

PONTIFICIA UNIVERSIDAD CATÓLICA DE VALPARAÍSO
FACULTAD DE INGENIERÍA
ESCUELA DE INGENIERÍA INFORMÁTICA

**Un Algoritmo Genético para Resolver el Yard
Allocation Problem en el Terminal Portuario de Arica**

DIEGO MAURICIO RUIZ ORTEGA

INFORME FINAL DEL PROYECTO
PARA OPTAR AL TÍTULO PROFESIONAL DE
INGENIERO DE EJECUCIÓN EN INFORMÁTICA

Diciembre, 2015

**PONTIFICIA UNIVERSIDAD CATÓLICA DE VALPARAÍSO
FACULTAD DE INGENIERÍA
ESCUELA DE INGENIERÍA INFORMÁTICA**

**Un Algoritmo Genético para Resolver el Yard
Allocation Problem en el Terminal Portuario de Arica**

DIEGO MAURICIO RUIZ ORTEGA

Profesor Guía: **Claudio Cubillos Figueroa**

Profesor Co-referente: **Ignacio Araya Zamorano**

Carrera: **Ingeniería de Ejecución en Informática**

Diciembre, 2015

El presente proyecto de título va en dedicatoria a toda mi familia, por su incondicional apoyo en los momentos de dificultad, por su amor y lealtad.

A mis tutores, Claudio Cubillos, gracias por confiar en mí y darme la oportunidad de autosuperarme como profesional y por sobre todo, por abrirme las puertas hacia la continuidad de mis estudios en la universidad. A Ignacio Araya, gracias por tomar la opción de ser mi co-referente y sus consejos para la elaboración del proyecto.

A Daniel Cabrera, por tus ánimos, apoyo y simpatía a lo largo de este proyecto, muchas gracias por haberme motivado en los momentos necesarios, contigo empezó todo.

A Rosa Gonzalez, por compartir su experiencia, su dedicación y motivación por las operaciones marítimas y nutrirme de información valiosa para el desarrollo del proyecto.

A mis formadores y colegas en la etapa como estudiante y persona dentro de la Escuela de Ingeniería Informática, gracias por sus enseñanzas, amistad y apoyo.

A mis amigos, gracias aunque a pesar de la distancia, siempre tenían una palabra de aliento.

A mis detractores, sin ellos no habría sido posible el éxito de este proyecto.

Finalmente, dedico este proyecto de título a la persona que dedique de su tiempo para leer sobre lo contenido en este trabajo, espero que sea de utilidad y pueda continuar con la línea de investigación que con mucho esfuerzo asentamos.

- Diego Ruiz Ortega

Resumen

Yard Allocation Problem (YAP) es un problema observado en la mayoría de los puertos marítimos alrededor del mundo, y consiste en definir un proceso de asignación temporal de contenedores de carga, tanto de exportación como de importación, dentro de bloques de acopio de carga ubicados en la zona de operaciones de los puertos marítimos. El objetivo central de la resolución del YAP corresponde a agilizar las operaciones tanto del terminal portuario como de las empresas navieras usuarias del terminal mismo, disminuyendo los costos operacionales y aumentando la eficiencia global. En este sentido, este trabajo propone resolver el YAP utilizando un algoritmo genético basado en restricciones para cada contenedor. El escenario de test está conformado por datos oficiales de operación del Terminal Portuario de Arica - Chile (TPA). El algoritmo genético propuesto representa una aproximación novedosa y prometedora, toda vez que permite obtener soluciones de manera robustas sin dejar de satisfacer el conjunto de restricciones que plantea el modelo general de operación.

Palabras Claves - Yard Allocation Problem, algoritmo genético, contenedor

Abstract

Yard Allocation Problem (YAP) is a frequently observed problem in port logistic operations, and consists in a temporary assignment of cargo containers, both export and import, into storage blocks located in the operation zone of seaports. The principal objective of YAP resolution is to improve the operations efficiency of both, the port terminal and the shipping companies that use the seaport, which allow to decrease operational costs and increasing the global efficiency. In this sense, this work proposes to solve the YAP using a genetic algorithm based in restrictions for each container. Official data from Arica Seaport - Chile was used for defining an experimental scenario, which allowed obtaining promising results. The genetic algorithm represents a novel approach because allows getting solutions in a robust way, while satisfying the existing set of constrains associated with the general operation model.

Index Terms - Yard Allocation Problem, genetic algorithm, container

Índice

Índice de Figuras	II
-------------------	----

Índice de Tablas	III
------------------	-----

1. Introducción	1
1.1. Objetivo del Proyecto	2
1.1.1. Objetivo General	2
1.1.2. Objetivo Especifico	2
1.2. Plan de Trabajo y Metodología	2
1.2.1. Primera Etapa: Investigación y Propuesta de una Solución Formal al Problema	2
1.2.2. Segunda Etapa: Modelamiento del Problema y Primer Prototipo	2
1.2.3. Tercera Etapa: Modelado y Mejora de Solución con Nuevo Prototipo	3
1.2.4. Cuarta Etapa: Retoques, Resultados y Conclusiones Finales	3
1.3. Estructura del Documento	3
2. ¿Porque Resolver el Yard Allocation Problem mediante un Algoritmo Genético?	4
3. Algoritmos Genéticos	5
3.1. El Algoritmo de Holland	6
3.1.1. Codificación	7
3.1.2. Función de Adaptación	7
3.1.3. Selección	7
3.1.4. Cruzamiento	9
3.1.5. Mutación	9
3.1.6. Criterio de Término	9
3.1.7. Reemplazo	10
3.1.8. Operadores Genéticos	11
3.2. Modificaciones al Algoritmo de Holland	11
3.2.1. Selección	11
3.2.2. Cruzamiento	12
4. Asignación de Espacio en un Patio de Contenedores	15
4.1. Conceptos Básicos	15
4.1.1. Puerto Marítimo	15
4.1.2. Puertos en Chile	16
4.1.3. Terminal Portuario de Arica	16
4.1.4. Maquinaria	18
4.1.5. Patio de Contenedores	20
4.2. Descripción de las Operaciones dentro del Patio	20
4.2.1. Proceso de Importación	20
4.2.2. Proceso de Exportación	21

5. Literatura Previa	23
5.1. Algoritmo Genético para YAP: Enfoque en la Minimización de los Tiempos de Almacenaje y Balanceo de Cargas entre Bloques	23
5.1.1. Definición Formal	24
5.1.2. Algoritmo Genético	26
5.2. Algoritmo Genético para YAP: Enfoque en la Diversidad de Contenedores . . .	28
5.2.1. Definición Formal	28
5.2.2. Algoritmo Genético	30
5.3. Algoritmo Genético para YAP: Enfoque en el Espacio Asignado a Contenedores	31
5.3.1. Definición Formal	32
5.3.2. Algoritmo Genético	33
6. Propuesta de Solución	35
6.1. Generar un Balance entre las Cantidades de Contenedores por Bloque	35
6.2. Disminuir el Tiempo de Permanencia de una Embarcación en el Puerto	35
6.3. Planificar la Ubicación de los Contenedores Dentro de un Bloque	36
6.4. Definición Formal de la Solución	37
6.4.1. Supuestos y Definiciones	37
6.4.2. Estructura de los Parámetros de Entrada	38
6.4.3. Modelo Matemático	43
6.4.4. Algoritmo Genético	44
6.5. Sistema de Pruebas	46
6.5.1. Prueba 1, Evaluación de la Probabilidad de Cruzamiento	47
6.5.2. Prueba 2, Evaluación de la Probabilidad de Reemplazo	49
6.5.3. Prueba 3, Evaluación de la Probabilidad de Reemplazo al 20 %	51
6.5.4. Prueba 4, Evaluación de la Probabilidad de Reemplazo al 25 %	51
6.5.5. Prueba 5, Evaluación de la Probabilidad de Reemplazo al 30 %	52
6.5.6. Prueba 6, Evaluación de la Probabilidad de Reemplazo en los 3 Mejores	53
7. Conclusión	57
Referencias	58

Índice de Figuras

1.	Pseudocódigo del Algoritmo Genético Simple	6
2.	Roulette Wheel Selection	8
3.	Cruzamiento en un Punto	9
4.	Selección por Muestreo Estocástico Universal	11
5.	Selección por Torneo	12
6.	Cruzamiento en un Punto	13
7.	Cruzamiento Uniforme	13
8.	Típica Distribución de un Patio de Contenedores	15
9.	Terminal Portuario de Arica	16
10.	Operaciones Portuarias	17
11.	Quay Crane	18
12.	Rail Mounted Gantry Crane	18
13.	Rubber Tired Gantry Crane	19
14.	Reach Stacker	19
15.	Sistema BAROTI	20
16.	Proceso de Importación	21
17.	Proceso de Exportación	22
18.	Matriz Referencial de Soluciones	45
19.	Estructura Solución YAP	45
20.	Resultados Prueba 1, Probabilidad de Cruzamiento	48
21.	Resultados Prueba 2, Probabilidad de Reemplazo	50
22.	Resultados Prueba 3, Probabilidad de Reemplazo al 20 %	51
23.	Resultados Prueba 4, Probabilidad de Reemplazo al 25 %	52
24.	Resultados Prueba 5, Probabilidad de Reemplazo al 30 %	52
25.	Resultados Prueba 6, Probabilidad de Reemplazo al 20 %, 25 % y 30 %	54
26.	Patio de Contenedores Después de la Estiba	55
27.	Patio de Contenedores con Población Insertada	56

Índice de Tablas

1.	Ejemplo de Selección por Ruleta	8
2.	Lista de Bloques	39
3.	Planificación Naviera	39
4.	Distancia de Bloques a Sitios	40
5.	Contenedores Iniciales	40
6.	Tiempos de Ejecución	55
7.	Segmentación por Niveles	56

1. Introducción

La globalización ha influido fuertemente en el aumento del comercio internacional. Lo anterior incide de manera directa en los puertos marítimos alrededor del mundo, dado que deben responder de manera ágil a las exigencias del mercado. En Chile, dada su característica de poseer más de 9.000 kilómetros de costa y estar en una ardua búsqueda hacia el desarrollo económico, se ha logrado concretar en la última década importantes tratados de libre comercio, permitiendo que el sector portuario diera un salto considerable respecto a la modernización de nuevas tecnologías que coordinan a la par su trabajo junto con maquinarias de última generación, lo que abre paso a la optimización y eficiencia en los tiempos de carga y descarga.

Al mismo tiempo, esta característica también trae consigo algunos aspectos negativos. Chile, al poseer un extenso borde costero, se permite la existencia de actualmente 56 puertos asociados a importantes ciudades del país. Las ciudades que tienen puertos marítimos tienden a basar su desarrollo alrededor de las operaciones portuarias, trayendo consigo consecuencias y restricciones al desarrollo urbano. La expansión de las operaciones portuarias es una dificultad para la mayoría de las ciudades puerto, dada la imposibilidad de extender su terreno a realizar sus operaciones, debiendo ajustarse al espacio que disponen y administrar el aumento de las operaciones de transferencia de carga. Extender las explanadas de puerto desde el océano hasta el continente atenta con la estructura de las ciudades, y adicionalmente, extender las explanadas hacia el océano posee un elevado costo monetario.

Dentro de las actividades principales de un puerto están: por un lado, minimizar los tiempos en que los puertos atracan en un muelle, dado que el tiempo de atraque tiene un costo para el puerto (no puede recibir otros barcos mientras no despeje un muelle de atraque), y para las compañías navieras (cada cierto tiempo extra que el barco continúe en puerto, se le cobrará una tarifa); y por otro lado, minimizar los costos operativos en relación al movimiento de contenedores ya que el innecesario movimiento de ellos genera un costo para las máquinas en la acción de mover un contenedor, por lo que hay que mover el mínimo de contenedores posibles.

Las novedades del presente trabajo son: 1) definir un algoritmo genético en base a la explotación para resolver el Yard Allocation Problem, 2) definir un escenario de test con datos reales para el Yard Allocation Problem, 3) analizar los resultados experimentales obtenidos mediante la aplicación del algoritmo genético en el escenario de test definido.

1.1. Objetivo del Proyecto

1.1.1. Objetivo General

- Desarrollo de la problemática sobre una eficiente asignación del espacio para contenedores a través de un algoritmo genético con datos del Terminal Portuario de Arica.

1.1.2. Objetivo Especifico

- Entender a plenitud la problemática a solucionar.
- Investigar y comprender el comportamiento de los parámetros de los algoritmos genéticos y su funcionamiento.
- Una vez comprendido el problema, recabar la información y proponer un prototipo para acercarse lo más posible a un algoritmo genético.
- Refinar el prototipo generado para obtener la primera versión de algoritmo genético.
- Obtener resultados mediante de los parámetros del algoritmo genético (probabilidades).

1.2. Plan de Trabajo y Metodología

Es importante esclarecer que esta documentación no está basada en un sistema de información, por lo que no se empleará una metodología tradicional para el desarrollo de un software operativo. Pero al tratarse de un problema en lo que se busca es la optimización, lo más relevante serán los resultados que entregue, no el software en si. No obstante a medida que avanza el documento, se irán desarrollando distintos algoritmos que irán hacia un mismo fin, regidos bajo una metodología incremental, complementando con otros algoritmos simultáneamente.

Este trabajo se irán subdividiendo en etapas acorde a plazos fijados que se irán explayando un poco más exhaustivamente.

1.2.1. Primera Etapa: Investigación y Propuesta de una Solución Formal al Problema

Buscar, investigar y comprender la más completa información sobre lo que abarca el problema, además del estudio sobre algoritmos genéticos, afectividad y sus aplicaciones en una lucha por enunciar una solución tentativa y acorde a lo que el problema necesita.

Duración: 2 meses (Marzo - Abril, 2015).

1.2.2. Segunda Etapa: Modelamiento del Problema y Primer Prototipo

Examinar la mayor cantidad de información real sobre el problema llegando a obtener un set considerable de datos del Terminal Portuario de Arica, lo que a priori servirá para generar un primer prototipo. Lo que finalmente se convierte en el primer acercamiento al YAP en este proyecto, como fue el algoritmo constructivo.

Duración: 2 meses (Mayo - Junio, 2015).

1.2.3. Tercera Etapa: Modelado y Mejora de Solución con Nuevo Prototipo

Mejorar el prototipo de algoritmo constructivo dejando de lado los óptimos locales y generar el primer acercamiento de algoritmo genético con las reglas del negocio de TPA sobre ubicación de contenedores y acercarse a un óptimo global.

Duración: 2 meses (Septiembre - Octubre, 2015).

1.2.4. Cuarta Etapa: Retoques, Resultados y Conclusiones Finales

Perfeccionar y mejorar los detalles del prototipo de algoritmo genético para obtener resultados y compararlos con algoritmo Greedy para generación de población.

Duración: 2 meses (Noviembre - Diciembre, 2015).

1.3. Estructura del Documento

En el capítulo uno se explica una breve introducción al problema junto con los objetivos tanto general como específicos, además del plan de trabajo y la metodología que se empleará.

En el capítulo dos se expone porque este problema común dentro de un puerto debe ser resuelto a través de algoritmos genéticos.

En el capítulo tres se hace una breve historia y desarrollo de los algoritmos genéticos con el paso de los años, además de sus métodos más utilizados y ventajas en relación a otros métodos de búsqueda y optimización.

En el capítulo cuatro aclara conceptos básicos dentro de un puerto, como son la maquinaria principal que se utiliza, sus procesos y métodos para el manejo de contenedores.

En el capítulo cinco se hace explicación de trabajos anteriores de resolución al YAP con algoritmo genético y en su conjunto generarán las ideas para la propuesta de solución

En el capítulo seis se propone y modela la solución para el Yard Allocation Problem y sus resultados finales.

2. ¿Porque Resolver el Yard Allocation Problem mediante un Algoritmo Genético?

El *Yard Allocation Problem (YAP)* es un problema de decisión para la asignación de recursos. Considerado como un problema de categoría NP-hard (problemas computacionales que pueden ser resueltos en tiempo polinómico), por lo que es factible el uso de algún tipo de heurística que pueda resolver el problema. En el universo de soluciones a este problema, se ha resuelto con distintas clases de heurísticas, como sistemas multi-agente, Búsqueda tabú, cálculo humano, por nombrar algunos. Pero hay una heurística no muy usada y que esta tomando mucha fuerza como son los algoritmos genéticos.

Algoritmos genéticos son una nueva tendencia dentro de las heurísticas, que trabaja intrínsecamente en paralelo (opera con varias soluciones en paralelo) dándole una gran diferencia con otras heurísticas, se ven poco afectadas ante los “óptimos locales” (soluciones falsas). Todo esto implementado para resolver distintos problemas de búsqueda y optimización. Pero siendo aún más específicos, dentro de los problemas de un puerto los algoritmos genéticos han abarcado resoluciones a problemas como *Berth Allocation Problem* (distribución óptima de barcos a muelle de atraque) y el *Truck Allocation Problem* y no al *Yard Allocation Problem*. Es por esto que nació la propuesta de usar este tipo de heurística.

3. Algoritmos Genéticos

El algoritmo genético es un conocido método de búsqueda y optimización que se remonta a ideas sobre la teoría de la evolución de Charles Darwin (1859) en los organismos vivos. En la naturaleza, las poblaciones van adaptándose a las exigencias del entorno haciendo que unos individuos sean más adaptables que otros generando una diferenciación entre los más fuertes y los más débiles, a esto se conoce como selección natural. Los individuos luchan constantemente por recursos naturales que por lo general son escasos. Por lo cual hay una lucha constante entre los individuos que puedan adaptarse mejor al entorno que les rodea. Esta adaptación esta llevada a cabo por características propias del individuo que le fueron heredadas por sus antecesores, estas características esta controlada por un gen que el conjunto de estos forman al cromosoma, los cuales son fundamentales a la hora de la competitividad.

Además de luchar por los recursos, también compiten por la búsqueda de una pareja con la cual reproducirse a lo que llamaremos al proceso de selección. Los seres que tengan más probabilidades de adaptarse, también tendrán más posibilidad de crear un gran número de descendientes. Por el contrario, los individuos que no tengan las mismas oportunidades de adaptación van a generar una baja cantidad de individuos. Aquí es donde se genera la evolución con el intercambio de material genético de los progenitores. Este intercambio es en si entre los cromosomas se llama cruza o crossover, creando una posibilidad de que la combinación haya sido correcta entre genes para crear mejores individuos. La selección y cruza iterativamente genera en la evolución continua del pozo de genes y la creación de individuos que se irán adaptando de mejor manera al ambiente que sus predecesores.

Ya desde los años 50' que estaban comenzando las primeras ideas de la simulación de la evolución en un ordenador por parte de brillantes científicos del área como Turing o Bremermann. Pero no fue hasta las últimas tres décadas, en que el interés por este método de búsqueda y optimización había llamado notoriamente la atención cuando un profesor de la Universidad de Michigan, John Henry Holland [Holland, 1975] propuso las reglas para la creación de un código genético que pudiese representar la estructura de cualquier programa computacional.

