

Caracterización de instancias difíciles del problema de Bin Packing orientada a la mejora de algoritmos metaheurísticos

Adriana Mexicano Santoyo¹, Joaquín Pérez Ortega², Gerardo Reyes Salgado²,
Nelva Nely Almanza Ortega²

¹ Instituto Tecnológico de Cd. Victoria,
México

² Centro Nacional de Investigación y Desarrollo Tecnológico, Cuernavaca,
México

mexicanoa@gmail.com, jpo_cenidet@yahoo.com.mx, nelvanely@cenidet.edu.mx

Resumen. En este trabajo se presenta una metodología para la caracterización de instancias difíciles del problema de *Bin Packing* usando Minería de Datos. El objetivo es que las características de las instancias proporcionen ideas para desarrollar nuevas estrategias para encontrar soluciones óptimas mediante la mejora de los algoritmos de solución actuales o mediante el desarrollo de nuevos. De acuerdo a la literatura especializada, en general, la caracterización de instancias ha sido utilizada para predecir qué algoritmo resuelve mejor una instancia o para mejorar el algoritmo asociando las características de la instancia con el desempeño de dicho algoritmo. A diferencia de los trabajos anteriores, este trabajo propone que el desarrollo de algoritmos de solución eficientes puede ser guiado por una previa identificación de las características que representan un alto impacto en la dificultad para obtener su solución. Para validar nuestro enfoque se utilizó un conjunto de 1,615 instancias, 6 algoritmos bien conocidos del problema de *Bin Packing* y 27 métricas iniciales. Después de aplicar técnicas de agrupamiento de Minería de Datos para la caracterización de las instancias, se encontraron 5 métricas que ayudaron a caracterizar 4 grupos con las instancias que no fueron resueltas por ninguno de los algoritmos usados en este trabajo. En base al conocimiento obtenido de la caracterización de las instancias, se propuso un nuevo método de reducción de instancias que contribuye a reducir el espacio de búsqueda de un algoritmo metaheurístico. Los resultados experimentales muestran que aplicando el método de reducción es posible encontrar más soluciones óptimas que las reportadas en el estado del arte por las mejores metaheurísticas.

Palabras clave. Metaheurísticas, bin packing, caracterización, agrupamiento, reducción, descubrimiento de conocimiento.

Characterization of Difficult Bin Packing Problem Instances Oriented to Improve Metaheuristic Algorithms

Abstract. This work presents a methodology for characterizing difficult instances of the Bin Packing Problem using Data Mining. Characteristics of such instances help to provide ideas for developing new strategies to find optimal solutions by improving the current solution algorithms or developing new ones. According to related work, in general, instance characterization has been used to make prediction of the algorithm that best solves an instance, or to improve one by associating the instance characteristics and performance of the algorithm that solves it. However, this work proposes the development of efficient solution algorithms guided by previous identification of characteristics that represent a greater impact on the difficulty of the instances. To validate our approach, we used a set of 1,615 instances, 6 well-known algorithms of the Bin Packing Problem, and 27 initial metrics. After applying our approach, 5 metrics were found relevant; these metrics helped to characterize 4 groups containing instances that could not be solved by any of the algorithms used in this work. Based on the gained knowledge from instance characterization, a new reduction method that helps to reduce the search space of a metaheuristic algorithm was proposed. Experimental results show that application of the reduction method allows finding more optimal solutions than those of best metaheuristics reported in the specialized literature.

Keywords. Characterization, clustering, metaheuristics, bin packing problem, reduction, knowledge discovery.

1. Introducción

En el área de la computación, uno de los problemas abiertos y vigentes es encontrar algoritmos eficientes y eficaces para resolver problemas de optimización discreta del tipo NP-hard [18, 24], como lo son *Bin Packing*, el Agente viajero, y el de la Mochila, entre otros, dichos problemas son comúnmente encontrados en situaciones prácticas como la planeación de la producción y logística. Este tipo de problemas son considerados de alta complejidad computacional ya que a la fecha no se conocen algoritmos que los puedan resolver en un tiempo polinomial. Con la finalidad de extraer conocimiento relevante que posibilite la generación de nuevas estrategias que sean *incorporadas en los algoritmos de solución actuales o que permitan la generación de nuevos algoritmos, en este trabajo se aborda el problema de la caracterización de instancias difíciles de resolver del problema de *Bin Packing*, cuya entrada es una secuencia de n objetos $I = \{w_1, w_2, \dots, w_n\}$ de diferentes tamaños. El objetivo es colocar los objetos en el mínimo número de contenedores de capacidad uniforme c , donde $0 < w_i \leq c$ [23]. El trabajo propone un enfoque basado en la hipótesis: “el desarrollo de algoritmos de solución eficientes, puede ser guiado por una previa identificación de las características que representan un alto impacto en la dificultad de las instancias”. En este sentido, se utilizaron técnicas de Minería de Datos con el propósito de caracterizar las instancias representativas del problema de *Bin Packing* que contribuyen en la obtención de patrones que pueden ser utilizados en el desarrollo de estrategias para la mejora del desempeño de los algoritmos de solución.

Para probar la hipótesis propuesta, se seleccionaron 1,615 instancias del problema de *Bin Packing*. Cada instancia fue resuelta con 6 algoritmos bien conocidos y caracterizada con 27 métricas iniciales. Al aplicar la Minería de Datos se obtuvieron los siguientes resultados:

- a) identificación de 5 características relevantes que permiten describir las instancias difíciles de resolver,

- b) identificación de cuatro patrones que contribuyen a la caracterización de 22 instancias cuya solución óptima no fue alcanzada por ninguno de los 6 algoritmos utilizados,
- c) Tres soluciones óptimas que no han sido reportadas por las mejores metaheurísticas de la literatura especializada.

Lo que resta del documento está estructurado de la siguiente forma: la Sección 2 muestra los trabajos relacionados con esta investigación, la Sección 3 presenta la metodología propuesta para la caracterización de instancias difíciles de resolver del problema de *Bin Packing* donde a manera de ejemplo se presenta la interpretación del conocimiento adquirido mediante el desarrollo un procedimiento de reducción de instancias el cual es generado a partir de la caracterización de las mismas y su incorporación sobre un algoritmo metaheurístico inspirado en HGGA_BP [33]. En la Sección 4 se presentan los resultados experimentales de ejecutar nuestro algoritmo sobre un conjunto de instancias difíciles de resolver, se destaca que fue posible encontrar más soluciones óptimas que las obtenidas inicialmente. Finalmente, la Sección 5 muestra las conclusiones obtenidas y los trabajos futuros.

