

# Métodos de optimización aplicados al problema de apilamiento de contenedores

## Optimization Methods applied to Container Stacking Problem

Laidy De Armas Jacomino<sup>1\*</sup>, Rafael Bello Pérez<sup>2</sup>, Carlos Morell Pérez<sup>2</sup>

**Abstract** En las terminales de contenedores marítimas se toman a diario múltiples decisiones tales como la planificación de la estiba en los buques, la asignación de atraques y grúas, el apilamiento de contenedores en la explanada o patio, etc. El apilamiento de contenedores es considerado por varios autores como una de las decisiones más importantes que puede afectar el desarrollo eficiente del resto de las actividades y procesos. Este proceso en una terminal marítima da lugar a la ocurrencia de distintos problemas de optimización conocidos como carga, descarga y recolocación o premarshalling de contenedores. Estos problemas de optimización han sido resueltos por varios autores usando diversos métodos matemáticos conocidos también como métodos exactos, entre ellos están la programación lineal, programación lineal entera, programación dinámica, etc. Además para su solución se han usado métodos computacionales de Inteligencia Artificial como programación de restricciones, metaheurísticas y técnicas de planificación. La presente revisión bibliográfica precisamente hace énfasis en el análisis de los métodos de optimización empleados para resolver estos problemas. Atendiendo a esto los trabajos analizados se agruparon según el método matemático o computacional empleado por los autores estudiados.

**Abstract** In the maritime container terminals they are taken daily multiple decisions such as planning stowage on ships, allocation of berths and cranes, stacking containers in the courtyard or patio, etc. Container stacking is considered by several authors as one of the most important decisions that can affect the efficient development of other activities and processes. This process in a marine terminal results in the occurrence of different optimization problems known as loading, unloading and relocation or premarshalling container. These optimization problems have been solved by several authors using different mathematical methods known also as exact methods, among them are linear programming, integer linear programming, dynamic programming, etc. In addition to its solution they have been used computational methods of Artificial Intelligence as constraint programming, metaheuristics and planning techniques. This literature review emphasizes precisely the analysis of the optimization methods used to solve these problems. In response to this the studies analyzed were grouped according to the mathematical or computational method used by the authors studied.

### Palabras Clave

Apilamiento de contenedores— Métodos de optimización— Métodos exactos —Metaheurísticas

<sup>1</sup> Facultad de Ciencias Informáticas, Universidad de Ciego de Ávila Máximo Gómez Báez, Ciego de Ávila, Cuba. [laidy@unica.cu](mailto:laidy@unica.cu)

<sup>2</sup> Centro de Estudios de Informática, Universidad Central "Marta Abreu" de Las Villas, Cuba.

\*Autor para Correspondencia

## Introducción

En las Terminales de Contenedores Marítimas (TCM) se toman a diario un grupo de decisiones importantes tales como [35]:

- Planificación de la estiba de contenedores en los buques.
- Asignación de atraques y de grúas de muelle.
- Planificación del transporte del muelle.
- Apilamiento de contenedores.

El apilamiento de contenedores (del inglés Container Stacking Problem) es considerado como una de las decisiones más importantes debido a que causa mayores retrasos y cuellos de botella en una TCM. Esto se debe, entre otras razones, al arribo de diferentes flujos<sup>1</sup> de contenedores y a la ocurrencia de movimientos improductivos de las grúas de patio. Los movimientos improductivos de las grúas ocurren cuando se desea descargar un contenedor, el cual tiene a su vez varios

<sup>1</sup> Aquí se hace referencia a los principales flujos de contenedores importación, exportación y trasbordo.

contenedores encima que deben ser recolocados para obtenerlo. Por ello los operarios y directivos al tomar las decisiones de apilamiento deben tener presente los indicadores de eficiencia a cumplir por la terminal de forma que se pueda optimizar el espacio de almacenamiento en el patio, se disminuya el tiempo de los buques en el atraque, el gasto de combustible, el deterioro de las grúas, etc.

La optimización del proceso de apilamiento es objeto de estudio desde hace varios años por distintos autores del campo académico de conjunto con personal de las TCM. Se registran en la literatura varias soluciones usando diversos métodos de optimización tanto exactos como heurísticos. Estas soluciones son clasificadas atendiendo a las principales decisiones de un patio de contenedores [11, 12]. Otras soluciones son clasificadas de acuerdo a la gestión de los recursos fundamentales a optimizar en un patio de contenedores: las grúas de patio, el espacio de almacenamiento y los vehículos usados para transportar y apilar contenedores [19]. También existen estudios donde se analizan soluciones atendiendo a problemas más específicos del patio de contenedores como los problemas de apilamiento relacionados con la carga, descarga y la recolocación o pre-marshalling de contenedores [8, 16, 33]. El presente estudio, a diferencia de los mencionados anteriormente, hace énfasis en el análisis de los métodos de solución de problemas que han sido usados por diferentes investigadores para la optimización de los problemas de apilamiento de contenedores en el patio de una TCM. La clasificación de los trabajos analizados está acorde a la clasificación del método que se emplea en el mismo. Dentro de los métodos de optimización exactos tratados están: la Programación Lineal (PL), Programación Lineal Entera (PLE), Programación Dinámica (PD) y los métodos computacionales de Inteligencia Artificial (IA) tales como Metaheurísticas, Programación de Restricciones y Técnicas de Planificación. Sobre estos métodos se explican algunas de sus principales características, especificando también para qué tipo de problema de apilamiento se han usado. Por último en las conclusiones se presentan algunas tendencias sobre su uso a partir de la literatura analizada.

## 1. Uso de métodos de optimización exactos en el apilamiento de contenedores

Muchos de los modelos matemáticos desarrollados para formalizar el problema de apilamiento de contenedores han sido resueltos usando métodos matemáticos de Investigación de Operaciones (IO) conocidos como métodos exactos. Estos métodos realizan una búsqueda exhaustiva tratando de identificar la solución óptima global para el problema. Este tipo de búsqueda en problemas pequeños resulta una estrategia aceptable. Sin embargo, en problemas como los que normalmente suceden en la vida real es prácticamente imposible llevarla a cabo [7].

