



CONSTANCIA DE VALIDACIÓN Y CONFIDENCIALIDAD DE MONOGRAFÍA A REPOSITORIO ACADÉMICO

1.- IDENTIFICACIÓN DEL TRABAJO ACADÉMICO

Tipo de monografía (marcar una opción): ☒ Memoria o trabajo de título ☐ Tesis de Postgrado

Título del trabajo: OPTIMIZACIÓN DE MOVIMIENTOS Y ROTACIÓN DE CONTENEDORES VACÍOS INTEGRADO A LA APLICACIÓN DINODOCK

Nombre del candidato(a): Joaquin Aguilar Olivares

Carrera / Grado: ingeniería en informática

Campus: Viña del mar Departamento: Electrotecnia e Informática

2.- VALIDACIÓN DEL PROFESOR GUÍA/DIRECTOR DE TESIS

Yo, Gabriel Jara Bulnes, en mi calidad de profesor(a) guía/director(a) del trabajo académico mencionado anteriormente **DEJO CONSTANCIA** que:

- He revisado esta versión del documento y corresponde a la versión final aprobada del trabajo.
- El trabajo cumple con los requisitos académicos y de formato establecidos por la institución.

3.- EVALUACIÓN DE CONFIDENCIALIDAD POR PROPIEDAD INDUSTRIAL (marcar una opción)

☒ El trabajo **NO contiene** información que amerite confidencialidad y puede ser publicado de inmediato en repositorio con acceso abierto.

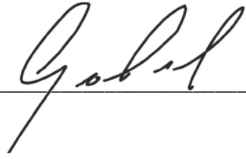
☐ El trabajo **CONTIENE** información con potenciales implicancias de propiedad industrial o intelectual y requiere un periodo de confidencialidad (**embargo**) por (marcar una opción):

☐ 6 meses ☐ 12 meses ☐ 2 años ☐ 3 años ☐ 5 años ☐ 10 años

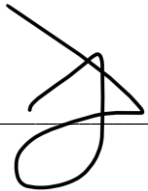
Fundamentación de la necesidad de confidencialidad (obligatorio si se solicita embargo):

4.- FIRMAS

Profesor(a) guía o director(a) de memoria o tesis:

Fecha: 05-11-2025 Firma: 

Estudiante o Candidato(a):

Fecha: 05-11-2025 Firma: 

Este formulario debe ser insertado como página 2 de la memoria o tesis, completado y firmado por estudiante y profesor(a) antes de la entrega en portal PRISMA de Biblioteca USM.



UNIVERSIDAD TÉCNICA
FEDERICO SANTA MARÍA

DEPARTAMENTO DE ELECTROTECNIA E INFORMÁTICA
INGENIERÍA EN INFORMÁTICA

OPTIMIZACIÓN DE MOVIMIENTOS Y ROTACIÓN DE CONTENEDORES VACÍOS INTEGRADO A LA APLICACIÓN DINODOCK

Joaquín Aguilar Olivares

Joaquin.aguilero@usm.cl

Gabriel Jara
Profesor Guía

Resumen: Las empresas dedicadas a la logística de contenedores vacíos deben cumplir con cuotas de rotación y objetivos ecológicos, establecidos tanto por sus clientes como por sus propias políticas internas. El equilibrio entre estos dos objetivos, que en muchos casos pueden ser contrapuestos, representa un desafío considerable. Para abordar esta problemática se implementó un sistema de optimización diseñado para mitigar dicha tensión. Este sistema, denominado DinoDock, utiliza funciones de optimización para asignar posiciones a los contenedores durante su ingreso al patio y para seleccionar cuáles serán entregados en el proceso de salida. Con el fin de evaluar su desempeño, este modelo de optimización se compara frente a algoritmos comúnmente utilizados en este tipo de escenarios.

Palabras clave: Optimización, Contenedores, Rotación

1 INTRODUCCIÓN

Las empresas de logística de contenedores almacenan estos hasta que son necesarios, asegurándose de que estén disponibles rápidamente cuando se requieran. Para ello, los concentran en grandes patios donde reposan hasta ser solicitados, siendo este último paso donde surge un problema: los contenedores no siempre salen en un orden adecuado, lo que deja algunos sin uso durante largos periodos. Esto, debido a la necesidad de movimiento de los contenedores, genera gastos adicionales, ya que es necesario realizar revisiones repetidas, lo que constituye una especie de "multa", además de requerir constante coordinación con los transportistas para agilizar el proceso. Aunque existen técnicas que buscan evitar demoras excesivas y problemas de falta de rotación, han demostrado ser insuficientes o presentar inconvenientes en su aplicación práctica, lo que conlleva costos tanto monetarios como de reputación para las empresas.

Otro problema creciente en este sector es el impacto ambiental, especialmente en lo relacionado con la contaminación generada por las grúas al no existir métodos óptimos para almacenar o extraer contenedores, generando movimientos innecesarios, los cuales al significar un mayor consumo de combustible se traduce en más emisiones de CO₂. Esta preocupación es recurrente, dado que muchas empresas de logística y navieras cuentan con políticas estrictas sobre emisiones de carbono. El

incumplimiento de estas metas ambientales puede ocasionar pérdidas tanto en la imagen de marca como en las relaciones comerciales. Por ello, encontrar soluciones eficientes para el almacenamiento y la extracción de contenedores vacíos resulta crucial para evitar diversas pérdidas.

En la actualidad, para mitigar las problemáticas operativas en las terminales logísticas, se recurre a la implementación de diversas estrategias. Entre ellas destacan el uso de la técnica de cross docking [1], que establece una zona de espera para agilizar el proceso de entregas; la zonificación, que consiste en dividir el espacio en áreas asignadas a cada tipo de contenedor para una mejor organización y distribución [2], y la aplicación de métodos de gestión de inventario como FIFO (First In, First Out) o LIFO (Last In, First Out) [2], seleccionados en función del contexto operativo. Asimismo, se emplean plataformas de planificación que permiten organizar los pedidos de manera más eficiente mediante sistemas de encargo que permiten seleccionar previamente los contenedores que se despacharán y permiten agendar horas al retiro. Además, se utiliza el seguimiento de los contenedores para apoyar la toma de decisiones, con el objetivo de optimizar los flujos de trabajo y reducir los tiempos de espera.

Si bien estas técnicas aportan beneficios significativos, su aplicación aislada no siempre resulta suficiente. Por ejemplo, la zonificación facilita la localización de contenedores y contribuye a un mejor aprovechamiento del espacio, pero no resuelve por sí sola los desafíos relacionados con la rotación adecuada de los mismos. Del mismo modo, los métodos de gestión de inventario dependen fuertemente del contexto operativo, el cual puede variar constantemente y estar sujeto a la interpretación o experiencia del operario en terreno. En cuanto a las plataformas de planificación de pedidos, estas suelen depender de terceros, lo que introduce un factor de incertidumbre si dichos actores no cumplen con los tiempos o procesos establecidos.

