***Ingeniería Investigación y Tecnología,*** volumen XVII (número 2), abril-junio 2016: 155-168
ISSN 1405-7743 FI-UNAM
(artículo arbitrado)
doi:10.1016/j.riit.2016.06.001

# Una nueva estrategia heurística para el problema de *Bin Packing*

## *A New Heuristic Strategy for the Bin Packing Problem*

Pérez-Ortega Joaquín
*Centro Nacional de Investigación y Desarrollo Tecnológico*
*Departamento de Ciencias Computacionales*
*Correo: jpo_cenidet@yahoo.com.mx*

Castillo-Zacatelco Hilda
*Benemérita Universidad Autónoma de Puebla*
*Centro Nacional de Investigación y Desarrollo Tecnológico*
*Departamento de Ciencias Computacionales*
*Correo: hildacz@gmail.com*

Vilariño-Ayala Darnes
*Benemérita Universidad Autónoma de Puebla*
*Facultad de Ciencias de la Computación*
*Correo: darnesv@yahoo.com*

Mexicano-Santoyo Adriana
*Instituto Tecnológico de Cd. Victoria*
*Correo: mexicanoa@gmail.com*

Zavala-Díaz José Crispín
*Universidad Autónoma del Estado de Morelos*
*Facultad de Contaduría, Administración e Informática*
*Correo: crispin_zavala@uaem.mx*

Martínez-Rebollar Alicia
*Centro Nacional de Investigación y Desarrollo Tecnológico*
*Departamento de Ciencias Computacionales*
*Correo: dcc@cenidet.edu.mx*

Estrada-Esquivel Hugo
*Fondo de Información y Documentación para la Industria*
*Correo: hestrada@dsic.upv.es*

Información del artículo: recibido: enero de 2015, reevaluado: mayo de 2015, aceptado: septiembre de 2015

**Resumen**

El problema de *Bin Packing* (BPP) es NP-duro, por lo que un método exacto para resolver instancias del BPP requiere un gran número de variables y demasiado tiempo de ejecución. En este trabajo se propone una nueva estrategia heurística para resolver instancias del BPP en donde se garantiza la solución óptima. La estrategia propuesta incluye el uso de un nuevo modelo exacto basado en arcos de flujo. En el modelo propuesto, el número de variables se redujo asignando objetos en contenedores. Adicionalmente se incluye una heurística que mediante el preprocesado de la instancia permite reducir su tamaño y con ello el espacio de búsqueda del algoritmo de solución. Para validar el enfoque propuesto, se realizaron experimentos usando los conjuntos de prueba hard28, 53nirup, bin1data, uniform, triplets y subconjuntos de otras instancias, todos ellos conocidos en el estado del arte. Los resultados muestran que empleando nuestro enfoque es posible encontrar la solución óptima de todas las instancias de prueba. Además, el tiempo de ejecución se redujo en relación con lo reportado por el modelo basado en arcos de flujo. Las reducciones de tiempo fueron de 19.7 y 43% para los conjuntos 53nirup y hard28, respectivamente.

**Descriptores:**

- heurística
- *Bin Packing*
- arcos de flujo
- modelo exacto para BPP
- límite inferior, instancias del
- BPP

***Abstract***

*The Bin Packing problem (BPP) is NP-hard, the use of exact methods for solving BPP instances require a high number of variables and therefore a high computational cost. In this paper a new heuristic strategy for solving the BPP instances, which guarantees obtain optimal solutions, is proposed. The proposed strategy includes the use of a new model based on flow arcs. In the proposed model, the number of variables was reduced by previous allocation of objects in bins. Additionally, our approach includes a heuristic that allows reducing the instance size and thereby the solution algorithm search space. To validate the proposed approach, experiments were performed using the test sets hard28, 53nirup, bin1data and falkenauer, all of them well known in the state of the art. The results show that it using our approach is possible to find the optimal solution for all test set. Also, the execution time was reduced in regard the reported time obtained by using the flow arc model. Time reductions were up to 19.7 and 43% for 53nirup and hard28 test set, respectively.*

***Keywords:***

*heuristic*
*bin packing*
*flow arcs*
*exact model for BPP*
*lower bound*
*instances of the BPP*

## Introducción

El problema de *Bin Packing* se puede formular de la siguiente manera: dados $n$ objetos de tamaño $w_1$, .., $w_n$ y contenedores de tamaño $C$, el objetivo es encontrar el menor número de contenedores en donde se coloquen todos los objetos. De acuerdo con Garey y Johnson (1979) el problema de *Bin Packing* (BPP) es un problema combinatorio, que pertenece a la clase de problemas NP-duros y existen problemas reales de la industria que pueden mapearse a dicho problema.

Algunos investigadores dirigen sus esfuerzos hacia el desarrollo de modelos matemáticos que representen el problema, con el objetivo de obtener soluciones óptimas, por ejemplo, los modelos desarrollados por Valerio (1999, 2000) y Martello y Toth (1990). El trabajo desarrollado por Shawn (2004) es otra alternativa que propone darle solución al problema BPP realizando una modificación al modelo desarrollado para resolver el problema de la mochila multidimensional. La limitante de esta propuesta es que no resuelve convenientemente todas las instancias de los conjunto uniform y triplets (Falkenahuer, 1996). Una mejora a este trabajo se propuso por Schaus en su tesis doctoral (2009), donde mediante el uso de arcos de flujo se resuelven todas las instancias de los conjuntos uniform y triplets, además se mejoran los tiempos de cómputo presentados por Shawn.

Cambazard y O'Sullivan (2010) propusieron introducir directamente en el modelo de arcos de flujo los criterios utilizados por Shawn, lo que les permitió reducir los tiempos de ejecución y resolver las instancias uniform y bin1data (Scholl *et al*., 1997), sin embargo, no lograron resolver de forma óptima todas las instancias de los conjuntos triplets y bin2data (Scholl *et al*., 1997). Alves (2012) por su parte, propuso un modelo basado en arcos de flujo, que limita el número de objetos por contenedor en las restricciones, lo cual provoca que tanto el número de variables como el tiempo de ejecución se reduzcan. Sin embargo, no garantiza que se llegue al óptimo para todas las instancias.

Los trabajos anteriores, en su mayoría, se basan en los modelos de Valerio, ya que dichos modelos utilizan la técnica de optimización mediante arcos de flujo (Taha, 2004), que permite representar todas las posibles formas de acomodar los objetos en los contenedores. Para construir el modelo de arcos de flujo se utiliza una representación del problema mediante un grafo, donde es necesario garantizar la ley de conservación del flujo. Sin embargo, la limitante general de estos algoritmos es la gran cantidad de variables utilizadas y el tiempo empleado para encontrar la solución óptima del problema, por lo que es importante diseñar nuevos modelos cuya formulación permitan generar algoritmos que consuman menos tiempo de cálculo, pero que permitan encontrar la solución óptima.

En este sentido, el presente trabajo propone un nuevo modelo para el problema BPP de una dimensión, en donde todos los contenedores deben tener la misma capacidad. El método propuesto se basa en los modelos formulados por Valerio (1999) para el BPP de una dimensión y de tamaño variable (Valerio, 2002). Cada arco en el grafo corresponde a una variable en el modelo. Para reducir la dimensión del grafo se fijan las variables, donde el peso es mayor a la mitad de la capacidad del contenedor. Además, se introducen restricciones considerando los límites propuestos por Martello y Toth (1990), Scholl *et al*. (1997) y Jarboui *et al*. (2012). Para encontrar el límite inferior de cada instancia de prueba, se propone una nueva metaheurística basada en las características de las instancias. Otra característica de la estrategia propuesta es la incorporación de dos

criterios de reducción de la instancia, inspirados en el trabajo de Alvim *et al*. (2004), que permiten fijar objetos a contenedores, provocando que el número de objetos de la instancia se reduzca, sin que por ello se pierda la optimización del resultado. Esto también provoca que se reduzca considerablemente el número de variables en el modelo. Todos los criterios comentados aceleran el proceso de generación de arcos en el modelo propuesto.