Holland, siempre pensante sobre la versatilidad que tienen los organismos vivos para resolver problemas de mejor manera que el mejor programa informático lo hicieron ver más a fondo a la selección natural. La selección natural simplifica enormemente al diseño de software porque se especifica de antemano todas las características del problema y del programa que debe tratar con ellas. Los algoritmos genéticos permiten abarcar más a fondo los problemas reales como lo haría un programa convencional. La mayoría de los organismos al momento de la reproducción, este se divide en las etapas de selección y cruzamiento, pero para el algoritmo genético esto no es lo más complejo. La parte difícil corresponde a la hora de generar el "código genético" que es la base para el algoritmo como el ADN para el organismo.

Todo lo antes mencionado sirvió para crear el algoritmo genético pionero en el área codifica las posibles soluciones en cadenas de bits en el alfabeto binario. Al igual que en la naturaleza, el proceso de selección permitirá llegar a las mejores soluciones y como es de costumbre, serán las soluciones que sobrevivan, cada posible solución posee un valor de fitness o amplitud

que muestra matemáticamente que tan óptima es para poder diferenciarse con el resto de la población y a la vez le dará más posibilidades de reproducirse y más "hijos". El intercambio de material genético es simulada a través del método de cruzamiento que intercambia porciones entre cromosomas. Otro método, la mutación causa alteraciones en los bits de cromosomas con la intención de generar diversidad en la población.

3.1. El Algoritmo de Holland

Ahora ya conocido como Algoritmo Genético Simple o SGA (en inglés, Simple Genetic Algorithm) se basa en la simplicidad de las operaciones y gran calidad de resolución. Comienza con la generación de una población inicial que se creará aleatoriamente y codificado en binario (unos y ceros), siendo cada uno de los individuos una posible solución al problema que se este planteando. Estas soluciones se evaluarán a través de una función de adaptación que mostrará la calidad misma de la solución. Una vez creados los individuos, se procede a seleccionar a los padres al azar aplicando un método que favorezca a los predecesores mejor adaptados, ya que a cada individuo le corresponde una probabilidad de ser seleccionado proporcional a su función de fitness. Posterior a esto se usan operadores genéticos como el cruzamiento y la mutación creando una nueva generación de individuos. Este proceso es iterativo hasta que un criterio de termino se cumpla [Sivanandam and Deepa, 2007].

En la Figura 1 se presenta la estructura básica del algoritmo genético clásico, el que resume su funcionamiento, el cual contiene los siguientes componentes: una población de cadenas binarias, parámetros de control, una función de aptitud, operadores genéticos, un mecanismo de selección y un mecanismo para codificar las soluciones como cadenas binarias.

```

BEGIN /* Algoritmo Genetico Simple */
  Generar una poblacion inicial.
  Computar la funcion de evaluacion de cada individuo.
  WHILE NOT Terminado DO
    BEGIN /* Producir nueva generacion */
      FOR Tamaño poblacion/2 DO
        BEGIN /*Ciclo Reproductivo */
          Seleccionar dos individuos de la anterior generacion,
          para el cruce (probabilidad de seleccion proporcional
          a la funcion de evaluacion del individuo).
          Cruzar con cierta probabilidad los dos
          individuos obteniendo dos descendientes.
          Mutar los dos descendientes con cierta probabilidad.
          Computar la funcion de evaluacion de los dos
          descendientes mutados.
          Insertar los dos descendientes mutados en la nueva generacion.
        END
      IF la poblacion ha convergido THEN
        Terminado := TRUE
      END
    END
  END

```

Figura 1: Pseudocódigo del Algoritmo Genético Simple. *Fuente: www.datateca.unad.edu.co*

3.1.1. Codificación

Asumimos que los individuos (o cromosomas), probables soluciones al problema específico pueden representarse como una agrupación de parámetros (o genes) que conforman una cantidad considerable de valores. Si damos por hecho que para representar individuos no necesariamente debe estar en base a 0 y 1, gran parte de la teoría en que están constituido los algoritmos genéticos usan este alfabeto. Estos individuos pueden ser considerados a dos niveles, fenotipo que son los valores que toman los parámetros a estimar en el dominio en el que se define la función objetivo, y genotipo corresponde a la representación del fenotipo a bajo nivel, es decir, la codificación del genotipo, generalmente una cadena de bits.

3.1.2. Función de Adaptación

Cada problema debe tener diseñada su función de adaptación de forma específica que evalúe cada individuo. Dado un cromosoma en específico, la función de adaptación le asignará un valor real, que presume indicar la calidad de la solución o grado de adaptabilidad del individuo al entorno para posteriormente ser usado por el mecanismo de selección para evaluar las cadenas de la población.

3.1.3. Selección

Holland dentro de los métodos para utilizar en el SGA para la selección proporcional (individuos en función de amplitud con respecto al total de la población) se decidió por el método de la ruleta o RWS (en inglés, Roulette Wheel Selection) que trabaja de la mano con la probabilidad de que sea seleccionado un individuo, ya que es proporcional a su función de adaptación.

1. Evaluamos la función de adaptación, f_i , para cada individuo en la población.
2. Procedemos a calcular la probabilidad de selección, p_i , para cada individuo de la población: $p_i = f_i / \sum_{j=1}^n f_j$, donde n es el tamaño de la población.
3. Calculamos la probabilidad acumulada, q_i , para cada individuo: $q_i = \sum_{j=1}^i p_j$.
4. Generamos un número random uniforme, $r \in (0, 1]$.
5. Si $r < q_1$, entonces se seleccionará el primer cromosoma, x_1 , de lo contrario se escogerá un x_i que cumpla: $q_{i-1} < r \leq q_i$.
6. Repetir los pasos 4-5 n veces hasta generar los n candidatos.

Cadena	Población Inicial	Base ₁₀	Aptitud, f	Probabilidad, p_i	Probabilidad Acumulada, q_i
1	01101	13	169	0,14	0,14
2	11000	24	576	0,49	0,63
3	01000	8	64	0,06	0,69
4	10011	19	361	0,31	1,00
Suma			1170	1,00	
Promedio			293	0,25	
Máximo			576	0,49	

Tabla 1: Ejemplo de Selección por Ruleta con $f(i) = i^2$. Fuente: www.edc.ncl.ac.uk

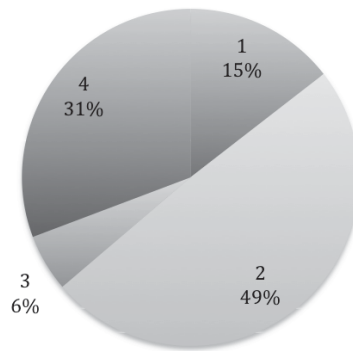


Figura 2: Roulette Wheel Selection. Fuente: *Elaboración Propia*

Ahora, si se generara un número random $r = 0,66$, entonces el tercer cromosoma sería seleccionado ya que $q_2 = 0,63 < 0,66 \leq q_3 = 0,69$.

Cada cadena con un valor de aptitud más alto que el promedio se le asigna más de una descendencia, mientras que a una cadena con un valor más bajo que el promedio se le asigna menos de una descendencia, esta es asignada a un sector de la ruleta con un ángulo subtendido por el sector en el centro de la rueda igual a 2π (ver Figura 2). A una cadena se le asigna una descendencia si el número generado al azar en el intervalo $[0, 2\pi]$ cae en un sector correspondiente a la cadena.

Hay inconvenientes cuando hay una pequeña población (en el peor de los casos $N = 1$) que posee un fitness excesivamente superior en comparación al resto de cromosomas, esto pierde diversidad en la población creando una convergencia prematura no representando a la población total, pero si a un grupo específico de la población como un óptimo local. En esos casos es mejor usar selección basada en ranking o por torneo.

Una vez que ya se ha seleccionado se realiza un emparejamiento de manera aleatoria para continuar con el método de cruzamiento.

3.1.4. Cruzamiento

El cruzamiento busca emular la herencia genética de cualidades de los progenitores a los hijos. Se realiza según la representación elegida para los individuos. De seleccionar la representación binaria, lo más apropiado (y lo que se usa en SGA) es hacer el cruzamiento entre dos individuos por el cruce en un punto: una vez que se decide cruzar dos individuos, se genera un número aleatoriamente uniforme r , y si $r \leq p_c$, los dos individuos se cruzarán. En caso contrario, es decir $r > p_c$, los dos hijos no serán más que unas copias a sus padres. El valor de p_c puede ser de forma experimental o puede ser en base a los principios de los teoremas de esquemas [Goldberg, 1989]. Luego se cambian las subcadenas a partir del punto de cruce (ver Figura 3).

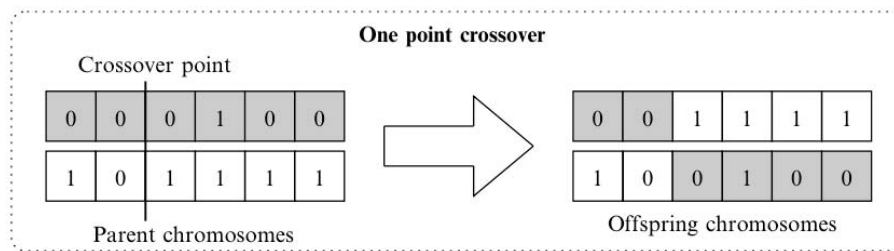


Figura 3: Cruzamiento en un Punto. Fuente: [Burke and Kendall, 2014]

Por ejemplo, si los individuos se representan por vectores binarios de longitud cinco, el cruce de los padres (1010001110) y (0011010010) a partir del cuarto gen dará como resultado los hijos (1010010010) y (0011001110). Tras el cruce los hijos resultantes sustituyen a los padres en la nueva población, se trata por lo tanto de un método de remplazamiento generacional.

3.1.5. Mutación

Se encarga de cambiar algunas características de los individuos de manera individual para introducir diversidad a la población. Consiste en invertir un bit (cuando la representación es binaria), es decir, cambia un 1 por un 0 y viceversa. Al igual que en cruzamiento, en mutación ocurre lo mismo en relación a la probabilidad de que un bit sea invertido, denominada por p_m .

Si bien en un principio se podría pensar que el operador de cruce es más relevante que el operador de mutación, ya que ofrece una exploración rápida del espacio de búsqueda, éste último asegura que ningún punto del espacio de búsqueda tenga probabilidad cero de ser examinado y la opción de recuperar material genético perdido (por ejemplo que todas las cadenas en una población convergen a cero en cierta posición y que la solución óptima tiene un 1 en esa posición), siendo de vital importancia para asegurar la convergencia de los AG.

3.1.6. Criterio de Término

Terminando con el SGA, tenemos que recordar que previo a la ejecución del algoritmo, este criterio tiene que estar definido. Se basa en que una vez finalizada una generación, se debe estudiar si el proceso de búsqueda debe ser detenido o no. Existen cuatro criterios que se utilizan

típicamente para finalizar el proceso de búsqueda y reconocer a una solución como buena. Son los siguientes:

1. Finalizar al momento en que se encuentra un individuo sobre un valor de aptitud suficientemente bueno.
2. También la ejecución podría ser terminada cuando la población alcance cierto nivel de homogeneidad, es decir gran parte de las cadenas tiene la mayoría de los bits idénticos.
3. Terminado una vez que hayan pasado cierta cantidad de generaciones.
4. Haber alcanzado un determinado tiempo de computación;

Se deben usar criterios de términos porque al ser una heurística, el objetivo de los AG no es encontrar óptimos, sino encontrar una buena solución en un tiempo considerable. En la situación de ser posible encontrar una solución óptima, en la mayoría de los casos no se puede detectar si esta se encontró.

3.1.7. Reemplazo

Una vez que las nuevas soluciones descendientes se han creado mediante el cruce y la mutación, hay que introducirlos en la población de los padres. Hay muchas maneras de abordar el tema. Hay que tener en cuenta que los individuos de los padres ya han sido seleccionados de acuerdo a su “condición física”, por lo que se espera que los niños se encuentren entre los más fuertes dentro de la población y que con el paso de las generaciones, la población en general también mejore su “estado físico”. Algunas técnicas más comunes son:

- Eliminarlos a todos: Esta técnica elimina a todos los miembros de la población actual y los reemplaza con el mismo número de cromosomas que acaban de ser creados. Esta es probablemente la técnica más común y será la técnica de elección para la mayoría de la gente debido a su relativa facilidad de aplicación.
- El estado de equilibrio: Elimina los n antiguos miembros y los reemplaza con n nuevos miembros. El número para eliminar y reemplazar, n , en un momento dado es un parámetro para esta técnica. Otra consideración es decidir que miembros borrar de la población actual.
- No hay duplicados en estado estacionario: El algoritmo comprueba que no se hayan añadido cromosomas duplicados para la población. Esto se suma a la sobrecarga de cálculo pero puede significar que más del espacio de búsqueda se explora.
- Eliminar a los peores padres: Se comprueba que al final de la generación, los nuevos hijos son mejores que los padres, de cumplirse esta condición, los nuevos hijos reemplazarán a los padres.
- Probabilidad de reemplazo: Como una forma de evitar que la población se pudiera volver elitista, existe una probabilidad que le da la posibilidad a los hijos con peor fitness que los padres tengan alguna oportunidad de ser beneficiados de pasar a la siguiente generación.

3.1.8. Operadores Genéticos

Cada vez que es ejecutado un AG, corresponde decidir con que clase de frecuencia se va a aplicar cada uno de los AG, en ciertas ocasiones como ocurre en mutación y cruzamiento, se debe añadir un parámetro adicional que indicará la frecuencia que se va a aplicar dentro de cada gen del cromosoma. Esta frecuencia para cada operador va a estar en relación al problema.

En general, la frecuencia de estos operadores no cambia al momento de la ejecución del algoritmo, pero hay que explicar que cada operador es más eficiente en un tiempo exacto de la ejecución, es decir, en un comienzo en la fase de exploración, los más activos son la mutación y el cruzamiento. Luego en el momento que la población casi ha convergido, el cruzamiento no es útil, porque al trabajar con individuos casi similares, la información que se estaría intercambiando (y relevante) sería muy poca. Por el contrario, si se produce un lentitud en el algoritmo la mutación tampoco sería útil porque estaría bajando a una búsqueda aleatoria.

3.2. Modificaciones al Algoritmo de Holland

En las últimas décadas, se ha investigado como mejorar el rendimiento de los algoritmos genéticos. Se han propuesto implementaciones más eficientes tales como:

3.2.1. Selección

Muestreo Estocástico Universal

El SUS (en inglés, Stochastic Universal Sampling) [Baker, 1987] es una modificación a la selección de la ruleta para disminuir la propagación. Los individuos se asignan a un segmento continuo con un espacio similar a su función de adaptación, F . La diferencia con la ruleta es que ahora se usarán punteros equidistantes que van sobre los individuos. Considerar N punteros que será el número de individuos seleccionados. La distancia entre los punteros será de F/N . La posición del primer puntero será un número aleatorio en el rango $[0, F/N]$. Este método es bastante útil ya que los individuos con menor adaptación tienen más oportunidad de ser seleccionados asegurando una mejor descendencia que en el método de ruleta.

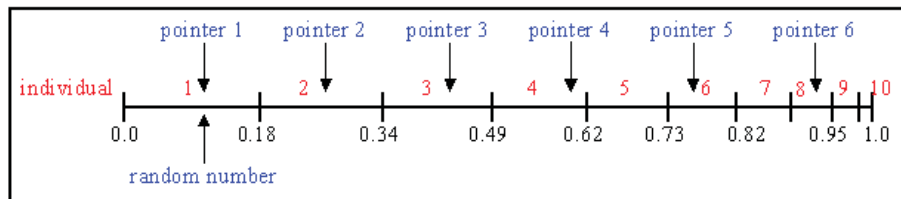


Figura 4: Ilustración de una selección por muestreo estocástico universal. Fuente: pg.gda.pl

En la Figura 4 se considera un $F = 1$, los seis punteros ($N = 6$) equidistantes a $1/6 = 0,167$ y el primer puntero fue colocado aleatoriamente $[0, 0,167]$ resultando 0,1. Entonces los individuos seleccionados fueron 1,2,3,4,6,8.

Por Torneo

Implementada por Goldberg, s cromosomas son elegidos al azar y entran a un torneo de uno contra uno. El individuo más apto en el grupo de cromosomas s gana el torneo y es seleccionado como padre (ver Figura 5). El valor más utilizado de s es 2.

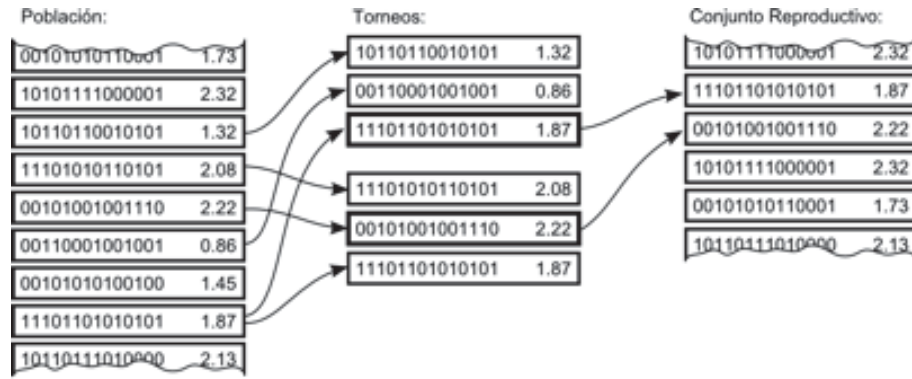


Figura 5: Ilustración de una selección por torneo. *Fuente: Wikipedia (Autor: Lukipuk)*

Este método es variado dependiendo de lo que se busque en el problema a desarrollar, porque cuando existen muchos individuos en el torneo, los peores individuos tendrán una baja oportunidad de ser seleccionados y se estaría centrando en una búsqueda de las mejores soluciones actuales en el entorno próximo. Pasa lo contrario cuando la población del torneo es baja, dándole más oportunidad a los peores individuos y permitiendo una exploración de nuevas regiones en el espacio de búsqueda.

Por Truncamiento

Están ordenadas por la función de adaptación y una proporción p ($1/2$, $1/3$, etc) se seleccionan los individuos más aptos y serán reproducidos $1/p$ veces. Ejemplo de una población de 100 individuos y se selecciona la mitad de ellos, todos los sobre o igual a $1/2$ serán seleccionados, dándole cero oportunidad a los con menor fitness de haberse convertido quizás en algo mejor. Es poco sofisticado que otros procesos de selección y en la práctica no se usa con mucha frecuencia.

3.2.2. Cruzamiento

De Dos Puntos

Tendrá dos puntos seleccionados al azar para escoger una cadena del individuo. Todo lo que se encuentra entre los dos puntos procede a intercambiarse. Este método es ideal en conjunto con el de un punto cuando se va a trabajar con poblaciones grandes, ya que permite la preservación de los esquemas (ver Figura 6).