Los métodos matemáticos exactos, principalmente de PL y sus derivados, se han empleado para resolver disímiles prob-

**Cuadro 1.** Términos usados para PL en [20]

$Z$	Valor de la medida global de desempeño
$x_j$	Nivel de la actividad $j$ (para $j = 1, 2, \dots, n$ ).
$c_j$	Incremento en $Z$ que se obtiene al aumentar una unidad en el nivel de la actividad $j$ .
$b_j$	Cantidad de recurso $i$ disponible para asignarse a las actividades (para $i = 1, 2, \dots, m$ ).
$a_{ij}$	Cantidad del recurso $i$ consumido por cada unidad de la actividad $j$ .

lemas de optimización. Entre estos problemas resueltos se encuentra el apilamiento de contenedores en el patio de una TCM. En este epígrafe se analizan varios trabajos donde se hacen uso de estos métodos de optimización y de técnicas de diseño de algoritmos como Ramificación y Acotamiento.

### 1.1 Programación lineal

La PL es uno de los métodos matemáticos ampliamente usados en la IO en las empresas y otras instituciones desde hace varios años. Este método “*involucra la planeación de actividades para obtener un resultado óptimo; esto es, el resultado que mejor alcance la meta especificada —de acuerdo con el modelo matemático— entre todas las alternativas factibles. El adjetivo lineal se refiere a que todas las funciones matemáticas del modelo deben ser lineales*” [20]. Los autores, a partir de la asunción de que la aplicación más común de programación lineal es la asignación de recursos a ciertas actividades, definen una terminología general para este tipo de problema que aparece referida en el cuadro 1.

A partir de esta terminología el modelo plantea el problema en términos de tomar decisiones sobre los niveles de las actividades, por lo que  $x_1, x_2, \dots, x_n$  se llaman variables de decisión. Los valores de  $c_j$ ,  $b_i$  y  $a_{ij}$  (para  $i = 1, 2, \dots, m$  y  $j = 1, 2, \dots, n$ ) son las constantes de entrada al modelo, también conocidas como parámetros del modelo. En un problema general de PL se deben elegir valores de  $x_1, x_2, \dots, x_n$  con el objetivo de

Maximizar o Minimizar  $Z = c_1x_1 + c_2x_2 + \dots + c_nx_n$

Sujeta a las restricciones

$$a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n \leq b_1$$

$$a_{m1}x_1 + a_{m2}x_2 + \dots + a_{mn}x_n \leq b_m$$

$$x_i \in \mathbb{R}$$

El problema de asignación es un tipo especial de problema de PL en el que los asignados son recursos que se destinan a la realización de tareas [20]. El problema de apilamiento de contenedores se puede ver como un problema de asignación de contenedores a posiciones o slot dentro del patio. En este problema los asignados son los contenedores, los cuales solo pueden ser ubicados en una posición de las disponibles en el patio. El objetivo que generalmente se desea optimizar en este caso es encontrar una asignación de contenedores a slots, de

forma que se minimicen los movimientos improductivos de las grúas cuando sean descargados.

El modelo matemático de un problema de asignación usa las variables de decisión binarias (toman valores entre 0 y 1) para especificar si un asignado  $i$  (contenedor) debe realizar la tarea  $j$  (estar en una posición). Es decir  $x_{ij}$  es 1 si se asigna  $i$  para realizar la tarea  $j$ , en otro caso toma valor 0. La función a optimizar según este modelo queda de la forma siguiente:

$$Z = \sum_{i=1}^n \sum_{j=1}^n c_{ij} x_{ij}$$

Sujeta a las restricciones:

$$\sum x_{ij} = 1 \text{ para } i = 1, 2, \dots, n$$

$$\sum x_{ij} = 1 \text{ para } j = 1, 2, \dots, n$$

$$x_{ij} \geq 0 \text{ para toda } i, j$$

En el problema de apilamiento de contenedores  $Z$  es el costo total de asignar los contenedores a las posiciones disponibles dentro del patio. El valor de  $c_{ij}$  significa el costo de asignar un contenedor determinado a una posición en función de los movimientos improductivos que pudiera provocar tal asignación.

En el problema de asignación general uno de los supuestos es que debe existir igual cantidad de asignados  $n$  que de tareas  $n$ . Sin embargo en el apilamiento de contenedores puede que esto se cumpla o puede que existan más posiciones que contenedores. En los modelos matemáticos estudiados se especifica la cantidad máxima de contenedores  $N_c$  y la cantidad máxima de posiciones en el patio  $N_p$ , estas cantidades son generalmente parámetros del modelo.

La función a optimizar para asignar contenedores a posiciones de forma que se minimicen los movimientos improductivos quedaría:

$$Z = \sum_{i=1}^{N_c} \sum_{j=1}^{N_p} c_{ij} x_{ij}$$

Sujeta a las restricciones:

$$\sum_{j=1}^{N_p} x_{ij} = 1 \text{ para } i = 1, 2, \dots, N_p$$

$$\sum_{i=1}^{N_c} x_{ij} = 1 \text{ para } j = 1, 2, \dots, N_c$$

$$x_{ij} \geq 0 \text{ para toda } i, j$$

La primera técnica matemática ampliamente aceptada en el medio de IO fue el Método Simplex de PL, desarrollado en 1947 por el matemático norteamericano George B. Dantzig. Para la implementación de esta técnica y otras de PL se cuenta hoy con varios lenguajes y sistemas informáticos que agilizan la conceptualización del modelo matemático pues en este tienden a crecer considerablemente el número de variables y restricciones. Uno de los lenguajes matemáticos usados para la PL es MPL (del inglés Mathematical Programming Language) y algunos de los sistemas informáticos que se emplean para programar en este lenguaje son: CPLEX, GUROBI, LINGO,

Mathematica y R<sup>2</sup>. Este último software cuenta con diversos paquetes disponibles para la optimización de procesos, entre ellos se encuentran cplexAPI y linprog.

En algunos problemas de PL el supuesto de divisibilidad no se cumple porque algunas o todas las variables de decisión deben restringirse a valores enteros. En este caso al modelo empleado se le conoce como PLE, el cual se diferencia de la PL solo en este aspecto pues las variables de decisión solo toman valores enteros. Por el contrario, si en un problema PL solo es necesario que algunas variables toman valores enteros entonces el modelo se nombra Programación Entera Mixta (PEM). Además existen problemas donde las variables de decisión solo pueden tomar valores booleanos, es decir, 0 y 1 o valores binarios. A estos tipos de problemas se les conoce como Programación Entera Binaria (PEB) o 0-1 de programación entera.