Los resultados muestran que utilizando los criterios de reducción propuestos y el modelo desarrollado, es posible encontrar las soluciones óptimas para los conjuntos de datos hard28, 53nirup (Schoenfields, 2002) y bin1data. Es importante mencionar que no todas las instancias cuentan con las características adecuadas para aplicar los criterios propuestos en este trabajo. En estas circunstancias se encuentra el conjunto triplets, sin embargo, con el modelo propuesto se encontraron todas las soluciones óptimas de las instancias de este conjunto. También se experimentó con el conjunto uniform y una muestra de los conjuntos bin2data, schwerin y dualdistribution. Se contrastaron los tiempos utilizados por el modelo propuesto por Valerio y el modelo desarrollado, y se observó que este último empleó 43% menos de tiempo que el modelo de Valerio para el conjunto hard28, y para el conjunto 53nirup el modelo propuesto utilizó 19.7% menos tiempo que el modelo de Valerio. El modelo de programación entera generado se resuelve utilizando LINGO (2008).

Lo que resta del documento se estructura de la siguiente forma: en la segunda sección se presentan dos modelos matemáticos que sirven como base del modelo propuesto para resolver el problema BPP. En la sección 3 se describe el modelo matemático, en la sección 4 se presenta una metaheurística para calcular el límite inferior, en la sección 5 se desarrolla una heurística de preprocesamiento que permite reducir el tamaño de la instancia. En la sección 6 se discute la implementación del sistema de pruebas, en la sección 7 se presentan los resultados experimentales y finalmente se discuten las conclusiones y el trabajo a futuro.

## Modelos matemáticos de Valerio J.M. y Martello y Toth para el BPP

De acuerdo con la literatura especializada, los modelos matemáticos más relevantes que se han desarrollado para dar solución al problema de *Bin Packing* son los de Martello y Toth (1990) y el de Valerio (1999, 2002).

El modelo de Martello y Toth, se puede clasificar como un modelo de programación binaria mixta, que involucra un gran número de variables binarias, por lo que es costoso computacionalmente encontrar la solución óptima (Balas, 1979).

La descripción del modelo de Martello y Toth (1990) se presenta a continuación

Dados $W=\{w_1, w_2, \ldots, w_n\}$,

que es el conjunto de pesos de los $n$ objetos de una instancia y también el número de contenedores, se deben asignar objetos a contenedores, de tal forma que la suma de los pesos de los objetos no exceda la capacidad del contenedor y el número de contenedores utilizados sea el mínimo. La capacidad de cualquier contenedor es $C$ y estos se representan en el modelo por $y_i$, que es una variable binaria, así si $y_i$ es igual a 0 significa que el contenedor $i$ no se utilizó. La formulación del modelo es la siguiente

Minimizar $z_1 = \sum_{i=1}^{n} y_i$

Sujeto a:

$$\sum_{j=1}^{n} w_j x_{ij} - C y_i \leq 0, \text{ para } i = \{1,\ldots,n\} \quad (1)$$

$$\sum_{i=1}^{n} w_{ij} = 1, \text{ para } j = \{1,\ldots,n\} \quad (2)$$

$y_i = 0$ ó $1$

$x_{ij} = 0$ ó $1$

donde

$$y_1 = \begin{cases} 1 \text{ si el contenedor } i \text{ se usa} \\ 0 \text{ en caso contrario} \end{cases}$$

$$x_{ij} = \begin{cases} 1 \text{ si el objeto } j \text{ se asigna al contenedor } i \\ 0 \text{ en caso contrario} \end{cases}$$

$w_j \in Z^+$, es el peso del $j$ - ésimo objeto

Las restricciones numeradas como (1) garantizan que los objetos que se ubican en un contenedor no excedan la capacidad del mismo, las restricciones numeradas como (2) garantizan que cada objeto se asigne una sola vez.

Un algoritmo que resuelve de manera óptima un problema que involucra variables binarias es el méto-

do de Balas (1979), el cual tiene un costo temporal exponencial. Este algoritmo introduce cuatro criterios de eliminación, considerando la forma en que construye la familia de problemas que genera el proceso en que se fija cada variable en cero o 1, y el valor del óptimo que se obtiene durante el proceso de ramificación.

Por otro lado, Valerio (1999, 2002) propuso un modelo basado en programación lineal entera que utiliza la técnica de arcos de flujo (Taha, 2004) para modelar el problema de *Bin Packing* de tamaño variable y el de una sola dimensión. A continuación se describe brevemente el modelo para el BPP de una dimensión, donde todos los contenedores tienen la misma capacidad.

En este modelo los diferentes pesos de los objetos de una instancia se agrupan en un conjunto $P = \{p_1, p_2, \ldots, p_m\}$, donde $m$ representa el número de valores distintos de los pesos de los objetos en una instancia y $p_1 > p_2 > \ldots > p_m$. Además, se define el conjunto $B = \{b_1, b_2, \ldots, b_m\}$, donde $b_i$ representa la demanda (número de objetos) asociada al peso $p_i$.

Este modelo permite minimizar el valor del flujo representado por $z_1$, que circula por todo el grafo. Cada arco en el grafo corresponde a una variable $x_{ij}$ , que representa la demanda del objeto con peso *j-i* en la solución final. En este contexto los nodos tienen un identificador que es el valor de la variable $d$ (del vértice del grafo), para crear el arco $(d, d+w_i)$, se suma el valor de $d$ al valor de $w_i$ y ese nuevo valor es el identificador del vértice destino del arco; algo similar ocurre con el nombre de las variables, por ejemplo en $x_{d,\, d+wi}$.

El modelo matemático basado en arcos de flujo propuesto por Valerio (1999) se formula a continuación

Minimizar $z$

Sujeto a:

$$\sum_{(d,e)\in A'} x_{de} - \sum_{(e,f)\in A'} x_{ef} = \begin{cases} -z, \text{ si } e = 0 \\ 0, \text{ para } e = 1, \ldots, C-1 \\ z, \text{ si } e = C \end{cases} \qquad (3)$$

$$\sum_{(d,d+w_i)\in A'} x_{d,d++w_i} \geq b_i, i = 1, \ldots, m \qquad (4)$$

$$x_{de} \in Z^+ \cup \{0\}, \forall (d,e) \in A'$$

donde

$x_{ij,} \in A'$ = al arco $(i, j)$ del grafo

$A' \subseteq A$, donde $A$ son los arcos que forman el grafo del modelo antes de aplicar los criterios de reducción 1, 2 y 3 definidos en Valerio (1999)

$b_i$ = demanda del $i$-ésimo objeto

C = capacidad del contenedor

$m$ = número de elementos de $P$

Las restricciones numeradas como (3) son las que garantizan la conservación del flujo sobre la red, es decir, para cada vértice el flujo que entra es exactamente igual al flujo que sale. El valor del flujo que circula sobre todo el grafo es $z$. Las restricciones numeradas como (4) son las que garantizan no asignar más objetos que la demanda dada para cada uno de ellos.

## Modelo propuesto para el BPP

En esta sección se presenta una de las principales contribuciones de esta investigación. El modelo propuesto en este trabajo se cimenta en los modelos formulados por Valerio para el BPP de una dimensión (1999) y de tamaño variable (2002). Cada arco en el grafo corresponde a una variable del modelo. En el modelo se fijan las variables cuyo peso es mayor a la mitad de la capacidad del contenedor, logrando con ello reducir la dimensión del problema. Esto se realizó porque para muchas instancias del *benchmark* para el BPP, se observó que el peso de los objetos es mayor a la mitad de la capacidad del contenedor, por lo que cada objeto que cumple con esta condición debe ubicarse en un contenedor donde no pueda ubicarse otro objeto con la misma característica, lo que provoca que el valor de las variables asociadas puedan tomar un valor constante.

Cada variable del modelo se asocia a un arco del grafo y el valor que toma representa el número de veces que un arco aparece en la solución, es decir, representa la demanda del objeto. Por lo que en el modelo propuesto, las variables asociadas a arcos donde el peso sobrepasa la mitad de la capacidad del contenedor se sustituyen por la demanda respectiva.