2. Trabajos relacionados

A continuación se describen algunos de los trabajos, donde la caracterización de instancias es de gran importancia para obtener soluciones óptimas de instancias difíciles de resolver, en problemas NP. Los trabajos relacionados se dividieron en problemas NP y *Bin Packing* para mayor claridad del documento.

2.1. Problemas NP

De acuerdo al teorema “No free lunch” [42], no existe un algoritmo que domine la solución de todas las instancias de un problema NP-duro. Trabajos como los desarrollados en [20, 31] sugieren que es indispensable el conocimiento de las características de las instancias a resolver para desarrollar algoritmos que sean eficientes en el cálculo de soluciones. Así por ejemplo, en el

&DUFMIJDFL'QG-HLQMDQFLVGI@LQMGHDSUREOPD;GH%Q3DFNLQJHUIHQDGDH;D;PHVLD;GH;D;D;RULV;R;P;H;M;K;M;@;M;R;V;

año 2000 Merz [31] analizó la trayectoria de los algoritmos meméticos aplicados a la solución de instancias del problema de partición de grafos. Concluyó que el agregar las características de las instancias, puede ayudar en el desarrollo de métodos heurísticos altamente eficientes. En el año 2004 Hoos [20], mientras estudiaba el desempeño de un algoritmo de búsqueda local para el problema de MAX-SAT, incorporó un conjunto de métricas que le permitieron valorar la dificultad de las instancias y con ello mejorar el desempeño del algoritmo.

2.2. Problema de Bin Packing

El problema de *Bin Packing* ha sido ampliamente estudiado; en trabajos como [1, 4, 7, 10, 11, 12,15, 16, 17, 19, 21, 25, 26, 27, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41] se han desarrollado diferentes algoritmos con el objetivo de encontrar procedimientos eficientes que contribuyan a resolver dicho problema, sin embargo, sólo investigaciones como [9, 10, 32, 33, 34, 37] se han enfocado en la identificación de las características que representan alto impacto en la dificultad de resolver instancias. Dicha caracterización ha sido utilizada para hacer predicciones sobre el algoritmo que puede resolver mejor un conjunto particular de instancias, o bien para mejorar un algoritmo dado, donde la caracterización se realiza a partir del desempeño del algoritmo sobre la instancia. El trabajo más relevante en cuanto a caracterización de instancias del problema de *Bin Packing* es el desarrollado por Schwerin y Wäscher en 1997 [37]. En ese trabajo se expresó que el intervalo de donde son extraídos los valores de los pesos de los objetos puede tener efectos severos en el tiempo de cómputo y la calidad de la solución de los algoritmos. Se reportan como indicadores de dificultad para el algoritmo *Fist Fit Decreasing* (FFD): a) el peso de los objetos; b) la variabilidad y c) el efecto de multiplicidad. Cruz en 2004 [9] propuso un procedimiento sistemático, basado en aprendizaje automático y estadística. En dicho trabajo se incorporaron y relacionaron características críticas del problema de *Bin Packing* que ayudaron a seleccionar el algoritmo que mejor resuelve un caso dado. Otro de los trabajos destacables es el desarrollado por

Quiroz en los años 2009-2013 [33, 34], en el cual se presenta la mejora del algoritmo HGGA-BP propuesto por primera vez en [10, 32]. Dicho trabajo presenta un análisis exploratorio sobre el desempeño del algoritmo y un análisis de las características de las instancias difíciles de resolver y haciendo uso de relaciones obtiene información que permite realizar la mejora de un algoritmo de solución.

Los trabajos anteriores sugieren dos factores clave en cuanto a la solución de las instancias difíciles de un problema, uno es la caracterización del tipo de instancia y el otro es el conocimiento que se tenga sobre el algoritmo que mejor resuelve ese tipo de instancias. En este trabajo, lo que se aborda es la caracterización de las instancias con el objetivo de encontrar los patrones que las hacen difíciles de resolver.

3. Caracterización de instancias difíciles mediante agrupamiento

La metodología desarrollada para la caracterización de instancias difíciles del problema de *Bin Packing* está basada en el modelo de referencia para Minería de Datos CRISP-DM [8] cuyas fases son: entendimiento del negocio y de los datos, preparación de los datos, modelado, evaluación y despliegue de resultados. La Figura 1 muestra el enfoque de solución propuesto que incluye desde el preprocesado de los datos hasta la aplicación del conocimiento que se extrae para la mejora o generación de los algoritmos de solución.

Se muestra además que dado un conjunto de algoritmos, un conjunto de instancias y un conjunto de métricas; la caracterización de las instancias se realiza a través de la aplicación de técnicas de Minería de Datos donde como parte de la preparación de datos: se extraen las soluciones de las instancias al aplicar diferentes algoritmos, esto con la finalidad de identificar las instancias más difíciles de resolver; a partir de un conjunto de métricas se extraen los valores correspondientes a cada instancia y se selecciona un conjunto de métricas que sean relevantes para caracterizar las instancias más difíciles de resolver. En la fase de modelado como su nombre lo indica, se modela la

METODOLOGÍA BASADA EN CRISP-DM [8]

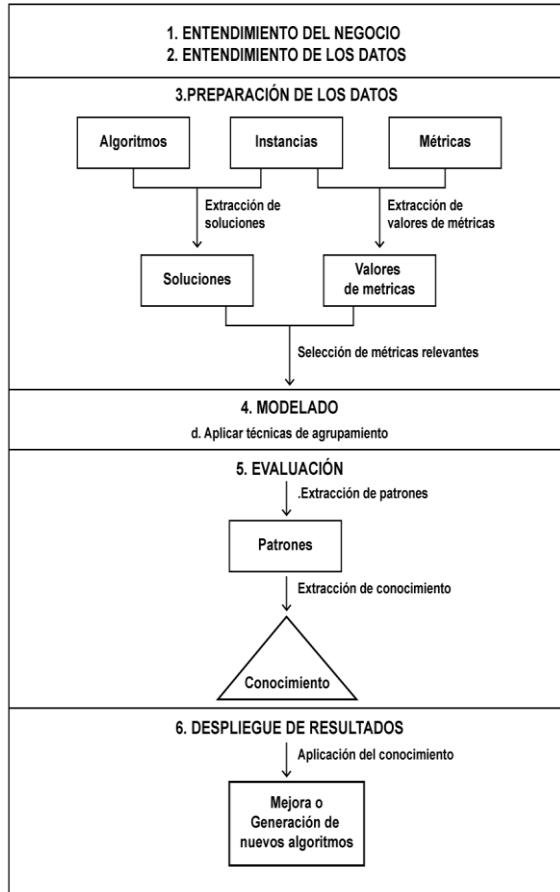


Fig. 1. Enfoque de solución propuesto

información, en este caso mediante agrupamiento y se obtienen los patrones que posibilitan describir las instancias difíciles de resolver. Durante la fase de evaluación, se analizan los patrones para extraer conocimiento.