En el artículo científico citado en [47] se presenta un modelo matemático para determinar las posiciones de almacenamiento para los contenedores que arriban a un patio y también para aquellos contenedores que han sido recolocados de una pila hacia otra. Esta investigación fue el primer modelo de PL exitoso para este tipo de problema específico, el cual está orientado al flujo de exportación. La función objetivo de su trabajo busca minimizar el número de remociones o movimientos innecesarios de las grúas de patio cuando se asignan posiciones a los contenedores de exportación que arriban al patio y a los que son recolocados. Además definen varias restricciones que limitan las condiciones físicas del patio, condiciones de estabilidad del apilado, por ejemplo, que un contenedor no pueda estar en el aire sino encima de otro o sobre el suelo, etc.

La PLE se usa para resolver problemas de recolocación o premarshalling de contenedores en los trabajos [32, 17]. La primera investigación de las antes mencionadas ha sido ampliamente referenciada en este campo y ha servido de base para otras investigaciones similares. En esta se usa PLE con un problema de flujo de red con múltiples demandas de flujos embebido (del inglés Multi-commodity flow problem). El principal objetivo que persigue es minimizar el número de movimientos innecesarios de las grúas de patio. Para desarrollar la solución usaron el lenguaje C++ combinado con CPLEX 9.0 y además se auxiliaron del procedimiento Ramas y Cotas (del inglés Branch and Bound (B&B)).

De una forma similar al trabajo referenciado en [32] se modela y resuelve este mismo problema en [17]. La solución ofrecida en este trabajo superó a la anteriormente mencionada en tiempo computacional de ejecución. Esto se debe entre otros factores a que el número de variables y restricciones de su modelo es inferior.

Cuando se tiene un problema de optimización muchas veces se necesitan optimizar varios objetivos de forma si-

<sup>2</sup><http://www-01.ibm.com/software/commerce/optimization/cplex-optimizer/> <http://www.gurobi.com/products/gurobi-optimizer>  
<http://www.lindo.com/>  
<http://www.wolfram.com/mathematica/> <http://cran.r-project.org/>

multánea. Esto significa que se necesita encontrar una solución que tenga en cuenta todos los objetivos y no una solución para cada uno de ellos, conociéndose como problemas de optimización multiobjetivos. Para solucionarlos existen dos enfoques fundamentales: suma ponderada de objetivos y los basados en la no-dominancia del frente de Pareto [1].

En el estudio citado en [48] se propone un modelo de PLE multiobjetivo para determinar las posiciones de almacenamiento de los contenedores de importación. En la función multiobjetivo propuesta se debe minimizar una suma de tres funciones particulares que tienen diferentes factores de peso. La primera función maximiza la circulación de contenedores entre el patio y la zona del muelle de la terminal. La segunda abarca la carga de trabajo de las grúas de patio, la cual debe ser minimizada para alcanzar la eficiencia y reducir los tiempos improductivos de las mismas. Por último con la tercera se desea minimizar los movimientos innecesarios de las grúas cuando se asigna una bahía<sup>3</sup> específica a un grupo de contenedores.

La PEM se usa en [12] para realizar la asignación de posiciones de almacenamiento a los contenedores de exportación en dos fases. En este trabajo [12] al igual que en el [48], se formula y resuelve un modelo matemático multiobjetivo mediante el enfoque de suma ponderada de objetivos pero con dos funciones. Se optimiza la distancia viajada por equipos que transportan los contenedores entre el atraque del puerto de la terminal y la ubicación asignada en el patio. Se asume que la distancia viajada depende del espacio de almacenamiento asignado a los contenedores de cada buque en la zona del patio. El segundo término de su función mide el desbalance en el número de contenedores apilados en cada bahía del patio por razones de estabilidad.

## 1.2 Programación dinámica

La PD (del inglés dynamic programming o multistage programming) es otro de los métodos matemáticos que se usa para resolver el apilamiento de contenedores. La PD no es más que un procedimiento para resolver problemas de optimización dividiéndolos o separándolos en subproblemas más simples [7]. De esta forma resuelve el problema en fases, tratando con todas las opciones en una fase en particular antes de moverse a la siguiente. El nombre se deriva de su popularidad en resolver problemas que requieren tomar decisiones en una secuencia de períodos de tiempo. Fue desarrollada en la década del 50's por el matemático norteamericano Richard Bellman y por ello esta técnica basa su funcionamiento en el principio de optimalidad de Bellman. Este principio dice que *“Una política es óptima si toda subpolítica de la misma es óptima para el problema residual correspondiente a la subpolítica”* [3].

Una política es uno de los conceptos esenciales de este método y no son más que un conjunto de reglas de decisión que

dependen de la etapa y del estado del sistema en cada etapa. Las etapas son subproblemas en los cuales puede dividirse el problema general a resolver. Los estados se consideran como diferentes situaciones en que puede encontrarse el sistema o problema en una etapa determinada a partir del cual se adopta una política de decisión específica. Para pasar de un estado a otro de cada etapa se usan las variables de decisión. Otro de los elementos importantes de un problema de PD es la ecuación funcional o función recursiva para cada etapa, la cual debe ser evaluada en cada una de estas. Los problemas que resuelve la PD pueden ser determinísticos o estocásticos. En el primer caso están los problemas en los cuales no existe incertidumbre o aleatoriedad en cuanto a los parámetros de estos, por consiguiente no necesitan ser estimados pues se conocen.

En las investigaciones [30, 38] se utiliza este método para resolver el apilamiento de contenedores. En la primera se usa para determinar una posición óptima para contenedores de exportación que arriban al patio. Se propone un modelo de optimización multi-fase bajo incertidumbre pues a los contenedores no se les conocen sus pesos con suficiente anterioridad, estos deben ser estimados probabilísticamente.

En la segunda investigación se emplea este mismo método para resolver otro problema de apilamiento, la recolocación de contenedores o premarshalling [38]. Este tipo específico de problema por naturaleza puede ser dividido en varias fases y estas pueden tener varios estados posibles donde se cumplen las restricciones del problema. En este problema generalmente se conoce el esquema inicial de los contenedores en el patio y a través de varias recolocaciones se desea llegar a otro esquema final en el que los contenedores con fecha próxima a ser descargados estén encima en el apilado. Esto generalmente se debe hacer con la menor cantidad de movimientos o recolocaciones posibles para no afectar el rendimiento de las grúas, gastar combustible, entre otros aspectos.

