x_{ij} = 1 \qquad i \in I \tag{2}$$

$$\sum_{i \in I} w_j x_{ij} \leq W_j y_j \tag{3}$$

Resultados y Discusión

En los resultados que se muestran, además de usarse *n* para indicar el número de ítems y *m* para el número de tipos de contenedores, también se incluye *k*, para especificar el número total de contenedores generados por la herramienta, independientemente de su tipo.

Conjunto 1

En la tabla 1, se establece el número de elementos *n*, el número de tipos de contenedor *m* y el número de contenedores *k*. Hay una columna para cada tipo de función de costo, lineal, cóncava y convexa, y dentro de

cada una de ellas el tiempo utilizado por el *solver* y el *gap*. Como se puede evaluar, hay cuatro instancias donde se halló el óptimo global en poco tiempo, pero, a medida que crece el tamaño de la instancia, el *gap* (diferencia estimada respecto al óptimo) también aumenta, lo que significa que el *solver* no pudo encontrar una solución óptima en una hora. En este sentido, existe una diferencia entre la mejor solución posible y la encontrada.

Tabla 1. Resultados en el conjunto 1 por funciones de costo. Fuente: autores

Características			Lineal		Cóncava		Proporcional	
n	m	k	Tiempo (sec)	Gap (%)	Tiempo (sec)	Gap (%)	Tiempo (sec)	Gap (%)
25	3	28	4,28	0,00	8,69	0,00	48,74	0,00
100	3	118	3600,01	3,92	3600,00	4,26	3600,00	3,91
200	6	517	3600,01	4,84	3600,01	4,84	3600,00	5,39
500	6	1306	3614,90	15,03	3621,94	54,03	3603,51	13,86
1000	6	2576	3624,00	∞	3623,36	∞	3650,17	∞

Los resultados de la tabla 1 confirman que el factor principal que aumenta la dificultad de las instancias (mayor *gap*), es la cantidad de contenedores (*k*), independientemente de su tipo. Nótese que el *gap* reflejado es la relación máxima entre la calidad de la solución del problema entero y su solución aproximada.

Sin embargo, existe un detalle que vale la pena analizar: el efecto de la función de costo. A pesar de que las tres instancias pequeñas (primera fila tabla 1) se resolvieron fácilmente, el tiempo muestra una ligera desviación de la función de costo proporcional con respecto a las demás. Por otro lado, para instancias con 100 o más ítems, el tiempo siempre ronda los 3600 milisegundos, pero el *gap* es mejor, siendo más significativa la diferencia a medida que aumenta el tamaño de la instancia. Esto demuestra que la relación de linealidad afecta la complejidad de las instancias y que el costo fijo en la función lineal condiciona peores resultados en el solver que las instancias cuya relación es proporcional.

Conjunto 2

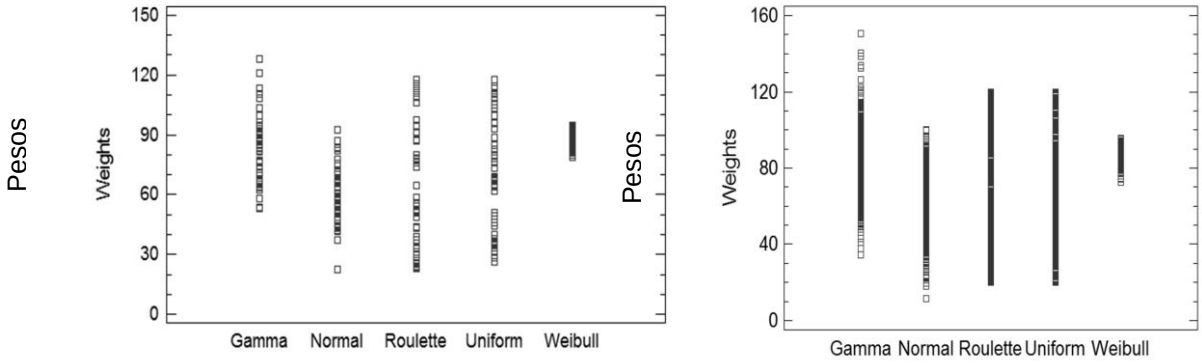
En la tabla 2, se presenta el componente de distribución estadística de los pesos de los ítems (columna IW-d) y el resto de las columnas siguen siendo las mismas.