En este modelo se introducen restricciones cuyo término independiente se corresponde con el límite inferior del problema. Para encontrar este límite inferior se desarrolla una nueva metaheurística, la cual se discute en la siguiente sección. La formulación del modelo propuesto es la siguiente

Minimizar z

sujeto a:

$$\sum_{(d,e)\in A',e-d\le C/2} x_{de} + \sum_{(d,e)\in A',e-d>C/2,w_t=e-d} b_t - \sum_{(d,e)\in A',e-d\le C/2} x_{de} - \sum_{(d,e)\in A',e-d>C/2,w_t=e-d} b_t = \begin{cases} -z, \text{ si } e=0 \\ 0, \text{ para } e=1,\cdots,C-1 \\ z, \text{ si } e=C \end{cases} \quad (5)$$

$$\sum_{(d,d+w_t)\in A',d+w_t-d\le C/2} x_{d,d+w_t} + \sum_{(d,d+w_t)\in A',d+w_t-d>C/2} b_t \ge b_i, \qquad i=1,\cdots,m; \quad t=1,\cdots,m \quad (6)$$

$$z \le lowerBound,$$

$$x_{de} \in Z^+ \cup \{0\}, \forall (d,e) \in A' \quad (7)$$

El objetivo principal del modelo es minimizar el flujo en el grafo generado a partir de los pesos de los objetos y la capacidad del contenedor. El grafo se construye de la misma forma que en los modelos de Valerio (1999, 2002), aplicando los criterios de reducción 1, 2 y 3, descritos en ese trabajo. Las restricciones (5) corresponden a la conservación del flujo en la red. En este modelo se propone además, que las variables que corresponden a objetos con pesos mayores a la mitad de la capacidad del contenedor, tomen el valor de la demanda asociada. Las restricciones (6), garantizan que no se coloquen objetos que excedan la capacidad del contenedor. Las restricciones (7) permiten limitar el número de contenedores utilizados, lo que reduce el espacio de búsqueda.

Las variables del modelo se defininen de la siguiente forma:

$C$ = capacidad de cualquier contenedor
$A'$ = conjunto de arcos asociados al grafo después de aplicar los criterios de reducción propuestos por Valerio
*lowerBound* = límite inferior de la instancia, el cálculo de este valor se describe más adelante
$x_{de}$ = variable que representa al arco $(d,e)$
$b_t$ = la demanda asociada al $t$-ésimo peso

La restricción de acotación de la variable $z$ con el límite inferior se creó a partir del modelo de Valerio para BPP de tamaño variable, pero con la diferencia de que en el modelo propuesto $z$ se restringe por una constante y no por una variable (Valerio, 2002). Mediante esta restricción se busca reducir el espacio de soluciones y garantizar la solución óptima de la instancia. Si por alguna razón el óptimo no coincide con el límite inferior, es decir, no se obtiene una solución factible, se modifica el valor del límite inferior agregando un contenedor.

El límite inferior *lowerBound* para una instancia del BPP representa el número de contenedores que al menos se deben utilizar para acomodar todos los objetos de esa instancia.

A continuación, se muestra un ejemplo de la aplicación del modelo desarrollado para solucionar una instancia del BPP.

Sea $W$={6, 5, 4, 3, 2, 1} el conjunto de pesos de los objetos, $b$={2, 2, 1, 2, 4, 3} la demanda correspondiente a cada peso ($w_i$ tiene una demanda de $b_i$), y $C$=7 la capacidad del contenedor. En total son 14 objetos.

Aplicando la técnica de arcos de flujo y los criterios de reducción propuestos en Valerio (1999) se obtiene el siguiente conjunto de arcos, *arcos*={(0,1), (0,2), (0,3), (0,4), (0,5), (0,6), (1,2), (2,3), (2,4), (3,4), (3,5), (3,6), (4,5), (4,6), (4,7), (5,6), (5,7), (6,7)}. Para esta instancia se calcula el límite inferior con la metaheurística que se describe en la siguiente sección. El límite inferior calculado es de 7 (*LowerBound*).

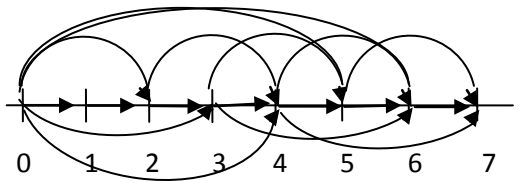


Figura 1. Grafo asociado al modelo de PLE ejemplo

En la figura 1 se presenta el grafo que corresponde al conjunto de arcos del ejemplo. Para este ejemplo se asocian variables $x_{de}$, a cada uno de los arcos. Las variables asociadas a arcos donde el peso es mayor a la mitad de la capacidad del contenedor son: $x_{04}$, $x_{05}$ y $x_{06}$. Por lo que estas variables toman el valor de la demanda correspondiente, es decir, la variable $x_{04}$ toma el valor 1, la variable $x_{05}$ toma el valor 2 y la variable $x_{06}$ también toma el valor 2.

El modelo de programación lineal entera (PLE) se formula para este ejemplo de la siguiente forma

Minimizar $z$
Sujeto a:

$$z - x_{01} - x_{02} - x_{03} - 1 - 2 - 2 = 0$$
$$x_{01} - x_{12} = 0$$
$$x_{02} + x_{12} - x_{23} - x_{24} = 0$$
$$x_{03} + x_{23} - x_{34} - x_{35} - x_{36} = 0$$
$$1 + x_{24} + x_{34} - x_{45} - x_{46} - x_{47} = 0$$
$$2 + x_{35} + x_{45} - x_{56} - x_{57} = 0$$
$$2 + x_{36} + x_{46} + x_{56} - x_{67} = 0$$
$$x_{47} + x_{57} + x_{67} - z = 0$$
$$x_{03} + x_{36} + x_{47} \geq 2$$
$$x_{02} + x_{24} + x_{35} + x_{46} + x_{57} \geq 4$$
$$x_{01} + x_{12} + x_{23} + x_{34} + x_{45} + x_{56} + x_{67} \geq 3$$
$$z \leq 7$$

El óptimo se alcanza para $z$=7 y el valor de la solución obtenida es: $x_{01}$=1, $x_{02}$=1, $x_{12}$=1, $x_{23}$=1, $x_{24}$=1, $x_{34}$=1, $x_{45}$=1, $x_{56}$=1, $x_{57}$=2, $x_{67}$=3. Se agregan a la solución las variables $x_{04}$, $x_{05}$ y $x_{06}$, que tomaron el valor de la demanda, de acuerdo con el criterio aplicado.

En la figura 2 se presenta una forma en donde se pueden acomodar los objetos en los contenedores.

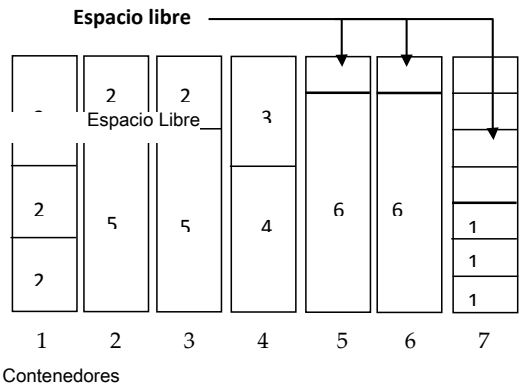


Figura 2. Ubicación de los objetos en los contenedores de acuerdo con la solución óptima encontrada

## Metaheurística para calcular el límite inferior

El límite inferior de una instancia del BPP, indica el número de contenedores que al menos deberán utilizarse para colocar los *n* objetos de la instancia. Si el límite inferior coincide con el número de contenedores óptimo que una instancia debe utilizar, sería de mucha utilidad como en algunos casos que a continuación se describen. Una solución obtenida del BPP al aplicar una heurística se puede decir que es la óptima si coincide con el valor de alguno de los límites inferiores.

Para el modelo propuesto en este trabajo, el uso del límite inferior permite reducir el espacio de soluciones. Por lo que conseguir un límite inferior lo más cercano al óptimo es importante.