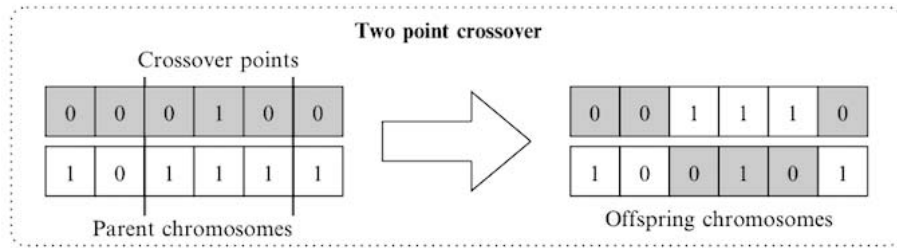


Figura 6: Cruzamiento en un Punto. *Fuente: [Burke and Kendall, 2014]*

Uniforme

Hay un intercambio entre trozos al azar del cromosoma de los individuos seleccionados, con cierta probabilidad p_e , llamada probabilidad de intercambio. Por lo general, esta probabilidad toma un valor de 0,5. Esto permite que los padres contribuyan al nivel de genes en vez de al nivel de segmento. Así con una probabilidad de 0,5 los hijos tendrán un 50% del primer padre y un 50% del segundo padre. Este método es recomendable cuando se está trabajando con poblaciones pequeñas (ver Figura 7).

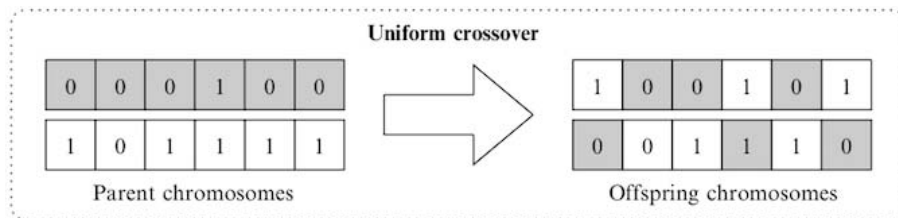


Figura 7: Cruzamiento Uniforme. *Fuente: [Burke and Kendall, 2014]*

Partially Mapped Crossover (PMX)

Esta técnica de cruzamiento fue propuesta para resolver el problema del vendedor viajero (TPS). Se hacen algunos ajustes para acomodar el genotipo. El PMX genera una descendencia tomando una sub secuencia de una asignación de AL proveniente de un padre y preservando el orden y posición de tantas AL como sea posible del otro padre [Sivanandam and Deepa, 2007]. La sub secuencia se determina al escoger dos puntos de corte aleatorios. Por ejemplo, considere estos padres:

$$p_1 = 012|3456|789$$

$$p_2 = 312|5740|968$$

Estos generan las siguientes descendencias. Primero, los segmentos entre los puntos de corte se intercambian:

$$o_1 = ---|5740|---$$

$$o_2 = ---|3456|---$$

Este intercambio define una serie de mapeos:

$$3 \leftrightarrow 5,$$

$$4 \leftrightarrow 7,$$

$$5 \leftrightarrow 4,$$

$$6 \leftrightarrow 0.$$

El resto se rellena con los AL originales de los padres, los que no tienen conflicto:

$$o_1 = -12|5740| - 89$$

$$o_2 = -12|3456|9 - 8$$

Finalmente, la primera posición en o_1 (debería ser 0, pero causa conflicto, ya que el 0 ya se encuentra presente en la posición 7) es remplazada por el 6 a causa del mapeo $6 \leftrightarrow 0$. Estos reemplazos son transitivos hasta que el conflicto se resuelva. Por ejemplo, la segunda posición de o_1 debería seguir los mapeos $7 \leftrightarrow 4$, $4 \leftrightarrow 5$ y $5 \leftrightarrow 3$, por lo que se reemplaza por el 3. Las descendencias finales son:

$$o_1 = 612|5740|389$$

$$o_2 = 712|3456|908$$

4. Asignación de Espacio en un Patio de Contenedores

Existen diversos aspectos que son deseables en los procesos operativos de un terminal portuario, específicamente en el patio de contenedores (el cual sirve de buffer para las operaciones de importación y exportación). La mayoría proviene de elementos directamente relacionados con el patio, pero no son propios de este. En este caso, se tomarán algunos aspectos de ambos, considerando asuntos exclusivamente de patios de contenedores y algunos aspectos que estén directamente relacionados con estos. Qué elementos se tomarán y cuáles no, serán descritos en el capítulo 6. En este capítulo se describen algunos procesos relacionados con el patio de contenedores, principalmente los de exportación e importación. Esto con el objetivo de contextualizar el problema y permitir una mayor comprensión del escenario formal a tomar en cuenta en la solución.

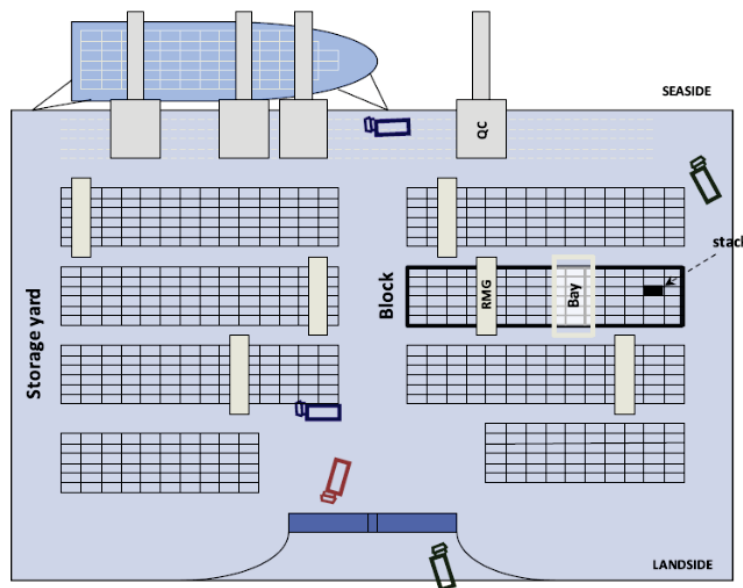


Figura 8: Típica Distribución de un Patio de Contenedores. *Fuente: www.e-archivo.uc3m.es*

4.1. Conceptos Básicos

Para introducir al lector al ámbito del proyecto, se presentan a continuación los conceptos básicos de las operaciones en un terminal de contenedores.

4.1.1. Puerto Marítimo

Físicamente, un puerto es una ubicación en la costa con la infraestructura necesaria para que las naves puedan atracar para la transferencia de personas y principalmente para el transporte de carga. Funcionalmente, son interfaces entre distintos modos de transporte. Son sistemas multifuncionales y hoy en día son verdaderos puntos de distribución de mercancías. Juegan un papel importante en el comercio, dado que gran parte del transporte de mercancías se hace a través del medio marítimo.

4.1.2. Puertos en Chile

Por sus llamativas características geográficas, Chile posee un borde costero muy extenso. En general nuestra costa chilena se caracteriza por ser abierta, desabrigada y de poca profundidad, esto abrió paso al desarrollo de la infraestructura necesaria para permitir el transporte de todo tipo de carga, por lo que en la actualidad existen 56 puertos a lo largo del país. Cada zona del país se caracteriza por ciertos tipos de productos, así mismo los puertos de estas zonas sirven para estos mismos propósitos. Para la zona norte los puertos son usados principalmente para el transporte de minerales dada la gran cantidad de extracción de mineral. En la zona central, se usan en general para productos agrícolas e importación de carga de valor agregado dado que se concentra la mayoría de la población. Más al sur están los puertos dedicados al transporte de productos derivados de la madera [Gobierno de Chile, 2009].

4.1.3. Terminal Portuario de Arica

Ubicado en la ciudad de Arica, XV Región de Arica y Parinacota, el Terminal Portuario de Arica (TPA) es uno de los principales terminales portuarios del norte de Chile, la Empresa Portuaria Arica ha servido tanto para el tránsito de mercaderías tanto de importación como exportación para los territorios de Chile y Bolivia [Arica, 2015].

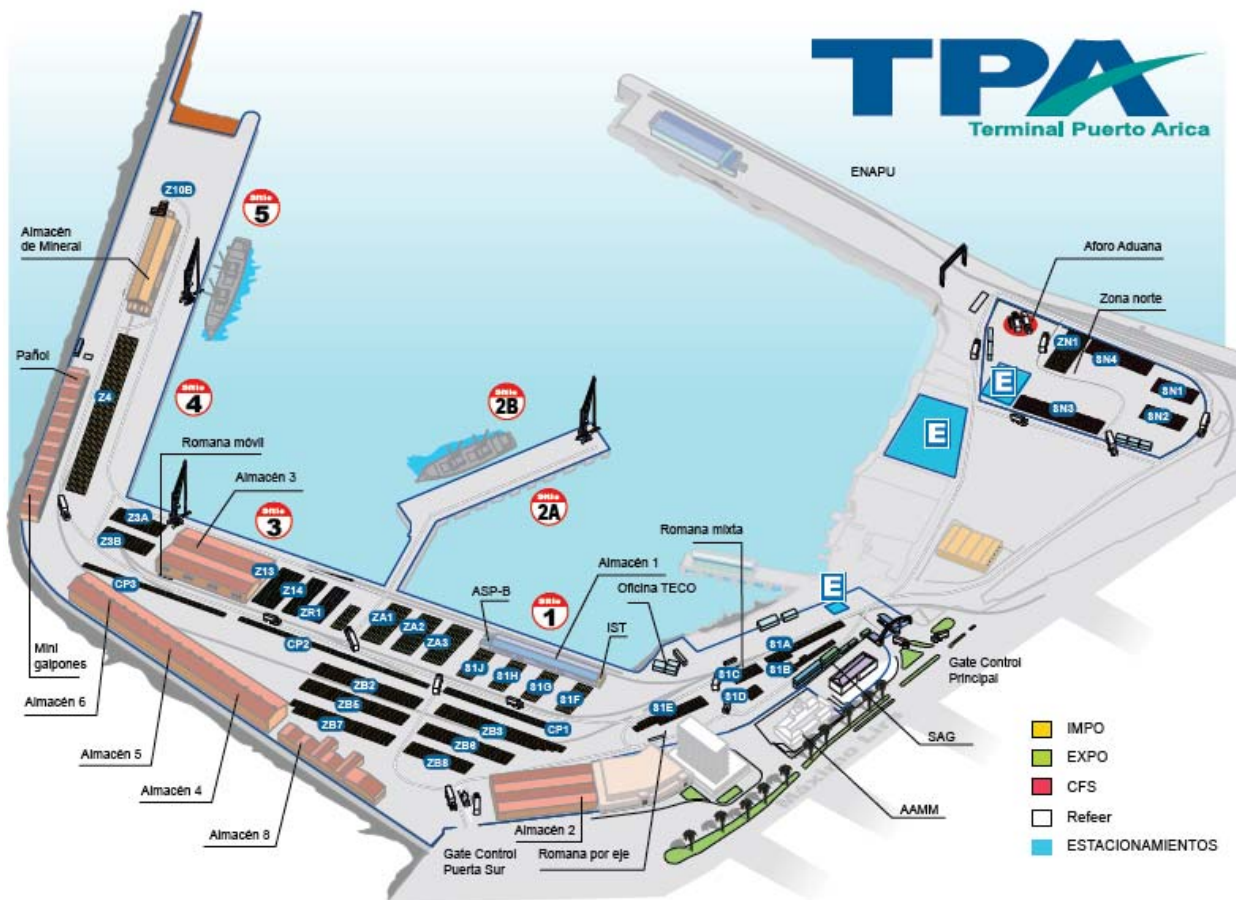


Figura 9: Terminal Portuario de Arica (TPA). Fuente: www.tpa.cl

Tiempo estimado de estadía de los contenedores

TPA posee un tiempo de permanencia mayor a diferencia de otros puertos en Chile, debido principalmente a las condiciones firmados en el Tratado de Paz y Amistad entre Chile y Bolivia. Los contenedores bolivianos poseen límites más extendidos de permanencia de la carga sin costo, registrándose tiempos promedios de 8,5 días durante el año 2011 [Maturana, 2014].

Operaciones Portuarias

Dentro de las operaciones principales que abarca el manejo de contenedores en un terminal portuario, las más importantes pueden resumirse en la Figura 10 como:

1. Atrake: Corresponde a la actividad que se realiza entre el mar y el terminal, conformado principalmente de los siguientes procesos: Llegada de las naves, asignación de un sector del puerto para el atraque de la nave y selección de la próxima nave que atracará.
2. Carga-Descarga: Se encarga de seleccionar el tipo y número de grúas de muelle que son necesarias para llevar a cabo la tarea de carga y descarga de los contenedores de la nave, de una forma eficiente y eficaz trabajando cuidadosamente con la estiba correspondiente a la nave. Los punto 1 y 2 son un problema de decisión para resolver el *Berth Allocation Problem*.
3. Recepción y Entrega Terrestre: Es la llegada de los camiones y el intervalo de tiempo, ya que los terminales portuarios cuentan con periodos determinados para la recepción de camiones y entrega correspondiente de contenedores. Este problema es conocido como el *Truck Allocation Problem*.
4. Almacenamiento: Trabaja con el punto 2 de Carga-Descarga y el punto 3 de Recepción, el objetivo principal es la manipulación y almacenamiento de contenedores. Como el espacio dentro de un terminal portuario es limitado, es necesario que se desarrollen políticas que permitan el optimo aprovechamiento del espacio y del tiempo. Este punto es conocido como el problema de asignación de patio o *Yard Allocation Problem*.

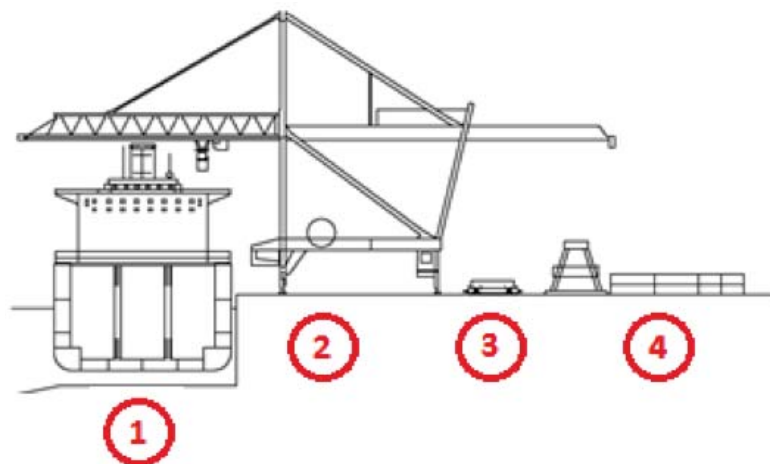


Figura 10: Operaciones Portuarias. Fuente: [Maturana, 2014]

4.1.4. Maquinaria

A continuación se describen algunas de las máquinas necesarias para las actividades en un terminal de contenedores. Fundamentales para las faenas de carga y descarga, las grúas son máquinas que tienen como propósito levantar una carga, ya sea para cambiarla de posición dentro de un espacio, o dejarla en otra máquina para que esta la transporte. Otro equipamiento importante son los tracto camiones los cuales se utilizan para el tránsito de contenedores dentro del puerto como para el transporte hacia el exterior. Para el caso de las grúas, en un patio de contenedores se utilizan de distintos tipos según sus funciones, características, y capacidades.

Las grúas de muelle o Quay Crane se encuentran en el lado del muelle del terminal portuario y se encarga de la carga y descarga de contenedores de los barcos atracados en el muelle (ver Figura 11).



Figura 11: Quay Crane. *Fuente: portstrategy.com*

Las grúas de patio tienen el propósito de ubicar y extraer los contenedores dentro del patio de contenedores. Existen de distintos tipos: las Rail Mounted Gantry Crane (ver Figura 12) y Rubber Tired Gantry Crane (ver Figura 13), ambas del tipo pórtico, difiriendo de su plataforma de movimiento, además están las Reachstacker y las Toplifter.



Figura 12: Rail Mounted Gantry Crane. *Fuente: portstrategy.com*

Las Rail Mounted (Montada sobre rieles), como su nombre indica, se desplazan sobre rieles. Su movimiento está limitado a un eje que se extiende a lo largo del riel. Se utilizan específicamente para el manejo de contenedores de patio, utilizan energía eléctrica, por lo que su operación es más limpia. La desventaja es el costo de la infraestructura necesaria. Las Rubber Tired, en cambio, como su nombre lo indica, se movilizan a través de ruedas, son más lentas que las Rail Mounted, pero más contaminantes y ruidosas.



Figura 13: Rubber Tired Gantry Crane. *Fuente: portstrategy.com*

Las Reachstacker (ver Figura 14) son grúas pequeñas más frecuentes en puertos que tratan menores volúmenes de contenedores, son del tipo tractor. Estas son capaces de apilar hasta aproximadamente 3 contenedores. Las Toplifter son del mismo tipo con la diferencia de que se usan para apilar contenedores vacíos, por lo que pueden apilar hasta un aproximado de 7 contenedores. El puerto a considerar usa solo grúas RMG para el patio.



Figura 14: Reach Stacker. *Fuente: portstrategy.com*

4.1.5. Patio de Contenedores

En general el patio de contenedores de un puerto se encuentra dividido en yardas destinadas para almacenamiento de estos. Dentro de cada yarda se utiliza el mismo sistema que se usa en la estiba de barcos: el sistema BAROTI, el cual se explica en [Kaps, 2012]. Para el caso del patio de contenedores, este sistema usa un sistema de coordenadas de 3 ejes, como lo ilustra la Figura 15, divide un sector en bays, los cuales corresponden a las columnas transversales respecto al suelo y que se enumeran cada 20 pies; en rows o filas, que corresponden a la ubicación del contenedor en un bay, en sentido longitudinal respecto al suelo; y en tiers, los cuales corresponden al nivel en el cual se encuentran los contenedores dentro de un sector respecto del suelo.



Figura 15: Sistema BAROTI. Fuente: [Kaps, 2012]

4.2. Descripción de las Operaciones dentro del Patio

El modelo a resolver si bien no será con la totalidad de la información de un puerto real, se proporcionará datos correspondientes al Terminal Portuario de Arica. A continuación se describen los procesos relacionados al patio de contenedores. Algunos aspectos han sido omitidos ya que no son relevantes para el transcurso y contextualización del problema.

4.2.1. Proceso de Importación

Como se aprecia en la Figura 16, el Proceso de Importación comienza previo al atraque de la nave, esto empieza con el envío manifiesto de la carga por parte de la empresa naviera, donde entrega el listado de contenedores que serán descargados (en inglés, *Container Announcement List (CAL)*). Esto abre una “ventana de stacking” de 72 horas en la que la nave puede enviar al terminal portuario la lista completa de los contenedores a descargar. Una vez terminadas estas 72 horas, comienzan 8 horas en que se realiza la planificación de faena determinando, por rangos de turnos, qué camiones serán atendidos para cada faena (recepción, despacho, consolidado, y des-consolidado) y el orden en que serán retirados los contenedores. A la llegada de la nave al sitio de atraque predefinido en un tiempo estimado de llegada (en inglés *Estimated Time of Arrival, (ETA)*), la grúa *Quay Crane* (ver Figura 11) procede a la descarga de los contenedores de

importación en los tracto camiones internos, los cuales deben transportar los contenedores desde el muelle hasta los sectores pre-asignados, luego el camión es descargado por la grúa Reach Stacker (ver Figura 14) en uno de los stacks, el cual puede ser segregados por carga directa, indirecta o vacío.