Finalmente en la fase de despliegue de resultados, el conocimiento proporcionado por los patrones se incorpora como estrategia que al combinarse con los algoritmos de solución actuales, permite obtener soluciones óptimas que previamente no han sido encontradas. Otra posibilidad de uso del conocimiento obtenido es la generación de nuevos algoritmos para la solución de problemas reales. Las Secciones 3.1-3.6 describen cada una de las fases.

3.1. Entendimiento del negocio

Durante esta fase se determinó la siguiente hipótesis: “el desarrollo de algoritmos de solución eficientes puede ser guiado por una previa identificación de las características que representan un mayor impacto en la dificultad de las instancias”. Dada la hipótesis la caracterización de las instancias fue planteada en términos de determinar cuáles son las instancias que son difíciles de resolver por los algoritmos de solución actuales, encontrar las características que mejor las describen, aplicar el conocimiento adquirido para la generación de estrategias que redunden en la mejora y/o desarrollo de nuevos algoritmos de solución. Para alcanzar el objetivo se generaron los siguientes objetivos particulares:

- a) Seleccionar un conjunto de instancias de prueba $I = \{i_1, \dots, i_n\}$, un conjunto de algoritmos $A = \{a_1, \dots, a_j\}$ y un conjunto de métricas $M = \{m_1, \dots, m_l\}$ que permitan la caracterización de las instancias.
- b) Calcular las soluciones de las instancias en I con los algoritmos en A , para obtener el conjunto de soluciones $R = \{r_{1,1}, \dots, r_{n,j}\}$ que permitan identificar cuáles son las instancias más difíciles de resolver.
- c) Reducir el conjunto inicial de métricas M a partir de los valores obtenidos al aplicarlas sobre I y encontrar las que describan mejor a las instancias difíciles de resolver.
- d) Modelar los datos mediante agrupamiento para encontrar un conjunto de patrones $P = \{p_1, \dots, p_x\}$, que aporten información sobre las características de los conjuntos de instancias difíciles de resolver del problema de *Bin Packing*.
- e) Interpretar el conocimiento extraído a partir de los patrones relacionados con las instancias difíciles de resolver.
- f) Mediante el conocimiento adquirido, implementar al menos una estrategia que permita obtener mayor número de soluciones óptimas.

3.2. Entendimiento de los datos

En esta etapa se recopiló y analizó la información que contribuyó para caracterizar las

métricas relevantes, las cuales se describen a continuación.

Para la descripción de las métricas considere que: n es el número de objetos en una instancia, w_i es el peso del objeto i , c es la capacidad del contenedor, $tw = \sum_{i=1}^n w_i$ es la suma total de los pesos de los objetos en una instancia y para toda i la función $factor(c, w_i)$ identifica si un objeto es factor o no del contenedor c

$$factor(c, w_i) = \begin{cases} 1 & (c \bmod w_i) = 0 \\ 0 & (c \bmod w_i) \neq 0 \end{cases}$$

1. **Tamaño de la instancia.** Expresa la relación entre el tamaño de una instancia y el tamaño máximo resuelto. El valor de n_{max} se asignó a 1,000, ya que corresponde con el número de objetos de la instancia mayor en I (ver expresión 1):

$$p = \frac{n}{n_{max}}. \quad (1)$$

2. **Factores.** Expresa la proporción de objetos cuyos tamaños son factores de la capacidad del contenedor; un objeto i es un factor cuando la capacidad del contenedor c es múltiplo de su correspondiente tamaño de objeto w_i . En general, las instancias con muchos factores se consideran fáciles de resolver (ver expresión 2):

$$f = \frac{1}{n} \sum_{i=1}^n factor(c, w_i). \quad (2)$$

3. **Uso de contenedores.** Cuantifica la proporción de la suma de los tamaños de los objetos que pueden asignarse en un contenedor de capacidad c , y expresa la cantidad ideal de contenedores requeridos en la solución. Cuando este cociente toma un valor mayor o igual a 1 significa que todos los objetos caben en un contenedor y se le asigna un valor de 1 a la métrica (ver expresión 3):

$$b = \begin{cases} 1 & c \geq tw \\ c/W & c < tw \end{cases}. \quad (3)$$

4. **Media aritmética.** Expresa el promedio de los pesos de los objetos (ver expresión 4):

$$avg = \frac{tw}{n}. \quad (4)$$

5. **Rango.** Expresa la diferencia entre el peso mayor y el peso menor de los objetos en la instancia respecto de c (ver expresión 5):

$$r = \frac{\max_{i=1, \dots, n} w_i}{c} - \frac{\min_{i=1, \dots, n} w_i}{c}. \quad (5)$$

6. **Coefficiente de asimetría de Pearson.** Expresa la comparación entre la media y los pesos de los objetos de la instancia para observar que tan simétricos son los datos (ver expresión 6):

$$A_p = \frac{\sum_{i=1}^n (w_i - avg)^3}{n\sigma^3}, \quad (6)$$

donde σ es la desviación estándar de los pesos de los objetos.

7. **Menor.** Expresa el peso del objeto más pequeño en la instancia, en relación al tamaño del contenedor y permite ubicar el inicio de la distribución de pesos (ver expresión 7):

$$m = \frac{\min_{i=1, \dots, n} \{w_i\}}{c}. \quad (7)$$

8. **Frecuencia de repetición máxima.** Expresa la frecuencia máxima con la que un peso se repite en el conjunto de objetos de la instancia (ver expresión 8). f_i es el número de veces en que w_i aparece en w_1, \dots, w_n , $V = \{v_1, \dots, v_k\}$ es el conjunto de valores diferentes en $\{f_1, \dots, f_n\}$:

$$U = \max_{i=1, \dots, k} v_i. \quad (8)$$

La información relacionada con: instancias, resultados obtenidos al ejecutar los algoritmos y valores de las métricas por instancia, fue almacenada en una base de datos para su posterior modelado.

3.4. Modelado

Para el modelado de los datos se utilizó la técnica de agrupamiento, en particular el algoritmo k-means [29]. Durante la construcción del modelo fue necesario experimentar con varios valores iniciales de k hasta encontrar subconjuntos formados por las instancias difíciles de resolver (D) y separados lo más posible de las instancias que se pudieron resolver de forma óptima ($I \setminus D$).