Tabla 2. Resultados en conjunto 2. Fuente: autores

Características			IW-d	Lineal		Cóncava		Proporcional	
n	m	k		Tiempo (sec)	Gap (%)	Tiempo (sec)	Gap (%)	Tiempo (sec)	Gap (%)
50	3	66	Γ(20,4)	342,20	0,00	198,12	0,00	293,56	0,00
50	3	60	N(60,15)	3600,00	7,84	3600,00	8,56	3600,00	6,16
50	3	28	R(45,15,40)	3600,00	4,23	3600,00	3,16	3600,00	4,20
50	3	77	W(32,90)	1,47	0,00	3,44	0,00	3,22	0,00
50	3	61	U(20,120)	3600,00	2,65	3600,00	3,17	3600,00	4,08
500	6	1396	Γ(20,4)	3613,52	17,09	3642,95	58,00	3627,63	15,86
500	6	1387	N(60,15)	3609,63	22,73	3645,92	84,66	3615,25	20,74
500	6	1263	R(45,15,40)	3622,05	18,22	3608,86	68,54	3604,89	16,61
500	6	1414	W(32,90)	3636,69	15,72	3610,54	50,67	3645,92	14,72

Este conjunto no contiene ninguna instancia fácil, por lo tanto, para medir el efecto de la distribución estadística y la función de costo, solo se utilizó el gap. Con respecto a la correlación entre la instancia de tamaño y el gap, se puede observar que tienen un comportamiento similar a las instancias descritas en el Conjunto 1. Sin embargo, como muestra la figura 1, los pesos para cada distribución son bastante diferentes, o al menos, sus medias aparecen en diferentes puntos. La instancia con Distribución Uniforme y 500 ítems, se tomó del Conjunto 1.

Hay un comportamiento interesante relacionado con la distribución de los pesos de los ítems, demostrado por el valor del gap. En los artículos revisados, la Distribución Uniforme es la única utilizada porque, en teoría, se espera que sea la más difícil de resolver [19]. No obstante, resulta ser la Distribución Normal la que incrementó la complejidad de las instancias resueltas, lo cual puede apreciarse en la figura 2 para el subconjunto que tiene 50 ítems. Para el subconjunto de instancias con 500 ítems este comportamiento es igual, como muestra la tabla 3.



(a) 50 Ítems (b) 500 Ítems

Fig. 1. Dispersión de pesos de ítems por distribuciones. Fuente: autores

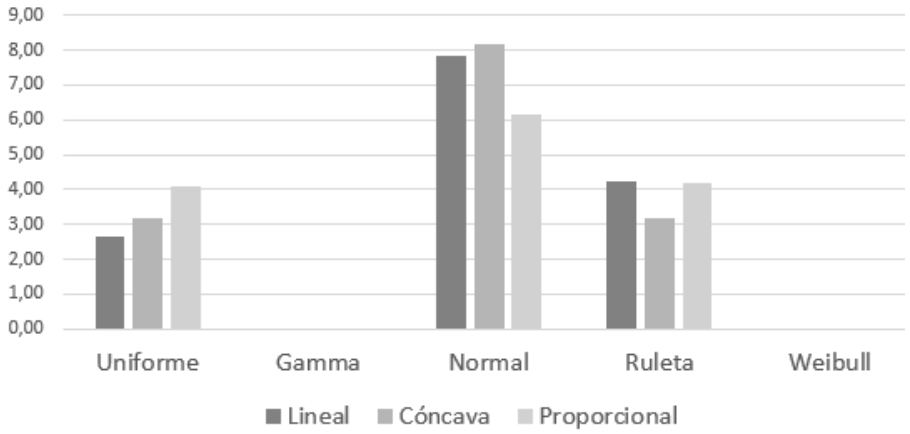


Fig. 2. Gap por función de costo y distribuciones para subconjunto 1 del conjunto 2. Fuente: autores