## 2. Uso de métodos computacionales de Inteligencia Artificial en el apilamiento de contenedores

Los métodos matemáticos anteriores usualmente garantizan encontrar la solución óptima a un problema de optimización. Sin embargo, cuando crece el número de variables y ecuaciones del modelo se torna más difícil resolverlo en un tiempo computacional aceptable. Por ello cuando se debe tomar en cuenta el costo computacional y el tiempo de terminación de un estudio se usan técnicas heurísticas. Las técnicas heurísticas son algoritmos que encuentran soluciones de buena calidad para problemas combinatorios complejos explotando el conocimiento del dominio de aplicación, aunque no garantizan encontrar la solución óptima global.

Las metaheurísticas constituyen un tipo específico de método heurístico, término que fue usado por primera vez en el año 1997 por Fred Glover [18]. Los algoritmos metaheurísticos son algoritmos de propósito general, que no dependen del

<sup>3</sup>Consiste en un grupo de pilas de contenedores, las cuales pueden ser relacionadas a partir del número de bahía donde están ubicadas. Las bahías pertenecen a bloque específico del patio y cada bloque puede tener varias bahías de contenedores.

problema y que ofrecen buenos resultados, pero que normalmente no ofrecen la solución óptima al igual que los heurísticos. Algunas metaheurísticas conocidas son:

- Algoritmos Genéticos (AG) (del inglés Genetic Algorithms (GA)) creada por el profesor John Holland en 1975[21].
- Algoritmos de Colonia de Hormigas (del inglés Ant Colony Optimization (ACO)) creada por Marcos Dorigo a finales de la década de los 90 [13].
- Optimización por Enjambre de Partículas (del inglés Particle Swarm Optimization (PSO)) creada por Eberhart y Kennedy en 1995 [29].

Para resolver el problema de apilamiento de contenedores se usan métodos heurísticos específicos del dominio de aplicación, así como la adaptación de alguna de las metaheurísticas mencionadas antes y otras desarrolladas posteriormente.

## 2.1 Empleo de algoritmos heurísticos y de metaheurísticas

El empleo del enfoque heurístico es una alternativa factible y de gran valor hoy día para solucionar problemas de optimización, especialmente en el caso del apilamiento de contenedores donde tiende a crecer el espacio de posibles soluciones. *“Una técnica heurística es un método que busca buenas soluciones (ej. Cercanas al óptimo) con un costo computacional razonable sin ser capaz de garantizar optimalidad...”* [39]. Los algoritmos que emplean heurísticas generalmente son construidos a partir de datos e información del dominio de aplicación y otros como las metaheurísticas siguen procedimientos ya establecidos ajustando sus parámetros al problema de optimización en cuestión.

Los algoritmos heurísticos han sido creados con diversos propósitos para encontrar soluciones al problema de apilamiento de contenedores. Estas heurísticas han sido incorporadas a las soluciones ofrecidas por varios autores independientemente del método de optimización empleado. Las heurísticas encontradas en la literatura fueron agrupadas en tres clases principales atendiendo al objetivo que persiguen las mismas. A continuación se mencionan estas clases y en el cuadro 2 se presentan las referencias bibliográficas de las contribuciones sobre cada una de ellas.

1. Minimizar el número de movimientos improductivos de las grúas cuando se asignan posiciones a los contenedores que arriban al patio y/o a los que han sido recolocados de una posición a otra.
2. Encontrar una secuencia mínima de movimientos de contenedores cuando se hace la descarga de estos de una bahía o cuando se resuelve el problema de recolocación o premarshalling.
3. Minimizar remociones o movimientos improductivos de las grúas de patio para lograr un tiempo computacional menor de la solución ofrecida.

**Cuadro 2.** Referencias sobre heurísticas aplicadas al apilamiento de contenedores.

<b>Clase 1</b>	[34, 23, 47, 46]
<b>Clase 2</b>	[28, 31, 40, 10, 14, 26, 25]
<b>Clase 3</b>	[32, 16]

En la primera clase se muestran varios estudios donde se usan heurísticas para determinar la posición específica de un contenedor dentro de un número significativamente grande de posibles posiciones que pudieran tener. No solo se usan para determinar la posición de un contenedor que arriba por primera vez al patio y necesita un lugar donde ubicarse sino también para aquellos que han sido recolocados producto de una remoción. Estos autores demuestran que luego que un contenedor es recolocado la estrategia que se siga determina en gran medida los movimientos improductivos que se harán en el futuro.

Otras heurísticas en la clase dos se orientan a encontrar una secuencia mínima de movimientos a la hora de realizar la descarga o la recolocación de contenedores. A diferencia de las heurísticas de la clase uno donde lo que interesaba era brindar una solución que disminuya el número de movimientos improductivos de las grúas, estas otras buscan una secuencia de pasos o movimientos óptima. Por último, en la clase tres se encuentran aquellas heurísticas cuyo objetivo es minimizar la cantidad de remociones para lograr un tiempo computacional menor que el que se alcanza con otros métodos. Entre las metaheurísticas usadas para resolver problemas de apilamiento de contenedores están los AG. Los AG son métodos de búsqueda basados en los principios de la selección natural y la genética. Estos incluyen conceptos como cromosomas, genes, población, operadores de cruce, mutación, etc. En el cuadro 3 se muestran varios de estos términos asociados comúnmente a los AG y sus significados.

**Cuadro 3.** Términos asociados a los AG.

<b>Cromosoma</b>	Una solución candidata.
<b>Gen</b>	Parámetros o variables del problema.
<b>Población</b>	Grupo de soluciones candidatas del problema.
<b>Fitness</b>	Grado de aptitud o calidad del cromosoma o de la solución.
<b>Selección</b>	Escoger individuos o soluciones basadas en su fitness.
<b>Cruzamiento</b>	Se toman dos padres, se intercambian partes de estos y se generan dos hijos.
<b>Mutación</b>	Se selecciona uno o varios genes del cromosoma de manera aleatoria y se modifican sus valores por otros de su dominio.

Los tres últimos elementos del cuadro 3 constituyen los operadores de búsqueda genéticos más destacados. El operador de selección o reproducción permite crear un conjunto intermedio de individuos a partir de la población actual. A partir de la selección de los mejores individuos de la población se aplica el operador de cruce, el cual permite combinar pares de individuos para formar soluciones nuevas intercambiando partes de un par de cromosomas. Por último, el operador de mutación se usa para modificar el valor de uno o varios genes del cromosoma por otros valores del dominio del gen. Este operador genera diversidad entre la población pero normalmente es usado con menos probabilidad que el de cruce como sucede en la vida real. Existen distintos tipos de operadores de cruce y mutación así como formas para realizar la selección de los individuos o cromosomas según la forma de representar el cromosoma, por ejemplo, binario, real, discreto, etc.

