Modified Harmony Search for a Truck Loading Problem Application

Nicolás Anabalón Romero, and Matías Barros Vásquez, and Rosa Medina Durán

***Abstract*—We must decide how to load boxes in a truck, to avoid damaging during the transportation and simplifying the delivery to multiple clients. We propose a constructive heuristic and a modified Harmony Search metaheuristic, using the open dimension approach, the length in this case. The constructive heuristic, based on the Wall Building approach, provides a very fast initial solution for the metaheuristic. The selection algorithm, guiding the metaheuristic, considers the optimization of the multi-client features originally. The results show the effectiveness of the metaheuristic in searching solutions that improve the considered features, in reasonable times according to the truck's load planning.**

***Index Terms*—Packing, Truck Loading Problem, Harmony Search.**

I. Introducción

Consideremos una empresa que debe entregar sus productos en cajas a sus clientes. Una vez resuelto el problema de ruteo, se debe decidir cómo ubicar las cajas en el camión de reparto, de manera que la entrega de las cajas sea expedita, por lo cual las cajas de un mismo cliente deben estar cercanas entre sí, sin bloquear las cajas de otros clientes. Las cajas de los últimos clientes en la ruta, serán las primeras en cargarse en el camión. Por otra parte, al estar apiladas, las cajas podrían dañarse; por lo cual una caja más pesada no debería estar completamente sobre una de menor peso. Además, para que las cajas no se muevan al desplazarse el camión, se debe garantizar la estabilidad, es decir, es necesario que exista una superficie mínima de contacto entre cajas que se encuentren una sobre otra; que podría reforzarse mediante cuerdas o arnés, para evitar un derrumbe de columnas muy altas o cerca de la puerta del camión. Se busca una solución que ubique cada caja en el camión cumpliendo con estas características.

En la literatura, el problema se relaciona con los problemas de *3D-packing* [1]–[3], *Container Loading Problem* (CLP) [4], [5] y *Truck Loading Problem* [6]–[8], para los cuales existen muchos estudios que abordan las diferentes características, resolviendo los problemas de manera exacta o heurística. De acuerdo a la clasificación de [9], para CLP, el problema corresponde a un problema de dimensión abierta con carga débilmente heterogenea (*Open Dimension Problem with Weakly Heterogenous assortment of cargo, ODP/W*); con *(i)* restricciones de apilado o *stacking (load bearing)*: una caja más pesada no debería ir arriba de una más liviana; *(ii)* restricciones de entregas múltiples o *multi-drop*: grupos de cajas ubicadas juntas y de acuerdo al orden en que serán entregadas a los clientes; y *(iii)* restricciones de estabilidad vertical o estática (*vertical stability/static stability*): la base de la caja debe ser sostenida total o parcialmente, por el piso del contenedor o por las caras superiores de otras cajas.

Los trabajos con una dimensión abierta, se relacionan con el *Strip Packing problem* (SPP), el cual puede tener diferentes dimensiones. También en Respen y Zufferey [10], resuelven un 2DSPP, comparando los métodos GA, TS y ANT (*Ant Algorithm*) con un *Look Ahead Greedy* (LAG) usado por la empresa dueña de los datos utilizados, demostrando que una combinación de las 3 primeras metaheurísticas es capaz de obtener soluciones que tengan más objetos en un largo igual o menor que el encontrado por LAG, además se muestra como su alternativa es al menos 5 veces más rápida. En [11], se resuelve el 2DSPP a través de TS y GA creando cotas para encontrar el mínimo valor posible. En el trabajo de Bansal et al. [12], se aborda el SPP en tres dimensiones, para minimizar la altura de un contenedor, gracias a transformaciones de restricciones de un problema de *Bin Packing* bidimensional. También se encuentran heurísticas constructivas en tres dimensiones, como en [3] que va creando la solución por capas desde el fondo del contenedor.

Dadas las múltiples aplicaciones estudiadas en la literatura, existen múltiples restricciones a la distribución de la carga. Por ejemplo, en [13], resuelven un CLP donde cada contenedor tiene un límite de peso máximo, proponiendo modelos matemáticos para su resolución. La fragilidad, limita el máximo de objetos en un contenedor, en el trabajo de [14], resolviéndolo mediante un *Branch and Bound*. Especial énfasis se le da a la estabilidad de los objetos apilados en el trabajo de [13]. Alonso et al. [7] utilizan la metaheurística GRASP (*Greedy Randomized adaptive Search Procedure*) para encontrar la mejor disposición de los objetos en pallet y cargarlos en el camión. En [15] se utiliza un modelo matemático enfocándose en cumplir restricciones de distribución de peso, para un problema de carga de múltiples contenedores con pallets.

Dentro de los *Truck Loading Problem*, hay trabajos que consideran otras características como la rotación de objetos o la distribución del peso de la carga. Junqueira et al. [8], proponen un enfoque basado en un modelo matemático para el CLP con restricciones de entregas múltiples. En [6], los autores se enfocan en la distribución del peso de la carga en el camión para entregar a un cliente, se propone una heurística constructiva basada en capas, para lograr cargar el máximo posible de cajas.

Nicolás Anabalón Romero, Departamento de Ingeniería Industrial, Universidad de Concepción, Chile, e-mail:nianabalon@udec.cl

Matías Barros Vásquez, Departamento de Ingeniería Industrial, Universidad de Concepción, Chile e-mail:mabarros@udec.cl

Rosa Medina Durán, Instituto Sistemas Complejos de Ingeniería y Departamento de Ingeniería Industrial, Universidad de Concepción, Chile e-mail:rosmedina@udec.cl

Podemos observar que, al tratarse de un problema NP-Hard, las heurísticas proveen soluciones más rápidas para las aplicaciones reales. Por otra parte, considerando los distintos tipos de problemas, aunque se realizan varias comparaciones entre métodos, al considerar que siempre las restricciones estudiadas varían, no es posible indicar cual es el mejor método para resolver este problema.