Como método de comprobación adicional, se realizó un promedio de rango (*rank average*) sobre el gap para cada distribución con cada función de costo. Las tablas 3 y 4 muestran que, efectivamente, la distribución normal con la media en el centro del rango de pesos de los ítems y una desviación estándar relativamente pequeña, aumenta la dificultad de las instancias. Como contraparte, Gamma y Weibull son las de menor complejidad. El estrecho rango de pesos de ítems para Weibull hace que el número de tipos de contenedores que el solver debe considerar sea mayor, es decir, dado que el ítem más pesado puede empacarse en el contenedor más pequeño, en general hay un mayor número de contenedores (considerando todos los tamaños) porque para cada tamaño de contenedor, debe haber suficientes contenedores para empacar todos los ítems. Sin embargo, el número de decisiones a tomar es menor ya que los pesos de los artículos se colocan en un rango muy pequeño y reduce el número de posibles opciones de contenedores. Esto permite pensar que a medida que el rango de pesos tiende a un solo valor, la complejidad disminuirá, siendo el mejor caso del problema.

Tabla 3. Promedio de rangos del gap para instancias de 50 ítems.
Fuente: autores

Función de costo	Uniforme	Gamma	Normal	Ruleta	Weibull
Lineal	3,00	1,50	5	4,00	1,50
Cóncava	4,00	1,50	5	3,00	1,50
Proporcional	3,00	1,50	5	4,00	1,50
Promedio	3,33	1,50	5	3,67	1,50

Tabla 4. Promedio de rangos del *gap* para instancias de 500 ítems
Fuente: autores

Función de costo	Uniforme	Gamma	Normal	Ruleta	Weibull
Lineal	1	3	5	4	2
Cóncava	2	3	5	4	1
Proporcional	1	3	5	4	2
Promedio	1,33	3	5	4	1,67

Hay otro hecho que debe destacarse. Las pequeñas instancias que siguen la Distribución Uniforme en los pesos, son similares en términos de espacio que la Ruleta, pero sucede algo diferente cuando el tamaño de las

instancias crece. En la tabla 4, las instancias dentro de la Distribución Uniforme son las de menor complejidad, y contradice la suposición hecha en otros trabajos [3, 6, 12] de que esta distribución es la más difícil de resolver.

Estos resultados están relacionados con la intersección entre la distribución de pesos y las capacidades disponibles, figura 1. Como se explicó anterirmente, cuando el generador crea los contenedores con el límite superior automático, se considera el número mínimo de contenedores de cada tamaño que se necesitan para almacenar todos los ítems en el conjunto que caben en él. Esto lleva a tener un número diferente de contenedores para cada tipo, lo que afecta la complejidad de la instancia generada.

Para dar una idea precisa de cuándo se eliminan ciertos tipos de contenedores de la instancia, en la tabla 5 (filas donde n = 50) se muestra el porcentaje de ítems del número total en las instancias. El rango de pesos de la columna refleja el subrango de pesos que influyen en el número de contenedores por tipo generado con el límite superior automático, y el resto de las columnas el porcentaje respecto al valor máximo en el rango. Nótese que la distribución de contenedores en los rangos de pesos no coincide estrictamente con los parámetros para la Ruleta descritos en Conjunto de datos. En este sentido es necesario señalar que esta opción funciona como una rueda, lo que significa que un valor aleatorio determina la posición del peso en uno de los tres percentiles definidos. Las filas donde n = 500 sigue la misma idea, pero para las instancias del subconjunto dos, el que tiene 500 elementos y seis tipos de contenedor.