En este contexto, se propone el desarrollo de DinoDock, una aplicación orientada a la optimización de los procesos de entrada y salida de contenedores, con un enfoque particular en la gestión de contenedores vacíos. DinoDock tiene como objetivo principal mejorar la eficiencia en la distribución y selección de contenedores dentro del patio, minimizando movimientos innecesarios, bloqueos futuros y tiempos de espera. Para lograrlo, la aplicación se basa en la implementación de algoritmos de optimización que permiten tomar decisiones inteligentes en dos momentos críticos del proceso: al momento de asignar una posición a un contenedor entrante y al decidir qué contenedor debe ser retirado del patio.

2 OBJETIVOS

Se presenta a continuación los objetivos que persigue el presente trabajo.

2.1 OBJETIVO GENERAL

Desarrollar un modelo de optimización para la rotación y movimientos de contenedores vacíos, integrado a la aplicación DinoDock, con potencial de ayudar a mejorar la eficiencia operativa y reducir costos tanto logísticos como ambientales.

2.2 OBJETIVOS ESPECÍFICOS

- Identificar las principales variables y restricciones que influyen en la rotación y movimientos de contenedores vacíos dentro de las operaciones logísticas.
- Desarrollar un algoritmo de optimización que minimice los costos de movimiento de grúa y la falta de rotación de contenedores.
- Integrar el modelo de optimización con la plataforma DinoDock para mejorar la gestión de flotas y rotación de contenedores vacíos.

- Evaluar la eficiencia del algoritmo propuesto mediante simulaciones de un patio de contenedores con movimientos instantáneos.

3 METODOLOGÍA

El desarrollo del modelo comprende varias etapas. En primer lugar, se realiza una revisión bibliográfica, que consiste en la recopilación y análisis de estudios previos relacionados con la optimización de contenedores vacíos, con el objetivo de comprender las metodologías existentes e identificar soluciones aplicadas en la logística.

Posteriormente, se lleva a cabo la identificación de variables y restricciones, definiendo los elementos clave que intervienen en la optimización, tales como variables (Ej.: información referente al contenedor) y restricciones (Ej.: limitaciones físicas), con el fin de estructurar adecuadamente el modelo de optimización.

A continuación, se procede al desarrollo del modelo de optimización, mediante la formulación matemática del problema de reubicación de contenedores vacíos, integrando las variables y restricciones previamente identificadas.

Una vez formulado, se realiza la integración con DinoDock, preparando las funciones necesarias para que el modelo pueda interactuar con la aplicación.

Finalmente, se ejecuta la validación y simulación del modelo, comprobando que representa adecuadamente la realidad del sistema logístico de contenedores vacíos. Mediante simulaciones con datos reales o escenarios hipotéticos, se evalúa el comportamiento del modelo, se ajustan los parámetros y se verifica que las soluciones propuestas sean viables y eficaces en la práctica.

4 MARCO TEÓRICO Y ESTADO DEL ARTE

El presente apartado tiene como finalidad establecer los fundamentos conceptuales y técnicos que sustentan esta investigación, así como ofrecer una visión general del conocimiento existente en torno al objeto de estudio. Para ello, se abordan las principales metodologías y enfoques relacionados con la gestión logística en patios de contenedores.

4.1 CONTEXTO GENERAL DEL PROCESO LOGÍSTICO

Cuando un contenedor llega a las instalaciones, inicia un proceso compuesto por varias etapas fundamentales para asegurar su correcto manejo y conservación. En primer lugar, se somete a una inspección técnica destinada a evaluar su estado general y detectar posibles daños o deterioros. Si durante esta revisión se identifican fallas o imperfecciones, el contenedor pasa a una fase de reparación, donde se realizan las intervenciones necesarias para dejarlo en condiciones óptimas de uso [3].

Finalizadas estas etapas, el contenedor es trasladado a una zona de almacenamiento, donde permanece a la espera de ser nuevamente requerido. La solicitud de contenedores por parte de los clientes se realiza según criterios específicos, principalmente por marca y tipo, ya que cada unidad está asociada a un propietario determinado. Este procedimiento permite mantener un control riguroso sobre la gestión y trazabilidad de los equipos, asegurando que cada contenedor sea entregado conforme a los requerimientos de su respectivo dueño.

Es importante destacar que el tratamiento de los contenedores varía según su tipo. Por ejemplo, los contenedores freeze requieren un mantenimiento especializado con estructuras particulares, además de

presentar necesidades de rotación distintas a las de los contenedores dry. Por esta razón, ambos tipos son gestionados de manera diferenciada dentro del sistema.

4.2 SOLUCIONES ACTUALES

Para considerar un sistema como una solución viable, se evaluó su capacidad para abordar problemáticas que inciden negativamente en la eficiencia operativa, tales como la baja rotación de contenedores y el exceso de movimientos innecesarios. Por ejemplo, las funcionalidades orientadas a mantener el orden del patio, ya sea mediante el seguimiento en tiempo real de los contenedores o a través de la asignación eficiente de posiciones dentro del área de almacenamiento.

SYNCROTESS: Es un software de optimización que utiliza IA para la toma de decisiones en distintos procesos, como la elección de donde guardar contenedores, como el retiro de estos. Es una solución integral en la toma de decisiones en el mundo de la logística.

Dynamic Yard: Es una herramienta de gestión de patios diseñada para facilitar las operaciones diarias en instalaciones medianas. Mejora la visibilidad del patio y optimiza funciones clave como el control en puertas, supervisión de muelles y despacho, además de permitir la generación de informes de rendimiento, todo con el objetivo de agilizar las operaciones.

SP depot: Es un sistema que ayuda en la gestión operativa, administrativa y en los procesos comerciales de un patio de contenedores para gestionar integralmente los procesos comerciales, administrativos y operativos de patios de contenedores. Cubre el ciclo completo de operaciones, desde el ingreso y salida hasta inspecciones, lavados, reparaciones y facturación.

En general, las herramientas actuales están diseñadas para apoyar la toma de decisiones y la gestión general del patio. Estas facilitan considerablemente el trabajo del encargado, ya que le permiten actuar de manera informada; sin embargo, aún persiste el desafío de encontrar soluciones óptimas para la ubicación y movimiento de cada contenedor.

4.3 TÉCNICAS DE OPTIMIZACION APLICADAS EN LA GESTIÓN DE CONTENEDORES

Se emplean diversas técnicas para optimizar las operaciones de entrada y salida de contenedores. Entre las más sencillas se encuentran los protocolos de decisión que únicamente consideran el momento de entrada del contenedor.

- **FIFO (First In, First Out):** Algoritmo de gestión utilizado para asegurar que los primeros contenedores en entrar sean también los primeros en salir. Este enfoque tiene como objetivo optimizar la rotación de los contenedores dentro del patio, garantizando que los contenedores que han permanecido más tiempo en el lugar sean los primeros en ser extraídos. Al seguir esta estrategia, se busca evitar la acumulación de contenedores que permanecen de manera prolongada, lo que podría afectar tanto la eficiencia operativa como los costos asociados al almacenamiento de contenedores sin uso.
- **LIFO (Last In, First Out):** Algoritmo de gestión en el que los últimos contenedores en entrar son los primeros en salir. Este enfoque se utiliza generalmente cuando hay una alta demanda de espacio o cuando se necesita priorizar la salida rápida de ciertos contenedores, en lugar de centrarse en la rotación de los mismos. LIFO es útil en situaciones donde los contenedores más recientes son los que requieren ser extraídos con mayor urgencia, como cuando se espera un envío rápido o una necesidad inmediata de espacio para nuevos contenedores. A diferencia del método FIFO, LIFO puede generar una menor rotación en el patio, ya que los contenedores más antiguos pueden permanecer durante más tiempo. Sin embargo, su ventaja radica en su capacidad para

gestionar demandas urgentes y acelerar la salida de contenedores recientes, lo que lo convierte en una opción efectiva en escenarios de alta presión.

