

PONTIFICIA UNIVERSIDAD CATÓLICA DE VALPARAÍSO

FACULTAD DE INGENIERÍA

ESCUELA DE INGENIERÍA INFORMÁTICA

**RESOLUCIÓN DEL PRE-MARSHALLING  
PROBLEM UTILIZANDO FRUIT FLY  
OPTIMIZATION ALGORITHM**

**SEBASTIÁN RODRIGO SÁEZ MANSILLA**

INFORME FINAL DEL PROYECTO

PARA OPTAR AL TÍTULO PROFESIONAL DE

INGENIERO DE EJECUCIÓN EN INFORMÁTICA

Agosto 2016

PONTIFICIA UNIVERSIDAD CATÓLICA DE VALPARAÍSO

FACULTAD DE INGENIERÍA

ESCUELA DE INGENIERÍA INFORMÁTICA

**RESOLUCIÓN DEL PRE-MARSHALLING  
PROBLEM UTILIZANDO FRUIT FLY  
OPTIMIZATION ALGORITHM**

**SEBASTIÁN RODRIGO SÁEZ MANSILLA**

Profesor Guía: **Ricardo Soto De Giorgis**

Profesor Co-referente: **Héctor Allende Cid**

**Ingeniería en Ejecución Informática**

Agosto 2016

## **Dedicatoria.**

A mi tía María Elena, por su entrañable preocupación y cariño. A mis padres, Erminda y Rosendo, por su incansable apoyo y sacrificio durante todos estos años.

# Índice

<b>Resumen y Abstract .....</b>	<b>iii</b>
<b>Resumen.....</b>	<b>iii</b>
<b>Abstract.....</b>	<b>iii</b>
<b>Lista de Figuras .....</b>	<b>iv</b>
<b>Lista de Tablas .....</b>	<b>v</b>
<b>1 Introducción .....</b>	<b>1</b>
<b>2 Definición de Objetivos.....</b>	<b>2</b>
<b>2.1 Objetivo General.....</b>	<b>2</b>
<b>2.2 Objetivos Específicos .....</b>	<b>2</b>
<b>3 Estado del Arte .....</b>	<b>3</b>
<b>4 Problemática.....</b>	<b>6</b>
<b>4.1 Descripción del Problema de Ordenamiento de Contenedores .....</b>	<b>6</b>
<b>4.2 Pre-Marshalling Problem.....</b>	<b>6</b>
4.2.1 Definición Formal del Pre-Marshalling Problem .....	9
<b>5 Descripción del Límite Inferior.....</b>	<b>10</b>
<b>5.1 Definiciones Límite Inferior .....</b>	<b>10</b>
<b>5.2 Ejemplo uso del límite inferior .....</b>	<b>13</b>
<b>6 Técnica Utilizada.....</b>	<b>17</b>
<b>6.1 Concepto básico de Fruit Fly Optimization Algorithm.....</b>	<b>17</b>
<b>6.2 Implementación al problema .....</b>	<b>19</b>
<b>6.3 Adaptación FOA .....</b>	<b>20</b>
6.3.1 Inicialización de la población .....	20
6.3.2 Búsqueda Basada en Olor y Visión Local .....	21
6.3.3 Búsqueda Basada en Visión Global.....	21

6.3.3 Operación de Reparación.....	21
6.3.5 Variables Utilizadas.....	22
<b>7 Evaluación.....</b>	<b>24</b>
<b>7.1 Experimentos con LC .....</b>	<b>24</b>
<b>7.2 Experimentos con CV .....</b>	<b>25</b>
<b>8 Conclusiones .....</b>	<b>26</b>
<b>9 Referencias.....</b>	<b>27</b>
<b>Anexos .....</b>	<b>29</b>
<b>A. Detalles de los Casos de Prueba .....</b>	<b>29</b>
<b>B. Detalle Resultados Obtenidos.....</b>	<b>29</b>
<b>C. Evidencia.....</b>	<b>34</b>

# Resumen y Abstract

## Resumen

En los terminales portuarios los contenedores se organizan en bahías, que en la práctica corresponden a conjuntos de pilas de contenedores. El objetivo del Pre-Marshalling Problem es minimizar la cantidad de recolocaciones para reorganizar una bahía de contenedores en función de su secuencia de envío. Esta reorganización es de gran importancia para la correcta operación de los puertos, dado que la secuencia de llegada de contenedores al puerto no es compatible con la secuencia de envío. En este proyecto, se utilizará la metaheurística Fruit Fly Optimization Algorithm para la resolución de este problema, la cual está inspirada en el comportamiento de la mosca de la fruta. Se presentan resultados prometedores para distintas instancias del problema.

*Palabras-claves: Pre-Marshalling problem, Metaheurística, Fruit Fly Optimization Algorithm, Contenedor.*

## Abstract

In port terminals, containers are organized in bays that in practice correspond to sets of container stacks. The goal of the Pre-Marshalling Problem is to minimize the amount of relocations for reorganizing a container bay according to their shipment sequence. This reorganization is of great relevance to the proper operation of ports because the arrival sequence of containers to the port does not match their departure sequence. In this project, we employ the Fruit Fly Optimization metaheuristic for solving this problem, which is inspired on the behavior of fruit flies. We present promising results for different instances of the problem.

*Keywords: Pre-Marshalling problem, Metaheuristic, Fruit Fly Optimization Algorithm, Container.*

## Lista de Figuras

Figura 4. 1 Ejemplo de Bahía.....	7
Figura 4. 2 Resolución del PMP tipo A .....	8
Figura 4. 3 Resolución del PMP tipo B .....	8
Figura 5. 1 Bahía inválida .....	13
Figura 5. 2 Ejemplo de obtención demanda para el grupo $g=4$ .....	14
Figura 5. 3 Ejemplo de obtención para la oferta potencial .....	15
Figura 5. 4 Elementos bien posicionados con índice menor al $g^*=3$ . .....	16
Figura 6.1 Búsqueda de la comida por parte de un grupo de moscas. ....	17
Figura C. 1 Ejemplo de bahía resultante para CV1_2.bay. ....	35
Figura C. 2 Ejemplo de recolocaciones resultantes para ordenar CV1_2.bay. ....	35

## Lista de Tablas

Tabla 5. 1 Términos y definiciones para el límite inferior.....	12
Tabla 5. 2 Valores de demanda y oferta para los contenedores de la bahía.....	13
Tabla 6. 1 Ejemplo de archivo contenedor de bahía.....	19
Tabla 6. 2 Principales variables para el desarrollo del FOA para PMP.....	22
Tabla 7. 1 Tabla comparativa resultados para LC (parte 1).....	24
Tabla 7. 2 Tabla comparativa resultados para LC (parte 2).....	25
Tabla 7. 3 Tabla comparativa de resultados para CV.....	25
Tabla A. 1 Detalle de los casos de prueba LC.....	29
Tabla A. 2 Detalle de los casos de prueba CV.....	29
Tabla B. 1 Detalle de los resultados obtenidos según ejecución para CV (1). .....	30
Tabla B. 2 Detalle de los resultados obtenidos según ejecución para CV (2). .....	30
Tabla B. 3 Detalle de los resultados obtenidos según ejecución para CV (3). .....	31
Tabla B. 4 Detalle de los resultados obtenidos según ejecución para CV (4). .....	31
Tabla B. 5 Detalle de los resultados obtenidos según ejecución para CV (5). .....	31
Tabla B. 6 Detalle de los resultados obtenidos según ejecución para CV (6). .....	32
Tabla B. 7 Detalle de los resultados obtenidos según ejecución para LC (1).....	32
Tabla B. 8 Detalle de los resultados obtenidos según ejecución para LC (2).....	33
Tabla B. 9 Detalle de los resultados obtenidos según ejecución para LC (3).....	33
Tabla B. 10 Detalle de los resultados obtenidos según ejecución para LC (4).....	33
Tabla B. 11 Detalle de los resultados obtenidos según ejecución para LC (5).....	34
Tabla B. 12 Detalle de los resultados obtenidos según ejecución para LC (6).....	34

# 1 Introducción

Los terminales portuarios juegan un rol crucial como intercambio intermodal de las redes globales de transporte de mercancía, así como puntos de destino y entrega de éstos. Dicho intercambio se opera a través de contenedores cuyo tráfico ha crecido de manera ininterrumpida en las últimas décadas, implicando un esfuerzo en cada terminal en la gestión adecuada de sus capacidades para el almacenamiento y manipulación.

En este ámbito, el patio de contenedores juega un papel decisivo en la organización de estos, ya que desde aquí se transportan y reorganizan, afectando directamente la productividad del terminal. Aquí los contenedores o artículos son ordenados uno sobre otros formando pilas de disímiles alturas y, que en su conjunto, conforman estructuras denominadas bahías. Estos artículos permanecerán allí hasta que, siguiendo un plan de estiba (orden en que deben ser cargados a las naves), sean embarcados para dirigirse a su destino final.

Sin embargo, a pesar de la existencia de este procedimiento de carga, las bahías comúnmente se encuentran con un orden o configuración distinta a lo requerido, debiendo ser reconfigurada a través de movimientos (denominados recolocaciones), que trasladan los contenedores de su pila actual a otra pila, llegando a una configuración o bahía válida. Esta serie de recolocaciones que busca lograr una configuración acorde al plan de estiba se denomina Pre-Marshalling.

El servicio de los terminales se ve afectado directamente por la capacidad que tienen para manejar los problemas del Pre-Marshalling. Las operaciones de recolocación de contenedores pueden retrasar considerablemente la partida de un barco, produciendo un efecto negativo en la eficiencia del terminal. Es así, como una reducción en el número de recolocaciones para obtener una bahía válida junto con una serie de requerimientos, conducen a los objetivos principales de este documento.

Aunque el problema del Pre-Marshalling ha sido estudiado bajo diversas aristas y criterios que han generado disímiles resultados y estudios, sigue siendo, aún; tema de investigación. A raíz de lo anterior, se propone el algoritmo de optimización de la mosca de la fruta para la resolución del problema de Pre-Marshalling. Esta metaheurística está basada en la forma en la cual las moscas de la fruta buscan su comida a través de sensibles órganos que las hacen superiores a otras especies. Es un algoritmo adaptable a variadas problemáticas y que tiene como principal estructura o filosofía, la búsqueda basada en olfato y en visión.

De este modo, el presente informe aborda el método propuesto para la resolución del problema de pre-ordenamiento de contenedores, entregando objetivos acordes a la investigación y la complejidad de ésta. Así mismo, haciendo énfasis en la adaptación del Fruit Fly Optimization Algorithm para este caso particular. A la vez, se describe el uso del límite inferior en la obtención del número de movimientos necesarios para llegar a una bahía válida y su implicancia en este estudio.