Tabla 5. Porciento de ítems en rangos por distribución (según n y m). Fuente: autores

n	m	Rango de pesos	Uniforme (%)	Gamma (%)	Normal (%)	Ruleta (%)	Weibull (%)
50	3	[0,50[	36	0	20	42	0
50	3	[51,100[	44	88	80	42	100
50	3	[101,150]	20	12	0	16	0
500	6	[25,50[	28	2	26	38	0
500	6	[50,75[	29	40	57	27	1
500	6	[75,100[	22	45	17	19	99
500	6	[100,125[	21	12	0	16	0
500	6	[125,150[	0	1	0	0	0
500	6	[150,175]	0	0	0	0	0

De acuerdo con los resultados expuestos se puede entonces asegurar que, las instancias más difíciles de resolver son aquellas donde los pesos siguen una Distribución Normal y tienen la mayor cantidad de contenedores (sin importar el tipo) disponibles para empaquetar.

Conjunto 3

Siguiendo la misma línea, la Tabla 6 muestra los resultados del Conjunto 3 donde se utilizaron diferentes distribuciones estadísticas para las capacidades de los contenedores. La columna *k* representa el número de ubicaciones disponibles (considerando todos los tipos) y BC-d es la distribución utilizada para los tipos de contenedores; el resto sigue igual y las instancias más grandes siguen siendo un desafío.

Para lograr una mejor comprensión del análisis para este conjunto, en la tabla 7 se muestran los tipos de contenedores generados para un número fijo de ítems *n*: para *n* = 50 hay 3 tipos de contenedor, mientras que para *n* = 500 hay 6 tipos. Estos valores cambian de una distribución a otra ya que el generador intenta cubrir automáticamente todo el rango de pesos que crea la demanda de contenedores disponibles.

Tabla 6. Resultados en conjunto 3 para funciones de costo Lineal y Cóncava.Fuente:autores

Características			BC-d	Lineal		Cóncava		Proporcional	
n	m	k		Tiempo (sec)	Gap (%)	Tiempo (sec)	Gap (%)	Tiempo (sec)	Gap (%)
50	3	85	U(90,150)	3600,00	9,65	3600,00	0,50	3600,00	6,95
50	3	77	N(110,25)	3600,00	4,93	3600,00	5,50	3600,00	8,09
50	3	93	W(70,130)	3600,00	10,16	3600,00	11,88	3600,00	5,93
500	6	1535	U(90,150)	3634,79	11,16	3602,07	36,56	3628,13	22,40
500	6	1350	N(110,25)	3613,55	18,63	3635,07	33,83	3657,63	23,09
500	6	1767	W(70,130)	3675,00	32,03	3644,45	35,24	3661,40	32,65

Tabla 7. Contenedores generados en el conjunto 3 para 50 ítems y de 500 ítems. Fuente: autores

n	Uniforme	Normal	Weibull
50	136,56	91,10	124,61
50	140,10	109,90	127,10
50	142,35	132,34	131,81
500	94,74	61,85	123,61
500	97,06	64,95	126,99
500	99,62	102,46	129,23
500	106,73	109,52	131,07
500	135,06	111,64	131,14
500	142,03	149,10	132,53

En estos resultados se puede observar que cuando la capacidad (tipos) de contenedores siguen una Distribución Weibull, el *solver* obtiene el peor *gap*, lo que está relacionado, con el tamaño de la instancia (un mayor número de contenedores disponibles) como muestra la columna *k* y la distribución de pesos en relación con los tamaños de los contenedores. Esto significa que el tipo de contenedor más pequeño es mayor que el artículo más pesado y se necesita para cada tamaño de contenedor, un mayor número de contenedores para empacar todos los artículos.

La tabla 8 muestra el valor promedio del gap en las instancias con 50 y con 500 ítems para el conjunto 3. El *gap* de la función de costo Proporcional con Distribución Uniforme en las instancias más grandes es peor que para las más pequeñas, incluso superando el *gap* de las instancias donde la capacidad sigue la Distribución de Weibull.

Tabla 8. Promedio de rangos del *gap* para instancias con 50 y 500 ítems en el conjunto 3. Fuente: autores

Función de costo	**n = 50**			**n = 500**		
	Uniforme	**Normal**	**Weibull**	**Uniforme**	**Normal**	**Weibull**
Lineal	2	1	3	1	2	3
Cóncava	1	2	2	3	1	1
Convexa	2	3	3	1	2	2
Promedio	1,67	2	2,67	1,67	1,67	2

Se realizaron varios experimentos para verificar este problema en particular, pero hasta ahora, no se encontró ningún patrón que lo explicara. Sin embargo, en la tabla 8 está comprobado que en casos generales en los que las capacidades del contenedor siguen la Distribución de Weibull, el *solver* alcanza el límite de tiempo y da un *gap* más alto a pesar del tamaño de las instancias.