- Aleatorio: Este método se utiliza en diversas áreas para organizar objetos que comparten alguna relación. Un ejemplo de ello es la memoria RAM, que asigna ubicaciones a los datos de manera aparentemente aleatoria, pero manteniendo una organización lógica que permite acceder a la información de forma eficiente [4]. También es utilizado en otros contextos de logística para mejorar el aprovechamiento del espacio. [5]

También existen modelos matemáticos de optimización, los cuales se fundamentan en la formulación de funciones de costo y restricciones con el objetivo de encontrar una solución óptima o lo más cercana posible a esta.

- Programación lineal: Es una técnica de optimización matemática que tiene como objetivo maximizar o minimizar una función lineal, sujeta a un conjunto de restricciones también lineales [6]. Resulta especialmente útil en problemas bien definidos y acotados; sin embargo, a medida que aumenta la complejidad del problema (ya sea por el número de variables o restricciones), los tiempos de cómputo pueden incrementarse considerablemente, afectando la eficiencia del proceso de resolución.
- Basado en prioridad: Estos algoritmos son utilizados dando un “peso” a cada variable para así priorizar o dar más “peso” a las más importantes.
- Heurística: Estos enfoques proporcionan soluciones aproximadas de alta calidad para problemas complejos, reduciendo considerablemente el tiempo de cómputo en comparación con los métodos de optimización exactos [7].
- Algoritmo genético: Este algoritmo está basado en los procesos biológicos de evolución. Para encontrar la solución a un problema, este algoritmo va seleccionando y cruzando soluciones. Ha sido ocupado en estudios con el problema de la logística de contenedores [8].
- Búsqueda Tabú: Este algoritmo busca soluciones y las agrega a una lista tabú, para así no pasar más por estas e ir encontrando mejores. Ha sido ocupado en estudios con el problema de la logística de contenedores [9].
- búsqueda exhaustiva: Técnica de optimización que implica explorar todas las posibles soluciones a un problema para encontrar la mejor opción. Este enfoque es particularmente útil en problemas acotados, donde el número de variables y soluciones posibles es limitado, lo que permite que el proceso de búsqueda sea manejable en términos de tiempo de cómputo. La búsqueda exhaustiva garantiza que se encuentre la solución óptima, ya que no se omite ninguna opción en el proceso. Sin embargo, cuando el problema es de mayor magnitud o tiene un gran número de variables, los tiempos de cómputo pueden volverse excesivos, lo que hace que este método no sea práctico. En tales casos, se prefieren métodos inexactos o heurísticos, que, aunque no garantizan la solución óptima, pueden proporcionar soluciones aproximadas de manera mucho más eficiente, incluso en problemas de gran escala.
- Simulaciones: Estas, son muy utilizadas al momento de optimizar algo, ya que permite analizar el comportamiento del sistema sin intervenir en el entorno real, además, mientras mayor es la complejidad del objeto a optimizar, mayor es la necesidad de una simulación para poder replicar las distintas variables del entorno [10], [11],

Se observa que existe una amplia variedad de métodos disponibles, y la elección del más adecuado depende tanto de la complejidad operativa del patio como de las capacidades computacionales con las que se cuenta.

4.4 VARIABLES

Las variables presentadas a continuación fueron seleccionadas a partir del análisis de la literatura y adaptadas al problema identificado, con el fin de representar los factores que influyen en el movimiento de contenedores dentro del patio.

- Días en patio: Correspondiendo al tiempo que ha estado un contenedor en el patio.
- Tipo contenedor: Correspondiendo a si el contenedor es Dry o Freeze [12]
- Tamaño contenedor: Correspondiendo a si el contenedor es de 20 o 40 pies. [12]
- Movimientos de grúa: Correspondiendo a la cantidad de movimientos totales de la grúa para poder acceder al contenedor que se requiere. [12]
- Marca del contenedor: Correspondiendo a la marca del contenedor.

4.5 RESTRICCIONES

A continuación, se describen las restricciones incorporadas en el modelo de optimización. Dichas restricciones permiten acotar las posiciones posibles de los contenedores, asegurando el cumplimiento de las condiciones físicas del sistema y de las políticas de gestión del negocio.

4.5.1 RESTRICCIONES FÍSICAS

No se permite ubicar contenedores en posiciones que no cuenten con un soporte inferior. En otras palabras, un contenedor solo puede colocarse sobre otro contenedor o directamente sobre el suelo del patio, para de esta forma respetar la gravedad.

No se pueden asignar varios contenedores a una misma posición dentro del patio. Cada ubicación solo puede estar ocupada por un contenedor a la vez.

Cada contenedor debe ser asignado a una única posición dentro del patio. No se permite que un mismo contenedor ocupe más de un lugar.

4.5.2 RESTRICCIONES DEL NEGOCIO

Se está tomando un enfoque de almacenamiento en bloques, por lo tanto, no se mezclarán tipos y tamaños de contenedores, ya que los contenedores tienen distintos tipos de tiempos de mantenimiento y rotación.

5 DESARROLLO

En este apartado se desarrolla el núcleo del trabajo, estructurado en tres secciones principales. En primer lugar, se aborda el desarrollo de la simulación, donde se describen los procesos y herramientas utilizados para modelar el sistema objeto de estudio. En la segunda sección, se presenta el desarrollo de la función de optimización, detallando su formulación matemática y los enfoques empleados para resolverla. Finalmente, la tercera sección se centra en la integración de la función a DinoDock.

5.1 SIMULACIÓN

La simulación tiene como objetivo servir como un entorno experimental para probar y evaluar el desempeño de la función de optimización, lo que la convierte en un componente fundamental de este trabajo. Al emular el funcionamiento básico de un patio de contenedores, permite recrear las condiciones y dinámicas típicas del sistema, proporcionando un espacio controlado donde se pueden identificar y resolver problemas antes de la implementación real.

Por esta razón, la simulación es la primera parte a desarrollar, ya que su correcta construcción es crucial para validar las siguientes etapas. La estructura del software diseñado para la simulación sigue el esquema presentado en la figura 1, el cual organiza las principales funcionalidades del modelo, asegurando una representación fiel y eficiente del sistema a analizar.