## **2 Definición de Objetivos**

En la presente sección se detalla la meta formal que se pretende alcanzar con este estudio, así como los objetivos específicos que contribuirán a la obtención de ésta.

### **2.1 Objetivo General**

Resolver el Container Pre-Marshalling Problem utilizando la metaheurística Fruit Fly Optimization Algorithm.

### **2.2 Objetivos Específicos**

- Comprender a cabalidad el Container Pre-Marshalling Problem.
- Comprender la Metaheurística Fruit Fly Optimization Algorithm.
- Desarrollar un modelo para resolver la problemática del Pre-Marshalling utilizando una variación del algoritmo de la mosca de la fruta.
- Realizar una implementación de Fruit Fly Optimization Algorithm para solucionar el Pre-Marshalling.
- Realizar experimentos y analizar resultados de la implementación realizada.

### 3 Estado del Arte

Como se describió anteriormente, el problema del pre-ordenamiento de contenedores o Pre-Marshalling Problem (PMP) puede ser estudiado bajo distintas premisas y de diversas formas. En torno a este ámbito y con el objetivo principal de optimizar los procesos de carga y descarga de contenedores, se debe dar respuesta en primer lugar; cómo solucionar el problema de recolocaciones y, en segundo; qué métodos utilizar para lograr la esperada optimización. Estos cuestionamientos son los que proporcionan el sentido al estudio en cuestión.

Sobre la resolución del PMP y la búsqueda del método más óptimo de manejo de contenedores (su carga y descarga), distintos autores han presentado una completa definición del problema y/o propuesto variadas soluciones al respecto. Algunos de estos estudios se señalan a continuación, junto a una breve descripción al respecto y su influencia en la investigación.

En [16] se aprecian detalles de investigaciones respecto a la problemática del post-apilamiento, así como sus contribuciones que encaminan a una solución a éste. Además, se evidencian algunas clasificaciones distintas al PMP, que ayudan a entender de mejor manera esta problemática. Es así como se describe el BRP (Block Relocation Problem), reubicación de bloques de contenedores y RMP (Re-Marshalling Problem), reubicación de contenedores dentro de un bloque. Estas problemáticas de mayor envergadura colaboran al acabado entendimiento del PMP y de la necesidad de una óptima solución de éste.

Dos tipos de PMP son introducidos en [15], los cuales tienen ciertas distinciones que deben ser consideradas a la hora de gestionar su solución. El PMP tipo A trata sobre un orden de contenedores según su índice de prioridad, en cambio; el tipo B se enfoca en el orden de contenedores según su característica. En este artículo se describen dos métodos heurísticos destinados a resolver cada una de estas problemáticas. Junto a esto y respecto al PMP tipo A, que es en definitiva la problemática de esta investigación, se presenta una definición formal del Pre-Marshalling Problem en [6].

Para la resolución del PMP es necesario aplicar una serie de movimientos de los contenedores, de forma que se cumpla de modo eficaz con las prioridades de salida de cada uno de ellos. A esta serie de movimientos se les denomina recolocaciones, los cuales pueden ser clasificados según su eficiencia. Bajo este antecedente, en el artículo [1] se nombran algunas características que deben cumplir las recolocaciones para que sean productivas y, por ende, poder re-organizar la bahía en el menor número de movimientos posibles. Esto implica, por lo demás, reducir la cantidad de movimientos improductivos, lo cual se puede tratar haciendo uso de un límite inferior que entrega una medición sobre la cantidad eficiente de movimientos de reubicación.

En [17], los autores describen el problema y proponen una heurística que junto a una serie de restricciones, era capaz de dar solución a éste. Siguiendo esta línea, en [5], los autores Lee y Chao presentan un proceso de búsqueda heurístico basado en vecindarios, realizado en conjunto a un modelo de programación entera y tres subrutinas

de menor tamaño. Los experimentos con esta propuesta fueron realizados en bahías más bien pequeñas o medianas (medidas según la cantidad de pilas, niveles y contenedores.)

Sobre esto último, en el artículo [8] se narran las características más relevantes del PMP. Allí se describen varios conflictos existentes al momento de realizar el proceso de recolocación, así como el efecto negativo o positivo que éstos puedan tener en el terminal portuario. Con respecto a lo anterior, en [10] se expone una solución siguiendo un árbol heurístico de búsqueda, con tal de obtener la cantidad mínima de recolocaciones, resultando en una bahía donde cada pila está ordenada de manera descendente. Los resultados conseguidos en éste artículo fueron comparados con los obtenidos por Lee y Chao en [5], mejorando en un gran porcentaje los resultados en comparación a éstos.

Según se describe en el artículo [18], el problema del Pre-Marshalling es resuelto de manera significativa en paralelo a la actual literatura, debido a las comparaciones realizadas con los resultados de otros métodos; tales como los expuestos en [4]. Con el fin de obtener instancias con diferentes grados de dificultad, que van desde lo más simple a lo más complejo, es que se tienen en cuenta dos características. Por un lado, se considera la tasa de ocupación de la bahía de contenedores, y por otro; para medir el grado de la instancia se considera el porcentaje de contenedores con alta prioridad que se encuentran por debajo de artículos con menor prioridad.

Referente al Fruit Fly Optimization Algorithm, este es presentado por Pan, principalmente en [11], donde exhibe una definición formal de este método junto a sus cuatro pasos o etapas principales. Éstas son la inicialización, búsqueda por olfato, evaluación de la población y búsqueda basada en la visión. Luego en [2], narra mayormente sus conceptos y variables claves. Además, se presentan una serie de aplicaciones en distintos niveles de complejidad, donde se aprecia la versatilidad de este algoritmo.

Si bien el algoritmo se presenta como altamente adaptable, la estructura propuesta por Pan en [2] requiere modificaciones para utilizarse en determinados casos. Debido a esto, en el artículo [7] se modifica el Fruit Fly Optimization Algorithm (FOA) siendo presentado como un algoritmo binario para poder resolver un problema vectorial. Conjuntamente, se utiliza un vector de generación de probabilidad para la creación de nuevas soluciones.

Por consiguiente, en [7] se describen características de búsqueda binaria. Allí, cada mosca es una solución válida y con soluciones iniciales según un vector de probabilidad. A partir de esto, se sigue un proceso de búsqueda de olfato y visión según vecindarios y un método de búsqueda de solución global, además; métodos de reparación ayudan en la obtención de soluciones acorde a los requerimientos del problema vectorial.

De igual modo, en [9]; los autores presentan éste método binario de búsqueda para dar solución a otra problemática, combinando diferentes procesos de binarización. Distintamente al proceso descrito en [7], los métodos de reparación de las soluciones dan cabida a la adaptación de esta búsqueda binaria.

Todo lo anteriormente descrito, guía y entrega las herramientas necesarias para entender y poder dar una sólida solución al PMP. Esto implica que las búsquedas heurísticas son la mejor alternativa para entregar una óptima solución al problema del pre-ordenamiento.

## 4 Problemática

### 4.1 Descripción del Problema de Ordenamiento de Contenedores

El patio de contenedores es uno de los lugares más críticos a nivel de gestión en los terminales portuarios. Aquí se almacenan las bahías (porción en dos dimensiones donde se depositan los contenedores), las cuales tienen un número limitado de pilas de contenedores junto con un nivel máximo de apilamiento (altura), donde se extraen para ser cargados y dirigidos a su destino (véase la figura 4.1). En la mayoría de los casos, estos contenedores están incluidos en un plan de salida para cada barco en particular, es decir, se puede establecer un orden de salida según la prioridad de cada contenedor.

Los contenedores inferiores en la bahía sólo pueden ser rescatados si antes son obtenidos los que se encuentren sobre él, debido a que éstos siguen un orden LIFO (último en entrar, primero en salir). Si bien los contenedores son ordenados teniendo en cuenta su prioridad de salida, existe una alta probabilidad de que contenedores con una alto índice de prioridad queden bajo contenedores de menor índice. Esta situación conlleva a la realización de movimientos de reubicación, con tal de tener acceso a aquellos contenedores con mayor prioridad lo que, en definitiva, genera el retraso en la salida de los buques.

Esta secuencia de reubicaciones es realizada con tal de obtener una bahía con todos los contenedores ordenados, haciendo así posible su extracción conforme al plan de estiba, sin necesidad de nuevas reubicaciones. En la literatura este problema, y como se mencionó anteriormente, es llamado Pre-Marshalling.

Existen tres tipos de problemas de organización de contenedores, cada cual con distintos propósitos. Si bien tienen un efecto y magnitud distinta en el manejo del terminal portuario, todas influyen directamente en la eficacia y eficiencia de éste. A continuación se detallan los distintos problemas:

- En primer lugar, se encuentra el problema del Block Relocation Problem (BRP), el cual tiene como objetivo el ordenamiento de contenedores en un conjunto de bloques.
- En segundo lugar, se tiene el Re-Marshalling Problem (RMP) referido a reubicaciones de contenedores dentro de un bloque.
- Y en tercer lugar; aparece el PMP el cual es objeto de esta investigación que se define resumidamente como reordenamiento de contenedores en una bahía.

### 4.2 Pre-Marshalling Problem

Se puede describir como; un problema de reorganización de una bahía con una configuración u orden inicial inválido, con el fin de obtener una bahía en la que las operaciones de recuperación puedan desarrollarse sin más traslados y, a su vez, reducir al mínimo el número final de reubicaciones requeridas para reorganizarlos. Estas recolocaciones son realizadas por una grúa especial y son efectuadas desde una pila de

origen a una nueva pila. Más adelante se presentará una definición formal sobre este problema.

Para poder lograr un mejor trabajo respecto a la resolución del PMP, se utilizan una serie de supuestos utilizados para su resolución, los cuales se presentan a continuación:

- El problema se limita a una sola bahía contenedora, es decir, las recolocaciones están restringidas a los límites de cada bahía, dada por su número de pilas y su nivel.
- Todos los contenedores poseen las mismas dimensiones y masa.
- Los índices de prioridad de cada contenedor están bien definidos, son números enteros y se conocen con antelación. Estos valores no son modificables entre contenedores.

Se debe considerar que existen dos clasificaciones para el Pre-Marshalling Problem. La primera (y por la cual se realiza esta investigación) se denomina PMP tipo A, ilustrada en la figura 4.2. Se describe como una organización de contenedores por índices de prioridad, donde a menor índice implica una mayor prioridad. Se desea, por tanto, encontrar la secuencia mínima de recolocaciones que transforme una configuración inicial a una en que no existan contenedores situados sobre otros con índice menor.