Algoritmo 1: Formación de grupos y reducción de métricas

Entrada: $\hat{M} = \{v_{1,1}, \dots, v_{n,l}\}$, l , n , $k_0 = 30$, $\Delta = 5$
 1: Inicializar $k \leftarrow k_0$,
 2: Calcular k -means(k, \hat{M})
 3: **Mientras** en k -means(k, \hat{M}) ($D \cap I \setminus D \neq \emptyset$) o el siguiente incremento de k no mejora la separación de los conjuntos **hacer**
 4: $k \leftarrow k + \Delta$
 5: Calcular k -means(k, \hat{M})
 6: **finMientras**

Después de obtener una buena separación de las instancias difíciles de resolver, de manera empírica se buscó reducir las métricas que permitieran mantener los grupos generados.

El procedimiento de búsqueda del número de grupos tiene como objetivo encontrar $(D \cap I \setminus D) = \emptyset$ o lo más cercano a 0, y además permite reducir el número de métricas de manera empírica. Dicho procedimiento es representado por el Algoritmo 1, el cual recibe como parámetros de entrada un conjunto inicial de $l \cdot n$ valores, obtenidos al aplicar las $l = 8$ métricas sobre las $n = 1,615$ instancias en I , k_0 es el valor inicial de k y Δ corresponde a las unidades que se incrementará k en cada iteración. La línea 1 indica la inicialización del número de grupos k a utilizar con k-means $k_0 = 30$, la línea 2 indica el cálculo de los k grupos utilizando k-means y los valores de las métricas calculadas para cada instancia \hat{M} .

El agrupamiento se repite mientras no sean separados los conjuntos I y $I \setminus D$ o el siguiente incremento de k no mejora la separación de los conjuntos (ver línea 3). La línea 4 indica el incremento al número de grupos a calcular por k-

means y Δ corresponde a incrementos de 5 unidades por iteración (línea 5).

Durante la experimentación se utilizaron valores de $k = 30, 35, 40, 45$ y 50 ; se utilizó como criterio de selección del número de grupos, el número de grupos que separara de manera conveniente las instancias en D del resto, es decir, cuando los grupos que reunieran las instancias que no se pudieron resolver contuvieran el menor número de instancias resueltas por los algoritmos probados en este trabajo.

3.5. Evaluación de resultados

Extracción de patrones

El mejor resultado se obtuvo al generar 45 grupos con k-means utilizando 5 métricas (avg, m, v, r, p) debido a que esta configuración permitió una buena separación de las instancias de D en 4 grupos denominados A, B, C y D los cuales se muestran a continuación: $A = \{\text{TEST0082}\}$; $B = \{\text{TEST0014, TEST0030, TEST0058, TEST0005, TEST0022, TEST0065, TEST0084}\}$; $C = \{\text{BPP60, BPP832, BPP13, BPP709, BPP785, BPP144, BPP561, BPP781, BPP900, BPP195, BPP14, BPP119}\}$ y $D = \{\text{BPP40, BPP645, BPP766, BPP181, BPP485, BPP419, BPP178, BPP360, BPP47, BPP640, BPP814, BPP742, BPP531, BPP359, BPP716, BPP175}\}$. Los grupos B, C y D contienen algunas instancias en letra cursiva que fueron resueltas previamente por los algoritmos utilizados.

En particular, los grupos A y B contienen 5 instancias del conjunto gau_1 que no fueron resueltas de forma óptima por los algoritmos de prueba y los grupos C y D contienen las 17 instancias no resueltas de forma óptima del conjunto $Hard28$. Los patrones correspondientes a

Tabla 1. Centroides de grupos A, B, C y D

Grupo	<i>avg</i>	<i>m</i>	<i>v</i>	<i>r</i>	<i>p</i>
A	(0.280	0.01	0.06	0.73	.900)
B	(0.231	0.006	0.05	0.48	.890)
C	(0.370	0.008	0.018	0.691	.183)
D	(0.398	0.008	0.014	0.788	.180)

cada uno de los grupos se muestran en la Tabla 1.

Extracción de conocimiento

Dados los patrones mostrados en la Tabla 1 se puede observar que la instancia en el grupo A contiene objetos en promedio del 28% del tamaño del contenedor c , el objeto menor es del 1% de c , en promedio los objetos se repiten el 6%, el rango entre los tamaños de los objetos es del 73% de c , lo que significa que hay objetos muy pequeños y muy grandes y además la instancia es pequeña, sólo 90 objetos. En el caso del grupo B el promedio del tamaño de las instancias es de 23% respecto de c , el objeto pequeño es del 0.6% respecto de c , sólo el 5% de los objetos se repiten, el rango de los tamaños de los objetos es del 48% respecto de c y la instancia también es pequeña, 89 objetos. El grupo C muestra que el promedio de los tamaños de los objetos es del 37% respecto de c , el objeto menor representa el 0.8% de c , sólo se repite el 1.8% de los objetos, el rango entre el tamaño de los objetos es del 69% respecto de c y la

instancia es pequeña, 183 objetos. El grupo D muestra que en promedio el tamaño de los objetos es del 39.8% respecto de c , el objeto menor utiliza el 0.8% de c , sólo el 14% de los pesos de los objetos se repiten, el rango de los pesos de objetos es del 78.8% lo cual indica que hay un rango amplio en el peso de los objetos y las instancias también son pequeñas, 180 objetos en promedio.

De acuerdo a las características antes mencionadas y dado que el grupo D contiene el mayor número de instancias difíciles de resolver, se determinó que a manera de pre procesamiento de la instancia, era posible que algunos contenedores se pudieran llenar totalmente o a un alto porcentaje de su capacidad utilizando 2 objetos. Lo anterior con el objetivo de reducir el tamaño de la instancia y el espacio de búsqueda del algoritmo metaheurístico que se le aplicará.

3.6. Despliegue de resultados: Incorporación del conocimiento adquirido para la mejora de algoritmos

Con la finalidad de aplicar el conocimiento adquirido sobre llenar contenedores en un alto porcentaje utilizando dos objetos, se implementó un procedimiento de reducción denominado Γ que fue incorporado a una metaheurística como parte del preprocesamiento de la instancia, dicho procedimiento es llamado de forma iterativa mientras no se alcance el valor óptimo o mientras la capacidad residual de la instancia lo permita. El objetivo de aplicar el procedimiento fue reducir el espacio de búsqueda del algoritmo metaheurístico.