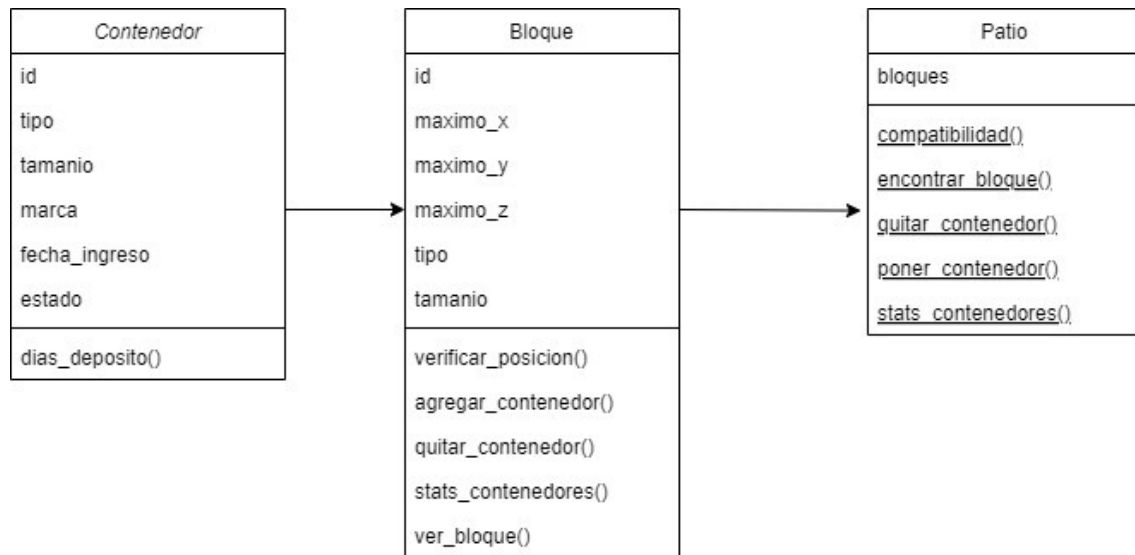


Figura 1. Esquema general de la simulación

Para el desarrollo de la simulación, se utilizó el lenguaje de programación Python, acompañado de las librerías numpy, random y datetime. Estas herramientas se seleccionaron por su versatilidad y eficiencia en el manejo de cálculos numéricos, generación de datos aleatorios y manipulación de fechas, respectivamente, lo que permitió construir un modelo que emula las dinámicas de un patio de contenedores de manera funcional.

Para validar el funcionamiento de la simulación, se realizaron pruebas que consistieron en efectuar movimientos específicos de contenedores dentro del modelo. Durante estas pruebas, se observaron los cambios generados en las posiciones, asegurándose de que el comportamiento del sistema replicara de manera adecuada las dinámicas reales de un patio de contenedores.

Entre las limitaciones de la simulación, no se considera la presencia de grúas ni los tiempos asociados a su operación, asumiendo que los movimientos se generan de manera instantánea durante el día, pero, bajo ciertas restricciones físicas, como el no poder mover contenedores debajo de otros sin antes mover el de arriba. Estas simplificaciones se han implementado para enfocar el modelo en los aspectos más relevantes para el proyecto: los movimientos necesarios para manejar los contenedores y los días de permanencia en el patio. Ambos factores son fundamentales para desarrollar y validar la función de optimización, ya que determinan tanto la eficiencia operativa como la capacidad de minimizar costos y bloqueos en el sistema.

Para simular la entrada de contenedores al patio, se generaron datos aleatorios que representaran tanto el ingreso de nuevos contenedores como su demanda. Este enfoque permite recrear de manera dinámica las variaciones que podrían ocurrir en un entorno real, como fluctuaciones en el número de contenedores que llegan y las solicitudes de extracción. Los datos generados incluyen características clave de los contenedores, como tipo, tamaño y marca.

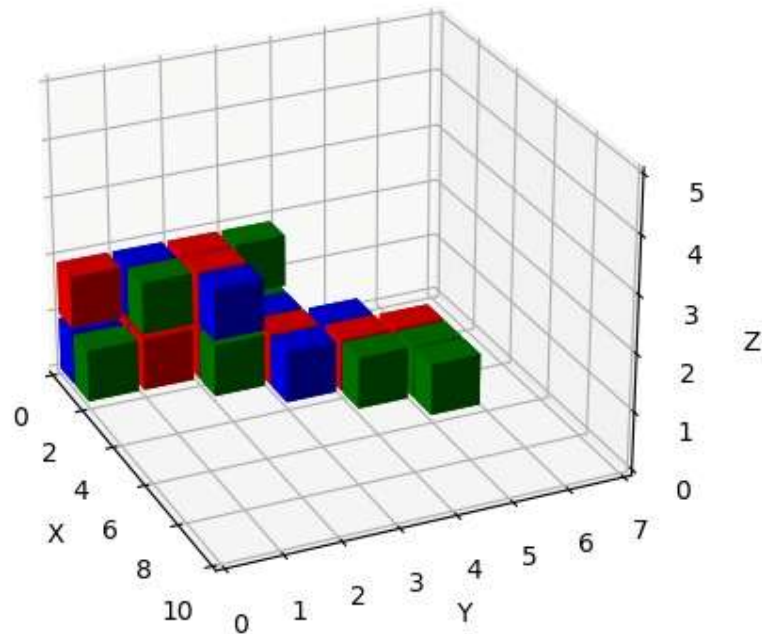


Figura 2. Vista de un bloque de contenedores

5.2 FUNCIÓN DE OPTIMIZACIÓN

La función de optimización constituye el núcleo de este trabajo, siendo el elemento más relevante dentro del desarrollo del proyecto. Su propósito es modelar matemáticamente el problema de gestión de contenedores para encontrar soluciones que minimicen los movimientos, a la vez que se aumenta la rotación.

Se desarrollaron dos funciones de optimización, cada una con un enfoque distinto pero complementario. La función de entrada se centra en garantizar la accesibilidad de los contenedores, organizándolos de manera que se minimicen los bloqueos futuros. Por otro lado, la función de salida prioriza la extracción del contenedor que requiera el menor número de movimientos o aquel que haya permanecido más tiempo en el patio.

Esta separación responde a la complejidad que representa abordar simultáneamente ambos problemas. No obstante, aunque las funciones estén diferenciadas, no operan de manera completamente independiente: existen mecanismos que permiten la interacción entre ellas. Ambas funciones comparten un objetivo central, que es optimizar la rotación del patio. La función de salida introduce un componente adicional, al enfocarse en minimizar la cantidad de movimientos necesarios para extraer un contenedor. Para compensar esto, la función de entrada se orienta a reducir los bloqueos, priorizando la accesibilidad de los contenedores con mayor antigüedad.

Además, al mantener las funciones separadas, es posible utilizarlas de forma independiente, lo cual reduce el tiempo de cómputo durante los procesos de entrada o salida. Esto resulta especialmente útil al emplear heurísticas como los algoritmos genéticos, donde las soluciones generadas para los cruces tienden a aumentar de tamaño y complejidad. Incluso si una parte de la función no se utiliza directamente, seguiría existiendo en memoria y participando en los cruces, afectando el rendimiento general del algoritmo.

5.2.1 FUNCIÓN DE ENTRADA

Este problema se aborda como un modelo binario, ya que las decisiones clave consisten en determinar si un contenedor debe ocupar una posición específica en el patio o no. Además de estas decisiones, es fundamental considerar ciertos atributos de los contenedores que influyen en el comportamiento del sistema. Entre ellos, destacan los días que el contenedor ha permanecido en el patio (siendo 0 para contenedores recién ingresados), el tipo y tamaño del contenedor, y la media de días que contenedores del mismo tipo suelen permanecer en el patio.

La función de optimización tiene como objetivo principal determinar la disposición de los contenedores en el patio de manera que se maximice su accesibilidad. Esto garantiza un acceso eficiente a los contenedores y optimiza la extracción de aquellos que llevan más tiempo en el patio.