La metaheurística Harmony Search (HS) [16] se basa en la improvisación musical, la cual mejora con cada práctica acercándose a la armonía perfecta. Dada una melodía, en una práctica se puede mantener la misma melodía, modificar parte de esta o modificar completamente, de acuerdo a la descripción de [17]. En el caso de los problemas de optimización, cada melodía corresponde a una solución y en las iteraciones se puede mantener la solución, modificar parcialmente o generar una aleatoriamente. Para ello la metaheurística define los parámetros r_{accept} y r_{pa}. Dado un número aleatorio $rand$, si $rand > r_{accept}$, la solución se mantiene. Si $r_{pa} \leq rand < r_{accept}$, la solución se modifica parcialmente. En otro caso, se genera una nueva solución de manera aleatoria. Para una adecuada exploración de las soluciones, [17] menciona que $r_{accept} = 0{,}7 \sim 0{,}95$ y $r_{pa} = 0{,}1 \sim 0{,}5$. Los trabajos de [18] y [19] revisan la teoría y las diferentes aplicaciones resueltas por esta metaheurística, observándose muchas modificaciones a la versión original para mejorar el desempeño del algoritmo en las diferentes aplicaciones. En [20] se resuelve un *Container Storage Problem* en tres dimensiones, modificando HS para minimizar el remanejo de contenedores en un puerto.

De acuerdo a nuestros conocimientos, la metaheurística HS no ha sido utilizada para resolver *Truck Loading Problems*. Las muchas aplicaciones resueltas por HS, muestran que es muy versátil en su estructura, adaptándose a los requerimientos de los problemas. Además, cuenta con pocos parámetros para su funcionamiento, facilitando su correcta selección. En el presente trabajo se propone una adaptación de HS para el problema descrito anteriormente, para mejorar la solución encontrada por una heurística constructiva, en términos de utilización del largo del camión y las características de entregas múltiples. El algoritmo de selección para guiar la búsqueda de buenas soluciones, utilizado en la metaheurística, considera de manera novedosa la optimización de las características de entregas múltiples del problema.

El artículo se organiza de la siguiente forma: en la Sección II Metodología, se describe cómo se abordan cada una de las características de la solución buscada, la heurística constructiva propuesta y cómo se adapta la metaheurística para mejorar esta solución. En la Sección III, se presentan las instancias utilizadas para la calibración de parámetros, los experimentos y los resultados. Finalmente, en la Sección IV se presentan las principales conclusiones del estudio.

II. Metodología

Para modelar el problema se considera un camión de dimensiones largo, ancho y alto, L, W y H, respectivamente. El origen del eje de coordenadas se ubica en la esquina inferior-izquierda-delantera del camión, como se muestra en la Fig. 1. Notar que la parte delantera del camión corresponde a la parte trasera de la zona de carga, por lo cual, para la carga, se considera "adelante" la zona más cercana a la puerta y "atrás" la zona más cercana al origen de las coordenadas.

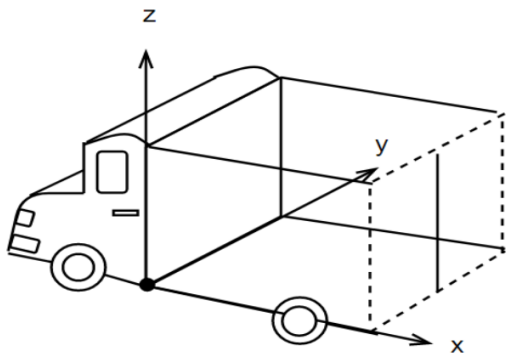


Fig. 1. Camión y ejes de coordenadas.

Se considera que existen n cajas a ubicar en el camión, cuyas caras son paralelas a los lados del camión y no pueden ser rotadas. Para cada caja i, se conocen sus dimensiones largo l_i, ancho w_i, alto h_i, y su peso p_i. Además, existen N clientes ordenados de acuerdo a la ruta de entrega. Se define Δ_j al conjunto de cajas del cliente j, en el cual se encuentran las cajas ordenadas por peso, en forma descendente.

Una solución debe determinar, para cada caja, la posición de la esquina inferior-izquierda-trasera; de manera que se respeten las restricciones físicas de factibilidad (contención, sobreposición y apoyo), además de las restricciones de apilado, estabilidad y de entregas múltiples.

Se utiliza el enfoque de ODP/W [9], mencionado en la Introducción, considerando minimizar el largo utilizado del camión. Es decir, inicialmente se considera un camión con ancho W y alto H, fijo; y de largo L, igual a la suma de los largos de todas las cajas. El algoritmo de selección busca, principalmente, minimizar el largo utilizado; pero ante igual largo, es preferible una solución donde las cajas de un mismo cliente se encuentren más cercanas y se respete el orden de entrega.

A continuación se describe como se abordan cada tipo de restricción, la heurística constructiva y la adaptación de la metaheurística utilizada.

A. Restricciones

A1. Restricciones Físicas: Para verificar que las cajas se encuentren completamente contenidas en el camión, por cada caja i, se debe respetar que:

$$0 \leq x_i < x_i + l_i \leq L \tag{1}$$

$$0 \leq y_i < y_i + w_i \leq W \tag{2}$$

$$0 \leq z_i < z_i + h_i \leq H \tag{3}$$

donde (x_i, y_i, z_i) es la posición de la caja i.