Carga directa corresponde a la carga que será retirada por el cliente dentro de un periodo de 24 horas antes del arribo de la nave. Si el o los contenedores no han sido retirados antes de 48 horas del arribo, se comienza a cobrar una tarifa por almacenaje.

Carga indirecta es la que no será retirada por el cliente dentro de las primeras 24 horas antes del arribo del buque, a la cual se le cobra almacenaje y donde puede estar sin ingresar un máximo de 90 días por ley.

Los contenedores vacíos son almacenados, para que el dueño los retire. El motivo de arribo de contenedores vacíos es para tener contenedores para exportación, ya que al enviar un contenedor de exportación, luego de que llegue a su destino, se tiene que traer vacío para poder exportar de nuevo. Cuando hay poco tiempo para efectuar el desembarque de contenedores, éstos son apilados en un sitio aledaño del muelle con el objetivo de minimizar el tiempo de descarga del buque, para luego ser trasladado al patio.

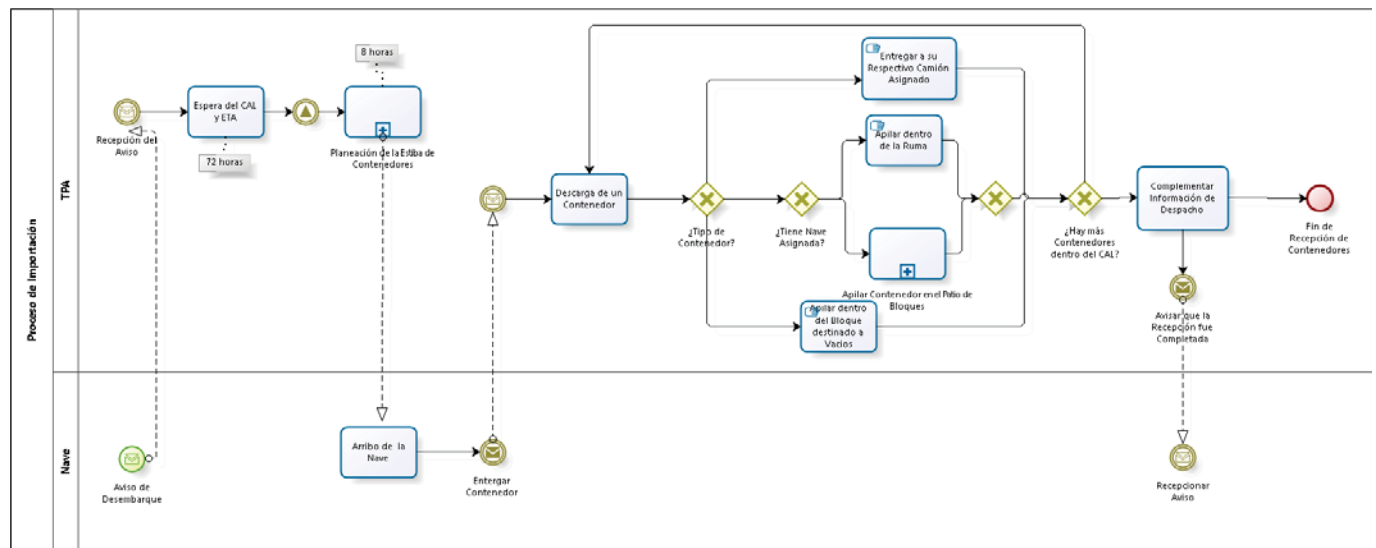


Figura 16: Proceso de Importación. Fuente: [Elaboración Propia]

4.2.2. Proceso de Exportación

Tal como se observa en la Figura 17, el Proceso de Exportación se inicia 72 horas previas al arribo del barco a puerto a un sector llamado sector de "pre-stacking", el cual no es más que un sector dentro del patio de contenedores reservado para recibir y almacenar temporalmente los contenedores de exportación traídos por los clientes. En este sector, los contenedores son

ordenados de manera inversa a la estiba del barco, para minimizar los movimientos innecesarios.

Aquí la descarga de estos camiones se realiza por la grúa Reachstacker. El plan de estiba se diseña considerando la estabilidad del barco en el proceso de carga, para esto debe considerarse el peso de los contenedores y el tipo de éstos, para lograr un centro de masa lo más equilibrado posible, para no comprometer la integridad del proceso de transporte. Los mismos aspectos deben considerarse para mantener la integridad de los mismo contenedores y/o su contenido. De este mismo modo, el puerto de destino de los contenedores también es un aspecto importante al momento de planificar la estiba. Se debe planificar tratando de satisfacer todos estos aspectos.

Cuando el barco haya atracado en el puerto, los contenedores van siendo cargados en los tracto camiones por las Reachstacker para ser llevados al sector del muelle donde las grúas QC's toman el contenedor y lo embarcan en el barco.

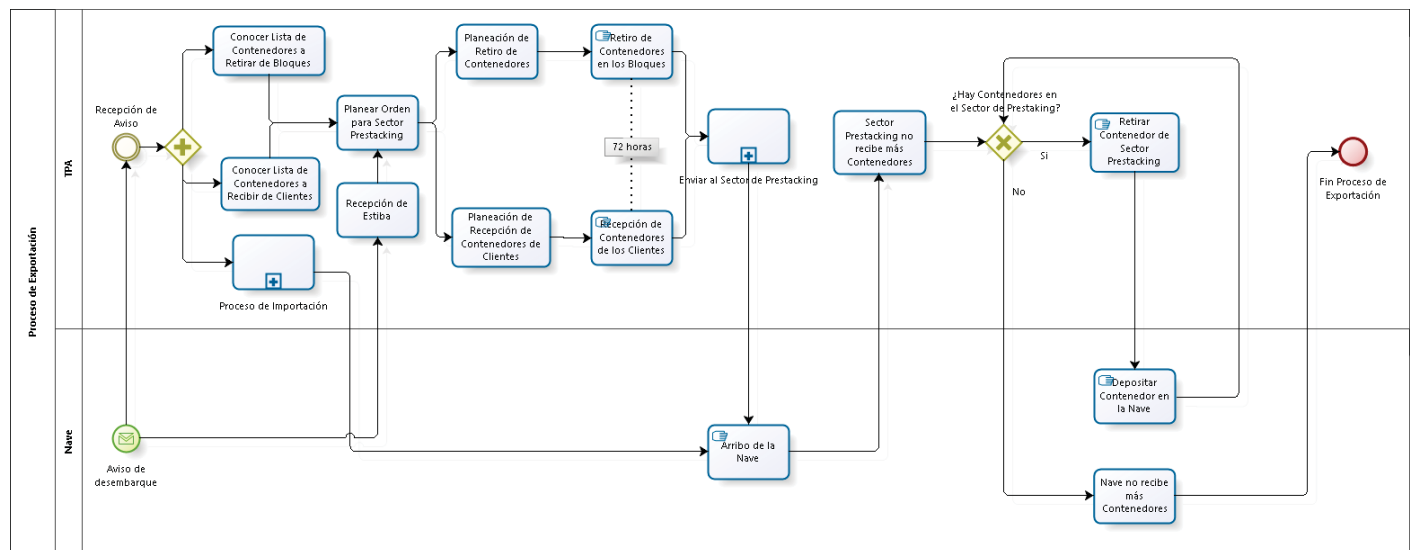


Figura 17: Proceso de Exportación. Fuente: [Elaboración Propia]

5. Literatura Previa

5.1. Algoritmo Genético para YAP: Énfasis en la Minimización de los Tiempos de Almacenaje y Balanceo de Cargas entre Bloques

Hecha por *Mohammad Bazzazi, Nima Safaei y Nikbakhsh Javadian* [Bazzazi et al., 2009] presentan una versión extendida de un SSAP (*Storage Space Allocation Problem*). Se puede definir como la asignación momentánea de contenedores de entrada y salida a los bloques de almacenamiento en cada periodo de tiempo, tratando así, de balancear las cargas entre bloques para minimizar los tiempos de almacenaje y retiro de los contenedores. Se considerará distinto tipo de contenedores y tamaños. Para este trabajo se trabajará con dos tipos de contenedores:

- Contenedores de importación que se encuentran en barcos antes de ser descargados y asignados al patio de contenedores (C_1).
- Contenedores que ya están en el patio de contenedores a la espera de ser retirados por los clientes (C_2).

El principal objetivo en este trabajo consiste en minimizar el tiempo promedio en que una embarcación permanece atracado en el puerto.

En el puerto en el que se desarrolló este trabajo, los contenedores C_1 y C_2 representaban el 85 % del total de la carga que se trabajaba. Los contenedores de exportación son ubicados en un solo bloque cercano al sitio del barco en donde serán embarcados dichos contenedores. Esta es el principal motivo por el cual no se considerarán contenedores de exportación.

La principal novedad en este trabajo consiste en el uso de un enfoque de “horizonte rodante”, donde el horizonte de la planificación será de 4 días. Cada día será dividido en 6 periodos de 4 horas. Al comenzar el día 1, un plan de asignación de espacio será determinado para los 24 periodos entre los días 1-4, esta asignación solo se ejecutará el primer día y un nuevo plan de 4 días se determinará al final del primer día, así obteniendo información más actualizada. Esto se irá repitiendo cada día.

Algo importante a destacar es que existen contenedores con tiempos de despacho desconocidos al momento de hacer la planificación, o contenedores con tiempos de despacho que están lejos del horizonte de planificación. Sus cargas de trabajo no se hacen efectivas dentro del horizonte de planificación, en consecuencia aquellos contenedores no son directamente incluidos en el modelo de asignación de espacios. Para tener en cuenta que esto puede afectar en el futuro, los contenedores son distribuidos en bloques en proporción del espacio disponible de cada bloque al principio del horizonte de planificación. Esta aproximación tiene un efecto marginal en el rendimiento general. La mayoría de los contenedores que son asignados, son de información conocida. En este caso se requerirán tres tipos de datos para cada contenedor de importación: la fecha en que el contenedor debe ser descargado del barco y llevado a los bloques, el tipo de contenedor y los bloques disponibles para ese contenedor. En el trabajo afirman que un balanceo de las cargas de los bloques asegura una disminución en los tiempos de permanencia de los barcos en el muelle.

5.1.1. Definición Formal

El SSAP extendido se formula como un modelo de programación matemática basado en las siguientes suposiciones. Básicamente, se debe determinar el número de contenedores C_1 y C_2 almacenados en cada bloque para cada período de planificación.

- Para la carga de trabajo solo se consideran contenedores de importación (C_1 y C_2), los que deben ser reubicados del barco al patio de contenedores.
- Existen suficientes recursos (Grúas de patio, grúas de muelle y camiones) para manejar las cargas de trabajos en el terminal considerado.
- Los contenedores son de distintos tipos y tamaños. Las cargas de trabajos para cada tipo y tamaño son medidos en términos del número de contenedores de ese tipo. Los contenedores de distinto tipo y tamaño pueden ser mezclados dentro de un bloque.
- Los bloques disponibles a cada tipo se conocen de antemano.

Parámetros de Entrada

- B : El número total de bloques en el patio de contenedores.
- T : El número de periodos de planificación dentro del horizonte de planificación, $T = 24$.
- R : La cantidad de tipos de contenedores.
- C_i : La capacidad del bloque i , $1 \leq i \leq B$.
- $\tilde{D}_{tkr}$: La cantidad esperada de contenedores C_1 de tipo r que son descargados en el periodo t y son retirados por los clientes en el periodo $t + k$, $1 \leq t \leq T, 0 \leq k \leq T - t$.
- β_{itr} : La cantidad esperada de contenedores C_1 de tipo r descargados en el periodo t , asignados al bloque i , con tiempos de retiro desconocido o con tiempo de retiro fuera del horizonte de planificación, $1 \leq t \leq T, 1 \leq i \leq B$.
- P_{itr}^0 : La cantidad total esperada de contenedores C_2 de tipo r que llegan al terminal en el periodo t y que deben ser embarcados en el periodo $t + k$, $1 \leq t \leq T, 0 \leq k \leq T - t$.
- V_{ir} : El inventario inicial del tipo r en el bloque i , $1 \leq i \leq B$.
- $S_{ir} = 1$: Si el contenedor tipo r puede ser asignado al bloque i ; de otra manera $S_{ir} = 0$.
- η : Densidad permitida para cada bloque.
- M : Un entero positivo arbitrariamente grande.

Variables de Decisión

- D_{itkr} : La cantidad de contenedores C_1 de tipo r con información completa almacenados en el bloque i que son descargados de los barcos en el periodo t y que deben ser retirados en el periodo $t + k$, $1 \leq i \leq B, 1 \leq t \leq T, 0 \leq k \leq T - t$.

- D_{itr} : El número total de contenedores C_1 de tipo r (de información completa o parcial) almacenados en el bloque i que son descargados en el período t , $1 \leq i \leq B, 1 \leq t \leq T$.
- P_{itr} : El número total de contenedores C_2 de tipo r almacenados en el bloque i que son retirados en el período t , $1 \leq i \leq B, 1 \leq t \leq T$.
- V_{itr} : El número de contenedores del tipo r (C_1 y C_2) dentro del bloque i al final del período t , $1 \leq i \leq B, 1 \leq t \leq T$.

Modelo Matemático

La función objetivo queda denotada como:

$$\min Z = \sum_{t=1}^T \sum_{r=1}^R (w_1 [\max_{i=1}^B \{D_{itr}\} - \min_{i=1}^B \{D_{itr}\}] + w_2 [\max_{i=1}^B \{D_{itr} + P_{itr}\} - \min_{i=1}^B \{D_{itr} + P_{itr}\}]) \quad (1)$$

La función objetivo busca un balance de los contenedores C_1 y el total de contenedores entre los bloques para cada tipo de contenedor y cada periodo de tiempo. El termino “ $\max(\cdot) - \min(\cdot)$ ” implica el desbalance entre las cargas de trabajo de cada bloque, por lo tanto, se busca que “ $\max(\cdot) = \min(\cdot)$ ” y quiere decir que los contenedores están distribuidos uniformemente entre los bloques. El primer termino se enfoca en el balance de los contenedores C_1 , y el segundo termino se enfoca en el balance de los contenedores C_1 y C_2 simultáneamente. La importancia de cada término está dada por los coeficientes de peso w_1 y w_2 . En la ecuación D_{itr} , es el total de contenedores relacionados a los barcos que deben ser almacenados en el bloque i en el periodo t y $D_{itr} + P_{itr}$ es el número total de contenedores (asociando tanto a los barcos como a los clientes) a ser almacenados en el bloque i en el periodo t . Por lo tanto, los dos términos en la ecuación miden el desbalance de los contenedores de descarga y el desbalance del total de contenedores en cada periodo de planificación, respectivamente. Los pesos de los términos, w_1 y w_2 se ajustan dependiendo de la importancia relativa, según como lo interprete el terminal. Sujeto a:

$$\sum_{i=1}^B D_{itkr} = \tilde{D}_{tkr} \quad (2)$$

Esta restricción (2) asegura que el total de contenedores C_1 de tipo r de información completa esperando a ser asignado, $\tilde{D}_{tkr}$, es la suma de los contenedores asignados a todos los bloques.

$$D_{itr} = \beta_{itr} + \sum_{k=0}^{T-t} D_{itkr} \quad (3)$$

Esta restricción (3) asegura que el total de contenedores C_1 de tipo r asignados al bloque i durante el período t , D_{itr} , es la suma del total de contenedores C_1 de información completa, D_{itkr} , y aquellos contenedores con tiempo de retiro desconocido en el horizonte de planificación, β_{itr} .

$$P_{itr} = \sum_{k=0}^{T-1} D_{i(t-k)kr} + P_{itr}^0 \quad (4)$$

Esta restricción (4) indica que el número de contenedores C_2 de tipo r asignados en el bloque i durante el período t , P_{itr} , consiste de dos partes. La primera son los contenedores transferidos desde los contenedores C_2 correspondientes que llegan en el horizonte de planificación. La segunda parte consiste en contenedores C_2 inicialmente almacenados en el bloque i para ser cargado en barcos en el período t en el horizonte de planificación actual.

$$V_{itr} = V_{i(t-1)r} + \tilde{D}_{itr} - P_{itr} \quad (5)$$

$$V_{itr} \leq \eta C_i \quad (6)$$

La restricción (5) representa la actualización de inventario, V_{itr} , período a período. La restricción (6) asegura que el inventario de cada bloque en cada período no exceda la densidad permitida. La densidad del bloque implica el hecho que una porción del bloque debe ser usado para transferir contenedores dentro del bloque.

$$\tilde{D}_{itr} \leq MS_{ir} \quad (7)$$

La restricción (7) asegura que cada tipo de contenedor poder ser asignado solo a los bloques permitidos.

5.1.2. Algoritmo Genético

Se escoge esta técnica dada su eficiencia para este tipo de problemas y el SSAP no había sido resuelto mediante esta técnica. Considerando la principal variable de decisión, D_{itkr} , se usa una estructura de 4 dimensiones para representar la solución del SSAP extendido. Estas 4 dimensiones indican los índices relacionados al bloque, período de descarga, período de despacho y tipo de contenedor.

Generación de la Población Inicial

Para generar soluciones aleatorias factibles para la población inicial, proponen el siguiente procedimiento:

- Generar valores D_{itkr} dentro del intervalo $[\text{máx}\{\tilde{D}_{tkr}\}, \text{mín}\{\tilde{D}_{tkr}\}]$, donde $\sum_{i=1}^B S_{ir} D_{itkr} = \tilde{D}_{tkr}, \forall i, t, k, r$.
- Calcular valores D_{itr} usando la restricción (3).
- Calcular valores P_{itr} usando la restricción (4).
- Calcular valores V_{itr} usando la restricción (5). Considerando la restricción (6). Si la restricción (6) es violada, entonces se agrega un valor $\alpha(V_{itr} - \eta C_i)$ a la función objetivo como una penalización donde α indica el coeficiente de penalización.

Cruzamiento Aritmético

Para mantener la factibilidad de las nuevas generaciones se propone el uso del siguiente mecanismo como operador de cruce. La cruce aritmética explora el espacio de solución manteniendo la factibilidad simultáneamente. Este operador genera la descendencia como la combinación lineal complementaria de los padres de la siguiente forma:

$$Descendencia \equiv \lambda x Padre_1 + (1 - \lambda)x Padre_2$$

Donde λ es un valor aleatorio dentro del intervalo $[0, 1]$. Así, los valores D_{itkr} son generados como:

$$D_{itkr}(Descendencia) = \lambda x D_{itkr}(Padre_1) + (1 - \lambda)x D_{itkr}(Padre_2)$$

Este mecanismo garantiza que la descendencia se mantendrá factible si sus padres lo son. Ocurre un problema ya que estos valores deben ser enteros. Por lo tanto la restricción (2) podría no cumplirse y se incurre en un error $\varepsilon = \sum_{i=1}^B [D_{itkr}(Descendencia)] - \bar{D}_{tkr}$. Para resolver esto, el valor de ε se suma heurísticamente al D_{itkr} con el valor mínimo.