**El proceso general de un AG se puede describir de la forma siguiente:**

Inicialmente se generan aleatoriamente o usando otras vías soluciones o cromosomas para el problema que se resuelve. Luego se realizan varias iteraciones del algoritmo. En estas se evalúan las soluciones candidatas o cromosomas de la población y se crea una población intermedia. A partir de esta población intermedia se realiza el cruce y de forma ocasional mutaciones. La condición de parada puede ser un número de generaciones determinado, un criterio basado en la calidad de las soluciones, o un criterio sobre la razón de cambio de la calidad de las soluciones.

De forma general los principales parámetros que se especifican cuando se adapta un AG al dominio de aplicación son:

- **Tamaño de la población.** Indica la cantidad de soluciones que forman la población.
- **Número de generaciones.** Cantidad de iteraciones que realizará el algoritmo para alcanzar una solución satisfactoria.
- **Probabilidad de cruce.** Valor entre 0 y 1 que permite determinar si procede o no el cruce. Usualmente como la frecuencia de cruce es alta este debe ser un valor alto.
- **Probabilidad de mutación.** Valor entre 0 y 1 que permite determinar si procede o no la mutación. Usualmente como la frecuencia de mutación es pequeña este debe ser un valor pequeño.

La metaheurística AG ha sido una de las más empleadas para resolver el problema de apilamiento de contenedores, fundamentalmente para resolver el problema de carga. En los trabajos citados en [5, 4, 24, 45, 37, 27] se demuestra su aplicación a diversos problemas reales relacionados con el apilamiento de contenedores. Los autores del primer trabajo citado [5] fueron pioneros en aplicar esta metaheurística al problema de apilamiento de carga. Ellos proponen un procedimiento de generación de soluciones aleatorias para la población inicial. Usan un operador aritmético de cruzamiento para obtener

nuevas soluciones y como operador de mutación Stepping Stone Mutation (SSM). A diferencia de trabajos similares posteriores ellos emplean dos criterios de terminación del AG: el número máximo de generaciones transcurridas y la desviación estándar del valor de fitness de los cromosomas en la generación actual.

En la investigación [4] se crea la población inicial de forma aleatoria pero respetando las restricciones del modelo matemático del problema. El operador de cruzamiento adoptado consistió en escoger dos padres de la población inicial mediante el método de selección de la ruleta (del inglés roulette-wheel) y de forma aleatoria los cruzaron en tres puntos (p crois-x, p crois-y, p crois-z) los cuales indican las coordenadas donde se encuentra un contenedor en el patio. El criterio de terminación del algoritmo fue cuando la calidad de la solución no mejorara luego de  $N$  iteraciones. Los resultados de este algoritmo fueron verificados evaluando la influencia del número de tipos de contenedores, el criterio de terminación a través de varios números de iteraciones diferentes y el tamaño de la población usado. Estos usaron una probabilidad de cruzamiento igual a [27] sin embargo equidistaron en la probabilidad de mutación. La mayoría de los autores usaron como método de selección de los individuos la ruleta. Por otro lado los criterios de terminación del AG fueron generalmente cuando se alcanza un número máximo de generaciones transcurridas o cuando la solución no ha mejorado luego de un número determinado de generaciones.

La representación de las soluciones o cromosomas en cada trabajo se realiza de forma diferente. En los problemas de carga donde se desea asignar una posición a cada contenedor que arriba al patio un cromosoma se representa como posibles posiciones o slots para los contenedores, que respeten las restricciones del modelo. Las posiciones sugieren a los operadores del patio coordenadas específicas en el patio  $(x, y, z, b)$ . La coordenada  $b$  indica el número del bloque,  $x$  la pila,  $y$  el número de contenedor en esa pila, comenzando a contar desde abajo hacia arriba en la pila y  $z$  la bahía de ese bloque.

Para resolver el problema de apilamiento de contenedores también se usan otras metaheurísticas tales como:

- ACO [37].
- Algoritmo de Abejas (del inglés Bee algorithm o BA) [36].
- Método Corridor (del inglés Corridor Method o CM) [9, 10].

## 2.2 Otras técnicas de IA usadas

Para modelar y optimizar el apilamiento de contenedores se emplean también las tecnologías de programación de restricciones conocidas como Problemas de Satisfacción de Restricciones (del inglés Constraint Satisfaction Problem o CSPs) y las técnicas de planificación de Inteligencia Artificial (AI Planning). La resolución de un CSPs consta de dos fases diferentes según [43]:



- Modelar el problema como un problema de satisfacción de restricciones.
- Procesar el problema de satisfacción de restricciones resultante.

La modelación del problema se hace mediante una sintaxis de CSPs, es decir, mediante la definición de las variables, dominios y restricciones del CSPs. Una vez que se tiene el problema modelado como CSPs existen dos maneras según los propios autores de procesar las restricciones usando:

- Técnicas de consistencia. Se trata de técnicas para la resolución de CSPs basadas en la eliminación de valores inconsistentes de los dominios de las variables.
- Algoritmos de búsqueda. Estos algoritmos se basan en la exploración sistemática del espacio de soluciones hasta encontrar una solución o probar que no existe tal solución.

Una solución en este tipo de problema es una asignación de valores a todas las variables de forma que se satisfagan todas las restricciones.

Esta técnica computacional está estrechamente relacionada con los métodos heurísticos pues casi siempre se necesita saber el orden en el que se van a analizar las variables, así como el orden en el cual estas se van a instanciar. Otra variante para la cual son usadas las técnicas heurísticas en estos problemas es para la ordenación de las restricciones del problema. Por otro lado las metaheurísticas también se han destacado para la búsqueda de soluciones cuando el problema a resolver es de gran magnitud y por lo tanto de gran complejidad computacional.