Los investigadores han propuesto varios métodos para calcular el límite inferior de una instancia, algunos de ellos se aproximan más al número de contenedores de la solución óptima que otros.

En Pérez *et al.* (2014) se propone una metaheurística que permite calcular el límite inferior de una instancia de tal manera que sea lo más aproximado posible al óptimo. La metaheurística hace uso de varios métodos para calcular el límite inferior y selecciona al más adecuado dependiendo de las características de la instancia.

En ese trabajo se realizó una evaluación de los límites propuestos por Martello y Toth (1990) (LB1, LB2 y LB3), los propuestos por Scholl *et al.* (1997) (LB4), y el límite propuesto por Jarboui *et al.* (2010) (LBJ). El límite LB3 para muchas instancias coincide con el valor óptimo, sin embargo la limitante de esta cota es su tiempo de respuesta para algunas instancias, por lo que se hicieron experimentos para averiguar qué tipo de instancias presentaban este comportamiento. Con este fin se evaluaron las métricas, para caracterizar instancias, propuestas por Cruz (2004) y Álvarez (2006). Las métricas que finalmente caracterizaron de manera adecuada a cada instancia estudiada fueron la Kurtosis y MaxRepe. A partir de los resultados obtenidos, se desarrolló una metaheurística para el cálculo del límite inferior, el pseudocódigo de esta se muestra en el Algoritmo 1, publicado en Pérez *et al.* (2014). La descripción de las variables del algoritmo se muestra en la tabla 1.

En esta metaheurística además de la Kurtosis y MaxRepe se introducen dos métricas más: porcItem1 y porcMaxRepe, para medir la proporción de lo que ocupa un objeto en un contenedor. La primera de ellas se refiere al porcentaje que el objeto más grande de la instancia ocupa dentro del contenedor, y la segunda se refiere al porcentaje que representa MaxRepe respecto al número de objetos de la instancia.

Como se mostró en el trabajo de Pérez *et al.* (2014), se realizaron pruebas a 16,413 instancias, a las que se les calculó cada uno de los límites, obteniéndose varios patrones de comportamiento. Este análisis permitió identificar qué límites inferiores deben calcularse en cada caso para obtener el límite inferior final. A continuación se explican estos patrones de comportamiento asociándolo al número de línea especificado en el Algoritmo 1:

*Línea 5.* Esta condición hace referencia a las instancias donde los objetos no ocupan más de 25% de capacidad del contenedor y donde el porcentaje de MaxRepe

**Algoritmo 1** Metaheurística LBMH para calcular el límite inferior

Entrada: $w = \{w_1, w_2, \ldots, w_n\}$, $W$
Salida: *lowerBound* valor del límite inferior calculado

```
1.   porcItem1=w1*100/C
2.   porcMaxRepe=MaxRepe*100/n
3.   kurt=Kurtosis()
4.   if
5.    (porcItem1<25 and porcMaxRepe<20) or
6.    (porcItem1<=57 and porcMaxRepe<10) or
7.    (porcItem1>57 and porcItem1<=59 and
8.     porcMaxRepe<3) or
9.    (porcItem1>59 and porcItem1 <=65 and
10.    porcMaxRepe<1.5) then
11.        lowerBound←max{LB1, LBJ}
12   else
13.    if
14.      (LB1=LB2 and kurt>8000) or (kurt>
15.       15000) then
16           LB3←0
17.    lowerBound←max {LB1, LB2, LB3, LBJ}
18.  return lowerBound
```

Tabla 1. Descripción de las variables empleadas en el algoritmo 1

| Variable | Descripción |
|---|---|
| *porcItem1* | Porcentaje del espacio que ocupa el objeto más grande de la instancia en el contenedor |
| *porcMaxRepe* | Porcentaje asociado con la métrica MaxRepe de acuerdo al número de objetos en la instancia |
| *MaxRepe* | Número máximo de repeticiones |
| *Kurt* | Valor de la curtosis de la instancia |
| $w_1$ | Peso del objeto más grande de la instancia |
| *lowerBound* | Límite inferior de la instancia |
| *N* | Número de elementos |
| *C* | Capacidad del contenedor |
| *LB1, LB2, LB3, LBJ* | Límites calculados |

es menor a 20%, esto indica que los objetos son muy pequeños y es posible encontrar objetos con el mismo peso.

*Línea 6.* Esta condición la cumplen aquellas instancias en donde el peso de objeto más grande no excede 57% de la capacidad del contenedor y MaxRepe es menor a 10%, indicando que el número de objetos con igual peso no debe exceder 10%.

*Línea 7 y 8.* Esta condición la cumplen aquellas instancias donde el objeto más grande ocupa entre 57 y 59% de la capacidad del contenedor y MaxRepe es menor a 3%.

*Línea 9 y 10.* En este caso se habla de las instancias, cuyo objeto más grande sobrepasa la mitad de la capacidad del contenedor (ocupa entre 59 y 65% de la capacidad del contenedor) y MaxRepe es menor a 1.5%, es decir, casi no hay objetos con el mismo peso.

Esto permite concluir que si se cumple alguna de las condiciones anteriores, entonces el límite inferior se calcula como el máximo valor de los valores obtenidos entre LB1 y LBJ. Se puede observar que entre mayores sean los objetos, el valor de MaxRepe disminuye.

En el caso de que no se cumpla ninguna de las condiciones anteriormente expuestas, entonces se verifica el valor de la Kurtosis, si esta es mayor a 8000 y los límites LB1 y LB2 son iguales o simplemente la Kurtosis excede los 15000, entonces esto indica que obtener LB3 puede ser un proceso demasiado costoso en tiempo, por lo que no se debe calcular LB3 y se asigna cero a LB3. En cualquier caso el límite inferior es el máximo de LB1, LB2, LB3 y LBJ.

## Heurística de preprocesamiento para reducir el tamaño de la instancia

Para solucionar instancias del BPP bajo cualquier enfoque, una de las principales dificultades es el tiempo utilizado en solucionar el problema, sobre todo cuando el número de elementos es muy grande. Por lo que es importante reducir el número de objetos de la instancia.

En este trabajo se presenta una heurística de preprocesamiento que permite disminuir el tamaño de la instancia, esta es una de las principales contribuciones de esta investigación. Esta heurística aplica dos criterios de reducción que eliminan de manera temporal objetos, provocando que el tamaño de la instancia se reduzca, sin que por ello se pierda la optimalidad del resultado. Esta heurística puede aplicarse como una etapa de preprocesamiento para cualquier método de solución del BPP. A la solución obtenida se le agregan los objetos eliminados temporalmente. Si se utiliza el modelo propuesto en este trabajo para solucionar la instancia reducida, entonces el número de variables es menor que si se modela la instancia completa.

Al analizar los datos de las instancias de los conjuntos hard28 y 53nirup, se observaron 2 patrones de comportamiento que permitieron proponer dos criterios de reducción (criterio 1A y criterio 2A), los cuales se presentan a continuación:

*Criterio1A*

Sea $w_i$ el peso asociado al i-ésimo objeto y $n$ el número de objetos de la instancia. Si $w_i$=1, entonces eliminarlo del conjunto de datos, $\{\forall\ i,\ 0 < i \leq n\}$. Las variables que representan los objetos con peso de longitud 1 no son consideradas en el modelo.

Cada objeto $i$ eliminado por el criterio 1A, se incorpora a la solución final, en los contenedores parcialmente llenos. Si a un contenedor le queda espacio, al menos deberá ubicarse un objeto de tamaño 1. Si no hay contenedores disponibles, entonces se abre un nuevo contenedor.

*Criterio 2A*

Sean $w_i$ y $w_j$ los pesos asociados al i-*ésimo* y al *j-ésimo* objeto de la instancia, y $C$ la capacidad del contenedor.

Si $w_i + w_j = C$ entonces eliminar los objetos $i$ y $j$ de la instancia y depositarlos en un contenedor. Esta operación se realiza para todos los objetos de la instancia.