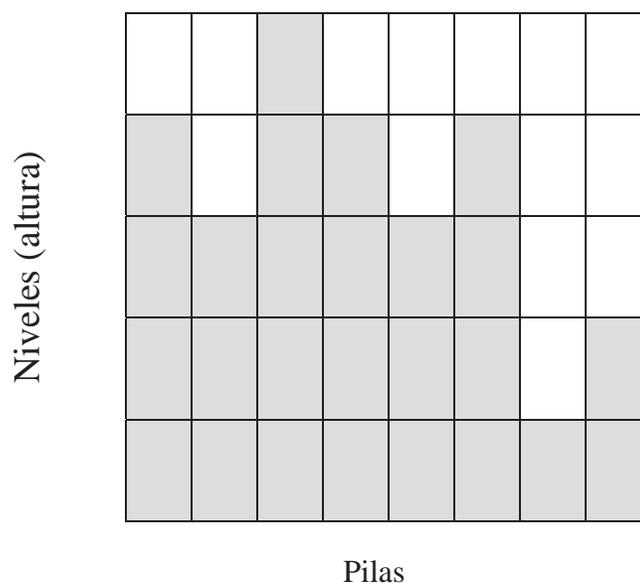


Figura 4. 1 Ejemplo de Bahía

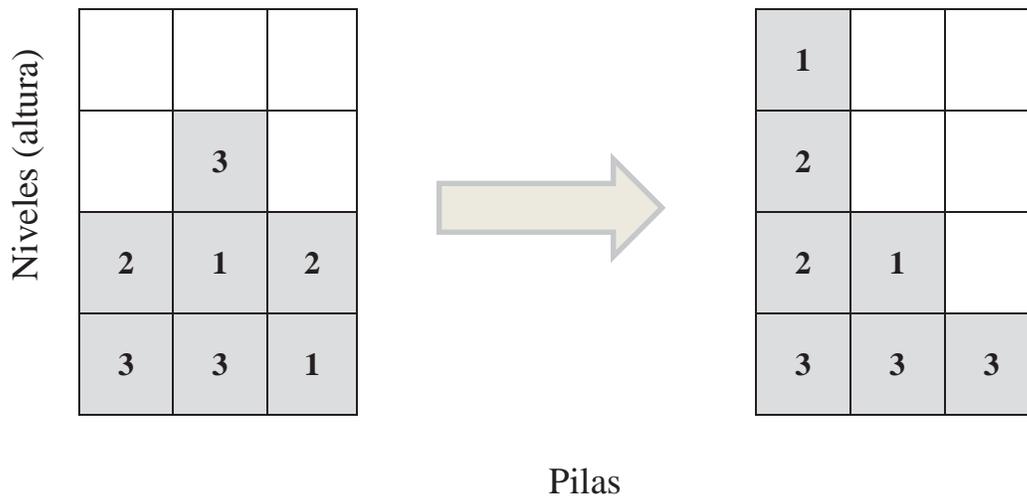


Figura 4. 2 Resolución del PMP tipo A

La segunda clasificación del Pre-Marshalling Problem corresponde a una organización de contenedores según sus características, por ejemplo, contenedores que deben ser cargados en un mismo buque, un mismo destino, similitud de productos, etc. El propósito de este tipo de PMP es agrupar los contenedores con categorías iguales en una misma pila, esto implica (y se asume) que cada bahía tiene tantas pilas como categorías existan en ella. Un ejemplo de la resolución de este tipo de PMP es ilustrado en la figura 4.3.

Debido a la logística utilizada y los tiempos computacionales elevados necesarios para la resolución de este tipo de problema (en especial el PMP tipo A), es preciso el uso de métodos que obtengan resultados acorde a las necesidades del terminal, donde una buena respuesta es de suma importancia. Es así que, a través de métodos heurísticos, se realizará la búsqueda de la secuencia de recolocaciones que permitan llegar a una bahía con una configuración valida. Además, cumplir con otro de los factores principales del PMP que es el hallar la secuencia mínima de movimientos.

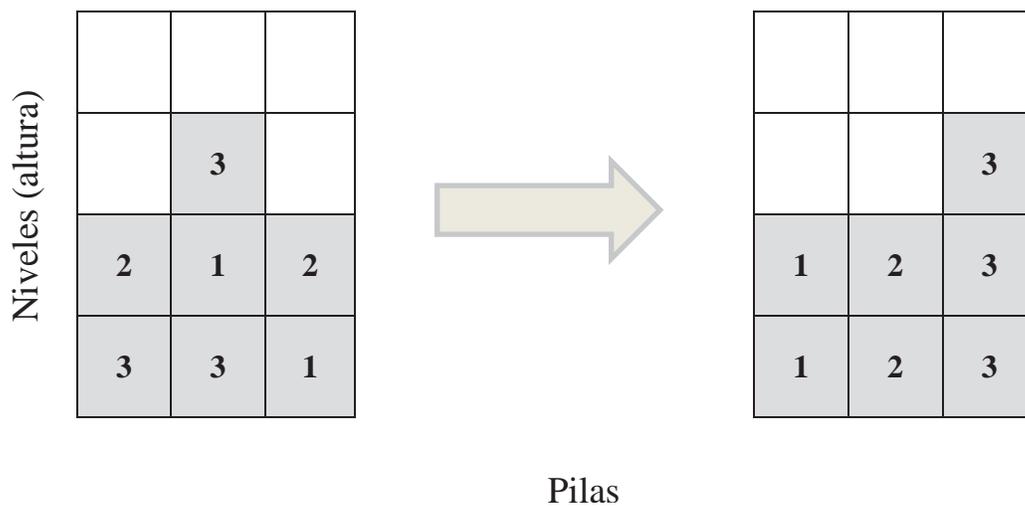


Figura 4. 3 Resolución del PMP tipo B

### 4.2.1 Definición Formal del Pre-Marshalling Problem

Según lo descrito en [5] junto a la descripción de índices de prioridades dadas en [9] y acorde al PMP tipo A que se estudia en esta investigación, el Pre-Marshalling Problem se describe de la siguiente forma. Se considera una bahía  $L$  con  $S$  pilas y una altura o nivel máximo dado por  $H$ . La bahía  $L$  posee un grupo de contenedores con índice  $g$  ( $0 \leq g \leq G$ ).

La configuración inicial  $L$  posee una serie de casillas o espacios de almacenamientos definidos como  $L_{s,h}$  con  $s = 1, \dots, S$  y  $h = 1, \dots, H$ , además si  $L_{s,h} = 0$  implica que el espacio  $L_{s,h}$  está vacío. Como se mencionó anteriormente, la bahía es modificada a través de recolocaciones de los contenedores que se encuentren en la parte superior de una pila a otra. Se denota dicho movimiento de la pila  $s$  a la pila  $s^*$  como  $m = (s, s^*)$  donde  $s, s^* = 1, \dots, S$ .

El contenedor  $g$  localizado en  $L_{s,h}$  se dice verticalmente bloqueado si existe otro contenedor  $g^*$  en  $L_{s,h'}$  con  $h < h'$  (contenedor  $g^*$  está por encima de  $g$ ) y  $g < g^*$ . A la pila  $s$  se le llama v-perfecta si no existen contenedores en esta pila verticalmente bloqueados.

Una solución a esta bahía está dada por la secuencia de movimientos  $\sigma = \{m_1, m_2, \dots, m_k\}$  con longitud igual a  $k$ .

## 5 Descripción del Límite Inferior

Para resolver exitosamente el Pre-Marshalling Problem es necesario cumplir con dos claros objetivos. Primero; transformar una bahía mal configurada de manera que los contenedores de mayor prioridad no tengan obstáculos para ser extraídos. En segundo lugar, esta transformación se realiza por una serie de recolocaciones, que a su vez debe ser optimizada de tal forma que se obtenga la mínima secuencia de movimientos posibles y, además, éstos tengan la menor cantidad de recolocaciones improductivas.

A partir de esto último, dentro de la búsqueda de la mínima secuencia de movimientos para lograr una configuración válida, se debe tener presente aquellos conceptos necesarios para la optimización. Es así, como la búsqueda de un mínimo a mejorar es de vital relevancia, ya que entrega una referencia inicial con parámetros tangibles. Además, para los propósitos de adaptación del FOA a este problema, el límite inferior entrega un estándar base a cumplir que proporciona una referencia a la búsqueda.

Debido a esto es que en el PMP, un límite inferior eficaz nos entrega la longitud mínima de la serie de recolocaciones, que permiten transformar una configuración inicial de contenedores. La conducción a una bahía ordenada se da, como se dijo anteriormente, por el orden de los índices de prioridad (llamados grupos  $g$ ), lo que implica que en cada pila los contenedores con índice  $g$  estén ordenadas de forma descendente.

El número que indica la cantidad de recolocaciones necesarias para lograr transformar una bahía mal configurada está dado, según lo descrito en el artículo [9], por tres fases principales. Estas fases están dadas por los tipos de movimientos existentes en una bahía (cuatro tipos de recolocaciones). El resultado de este cálculo es un número entero, dicho valor se ocupará dentro de la investigación para la búsqueda de una solución óptima dada una bahía en específico. A continuación se describe el método que permite el cálculo de este valor.

### 5.1 Definiciones Límite Inferior

En esta sección se pretende entregar la definición formal del límite inferior, sus variables claves, así como los conceptos que están relacionados con ella. Los valores intrínsecos para el cálculo del largo de la secuencia de movimiento son descritos a continuación junto con su rol en el resultado final.

Todos aquellos datos utilizados en esta investigación, con el fin de hallar la mínima secuencia de recolocaciones, están respaldados por lo descrito en el artículo [9]. Las tres fases que son necesarias para la conjetura del límite inferior toma en cuenta, principalmente, los tipos de movimientos que pueden ser realizados a partir de una bahía inicial. Primero que todo, se describen ciertos términos claves que se manejan en este método y que ayudan a la comprensión y cálculo del límite inferior:

- Sea  $L_{s,h} = g > 0$ . Se le dice mal posicionado al elemento del grupo  $g$  ubicado en ranura  $(s, h)$ , si  $h > 1$  y el elemento ubicado en  $L_{s,h-1} < g$ , o si hay un artículo mal posicionado en una ranura  $(s, h')$ ,  $1 < h' < h$ . De lo contrario el elemento de grupo  $g$  en la ranura  $(s, h)$  está bien situado.
- Una ranura  $L_{s,h}$  es una ranura de suministro potencial de grupo  $g$  si y sólo si (i) la pila  $s$  no está vacía y el contenedor bien ubicado en la parte más alta de la pila es igual a  $g$  y (ii) la ranura  $L_{s,h}$  se encuentra por encima del último contenedor bien situado en la pila  $s$ .