Las investigaciones encontradas en la literatura que resuelven el problema de apilamiento mediante CSPs son citadas en [22, 41]. En la primera se ofrece solución al problema de carga específicamente y en la postrera al de premarshalling. En el primer estudio se provee de un modelo de CSPs donde las variables de decisión son los contenedores que necesitan ser apilados, las ubicaciones en las pilas del patio constituyen el dominio o valores posibles a tomar para estas variables y las restricciones del modelo definidas como restricciones del CSPs propiamente. A partir de esta modelación los autores diseñaron una solución híbrida entre CSPs y el algoritmo de aproximación del Mejor Ajuste (del inglés Best-Fit approximation).

En el trabajo [41] se definen dos tipos de variables atendiendo a la naturaleza del tipo de problema de apilamiento que ellos resuelven, recolocación o premarshalling. Por ello tiene una variable que representa el estado de una bahía de contenedores luego de que se realiza un movimiento de una posición hacia otra y otra variable binaria que indica si un contenedor ha sido movido o no a una determinada posición. El dominio de las variables son las prioridades de los contenedores. Proponen además una variante de modelación de CSPs más robusta donde se prevé incertidumbre en el arribo de los buques a las

terminales marítimas. En este segundo caso el dominio ya no serían las prioridades de los contenedores a apilar sino el id de los contenedores. Las variables por el contrario se mantienen iguales.

Los trabajos [22, 41] coinciden en que presentan a la comunidad científica de CSPs el problema de apilamiento en años consecutivos, en el primer caso el de carga y en el segundo el de premarshalling. En los dos trabajos se dejan varios temas por tratar en esta línea para mejorar los resultados obtenidos a partir de su integración con otras técnicas computacionales. Además es importante resaltar que la modelación de un CSPs depende del patio de contenedores que se trate por lo que siempre pueden existir otras variantes de modelar las variables, dominios y restricciones, así como métodos o heurísticas nuevas para seleccionar las variables y valores.

Las técnicas de planificación de IA son otras de las formas en que se ha resuelto el apilamiento de contenedores atendiendo a su similitud con el problema de planificación clásico *Blocks World*.

La planificación es el problema de sintetizar un curso de acciones que, cuando son ejecutadas, conducen a un agente desde un estado inicial hasta el estado objetivo deseado. La síntesis de planes de forma automática ha sido una meta de investigación importante en la IA por más de 30 años. Informalmente, un plan puede ser visto como una serie de acciones que el agente decide sobre la base de sus metas finales, la información sobre el estado actual del ambiente, y la dinámica de evolución de ese ambiente [6].

Un problema clásico de planificación se representa mediante la definición de:

- Un conjunto enumerable de estados.
- Un conjunto de acciones.
- Un conjunto de eventos.
- Una función de transición de estados.

Los planificadores pueden ser específicos o independientes del dominio de aplicación [15]. En el caso de los primeros son diseñados para un problema en particular y no suelen trabajar bien en otro contexto. El planificador independiente del dominio por el contrario trabaja en cualquier dominio de planificación pues no usa conocimiento específico de este, solo las definiciones de las acciones básicas.

El lenguaje para formalizar lo anterior es una sintaxis ampliamente aceptada PDDL (del inglés Planning Domain Definition Language) [15]. Este lenguaje ha sido usado en varias competencias de planificadores las dos últimas ICAPS 2011 e ICAPS 2014 <sup>4</sup>, séptima y octava edición respectivamente.

A partir de la revisión realizada se encontraron los trabajos [44, 42, 2] donde se emplean las técnicas de planificación de IA usando diferentes lenguajes para modelar el problema. En [44, 42] se definen los tipos de objetos a manipular, en este dominio los objetos son los contenedores y las pilas donde

<sup>4</sup><https://helios.hud.ac.uk/scommv/IPC-14/index.html>

estos pueden estar. Los tipos de predicados para describir este dominio también se especifican, donde ?x y ?y, son parámetros de los predicados:

- on ?x - container ?y - (either row container). Este predicado indica que el contenedor ?x está sobre el contenedor ?y, el cual también puede estar directamente sobre el suelo.
- clear ?x - (either row container). Este predicado indica que ?x, el cual puede ser un slot o un contenedor, está vacío, es decir, no hay contenedores apilados sobre este.
- crane-empty. Indica que la grúa de patio usada para mover los contenedores no está moviendo ningún contenedor.
- holding ?x - container. Este predicado indica que la grúa está moviendo el contenedor ?x.

En el trabajo también se definen un grupo de acciones para mover un contenedor de un lugar hacia otro del patio. En este caso se agrega el parámetro ?r:

- pick (?x - container ?r - row). Con esta acción la grúa toma el contenedor ?x que está en el suelo de la pila ?r.
- put (?x - container ?r - row). La grúa pone el contenedor ?x en la pila ?r.
- unstack (?x ?y - container ?r - row). La grúa descarga el contenedor ?x que está encima del contenedor ?y.
- stack (?x ?y - container ?r - row). La grúa apila el contenedor ?x que está sosteniendo, sobre el contenedor ?y en la pila ?r.

El estado inicial especificado en este trabajo consiste en un esquema inicial de contenedores que han sido apilados en el patio con anterioridad. De estos contenedores un grupo son seleccionados para ser asignados a las posiciones en el tope de las pilas o debajo de otros contenedores seleccionados. Esto se hace para cuando se descarguen estos contenedores los movimientos improductivos de las grúas sean mínimos, lo cual constituye la función a optimizar en este estudio.

El planificador que proponen incluye un Escalador de colinas reforzado (del inglés Enforced Hill-Climbing) el cual permite encontrar soluciones rápidamente. Esta propuesta es mejorada con la inclusión de una función heurística dependiente del dominio de aplicación en el trabajo [44]. Los resultados obtenidos con las técnicas heurísticas propias del dominio de aplicación en el planificador superan a las obtenidas por la variante independiente del dominio.

En el trabajo [2] se usan las técnicas de planificación de IA para optimizar las operaciones de carga y descarga de contenedores en una terminal pequeña, operada por grúas más pequeñas que las usadas en el estudio anterior. Para representar el problema de planificación se usa el lenguaje STRIPS<sup>5</sup>

<sup>5</sup>STRIPS es un generador de planes automatizado. Este mismo nombre fue utilizado más tarde para referirse al lenguaje formal de las entradas de este generador de planes.

(del inglés Stanford Research Institute Problem Solver) en lugar de PDDL. Esta propuesta se adapta mejor a las condiciones actuales de Cuba y de los Centros de Carga y Descarga de Contenedores (CCD) que esta tiene en el interior del país.