Mutación

La principal tarea de la mutación es mantener la diversidad de la población al pasar las generaciones. En este trabajo se utiliza el mecanismo de mutación llamado “*Stepping Stone mutation*”. Este mecanismo asegura que las soluciones mutadas se mantendrán factibles si la solución sujeta a mutación lo es. El procedimiento es el siguiente:

- Seleccionar aleatoriamente una solución.
- Para cada r, k y t . Sea $D_{ptkr} = \max_{i=1}^B D_{itkr}$, $D_{qtkr} = \min_{i=1}^B D_{itkr}$ y $\delta_{tkr} = D_{ptkr} - D_{qtkr}$.
- Cambiar los valores de D_{ptkr} y D_{qtkr} a $D_{ptkr} \longrightarrow D_{ptkr} - \frac{\delta_{tkr}}{2}$ y $D_{qtkr} \longrightarrow D_{qtkr} + \{\delta_{tkr} - \frac{\delta_{tkr}}{2}\}$, respectivamente.

Selección

Para la selección, se utiliza el típico mecanismo de selección por ruleta, explicada en detalle en el capítulo 3.1.3.

Estrategia de Aceptación de Descendencia

Se usa una estrategia tipo semi-greedy para aceptar las descendencias. En este caso una descendencia es aceptada para la siguiente generación si su aptitud es menor que la del promedio de sus padres.

Criterio de Terminación

Aquí se usan dos criterios: el típico mediante ciertas cantidad de generaciones y una cierta desviación estándar de los valores de aptitud de los genotipos de la generación actual.

Resultados

Se efectuaron pruebas para 22 set de datos, con distintos tamaños, y para comparar se utilizó el método de Branch & Bound. Este método solo pudo resolver los set de datos más pequeños, ya que para los set más grandes se excedió en el tiempo razonable (3 horas). El algoritmo genético

no tuvo problemas en resolver todo el set. El set más grande que pudo resolver el método B&B (7.650 segundos), fue resuelto por el GA en 30 segundos. La implementación del GA para este problema tuvo excelentes resultados.

5.2. Algoritmo Genético para YAP: Énfasis en la Diversidad de Contenedores

I. Ayachi, R. Kammarti, M. Ksouri y P. Borne [Ayachi et al., 2013] abarcaron el problema mediante la implementación de un algoritmo genético basado en respetar la fecha de entrega de los clientes, reducir las operaciones con los contenedores en el patio del terminal, en su fuerte consideraban varios tipos de contenedores (normal, vacío, con techo abierto, con su lado abierto, tanque y refrigerado), y el orden final del patio de contenedores se rige por algunas restricciones según el tipo de contenedor. Los resultados de este proyecto se compararán con LIFO (proceso aún usado en los terminales portuarios que consiste en colocar el contenedor en el último lugar donde hubo un contenedor, apoyado por la planificación manual en la experiencia humana).

5.2.1. Definición Formal

Parámetros de Entrada

- i : Índice de contenedores.
- n_1 : Número máximo de contenedores en el eje X
- n_2 : Número máximo de contenedores en el eje Y
- n_3 : Número máximo de contenedores en el eje Z
- $N_{c_{piso}}$: Número máximo de contenedores por piso, $N_{c_{piso}} = n_1 * n_2$
- $N_{c_{piso}}(k)$: Número de contenedores en el piso k
- $N_{stock_{reg}}$: Número de bloques para contenedores que no requieren acceso a electricidad
- $N_{stock_{refrig}}$: Número de bloques para contenedores refrigerados
- j : Índice para bloques de contenedores, $j = 1, \dots, N_{stock_{reg}} + N_{stock_{refrig}}$
- d_i : Fecha de entrega de cada contenedor i
- $N_c(T)$: Número de contenedores para cada tipo, con T indica el tipo de contenedor, $T = 1, 2, \dots, 6$

$$T = \begin{cases} 1 & \text{si es un contenedor Dry} \\ 2 & \text{si es un contenedor vacío} \\ 3 & \text{si es un contenedor de techo abierto} \\ 4 & \text{si es un contenedor de lado abierto} \\ 5 & \text{si es un contenedor tanque} \\ 6 & \text{si es un contenedor refrigerado} \end{cases}$$

- N_T : Número de tipo de contenedores
- $N_{c_{max}}$: Número máximo de contenedores, con $N_{c_{max}} = (N_{stock_reg} + N_{stock_refrig})n_1 * n_2 * n_3$

Variables de Decisión

$Pos(x, y, z, j)$ es una variable de decisión. Es una estructura con dos variables tales como V y $Tipo$. Donde $x \in [1, \dots, n_1], y \in [1, \dots, n_2], z \in [1, \dots, n_3], j \in [N_{stock_reg} + N_{stock_refrig}]$.

$$Pos(x, y, z, j).V = \begin{cases} 1 & \text{si tenemos un contenedor en esta posición} \\ 0 & \text{otro} \end{cases} \quad (1)$$

$$Pos(x, y, z, j).Tipo = \begin{cases} T & \text{si tenemos un contenedor en esta posición} \\ 0 & \text{otro} \end{cases} \quad (2)$$

Modelo Matemático

Se consideró una cantidad finita de bloques. La función de aptitud tiene como objetivo minimizar la manipulación de contenedores y la minimización del tiempo en que el barco lleva en el muelle. Para esto, se utilizó la siguiente función:

$$\begin{aligned} \min \sum_{t=1}^{N_T} \sum_{i=1}^{N_c(T)} \sum_{j=1}^{N_{stock_reg} + N_{stock_refrig}} P_{i,t} m_{i,t,j} Pos(x, y, z, j).V \\ \forall x = 1, \dots, n_1, \forall y = 1, \dots, n_2, \forall z = 1, \dots, n_3 \end{aligned}$$

Donde:

- P_i : Valor de prioridad dependiendo de la fecha de entrega d_i del contenedor i al cliente, con $P_i = 1/d_i$
- $m_{i,j}$: El mínimo número de contenedores que hacen interferencia para descargar el contenedor i que se encuentra en el bloque j

Constantes

Las constantes (3) y (4) asegura que el nivel de la piso del bloque contiene más contenedores que directamente encima.

$$N_{c_{piso}}(k) \geq N_{c_{piso}}(k+1), k = 1, \dots, N_{piso} \quad (3)$$

$$\text{Si } Pos(x, y, z, j) = 0, \text{ entonces } Pos(x, y, z-1, j) = 0 \quad (4)$$

La constante (5) indica que un contenedor con techo abierto no puede tener otro contenedor encima.

$$\text{Si } \begin{cases} Pos(x, y, z, j).V=1 \\ y \\ Pos(x, y, z, j).Tipo=3 \end{cases} \text{ entonces } Pos(x, y, z+1, j).V = 0 \quad (5)$$

La constante (6) indica para un contenedor de lado abierto.

$$Si \left\{ \begin{array}{l} \text{Pos}(x,y,z,j).V=1 \\ y \\ \text{Pos}(x,y,z,j).Tipo=4 \end{array} \right. \quad entonces \left\{ \begin{array}{l} \text{Pos}(x,y,z+n,j).V=0 \\ \forall n \in [1, \dots, n_3 - z[\\ \text{Pos}(x+m,y,z,j).V=0 \\ \forall m \in [1, \dots, n_1 - x] \end{array} \right. \quad (6)$$

La constante (7) indica que puede haber otro contenedor vacío sobre uno vacío, o que no haya ninguno.

$$Si \left\{ \begin{array}{l} \text{Pos}(x,y,z,j).V=1 \\ y \\ \text{Pos}(x,y,z,j).Tipo=2 \end{array} \right. \quad entonces \left\{ \begin{array}{l} \text{Pos}(x,y,z+1,j).V=0 \\ \text{ó} \\ \text{Pos}(x,y,z+1,j).Tipo=2 \end{array} \right. \quad (7)$$

La constante (8) sugiere que un contenedor tanque debe ir sobre otro contenedor tanque.

$$Si \left\{ \begin{array}{l} \text{Pos}(x,y,z,j).V=1 \\ y \\ \text{Pos}(x,y,z,j).Tipo=5 \end{array} \right. \quad entonces \left\{ \begin{array}{l} \text{Pos}(x,y,z+1,j).Tipo=5 \\ \text{ó} \\ \text{Pos}(x,y,z+1,j).V=0 \end{array} \right. \quad (8)$$

La constante (9) indica que los contenedores refrigerados deben asignarse a bloques equipados para la toma de corriente.

$$Si \left\{ \begin{array}{l} \text{Pos}(x,y,z,j).V=1 \\ y \\ \text{Pos}(x,y,z,j).Tipo=6 \end{array} \right. \quad entonces \quad j \in [1, \dots, N_{stock_refrig}] \quad (9)$$

5.2.2. Algoritmo Genético

Genotipo

De acuerdo con la variable de decisión $\text{Pos}(x, y, z, j)$, se utiliza una estructura de representación de cuatro dimensiones, cada una representa el área de almacenamiento de contenedores. Estas dimensiones indican, respectivamente, el contenedor de coordenadas en el plan de (X, Y, Z) y el número del bloque asignado.

Generación de la Población Inicial

La población inicial se generó aleatoriamente, respetando las condiciones explicadas de (1) a (9).

Selección

En este trabajo se utilizó el mecanismo de selección por ruleta, explicado en el capítulo 3.1.3.

Cruzamiento

Para el crossover se utilizó el mecanismo de dos puntos (explicado en el capítulo 3.2.2) con una probabilidad de crossover fija de un 70 %.

Mutación

El operador de mutación consiste en permutar dos contenedores seleccionados al azar. La probabilidad de mutación se estableció en un 20 %.

Criterio de Terminación

El proceso termina cuando la solución no mejora después de ciertas iteraciones.

Resultados

Para este trabajo se comparó con el LIFO (almacena en el primer lugar donde estuvo el último contenedor), siendo un método aún usado en terminales portuarios obteniendo el GA mejores resultados que aplicando el uso de LIFO.

5.3. Algoritmo Genético para YAP: Énfasis en el Espacio Asignado a Contenedores

Plantado por Christopher O'Shee [O'Shee, 2013], postulando al título de Ingeniero Ejecución en Informática de la Pontificia Universidad Católica de Valparaíso, presenta una solución al Yard Allocation Problem a través de la implementación de un algoritmo genético, basándose en los objetivos de minimizar el tiempo en que un barco lleva atracado en el puerto y optimizar la distribución de contenedores en el patio de estos.

Para el primer objetivo estuvo basado en la publicación de Mohammad Bazzazi, Nima Saffae y Nikbakhsh Javadian, *A genetic algorithm to solve the storage space allocation problem in a container terminal* [Bazzazi et al., 2009] proponiendo que la distribución de cargas de trabajo y una buena distribución de los contenedores en los bloques función de buena manera para minimizar el tiempo de un barco en el puerto. Por lo que se hace una distribución de trabajo segregada en 5 días, considerando solo el proceso de exportación de contenedores (descrito en el capítulo 5.2.2), proponiendo una solución a través del uso de un algoritmo genético.

Para el segundo objetivo, propuso según las coordenadas BAROTI un orden similar al que se trabajó en el proceso de estiba de un barco un algoritmo tipo greedy que en base a las interferencias para retirar un contenedor, el peso de un contenedor, minimizará los movimientos innecesarios.

5.3.1. Definición Formal

Parámetros de Entrada

Previo a la ejecución del algoritmo, se deberán tener almacenados estos datos en una base de datos.

- Planificación naviera:
 - o Nave.
 - o Viaje.
 - o ETA.
 - o Fecha de arribo.
 - o Sitio del muelle.
- Distribución del puerto:
 - o Cantidad de bloques.
 - o Tamaño de cada bloque.
 - o Distancia entre bloques a sitio de atraque.
- Container Assignment List (CAL):
 - o Contenedor.
 - o Peso.
 - o Puerto de destino.
 - o Fecha de viaje.

Variables de Decisión

La variable de decisión será el bloque en el que se almacenará cada contenedor, para calcular las cargas de trabajo se suman los flujos de entrada y salida de un bloque en un periodo de tiempo determinado, la distancia bloque-muelle correspondiente al contenedor queda determinada por la distribución del terminal, la cantidad de interferencias al retirar un contenedor de un bloque, será calculado por el algoritmo tipo greedy.

Modelo Matemático

El objetivo es minimizar tres aspectos importantes:

- La diferencia de cargas de trabajo entre bloques. El peso estará determinado por α .
- Las distancias entre cada contenedor almacenado en el patio y el respectivo sitio a donde será embarcado. El peso está determinado por β .

- La cantidad de interferencias encontradas al extraer un contenedor de su respectivo bloque para transportarse al respectivo sitio de atraque. El peso está determinado por γ .

Sea:

- C_{it} : Número de contenedores que debe procesar el bloque i en el periodo t .
- d_i : Distancia entre el bloque asignado al contenedor i y el sitio del muelle correspondiente.
- I_i : Número de interferencias detectadas para sacar el contenedor i del patio de contenedores.

$$\min Z = \alpha \sum_{t=1}^T (\max_{i \in B} C_{it} - \min_{i \in B} C_{it}) + \beta \sum_{i=1}^C d_i + \gamma \sum_{i=1}^C I_i \quad (10)$$

5.3.2. Algoritmo Genético

Genotipo

Se determinará en que bloque deberá ir ubicado cada uno de los contenedores que lleguen dentro del periodo de planificación, por lo que cada posible solución será representada por un individuo compuesto por números enteros, por ejemplo:

$$G = C_1 C_2 C_3 C_4 C_5 C_6 \dots C_n$$

La posición de cada gen representa un contenedor, el valor de cada gen representa el bloque en el que podría ser almacenado el contenedor, por lo que su valor debe fluctuar entre los identificadores de bloques.

Generación de la Población Inicial

Se opta por una población más dispersa para poder abarcar en gran parte el espacio de búsqueda, por lo que aleatoriamente se crearon cromosomas sin importar si estaban repetidos. El tamaño de la población será proporcional a la cantidad de contenedores en el periodo de tiempo a evaluar.

Selección

Para la selección se utilizó el mecanismo de torneo, descrito en el capítulo 3.2.1.

Cruzamiento

Se utilizó cruzamiento en un punto (descrito en el capítulo 3.1.4. Se aplicó un valor base para la probabilidad de cruzamiento (p_c) de 90 %.

Mutación

Cada gen de cada individuo tiene una probabilidad de ser mutado. Si un gen es seleccionado para la mutación, se toma un par de genes aleatorios y se intercambian, no generando conflicto de factibilidad.

Criterio de Termino

Se explica que el valor óptimo no se conoce, por lo que al algoritmo genético se detendrá al cabo de cierto número de generaciones.

Resultados

Luego del termino de 1000 iteraciones, se realizaron pruebas de rendimiento con distintos peaks considerando los factores que se estaban optimizando, para luego comparar los resultados con datos extraídos del Terminal Puerto de Arica obteniendo buenos resultados.

6. Propuesta de Solución

Para este proyecto se hará una continuación del caso de estudio de Christopher O'Shee [O'Shee, 2013], habrá un ordenamiento de los datos y habrá una implementación extensible aunque de todas formas hay procesos usados que dieron buenos números. Además, se va a priorizar en dos objetivos principales además de una novedad adicional al problema, para el primero mediante la implementación de un algoritmo genético y para el segundo con un algoritmo tipo *Greedy*.

- Generar un Balance entre las Cantidades de Contenedores por Bloque.
- Disminuir el Tiempo de Permanencia de una Embarcación en el Puerto.
- Planificar la Ubicación de los Contenedores Dentro de un Bloque.

Este trabajo no considerará en cuenta otros factores que intervengan dentro del patio de contenedores, por lo que se debe asumir que los recursos siempre serán suficientes. Por ejemplo, se asumirá la existencia de suficientes camiones para el transporte de contenedores que permitan en tráfico expedito de un contenedor, tanto para exportación como importación. Para la formulación básica y set de datos, existieron las opciones de trabajar con datos e información del puerto STI de San Antonio, Chile, utilizados por [Covarrubias and Tapia, 2012] y [O'Shee, 2013]. Pero la información que se encontraba más abundante resultó ser la del Terminal Portuario de Arica, Chile, Por lo que se toma la decisión de usar solo información de procesos y set de datos pertenecientes a este último puerto.

6.1. Generar un Balance entre las Cantidades de Contenedores por Bloque

Esto es un inconveniente dentro del patio de contenedores, ya que el objetivo que se busca es una correcta distribución de los contenedores por el mismo patio. Existe el problema cuando todos los contenedores tienden hacia un mismo bloque en un periodo t , esto genera que la maquinaria genere un cuello de botella y los contenedores tengan que esperar a ser ubicados en los bloques, ocasionando un mal uso de la infraestructura portuaria y un costo innecesario.

Para solucionar este problema se propone generar un balance, que todas las maquinarias destinadas al movimiento de contenedores hagan un uso equilibrado y manipulen la cantidad suficiente de contenedores, con la finalidad de eliminar el cuello de botella y hacer un uso eficiente de las maquinarias del puerto.

6.2. Disminuir el Tiempo de Permanencia de una Embarcación en el Puerto

Esto es un problema tanto para la empresa portuaria y la empresa naviera. Para la primera porque esto genera un retraso en sus operaciones y no podrá atracar un nuevo barco a puerto hasta que el primer barco haya zarpado. Para la segunda porque cada cierto tiempo que el barco

sigue permaneciendo atracado en el puerto, la empresa naviera deberá costear una tarifa adicional.

Este objetivo visto a gran escala, hace investigar minuciosamente cuales son motivos principales que generan este retraso y se llegó a la conclusión de que la primera causa es el tiempo en embarco/desembarco de contenedores.

Para dar una óptima solución se generará un algoritmo genético de “tres niveles” . En *A genetic algorithm to solve the storage space allocation problem in a container terminal* [Bazzazi et al., 2009] se ve que el balance de las cargas de trabajo de los bloques funciona muy bien para minimizar el tiempo de permanencia de una nave. En ese trabajo solo se mencionó el flujo de importación ya que este flujo representa el 85 % en el terminal considerado (Shahid Rajaei Terminal, Irán). En este trabajo se tomará en cuenta solo el proceso de exportación, ya que en el de importación hay un despacho relativamente rápido después de la descarga, como indica en [O’Shee, 2013] y explicado en el capítulo 3.2.1 los retiros de los contenedores es relativamente rápida mediante el sistema de citas, por lo que se asumirá que los contenedores tiene asignada una cita a un cliente que va a retirar el contenedor en el momento preciso.

También para evitar complicar más el proyecto, no se considerará la distribución de los barcos alrededor del puerto, ya que eso es otro problema que es conocido como *Berth Allocation Problem*. Pero no dejará de ser importante la distancia entre los puertos de atraque y la ubicación del bloque más óptimo para recibir dicho contenedor.