La sobreposición de las cajas verifica que las cajas no queden dentro de otras, creando una solución infactible. Dadas dos cajas i y u; por cada eje, se verifican las siguientes condiciones ejemplificadas para el eje X:

$$x_i \leq x_u \leq x_u + l_u \leq x_i + l_i \quad (4)$$
$$x_u \leq x_i \leq x_i + l_i \leq x_u + l_u \quad (5)$$
$$x_i \leq x_u < x_i + l_i \leq x_u + l_u \quad (6)$$
$$x_u \leq x_i < x_u + l_u \leq x_i + l_i \quad (7)$$

Si para cada eje, al menos una de las condiciones anteriores se cumple, entonces las cajas i y u se consideran sobrepuestas y se mueve una de las cajas hacia una nueva posición factible, mediante la función $mover(i)$,ver Algoritmo 2, explicado en la Sección A3.

También, para abordar el apoyo y estabilidad de la carga, se verifica que la cara inferior de las cajas se encuentre al menos en un 50 % en contacto con otra superficie, que puede ser el plano XY (base del camión) o superficies de otras cajas, ver Algoritmo 1, líneas 3 a 5. La función $contactoXY(i, u)$ considera las cajas i y u, y calcula la intersección de las superficies de las cajas en el plano XY:

$$\begin{aligned}(min\{x_i + l_i, x_u + l_u\} - max\{x_i, x_u\}) \times \\ (min\{y_i + w_i, y_u + w_u\} - max\{y_i, y_u\})\end{aligned} \quad (8)$$

A2. Restricciones de Apilado: Para verificar el apilado de las cajas, se considera el peso p_i de las cajas y su posición, de manera que una caja de mayor peso no debería estar sobre otra de menor peso. Se revisan los pares de cajas que se encuentren en contacto en sus caras superiores e/o inferiores. Si la caja que está arriba tiene mayor peso que la que se encuentra abajo y el área en contacto es más del 50 % de la caja de abajo; entonces, se mueve la caja de arriba a una posición factible, ver Algoritmo 1, líneas 7 a 9.

Algoritmo 1 Apoyo entre cajas y restricciones de apilado

1: **if** $z_i + h_i = z_u$ & ($x_i \leq x_u \leq x_u + l_u \leq x_i + l_i$ ‖ $x_u \leq x_i \leq x_i + l_i \leq x_u + l_u$ ‖ $x_i \leq x_u < x_i + l_i \leq x_u + l_u$ ‖ $x_u \leq x_i < x_u + l_u \leq x_i + l_i$) & ($y_i \leq y_u \leq y_u + w_u \leq y_i + w_i$ ‖ $y_u \leq y_i \leq y_i + w_i \leq y_u + w_u$ ‖ $y_i \leq y_u < y_i + w_i \leq y_u + w_u$ ‖ $y_u \leq y_i < y_u + w_u \leq y_i + w_i$) **then**
2: **if** $p_i < p_u$ **then**
3: **if** $contactoXY(i, u) > 0,5 \times (l_i \cdot w_i)$ **then**
4: $mover(u)$
5: **end if**
6: **else**
7: **if** $contactoXY(i, u) < 0,5 \times (l_u \cdot w_u)$ **then**
8: $mover(u)$
9: **end if**
10: **end if**
11: **end if**

A3. Restricciones de Entregas Múltiples: Las restricciones de entregas múltiples consideran la proximidad de las cajas de un mismo cliente y el orden de las entregas.

Para que las cajas de un mismo cliente se mantengan cercanas, se busca disminuir la distancia entre ellas. Por cada caja, se revisa que alguna de las cajas adyacentes sea del mismo cliente, asegurando pares de cajas de un mismo cliente. Para considerar dos cajas adyacentes, se verifica que al menos en un eje coincidan sus dimensiones iniciales y finales. Si no existe ninguna caja adyacente del mismo cliente, se mueve la caja intentando ubicarla junto a una caja del mismo cliente. Si esto no es posible, se intenta junto a cajas del cliente anterior o posterior y, si lo anterior no es posible, se ubican de manera aleatoria pero cumpliendo las restricciones físicas, ver Algoritmo 2. Este movimiento de cajas se utiliza cada vez que se requiere encontrar una nueva posición para una caja.

El Algoritmo 2 intenta poner la caja i sobre una caja u que sea del mismo cliente, líneas 2-26. Primero controla que sea factible por altura y ancho, con las condiciones de la línea 5. En el ciclo for de las líneas 6-10, verifica si existe alguna otra caja k que se encuentre sobre u, que impediría poner a i. En el ciclo for de las líneas 12-16, verifica si existe otra caja k que podría quedar traslapada con i al ubicarla sobre u. Si estas condiciones anteriores no se cumplen, se ubica la caja i sobre u en la línea 22. De lo contrario, en forma análoga a las verificaciones para moverla arriba de u, se busca mover i a la derecha (línea 28), izquierda (línea 31), adelante (línea 34) o atrás (línea 37) de otra caja del mismo cliente. Si no es posible, se ubica de manera aleatoria relajando la restricción de proximidad a cajas del mismo cliente, pero respetando las restricciones de factibilidad.

Las restricciones de orden de entrega se abordan en el algoritmo de selección que guía la búsqueda de soluciones, explicado en la Sección C1.

B. Heurística Constructiva

Al construir una solución se sigue un enfoque determinista de *wall building*. Dado que las cajas se encuentran ordenadas por orden de entrega y, para cada cliente, por peso descendente; este enfoque permite generar una solución factible para el problema al respetar las restricciones físicas, de empilado y estabilidad, dejando las cajas de un mismo cliente cercanas, respetando las restricciones de entregas múltiples; pero con un largo significativo.