Esto se explica de la siguiente manera: si dos objetos llenan el contendor, no existe otra combinación de objetos que permita mejorar esta solución, por lo que es conveniente no considerar estos objetos para la construcción del modelo.

A continuación se aplican los criterios de reducción (1A y 2A) en la etapa de preprocesamiento a la instancia de entrada del ejemplo de la sección 3. La instancia reducida se resuelve con el modelo propuesto.

Al aplicar el criterio 1A, la instancia reducida queda de la siguiente manera: $W$={6, 5, 4, 3, 2} y $b$={2, 2, 1, 2, 4}, se eliminaron temporalmente 3 objetos cuyo peso es 1.

A la instancia reducida se le aplica el criterio 2A. Con la aplicación del criterio 2A se eliminaron 6 objetos, y con ellos se llenaron 3 contenedores: *Bin*1={5,2}, *Bin*2={5,2}, *Bin*3={4,3}. Por consiguiente, la instancia reducida se presenta a continuación: $W$={6,3,2} y $b$={2,1,2}, esta instancia cuenta ahora con 5 objetos. La instancia reducida se soluciona mediante la técnica de arcos de flujo utilizando el modelo propuesto. El conjunto de arcos, modelo y solución se describen a continuación.

Para la instancia reducida, los arcos correspondientes son: *arcos*={(0,2), (0,3), (0,6), (2,3), (2,4), (3,4), (3,5), (4,5), (5,6), (5,7), (6,7)}. Los arcos se representan gráficamente en la figura 3.

En este caso se fija en el modelo la variable $x_{06}$ que está asociada al arco (0,6), esta variable se sustituye por su demanda, $x_{06}$=2. El modelo se presenta a continuación

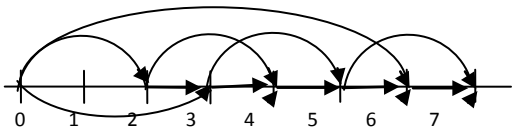


Figura 3. Grafo asociado al modelo ejemplo, después de aplicar los criterios de reducción propuestos en este trabajo

Minimizar $z$
Sujeto a:

$$z - x_{02} - x_{03} - 2 = 0$$
$$x_{02} - x_{23} - x_{24} = 0$$
$$x_{03} + x_{23} - x_{34} - x_{35} = 0$$
$$x_{24} + x_{34} - x_{45} = 0$$
$$x_{35} + x_{45} - x_{56} - x_{57} = 0$$
$$2 - x_{67} = 0$$
$$x_{57} + x_{67} - z = 0$$
$$x_{03} \geq 1$$
$$x_{02} + x_{24} + x_{57} \geq 2$$
$$z \leq 4$$

Resolviendo el modelo para esta instancia reducida, se obtiene después: $z$=4, $x_{03}$=2, $x_{35}$=2, $x_{57}$=2, $x_{67}$=2. Se agrega la variable que se fijó, $x_{06}$=2. Una vez que se obtiene la solución del modelo se interpreta para ubicar de manera adecuada los objetos en los contenedores.

Una forma de ubicar los objetos en los contenedores se muestra en la figura 4a. Finalmente se agregan a los contenedores parcialmente llenos los objetos con peso 1 y se agregan los contenedores que se llenaron con el criterio 2A. Este proceso se muestra en la figura 4b y c.

En la figura 4 se puede observar claramente que el número de contenedores mínimos necesarios para ubicar los objetos se mantuvo.

## Implementación de un sistema para pruebas

El sistema de pruebas que se desarrolló incluye: el algoritmo para resolver el modelo y los criterios de reducción propuestos por Valerio (1999), el modelo y los criterios 1A y 2A propuestos en este trabajo. Los algoritmos se implementaron en el lenguaje de programación C++. Sin pérdida de generalidad, para todas las ejecuciones se asume que los datos de entrada son números enteros.

El equipo de pruebas tiene las siguientes características: procesador Intel Core Quad y 2Gb de memoria RAM. Para resolver el modelo de programación entera construido se utilizó el *software* Lingo 11.0 para Windows.

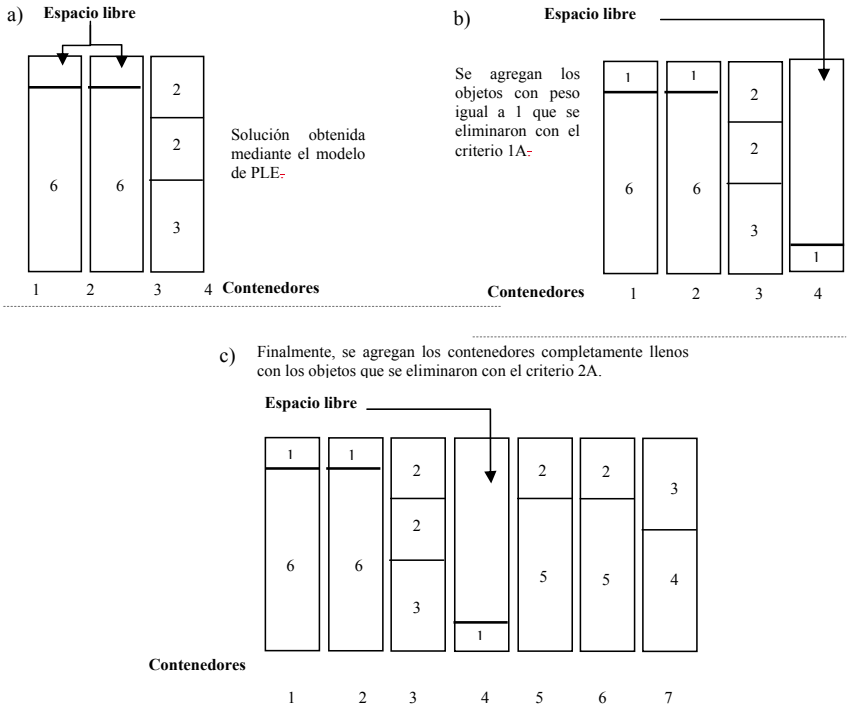


Figura 4. a) ubicación de los objetos de acuerdo con la solución del modelo reducido, b) ubicación de los objetos al agregarse los 1's eliminados con el criterio 1A, c) a la solución parcial se le agregan los contenedores con los objetos eliminados con el criterio 2A, obteniéndose la solución final

El sistema posee tres módulos principales. El primer módulo denominado *PreInstancia* se encarga de preparar la entrada de datos al formato requerido, para posteriormente aplicar los criterios 1A y 2A a la instancia de entrada. Como los objetos se encuentran ordenados en forma decreciente, el criterio 1A se aplica únicamente si el último elemento es 1. Y el criterio 2A se aplica únicamente si el primer elemento es mayor o igual a la mitad de la capacidad del contenedor.

El segundo módulo toma como entrada de datos la instancia de salida del módulo *PreInstancia* y genera el modelo desarrollado. Este modelo se soluciona utilizando Lingo, la salida que proporciona este *software* se interpreta al formato que utiliza el sistema de pruebas. El tercer módulo, *PostInstancia*, se encarga de ubicar los objetos con peso igual a 1 en los contenedores que no estén completamente llenos, en el caso de que no existan contenedores parcialmente llenos entonces se abre un nuevo contenedor. Además se encarga de generar la solución final agregando los contenedores completamente llenos, resultantes de la aplicación del criterio 2A, a la solución obtenida del modelo. En la figura 5 se representan las fases con las que cuenta el sistema de pruebas.