En particular, y como se introdujo anteriormente, existen distintos tipos de movimientos que pueden llevarse a cabo en una bahía. Su existencia dependerá de la disposición inicial de los contenedores, pero en definitiva se obtienen dos movimientos principales. El primero es denotado por B (*BAD*) que es un movimiento que se le realiza a un contenedor mal ubicado dentro de una bahía, es decir, que existe bajo este contenedor uno de mayor prioridad. El segundo es denotado por G (*GOOD*) que es un movimiento que se le aplica a un contenedor que está bien ubicado dentro de la bahía, es decir, que todos los contenedores bajo él son de una prioridad más baja. Es así como se obtiene lo siguiente:

- BB: implica mover un contenedor mal ubicado a otra pila, quedando igualmente mal posicionado.
- BG: implica mover un contenedor mal ubicado a otra pila, quedando bien posicionado en relación a los contenedores que estén debajo de él.
- GB: implica mover un contenedor bien ubicado a otra pila, quedando inmediatamente mal ubicado por los contenedores que queden bajo él.
- GG: implica mover un contenedor bien ubicado a otra pila, quedando igualmente bien posicionado.

En general, los movimientos se clasifican en BX que puede ser un movimiento BB o bien BG. De forma análoga se definen y clasifican los movimientos GX. Dicho lo anterior, es el momento de definir las variables necesarias para el entendimiento y desarrollo del límite inferior. La tabla 5.1 ilustra las variables y definiciones usadas para este procedimiento.

Las tres fases para el cálculo del límite inferior se logran con los datos entregados en la tabla 5.1. En primer lugar, está el cálculo del límite o cota inferior para el número de movimientos BX necesarios para llegar a un diseño final válido. En segundo lugar, está el cálculo de la cota inferior para el número de movimientos GX, necesarios para llegar a una configuración válida y, en último lugar, está el cálculo de todos los movimientos necesarios para llegar a una disposición válida, la cual es la suma del primer y segundo punto. Razones por las cuales se definen los siguientes puntos:

- $n'_{BX} = n_b + \min\{n_b(s)/s = 1, \dots, S\}$ . Cota inferior para los movimientos BX.
- $n'_{GX} = \sum_{s=1}^{n(S,GX)} n^{g^*}(s) / D_s(g^*) \geq D_s(g)$  para  $g = 1, \dots, G$ . Cota inferior para los movimientos GX.

- $n'_m = n'_{BX} + n'_{GX}$ .

Tabla 5. 1 Términos y definiciones para el límite inferior

Término	Variable	Definición
Pila	$S$	Cantidad de pilas existentes en la bahía.
Nivel	$H$	Altura final de la bahía
Grupo $G$	$G$	Cantidad de prioridades existentes en una bahía. $G = \{1, \dots, G\}$
-	$n_b$	Número de contenedores mal posicionados en la bahía.
-	$n_b(s)$	Número de contenedores mal posicionados en la pila $s$ .
-	$n^g(s)$	Número de contenedores bien posicionados en la pila $s$ .
Demanda del grupo $g$	$d(g)$	Número de contenedores mal posicionados del grupo $g$ .
Demanda acumulada del grupo $g$	$D(g)$	$D(g) = d(g) + d(g + 1) + \dots + d(G)$
Oferta potencial del grupo $g$	$s_p(g)$	Número de todos los espacios que podrían servir para almacenar al grupo $g$ .
Oferta potencial acumulada del grupo $g$	$Sp(g)$	$Sp(g) = s_p(g) + s_p(g + 1) + \dots + s_p(G) + H \times n_{s,empty}$ ; donde $n_{s,empty}$ Es el número de pilas vacías de la bahía.
Demanda acumulada por exceso del grupo $g$	$D_s(g)$	$D(g) - Sp(g)$
Oferta limpia del grupo $g$	$s_c(g)$	Número de todos los espacio de oferta vacía del grupo $g$ .
Oferta limpia de la bahía	$S_c(g)$	Es la suma ponderada de; $10^0 s_c(1) + 10^1 s_c(2) + \dots + 10^{G-1} s_c(G)$

## 5.2 Ejemplo uso del límite inferior

Se toma como ejemplo la bahía utilizada en [9] la cual cuenta con una altura de  $H=4$  y una cantidad  $S=6$  de pilas. Esto se ilustra en la figura 5.1, y los datos obtenidos de dicha bahía que son necesarios para el cálculo del límite inferior son descritos en la tabla 5.1.

Niveles	1					3
	4				3	3
	2	4	3	2	4	3
	2	3	1	1	1	1
	Pilas					

Figura 5. 1 Bahía inválida

La figura 5.1 muestra una bahía desordenada, la cual posee cuatro grupos en total ( $G=1,\dots,4$ ). En ella se puede observar, en color más oscuro, los contenedores bien ubicados para cada pila, al contrario del tono más claro, el cual ilustra los contenedores mal ubicados. A partir de esto se obtiene, por ejemplo, la demanda del grupo tres (3), que es igual a 5. A continuación se describe la forma en que se obtienen los datos detallados en la tabla 5.2.

Tabla 5. 2 Valores de demanda y oferta para los contenedores de la bahía

Grupo	Demanda $d(g)$	Demanda acumulada $D(g)$	Oferta Potencial $s_p(g)$	Oferta Potencial Acumulada $Sp(g)$	Exceso de Demanda acumulada $D_s(g)$
4	3	3	0	0	3
3	5	8	3	3	5
2	1	9	2	5	4
1	1	10	12	17	-7

En forma general, la tabla 5.2 se llena de forma desde la parte superior a la inferior. Así, la primera columna corresponde a los grupos  $g$  presentes en la bahía ordenadas de menor a mayor prioridad. Para el resto de los datos se requiere de mayor análisis de la bahía.

La segunda columna correspondiente a la demanda del grupo  $g$ , que indica la cantidad de contenedores mal posicionados para cada índice  $g$  en la bahía. Por ejemplo, y como ilustra la figura 5.2, la demanda para el grupo  $g=4$  es igual a 3, ya que existen tres contenedores mal posicionados para este grupo (se cuentan los contenedores que se encuentran en la parte más clara de la bahía). Con esto, se calcula también la tercera columna; que es la suma de las demandas de un grupo  $g$  y de los precedentes.

	1					3
	4				3	3
Niveles	2	4	3	2	4	3
	2	3	1	1	1	1
	Pilas					

Figura 5. 2 Ejemplo de obtención demanda para el grupo  $g=4$

En la cuarta columna se cuentan todos aquellos lugares que son oferta o espacios potenciales para un grupo  $g$ . Dicho de otra manera, un espacio es oferta potencial del grupo  $g$  si y sólo si:

- La pila  $S$  no está vacía.
- El último contenedor bien posicionado de la pila  $S$ , sea del grupo  $g$ .

La figura 5.3 servirá como ejemplo para ilustrar la forma en que se obtiene estos valores. En la tabla 5.2 se tiene un  $S_p(3) = 3$ , lo cual es resultado de contar los espacios potenciales para este grupo en la bahía. Primero se ubican aquellos contenedores del grupo  $g$  (en este caso, el grupo  $g=3$ ) que estén bien posicionados o, como se muestra en la figura 5.3, en un color más oscuro. Dentro de esta bahía se tiene sólo un contenedor bien posicionado para este grupo y se encuentra en la pila  $S=2$  y se cumple, además, que es el último bien posicionado. Ahora sólo queda numerar los espacios hacia arriba de esa pila, que son en total 3, por tanto;  $S_p(3) = 3$ . Sin embargo, si existen pilas vacías, a



- 3) Por último, el  $n'_m$  es consecuencia de sumar los dos anteriores resultados. Así,  $n'_m = 11 + 2 = 13$ , siendo éste el límite inferior para esta bahía.

Niveles	1					3
	4				3	3
	2	4	3	2	4	3
	2	3	1	1	1	1
	Pilas					

Figura 5. 4 Elementos bien posicionados con índice menor al  $g^*=3$ .

Es así como se tiene una manera sencilla para calcular un número mínimo de relocalizaciones a realizar y un parámetro claro para el inicio de la optimización y resolución del PMP.

## 6 Técnica Utilizada

El Pre-Marshalling Problem puede ser abordado y resuelto mediante variadas técnicas, ya que su complejidad no ha permitido soluciones totalmente factibles y/u óptimas. Cada método utilizado enfoca su búsqueda de buenas relocalaciones de disímiles maneras, generando resultados acorde a lo esperado. Además, donde la optimización juega un rol clave para mejorar la eficiencia del terminal portuario, las variables a utilizar por estos métodos inciden directamente en la obtención de una secuencia mínima de reubicaciones o relocalaciones.

A partir de lo anterior se propone el uso del algoritmo de la mosca de la fruta (FOA) para la resolución de este problema. Este algoritmo ha sido destacado por su adaptabilidad a diversos problemas, debido a que proporciona las herramientas para una amplia búsqueda de óptimos globales. Es así como, junto con el límite inferior y una adaptación del FOA, se propone una resolución al problema de pre-ordenamiento de contenedores.

### 6.1 Concepto básico de Fruit Fly Optimization Algorithm

FOA, como es comúnmente conocido este algoritmo, es un método basado en la forma en que las moscas de la fruta buscan y encuentran su comida. Éstas tienen órganos sensibles que les brindan una capacidad superior en comparación a otras especies. Uno de los principales métodos de búsqueda de las moscas es a partir del olfato que es capaz de oler una fuente de alimento a más de 40 kilómetros de distancia. Luego de hallar su comida, se dirige hacia ella y gracias a su visión, el resto de moscas del grupo puede dirigirse hacia ese lugar también y reunirse con sus compañeras. La forma de esta búsqueda es ilustrada en la figura 6.1 donde a partir de una cierta posición inicial, pueden realizar un forrajeo para encontrar su comida.