La Figura 2 muestra cómo se aplica el enfoque de solución, donde la parte superior de la figura corresponde a la instancia $I = \{w_1, w_2, \dots, w_n\}$ que será resuelta, 1) representa la ejecución del procedimiento de reducción sobre I de lo cual se obtiene la solución parcial s_1 y una nueva instancia reducida $I'' = I \setminus s_1$, 2) corresponde a solución parcial s_2 , producto de aplicar el algoritmo metaheurístico; 3) corresponde a la solución final $S = s_1 \cup s_2$. Si el número de contenedores obtenidos por S no corresponde al límite teórico preestablecido (L_2 propuesto por Martello y Toth en [30]), se vuelve

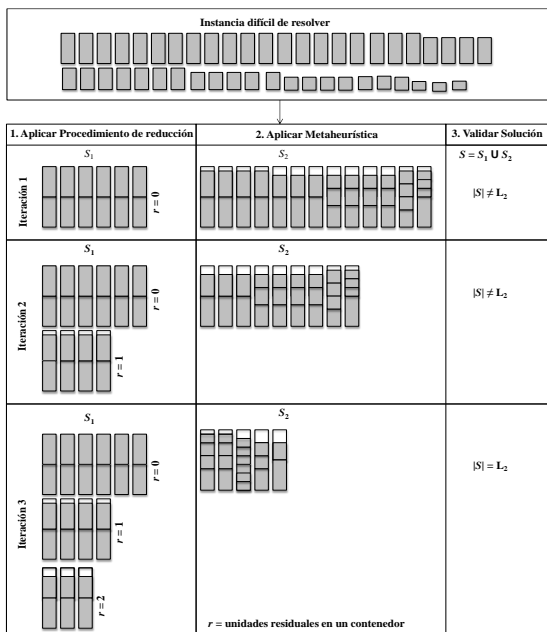


Fig. 2. Enfoque de solución para instancias difíciles de Bin Packing de una dimensión

a repetir el procedimiento desde 1) actualizando $I = I''$ y aumentando la capacidad residual del contenedor en una unidad.

Procedimiento de reducción Γ

De acuerdo a las características de las instancias en el grupo D , el mecanismo para reducir instancias consistió en llenar los contenedores con capacidad c usando un par de objetos (w_a and w_b) tales que $w_a + w_b = (c - umbral)$, donde $a \neq b$. La variable *umbral* es un parámetro adaptable, en esta experimentación, fue inicializado en 0 e incrementado en una unidad por iteración, el *umbral* desempeña la labor de administrador de la capacidad residual que le es permitida a un contenedor utilizando sólo dos objetos en una iteración dada. Básicamente encuentra todos los pares de objetos que llenen la capacidad c del contenedor tanto como sea posible con el objetivo de reducir el espacio de búsqueda de los algoritmos de solución del problema de *Bin Packing*.

El procedimiento de reducción Γ está presentado por el Algoritmo 2, el cual recibe como parámetros de entrada la instancia I con los objetos ordenados de forma decreciente, el valor de la variable *umbral* inicializada en 0 en la primera iteración del algoritmo que llamará a Γ , la capacidad del contenedor c y el número de contenedores a los que se les asignan objetos en la iteración actual $nBin$. Para ubicar los pares de objetos, se colocan dos apuntadores, uno apunta al objeto con el peso mayor de la instancia (*max*) y el otro apunta al objeto con el peso menor de la instancia (*min*) (líneas 1 y 2). Mientras existan objetos que no han sido evaluados en la instancia y mientras los índices de los apuntadores sean diferentes, el proceso de combinar objetos se repite (línea 3). Si la suma de los pesos de los objetos del conjunto par actuales igual a $c - umbral$, el par de objetos es extraído de I y los objetos son asignados al contenedor $nBin$ en la solución y agregados a la instancia I' (líneas 4, 5, 6 y 7). En caso contrario, si la suma es menor que $c - umbral$, el apuntador *min* apunta al siguiente objeto menor más cercano (líneas 8 y 9). Para el caso del apuntador mayor (*max*), si la suma es mayor que $c - umbral$ el apuntador *max* cambia de posición y apunta al siguiente objeto

Algoritmo 2: Procedimiento de reducción Γ

Entrada: $I = \{w_1, w_2, \dots, w_n\}; w_1 \geq w_2 \geq \dots \geq w_n, umbral, I', c, nBin$

```

1:  $max \leftarrow 1$ 
2:  $min \leftarrow n$ 
3: mientras ( $(w_{max} \neq \text{NULL})$  and ( $max \neq min$ ))
   hacer
4:   si ( $w_{max} + w_{min} = c - umbral$ ) entonces
5:      $nBin \leftarrow nBin + 1$ 
6:      $B_{nBin} \leftarrow \{w_{max}, w_{min}\}$ 
7:      $I' \leftarrow I' \cup B_{nBin}$ 
8:   sino, si ( $w_{max} + w_{min} < c - umbral$ )
     entonces
9:      $min \leftarrow min - 1$ 
10:  sino, si ( $w_{max} + w_{min} > c - umbral$ )
    entonces
11:     $max \leftarrow max + 1$ 
12:  fin si
13: fin mientras
14: return  $I'$ 

```

menor (líneas 10 y 11). Al terminar el procedimiento de reducción, la variable $nBin$ contiene el número de contenedores a los cuales se les asignaron objetos y regresa el subconjunto I' que corresponde a una solución parcial S_1 de la instancia I (línea 14).

El procedimiento fue implementado de tal forma que evita iterar más de lo necesario ya que cada par de objetos es evaluado como máximo una vez.

4. Experimentación computacional

En esta sección se muestran los resultados obtenidos al aplicar el procedimiento de reducción propuesto en este trabajo, producto de la caracterización de las instancias difíciles de resolver. El enfoque utilizado consistió en aplicar Γ seguido de una metaheurística para solucionar instancias de Bin Packing de una dimensión. El enfoque se aplicó sobre el grupo de instancias D , el cual contiene 7 instancias difíciles de resolverse (BPP40, BPP645, BPP766, BPP181, BPP485, BPP419, BPP178). Para la realización de los experimentos computacionales el procedimiento de reducción y la metaheurística de solución para *Bin Packing* de una dimensión fueron implementadas en lenguaje C. Se utilizó un equipo con procesador Intel core i7, 2GHz,

4GB en RAM, 32 bits y el sistema operativo Windows 7.

La metaheurística utilizada durante la experimentación fue inspirada en el algoritmo de solución HGGA-BP [33], en lo sucesivo nos referiremos a ella como H. El enfoque de solución de instancias consistió en aplicar de forma alternativa el procedimiento Γ y la metaheurística H. Después de aplicar el enfoque de solución sobre las instancias del grupo *D* fue posible obtener los valores óptimos alcanzados por los algoritmos de prueba y adicionalmente se encontraron las soluciones óptimas para las instancias BPP181, BPP485 y BPP178, dichas soluciones no han sido reportadas en la literatura por ninguna de las metaheurísticas conocidas. El tiempo en encontrar dichos valores fue de 15.53, 6.98 y 6.34 segundos respectivamente.