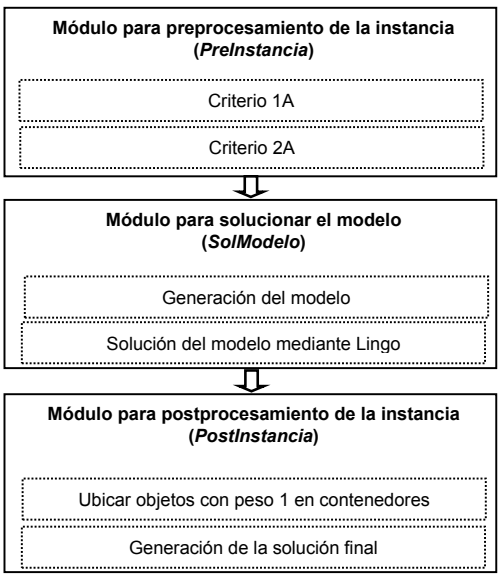


Figura 5. Fases de las que consta el sistema de pruebas

## Resultados experimentales

Para validar el modelo desarrollado y los criterios 1A y 2A propuestos se realizó una gran cantidad de experimentos. La experimentación se dividió en dos partes, en la primera se evaluaron los criterios 1A y 2A y en la segunda se evaluó el modelo propuesto. A continuación se describen ambas partes de la experimentación.

En la primera parte los criterios 1A y 2A se aplicaron a 16413 instancias, estas instancias se agrupan en los conjuntos que se enlistan a continuación: Falkenauer (1996), que contiene a los conjuntos conocidos

como uniform y triplets; bin1data, bin2data y bin3data (Scholl *et al*., 1997); Schwerin y Waescher (Schwerin y Wäscher,1999); Dual_distribution (Burke, 2010); ffd-hardA, ffd-hardB, ffd-hardC y ffd-hardD (Schwerin y Wäscher, 1999); hard28 y 53nirup (Schoenfield, 2002).

La aplicación de estos criterios hizo posible la reducción del número de objetos de las instancias que contenían objetos con peso igual a 1 y objetos cuyo peso es mayor a la mitad de la capacidad del contenedor. De acuerdo con la evaluación realizada, los conjuntos de instancias de nuestro caso de estudio que no cumplen con estas condiciones son: bin3data, schwerin y waescher. En total, 51.6% de las instancias no cuentan con las características antes mencionadas.

En la tabla 2 se presenta el número de objetos en promedio que se eliminaron aplicando los criterios 1A y 2A en cada conjunto de instancias. La primera columna, *NomInstancia*, representa el nombre del conjunto de instancia, *nObjetos* es el número de objetos en promedio del conjunto de instancias y *nObjetos eliminados* es el promedio de objetos que se eliminaron por conjunto de instancias. Se observa que los conjuntos de instancias más favorecidos con la aplicación de estos criterios fue dualdistribution cuya reducción fue de 50.95%, y falkenauer con una reducción de 33.03%.

El tiempo empleado para la reducción en cada uno de los conjuntos de datos se presenta en la figura 6. En la gráfica se puede observar que el tiempo empleado para llevar a cabo la eliminación temporal de objetos mediante el criterio 1A y 2A no es significativo. Es importante mencionar que existen conjuntos de instancias que no cumplen con las características necesarias para aplicarles los criterios 1A y 2A, tales conjuntos son bin3data, triplets que pertenece al conjunto falkenauer, Schwerin y waescher.

La segunda parte de la experimentación consistió en comparar los resultados que ofrece el sistema e integra los criterios 1A y 2A y la implementación del modelo desarrollado con el modelo de Valerio. Para esta actividad se seleccionaron a los conjuntos hard28, por considerarse uno de los más difíciles en calcular su solución, así como 53nirup, porque los límites inferiores calculados no coinciden con el número de contenedores de las soluciones obtenidas, además de ser interesante calcular las soluciones mediante el modelo de Valerio y el modelo propuesto en este trabajo.

Tabla 2. Número de objetos en promedio eliminados en cada conjunto de instancia y el porcentaje de reducción

| NomInstancias | nObjetos | nObjetos eliminados | % reducción |
|---|---|---|---|
| 53nirup | 124 | 15 | 12.47 |
| bin1data | 212 | 112 | 22.34 |
| bin2data | 213 | 26 | 12.28 |
| bin3data | 200 | 0 | 0.00 |
| dualdistribution | 4654 | 2371 | 50.95 |
| falkenauer | 383 | 126 | 33.03 |
| ffd-hardA | 150 | 15 | 9.84 |
| ffd-hardB | 143 | 14 | 9.93 |
| ffd-hardC | 121 | 18 | 15.11 |
| ffd-hardD | 118 | 25 | 21.17 |
| hard28 | 181 | 25 | 13.72 |
| schwerin | 110 | 0 | 0.00 |
| waescher | 127 | 0 | 0.00 |

Se realizó una comparación del tiempo y la cantidad de arcos que se generaron utilizando el modelo de Valerio frente al modelo desarrollado en este trabajo, incluyendo las reducciones propuestas. Mediante estas evaluaciones se llegó a la conclusión de que utilizando esta propuesta es posible reducir el número de arcos y el tiempo de solución de la mayoría de las instancias.

En las figuras 7 y 8 se observa que el número de arcos para los conjuntos hard28 y 53nirup es mucho menor que el utilizado en el modelo de Valerio. En las figuras siguientes se denomina al modelo de Valerio únicamente como Modelo Valerio.

Se obtuvieron las soluciones óptimas de todas las instancias del conjunto hard28 reportadas en la literatura

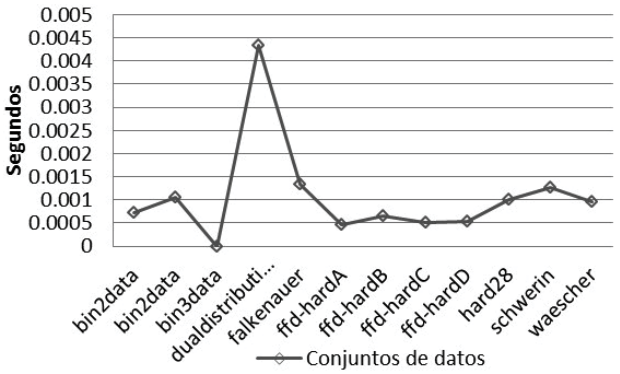


Figura 6. Tiempo utilizado en la aplicación de los criterios 1A y 2A a los diferentes conjuntos de datos

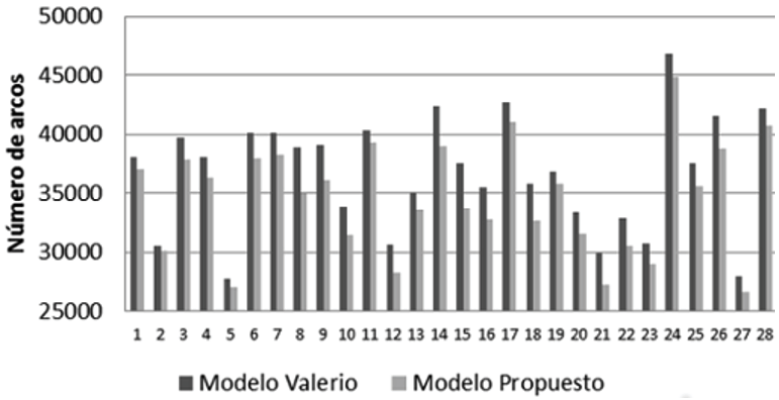


Figura 7. Número de arcos generados por el modelo desarrollado y el modelo de Valerio para hard28

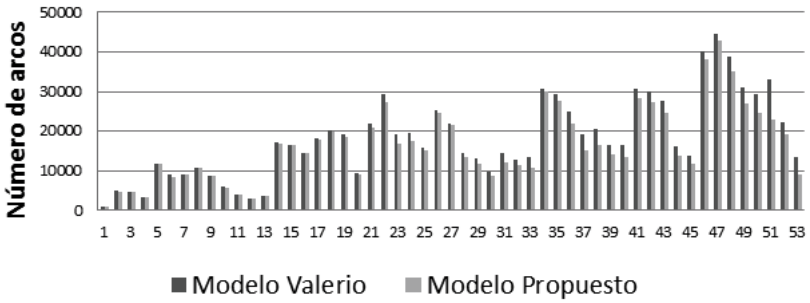


Figura 8. Número de arcos generados por el modelo desarrollado y el modelo de Valerio para 53nirup

(Mexicano, 2004). Las soluciones óptimas que se encontraron para el conjunto de instancias 53nirup, son las mismas que utilizando el modelo de Valerio.