Considerando los resultados expuestos, cuando las capacidades de los contenedores siguen una Distribución Weibull, las instancias se complejizan proporcionalmente al tamaño. Si a esto se le adiciona el resultado del análisis del Conjunto 2, pudieran obtenerse instancias mucho más complejas cuando los pesos de los ítems sigan una Distribución Normal y las capacidades de los contenedores la Distribución Weibull.

Conclusiones

El conocimiento, en cuanto a complejidad de las instancias del VSCBPP, resulta vital para seleccionar la herramienta o método de solución. En el presente trabajo, se utilizó un *solver* que no logró resolver las instancias más grandes, pero a los efectos de la investigación obtuvo los resultados necesarios para probar la influencia de diferentes distribuciones estadísticas en la complejidad de las instancias analizadas.

Los resultados exhibidos son concluyentes en cuanto a que demuestran que las instancias de mayor complejidad son aquellas donde los pesos de los ítems siguen la Distribución Normal. Lo mismo ocurre cuando las capacidades de los contenedores siguen la Distribución Weibull. De esta forma se demuestra que la distribución estadística de dos de los tres atributos principales que caracterizan el problema (peso del artículo, capacidad del contenedor, costo del contenedor) afectan la complejidad del problema en términos de eficiencia y eficacia.

Por otro lado, también pudo observarse que la función de costo es otro factor que incluye en la dificultad del problema independientemente del tamaño de las instancias y la distribución estadística que sigan los tipos de contenedores o los pesos de los ítems. La causa de este hecho es desconocida para los autores, por lo tanto,

es un asunto que se considera debe ser abordado por investigaciones adicionales. Otro aspecto que podría considerarse en estudios futuros es la variación de los parámetros para las distribuciones estadísticas que también podría tener un impacto en la complejidad.