La Tabla 2 presenta una comparativa entre las soluciones obtenidas por los mejores algoritmos metaheurísticos reportados en la literatura (HI_BP [4], Pert-SAWMBS [16] y HGGA-BP [11]) y los resultados obtenidos después de aplicar nuestro enfoque. Los resultados corresponden a las instancias del grupo *D*. En la Tabla 2 la columna *Inst* corresponde al nombre de la instancia, la columna *opt* corresponde al valor óptimo conocido y las columnas *HI_BP*, *Pert*, *HGGA-BP* y *EsteTrabajo* corresponden a cada uno de los algoritmos que se comparan. En la columna *EsteTrabajo* los asteriscos (*) indican las instancias para las cuales sólo nuestro algoritmo alcanzó los valores óptimos (remarcado en gris) y el signo positivo (+) sirve de identificador para aquellas instancias que más de un algoritmo alcanza los valores óptimos.

Las soluciones óptimas encontradas son presentadas a continuación. Por cada instancia se muestra el nombre de la misma, separado por un guión, el número de contenedores utilizados en la solución, el valor hasta el cual llegó la variable (*umbral*) y el tiempo consumido en obtener la solución. Posteriormente se muestran los contenedores que fueron llenados al utilizar el procedimiento de reducción Γ y los contenedores utilizados al aplicar el algoritmo metaheurístico H.

Solución de la instancia BPP178 - 80, *umbral* = 1, *tiempo* = 6.34 seg.

Procedimiento de reducción Γ :

(790 210) (786 214) (743 257) (724 276) (646 354) (640360) (637 363) (635 365) (629 371) (613 387) (605 395) (541 459) (531 469) (518 482) (515 485) (506 494) (724275) (678 321) (662 337) (644 355) (583 416) (580 419)(568 431) (523 476) (512 487)

Algoritmo metaheurístico H:

(723 174 103) (704 292 4) (691 168 141) (189 709 102)(223 673 104) (423 518 59) (164 278 558) (519 408 72 1)(699 240 61) (436 157 407) (667 328) (116 184 700) (147347 506) (793 146 61) (732 266) (213 179 608) (602 396)(196 608 196) (504 446 50) (658 93 249) (119 415 466)(595 400) (212 443 345) (736 220 44) (94 170 736) (518232 250) (772 226) (756 242) (471 468 61) (628 370) (45278 677) (713 221 66) (662 336) (764 233) (535 462) (740209 51) (692 185 123) (133 250 617) (712 233 55) (517447 36) (185 21 794) (517 171 312) (703 181 116) (659334 7) (622 369 9) (572 423) (742 254) (509 487) (633 246120) (612 346 41) (740 252) (691 208 99) (625 370) (595399) (521 313 166)

Solución de la instancia BPP485 - 71, *umbral* = 1, *tiempo* = 6.89 seg.

Procedimiento de reducción Γ :

(800 200) (782 218) (756 244) (751 249) (747 253) (742258) (713 287) (693 307) (645 355) (621 379) (600 400) (594 406) (586 414) (584 416) (579 421) (577 423) (548452) (777 222) (759 240) (717 282) (694 305) (661 338)(657 342) (654 345) (607 392)

Algoritmo metaheurístico H:

(540 14 446) (569 28 403) (570 54 376) (784 188 28) (42256 702) (138 276 586) (627 316 55) (335 574 91) (780117 103) (765 226 9) (688 263 47) (59 654 286 1) (758240) (633 10 357) (126 631 243) (559 317 124) (581 237182) (127 606 267) (351 644 5) (159 782 59) (505 482 13)(790 208) (612 380 8) (77 272 651) (116 765 119) (574277 149) (128 132 740) (788 141 70) (705 58 237) (103264 633) (639 359) (77 224 697) (405 405 190) (17 444539) (161 444 395) (752 245) (675 322) (541 455) (529467) (600 396) (796 197) (523 450 27) (90 404 506) (224706 68) (504 450 43) (445 289 266)

Solución de la instancia BPP181 - 72, *umbral* = 4, *tiempo* = 15.53 seg.

Procedimiento de reducción Γ :

Tabla 2. Instancias Hard28

Inst	opt	HI_BP	Pert	HGGA-BP	Este Trabajo
hBPP40	59	60	60	60	60
hBPP360	62	63	63	62	62+
hBPP645	58	59	59	59	59
hBPP742	64	65	65	64	64+
hBPP766	62	63	63	63	63
hBPP47	71	72	72	71	71+
hBPP181	72	73	73	73	72*
hBPP359	76	76	76	76	76+
hBPP485	71	72	72	72	71*
hBPP640	74	75	75	74	74+
hBPP716	76	76	76	76	76+
hBPP175	84	84	84	84	84+
hBPP178	80	81	81	81	80*
hBPP419	80	81	81	82	81
hBPP531	83	84	84	83	83+
hBPP814	81	82	82	81	81+

(799 201) (745 255) (733 267) (721 279) (713 287) (703 297) (673 327) (660 340) (648 352) (646 354) (603 397) (545 455) (532 468) (523 477) (797 202) (770 229) (651 348) (641 358) (623 376) (548 451) (503 496) (706 292) (702 296) (680 318) (597 401) (758 239) (684 313) (629 368) (610 387) (561 436) (525 472) (679 317) (603 393) (525 471)

Algoritmo metaheurístico H:

(32 659 309) (158 66 776) (699 278 23) (572 363 65) (45 372 583) (118 127 755) (51 750 199) (305 30 665) (39 407 554) (103 183 714) (144 755 101) (458 200 342) (214 120 666) (307 171 522) (276 109 615) (79 711 210) (120 128 752) (172 399 429) (697 281 22) (201 494 305) (661 276 63) (183 776 41) (391 197 412) (108 793 99) (328 215 457) (536 434 30) (457 278 265) (714 219 66) (531 432 36) (639 326 34) (51 372 577) (736 169 94) (610 307 82) (666 326 7) (620 257 123) (583 411) (572 407 20) (513 481)

5. Conclusiones y trabajos futuros

En este trabajo se muestra que la metodología propuesta para la caracterización de instancias

difíciles del problema de *Bin Packing* de una dimensión contribuye a la generación de estrategias para la mejora de los algoritmos de solución actuales.