## Conclusiones

A partir de la presente investigación se comprobó que los problemas de optimización de apilamiento de contenedores han sido resueltos a través de un variado grupo de métodos tanto matemáticos como computacionales. Los métodos matemáticos exactos empleados fueron generalmente la PL, PLE, PEM y PD. Estos métodos han sido usados principalmente cuando la cantidad de instancias del problema es pequeña o mediana, a diferencia de las metaheurísticas las cuales se han usado cuando existen problemas combinatorios complejos. De esta forma los estudios realizados con métodos exactos han servido en varias ocasiones como comparación o como base para la aplicación de métodos de IA.

Dentro de los métodos computacionales registrados en la literatura revisada se encontraron los heurísticos, las metaheurísticas, la programación de restricciones y las técnicas de planificación. De las metaheurísticas analizadas, los AG han sido uno de los más empleados para resolver este tipo de problema. Esto fue comprobado a partir del estudio de varias soluciones sobre el apilamiento de contenedores implementados con esta metaheurística.

Otra tendencia encontrada a partir del estudio es que existen diferencias en la aplicación de metaheurísticas sobre los tipos de problemas de apilamiento carga, descarga y premarshalling. Esto se evidencia en que los AG han sido más usados para resolver problemas de carga donde se debe asignar una ubicación a cada contenedor, mientras que CM solo ha sido usada en problemas de descarga. Esto no significa en ninguna medida que no puedan ser empleadas para resolver problemas distintos.

En resumen se puede afirmar a partir del estudio que aún existen temas abiertos a explorar en cuanto al uso de estos métodos de optimización para el apilamiento de contenedores. Esto se fundamenta en que cada problema de optimización se ajusta a un patio de contenedores en la mayoría de los casos, al tipo de equipamiento para manipular los contenedores, a las restricciones y objetivos específicos del modelo, etc. Esto hace que cada modelación sea diferente y requiera por tanto que las heurísticas, las estrategias de búsqueda a emplear y otras técnicas también necesiten evaluarse para cada contexto en particular.

## Agradecimientos

A los compañeros del Centro de Carga y Descarga de Contenedores (CCD) de Almacenes Universales SA. en Villa Clara por su cooperación en la investigación.



## Referencias

- [1] Multi-Objective Optimization. In Edmund K. Burke and Graham Kendall, editors, *Search methodologies. Introductory tutorial in optimization and Decision Support Techniques*, pages 274–316. Springer, 2005. 1.1
- [2] Galuzca Adam and Daniec Krysztof. Optimization of Re-handling and Load/Unload Operations in Small Container Terminal Operated by Reachstackers. *Latest Trends on Systems*, 2:616–621, 2010. 2.2
- [3] Pedro Félix Alvarez Rivero. *Programación Dinámica*. ENPES, 1991. 1.2
- [4] Imen Ayachi, Ryan Kammarti, M. Ksouri, and Pierre Borne. A Genetic algorithm to solve the container storage space allocation problem. pages 1–4, 2010. 2.1
- [5] Mohammad Bazzazi, Nikbakhsh Javadian, and Nima Safaei. A genetic algorithm to solve the storage space allocation problem in a container terminal. *Computers & Industrial Engineering*, 56:44–52, 2009. 2.1
- [6] Rafael Esteban Bello Perez, Zoila Zenaida Garcia Valdivia, Maria Matilde Garcia Lorenzo, and Antonio Reynoso Lobato. *Aplicaciones de la Inteligencia Artificial*. México, second edition, 2002. 2.2
- [7] Edmund K. Burke and Graham Kendall. *Search methodologies. Introductory tutorial in optimization and Decision Support Techniques*. Springer, 2005. 1, 1.2
- [8] Marco Caserta, Silvia Schwarze, and Stefan Voss. Container Rehandling at Maritime Container Terminals. In J. W. Böse, editor, *Handbook of Terminal Planning*, pages 247–269. Springer Science+Business Media, 2011. (document)
- [9] Marco Caserta and Stefan Voss. A Corridor Method-Based Algorithm for the Pre-marshalling Problem. In Giacobini, editor, *Workshops on Applications of Evolutionary Computation*, pages 788–797. Springer Berlin Heidelberg, 2009. 2.1
- [10] Marco Caserta, Stefan Voss, and Moche Sniedovich. Applying the corridor method to a blocks relocation problem. *OR Spectrum*, 33(4):915–929, 2011. 2, 2.1
- [11] Chuanyu Chen, Wen-Jing Hsu, and Shell-Ying Huang. Simulation and Optimization of Container Yard Operations: A survey. In *Proc. of the International Conference on Port and Maritime R and D and Technology*, pages 23–29, 2003. (document)
- [12] Lu Chen and Zhiqiang Lu. The storage location assignment problem for outbound containers in a maritime terminal. *International Journal of Production Economics*, 135(1):73–80, 2012. (document), 1.1
- [13] Marco Dorigo, Gianni Di Caro, and Luca M. Gambardella. Ant Algorithms for Discrete Optimization. *Artificial Life*, 5(3):137–172, 1999. 2
- [14] Christopher Expósito Izquierdo, Belén Melián Batista, and J. Marcos Moreno Vega. Una heurística eficaz para problemas de pre-marshalling en una terminal marítima de contenedores. 2012. 2
- [15] Malik Ghallab, Dana Nau, and Paolo Traverso. *Automated Planning. Theory and Practice*. Elsevier Inc., 2004. 2.2
- [16] Mohamed Gheith, Amr Eltawil, N. A. Harraz, and S. Mizuno. Container Pre-marshalling Problem: A Review and Solution Framework. In *Proc. of the IEEE International Conference of Industrial Engineering and Engineering Management*, pages 283–287, Hong Kong, 2012. (document), 2
- [17] Mohamed Gheith, Amr Eltawil, N. A. Harraz, and S. Mizuno. An Integer Programming Formulation and Solution for the Container Pre-marshalling Problem. pages 2047–2056, Istanbul, Turkey, 2014. 1.1
- [18] Fred Glover. Tabu search and adaptive memory programming—advances, applications and challenges. In *Interfaces in computer science and operations research*, pages 1–75. Springer US, 1997. 2
- [19] Loo Hay Lee, Xijia Jiang, Lu Zhen, and Ek Peng Chew. A Review on Yard Management in Container Terminals. *Industrial Engineering & Management Systems*, 12(4):289–305, 2013. (document)
- [20] Frederick S. Hillier and Gerald J. Lieberman. *Introduction to operations research*. The McGraw-Hill Companies, Inc, ninth edition, 2010. 1, 1.1, 1.1
- [21] Jhon Holland. *Adaptive in natural and artificial systems*. Ann Arbor, Univ. Michigan Press, 1975. 2
- [22] Dong-liang Hou and Fang-rong Chen. Constraint Satisfaction Technology for Stacking Problem with Ordered Constraints. *Procedia Engineering*, 29:3317–3321, 2012. 2.2
- [23] Kap Hwan Kim and Gyu-Pyo Hong. A heuristic rule for relocating blocks. *Computers & Operations Research*, 33:940–954, 2006. 2
- [24] Hyeonuk Jang, Ri Choe, and Kwang Ryel Ryu. Deriving a Robust Policy for Container Stacking Using a Noise-Tolerant Genetic Algorithm. In *Proc. of the 2012 ACM Research in Applied Computation Symposium*, pages 31–36, San Antonio, TX, USA., 2012. 2.1
- [25] Bo Jin, Wenbin Zhu, and Andrew Lim. Solving the container relocation problem by an improved greedy look-ahead heuristic. *European Journal of Operational Research*, 240(3):837–847, 2015. 2