En cuanto a las restricciones, se asegura el respeto por la gravedad, es decir, no se permite colocar un contenedor sobre un espacio vacío. Además, se establece que no puede haber más de un contenedor en una misma posición, y un contenedor no puede ser asignado a más de una posición. También se incluye una restricción adicional que considera las posiciones previamente ocupadas, asegurando que se tenga en cuenta el estado actual del patio al tomar decisiones sobre la colocación de nuevos contenedores.

Para reducir las necesidades de cómputo, se implementa un paso previo de filtrado, en el que se eliminan los bloques que no son aptos para recibir un contenedor siguiendo las restricciones de la zonificación de los bloques. Esto permite simplificar el proceso de optimización al centrarse solo en los bloques viables para cada contenedor.

Variables de decisión:

- x_{ij} , variable binaria, igual a 1 si el contenedor i se pone en la posición j .
- d_i , días que lleva en contenedor i en el patio.
- t_i , tipo del contenedor i .
- s_i , tamaño del contenedor i .
- m_i , media de días en los que un contenedor similar pasa sin ser solicitado i .
- b_{ij} , días acumulados por los contenedores bloqueados por el contenedor i en posición j .

Función objetivo:

$$\text{Min}Z = \sum_{ij} (-d_i + m_i + b_{ij})x_{ij} \quad (1)$$

Se busca minimizar esta función, ya que mientras menor es el valor, menos “molesta” el contenedor en su posición, si el valor de b es muy alto, es porque se tapan muchos contenedores o uno que lleva muchos días sin ser movido. Mientras menor sea el valor de m es más probable que sea necesario pronto y d se utiliza en caso de reubicación de un contenedor dándole prioridad de accesibilidad a los que llevan mucho tiempo en el patio. (Anexo 1.1)

Restricciones:

$$\sum_j x_{ij} = 1 \quad (2)$$

$$\sum_i x_{ij} \leq 1 \quad (3)$$

$$\sum_i x_{ij} \leq \sum_i x_{i(\text{bloque}, x, y, z-1)} \quad (4)$$

$$x_j = 1, \text{ si ocupado}(j) \quad (5)$$

La “Formula 2” es una restricción para que cada contenedor sea puesto en una sola posición. La “Formula 3” es para que cada posición tenga como mucho un solo contenedor, la “Formula 4” es para que los

contenedores no queden en el aire y respeten la gravedad y finalmente la “Formula 5” es para que las posiciones previamente ocupadas en el bloque queden como ocupadas, es una forma de “sacar una foto” al estado actual del bloque. (Anexo 1.2)

5.2.2 FUNCIÓN DE SALIDA

Para la salida de los contenedores, también se utiliza una variable binaria, que indica si un contenedor será extraído o no. Además, se consideran los movimientos necesarios para sacar el contenedor, los días que lleva en el patio y dos pesos variables, los cuales dependen de la prioridad asignada a los movimientos y a la rotación de los contenedores.

La función de optimización tiene como objetivo principal seleccionar el contenedor que requiera el menor número de movimientos para ser extraído, priorizando aquellos que llevan más días en el patio. De esta manera, se busca optimizar tanto la eficiencia en la manipulación de los contenedores como la rotación del patio, asegurando que los contenedores con mayor tiempo de permanencia sean los primeros en ser retirados, reduciendo así los costos asociados con los movimientos y mejorando el uso del espacio disponible.

En cuanto a las restricciones, es necesario que el contenedor seleccionado sea compatible con lo solicitado, es decir, que cumpla con los requisitos específicos para ser extraído. Además, se establece que solo puede ser elegido un contenedor para su extracción en cada instancia.

Variables de decisión:

- x_i , variable binaria, igual a 1 si se saca el contenedor i .
- m_i , movimientos necesarios para sacar al contenedor i .
- d_i , días del contenedor i en el patio.
- α , peso para importancia de movimientos.
- β , peso para importancia de días.

Función objetivo:

$$MinZ = \sum_i \left(\alpha \cdot \frac{m_i}{10} x_i - \beta \cdot \frac{d_i}{30} (x_i - 1) \right) \quad (6)$$

Restricciones:

$$\sum_i x_i = 1$$

$$x_i \leq \text{compatibilidad}(i)$$

Como se puede observar en la fórmula 6, los movimientos y los días están divididos por ciertos valores. Estos valores representan una aproximación del valor máximo posible que pueden alcanzar cada uno de estos parámetros, lo que permite normalizarlos y trabajar con ambos en una misma escala, facilitando así la comparación y la integración de ambos factores en el modelo de optimización. (Anexo 1.3)

También cabe destacar que la fórmula 2 tiene la flexibilidad de operar tanto bajo el enfoque FIFO (First In, First Out) como LIFO (Last In, First Out), dependiendo de los valores asignados a sus pesos alfa y beta. Si alguno de los pesos toma el valor de 0, la fórmula adopta automáticamente el comportamiento correspondiente a uno de estos métodos de gestión tradicionales, priorizando ya sea la rotación o la salida inmediata de los contenedores. En caso de asignar valores diferentes a ambos pesos, la función logra un

equilibrio entre los enfoques FIFO y LIFO, evaluando cuál es la opción más adecuada según el contexto del patio de contenedores. Esta capacidad de adaptación permite que la fórmula sea útil en una variedad de escenarios, ofreciendo una solución personalizada que combina lo mejor de ambos métodos para elegir el contenedor óptimo a retirar.

5.2.3 MODELO DE OPTIMIZACIÓN

Para la primera fase del estudio, se optó por utilizar PuLP, una biblioteca de programación lineal que permite modelar y resolver problemas de optimización exacta de forma sencilla y eficiente en entornos controlados. Esta herramienta fue aplicada a una versión reducida del problema, donde las dimensiones del modelo y la cantidad de restricciones eran manejables. Los resultados obtenidos en esta etapa fueron satisfactorios: se logró resolver el problema de optimización con rapidez y precisión, validando así la formulación del modelo y permitiendo identificar los parámetros críticos del sistema.

```
Contenedores por altura (z):  
Nivel z=0:  
  Contenedor 17 en posición (0, 0, 0) - Marca: CMA CGM, Días: 0  
  Contenedor 13 en posición (0, 1, 0) - Marca: MSC, Días: 0  
  Contenedor 4 en posición (1, 0, 0) - Marca: Maersk, Días: 0  
  Contenedor 12 en posición (1, 1, 0) - Marca: CMA CGM, Días: 0  
  Contenedor 22 en posición (2, 0, 0) - Marca: Maersk, Días: 0  
  Contenedor 16 en posición (2, 1, 0) - Marca: CMA CGM, Días: 0  
  Contenedor 18 en posición (3, 0, 0) - Marca: CMA CGM, Días: 0  
  Contenedor 3 en posición (3, 1, 0) - Marca: Maersk, Días: 0  
  Contenedor 21 en posición (4, 0, 0) - Marca: MSC, Días: 0  
  Contenedor 15 en posición (4, 1, 0) - Marca: Maersk, Días: 0  
Nivel z=1:  
  Contenedor 8 en posición (0, 0, 1) - Marca: CMA CGM, Días: 0  
  Contenedor 11 en posición (0, 1, 1) - Marca: CMA CGM, Días: 0  
  Contenedor 2 en posición (1, 0, 1) - Marca: CMA CGM, Días: 0  
  Contenedor 20 en posición (2, 0, 1) - Marca: CMA CGM, Días: 0  
  Contenedor 7 en posición (2, 1, 1) - Marca: CMA CGM, Días: 0  
  Contenedor 5 en posición (3, 0, 1) - Marca: CMA CGM, Días: 0  
  Contenedor 19 en posición (3, 1, 1) - Marca: Maersk, Días: 0  
  Contenedor 14 en posición (4, 0, 1) - Marca: Maersk, Días: 0  
  Contenedor 1 en posición (4, 1, 1) - Marca: MSC, Días: 0  
Nivel z=2:  
  Contenedor 23 en posición (0, 0, 2) - Marca: CMA CGM, Días: 0  
  Contenedor 6 en posición (0, 1, 2) - Marca: CMA CGM, Días: 0  
  Contenedor 0 en posición (1, 0, 2) - Marca: CMA CGM, Días: 0  
  Contenedor 9 en posición (2, 0, 2) - Marca: MSC, Días: 0  
  Contenedor 10 en posición (3, 0, 2) - Marca: CMA CGM, Días: 0
```