6.3. Planificar la Ubicación de los Contenedores Dentro de un Bloque

Dentro de los bloques de contenedores, un problema habitual es la existencia de realizar movimientos innecesarios de contenedores (principalmente su uso radica en el retiro de otro contenedor que necesita ser retirado y se encuentra debajo del primero). Esto tiene su origen en una mala distribución de contenedores dentro de un bloque, generando un alto costo en electricidad (si se realiza el movimiento del contenedor mediante la grúa RMG) o en combustible (si se usa la grúa RGT) levantar un contenedor (considerar que un contenedor estándar de 20’ puede pesar hasta casi 27 toneladas) además de movimientos innecesarios dentro del bloque. Por esta razón, se considera dentro de los objetivos el minimizar estos movimientos ante el alto costo que generan.

El patio de contenedores será dividido en bloques, como muestra. La mejor solución será usar el sistema de coordenadas BAROTI (*bays, rows y tiers*) y será ubicado en un stack. Estos movimientos de contenedores serán efectuados por las grúas RMG o RGT, por lo que cada bloque estará solventado por una o más de estas máquinas. Se buscará la mejor solución para los contenedores tipo Dry con la importancia del peso que tenga cada contenedor, ya que por ejemplo el o los contenedores de abajo no pueden pesar menos que el de arriba. Además se considerará el tamaño de cada contenedor que serán para este trabajo de 20 y 40 pies de largo.

El motivo principal para usar contenedores solo de tipo Dry es que son de mayor flujo que el resto (tanque, cielo abierto, lado abierto, refrigerado, vacío) y por sus requerimientos espe-

ciales simplemente se trabajarán en un bloque dedicado para cada tipo o con algún otro tipo de contenedor en los que no haya ningún proceso que optimizar por lo cual no se considerarán.

6.4. Definición Formal de la Solución

En base a lo anterior, se procede a definir formalmente el problema dividido en dos etapas, la solución final de la primera etapa servirá para la ejecución de la segunda, para cada nivel se ejecutará un algoritmo correspondiente.

Nivel 1

Se generará un algoritmo tipo *Greedy* que servirá para la generación de la población inicial. En base a las distintas exigencias que tiene el TPA respecto a la ubicación de sus contenedores es que se propone generar una población que respete dichas reglas y distribuya todos los contenedores que vayan llegando alrededor del patio de forma de generar individuos lo más diversos posibles.

Nivel 2

La población generada en el nivel 1 será de mucha importancia ya que en este nivel, apoyará la ejecución del algoritmo genético, el cual determinará mediante iteraciones, cuales serán las posiciones en donde debe ubicarse cada contenedor, de tal manera que logre un balance de los contenedores por bloque, que logre minimizar la distancia entre el muelle de atraque y el bloque a asignar y que logre minimizar las interferencias para el retiro de los contenedores. Todo esto demostrado por la función objetivo.

6.4.1. Supuestos y Definiciones

- Los parámetros (N° de sectores, capacidad, tipo y distancia entre ellos) de los sectores de almacenamiento se basarán en los datos reales del TPA.
- Existen suficientes recursos para el tránsito expedito entre *quay cranes* y el patio de contenedores (camiones), el tiempo para estos efectos es despreciable.
- Aunque en el patio de contenedores del TPA utilizan maquinas *Reach Stacker* para el manejo de contenedor hacia el bloque, las grúas de patio para este problema serán del tipo *Rail Mounted Gantry* (RMG) y *Rubber Tired Gantry* (RTG) pueden mover contenedores a una tasa fija constante.
- La tasa de las grúas RMG y RTG incluye movimientos tanto de recepción como para embarques. Esto se considera para calcular las cargas de trabajo.
- Las unidades de tiempo son atómicas, nada pasa entre ellas. Los barcos llegan en el tiempo estimado entregado en la planificación naviera (ETA).

- Una vez ubicado un contenedor, este no puede cambiar de posición hasta que deba ser embarcado (Excepto en casos de reacomodo).
- Solo existen contenedores de 20 y 40 pies del tipo DRY.
- La estiba del barco se hará ordenadamente según los criterios de puerto de destino y peso, teniendo este último mayor relevancia para mantener el balance del barco.
- Para el muelle se considera una cantidad definida de sitios de atraque, hay una distribución de sitios de atraque al rededor del puerto que están a una determinada distancia de cada bloque.
- La diferencia entre las distancias de los distintos bloques con el sitio de embarque se considera para tomar decisiones, para esto se extraen datos de la distribución del terminal.
- Cada vez que se deba remover un contenedor de un bloque para ser embarcado, se deben tener en consideración las interferencias, se cuenta uno por cada contenedor que esté encima del contenedor a remover.
- Se considerarán viajes, los cuales relacionan un barco, su sitio asignado y el tiempo de llegada. Cada contenedor tiene pre-asignado un viaje en el cual se debe embarcar.
- Por las distintas políticas dentro del TPA, un contenedor puede llegar sin tener una nave asignada para su posterior retiro, esto no ocurrirá ya que todo contenedor debe tener una nave asignada, para así, evitar el uso de la *Ruma*.
- Solo se utilizarán bloques de exportación.
- Dentro del lapso de pre-stacking (72 horas), pueden recibir en todo ese periodo la lista de contenedores que se van a desembarcar, lo cual es más prominente en las horas finales del pre-stacking. Para este modelo se asumirá un ingreso constante de contenedores.

6.4.2. Estructura de los Parámetros de Entrada

Se detallará los datos de entrada que son necesarios en la ejecución del algoritmo. Esta información al igual que en los proyectos [Elorrieta, 2011] y [Maturana, 2014] utilizaron archivos en formato Excel. Esto último será de utilidad para conformar los set de datos disponibles del TPA, dentro de esta información se encuentran: Lista de Bloques, Planificación Naviera, Distancia entre Bloques y Sitios y Contenedores Iniciales.

Lista de Bloques

Esta lista corresponde a los bloques que están destinados a almacenar contenedores de exportación (EXPO) en el Terminal Portuario de Arica (ver Tabla 2). Estos bloques tienen asignado un código único de identificación, junto con la capacidad máxima de contenedores que pueda almacenar, la que está directamente relacionada con los valores asociados a dicho bloque, es decir bay, row y tier. Esta lista es utilizada para crear computacionalmente las matrices tridimensionales en las que serán almacenados los contenedores.

Código Bloque	Contenido	Tipo	Bay [u]	Row [u]	Tier [u]	Capacidad [u]
CP1	EXPO	DRY	2	24	5	240
CP2	EXPO	DRY	2	27	5	270
Z13	EXPO	DRY	20	3	5	300
Z14	EXPO	DRY	20	3	5	300
ZA1	EXPO	DRY	8	9	5	360
ZA2	EXPO	DRY	8	9	5	360
ZA3	EXPO	DRY	8	9	5	360
						2190

Tabla 2: Lista de Bloques. *Fuente: [Elaboración Propia]*

Planificación Naviera

La Tabla 3 muestra el orden de llegada de las naves a los distintos sitios de atraque. Además, contiene el sitio al que atracará cada nave y la cantidad de contenedores que se van a retirar. Lo anterior es necesario para decidir donde posicionar los contenedores en los bloques, ya que la distancia sitio-bloque influye en la decisión de ubicación de contenedores. La tabla se descompone en 6 columnas, la primera columna Código indica el código de la nave, la segunda columna Nombre señala el nombre de la nave, la tercera columna Ruta muestra la ruta posterior a la que se dirigirá la nave, la cuarta columna Cantidad Contenedores apunta a la cantidad de contenedores que serán retirados del puerto, la quinta columna ETA menciona la hora estimada de llegada de la embarcación y la sexta columna Muelle avisa en que muelle va a atracar la nave.

Código	Nombre	Ruta	Cantidad Contenedores [u]	ETA	Muelle
9327671	ARIANA	1243-A	97	30-10-12 8:00	S3
9200031	COLAN	026-N	13	30-10-12 22:00	S2B
7825411	MSC KYOTO	1243-R	40	31-10-12 6:00	S1

Tabla 3: Planificación Naviera. *Fuente: [Elaboración Propia]*

Distancia entre Bloques a Sitios

Los datos siguientes corresponden a las distancias matriciales relativas en que se encuentran los bloques respecto de los sitios de atraque de las naves en metros (ver Tabla 4). Estos datos son utilizados para minimizar la distancia, lo que es considerado al momento de evaluar la función objetivo. La tabla se divide tanto en filas como columnas, las filas indican el nombre de cada uno de los sitios de atraque y las columnas contienen los nombres de los bloques que almacenarán contenedores.

Bloque/Sitio [m]	S1	S2A	S2B	S3	S4	S5
CP1	281	348	364	299	584	719
CP2	304	368	381	317	474	598
Z13	333	396	418	351	486	619
Z14	296	361	384	316	512	645
ZA1	282	347	356	291	591	723
ZA2	320	384	392	326	623	760
ZA3	337	413	419	356	657	789

Tabla 4: Distancia de Bloques a Sitios. *Fuente: [Elaboración Propia]*

Contenedores Iniciales

Todos los días en TPA a las 6 AM, se toma una “imagen” del estado actual del patio del terminal, es decir, una lista de la ubicación de todos los contenedores existentes en dicho momento. Estos contenedores son datos de entrada, ya que están almacenados en los bloques que serán considerados en cada nueva asignación de contenedores (ver Tabla 5). La tabla se distribuye en 8 columnas, la primera columna Código indica el código único que posee cada contenedor, la segunda columna Tamaño muestra el tamaño de dicho contenedor, este puede ser de 20 o 40 pies, la tercera columna Peso señala el peso de cada contenedor en kilogramos, la cuarta columna Destino menciona el siguiente puerto de destino al que se dirige el contenedor, la quinta columna Nave dice que nave va a retirar al contenedor, la sexta columna Nave apunta a la dirección que tomará la nave para el siguiente destino, la séptima columna Bloque nos orienta en que bloque se encuentra el contenedor y la octava columna BAROTTI informa sobre la posición matricial respecto a la ubicación del contenedor en el bloque.

Código	Tamaño [pie]	Peso [kg]	Destino	Nave	Viaje	Bloque	BAROTTI
MEDU138844-4	20	22.670	BSFPO	MSC KYOTO	1243-R	CP1	010184
BMOU208326-4	20	23.220	PECLL	COLAN	26-N	CP1	012784
CMAU020198-8	20	22.230	CNXXX	COLAN	26-N	CP1	012380

Tabla 5: Contenedores Iniciales. *Fuente: [Elaboración Propia]*

Datos de Entrada

Observándolo desde otro modo, de los datos mencionados anteriormente desprendidos en información real del TPA en formato *.xls* quedan resumidos como:

- Distribución del Puerto:
 - Cantidad de Bloques.
 - Cantidad de Sitios de Atraque.
 - Distancia entre Bloques y Sitios.
 - Inventario Inicial de Contenedores.

- Bloques:
 - Código.
 - Contenido.
 - Tipo de Contenedor.
 - Bay.
 - Row.
 - Tier.
 - Capacidad.
- Planificación Naviera:
 - Código.
 - Nombre.
 - Ruta.
 - Cantidad de Contenedores.
 - ETA.
 - Sitio de Atraque.
- CAL (Container Assignment List):
 - Código.
 - Tamaño.
 - Peso.
 - Puerto Destino.
 - Nave.
 - Ruta.

Parámetros

1. $S = \{1..s\}$: Conjunto de segregaciones de contenedores. Cada segregación corresponde a un conjunto de contenedores con cualidades similares. Estas cualidades son:

$$S = \left\{ \begin{array}{l} \text{Nave en la que será embarcado} \\ \text{Puerto de destino} \\ \text{Nave en que será embarcado} \\ \text{Tamaño del contenedor} \end{array} \right.$$

2. $B = \{1..b\}$: Conjunto de bloques (sectores) dentro del patio de contenedores.
3. $I = \{1..i\}$: Conjunto de contenedores dentro del patio de contenedores.

4. $T = \{1..t\}$: Periodos de tiempo en la planificación.
5. $D = \{1..d\}$: Conjunto de destinos para cada contenedor.
6. $N_{Bloques}$: Número máximo de bloques dentro del patio.
7. C_j : Representa la cantidad de contenedores en el bloque j .
8. $C_{20} \subseteq S$: Conjunto de contenedores de 20 pies.
9. $C_{40} \subseteq S$: Conjunto de contenedores de 40 pies.
10. ROW_j : Número máximo de contenedores que puede ser localizado dentro del ancho del bloque j .
11. BAY_j : Número máximo de contenedores que puede ser localizado dentro del largo del bloque j .
12. $TIER_j$: Número máximo de contenedores que puede ser localizado dentro del alto del bloque j .
13. x_j : Ancho de contenedores del bloque j .
14. y_j : Largo de contenedores del bloque j .
15. z_j : Alto de contenedores del bloque j .
16. $N_{max_{nivel}} j$: Número máximo de contenedores en cada nivel del bloque j ($ROW_j * BAY_j$).
17. p_{ij} : Peso del contenedor i en el bloque j .
18. d_{ij} : Destino del contenedor i en el bloque j .

Variables

1. x_{ij}^t : Representa la cantidad de contenedores i que se almacena en el bloque j durante el periodo t .
2. y_{ij}^t : Toma el valor 1 si en el bay j se asignarán contenedores del tipo i durante el período de tiempo t .
3. C_{it} : Número de contenedores que debe procesar el bloque i en el periodo t .
4. D_i : Distancia desde el sitio donde se encuentra el contenedor i al bloque respectivo.
5. I_i : Número de interferencias de retirar el contenedor i del patio de contenedores.
6. $pos_{(x_j, y_j, z_j)}^d(u_{ij})$: Tomará el valor 1 si hay un contenedor i dentro del bloque j que va a un destino d en la ubicación u (x, y, z), 0 si no cumple.
7. $N_{c_{piso}}(k_j)$: Número de contenedores en el piso k del bloque j .

8. $pos_{(x_j, y_j, z_j)}(p_{ij})$: Peso p del contenedor i en el bloque j con la posición (x, y, z) .
9. $pos_{(x_j, y_j, z_j)}(d_{ij})$: Toma el valor 1 si todos los destinos d del tier además del contenedor i en la posición (x, y, z) del bloque j son iguales, 0 si no cumple.
10. $pos_{(x_j, y_j, z_j)}(t_{ij})$: Toma el valor 1 si el contenedor i en la posición (x_j, y_j, z_j) del bloque j tiene un tamaño de 20 pies (1 TEU), 0 si es de 40 pies (2 TEU) de tamaño.

6.4.3. Modelo Matemático

Función Objetivo

$$\text{mín} Z = \alpha \sum_{t=1}^T (\text{máx}_{i \in B} C_{it} - \text{mín}_{i \in B} C_{it}) + \beta \sum_{i=1}^C D_i + \gamma \sum_{i=1}^C I_i$$

La función objetivo busca 3 objetivos, primero balancear las cargas de trabajo entre bloques. El peso quedará determinado por α . Segundo, la distancia entre el sitio de atraque y el bloque donde se ubicará el contenedor quedará determinado por el peso β . Tercero, la cantidad de interferencias encontradas al extraer el contenedor de su respectivo bloque al sitio en el que será embarcado. Su peso queda determinado por γ .

Sujeto a:

$$\sum_{i \in S} x_{ij}^t \leq ROW_j * BAY_j * TIER_j - C_j \quad \forall j, t \quad (1)$$

(1) La cantidad total de contenedores i que existe dentro de un bloque j en un periodo t debe ser menor a la capacidad total de contenedores que pueden estar ubicados dentro del bloque j .

$$\sum_{i \in S} y_{ij}^t \leq 1 \quad \forall j, t \quad (2)$$

(2) Dentro de todo el patio de contenedores (segregación de bloques), el contenedor i solo puede estar físicamente ubicado en una posición de un bloque j en un periodo t .

$$x_{ij}^t \leq M pos_{(x_j, y_j, z_j)}^d(u_{ij}) \quad (3)$$

$$M \gg 0 \quad \forall d \in D, i \in S, t \in T, x_j \in [1, \dots, ROW_j], y_j \in [1, \dots, BAY_j], z_j \in [2, \dots, TIER_j], j \in [1, \dots, N_{Bloques}]$$

$$N_{c_{piso}}(k_j) \geq N_{c_{piso}}(k_j + 1) \quad \forall k = 1, \dots, TIER_j - 1, j \in [1, \dots, N_{Bloques}] \quad (4)$$

(4) Busca que la cantidad de contenedores en el piso k del bloque j debe ser necesariamente mayor o igual que la cantidad de contenedores en el piso $k + 1$ del mismo bloque j para evitar la existencia de contenedores "en el aire".

$$pos_{(x_j, y_j, z_j)}(u_{ij}) \leq pos_{(x_j, y_j, z_j - 1)}(u_{ij}) \quad (5)$$

$$\forall x_j \in [1, \dots, ROW], y_j \in [1, \dots, BAY], z_j \in [2, \dots, TIER], j \in [1, \dots, N_{Bloques}], i \in S$$

(5) De existir un contenedor i en una ubicación u del bloque j esta cerciora que bajo esta ubicación debe existir otro contenedor i dentro del mismo bloque j pero en la posición inferior. Exceptuando este caso para el primer piso del bloque j .

$$\begin{aligned} pos_{(x_j, y_j, z_j)}(p_{ij}) &\geq pos_{(x_j, y_j, z+1_j)}(p_{ij}) \\ \forall x_j \in [1, \dots, ROW_j], y_j \in [1, \dots, BAY_j], z_j \in [1, \dots, TIER_j - 1], j \in [1, \dots, N_{Bloques}] \end{aligned} \quad (6)$$

(6) El peso p del contenedor i en el bloque j necesariamente debe ser mayor o igual que el contenedor que está posicionado sobre el.

$$\begin{aligned} \sum_{z_j=1}^{TIER_j} pos_{(x_j, y_j, z_j)}(d_{ij}) &\leq 5 \\ \forall i \in S, x_j \in [1, \dots, ROW_j], y_j \in [1, \dots, BAY_j], j \in [1, \dots, N_{Bloques}], d \in D \end{aligned} \quad (7)$$

(7) Los contenedores i dentro de un bloque j dentro de un mismo tier (o pila) deben ser del mismo destino d .

$$\begin{aligned} \sum_{z_j=1}^{TIER_j} \left(\sum_{x_j=1}^{ROW_j} \left(\sum_{y_j=1}^{BAY_j} pos_{(x_j, y_j, z_j)}^d(u_{ij}) \right) \right) &\leq N_{c_{piso}}(z_j) \\ \forall i \in S, j \in [1, \dots, N_{Bloques}], d \in D \end{aligned} \quad (8)$$

(8) La suma de los contenedores i dentro del bloque j en el mismo largo y ancho deben ser menor o igual que la cantidad máxima de contenedores disponibles para ubicar en ese piso z .

$$\begin{aligned} \sum_{x_j=1}^{ROW_j} \left(\sum_{y_j=1}^{TIER_j} pos_{(x_j, y_j, z_j)}^d(u_{ij}) \right) &\leq 0 \\ pos_{(x_j, y_j, 1)}(t_{ij}) &\leq \sum_{x_j=1}^{ROW_j} \sum_{z_j=1}^{TIER_j} pos_{(x_j, y_j, z_j)}(t_{ij}) \\ \forall i \in S, j \in [1, \dots, N_{Bloques}], y_j \in [1, \dots, BAY_j], d \in D \end{aligned} \quad (9)$$

(9) Si un BAY de un ROW está vacío dentro de un bloque j , esto sentencia que el primer contenedor que llegue a esa posición determinará el tamaño de contenedores que pueden estar en las pilas de dicho ROW.