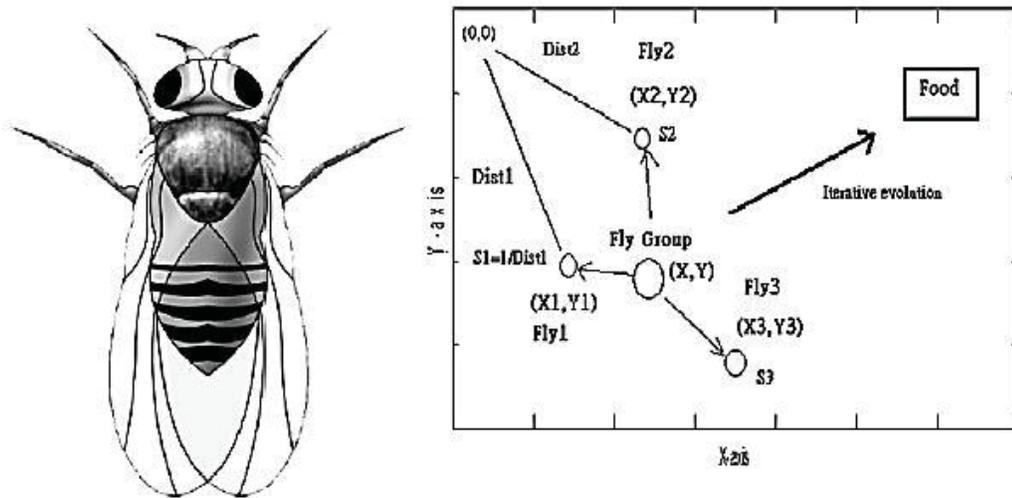


Figura 6.1 Búsqueda de la comida por parte de un grupo de moscas.

Estas características de forrajeo, utilizando órganos tan sensibles como su olfato y visión han sido resumidos y programados de la siguiente forma de acuerdo a Pan [2]:

- Se inicia aleatoriamente la posición del grupo de moscas (por ejemplo, el Fly Group graficado en la figura 6.1).

$$Init X_{axis}; Init Y_{axis}$$

- A cada mosca del grupo se le asigna una dirección y distancia de forma aleatoria, de manera que puedan buscar el alimento con su órgano olfativo.

$$X_i = X_{axis} + RandomValue$$

$$Y_i = Y_{axis} + RandomValue$$

- Debido que la distancia al alimento es desconocida, se estima primero el trayecto al origen, para luego calcular el valor de juicio de concentración ( $S$ ) que es inverso a la distancia ( $Dist$ ).

$$Dist_i = \sqrt{X_i^2 + Y_i^2} ; S_i = \frac{1}{Dist_i}$$

- Posteriormente, se sustituye los valores de juicio de concentración ( $S$ ) en la función de juicio de concentración de olor para obtener la concentración de olor ( $Smell$ ) de cada mosca del enjambre.

$$Smell_i = Function(S_i)$$

- Luego se identifica la posición de la mosca de la fruta que contenga la mayor concentración de olor

$$[bestSmell, bestIndex] = \max(Smell)$$

- Se toma la posición que contenga la mejor concentración de olor y se le indica a las moscas que vuelen a dicho lugar.

$$SmellBest = bestSmell$$

$$X_{axis} = X_{bestIndex}$$

$$Y_{axis} = Y_{bestIndex}$$

- A partir de los datos detallados anteriormente, se ingresa a un ciclo iterativo, el cual se detendrá hasta que encuentre una mejor concentración de olor.

Los pasos detallados anteriormente son una base conceptual para lograr implementar una solución, sin embargo, poseen algunas diferencias de aplicación a la hora de llevar la estrategia de las moscas al objetivo de este trabajo, el Pre-Marshalling

Problem. Esto particularmente, porque las moscas en el PMP, no pueden ser tomadas como valores en solitario, sino que como un conjunto de valores que representarán recolocaciones, lo que implica, además; que el proceso investigativo de este documento no concluye del todo.

## 6.2 Implementación al problema

Llegado a este punto y como se describe antes, el algoritmo de la mosca de la fruta presentado anteriormente es un método conceptual, ya que puede ser drásticamente modificado, pero manteniendo su esencia de búsqueda. Tanto es así que Wang en el artículo [7] propone un algoritmo de la mosca de la fruta binaria para resolver un problema de asignación.

Éste último es la base para el algoritmo propuesto para la resolución al problema del pre-ordenamiento. La estructura de este nuevo algoritmo junto con algunas referencias del documento [9], colaboran para la adaptación del FOA para la resolución del PMP.

Para dar una respuesta que cumpla con las necesidades del terminal, se tomará la bahía como una matriz de  $S$  por  $H$  (*stack* y *height*). En ella se distribuirán todos los contenedores con sus respectivas prioridades o índices de salida. Esta información será cargada desde un archivo que contendrá, además del número de pilas y la altura de la bahía e índices de los contendores, un identificador de la bahía y la cantidad total de contenedores que posee dicha bahía o instancia. A continuación, la tabla 6.1 bosqueja la situación antes descrita, utilizando los datos ilustrados en la figura 5.1.

Tabla 6. 1 Ejemplo de archivo contenedor de bahía

bay	cpmp_16_5_48_10_29_1.bay
Stacks	6
Height	4
Containers	17
Stack 1	2 2 4 1
Stack 2	3 4
Stack 3	1 3
Stack 4	1 2
Stack 5	1 4 3
Stack 6	1 3 3 3

## 6.3 Adaptación FOA

Anteriormente se había descrito una forma básica del algoritmo de la mosca de la fruta los cuales para los propósitos de esta investigación son insuficientes. Es por esto que, siguiendo el método descrito por [7], se presentará en esta primera instancia la adaptación del FOA para resolver el Pre-Marshalling Problem.

En primer lugar y a diferencia del FOA original, en el FOA para PMP adoptamos una matriz dinámica con tamaño inicial de  $2 \times n_m$  (número obtenido por el cálculo del límite inferior) para representar la solución, es decir, la secuencia de recolocaciones. Este vector que, si bien tendrá un tamaño inicial dado por un valor, crece hasta encontrar una solución al problema de reordenamiento de contenedores para una instancia en específico.

Segundo, [7] propone la utilización de un vector de probabilidades para la generación de la población de moscas, la cual fue usada en anteriores entregas. Sin embargo se dejó de utilizar debido a su discutible utilidad. Esto último no quiere decir que a futuro se pueda utilizar, de similar u distinta manera, en esta investigación.

### 6.3.1 Inicialización de la población

En este FOA para PMP, cada mosca es una solución representada por una matriz de 2 por  $N$  (donde  $N$  es igual al número de recolocaciones necesarias para llegar a una configuración válida). Cada movimiento, como sabemos, puede ser BB, BG, GB y GG. Según esto y con la premisa de optimizar la secuencia de movimientos se debe, a su vez, tratar de reducir al mínimo las recolocaciones improductivas. A partir de lo anterior, cada reubicación tendrá una probabilidad de cambio que dependerá de las características de la bahía a estudiar. Esto se explica con mayor énfasis más adelante, por ahora, las probabilidades de cambio de cada recolocación siguen la siguiente regla:

- Para bahías con una altura mayor al número de pilas los porcentajes son los siguientes:
  - Recolocación del tipo BG tiene posibilidad de cambio o de realización igual a 0.25.
  - Recolocación del tipo BB tiene posibilidad de cambio o de realización igual a 0.25.
  - Recolocación del tipo GG tiene posibilidad de cambio o de realización igual a 0.25.
  - Recolocación del tipo GB tiene posibilidad de cambio o de realización igual a 0.25.
- En caso de que el número de pilas sea mayor a los niveles existentes en la bahía en cuestión, los porcentajes serán los siguientes:
  - Recolocación del tipo BG tiene posibilidad de cambio o de realización igual a 0.55.

- Recolocación del tipo BB tiene posibilidad de cambio o de realización igual a 0.20.
- Recolocación del tipo GG tiene posibilidad de cambio o de realización igual a 0.15.
- Recolocación del tipo GB tiene posibilidad de cambio o de realización igual a 0.10.

Cada mosca  $F_i^d$  (donde  $i = 1, \dots, FN$  y  $d = 1, \dots, N$ ) es generada por esta serie de probabilidades y variará su tamaño dependiendo de la generación en que se encuentre. Por tanto en una primera instancia; todas las recolocaciones tienen igual probabilidad de existir, favoreciendo la aleatoriedad en la búsqueda inicial, es decir:

- $F_i^d = \text{Rand}(BB, GG, BB, GB)$ .

En la siguiente iteración, las moscas ( $FN$ ) se generan utilizando la mejor mosca encontrada en dicha iteración y la probabilidad de cambio de sus recolocaciones. Además, para efecto de futuras pruebas, cada recolocación en el vector de solución de una mosca tendrá un identificador, con el fin de reconocer de mejor manera aquellos movimientos que deban ser sustituidos en cada generación.

### 6.3.2 Búsqueda Basada en Olor y Visión Local

Se crearán vecindarios para ampliar la búsqueda local de los óptimos, es así como un vecindario de moscas llamado  $FS$  tiene  $FN$  moscas de la fruta. A partir de lo extraído en [8], el número de barrios a utilizar en este trabajo varía entre 2 y 10. Luego, la búsqueda basada en olor local se trata de (luego de inicializar todas las moscas) encontrar al mejor vecino del vecindario para posteriormente, utilizando la búsqueda por visión, igualar las moscas a la mejor del barrio. Sin embargo, antes de realizar la búsqueda de visión local, se deben evaluar mediante la función objetivo (Fitness o Smell Function) cada mosca del vecindario para comprobar que sean soluciones válidas. De no ser así, se debe ampliar el tamaño de la mosca hasta que cumpla con los criterios dados.

### 6.3.3 Búsqueda Basada en Visión Global

Este paso incluye la búsqueda de la mejor mosca global y la generación de las nuevas moscas para la siguiente iteración. La mejor mosca de la iteración o generación  $gen$  será la que tenga el menor número de recolocaciones. Luego, la generación de nuevas moscas está dada por las recolocaciones y las probabilidades de cambio que tiene dicha mejor mosca.

### 6.3.3 Operación de Reparación

A partir de los métodos descritos en las secciones anteriores, es necesario hacer hincapié en que los resultados obtenidos por cada generación (nueva generación de soluciones o moscas), pueden estar fuera de lo esperado debido a los factores de cálculo que éstos implican.

Referente a las nuevas soluciones generadas y a la dificultad del problema, la posibilidad de que éstas no sean soluciones válidas es alta. Debido a esto, se debe emplear un operador de reparador para tres casos significativos, el primero es; la utilización de recolocaciones inválidas (por ejemplo, sobrepasar la altura de una bahía), en segundo lugar; que los nuevos movimientos tengan un identificador distinto al tipo de recolocación que representan y en tercer lugar; corroborar que estas soluciones sean válidas, es decir; que logren una bahía con una configuración final óptima.