En la figura 9 se observa una gráfica de tiempo para el conjunto hard28, por motivos de espacio no se muestran los nombres de todas las instancias. La gráfica muestra el tiempo en minutos empleado por el modelo de Valerio para resolver el conjunto de instancias hard28 y el tiempo empleado por el modelo desarrollado, que incluye los criterios de reducción 1A y 2A. El tiempo que consume el modelo es en promedio 43% menor al empleado utilizando el modelo de Valerio.

En la figura 10 se muestra una gráfica del tiempo utilizado por el modelo de Valerio y el tiempo utilizado por el modelo desarrollado junto con las reducciones, para el conjunto de instancias 53nirup. En este caso, el modelo utiliza en promedio 19.7% menos tiempo que el que utiliza el modelo de Valerio (1999). El nombre de las instancias del conjunto 53nirup (Schoenfields, 2002) se numeraron de acuerdo con su orden lexicográfico, y

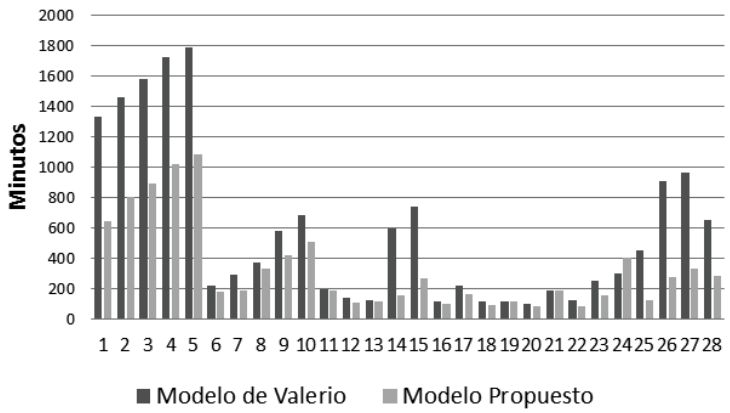


Figura 9. Comparación del tiempo empleado utilizando el modelo de Valerio y el modelo desarrollado para el conjunto de instancias hard28

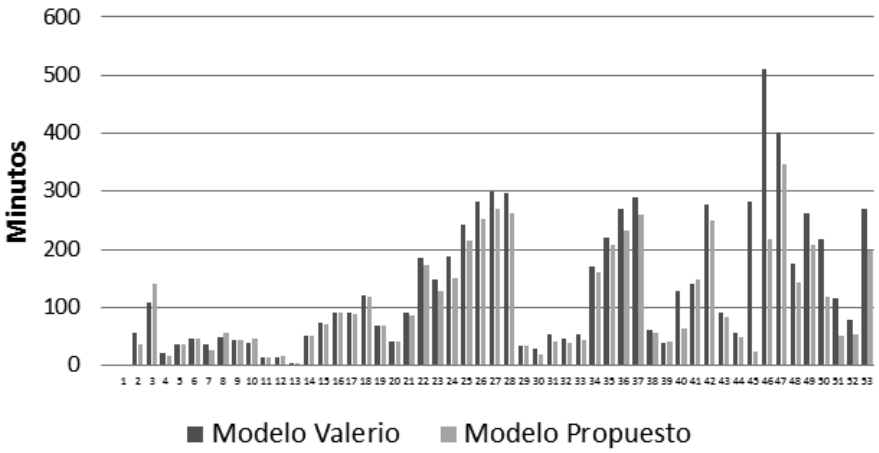


Figura 10. Comparación de los tiempos empleados por el modelo de Valerio y el modelo propuesto, aplicados al conjunto de instancias 53nirup

este número es el que aparece como etiqueta en el eje horizontal.

Se redujo el número de arcos en un 9.4% para el conjunto 53nirup y 5.5% para el conjunto hard28. Reduciendo el tamaño de las instancias se logra que el número de arcos utilizados en el modelo desarrollado sea menor, provocando con ello disminuir el número de variables finales utilizadas en el modelo.

En aras de ampliar la experimentación con el sistema se aplicaron los criterios de reducción y el modelo a la familia de problemas: bin1data, falkenauer (uniform y triplets) y un subconjunto de bin2data, dualdistribution y schwerin. En torno a los resultados obtenidos se llegó a las siguientes conclusiones:

1. Para bin1Data y el subconjunto de bin2data se encontró la solución óptima reportada.
2. En el caso del conjunto triplets de falkenauer y para schwerin se encontró la solución óptima de todas las instancias tomando como base que se llega al óptimo cuando el número de contenedores de la solución encontrada coincide con el límite inferior calculado.
3. A los conjuntos triplets y schwerin no se les pueden aplicar los criterios de reducción, porque los objetos a ubicar tienen un peso menor a la mitad de la capacidad del contenedor y no tienen objetos con peso igual a 1.
4. Una particularidad del conjunto dualdistribution es su dimensión, es decir, el número de objetos a ubicar es de 5000. Gracias a los criterios de reducción aplicados, el tiempo para encontrar la solución óptima para este subconjunto fue razonable.
5. Los resultados obtenidos para el conjunto dualdistribution puede ser el valor óptimo, pero no está garantizado debido a que el valor del límite inferior no siempre coincidió con el valor óptimo alcanzado.
6. El modelo y los criterios desarrollados permiten alcanzar la solución óptima de 6 conjuntos de datos con características diferentes. En la tabla 3 se muestra el número de instancias en donde se asume que se llegó al óptimo debido a que el número de contenedores de la solución coincide con el límite inferior. Para el caso de bin1data y de bin2data se encontraron los óptimos de todas las instancias de prueba.

Tabla 3. Número de instancias por conjunto de datos en donde se encontró garantizada la solución óptima

| Conjuntos de instancias | Núm. de instancias de la muestra | Núm. de instancias que coincidieron con el límite inferior calculado | Núm. de instancias que coincidieron con el valor Núm. óptimo conocido |
|---|---|---|---|
| hard28 | 28 | 23 | 28 |
| 53nirup | 53 | 3 | 53 |
| bin1data | 720 | 664 | 720 |
| bin2data | 55 | 55 | 55 |
| dualdistribution | 68 | 46 | - |
| falkenauer | 140 | 139 | - |
| schwerin | 11 | 11 | - |

## Conclusiones

En este trabajo se muestra que es posible resolver el problema de *Bin Packing* de una sola dimensión, mediante la reducción de instancias y la preasignación de objetos en contenedores, aplicando la técnica de arcos de flujo a la instancia reducida, sin perder la optimización del problema original para aquellos conjuntos en donde el óptimo coincide con el límite inferior o es conocido.

El nuevo modelo propuesto en este trabajo, incorpora directamente a las restricciones del problema el límite inferior, permitiendo reducir el espacio de soluciones.

Las suposiciones en cuanto a la aplicación de los criterios 1A y 2A propuestos en este artículo son válidas de acuerdo con los resultados obtenidos en el caso de estudio, y se pueden utilizar como parte del preprocesamiento de una instancia de entrada.

Con base en los resultados experimentales se mostró que el uso de los criterios 1A y 2A, junto con el modelo propuesto, reducen el tiempo de solución comparado con el tiempo utilizado por el modelo de Valerio.

El modelo propuesto es particularmente útil en aquellas instancias que incluyan objetos donde el peso sea mayor a la mitad de la capacidad del contenedor. En el caso de arcos de flujo, al reducirse el tamaño de la instancia, el número de arcos del modelo también se reduce.

La introducción de los criterios de reducción desarrollados en este trabajo permitió para el conjunto hard28 reducir sustancialmente el tiempo de cálculo para la obtención de las soluciones óptimas de casi todas las instancias. Se destaca la reducción en tiempo para el conjunto hard28, que fue de 43% respecto al tiempo empleado por el modelo de Valerio. Para este conjunto se encontraron las soluciones óptimas de todas las instancias.

Al conjunto bin1data se le pueden aplicar los criterios de reducción y se llega al óptimo de todas las instancias. Se tomó una muestra del conjunto bin2data y ofreció el mismo comportamiento.