Referencias

1. Christensen H, Khan A, Pokutta S, et al. Approximation and online algorithms for multidimensional bin packing: A survey. Computer Science Review. 2017;24:63-79. DOI 10.1016/j.cosrev.2016.12.001.
2. Balogh J, Békési J, Dósa G, et al. Online bin packing with cardinality constraints resolved. Journal of Computer and System Sciences. 2020;112:34-49. DOI 10.1016/j.jcss.2020.03.002.
3. Crainic TG, Perboli G,. Rei W, et al. Efficient lower bounds and heuristics for the variable cost and size bin packing problem. Computers & Operations Research. 2011;38(11):1474-1482. DOI https://doi.org/10.1016/j.cor.2011.01.001.
4. Hemmelmayr V, Schmida V, Blum C. Variable neighbourhood search for the variable sized bin packing problem. Computers & Operations Research. 2012;39(5):1097-1108. DOI https://doi.org/10.1016/j.cor.2011.07.003
5. Fraire Huacuja HJ, Padilla AE, Cruz-Reyes L, et al. Variable Neighborhood Search Algorithm for the Variable Cost and Size Bin Packing Problem. In: Handbook of Research on Metaheuristics for Order Picking Optimization in Warehouses to Smart Cities. Hershey PA: IGI Global; 2019. DOI 10.4018/978-1-5225-8131-4.
6. Haouari M,Serairi M. Heuristics for the variable sized bin-packing problem. Computers & Operations Research. 2009;36(10):2877-2884. https://doi.org/10.1016/j.cor.2008.12.016
7. Correia I, Gouveia L, Da Gama FS. Solving the variable size bin packing problem with discretized formulations. Computers & Operations Research. 2008;35(6):2103-2113. https://doi.org/10.1016/j.cor.2006.10.014
8. Castelli M, Vanneschi LA hybrid harmony search algorithm with variable neighbourhood search for the bin-packing problem. In: 6th World Congress on Nature and Biologically Inspired Computing (NaBIC). Porto, Portugal: IEEE; 2014.
9. Brandão F, Pedroso JP. Bin packing and related problems: General arc-flow formulation with graph compression. Computers & Operations Research. 2016;69:56-67. https://doi.org/10.1016/j.cor.2015.11.009
10. Sridhar R, Chandrasekaran M, Sriramya C, et al. Optimization of heterogeneous Bin packing using adaptive genetic algorithm. In: IOP Conference Series: Materials Science and Engineering. IOP Publishing; 2017.
11. Fraire Huacuja HJ, Estrada Padilla A, Cruz Reyes L, et al. Variable neighborhood search algorithm for the variable cost and size bin packing problem In: Handbook of Research on Metaheuristics for Order Picking Optimization in Warehouses to Smart Cities. IGI Global; 2019. DOI 10.4018/978-1-5225-8131-4.
12. Bang-Jensen J, Larsen R. Efficient algorithms for real-life instances of the variable size bin packing problem. Computers & Operations Research. 2012;39(11):2848-2857. https://doi.org/10.1016/j.cor.2012.02.018
13. Beasley JE. OR-Library. London, UK.: Brunel Unversity; 2009 Disponible en: http://people.brunel.ac.uk/~mastjjb/jeb/info.html
14. Casazza M, Ceselli A. Column generation for the variable cost and size bin packing problem with fragmentation. Electronic Notes in Discrete Mathematics. 2016;55:61-64.. https://doi.org/10.1016/j.endm.2016.10.016
15. Casazza M. New formulations for Variable Cost and Size Bin Packing Problems with Item Fragmentation. Optimization Letters. 2019;13(2):379–398. DOI 10.1007/s11590-018-1327-x.
16. Herrera Franklin J, Rosete A, García Borroto M. et al. On the impact of fuzzy constraints in the variable size and cost bin packing problem In: Information Processing and Management of Uncertainty in Knowledge-Based Systems Springer International Publishing Cham; 2020 DOI 10.1007/978-3-030-50146-4_18.
17. Schwarz C. An introduction to SCIP. In: SCIP Workshop 2018. Germany: Zuse Institute Berlin; 2018.
18. Koch T. Zimpl user guide (zuse institute mathematical programming language). Berlin, Germany: Technische Universität Berlin; 2004. Disponible en: http://www.zib.de/Publications/abstracts/ZR-04-58
19. Jin Y, Wang H, Chugh T. et al. Data-Driven Eutionary Optimization: An Overview and Case Studies. IEEE Transactions on Eutionary Computation. 2019;23(3):442-458. DOI 10.1109/TEVC.2018.2869001.

Conflicto de intereses

Los autores declaran que no existen conflictos de intereses

Contribución de los autores

Jorge Herrera Franklin, https://orcid.org/0000-0002-3694-499X

Participó en la preparación y en el diseño de los experimentos. Realizó la programación de los algoritmos utilizados en los experimentos. Realizó contribuciones en el análisis e interpretación de los datos. Participó en la búsqueda de información, en el diseño de la investigación, en la recolección de los datos, análisis de los resultados y en la revisión crítica de su contenido así como en la redacción y aprobación del informe final.

Alejandro Rosete Suárez. https://orcid.org/0000-0002-4579-3556

Participó en el diseño de los experimentos. Realizó contribuciones en el análisis e interpretación de los datos. Participó en la búsqueda de información, análisis de los resultados y en la revisión crítica de su contenido así como en la redacción, revisión y aprobación del informe final.

Milton García Borroto. https://orcid.org/0000-0002-3154-177X

Participó en el diseño de los experimentos. Realizó contribuciones en el análisis e interpretación de los datos. Participó en la búsqueda de información, análisis de los resultados y en la revisión crítica de su contenido así como en la revisión y aprobación del informe final

Suitberto Cabrera García https://orcid.org/0000-0003-1704-8361

Realizó contribuciones en el análisis e interpretación de los datos. Participó en la búsqueda de información, análisis de los resultados y en la revisión crítica de su contenido así como en la revisión y aprobación del informe final.