Inicialmente, se considera el camión vacío. Mientras las cajas sean del mismo cliente, la siguiente caja en la lista se coloca en la posición más atrás, a la izquierda y abajo posible. Si corresponde a un nuevo cliente, se inicia una nueva fila, en la base del camión y las cajas se ponen arriba mientras no se sobrepase la altura del camión. Si se sobrepasa la altura, la siguiente caja se ubica a la derecha de la columna anterior, inicializando una nueva columna, mientras no se sobrepase el ancho del camión. Si se sobrepasa el ancho, se inicializa una nueva fila de cajas. Esto se repite, mientras existan cajas sin cargar en el camión.

C. Adaptación de Harmony Search

Se utiliza una modificación de la metaheurística *Harmony Search*, ver Algoritmo 3, para mejorar la solución encontrada por la heurística constructiva, en términos de largo utilizado del camión y restricciones de entregas múltiples.

Para obtener un mejor desempeño, se modifica *Harmony Search* limitando el uso de la aleatoriedad en el posicionamiento de las cajas, ya que empíricamente se observa que esto perjudica la búsqueda de soluciones factibles. Primero, se cambia el enfoque de población por el de solución única,

Algoritmo 2 Mover caja *mover(i)*

```
j=0
for u ∈ N\{i} do
  if N_i = N_u then
    j=0
    if (z_u + h_u + h_i ≤ H) & (y_u + w_i ≤ W) then
      for k ∈ N\{i,u} do
        if z_u + h_u = z_k & contactoXY(u,k) then
          j=1, break
        end if
      end for
      if j=0 then
        for k ∈ N\{i,u} do
          if ( x_u ≤ x_k ≤ x_k + l_k ≤ x_u + l_i || x_k ≤ x_u ≤ x_u+l_i ≤ x_k+l_k || x_u ≤ x_k < x_u+l_i ≤ x_k + l_k || x_k ≤ x_u < x_k + l_k ≤ x_u + l_i ) & ( y_u ≤ y_k ≤ y_k + w_k ≤ y_u + w_i || y_k ≤ y_u ≤ y_u + w_i ≤ y_k + w_k || y_u ≤ y_k < y_u + w_i ≤ y_k + w_k || y_k ≤ y_u < y_k + w_k ≤ y_u + w_i) & ( z_u + h_u ≤ z_k ≤ z_k + h_k ≤ z_u + h_u + h_i || z_k ≤ z_u + h_u ≤ z_u + h_u + h_i ≤ z_k + h_k || z_u + h_u ≤ z_k < z_u + h_u + h_i ≤ z_k + h_k || z_k ≤ z_u + h_u < z_k + h_k ≤ z_u + h_u + h_i ) then
            j=1, break
          end if
        end for
      end if
    else
      j=1, break
    end if
    if j=0 then
      x_i = x_u, y_i = y_u, z_i = z_u + h_u
      j=2, break
    end if
  end if
end for
if j!=2 then
  j = mover.derecha(i)
end if
if j!=2 then
  j = mover.izquierda(i)
end if
if j!=2 then
  j = mover.adelante(i)
end if
if j!=2 then
  j = mover.atras(i)
end if
if j!=2 then
  for u ∈ N\{u} do
    mover.aleatorio(i)
  end for
end if
```

para lo cual se considera una única solución inicial, obtenida mediante la heurística constructiva, que corresponde a una solución factible para el problema.

Segundo, en la etapa de la modificación parcial de la solución ($r_{pa} \leq rand < r_{accept}$), se utilizan dos vecindarios con igual probabilidad de selección.

El primer vecindario, selecciona algunas cajas aleatoriamente y cambia sus posiciones, utilizando el Algoritmo 2. Luego, aplica el segundo vecindario. Este vecindario se enfoca en disminuir el largo utilizado. Por cada cliente, selecciona las cajas que tengan una mayor posición x_i e intenta disminuir este valor, es decir, ubicar la caja más atrás. Luego de pasar por la modificación de HS, la solución actual pasa por el proceso de factibilidad gracias al Algoritmo 1 y las restricciones físicas del problema.

Finalmente, el algoritmo de selección de soluciones permite considerar las mejoras en el largo del camión, como también mejoras en la proximidad de las cajas de un mismo cliente y en el orden de las cajas según la ruta. A continuación, se explica con detalle este algoritmo.

Algoritmo 3 Harmony Search modificado

```
while tiempo < tiempo.total do
  rand ← número aleatorio entre [0,1]
  if rand > r_accept then
    sol.Actual ← solución existente
  else if rand > r_pa then
    if rand > r_pa + (r_accept − r_pa)/2 then
      mover un número aleatorio de cajas de sol.Actual
    end if
    mover cajas con mayor posición x_i de sol.Actual
  else
    sol.Actual ← Solución creada aleatoriamente
  end if
  seleccionar soluciones
end while
```

C1. Selección de Soluciones: Para guiar la búsqueda de buenas soluciones, se consideran la disminución del largo y las restricciones de entregas múltiples. El algoritmo selecciona las soluciones de menor largo, pero, cuando el largo es igual, prefiere las soluciones con menor penalización o menor volumen de envolvente.

Para disminuir el largo de la solución utilizada, se busca en la solución un plano vertical YZ que se encuentre vacío debido al movimiento de las cajas. Todas las cajas que se encuentran delante del plano, se mueven hacia atrás para eliminar el plano.