6.4.4. Algoritmo Genético

Mapa Referencial de Soluciones

Para conocer las soluciones factibles para cada contenedor contenido en el CAL de las naves, primero se genera un *mapa de ubicación referencial* de contenedores destinados para la carga y descarga, es decir, se crea un mapa de posiciones x, y, z en la que cada posición puede ser probablemente factible para que pueda recibir a algún contenedor del CAL. La Figura 18 ilustra una vista aérea del TPA en los ejes x, y . Las coordenadas coloreadas de forma diferente al color

blanco, están relacionadas a bloques válidos para recibir contenedores. Las posiciones de un mismo color poseen la misma distancia proporcional en la que está basada el modelo, a lo cual, cada posición dentro del *mapa de ubicación referencial* coloreada, es una posición existente dentro del mapa (cada color representa a un bloque de exportación diferente en posiciones reales del TPA, como muestra la Tabla 2.

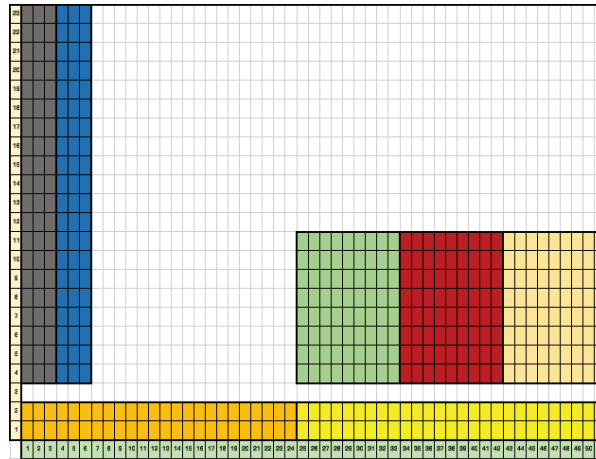


Figura 18: Matriz Referencial de Soluciones. *Fuente: Elaboración Propia*

Estructura del Algoritmo Genético

Como se muestra en la Figura 19, el algoritmo genético estará conformado por 100 individuos, los que cada uno tendrá todas las posiciones factibles del Mapa Referencial (438) además de su pila de contenedores correspondiente (5).

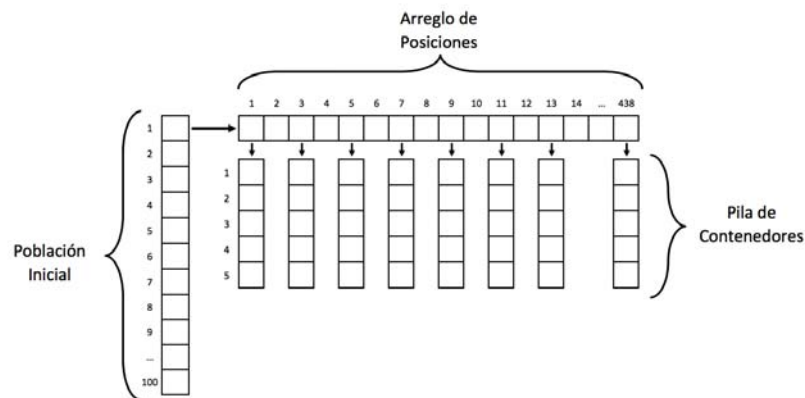


Figura 19: Estructura Solución YAP. *Fuente: Elaboración Propia*

Genotipo

En esta etapa, el algoritmo genético debe determinar en que posición debe ir ubicado cada uno de los contenedores que lleguen al terminal portuario, por lo que cada posible solución es

representada por un individuo a través de un genotipo por números enteros que representan la coordenada row y bay, cada posición row y bay tiene asociada la pila en donde se depositan los contenedores, es decir el tier.

$$G = X_1Y_1, X_2Y_2, X_3Y_3, \dots, X_{438}Y_{438}$$

Generación de la Población Inicial

Para la generación de la población inicial, estas serán generadas por el algoritmo Greedy, serán generadas acorde a las restricciones dispuestas por el TPA, serán todas las posiciones factibles existentes dentro del Mapa Referencial.

Selección

Para la selección, se utilizó el mecanismo de selección por ruleta, pero con la variante de que se mantendrá cierto porcentaje de los mejores individuos de la población pasen automáticamente a la siguiente generación (en este caso, 5%). Por ruleta todos tienen alguna posibilidad de heredar un hijo, lo que permite diversidad.

Cruzamiento

Para el cruzamiento, se utilizó PMX con 1 punto, se seleccionan dos padres al azar, los que se cruzarán generando dos nuevos hijos, sin olvidar la posibilidad de cruzamiento, en la que no asegura que se generen nuevos hijos y serán los padres quienes pasen al siguiente ciclo.

Mutación

En la mutación, cada gen de cada individuo tiene una probabilidad de ser mutado. Si un gen es seleccionado para la mutación, se toma un par de genes aleatorios y se procede a intercambiar, esto no genera conflicto de factibilidad.

Reemplazo

Para el reemplazo, se comparan los fitness de los hijos con sus padres, si el fitness de los hijos es mejor que el de los padres, serán reemplazados por sus hijos. En caso contrario, aparecerá la probabilidad de reemplazo, la que permite que los hijos a pesar de tener un fitness peor, de todas maneras puedan reemplazarlos. Esto mejora considerablemente la diversidad en la población.

6.5. Sistema de Pruebas

Dada la problemática del problema, es complicado comparar los resultados de este trabajo con otros, sin embargo, a través de un algoritmo greedy, se podrá comparar los resultados obte-

nidos con el valor del fitness de la población inicial entregados por este último.

Para la generación del algoritmo genético, se optó por programar en lenguaje Java por ser ideal para el problema que se estaba intentando solucionar, además que el autor llevaba al comienzo del proyecto, más de 2000 líneas de código, lo que hacía con mayor razón el uso de este lenguaje. El entorno de desarrollo utilizado, bajo una amplia gama de opciones se decidió por utilizar Netbeans ya que era el que más prestaciones entregaba y era más amigable en temas de usabilidad con el usuario.

Para evaluar el funcionamiento del algoritmo genético, se iteraron en varias ocasiones entre 500 y 10.000 iteraciones para poder tener resultados más representativos, estudiando en cada una de esas ocasiones se fueron modificando ciertos parámetros de entrada, principalmente las probabilidades de cruzamiento, mutación y reemplazo.

Para todas las iteraciones, se les dió la misma prioridad de 33,3 % a α , β y γ , esto significa que tanto el balanceo como las distancias e interferencias tendrán la misma relevancia.

Las gráficas posteriores tendrán el número de iteraciones en la barra horizontal y en la vertical corresponderá al valor del fitness, este valor adimensional será calculado mediante la función objetivo.

6.5.1. Prueba 1, Evaluación de la Probabilidad de Cruzamiento

Para esta prueba se evaluaron 5 instancias en la que se ejecutó el algoritmo genético en 10.000 iteraciones, en cada instancia se fue modificando la probabilidad de cruzamiento entre un 50 % a un 90 %, la probabilidad de mutación se mantuvo en un 5 % y la probabilidad de reemplazo del 100 %. La línea anaranjada muestra el resultado del algoritmo greedy para la creación de la población inicial, el resto de las ejecuciones tuvieron una población inicial randómica, en cambio en greedy, su población fue asignada a través de las reglas exigidas por el TPA.

Resultados

Como se ve en el siguiente gráfico (ver Figura 20), que a una alta probabilidad de cruzamiento afectaba negativamente el rendimiento del algoritmo. Se aprecia que mientras más bajo esta probabilidad permite obtener un mejor valor del fitness, por lo que para las siguientes pruebas fijaremos la probabilidad de cruzamiento en un 50 % ya que es mucho mayor que el greedy, el resto están alcanzando óptimos locales y esto se debe al reemplazo, porque la regla usada de los hijos mejores que los padres hace que sea muy elitista, los peores no logran avanzar a la siguiente generación, por lo que la población comienza a llenarse de óptimos locales.

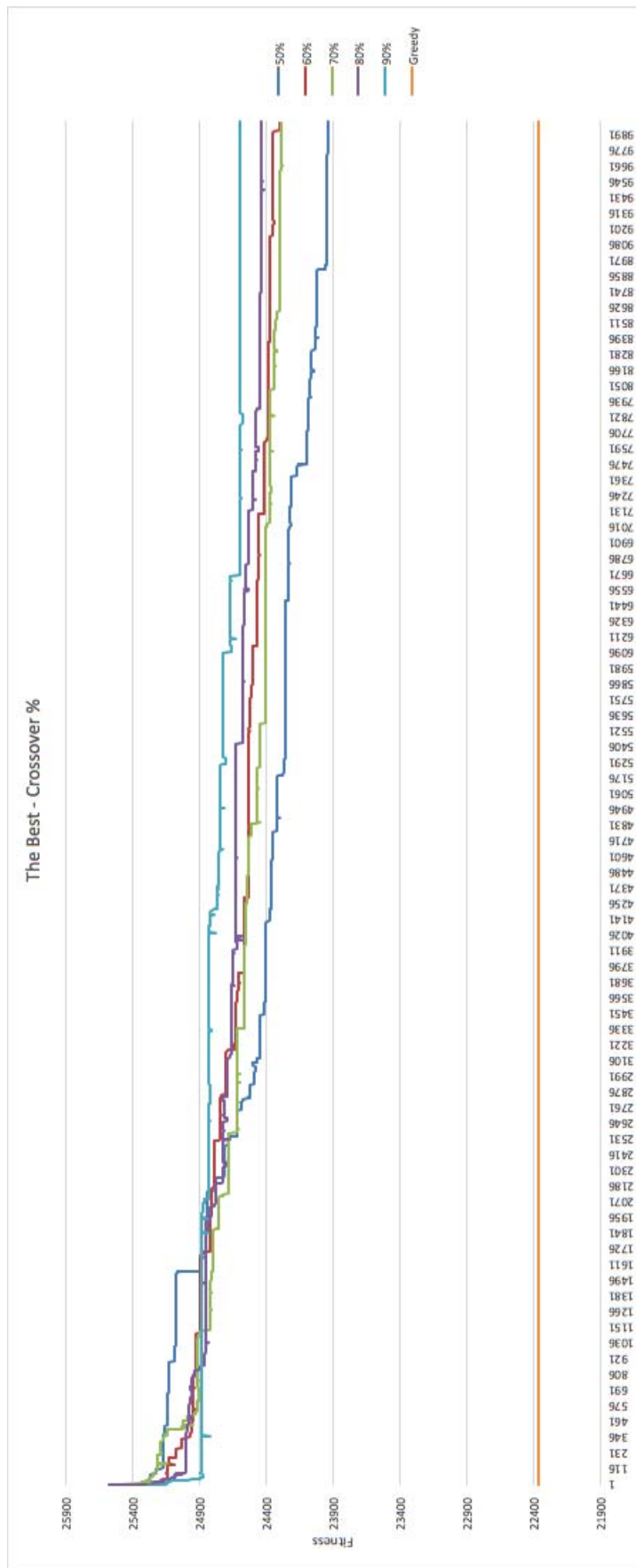


Figura 20: Resultados Prueba 1, Probabilidad de Cruzamiento. *Fuente: Elaboración Propia*48

6.5.2. Prueba 2, Evaluación de la Probabilidad de Reemplazo

Aquí se evaluaron 12 instancias en las que se ejecutaron 500 iteraciones por cada una de estas. Mencionado anteriormente, se mantuvo una probabilidad de cruzamiento del 50 %, una probabilidad de mutación constante del 5 % y en esta prueba aparece la probabilidad de reemplazo entre el 10 % y el 90 %, en la que permite que los hijos peores que el padre puedan ser parte de la nueva población, así que la población será más diversa que la de la última prueba.

Resultados

La idea de esto es que en la gráfica (ver Figura 21) no se este encaminando hacia óptimos locales muy luego, por los que tienen una pendiente muy abrupta caen directamente en esto, sin embargo, si bien se puede ver que el que obtuvo el mejor fitness fue el que tenía de parámetro un 30 % de probabilidad de reemplazo, no hay que dejar de lado otros resultados de la misma gráfica que son interesantes, como es la mejora progresiva de los resultados que tuvieron un 20 % y 25 % en la probabilidad de reemplazo, también fueron mejorando su fitness, así que esto no significa que el 30 % de probabilidad de reemplazo sea la ideal, sino que falta un estudio más recabado de estos tres y es lo que se hará posteriormente.

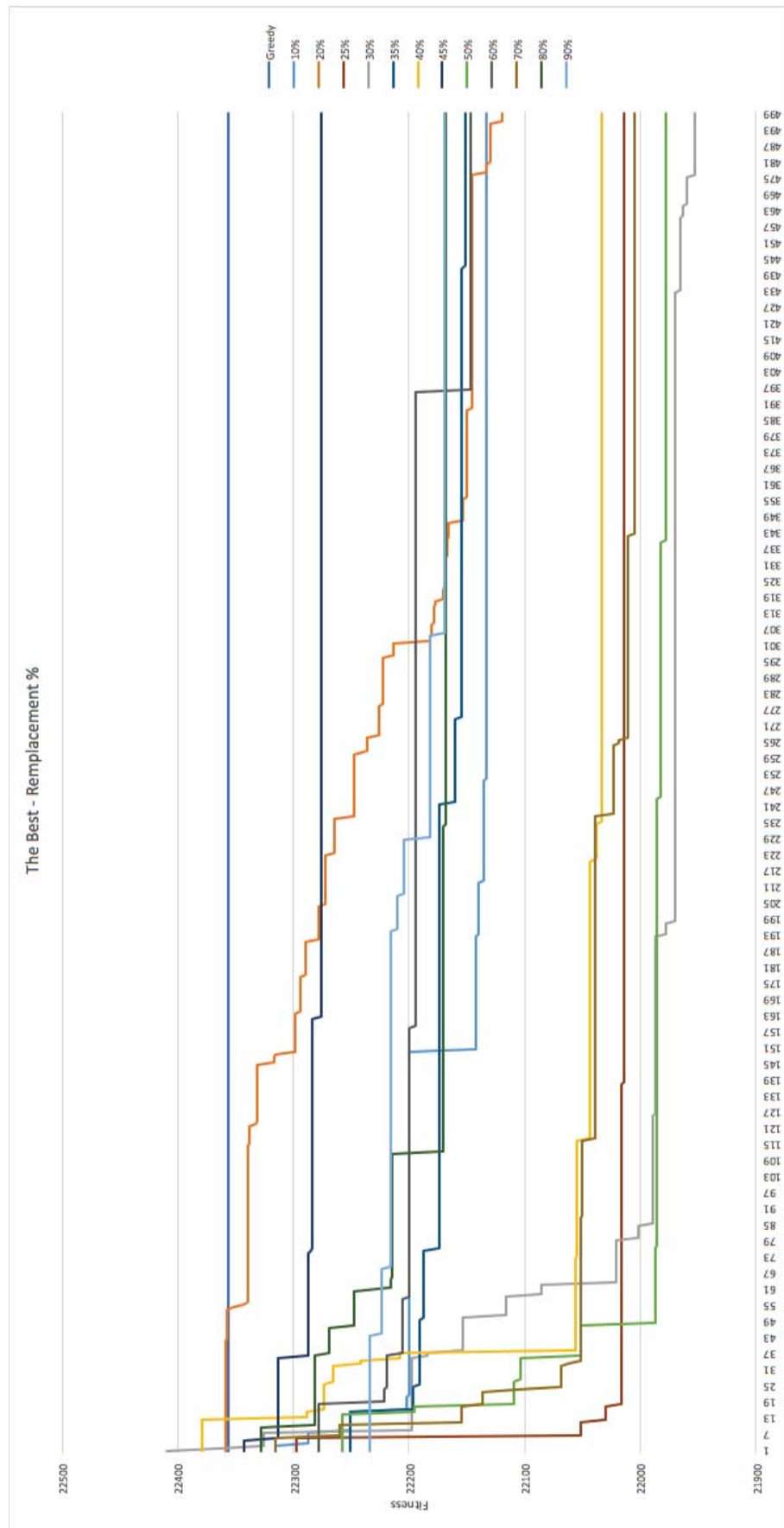


Figura 21: Resultados Prueba 2, Probabilidad de Reemplazo. Fuente: *Elaboración Propia*⁵⁰

6.5.3. Prueba 3, Evaluación de la Probabilidad de Reemplazo al 20 %

Se efectuó una instancia con una ejecución de 3.000 iteraciones, en la gráfica de la Figura 22 se aprecia el fitness del algoritmo greedy, el fitness del mejor cromosoma de la población y el fitness promedio de la población. Como parámetros de entrada, el algoritmo genético tiene un 50 % de probabilidad de cruzamiento, 5 % de probabilidad de mutación y 20 % de probabilidad de reemplazo.

Resultados

Se aprecia que a comienzos de la ejecución del algoritmo genético la población se mantiene bastante diversa, pero ya cerca de la iteración 1.700 tiende a estancarse y localizarse en un óptimo local no logrando una diversificación adecuada ya que no explora todo el mapa de soluciones.