Después de aplicar la metodología propuesta sobre un conjunto de 1,615 instancias y un conjunto inicial de 27 métricas para el problema de *Bin Packing*, se encontraron 5 métricas relevantes (promedio de los pesos de los objetos avg , peso del objeto menor m , frecuencia de repetición de los objetos v , rango entre el cual se encuentran los pesos de los objetos r y tamaño de la instancia p) que permitieron caracterizar las instancias difíciles de resolver por 6 algoritmos bien conocidos del estado del arte, dos de ellos considerados de los más robustos. Con las 5 métricas relevantes fue posible identificar 4 patrones correspondientes a las instancias difíciles de resolver. Las características de uno de los grupos sugirieron la posibilidad de desarrollar un nuevo procedimiento de reducción del tamaño de las instancias al cual denominamos Γ . Al aplicar el procedimiento de reducción Γ con un algoritmo metaheurístico H sobre uno de los grupos con instancias difíciles de resolver fue

posible encontrar las soluciones óptimas previamente reportadas y 3 adicionales que no han sido reportadas por ninguno de los algoritmos metaheurísticos conocidos.

Consideramos que el enfoque utilizado para la caracterización de instancias, también puede ser útil en la caracterización de instancias difíciles de otros problemas NP. Como trabajo futuro se propone realizar un análisis más profundo sobre las características específicas de los patrones encontrados para los grupos de instancias *C* y *D*, con el objetivo de generar estrategias que permitan encontrar los valores óptimos para las instancias cuyo valor óptimo no se alcanzó.

Referencias

1. Afza, N., Zulkipli, F., Sarah, S., & Radiah, S. (2014). An Alternative Heuristics for Bin Packing Problem. *Proceedings of the 2014 International Conference on Industrial Engineering and Operations Management*, Bali, Indonesia, January 7–9.
2. Álvarez, V. (2006). *Modelo para representar la complejidad del problema y el desempeño de algoritmos*. Tesis de maestría, Instituto Tecnológico de Cd. Madero, México.
3. Alvim, A. (2003). *Uma heurística híbrida de melhoria para o problema de Bin Packing e sua aplicação ao problema de escalonamento de tarefas*. Tesis de doctorado, Pontificia Universidad Católica de Rio de Janeiro, Brasil.
4. Alvim, A., Glover, F., Ribeiro, C., & Aloise, D. (2004). A hybrid improvement heuristic for the one-dimensional Bin Packing problem. *Journal of Heuristics*, Vol. 10, pp. 205–229. DOI: 10.1023/B:HEUR.0000026267.44673.ed
5. Beasley, J. (2015). The operational research library, <http://people.brunel.ac.uk/~mastijb/jeb/orlib/binpac/kinfo.html>, última visita: enero de 2015.
6. Belov, G. (2004). *Problems, models and algorithms in one and two dimensional cutting*. Tesis de doctorado, Fakultät Mathematik und Naturwissenschaftender Technischen Universität Dresden, Alemania.
7. Bhatia, A. & Basu, S. (2004). Packing bins using multi-chromosomal genetic representation and better-fit heuristic. *Neural Information in Processing*, Springer, pp. 181–186. DOI: 10.1007/978-3-540-30499-9_26
8. Chapman, P., Clinton, J., Kerber, R., Khabaza, T., Reinartz, T., Shearer, C., & Wirth, R. (2000). *Cross industry standard process for datamining version 1.0 step by step datamining guide*, <ftp://ftp.software.ibm.com/software/analytics/spss/support/Modeler/Documentation/14/UserManual/CRISPDM.pdf>, última visita: enero de 2015.
9. Cruz, L. (2004). *Clasificación de algoritmos heurísticos para la solución de problemas de BinPacking*. Tesis de doctorado, Centro Nacional de Investigación y Desarrollo Tecnológico, México.
10. Cruz, L., Nieto, Y., Tomás, S., & Castilla, V. (2007). Solving Bin Packing problem with a hybridization of hard computing and soft computing. *Innovations in Hybrid Intelligent Systems*, Springer, pp. 223–230. DOI:10.1007/978-3-540-74972-1_30
11. Cruz, L., Quiroz, M., Alvim, A., Fraire, H. Gómez, S., & Torres, J. (2012). Heurísticas de agrupación híbridas eficientes para el problema de empaqueo de objetos en contenedores. *Computación y Sistemas*, Vol. 16, No. 3, pp. 349–360.
12. Ducatelle, F. & Levine, J. (2004). Ant colony optimization and local search for Bin Packing and Cutting Stock problems. *Journal of the Operational Research Society*, Vol. 55, pp. 705–716. DOI:10.1109/ICSPC.2007.4728416
13. Dresden University. (2015). *Cutting and packing*, http://www.math.tudresden.de_capad/cpd-ti.html#pmp, última visita: enero de 2015.
14. ESICUP. (2015). *Euro especial interest group on cutting and packing (esicup)*, http://paginas.fe.up.pt/_esicup/tikilist_file_gallery.php?galleryId=1, última visita: enero de 2015.
15. Falkenauer, E. (1999). A hybrid grouping genetic algorithm for Bin Packing. *Journal of Heuristics*, Vol. 2, pp. 5–30. DOI: 10.1007/BF00226291
16. Fleszar, K. & Charalambous, C. (2011). Average-weight-controlled bin oriented heuristics for the one-dimensional Bin-Packing problem. *Discrete Optimization*, Vol. 210, pp. 176–184. DOI:10.1016/j.ejor.2010.11.004
17. Fleszar, K. & Hindi, K. (2002). New heuristics for one-dimensional Bin-Packing. *Computers & Operations Research*, Vol. 29, pp. 821–839. DOI:10.1016/S0305-0548(00)00082-4
18. Garey, M.R., & Johnson, D.S. (1979). *Computers and Intractability: a Guide to the Theory of NP-Completeness*. W. H. Freeman and Company, New York.
19. Gómez, M. & Randall, M. (2009). A hybrid extremal optimization approach for the Bin Packing