Para mantener las cajas de un mismo cliente próximas, se considera una penalización por su posición y el volumen de la envolvente de las cajas. Para el cálculo de la penalización, por cada cliente $j \in \{2, ..., N\}$, se define λ_j un límite en el eje X de la posición de sus cajas de acuerdo a la Ecuación 9.

$$\lambda_j = max\{x_i + l_i : i \in \Delta_{j-1}\} \forall j = \{2, ..., N\} \qquad (9)$$

Por ejemplo, para el cliente 1 no existe penalización. Para el cliente 2 se penaliza cada caja que este antes de una caja del cliente 1, y así sucesivamente. De este modo, las cajas del cliente 1 quedarán cercanas al origen, ya que son las últimas en entregarse. Análogamente, se considera λ_j, Ecuación 10, el

límite hacia adelante que deben respetar las cajas, para todos los clientes j menos el último.

$$\dot{\lambda}_j = min\{x_i : i \in \Delta_{j+1}\} \forall j = \{1, ..., N-1\} \quad (10)$$

Dadas estas definiciones, la penalización por orden de entrega se define como:

$$Pen = \frac{\sum_{j=2}^{N} \sum_{i \in \Delta_j} max\{0, \lambda_j - x_i\}}{2L} + \frac{\sum_{j=1}^{N-1} \sum_{i \in \Delta_j} max\{0, x_i + l_i - \dot{\lambda}_j\}}{2L} \quad (11)$$

correspondiendo a una fracción del largo utilizado en la solución. Para no considerar la misma posición penalizada hacia adelante y hacia atrás, el largo se multiplica por dos.

También, se considera en el algoritmo de selección, la suma de los volúmenes de la envolvente de las cajas de cada cliente. Un menor volumen total, indicará que las cajas de los clientes están más próximas. Para su cálculo, se consideran las posiciones mínimas y máximas de las cajas del cliente, ver Ecuación 12, y luego se calcula el volumen por cliente, para finalmente sumar todos los volúmenes de los clientes, ver Ecuación 13.

$$V_j = (max_{i \in \Delta_j}\{x_i + l_i\} - min_{i \in \Delta_j}\{x_i\}) * (max_{i \in \Delta_j}\{y_i + w_i\} - min_{i \in \Delta_j}\{y_i\}) * (max_{i \in \Delta_j}\{z_i + h_i\} - min_{i \in \Delta_j}\{z_i\}) \quad (12)$$

$$V = \sum_{j=1}^{N} V_j \quad (13)$$

El algoritmo de selección, ver Algoritmo 4, analiza el largo obtenido en cada solución, actualizando la "mejor solución" cada vez que este largo disminuye. Además, cuando el largo obtenido es igual al de la mejor solución, se verifica si la penalización por orden de entrega o el volumen de las envolventes son menores, actualizando la mejor solución.

Algoritmo 4 Selección de soluciones

```
if solActual.Largo < mejorSol.Largo then
  mejorSol ← solActual
else
  if solActual.Largo = mejorSol.Largo then
    if solActual.Pen < mejorSol.Pen or solActual.Vol <
    mejorSol.Vol then
      mejorSol ← solActual
    end if
  end if
end if
```

III. Experimentos y Resultados

Se implementó el algoritmo descrito en el IDE Spyder 3.3.3, usando Python 3.7. Se utilizó un servidor con un procesador Intel(R) Xeon(R) CPU E3-1270 v6, 3.80 GHz, 62 GB de ram y 30 nodos. El código fuente se encuentra en el siguiente link: https://github.com/nicolas16e/Strip_Packing_Problem/blob/master/SPP/C%C3%B3digo_SPP.py.

Los datos y rutas son obtenidos de una empresa distribuidora de productos en cajas. Se considera un camión con un ancho de 150 cm y un alto de 215 cm. Se utilizaron 9 instancias para calibrar parámetros y 12 más para analizar el desempeño de la propuesta , de acuerdo a la disponibilidad de los datos entregados por la empresa.

El número de clientes, N, varía entre [1, 7]. El número de cajas totales, n, varía entre [29, 681], con un mínimo de una caja por cliente, hasta un máximo de 490 cajas por cliente. El largo de las cajas varía entre [10, 60] cm, el ancho entre [10, 50] cm, el alto entre [10, 60] cm y el peso entre [0.26; 76] kg.

A. Calibración de Parámetros

Se requiere calibrar las probabilidades r_{accept} y r_{pa} que regulan la metaheurística. Como criterio de término se utilizaron 8 horas para calibrar los parámetros.

TABLA I
COMBINACIONES DE PARÁMETROS Y SOLUCIÓN OBTENIDA

Combinación	I	II	III	IV	V	VI	VII	VIII	IX
r_{pa}	0,10	0,20	0,30	0,10	0,20	0,30	0,10	0,20	0,30
r_{accept}	0,70	0,70	0,70	0,80	0,80	0,80	0,95	0,95	0,95
Solución (cm)	3808	3861	3867	3843	3826	3877	3891	3867	3923

La Tabla I muestra la combinación de parámetros utilizada. Para cada parámetro, se seleccionaron tres valores de los sugeridos por la literatura y que preliminarmente entregaban buenos resultados. En la Tabla I, la fila Solución presenta la suma de los largos obtenidos en las soluciones de las nueve instancias de parametrización. Se observa que la mejor solución se obtiene con $r_{pa} = 0{,}10$ y $r_{accept} = 0{,}70$, que además obtiene el menor largo para cuatro instancias y un menor tiempo.

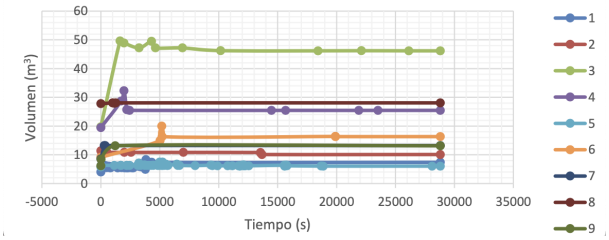
La Tabla II muestra el detalle de las soluciones encontradas con estos parámetros. La primera columna corresponde al número de la instancia. Las siguientes columnas presentan el largo obtenido, el volumen de la envolvente, la penalización por la posición de las cajas adelante o atrás de otro cliente, el tiempo de cómputo e iteración en que se obtuvo la mejor solución.