- [26] Raka Jovanovic, Milan Tuba, and Stefan Voss. A Multi-Heuristic Approach for Solving the Pre-Marshalling Problem. 2015. 2
- [27] Shuding Kang and Weimin Wu. A Genetic Algorithm to Solve the Containers Location Allocation Problem in Rear Storage Yard. In *Proc. of the IEEE 12th International Conference on Networking, Sensing and Control*, pages 33–38, Taipei, Taiwan, 2015. 2.1
- [28] M. Kefi, O. Korbaa, K. Ghedira, and P. Yim. Heuristic based model for container stacking problem. In *Proc. of the 19 International Conference on Production Research*, volume 7, 2007. 2
- [29] James Kennedy and Russell Eberhart. A New Optimizer Using Particle Swarm Theory. In *Proc. of the Sixth International Symposium on Micro Machine and Human Science*, 1995. 2
- [30] Kap Hwan Kim, Young Man Park, and Kwang-Ryul Ryu. Deriving decision rules to locate export containers in container yards. *European Journal of Operational Research*, 124:89–101, 2000. 1.2
- [31] Yusin Lee and Shih-Liang Chao. A neighborhood search heuristic for pre-marshalling export containers. *European Journal of Operational Research*, 196(2):468–475, 2009. 2
- [32] Yusin Lee and Nai-Yun Hsu. An optimization model for the container pre-marshalling problem. *Computers & Operations Research*, 34(11):3295 – 3313, 2007. 1.1, 2
- [33] Jana Lehnfeld and Sigrid Knust. Loading, unloading and premarshalling of stacks in storage areas: Survey and classification. *European Journal of Operational Research*, 239(2):297– 312, 2014. (document)
- [34] Richard Linn, Jiyin Liu, Yat-wah Wan, and Katta G. Murty. A decision support system for operations in a container terminal. *Decision Support Systems*, 39:309 – 332, 2005. 2
- [35] Patrick J.M Meersmans and Rommert Dekker. Operation Research supports container handling. Econometric Institute Report EI 2001-22, Econometric Institute, Erasmus University Rotterdam, 2001. (document)
- [36] Ndèye Fatma Ndiaye, Adnan Yassine, and Ibrahima Diarrassouba. Hybrid Algorithms to Solve the Container Stacking Problem at Seaport. *Journal of Mathematics, Statistics and Operations Research (JMSOR)*, 2(2):44–56, 2014. 2.1
- [37] Ndèye Fatma Ndiaye, Adnan Yassine, and Ibrahima Diarrassouba. A Hybrid Ant Colony and Genetic Algorithm to solve the Container Stacking Problem at Seaport Terminal. In *Proc. of the 2014 International Conference on Advanced Logistics and Transport (ICALT)*, pages 247–252. IEEE, 2014. 2.1
- [38] Matthias Prandtstetter. A Dynamic Programming Based Branch-And-Bound Algorithm for the Container Pre-Marshalling Problem. Technical report, AIT Austrian Institute of Technology,, 2013. 1.2
- [39] C. R. Reeves. Modern Heuristic Search Methods. In *Modern heuristic techniques*, pages 1–25. Wiley, New York, 1996. 2.1
- [40] Rui Jorge Rei and Joao Pedro Pedroso. Heuristic search for the stacking problem. *Internacional Transactions in Operational Research*, 19(3):379–395, 2011. 2
- [41] Andrea Rendl and Matthias Prandtstetter. Constraint Models for the Container Pre-Marshaling Problem. In *Proc. of the ModRef 2013: The Twelfth International Workshop on Constraint Modelling and Reformulation*, pages 44–55, Uppsala, Sweden, 2013. 2.2
- [42] Mario Rodriguez-Molins, Miguel A. Salido, and Federico Barber. Intelligent planning for allocating containers in maritime terminals. *Expert Systems with Applications*, 39(1):978–989, 2012. 2.2
- [43] Miguel A. Salido and Federico Barber. Introduction to constraint programming. *Revista Iberoamericana de Inteligencia Artificial*, (20):13–30, 2003. 2.2
- [44] Miguel A. Salido, Oscar Sapena, and Federico Barber. An Artificial Intelligence Planning tool for The Container Stacking Problem. pages 1–4, September 2009. 2.2
- [45] Phatchara Sriphrabu, Kanchana Sethanan, and Banchar Arnonkijpanich. A Solution of the Container Stacking Problem by Genetic Algorithm. *IACSIT International Journal of Engineering and Technology*, 5(1):45–49, 2013. 2.1
- [46] Lixin Tang, Wei Jiang, Jiyin Liu, and Yun Dong. Research into container reshuffling and stacking problems in container terminal yards. *IIE Transactions*, 47(7):751–766, 2015. 2
- [47] Yat-wah Wan, Jiyin Liu, and Pei-Chun Tsai. The Assignment of Storage Locations to Containers for a Container Stack. *Naval Research Logistics*, 56(8):700–713, 2009. 1.1, 2
- [48] Xiaoming Yang, Ning Zhao, Zhicheng Bian, Jiaqi Chai, and Mi Chao. An Intelligent Storage Determining Method for Inbound Containers in Container Terminals. *Journal of Coastal Research*, 73(sp1):197–204, 2015. 1.1