Figura 3. Resultados de la función de entrada con Pulp.

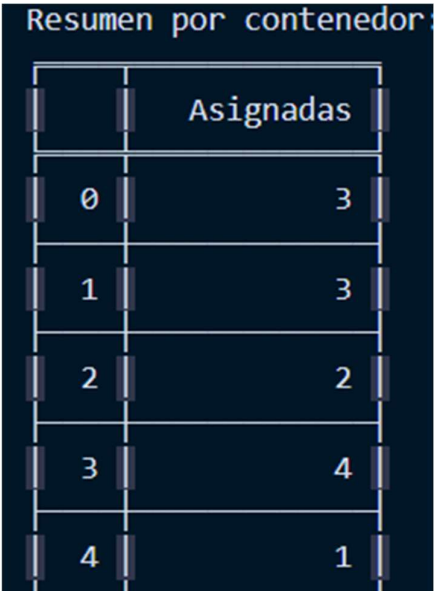
Sin embargo, se anticipó que, al intentar escalar el modelo a un entorno de operación realista, con un mayor volumen de datos y una estructura más dinámica, comenzarían a manifestarse problemas de

escalabilidad. El tiempo de cómputo aumentaría considerablemente, y en algunos casos, el modelo no lograría converger en un tiempo razonable. Estos hallazgos pusieron en evidencia las limitaciones de los métodos exactos para abordar problemas de gran escala en contextos logísticos complejos.

Frente a estas limitaciones, se exploró el uso de métodos heurísticos como una alternativa viable para enfrentar la creciente complejidad del problema. A diferencia de los enfoques exactos, las heurísticas ofrecen mayor flexibilidad, velocidad de cómputo y adaptabilidad, lo que las hace especialmente útiles en escenarios donde la obtención de soluciones óptimas no es necesariamente prioritaria, pero sí lo es alcanzar buenas soluciones en tiempos razonables.

Con este enfoque, se decidió implementar el modelo utilizando Mealpy (Anexos 1.3 - 1.5), una biblioteca en Python especializada en algoritmos metaheurísticos. En particular, se empleó el algoritmo genético, debido a su capacidad para explorar grandes espacios de búsqueda y generar soluciones de alta calidad en problemas no lineales y con múltiples restricciones. La implementación de este algoritmo permitió reducir significativamente los tiempos de cómputo y obtener soluciones competitivas incluso en instancias de mayor tamaño, lo cual validó su idoneidad para aplicaciones en entornos reales.

Al trasladar el problema a Mealpy, se detectó inicialmente un problema en la asignación de las posiciones de los contenedores: algunas ubicaciones quedaban vacías, mientras que otras recibían múltiples contenedores. Este comportamiento se debía tanto a la baja severidad de las penalizaciones aplicadas por no cumplir con las restricciones como a la limitada cantidad de soluciones exploradas por el algoritmo. Para corregir esta situación, se reforzaron las penalizaciones, haciéndolas más estrictas, y se ajustaron los parámetros iniciales del algoritmo, lo que permitió mejorar la calidad de las soluciones y lograr una asignación más coherente y realista de los contenedores.



	Asignadas
0	3
1	3
2	2
3	4
4	1

Figura 4. Muestra de la cantidad de posiciones asignados a un contenedor

24	2
25	1
26	3
27	1
28	1

Figura 5. Muestra de la cantidad de contenedores asignados a una posición

En esta primera iteración de la solución, se utilizaron 1500 épocas y una población inicial de 50. Sin embargo, al observar que la mejor solución global tendía a encontrarse alrededor de la época 600, se decidió reducir la cantidad de épocas y aumentar el tamaño de la población inicial, con el objetivo de explorar una mayor diversidad de soluciones desde las primeras generaciones y optimizar el rendimiento del algoritmo.

Resumen por contenedor:	
	Asignadas
0	1
1	1
2	1
3	1
4	1

Figura 6. Muestra de la cantidad de posiciones asignados a un contenedor posterior a los cambios

24	1
25	0
26	1
27	1
28	1

Figura 7. Muestra de la cantidad de contenedores asignados a una posición posterior a los cambios

Como se puede observar existió una mejora gracias a los cambios previos, aún existen posiciones vacías ya que para las 30 posiciones solo se ingresaron 27 contenedores, esto, debido a que el objetivo en primera instancia es que la función cumpla con las restricciones en su mayoría, ya que, aún existen contenedores con varias posiciones asignadas, pero, es una cantidad manejable que además nos da capacidad de maniobra.

7	3
8	1
9	1

Figura 8. Contenedor con varias posiciones asignadas

En cuanto a los tiempos de ejecución, el algoritmo genético rondaba los 500 segundos variando según la cantidad de datos dados, en cambio SIMPLEX podía pasar desde pocos segundos a varias horas.

Tiempo total de ejecución: 473.20 segundos

Figura 9. Ejemplo del tiempo de ejecución del algoritmo genético

(6751.35 seconds)

Figura 10. Marca de tiempo de simplex

Cabe destacar que el mayor incremento en los tiempos de ejecución del método SIMPLEX se produjo al incorporar contenedores que ya llevaban varios días en el patio. Al enfrentar esta situación, el tiempo de cálculo aumentó considerablemente, evidenciando una mayor sensibilidad del modelo ante condiciones operativas más complejas.

Para evaluar la efectividad del modelo propuesto, se ha diseñado un experimento comparativo frente al enfoque tradicional FIFO, ampliamente utilizado en la gestión de patios de contenedores debido a su simplicidad y facilidad de implementación. Esta estrategia consiste en despachar primero aquellos contenedores que han ingresado con mayor antigüedad, sin considerar factores como la estructura del apilamiento, la accesibilidad o la futura rotación de inventario.

El objetivo de esta comparación es determinar si el modelo heurístico desarrollado, basado en técnicas de optimización, puede ofrecer mejoras sustanciales en términos de eficiencia operativa, reducción de movimientos innecesarios y optimización del espacio disponible. Además, se busca validar si este enfoque puede adaptarse mejor a situaciones más complejas, donde las decisiones deben considerar múltiples variables, como la cantidad de días en el patio, la rotación promedio de contenedores similares, y las penalizaciones por bloqueos en la estructura del patio.