TABLA II
SOLUCIONES PARA $r_{pa} = 0{,}10$ Y $r_{accept} = 0{,}70$

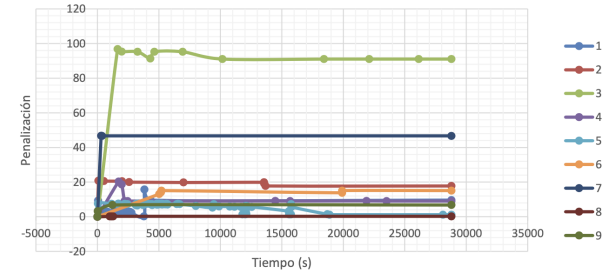
Instancia	L (cm)	Volumen (m^3)	Penalización	Tiempo (s)	Iteración
1	141	7,4	9,6	4322	1070
2	300	10,1	17,8	13673	1089
3	692	46,2	91	26141	62
4	676	25,4	9,1	23518	119
5	185	6,0	1,2	28106	2819
6	394	16,3	15	19906	602
7	309	13,2	46,7	361	6
8	881	28,0	0,3	1274	8
9	230	13,1	6,8	1222	206

Los resultados de la parametrización confirman las observaciones preliminares, que el uso de la componente aleatoria debe ser limitado en la exploración de soluciones, obteniendo una probabilidad de 0,10 el caso de la generación de una nueva

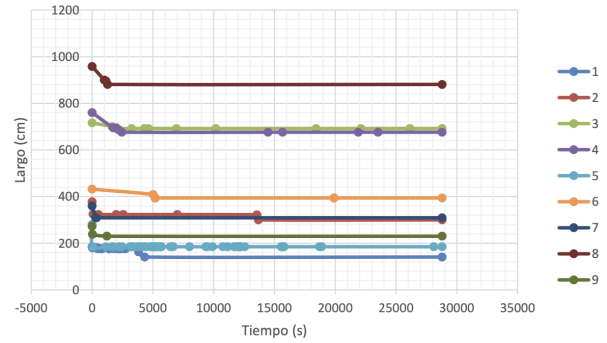
solución completamente aleatoria. La modificación parcial de la solución será con una probabilidad de 0,60 y la copia de la solución actual, con probabilidad 0,30.



(a) Volumen vs Tiempo



(b) Penalización vs Tiempo



(c) Largo vs Tiempo

Fig. 2. Gráficos de convergencia de las componentes de la función de evaluación.

En la Fig. 2, se presentan los gráficos de los componentes de la función de evaluación versus el tiempo: volumen de la envolvente de las cajas de un mismo cliente 2a, penalización por la posición delante o atrás de las cajas de otro cliente 2b y largo utilizado por la solución 2c. Se observa que las soluciones mejoran hasta los 20.000 segundos aproximadamente, en particular, para la Penalización vemos variaciones entre los 15.000 y 20.000 segundos; por lo cual se establece este como el tiempo límite.

B. *Evaluación del Algoritmo*

Para la evaluación del algoritmo, se utilizaron 12 instancias adicionales. Se compara la solución de la heurística constructiva con la solución encontrada por la metaheurística modificada.

La Tabla III muestra ambas soluciones para cada instancia. La primera columna, Ins, indica la instancia. La segunda y tercera columna muestra la cantidad de clientes (N) y cajas (n) respectivamente. Las cuarta, quinta y sexta columnas indican los valores de la solución encontrada por la heurística constructiva para el largo (L), la suma de los volúmenes de las envolventes de las cajas (Vol) y el tiempo de cómputo. Esta solución no tiene penalización ya que las cajas no bloquean a las de otros clientes. Las siguientes columnas detallan la solución encontrada por la metaheurística. Además de las características descriptivas de la solución: largo (L), volumen de la suma de las envolventes de las cajas (Vol) y penalización por posición (Penalización); se presenta, en la novena columna, la iteración en que se encuentra esa solución. En algunas instancias, se presenta entre paréntesis la solución anterior a la mejor solución, donde se mantiene el largo pero se mejora el volumen de envolventes o penalización por posición.

C. *Discusión de los Resultados*

En los resultados obtenidos, podemos observar que la heurística constructiva entrega una solución muy rápido, con 3,2 ms en promedio. Además, la metaheurística mejora el largo de la solución de la heurística constructiva en todas las instancias, excepto en la instancia 10, ver Fig. 3, para la cual mantiene la misma solución. Esta instancia tiene un sólo cliente y una de las cajas tiene largo 40 cm, por lo cual, ambas soluciones coinciden.

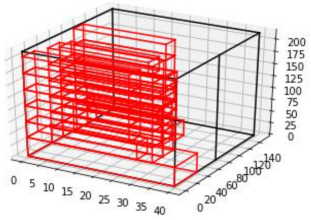


Fig. 3. Solución para la instancia 10.

En promedio, el largo disminuye un 11,6 %, aumentando el volumen de las envolventes un 47,4 % en promedio y con un promedio de la penalización por posición de 18,3. La mayor mejora en el largo, se alcanza en la instancia 5, Fig. 4, en donde se disminuye en un 33,9 % respecto a la solución de la heurística constructiva, aumentando el volumen de las envolventes un 5,2 % y la penalización por posición 1,2.