Para este último caso, y con el fin de mantener los criterios de búsqueda de una disposición final legítima para una bahía, se seleccionan tantas recolocaciones (de manera aleatoria) como sean necesarias para llegar a esta disposición final. Sin embargo, se le da mayor porcentaje de aparición a los movimientos del tipo BG.

### 6.3.5 Variables Utilizadas

Este método utilizado tiene ciertas variables que son manejadas y alteradas constantemente conforme se desarrolla el algoritmo. Para tener un mejor conocimiento de esto, la tabla 6.2 describe con detalle estas variables.

Tabla 6. 2 Principales variables para el desarrollo del FOA para PMP

Variable	Definición
Sizepop	Tamaño del vecindario de moscas. Está dado, en esta oportunidad, por el porcentaje de ocupación de la bahía y representado por la variable $FN$ . A mayor porcentaje de ocupación, menor población.
Sizefly	Tamaño del vector (mosca) dada por el límite inferior. Cabe decir que cada mosca es dinámica, por lo que el tamaño es sólo una referencia inicial.
Maxgen	Es el número máximo de iteraciones para realizar la búsqueda global. Todas las pruebas y resultados descritos más adelante fueron realizados con 15 mil iteraciones.

Otras variables o conceptos clave están dados por la metaheurística y se describen a continuación:

- Solución inicial: Consta de una serie de recolocaciones “aleatorias” por cada mosca en una primera instancia que conlleven una solución al PMP. Estas recolocaciones, que si bien pueden ser realizadas al azar, se preocupa de buscar movimientos productivo antes que uno improductivo, es decir; se busca cualquier reubicación BG a realizar, de no ser posible se busca cualquier movimiento GG y así sucesivamente.

- **Recolocación:** Como ha descrito antes, una recolocación está dada por la dupla  $(s,s')$ , donde  $s$  es la pila de origen y  $s'$  es la pila destino. Estas deben cumplir con las restricciones propias del PMP, por ejemplo, estar dentro de los límites dados por  $S$  y  $H$ . Téngase en cuenta que una mosca está conformada por un conjunto de duplas del tipo  $(s,s')$ .
- **Grupo de Moscas:** El grupo de moscas está dividido por vecindarios y cada uno de estos está conformado por una lista de nodos enlazados, donde en cada nodo se encuentra una mosca. El objetivo de esto, para bahías grandes con alto número de contenedores, es poder utilizar tamaños variables de la población, según iteración lo que se espera lleve a una búsqueda con mayor y mejores resultados.
- **Smell Function:** Conocida principalmente como Fitness, evaluará la calidad de la solución conseguida en la búsqueda por olor local. De forma tal de obtener soluciones acorde a las necesidades del problema.
- **Criterio de parada:** Principalmente se toma el criterio de parada como el momento en que se encuentre una solución óptima válida y que su largo (cantidad de recolocaciones) sea menor al límite inferior (en lo posible). Sin embargo, también se considera como criterio de parada el valor de Maxgen que es el número máximo de iteraciones o tope.

## 7 Evaluación

Para comparar el rendimiento de la adaptación del FOA al problema del pre-ordenamiento de contenedores y con el fin de obtener las conclusiones y futuras nuevas líneas de trabajo, se utilizaron los datos entregados y estudiados en [10] con algoritmos que han sido propuestos hasta ahora en la literatura. Los casos de prueba a estudiar se dividen en dos categorías (CV y LC) cuyo detalle y características se entregan a continuación. Todas las pruebas se realizaron con 10 ejecuciones del programa y 15 mil iteraciones dentro de él.

Ambos casos de prueba en estudio (LC y CV) presentaron una baja variabilidad en los resultados obtenidos dentro de las 10 ejecuciones. Esto queda demostrado por la desviación estándar resultante de cada instancia. Así, por ejemplo, el caso de LC, la desviación estándar tiene valores entre 0.52 y 2.78 (valores aproximados) del promedio. Por otro lado, los casos CV tienen valores entre 0.0 y 2.90.

### 7.1 Experimentos con LC

En esta entrega, las pruebas se realizaron en comparación con los métodos expuestos por Lee y Chao en [5] y con el resultado entregado por el límite inferior. Así, [5] propone a LC1, LC2 y LC3 (estas dos modificadas por Bortfeldt y Forster [10] por falta de datos sobre los contenedores mal posicionados, generando LC2a, LC2b, LC3a y LC3b) como casos de prueba, donde el primero posee una instancia, mientras que las siguientes poseen 10 instancias cada una. Los resultados obtenidos por cada caso de prueba son comparados con LC por Lee y Chao, BF por Bortfeldt y Forster y TS (Tabu Search) por Marcelo Matus, son detallados en la tabla 7.1.

Como se detalló anteriormente, los porcentajes de cambio en las recolocaciones dependerán del número de niveles y pilas que tengan las instancias. En el caso de los casos de prueba LC, el número de pilas es mayor a la altura de estos (componentes de cada instancia detallados en la tabla A.1, en el anexo), por lo que los esfuerzos se avocan a encontrar la mayor cantidad de movimientos BG. Aunque no se pudo llegar al óptimo en estas instancias, los datos obtenidos se acercan bastante y son un resultado satisfactorio para la presente investigación.

Tabla 7. 1 Tabla comparativa resultados para LC (parte 1).

Caso de Prueba	Caso de Prueba	Número de recolocaciones			
		LC	BF	TS	FOA
LC1	15.00	31.00	17.00	15.00	19.00
LC2a	22.60	61.00	22.60	25.10	31.70

Tabla 7. 2 Tabla comparativa resultados para LC (parte 2).

Caso de Prueba	Caso de Prueba	Número de recolocaciones			
		LC	BF	CV	FOA
LC2b	37.20	61.00	38.40	47.40	48.30
LC3a	23.70	75.80	23.70	30.60	47.00
LC3b	39.80	75.80	42.70	73.00	70.50

## 7.2 Experimentos con CV

Los siguientes casos de pruebas cuentan con 10 instancias cada uno y presentan mejoría respecto a los resultados obtenidos, en comparación con los casos de prueba anteriores. A diferencia de las anteriores instancias, éstas presentan un número de niveles mayor al número de pilas, lo que implica que en la totalidad de las iteraciones se buscará, en igual medida algún tipo de recolocación que lleve a una configuración final válida.

La tabla 7.2 muestra la comparativa de resultados obtenidos para estos casos, en donde se evidencia una cercanía a los mejores resultados.

Tabla 7. 3 Tabla comparativa de resultados para CV

Caso de Prueba	Promedio del Límite Inferior	Número de recolocaciones			
		CV	BF	TS	FOA
CV1	7.90	21.30	10.50	10.10	10.10
CV2	13.60	28.27	19.10	21.10	25.80
CV3	21.50	49.61	30.40	36.30	42.50
CV4	28.60	50.14	44.40	60.50	67.50

Los resultados descritos en la columna FOA, corresponden al promedio de los mejores valores obtenidos en cada ejecución para ese caso de prueba en particular. Como se puede evidenciar en el detalle de los resultados obtenidos presentados en el capítulo de anexos, la variabilidad entre ejecución y ejecución es considerablemente baja. Esto último implica que la adaptación del algoritmo de la mosca de la fruta al problema del Pre-Marshalling es confiable y si bien, tiene un importante componente aleatorio en la obtención de resultados y generación de éstos, no significa que los resultados sean dados al azar.

## 8 Conclusiones

Como resultado de la investigación sobre el Pre-Marshalling problem, es posible concluir que sus características intrínsecas han sido comprendidas a cabalidad. Esto permite, entre otras cosas, poder lograr una solución sin caer en errores de concepto. A su vez, lograr ciertos resultados que puedan generar nuevas líneas de estudio a través, por ejemplo; de la modificación de las probabilidades de cambio, número de vecindarios y moscas, etc. Todo lo anterior implica siempre encontrar el límite inferior de la manera más eficiente posible, ya que es una pieza clave en la búsqueda de soluciones que contengan el menor número de recolocaciones posible.

Sobre la metaheurística utilizada, el algoritmo de optimización de la mosca de la fruta, gracias a la literatura existente y el trabajo de sus creadores demostró ser, con éxito, adaptable. Es capaz de encontrar una solución acorde a lo exigido, aunque en algunos casos y en sentido crítico, insuficiente. Esto en el sentido de la optimización por estar aún lejos de lo esperado, si bien en algunos casos se ha alcanzado un óptimo. Por lo tanto, este algoritmo puede aún ser refaccionado con tal de perfeccionar sus resultados, tanto a nivel de la problemática como a nivel computacional.

En líneas generales, la conjunción del límite inferior a la investigación es crucial. Esta última permite de gran forma indicar qué tan próximo se está de una solución óptima. No obstante, es necesario mejorar la implementación de este método, debido a que las partes que lo componen son relevantes a la hora de llegar a una conclusión durante esta investigación, no sólo a nivel de resultados. Además, dichas partes del límite inferior pueden ser parte importante a la hora de obtener estadísticas de resolución y, por lo tanto, otorgar datos importantes para este estudio.

Finalmente, luego de evidenciar un progreso en los resultados, en comparación con entregas y estudios anteriores, se confirma aún más; que el FOA es una metaheurística altamente adaptable y competente. Si bien algunas instancias no fueron resueltas dentro de los parámetros esperados, queda abierta a nuevas líneas investigativas que podrían permitir la obtención de resultados significativamente mejores.

Es así como métodos de este tipo proveen de herramientas flexibles para resolver la problemática del pre-ordenamiento. Si bien el FOA no puede resolver de manera óptima un número determinado de casos, diferentes metaheurísticas como esta permitirán una resolución eficaz del Pre-Marshalling problem. De este modo, permitiendo que exista un cambio importante en el manejo portuario al entregarles herramientas eficaces para la administración de los contenedores que deben ser embarcados.