A pesar de que al conjunto triplets no se le pueden aplicar los criterios de reducción se llegó al óptimo en todos los casos.

Los resultados muestran que mediante esta estrategia para solucionar el BPP es posible mejorar los resultados en cuanto a calidad de la solución, debido a que va encaminada a obtener siempre la solución óptima. Consideramos que la metaheurística utilizada para calcular el límite inferior se puede mejorar mediante el uso y el análisis de nuevas métricas de caracterización de instancias. También es posible mejorar el proceso de generación de arcos mediante la modificación del modelo propuesto. Es importante mencionar que se está trabajando en el desarrollo de nuevos criterios para reducir el tamaño de la instancia.

## Referencias


Álvarez V.M. *Modelo para representar la complejidad del problema y el desempeño de algoritmos*, (tesis de maestría), México, Instituto Tecnológico de Ciudad Madero, 2006.

Alves F.D. *Bin packing and related problems:pattern-based approaches*, (tesis de maestría), Portugal, Faculdade de Ciencias da Universidade do Porto, 2012.

Alvim A.C.F., Glover F., Ribeiro C.C., Aloise D.J. A hybrid improvement heuristic for the *Bin Packing* problem. *Journal of Heuristics*, volumen 10, 2004: 205-229.

Balas E. Disjuntive programming. *Annals of Discrete Mathematics*, volumen 5, 1979: 3-51.

Burke E.K., Hyde M.R., Kendall G. Providing a memory mechanism to enhance the evolutionary design of heuristics, Proceedings of the IEEE World Congress on Computational Intelligence (CEC 2010), 2010.

Cambazard H. y O'Sullivan B. Propagating the bin packing constraint using linear programming, CP'10 Proceedings of the 16th international conference on Principles and practice of constraint programming, pp. 129-136.

Cruz L. *Clasificación de algoritmos heurísticos para la solución de problemas de Bin Packing*, (tesis doctorado), México, Centro Nacional de Investigación y Desarrollo Tecnológico (CENIDET), 2004.

Falkenauer E. A hybrid grouping genetic algorithm for *Bin Packing*. *Journal of Heuristics*, volumen 2, 1996: 5-30.

Garey M.R. y Johnson D.S. *Computers and intractability: A guide to the theory of NP-completeness*, Nueva York, W.H. Freeman and Company, 1979.

Jarboui B., Ibrahim S., Rebai A. A new destructive bounding scheme for the bin packing problem. *Annals of Operations Research*, volumen 179, 2010: 187-202.

LINGO user's guide. Lindo Systems INC., USA, 2008.

Martello S., Toth P. *Knapsack problems. Algorithms and computer. Implementations*, Inglaterra, DEIS University of Bologna, John Wiley & Sons, 1990.

Mexicano A. *Caracterización de conjuntos de instancias difíciles del problema de Bin Packing orientada a la mejora de algoritmos metaheurísticos mediante el uso de técnicas de minería de datos*, (tesis doctorado), México, Centro Nacional de Investigación y Desarrollo Tecnológico (CENIDET), 2004.

Pérez J., Castillo H., Vilariño D., De-la-Rosa R., Mexicano A., Zavala J.C., Santaolaya R., Estrada H. Metaheuristic for selecting lower bound applied to the problema of bin packing, Conference Proceedings International Conference on Innovative Trends in Science, Engineering and Management 2014 (ICITSEM2014), Dubai, UAE, pp. 196-201, 2014.

Schaus P. *Solving balancing and Bin Packing problems with constraint programming*, (PhD thesis), Belgica, 2009.

Shaw P. *A constraint for bin packing*, CP 2004, LNCS 3258, pp. 648-662, 2004.

Schoenfields J.E. *Fast, exact solution of open packing problems without linear programming*, Draft, US Army Space & Missile Defense Command, Huntsville, Alabama, USA, 2002.

Scholl A., Klein R., Jürgens C. BISON: A fast hybrid procedure for exactly solving the one-dimensional bin packing problem. *Computers Operations Research*, volumen 24 (número 7), 1997: 627-645.

Schwerin P. y Wäscher G. A new lower bound for the *Bin Packing* problem and its integration into MTP. *Pesquisa Operacional*, volumen 19 (número 2), 1999: 111-130.

Taha H.A. *Investigación de operaciones y flujo en redes*, 5a ed., Alfaomega, 2004.

Valerio-De Carvalho J.M. Exact solution of bin packing problems using column generation and branch and bound. *Annals of Operation Research*, volumen 86, 1999: 629-659.

Valerio-De Carvalho J.M. LP models for bin packing and cutting stock problems. *European Journal of Operational Research*, volumen 141, 2002: 253-273.

**Este artículo se cita:**

**Citación estilo Chicago**

Pérez-Ortega, Joaquín, Hilda Castillo-Zacatelco, Darnes Vilariño-Ayala, Adriana Mexicano-Santoyo, José Crispín Zavala-Díaz, Alicia Martínez-Rebollar, Hugo Estrada-Esquivel. Una nueva estrategia heurística para el problema de *Bin Packing*. *Ingeniería Investigación y Tecnología*, XVII, 02 (2016): 155-168.

**Citación estilo ISO 690**

Pérez-Ortega J., Castillo-Zacatelco H., Vilarino-Ayala D., Mexicano-Santoyo A., Zavala-Díaz J.C., Martínez-Rebollar A., Estrada-Esquivel H. Una nueva estrategia heurística para el problema de *Bin Packing*. *Ingeniería Investigación y Tecnología*, volumen XVII (número 2), abril-junio 2016: 155-168.

## Semblanzas de los autores

*Joaquín Pérez-Ortega.* Es profesor investigador en el área de ingeniería de software del Centro Nacional de Investigación y Desarrollo Tecnológico (CENIDET), México. Egresado del Instituto Tecnológico y de Estudios Superiores de Monterrey, México. Sus áreas de interés son la ingeniería de software, la minería de datos, la algoritmia y la modelación matemática.

*Hilda Castillo-Zacatelco.* Es profesora de la Benemérita Universidad Autónoma de Puebla, en la Facultad de Ciencias de la Computación en Puebla. Actualmente está inscrita en el programa de doctorado en ciencias de la computación que imparte el Centro Nacional de Investigación y Desarrollo Tecnológico (CENIDET), México. Su área de interés es el diseño y desarrollo de Algoritmos.

*Darnes Vilariño-Ayala.* Es profesora investigadora de la Benemérita Universidad Autónoma de Puebla en la Facultad de Ciencias de la Computación. Sus áreas de interés son el procesamiento de lenguaje natural, los sistemas distribuidos y la programación lineal.

*Adriana Mexicano-Santoyo*. Es profesora del Instituto Tecnológico de Cd. Victoria, egresada del Centro Nacional de Investigación y Desarrollo Tecnológico (CENIDET), México. Sus áreas de investigación son la minería de datos, reconocimiento de patrones y la algoritmia.

*José Crispín Zavala-Díaz.* Es profesor-investigador en el área de informática de la Facultad de Contaduría, Administración e Informática de la Universidad Autónoma del Estado de Morelos (FCAEI-UAEM), México. Egresado del Instituto Tecnológico y de Estudios Superiores de Monterrey, México. Sus áreas de interés son: modelado, algoritmia y aplicaciones en investigación de operaciones y cómputo paralelo.

*Alicia Martínez-Rebollar.* Es profesora-investigadora del Centro Nacional de Investigación y Desarrollo Tecnológico (CENIDET), México. Egresada de la Universidad Politécnica de Valencia, España y de la Universidad de Trento, Italia. Sus líneas de investigación son: sistemas distribuidos, en particular, inteligencia ambiental enfocada a la salud, cómputo móvil y redes sociales.

*Hugo Estrada-Esquivel.* Es investigador del Fondo de Información y Documentación para la Industria de México (INFOTEC), es egresado de la Universidad Politécnica de Valencia, España y de la Universidad de Trento, Italia. Sus líneas de investigación son: modelado organizacional, web semántica y ontologías.