- problem. *Artificial Intelligence*, Springer, pp. 242–251. DOI:10.1007/978-3-642-10427-5_24
20. **Hoos, H. Smyth, K., & Stutzle, T. (2004).** Search space features underlying the performance of stochastic local search algorithms for MAX-SAT. *Parallel Problem Solving from Nature*, Springer, pp. 51–60. DOI:10.1007/978-3-540-30217-9_6
 21. **Jing, X., Zhou, X., & Xu, Y. (2006).** A hybrid genetic algorithm for Bin Packing problem based on item sequencing. *Journal of Information and Computing Science*, Vol. 1, pp. 61–64.
 22. **Johnson, D. (1973).** *Near-optimal Bin Packing algorithms*. PhD Thesis, Massachusetts Institute of Technology, Massachusetts.
 23. **Kamali, S. & López, A. (2015).** Efficient Online Strategies for Renting Servers in the Cloud. *SOFSEM 2015, LNCS*, Vol. 8939, Springer, pp. 277–288. DOI:10.1007/978-3-662-46078-8_23
 24. **Kacprzak, Ł., Rudy, J., & Żelazny, D. (2015).** Multicriteria 3-dimension Bin packing Problem. *Research in logistics and production*, Vol. 5, pp. 85–94.
 25. **Kasap, N. & Agarwal, A. (2004).** Augmented-neural networks approach for the Bin Packing problem, *Proceedings of the 4th International Symposium on Intelligent Manufacturing Systems*, pp. 348–358.
 26. **Layeb, A. & Chenche, S. (2012).** A Novel GRASP Algorithm for Solving the Bin Packing Problem. *I.J. Information Engineering and Electronic Business*, Vol. 2, pp. 8–14.
 27. **Loh, K., Golden, B., & Wasil, E. (2008).** Solving the one-dimensional Bin Packing problem with a weight annealing heuristic, *Computers & Operations Research*, Vol. 35, No. 7, pp. 2283–2291. DOI:10.1109/ICSPC.2007.4728416
 28. **López, E., Terashima, H., Ross, P., Ochoa, G. (2014).** A unified hyper-heuristic framework for solving bin packing problems. *Expert Systems with Applications*, Vol. 41, pp. 6876–6868. DOI:10.1016/j.eswa.2014.04.043 89
 29. **MacQueen, J. (1967).** Some methods for classification and analysis of multivariate observations. *Proceedings of the fifteenth Berkeley Symposium on Mathematical Statistics and Probability*, pp. 281–298.
 30. **Martello, S. & Toth, P. (1990).** Lower bounds and reduction procedures for the Bin Packing problem. *Discrete Applied Mathematics*, Vol. 28, pp. 59–70. DOI:10.1016/0166-218X(90)90094-S
 31. **Merz, P. & Freisleben, B. (2000).** Fitness landscapes, memetic algorithms and greedy operators for graph bi-partitioning. *Evolutionary Computation*, Vol. 8, pp. 61–91. DOI:10.1162/106365600568103
 32. **Nieto, D. (2007).** *Hibridación de algoritmos metaheurísticos para problemas de Bin Packing*. Tesis de Maestría, Instituto Tecnológico de Cd. Madero, México.
 33. **Quiroz, M. (2009).** *Caracterización de factores de desempeño de algoritmos de solución de Bin Packing*. Tesis de maestría, Instituto Tecnológico de Cd. Madero, México.
 34. **Quiroz, M. Cruz, L., Torres, J., Gómez, C., Fraire, H., & Alvim A. (2015).** A grouping genetic algorithm with controlled gene transmission for the bin packing problem. *Computers & Operations Research*, Vol. 55, pp. 52–64.
 35. **Quiroz, M., Cruz, L., J. Torres, Gómez, C.G., Fraire, H.J., & Melin, P. (2013).** Improving the Performance of Heuristic Algorithms Based on Exploratory Data Analysis. *Recent Advances on Hybrid Intelligent Systems*, Vol. 451, pp. 361–375. DOI:10.1007/978-3-642-33021-6_29
 36. **Scholl, A., Klein, R., & Jürgens, C. (1997).** Bison: a fast hybrid procedure for exactly solving the one-dimensional Bin Packing problem. *Computers & Operations Research*, Vol. 24, No. 7, pp. 627–645. DOI:10.1016/S0305-0548(96)00082-2
 37. **Schwerin, P. & Wäscher, G. (1997).** The Bin-Packing problem: a problem generator and some numerical experiments with FFD packing and MTP. *International Transactions in Operational Research*, Vol. 4, No. 5-6, pp. 337–389. DOI: 10.1111/j.1475-3995.1997.tb00093.x
 38. **Sim, K., Hart, E., & Paechter, B. (2013).** Learning to Solve Bin Packing Problems with an Immune Inspired Hyper-heuristic. *Artificial Immune Systems – ICARIS*, pp. 856–863. DOI: 10.7551/978-0-262-31709-2-ch126
 39. **Singh, A. & Gupta, A. (2007).** Two heuristics for the one-dimensional Bin-Packing problem. *OR Spectrum*, Vol. 29, No. 4, pp. 765–781. DOI:10.1007/s00291-006-0071-2
 40. **Stawowy, A. (2008).** Evolutionary based heuristic for Bin Packing problem. *Computers & Industrial Engineering*, Vol. 55, No. 2, pp. 465–474. DOI:10.1016/j.cie.2008.01.007
 41. **Ülker, O., Korkmaz, E., & Özcan, E. (2008).** A grouping genetic algorithm using linear linkage encoding for Bin Packing. *Parallel Problem Solving from Nature*, Springer, pp. 1140–1149. DOI:10.1007/978-3-540-87700-4_113
 42. **Wolpert, D., & Macready, W. (1996).** No free lunch theorems for optimizations. *IEEE*

Transactions on Evolutionary Computation, Vol. 1, pp. 67–82. DOI:10.1109/4235.585893

Adriana Mexicano Santoyo recibió el grado de Doctora por el Centro Nacional de Investigación y Desarrollo Tecnológico. Actualmente es profesora en el Instituto Tecnológico de Cd. Victoria. Sus intereses de investigación incluyen análisis de algoritmos y optimización combinatoria entre otros.

Joaquín Pérez Ortega recibió el grado de Doctor por el Tecnológico de Monterrey. Es profesor investigador en el Centro Nacional de Investigación y Desarrollo Tecnológico. Actualmente pertenece al Sistema Nacional de Investigadores (SNI), es miembro Senior del IEEE y autor de más de 100 publicaciones en bases de datos, heurísticas y optimización.

Gerardo Reyes Salgado recibió el grado de Doctor por el Instituto Nacional Politécnico de Grenoble (Francia). Es profesor-investigador en el Centro Nacional de Investigación y Desarrollo Tecnológico. Autor de más de 60 publicaciones en temas de predicción y optimización con diferentes técnicas de *machine learning* e inteligencia artificial.

Nelva Nely Almanza Ortega recibió el grado de Maestra en Ciencias por el Centro Nacional de Investigación y Desarrollo Tecnológico. Actualmente estudia el Doctorado en la misma institución. Sus intereses de investigación incluyen Ingeniería de Software, Análisis de Algoritmos y Minería de Datos.

*Artículo recibido el 26/08/2013; aceptado el 21/04/2015.
Autor de correspondencia es Adriana Mexicano Santoyo.*