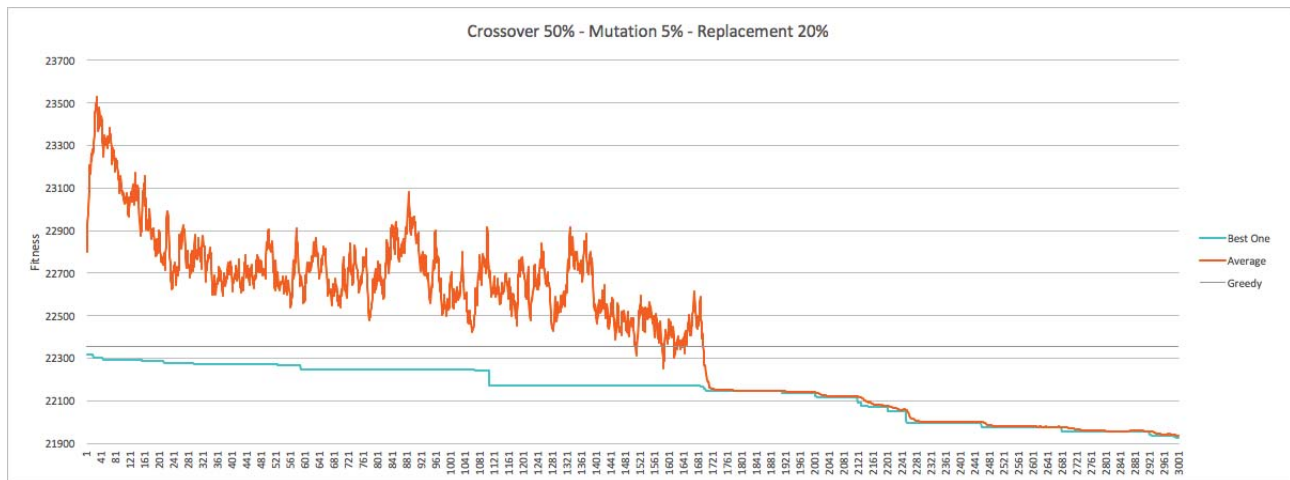


Figura 22: Resultados Prueba 3, Probabilidad de Reemplazo al 20 %. Fuente: *Elaboración Propia*

6.5.4. Prueba 4, Evaluación de la Probabilidad de Reemplazo al 25 %

Al igual que en la anterior, se efectuó una instancia con 3.000 iteraciones, en la gráfica de la Figura 23 vemos una franja que representa el fitness en el algoritmo greedy, otra franja con el promedio del fitness de la población y la tercera franja con el fitness del mejor individuo de la población. Los parámetros de entrada del algoritmo genético son un 50 % para la probabilidad de cruzamiento, un 5 % para la probabilidad de mutación y un 25 % en la probabilidad de reemplazo.

Resultados

Se ve de inmediato un óptimo local mucho antes que en la prueba 3, casi 1.000 iteraciones antes incluso, la solución se encasilla y no puede recorrer todo el mapa solución, por lo que progresaría más lentamente en encontrar una solución mejor a diferencia del resto.

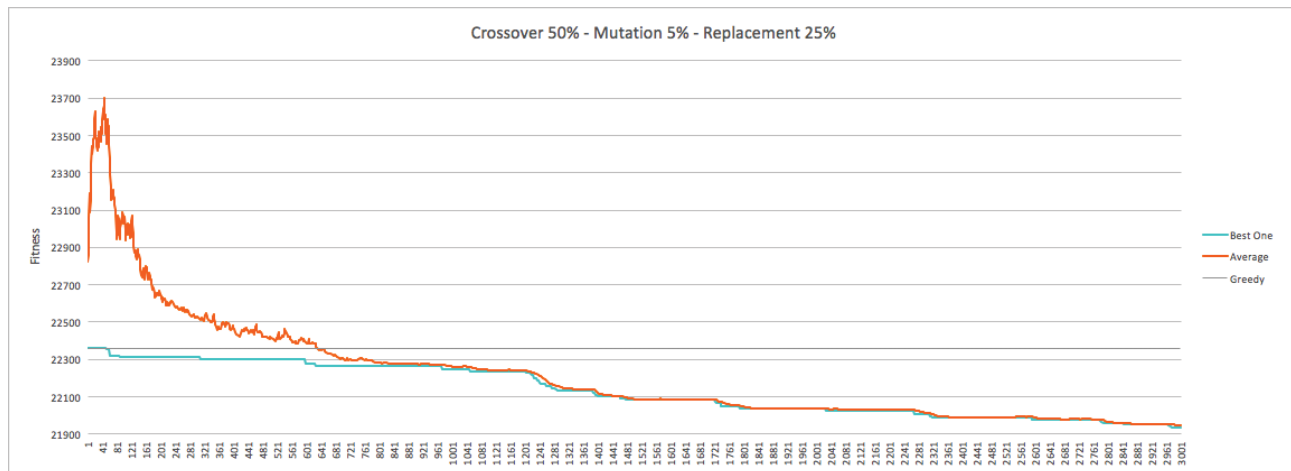


Figura 23: Resultados Prueba 4, Probabilidad de Reemplazo al 25 %. Fuente: Elaboración Propia

6.5.5. Prueba 5, Evaluación de la Probabilidad de Reemplazo al 30 %

Se efectuó una instancia con 3.000 iteraciones, en la gráfica de la Figura 24 vemos tres franjas, una con el fitness del algoritmo greedy, otra con el fitness promedio de la población del AG y otra que representa el fitness del mejor individuo en la población. Los parámetros de entrada del algoritmo genético para esta instancia fueron un 50% de probabilidad de cruzamiento, un 5% para la probabilidad de mutación y un 30% de probabilidad de reemplazo.

Resultados

Se nota una mejora considerable desde el punto de la diversidad, esto permite que se vayan explorando en gran parte el mapa de soluciones, no ha caído en algún optimo local, por lo que tiene el potencial de seguir mejorando el fitness con forme hayan más iteraciones.

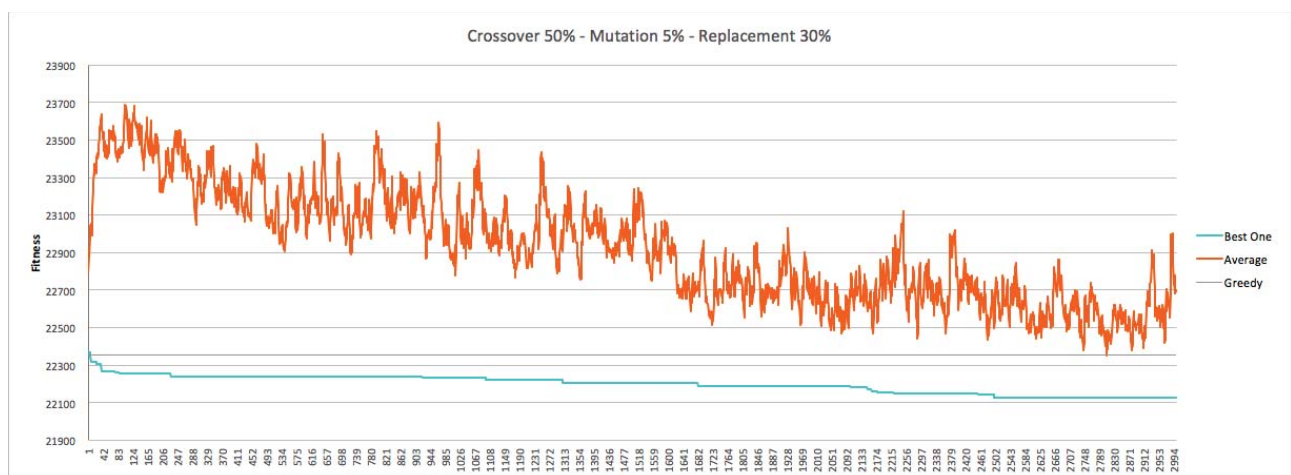


Figura 24: Resultados Prueba 5, Probabilidad de Reemplazo al 30 %. Fuente: Elaboración Propia

6.5.6. Prueba 6, Evaluación de la Probabilidad de Reemplazo en los 3 Mejores

Muestra los mejores fitness de cada uno de los individuos con una probabilidad de mutación del 50 %, probabilidad de mutación del 5 % y las probabilidad de reemplazo al 20 %, 25 % y 30 % además del fitness obtenido por el algoritmo greedy en las 3.000 iteraciones.

Resultados

Claramente se aprecia en la Figura 25 que los tres son notoriamente mejores que el fitness obtenido por el algoritmo greedy y que los que obtuvieron el mejor fitness fueron los del 20 % y 25 %, sin embargo, estos dos fueron los que en las 1.700 y 700 iteración terminan encaminados hacia óptimos locales y el de 30 % fue el que tuvo una población más diversa a lo que a parecer del autor, conviene seguir explorando la probabilidad de reemplazo del 30 % aunque tampoco dejar de lado las probabilidades del 20 % y 25 %.

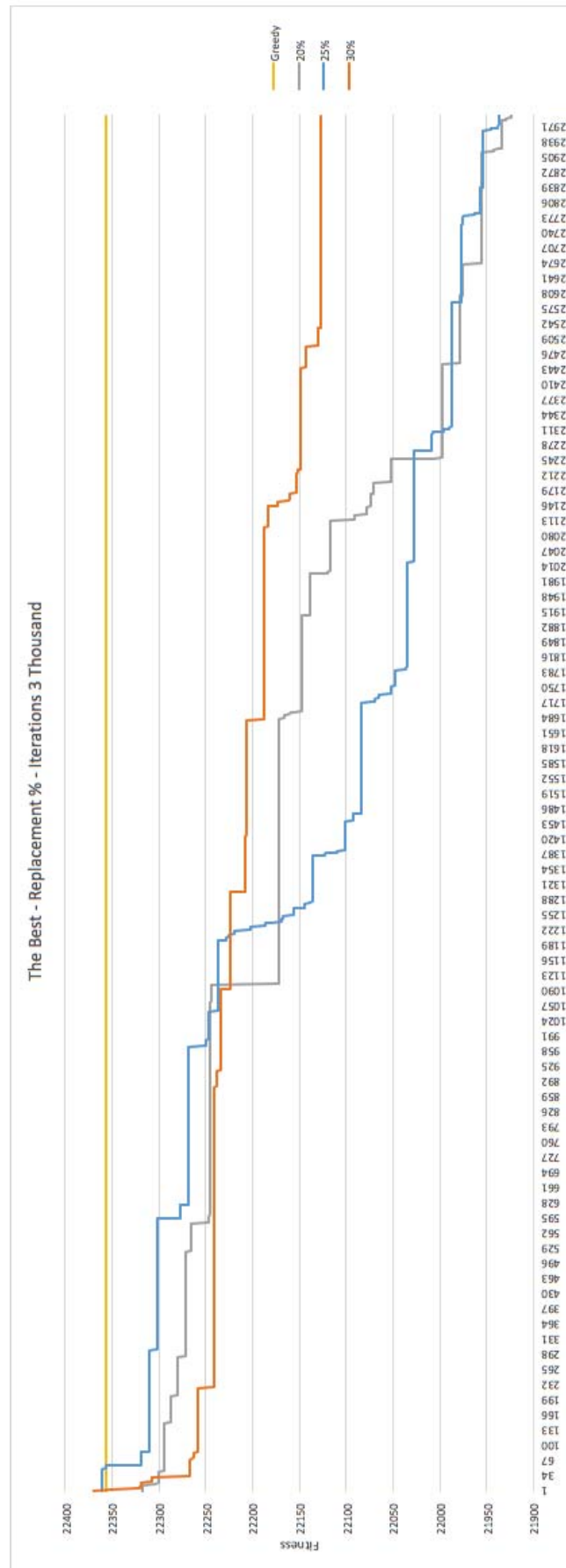


Figura 25: Resultados Prueba 6, Probabilidad de Reemplazo al 20%, 25% y 30%. Fuente: *Elaboración Propia*

Tiempos de Ejecución

	Generación [s]	Selección [s]	Cruzamiento [s]	Mutación [s]	Reemplazo [s]
Promedio	0,9009784	0,0003915	40,4727081	0,0005331	1,45025E+25
Máximo	1,095766	0,001905	53,158097	0,005049	1,45025E+25
Mínimo	0,704015	0,00005	31,539668	0,00023	1,45025E+25

Tabla 6: Tiempos de Ejecución. *Fuente: [Elaboración Propia]*

La Tabla 6 muestra los tiempos de ejecución en segundos de cada función relevante en el algoritmo genético, si bien la mayoría de las funciones tienen tiempos de ejecución razonables, no deja de llamar la atención el tiempo que se demora la función de cruzamiento. Esto se debe a que cada cromosoma al tener 438 posiciones X e Y, y cada posición X e Y posee una pila de 5 contenedores, esto hace generar un arreglo de 2.190 contenedores a intercambiar durante el cruzamiento de PMX en un punto.

Mapa Final Patio de Contenedores

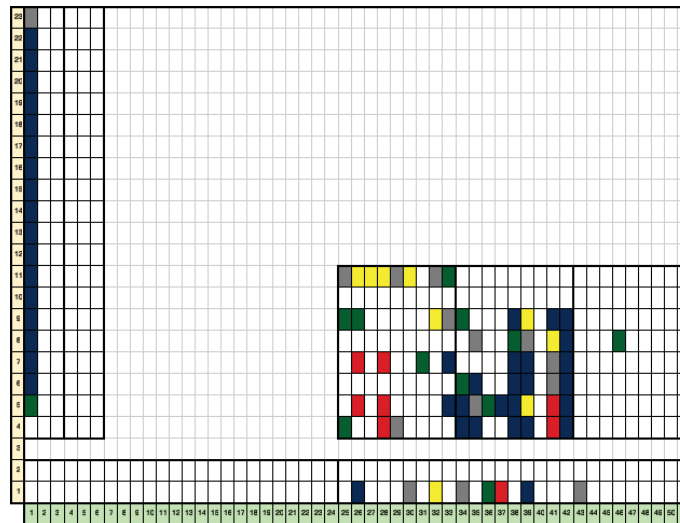


Figura 26: Patio de Contenedores Después de la Estiba. *Fuente: Elaboración Propia*

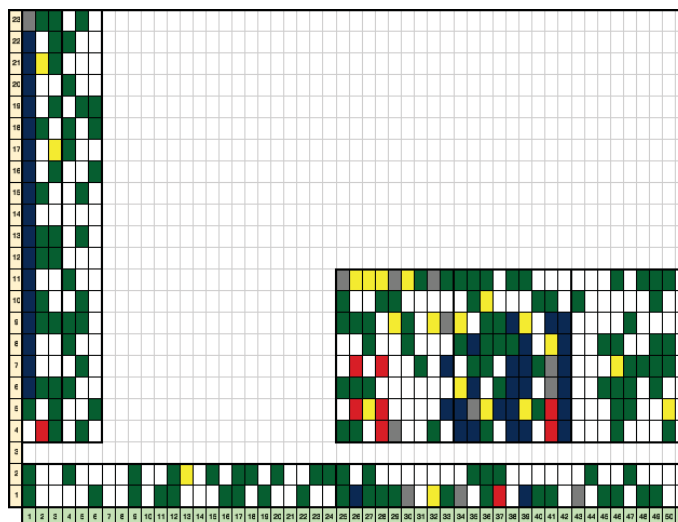


Figura 27: Patio de Contenedores con Población Insertada. *Fuente: Elaboración Propia*

Las Figuras 26 y 27 representan el Mapa Referencial de Soluciones o el patio mismo, La Figura 26 muestra la cantidad de contenedores en el patio después de que han sido retirados los contenedores de la estiba y la Figura 27 muestra los contenedores ya insertados al finalizar al algoritmo genético.

Color	Cantidad de Contenedores [u]
Blanco	0
Verde	1
Amarillo	2
Rojo	3
Plateado	4
Azul	5

Tabla 7: Segmentación por Niveles. *Fuente: [Elaboración Propia]*

Como se muestra en la tabla 7 , cada color representa cierta altura respecto a las unidades de contenedores apilados, el color blanco representa que hay ningún contenedor en dicha pila, el color verde nos dice que hay 1 contenedor en la pila, el color amarillo muestra que hay 2 contenedores la pila, el color rojo muestra que hay 3 contenedores en la pila, el color plateado muestra que hay 4 contenedores en la pila y el color azul indica que la pila tiene 5 contenedores y que ya está completa.

7. Conclusión

En el presente documento, se presentó la descripción del Yard Allocation Problem, el cual reveló la complejidad que tiene la resolución de dicho problema y los factores que son relevantes al momento de llevarse a cabo. Para el problema se planteó un algoritmo genético programado en Java, el que permitió buscar, mejorar y encontrar mejores soluciones que si hubieran sido resueltas por métodos más simples. Además se pueden entender claramente los costos en tiempo y gestión de las operaciones de un terminal portuario como el de Arica.

Como se pudo observar, se propuso una solución de dos niveles, la primera la creación de un algoritmo Greedy para generar una población acorde a las exigencias de las operaciones de asignación de contenedores del TPA, y la segunda un algoritmo genético que pudiera encontrar el mejor ordenamiento de estos contenedores con la finalidad de minimizar sus costos operacionales tanto en mover un contenedor, priorizar despejes y maximizar el uso de su maquinaria en bloques.

Se hicieron instancias considerables de prueba para encontrar los parámetros correctos que hicieran llegar con la mejor solución posible. Se pudo apreciar la efectividad del algoritmo genético.

Aplicar este tipo de algoritmo en la vida real es complejo, debido a que coexisten muchos factores externos que no fueron involucrados en este modelo de solución, pero el algoritmo Greedy de planificación puede ser de utilidad para explotarlo en trabajos futuros. Los algoritmos para este tipo de problemáticas debe acomodarse a cualquier tipo de evento que pudiera entorpecer las operaciones portuarias.

Referencias

- [Arica, 2015] Arica, P. (2015). Terminal Portuario de Arica. [Web; accedido el 16-12-2015].
- [Ayachi et al., 2013] Ayachi, I., Kammarti, R., Ksouri, M., Borne, P., et al. (2013). A genetic algorithm to solve the container storage space allocation problem. *arXiv preprint arXiv:1303.1051*.
- [Baker, 1987] Baker, J. E. (1987). Reducing bias and inefficiency in the selection algorithm. In *Proceedings of the second international conference on genetic algorithms*, pages 14–21.
- [Bazzazi et al., 2009] Bazzazi, M., Safaei, N., and Javadian, N. (2009). A genetic algorithm to solve the storage space allocation problem in a container terminal. *Computers & Industrial Engineering*, 56(1):44–52.
- [Burke and Kendall, 2014] Burke, E. K. and Kendall, G. (2014). *Search methodologies*. Springer.
- [Covarrubias and Tapia, 2012] Covarrubias, R. and Tapia, F. (2012). Desarrollo, Implementación y Aplicación de un Enfoque Multi-Objetivo para la Planificación de Operaciones de un Patio de Contenedores. Memoria de Título.
- [Elorrieta, 2011] Elorrieta, J. (2011). Planificación y Gestión de Operaciones Portuarias en Patio de Contenedores. Memoria de Título.
- [Gobierno de Chile, 2009] Gobierno de Chile, Ministerio de Obras Públicas, P. M. S. (2009). Infraestructura Portuaria y Costera Chile 2020. Technical report.
- [Goldberg, 1989] Goldberg, D. (1989). *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison Wesley, Reading, Massachusetts.
- [Holland, 1975] Holland, J. H. (1975). *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence*. U Michigan Press.
- [Kaps, 2012] Kaps, H. (2012). *Container Handbook*. <http://goo.gl/qxn7oz>. Gesamtverband der Deutschen Versicherungswirtschaft e.V. Referenciado el 19 de abril del 2015.
- [Maturana, 2014] Maturana, K. (2014). Sistema Multiagente para el Problema de Asignación de Patio. Memoria de Título.
- [O’Shee, 2013] O’Shee, C. (2013). Desarrollo de Algoritmos Genéticos para la Asignación de Espacios en un Patio de Contenedores. Memoria de Título.
- [Sivanandam and Deepa, 2007] Sivanandam, S. and Deepa, S. (2007). *Introduction to genetic algorithms*. Springer Science & Business Media.