## 9 Referencias

- [1]. S. Voß, “Extended Mis-overlay Calculation for Pre-marshalling Containers”. *Computational Logistics*. Vol. 7555, pages 86-91, 2012.
- [2]. W. Pan, “Fruit Fly Optimization Algorithm, Second Edition”, *Knowledge-Based Systems*. Vol. 26, pages 69-74, 2012.
- [3]. M. Caserta, S. Schwarze and S. Voß, “A New Binary Description of the Blocks Relocation Problem and Benefits in a Look Ahead Heuristic”, *Evolutionary Computation in Combinatorial Optimization*. Vol. 5489, pages 37-48, 2009.
- [4]. M. Caserta and S. Voß, “A corridor method-based algorithm for the pre-marshalling problem”, *Proceedings of the EvoWorkshops*. Vol. 5448, pages 788-797, 2009.
- [5]. Y. Lee and S. Chao, “A neighborhood search heuristic for pre-marshalling export containers”, *European Journal of Operational Research*. Vol. 196, pages 468–475, 2009.
- [6]. A. Tus, A. Rendl and G. Raidl, “Metaheuristics for the Two-Dimensional Container Pre-Marshalling Problem”, *Learning and Intelligent Optimization*. Vol. 8994, pages 186-201, 2015.
- [7]. L. Wang, X. Zheng and S. Wang, “A novel binary fruit fly optimization algorithm for solving the multidimensional knapsack problem”, *Knowledge-Based Systems*. Vol. 48, pages 17–23, 2013.
- [8]. C. Expósito, B. Melián and M. Moreno, “Una heurística eficaz para problemas de pre-marshalling en una terminal marítima de contenedores”, *Dpto. de Estadística, I.O. y Computación. Escuela Técnica Superior de Ingeniería Informática. Universidad de La Laguna*. 2012.
- [9]. B. Crawford, R. Soto, C. Torres-Rojas, C. Peña, M. Riquelme-Leiva, S. Misra, F. Johnson and F. Paredes, “A Binary Fruit Fly Optimization Algorithm to Solve the Set Covering Problem”, *Computational Science and Its Applications*. Vol. 9158, pages 411-420, 2015.
- [10]. A. Bortfeldt and F. Forster, “A tree search procedure for the container pre-marshalling problem”, *European Journal of Operational Research*, Vol. 217, pages 531–540, 2012.
- [11]. W. Pan. “A new Fruit Fly Optimization Algorithm: Taking the financial distress model as an example”, *Knowledge-Based Systems*. Vol. 26, pages 69–74, 2012.
- [12]. P. Zhang and L. Wang, “An Improved Fruit Fly Optimization Algorithm for Solving Lot-streaming Flow-shop Scheduling Problem”, *Journal of Donghua University*. Vol. 31, pages 165-170, 2014.
- [13]. S. Lin,” Analysis of service satisfaction in web auction logistics service using a combination of Fruit fly optimization algorithm and general regression neural network”, *Neural Computing and Applications*. Vol. 22, pages 783–791, 2013.
- [14]. J. Han, P. Wang and X. Yang, “Tuning of PID Controller Based on Fruit Fly Optimization Algorithm”, In *International Conference on Mechatronics and Automation*, pages 5 - 8, 2012.
- [15]. S. Huang and T. Lin, “Heuristic algorithms for container pre-marshalling problems”, *Computers & Industrial Engineering*, Vol. 62, pages 13–20, 2012.

- [16]. M. Caserta, S. Schwarze, and S. Voß, “Container Rehandling at Maritime Container Terminals”, Operations Research/Computer Science Interfaces Series, Vol. 49, pages 247-269, 2011.
- [17]. R. Stahlbock and S. Voß, “Operations research at container terminals: A literature update”, OR spectrum, Vol. 30, pages 1-52, 2008.
- [18]. C. Expósito, B. Melián and M. Moreno, “Heuristic algorithms for container pre-marshalling problems”, Expert Systems with Applications, Vol. 39, pages 8337–8349, 2012.

## Anexos

### A. Detalles de los Casos de Prueba

Tabla A. 1 Detalle de los casos de prueba LC

Caso de prueba	N° de instancias	N° de pilas	Altura de la pila	N° de grupo de cont.	N° de contenedores	N° de contenedores mal posicionados
LC1	1	10	5	10	35	13
LC2a	1	12	6	10	50	19
LC2b	10	12	6	10	50	35
LC3a	10	12	6	10	54	21
LC3b	10	12	6	10	54	36

Tabla A. 2 Detalle de los casos de prueba CV

Caso de Prueba	No. de Instancias	No. de Pilas	Altura de la Pila	No. de contenedores	No. de índices de prioridad	No. de contenedores mal ubicados
CV1	10	3	9	9	9	5,40
CV2	10	4	16	16	16	10,60
CV3	10	5	25	25	25	17,50
CV4	10	6	36	36	36	26,50

### B. Detalle Resultados Obtenidos

Las siguientes tablas presentan el detalle de los resultados obtenidos para cada instancia de cada caso. La primera columna indica el caso de prueba, la segunda; la instancia, luego; desde la tercera a la séptima columna se detallan los resultados obtenidos según ejecución (con  $E = 1, \dots, 10$ ). La octava columna indica el promedio de los resultados obtenidos y la última columna, indica la desviación estándar de los resultados obtenidos en las 10 ejecuciones.

La tabla B.1 a B.3 muestran los resultados de las 10 ejecuciones realizadas con los 2 primeros casos de prueba propuestos en CV. La primeras dos tablas detallan las primeras 5 ejecuciones, mientras que la última detalla las faltantes.

Tabla B. 1 Detalle de los resultados obtenidos según ejecución para CV (1).

Caso	Instancia	E1	E2	E3	E4	E5	Prom.	D.E
CV1	CV1_1.bay	14	12	15	12	14	13.10	1.10
	CV1_2.bay	8	8	8	8	8	8.0	0.0
	CV1_3.bay	13	13	13	14	14	13.30	0.48
	CV1_4.bay	11	12	11	13	11	11.80	0.92
	CV1_5.bay	13	15	14	13	13	14.0	0.94
	CV1_6.bay	8	9	8	8	8	8.20	0.42
	CV1_7.bay	8	8	8	8	8	8.10	0.32
	CV1_8.bay	11	12	11	11	12	11.40	0.51
	CV1_9.bay	9	10	10	10	9	9.50	0.53
	CV1_10.bay	11	8	8	10	11	9.70	1.25
CV2	CV2_1.bay	23	25	24	25	25	24.80	0.92
	CV2_2.bay	25	24	24	23	22	23,60	1.07
	CV2_3.bay	20	20	19	19	19	19.20	0.79
	CV2_4.bay	25	28	24	25	28	25.50	1.78
	CV2_5.bay	29	31	29	31	32	30.40	1.26
	CV2_6.bay	27	27	24	25	25	25.50	1.08
	CV2_7.bay	32	31	31	31	31	31.90	0.88
	CV2_8.bay	27	27	29	28	28	27.90	0.88
	CV2_9.bay	29	29	25	25	29	27.30	2.0
	CV2_10.bay	28	27	24	24	24	25,40	1.84

Tabla B. 2 Detalle de los resultados obtenidos según ejecución para CV (2).

Caso	Instancia	E6	E7	E8	E9	E10	Prom.	D.E
CV1	CV1_1.bay	12	13	13	14	13	13.20	1.10
	CV1_2.bay	8	8	8	8	8	8.0	0.0
	CV1_3.bay	13	14	13	13	13	13.30	0.48
	CV1_4.bay	13	13	12	11	11	11.80	0.92
	CV1_5.bay	15	13	14	15	15	14.0	0.94
	CV1_6.bay	9	8	8	8	8	8.20	0.42
	CV1_7.bay	8	8	8	9	8	8.10	0.32
	CV1_8.bay	11	11	12	12	11	11.40	0.51
	CV1_9.bay	9	9	10	10	9	9.50	0.53
	CV1_10.bay	11	8	10	10	10	9.70	1.25
CV2	CV2_1.bay	26	26	25	25	24	24.80	0.92
	CV2_2.bay	25	24	24	22	23	23,60	1.07
	CV2_3.bay	20	18	18	20	19	19.20	0.79
	CV2_4.bay	24	25	24	24	28	25.50	1.78
	CV2_5.bay	29	32	29	31	31	30.40	1.26
	CV2_6.bay	25	25	27	25	25	25.50	1.08
	CV2_7.bay	32	32	33	33	33	31.90	0.88

Tabla B. 3 Detalle de los resultados obtenidos según ejecución para CV (3).

Caso	Instancia	E6	E7	E8	E9	E10	Prom.	D.E
CV2	CV2_8.bay	29	28	27	27	29	27.90	0.88
	CV2_9.bay	29	25	28	29	25	27.30	2.0
	CV2_10.bay	28	27	24	24	24	25,40	1.84

Las tablas B.4 a B.6 muestran los resultados de las 10 ejecuciones realizadas con los 2 faltantes casos de prueba propuestos en CV. La primera tabla detalla las primeras 5 ejecuciones, mientras que las últimas dos detallan las faltantes.

Tabla B. 4 Detalle de los resultados obtenidos según ejecución para CV (4).

Caso	Instancia	E1	E2	E3	E4	E5	Prom.	D.E
CV3	CV3_1.bay	33	35	35	35	37	35.20	1.48
	CV3_2.bay	38	38	40	42	40	40,9	1.91
	CV3_3.bay	38	38	39	40	40	38.40	0.97
	CV3_4.bay	44	38	42	38	38	41.30	2.91
	CV3_5.bay	40	37	37	38	38	38.20	1.13
	CV3_6.bay	43	44	40	42	43	43.0	1.3
	CV3_7.bay	48	45	44	45	42	45.0	1.82
	CV3_8.bay	40	38	38	43	38	39.20	2.10
	CV3_9.bay	40	38	35	37	37	37.0	1.41
	CV3_10.bay	42	38	37	40	40	11,4	1.62
CV4	CV4_1.bay	35	37	37	34	34	35.60	2.30
	CV4_2.bay	55	54	54	52	55	54.70	1.49
	CV4_3.bay	65	66	62	66	65	64.60	1.43
	CV4_4.bay	68	69	68	67	69	68.30	0.67
	CV4_5.bay	67	70	68	68	68	68.30	0.95
	CV4_6.bay	68	70	70	73	68	70.30	2.06
	CV4_7.bay	67	64	64	66	66	65.80	1.32
	CV4_8.bay	68	65	65	64	65	65.20	1.14
	CV4_9.bay	68	67	66	67	68	67.10	0.88
	CV4_10.bay	65	63	64	64	64	63.50	0.97

Tabla B. 5 Detalle de los resultados obtenidos según ejecución para CV (5).

Caso	Instancia	E6	E7	E8	E9	E10	Prom.	D.E
CV3	CV3_1.bay	33	37	35	37	35	35.20	1.48
	CV3_2.bay	43	40	38	42	38	40,9	1.91
	CV3_3.bay	38	38	37	38	38	38.40	0.97
	CV3_4.bay	44	44	43	38	44	41.30	2.91
	CV3_5.bay	38	37	40	39	38	38.20	1.13
	CV3_6.bay	43	43	43	44	45	43.0	1.3
	CV3_7.bay	45	45	48	44	44	45.0	1.82
	CV3_8.bay	43	38	38	38	38	39.20	2.10

Tabla B. 6 Detalle de los resultados obtenidos según ejecución para CV (6).