Para llevar a cabo esta comparación, ambas estrategias fueron implementadas dentro de un entorno de simulación desarrollado específicamente para este propósito. En el caso del modelo FIFO, la función de entrada se encarga de posicionar los contenedores en el patio respetando estrictamente el orden de llegada, siguiendo también una política que prioriza el llenado de las columnas interiores antes de utilizar las exteriores (Anexo 1.6). Esta decisión se toma con el fin de evitar la creación de espacios vacíos inaccesibles que podrían aumentar los movimientos requeridos para futuras extracciones.

5.3 INTEGRACIÓN CON DINODOCK

Para integrar el modelo de optimización con DinoDock se prepararon las salidas de la función de optimización de manera que fueran compatibles con los formatos esperados por la API de DinoDock. Esto asegura una comunicación fluida entre el modelo y el sistema, permitiendo la correcta implementación y ejecución de la optimización dentro del entorno de DinoDock.

La forma de validar el funcionamiento de esta parte fue mediante llamadas a la API, verificando que las respuestas fueran las esperadas y que los datos devueltos coincidieran con los resultados previstos. Este proceso de validación garantizó que la integración entre el modelo de optimización y DinoDock funcionara correctamente.

Para la entrada de contenedores, se espera recibir el contenedor como dato de entrada, con información como su tipo, tamaño y tiempo de permanencia. El patio, por su parte, se obtiene directamente desde la base de datos, donde se almacena la disposición actual de los contenedores. Esta información se utiliza para realizar el proceso de optimización, asegurando que la disposición de los nuevos contenedores sea lo más eficiente posible en cuanto a accesibilidad y minimización de movimientos, considerando las restricciones del espacio disponible y los contenedores previamente ubicados.

```
Ingreso de contenedor:
Contenedor(ID: 001, Tipo: Seco, Tamaño: 20, Marca: Evergreen, Fecha Ingreso: 2023-01-07 13:16:00, Estado: 1)
Salida esperada (Bloque, coordenada_x, coordenada_y, coordenada_z)
('A', 0, 0, 2)
PS C:\Users\Alexander\Documents\DinoDock_V0.1>
```

Figura 11. Ingreso contenedor

Para la salida de un contenedor, se consulta información clave como la marca, tipo y tamaño del contenedor que se desea extraer. Estos datos son fundamentales para garantizar que se seleccione el contenedor adecuado según los criterios de optimización establecidos. En la Figura 4 se muestra cómo se ha adaptado el sistema para solicitar esta información desde la consola.

```
Egreso de contenedor con los siguientes parámetros:
Marca: Evergreen
Tipo: Seco
Tamaño: 20
Salida esperada (Contenedor elegido, Bloque, coordenada_x, coordenada_y, coordenada_z)
(Contenedor(ID: 21, Tipo: Refrigerado, Tamaño: 20, Marca: Evergreen, Fecha Ingreso: 2024-08-12 09:00:00, Estado: 1), 'A', 2, 4, 0)
PS C:\Users\Alexander\Documents\DinoDock_V0.1>
```

Figura 12. Salida contenedor

Como se ha mencionado anteriormente, el sistema interactúa con una API, cuya implementación y detalles escapan del alcance de esta tesis. Por lo tanto, en este documento nos centraremos exclusivamente en el proceso de entregar la información solicitada. El objetivo principal aquí es describir cómo se gestionan y procesan los datos relevantes para la entrada y salida de los contenedores, asegurando que se optimicen los movimientos y la disposición en el patio.

6 RESULTADOS

En una primera etapa, los resultados obtenidos mediante la función de optimización fueron equivalentes a los del enfoque FIFO. Esto se debe a que, en sus aspectos fundamentales, ambos métodos comparten una lógica de operación similar. Para esta prueba, se utilizaron valores de 0.5 tanto para los parámetros alfa como beta.

2023-01;6	2023-01;6
2023-02;6	2023-02;6
2023-03;8	2023-03;8
2023-04;9	2023-04;9
2023-05;8	2023-05;8
2023-06;3	2023-06;3
2023-07;6	2023-07;6
2023-08;12	2023-08;12
2023-09;7	2023-09;7
2023-10;6	2023-10;6
2023-11;5	2023-11;5
2023-12;3	2023-12;3
2024-01;4	2024-01;4
2024-02;0	2024-02;0
2024-03;3	2024-03;3

Figura 13. Movimientos agrupados por mes, primero se ve la fecha y luego la cantidad de movimientos

Esto se atribuyó al tamaño limitado de la simulación, por lo que se amplió el área de 'juego' y se agregaron más movimientos para evaluar el comportamiento del modelo en condiciones más complejas. A pesar de estos ajustes, los resultados siguieron siendo similares a los de FIFO, lo que indicó que, aunque la simulación se expandiera, las diferencias entre ambos enfoques no eran suficientes en este contexto inicial.

La simulación se diseñó para replicar el comportamiento de un patio logístico con diferentes cargas operativas. En una primera etapa, se analizaron escenarios con aproximadamente 100 movimientos totales, y posteriormente se incrementó la complejidad hasta alcanzar los 200 movimientos. Utilizando valores constantes para los parámetros alfa y beta (0.5), el modelo de optimización logró reducir los movimientos de 173 (FIFO) a 138, lo que representa una mejora cercana al 20%. Esta diferencia, aunque

moderada, evidencia un impacto operativo relevante, especialmente en patios de alta rotación, donde cada movimiento representa un costo en tiempo y recursos. Este resultado refleja el potencial del modelo para mejorar la eficiencia operativa, validando su utilidad frente a métodos tradicionales.

Total de movimientos realizados: 173

Figura 14. Cantidad de movimientos del enfoque FIFO

Total de movimientos realizados: 138

Figura 15. Cantidad de movimientos del modelo

En la práctica, estos resultados se traducen en una menor cantidad de movimientos de grúa, lo que implica una reducción significativa en varios aspectos operativos: menor tiempo de operación, menor consumo de combustible y, por ende, una disminución en las emisiones de carbono. De esta manera, el modelo no solo mejora la eficiencia logística, sino que también contribuye al cumplimiento de los objetivos ecológicos establecidos por las empresas.

7 CONCLUSIONES

Las pruebas evidenciaron una mejora en el desempeño del modelo en comparación con el enfoque FIFO. Esta ventaja puede incrementarse ajustando los pesos de los parámetros alfa y beta, los cuales permiten adaptar la función de optimización a las condiciones específicas y prioridades operativas del patio, cumpliendo así con el objetivo de reducir los movimientos de contenedores y mejorar su rotación.

El modelo desarrollado logra incrementar la eficiencia en los movimientos de contenedores, contribuyendo tanto a la toma de decisiones en su colocación y extracción como al ámbito ambiental, al reducir el uso de la grúa y, por ende, las emisiones de CO₂.

Los tiempos de ejecución fueron mayores al utilizar la función de optimización en comparación con los métodos tradicionales, como FIFO. Esto indica que, en términos de velocidad, los métodos tradicionales siguen siendo preferibles, ya que proporcionan resultados más rápidos, lo que es crucial en entornos donde la eficiencia temporal es una prioridad.