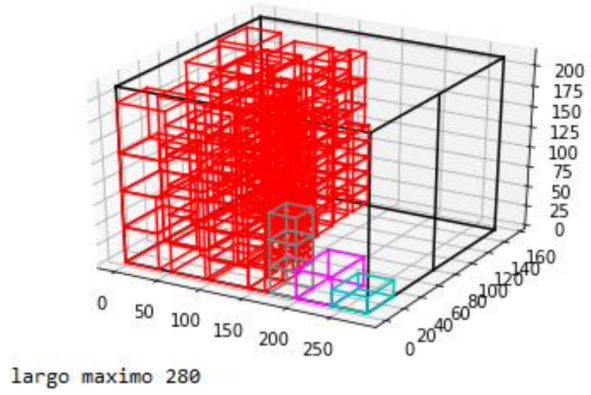
En diez instancias (instancias 3, 5, 6, 7, 13, 15, 16, 17, 20 y 21), podemos observar como la metaheurística mejora las características de entregas múltiples mediante la función objetivo. Para estas soluciones, el largo se mantiene. En cinco de estas instancias (3, 5, 15, 16 y 17), se mejoran tanto el volumen de las envolventes como la penalización por posición, 8,9 % y 40,1 % en promedio, respectivamente. Cuatro, mejoran sólo la penalización por posición, 8,7 % en promedio. La instancia 6 mejora el volumen de las envolventes en un 18,5 % y aumenta la penalización en un -6,4 %.

IV. Conclusiones

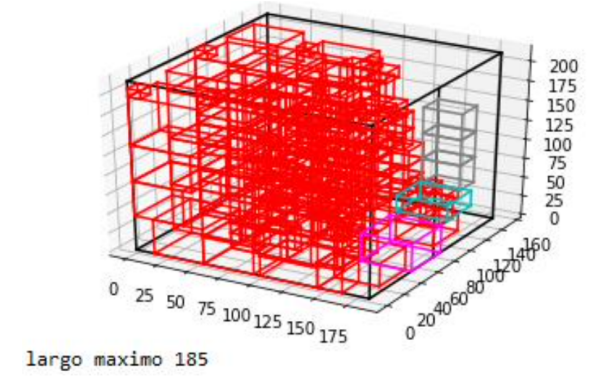
Se resuelve el problema de carga de un camión con cajas heterogéneas, respetando restricciones que facilitan el despacho de las cajas a distintos clientes, utilizando un enfoque de una dimensión abierta. Se proponen una rápida heurística

TABLA III
RESULTADOS DE LA HEURÍSTICA CONSTRUCTIVA Y *Harmony Search*

Ins.	N	n	Heurística Constructiva			*Harmony Search*			
			L (cm)	Vol (m^3)	Tiempo (ms)	L(cm)	Vol (m^3)	Penalización	Iteración
1	3	83	185	4,0	20,1	141	7,4	9,6	1070
2	3	141	378	11,4	1,4	300	10,1	17,8	1089
3	6	681	716	19,7	5,0	692	46,2 (48,9)	91 (95,3)	44 (7)
4	7	573	760	19,5	4,3	676	25,5	9,1	83
5	4	132	280	5,8	1,4	185	6,1 (6,3)	1,2 (7,6)	1928 (7)
6	6	222	432	9,1	2,4	394	16,3 (20)	15 (14,1)	602 (144)
7	5	300	360	8,6	3,1	309	13,2	46,7 (46,8)	6 (5)
8	4	435	958	27,8	3,0	881	28,1	0,27	8
9	4	103	273	6,2	1,1	230	13,1	6,8	206
10	1	29	40	1,0	0,3	40	1,0	0	0
11	3	84	223	5,4	0,9	178	5,7	10	4877
12	3	160	210	5,2	1,7	164	6,5	18,6	992
13	4	126	239	5,2	1,3	176	9,8	16,6 (23,1)	2032 (1799)
14	5	387	670	18,1	2,7	613	36,7	77	90
15	5	170	245	5,3	1,8	192	8,4 (10,6)	7,5 (14,1)	1233 (54)
16	4	389	566	16,3	2,6	564	28,3 (28,4)	33 (33,5)	10 (7)
17	6	528	694	18,9	3,9	616	28,8 (33,7)	16,3 (44,6)	113 (106)
18	3	501	604	16,5	3,2	567	17,9	0,6	7
19	4	387	648	17,1	2,7	628	18,6	0,7	103
20	4	195	250	5,7	2,1	218	7,2	3,7 (3,8)	684 (90)
21	4	125	215	3,9	1,3	145	5,2	2,6 (2,7)	1785 (58)



(a) Solución Heurística Constructiva



(b) Solución *Harmony Search*

Fig. 4. Soluciones para la instancia 5.

constructiva y una modificación de la metaheurística *Harmony Search* que permiten encontrar soluciones factibles para las necesidades de la compañía en tiempos adecuados a la etapa de planificación de la carga del camión. El algoritmo de selección de soluciones, que guía la metaheurística, permite mejorar no sólo el largo utilizado del camión, sino también las características asociadas a la entrega a múltiples clientes, manteniendo las cajas de un mismo cliente cercanas y ordenadas de acuerdo a la secuencia de entrega.

Como trabajo futuro, se considera la evaluación respecto a los modelos matemáticos en instancias pequeñas considerando la rotación de las cajas y la distribución de la carga en el camión. También, se considera la comparación de soluciones con diferentes heurísticas o metaheurísticas, para evaluar cual tiene un mejor desempeño en la resolución del problema.

AGRADECIMENTOS

Los autores agradecen el apoyo financiero del Instituto Sistemas Complejos de Ingeniería, ANID PIA/BASAL AFB180003. También agradecemos los comentarios y sugerencias de los revisores en el proceso editorial.

REFERENCIAS

[1] A. Bortfeldt and D. Mack, "A heuristic for the three-dimensional strip packing problem," *European Journal of Operational Research*, vol. 183, pp. 1267 – 1279, 2007.