Caso	Instancia	E6	E7	E8	E9	E10	Prom.	D.E
CV3	CV3_9.bay	37	37	35	37	37	37.0	1.41
	CV3_10.bay	37	38	40	40	40	39.20	1.62
	CV4_1.bay	34	37	37	37	30	35.60	2.30
	CV4_2.bay	57	54	57	54	55	54.70	1.49
	CV4_3.bay	65	62	65	65	65	64.60	1.43
CV4	CV4_4.bay	68	68	69	69	68	68.30	0.67
	CV4_5.bay	68	68	70	68	68	68.30	0.95
	CV4_6.bay	73	70	70	68	73	70.30	2.06
	CV4_7.bay	64	66	67	67	67	65.80	1.32
	CV4_8.bay	64	65	65	66	65	65.20	1.14
	CV4_9.bay	66	66	68	67	68	67.10	0.88
	CV4_10.bay	62	63	64	64	62	63.50	0.97

A continuación, las tablas B.7 a B.9 muestran los resultados de las 10 ejecuciones realizadas con los 3 primeros casos de prueba propuestos en LC. La primera tabla detalla las primeras 5 ejecuciones, mientras que las dos siguientes detallan los datos faltantes.

Tabla B. 7 Detalle de los resultados obtenidos según ejecución para LC (1).

Caso	Instancia	E1	E2	E3	E4	E5	Prom.	D.E
LC1	lc1.bay	19	21	20	19	19	20.0	1.05
LC2a	lc2a_1.bay	33	27	27	32	32	30.10	2.69
	lc2a_2.bay	25	27	28	27	27	27.10	0.88
	lc2a_3.bay	33	33	32	32	32	32.0	1.49
	lc2a_4.bay	30	29	28	30	29	28.9	0.88
	lc2a_5.bay	28	32	30	32	32	30.0	2.05
	lc2a_6.bay	33	30	30	33	33	31.80	1.55
	lc2a_7.bay	31	31	33	31	31	31.30	0.95
	lc2a_8.bay	28	30	31	34	34	31.80	2.10
	lc2a_9.bay	32	31	32	32	32	31.50	0.53
	lc2a_10.bay	34	32	32	34	32	32.60	1.35
LC2b	lc2b_1.bay	47	47	48	47	48	47.40	0.52
	lc2b_2.bay	49	51	51	48	48	49.50	1.35
	lc2b_3.bay	52	49	52	52	50	50.90	1.44
	lc2b_4.bay	53	51	51	51	54	52.80	1.32
	lc2b_5.bay	47	50	51	48	48	48.80	1.55
	lc2b_6.bay	51	49	49	53	53	51.40	2.12
	lc2b_7.bay	47	47	48	47	51	48.80	2.10
	lc2b_8.bay	52	55	51	52	52	52.70	1.34
	lc2b_9.bay	49	49	51	51	48	50.20	1.32
	lc2b_10.bay	48	46	48	47	48	47.50	0.71

Tabla B. 8 Detalle de los resultados obtenidos según ejecución para LC (2).

Caso	Instancia	E6	E7	E8	E9	E10	Prom.	D.E
LC1	lc1.bay	22	20	21	19	20	20.0	1.05
LC2a	lc2a_1.bay	27	27	32	32	32	30.10	2.69

Tabla B. 9 Detalle de los resultados obtenidos según ejecución para LC (3).

Caso	Instancia	E6	E7	E8	E9	E10	Prom.	D.E
LC2a	lc2a_2.bay	27	28	28	27	27	27.10	0.88
	lc2a_3.bay	32	33	28	32	33	32.0	1.49
	lc2a_4.bay	29	30	28	28	28	28.9	0.88
	lc2a_5.bay	28	31	27	28	32	30.0	2.05
	lc2a_6.bay	33	33	33	30	30	31.80	1.55
	lc2a_7.bay	30	31	33	31	31	31.30	0.95
	lc2a_8.bay	34	31	31	34	31	31.80	2.10
	lc2a_9.bay	31	31	32	31	31	31.50	0.53
	lc2a_10.bay	32	34	34	30	32	32.60	1.35
	LC2b	lc2b_1.bay	48	47	47	47	48	47.40
lc2b_2.bay		49	48	51	51	49	49.50	1.35
lc2b_3.bay		52	49	49	52	52	50.90	1.44
lc2b_4.bay		53	53	54	54	54	52.80	1.32
lc2b_5.bay		47	50	48	51	48	48.80	1.55
lc2b_6.bay		48	51	53	53	54	51.40	2.12
lc2b_7.bay		48	51	48	53	48	48.80	2.10
lc2b_8.bay		53	55	52	52	53	52.70	1.34
lc2b_9.bay		52	51	49	51	51	50.20	1.32
lc2b_10.bay		48	47	47	48	48	47.50	0.71

Ahora, las tablas B.10 a B.12 muestran los resultados de las 10 ejecuciones realizadas con los últimos 2 casos de prueba propuestos en LC. La primeras dos tablas detallan las primeras 5 ejecuciones, mientras que la última detalla las faltantes.

Tabla B. 10 Detalle de los resultados obtenidos según ejecución para LC (4).

Caso	Instancia	E1	E2	E3	E4	E5	Prom.	D.E
LC3a	lc3a_1.bay	45	48	48	48	45	46.80	1.62
	lc3a_2.bay	47	48	47	53	47	49.10	2.77
	lc3a_3.bay	48	47	47	48	52	49.0	2.11
	lc3a_4.bay	46	47	49	51	47	48.10	1.73
	lc3a_5.bay	47	53	53	49	47	49.70	2.41
	lc3a_6.bay	51	53	47	53	48	50.0	2.49
	lc3a_7.bay	53	53	46	51	49	50.30	2.31
	lc3a_8.bay	54	48	48	48	49	49.30	1.95
	lc3a_9.bay	53	51	51	49	51	50.90	1.79
	lc3a_10.bay	48	54	53	54	54	52.70	1.95

Tabla B. 11 Detalle de los resultados obtenidos según ejecución para LC (5).

Caso	Instancia	E1	E2	E3	E4	E6	Prom.	D.E
LC3b	lc3b_1.bay	68	67	71	72	67	68.70	1.89
	lc3b_2.bay	73	74	68	69	72	71.40	2.27
	lc3b_3.bay	69	75	73	73	71	72.20	2.0
	lc3b_4.bay	68	70	71	68	72	69.80	1.67
	lc3b_5.bay	74	73	72	69	70	71.50	1.96
	lc3b_6.bay	76	73	71	75	73	73.30	1.63
	lc3b_7.bay	73	71	72	68	72	71.20	1.93
	lc3b_8.bay	74	74	75	75	74	74.80	0.79
	lc3b_9.bay	75	76	73	74	74	75.30	1.65
	lc3b_10.bay	77	75	77	76	75	76.0	0.81

Tabla B. 12 Detalle de los resultados obtenidos según ejecución para LC (6).

Caso	Instancia	E6	E7	E8	E9	E10	Prom.	D.E
LC3a	lc3a_1.bay	45	47	48	49	45	46.80	1.62
	lc3a_2.bay	49	53	47	47	53	49.10	2.77
	lc3a_3.bay	52	52	48	48	48	49.0	2.11
	lc3a_4.bay	48	47	51	48	47	48.10	1.73
	lc3a_5.bay	48	53	49	49	49	49.70	2.41
	lc3a_6.bay	47	53	49	48	51	50.0	2.49
	lc3a_7.bay	49	49	51	53	49	50.30	2.31
	lc3a_8.bay	48	48	49	50	51	49.30	1.95
	lc3a_9.bay	53	48	51	53	49	50.90	1.79
	lc3a_10.bay	53	54	52	54	51	52.70	1.95
LC3b	lc3b_1.bay	67	71	68	68	68	68.70	1.89
	lc3b_2.bay	68	72	74	72	72	71.40	2.27
	lc3b_3.bay	72	74	69	73	73	72.20	2.0
	lc3b_4.bay	68	72	70	71	68	69.80	1.67
	lc3b_5.bay	73	74	69	70	71	71.50	1.96
	lc3b_6.bay	73	75	73	73	71	73.30	1.63
	lc3b_7.bay	74	68	71	71	72	71.20	1.93
	lc3b_8.bay	76	76	75	74	75	74.80	0.79
	lc3b_9.bay	76	78	74	75	78	75.30	1.65
	lc3b_10.bay	75	77	76	76	76	76.0	0.81

## C. Evidencia

A continuación las figuras C.1 y C.2 ilustran el resultado obtenido para CV1\_2.bay.

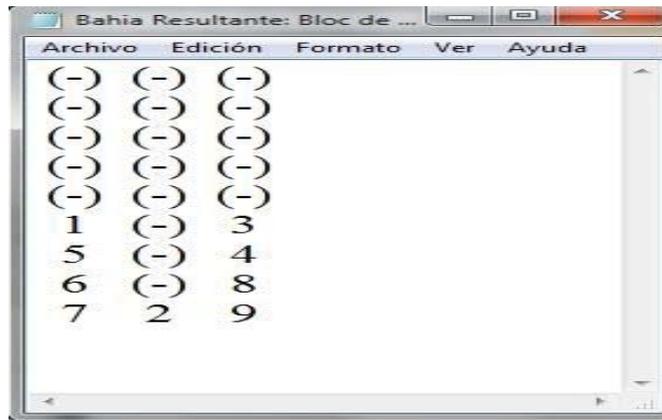


Figura C. 1 Ejemplo de bahía resultante para CV1\_2.bay.

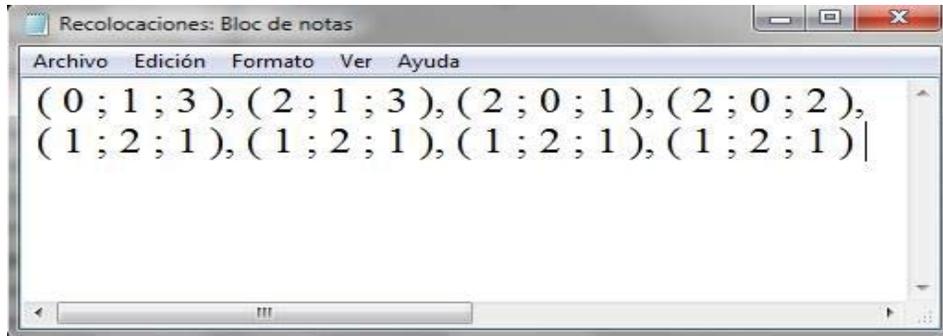


Figura C. 2 Ejemplo de recolocaciones resultantes para ordenar CV1\_2.bay.