El método SIMPLEX puede presentar tiempos de ejecución considerablemente largos, especialmente a medida que aumenta la cantidad de contenedores. Sin embargo, resulta útil en escenarios donde se agregan pocos contenedores a la vez. En contraste, el algoritmo genético tiene un tiempo de ejecución base más elevado, pero su crecimiento en tiempo es más moderado en comparación con SIMPLEX a medida que aumenta el volumen de datos, lo que lo hace más escalable para problemas de mayor tamaño.

Si bien la función de optimización ha logrado igualar los resultados de los métodos tradicionales, para obtener una mejora significativa, se recomienda explorar enfoques alternativos, como el aleatorio. Este enfoque podría proporcionar una mayor flexibilidad y eficiencia en ciertas condiciones, permitiendo superar las limitaciones actuales del modelo de optimización, especialmente en términos de tiempos de ejecución. Sin embargo, esto para patios con grandes flujos de contenedores o sin un sistema de planeación de recepción y entrega de contenedores del día.

El modelo podría enriquecerse incorporando variables que se ajusten según la temporada mediante la integración de un modelo de regresión, con el fin de generar predicciones más precisas. Asimismo, incluir

tiempos reales de operación durante la simulación permitiría que los resultados reflejen de manera más fiel la realidad operativa del patio.

8 REFERENCIAS

- [1] ¿Qué es el Cross-docking? Conoce cuándo y cómo se utiliza. www.netlogistik.com. Consultado el 12 de junio de 2025.
- [2] Zonificación de mercancía en una bodega. Ar-racking. <https://www.ar-racking.com>
- [3] Duran, S. (2012). ESTUDIO DEL CICLO DEL CONTENEDOR y SU OPERATIVA EN DEPÓSITOS y TERMINALES [Tesis de diplomado]. Universidad politécnica de cataluña.
- [4] *Random Access Memory (RAM)*. GeeksforGeeks (en inglés estadounidense). 14 de junio de 2021. Consultado el 12 de junio de 2025
- [5] Systems, N. (2022, 18 julio). ¿Cuáles son los métodos para la ubicación de los productos en el almacén? Noega Systems. <https://www.noegasystems.com/blog/logistica/metodos-para-la-ubicacion-de-los-productos-en-el-almacen>
- [6] *Programación lineal*. (s. f.). Universidad de Barcelona. Recuperado 12 de junio de 2025, de <http://www.ub.edu/>
- [7] Msmk. (2024, 22 octubre). Heuristic. MSMK. <https://msmk.university/heuristic/>
- [8] De Armas, Laidy & Bello, Rafael & Morell, Carlos. (2017). Métodos de optimización aplicados al problema de apilamiento de contenedores. *Ciencias Matemáticas*. 31. 25.
- [9] Glover, Fred & Laguna, Manuel. (1999). Tabu search I. 10.1287/ijoc.1.3.190.
- [10] López-Sánchez, Alicia Yesenia, González-Lara, Aida Lucina, & Alcaraz-Corona, Sergio. (2019). Simulación para la optimización de la producción de ejes en la línea de ensamblaje de una empresa de manufactura. *Ingeniería, investigación y tecnología*, 20(1) <https://doi.org/10.22201/fi.25940732e.2019.20n1.005>
- [11] Moraga Rebolledo, Milton, & Baesler Abufarde, Felipe. (2001). USO DE INTELIGENCIA ARTIFICIAL PARA LA OPTIMIZACIÓN DE UN MODELO DE SIMULACIÓN APLICADO A UN PROCESO DE REMANUFACTURA DE PINO RADIATA. *Maderas. Ciencia y tecnología*, 3(1-2), 52-62. <https://dx.doi.org/10.4067/S0718-221X2001000100006>
- [12] Andrés, W. P., De Ciencias Físicas y Matemáticas, F., De Ingeniería Matemática, D., De Ingeniería Industrial, D., Daniel, E. G., Fernando, O. P., & Roberto, C. C. (2009, 30 julio). Optimización del movimiento de contenedores vacíos bajo incertidumbre. <https://repositorio.uchile.cl/handle/2250/102154>

9 ANEXO

1.1 Función de entrada lineal

```
objetivo = lpSum(
    x[i][j] * (-contenedores[i].dias + contenedores[i].media)
    for i in range(n_conts)
    for j in range(n_pos)
) + lpSum(dias_tapados)
prob += objetivo
```

1.2 Restricciones entrada

```
for i in range(n_conts):
    prob += lpSum(x[i][j] for j in range(n_pos)) == 1

    for j in range(n_pos):
        prob += lpSum(x[i][j] for i in range(n_conts)) <= 1

    for j, (x_pos, y_pos, z_pos) in enumerate(pos_map):
        if z_pos > 0:
            try:
                j_lower = pos_map.index((x_pos, y_pos, z_pos - 1))
                for i in range(n_conts):
                    prob += x[i][j] <= lpSum(x[k][j_lower] for k in
range(n_conts))

            except ValueError:
                pass
```

1.3 Función salida lineal

```
prob += pulp.lpSum([
    (self.alpha * m[i] / 10) * x[i] - (self.beta *
contenedores[i][2] / 30) * (x[i] - 1)
    for i in range(n)
])
```

1.4 Función entrada heurística

```
def objective(self, solution):
    try:
```

```

        x = np.array(solution).reshape((self.n_contenedores,
self.n_posiciones))
    except Exception:
        return 1e10

    row_sums = x.sum(axis=1)
    col_sums = x.sum(axis=0)

    penalty = 0

    for s in row_sums:
        if s != 1:
            penalty += 1e6 * abs(1 - s)

    for s in col_sums:
        if s > 1:
            penalty += 1e6 * (s - 1)

    penalty += self.check_stacking_constraint(x)

    self.asignar_solucion_al_bloque(solution)

    z_obj = 0
    for x_idx in range(self.bloque.positions.shape[0]):
        for y_idx in range(self.bloque.positions.shape[1]):
            for z_idx in range(self.bloque.positions.shape[2]):
                cont = self.bloque.positions[x_idx, y_idx, z_idx]
                if cont is not None:
                    z_obj += (-cont.dias + cont.media)
    return z_obj + penalty

```

1.5 Función salida heurística

```

def objective(solution):
    x = np.array(solution)

```

```

total = 0
penalty = 0

for i in range(len(x)):

    suma_x = np.sum(x)
    if suma_x != 1:
        penalty += 1e6 * abs(suma_x - 1) # Penalización proporcional

    mi_term = alpha *
bloque.movimientos_para_extraer(contenedores[i][0][0], contenedores[i][0][1]
, contenedores[i][0][2]) / 10 * x[i]

    di_term = beta * contenedores[i][2] / 30 * (x[i] - 1)
    total += mi_term - di_term

return total + penalty

```

1.6 FIFO

```

idx = 0
total = len(df_contenedores)
for f in range(filas):
    for c in range(columnas):
        for h in range(alturas):
            if idx < total and bloque.comprobar_espacio(f, c, h):

                row = Contenedor(idx, m[idx], d[idx], marca[idx])

                bloque.add_container(row, f, c, h)
                idx += 1
            if idx >= total:
                return

```