[2] S. Allen, E. K. Burke, and G. Kendall, "A hybrid placement strategy for the three-dimensional strip packing problem," *European Journal of Operational Research*, vol. 209, pp. 219 – 227, 2011.

[3] L. Wei, W.-C. Oon, W. Zhu, and A. Lim, "A reference length approach for the 3d strip packing problem," *European Journal of Operational Research*, vol. 220, pp. 37 – 47, 2012.

[4] O. X. do Nascimento, T. Alves de Queiroz, and L. Junqueira, "Practical constraints in the container loading problem: Comprehensive formulations and exact algorithm," *Computers & Operations Research*, vol. 128, pp. 105–186, 2021.
[5] C. A. Vega-Mejía, R. G. García-Cáceres, and J. P. Caballero-Villalobos, "Hacia la optimización integral del problema de carga de contenedores," in *Congreso Latino-Iberoamericano de Investigación Operativa, Simposio Brasileiro de Pesquisa Operacional*, pp. 482–493, september 2012.
[6] M. da Graça Costa and M. E. Captivo, "Weight distribution in container loading: a case study," *International Transactions in Operational Research*, vol. 23, pp. 239 – 263, 2016.
[7] M. T. Alonso, R. Alvarez-Valdes, F. Parreño, and J. M. Tamarit, "Algorithms for pallet building and truck loading in an interdepot transportation problem.," *Mathematical Problems in Engineering.*, vol. 2016, pp. 1 – 11, 2016.
[8] L. Junqueira, R. Morabito, and D. S. Yamashita, "Mip-based approaches for the container loading problem with multi-drop constraints," *Annals of Operations Research*, vol. 199, pp. 51–75, 2012.
[9] A. Bortfeldt and G. Wäscher, "Constraints in container loading – a state-of-the-art review," *European Journal of Operational Research*, vol. 229, no. 1, pp. 1 – 20, 2013.
[10] J. Respen and N. Zufferey, "Metaheuristics for truck loading in the car production industry," *International Transactions in Operational Research*, vol. 24, pp. 277 – 301, 2017.
[11] M. Iori, S. Martello, and M. Monaci, "Metaheuristic algorithms for the strip packing problem," in *Optimization and Industry: New Frontiers. Applied Optimization* (P. Pardalos and V. Korotkikh, eds.), vol. 78, Springer, Boston, MA, 2013.
[12] N. Bansal, X. Han, K. Iwama, M. Sviridenko, and G. Zhang, "A harmonic algorithm for the 3d strip packing problem.," *SIAM Journal on Computing*, vol. 42, pp. 579 – 592, 2013.
[13] D. V. Kurpel, C. T. Scarpin, J. E. P. Junior, C. M. Schenekemberg, and L. C.Coelho, "The exact solutions of several types of container loading problems.," *European Journal of Operational Research*, vol. 284, pp. 87 – 107, 2020.
[14] M. A. Alba Martínez, F. Clautiaux, M. Dell'Amico, and M. Iori., "Exact algorithms for the bin packing problem with fragile objects.," *Discrete Optimization*, vol. 10, pp. 210 – 223, 2013.
[15] M.T.Alonso, R.Alvarez-Valdes, M.Iori, F.Parreño, and M.Tamarit, "Mathematical models for multicontainer loading problems.," *Omega.*, vol. 66, pp. 106 – 117, 2017.
[16] Z. W. Geem, J. H. Kim, and G. V. Loganathan, "A new heuristic optimization algorithm: Harmony search," *Simulation*, vol. 76, no. 2, pp. 60 – 68, 2001.
[17] X.-S. Yang, "Harmony search as a metaheuristic algorithm," in *Music-Inspired Harmony Search Algorithm: Theory and Applications* (Z. W. Geem, ed.), ch. 1, pp. 1–14, Studies in Computational Intelligence, Springer Berlin, vol 191, 2009.
[18] X. Gao, V. Govindasamy, H. Xu, X. Wang, and K. Zenger, "Harmony search method: Theory and applications," *Computational Intelligence and Neuroscience*, vol. 2015, p. 10 pages, 2015.
[19] Z. W. Geem, *Music-Inspired Harmony Search Algorithm: Theory and Applications*. Studies in Computational Intelligence, Springer Berlin, vol 191, 2009.
[20] I. Ayachi, R. Kammarti, M. Ksouri, and P. Borne, "Harmony search algorithm for the container storage problem.," *8th International Conference of Modeling and Simulation*, 2010.

Nicolás Anabalón Romero Nacido en Bulnes el 16 de mayo de 1997, con estudios básicos y medios en la misma ciudad, para luego estudiar en la Universidad de Concepción, obteniendo el título de Ingeniero Civil Industrial, al mismo tiempo que realiza algunos trabajos y participa en algunas actividades de la carrera. Recientemente titulado y con residencia en Concepción.

Matías Barros Vásquez Ingeniero Civil Industrial de la Universidad de Concepción. Durante sus estudios trabajo en dos artículos sobre optimización de redes de contacto, basando sus estudios e intereses en Investigación de operaciones. Realiza su Memoria de Título en optimización de reparto de productos, mediante un método exacto, estudio que continua, con métodos heurísticos para la Tesis de Magíster en Ingeniería Industrial, grado que obtiene dos años después.

Rosa Medina Durán Ingeniera Civil Industrial de la Universidad de Concepción, Magíster en Ingeniería Industrial de la misma casa de estudios. Doctora en Investigación de Operaciones de la Universidad de Bologna. Sus áreas de investigación corresponden a Optimización Combinatoria con aplicaciones en problemas de corte, empaquetamiento, forestal, minería y turismo.