Modelos y algoritmos para el problema de la asignación de atraques y grúas en las terminales de contenedores

Juan F. Correcher*, Ramon Alvarez-Valdes, Jose M. Tamarit y Alexei Lescaylle

Departamento de Estadística e Investigación Operativa
Universidad de Valencia

Resumen En este estudio proponemos un nuevo modelo de programación matemática lineal entero mixto para el problema de la asignación de atraques y grúas a los barcos que solicitan atracar en una terminal de contenedores. Consideramos un muelle continuo, llegada de buques dinámica, asignación de grúas invariable en el tiempo y tiempo de procesamiento de los barcos dependiente del número de grúas asignadas a cada uno. Asimismo, a fin de obtener buenas soluciones en aquellos problemas cuyo tamaño nos impide alcanzar una solución óptima mediante el modelo, proponemos un algoritmo genético de claves aleatorias sesgado. La eficacia de ambos enfoques es evaluada en diversos experimentos computacionales sobre distintos conjuntos de problemas generados aleatoriamente. Según los resultados, nuestro modelo permite alcanzar el óptimo en problemas de hasta 40 buques en muy poco tiempo, mientras que el genético consigue buenas soluciones en problemas de cualquier tamaño.

Palabras clave: Berth Allocation Problem, Quay Crane Assignment Problem, Mixed Integer Linear Model, Biased Random-key Genetic Algorithm, optimización en terminales de contenedores.

1. Introducción

Los puertos marítimos son un factor clave en la economía de mercado mundial. Una parte considerable de las mercancías son transportadas en contenedores estandarizados que pueden ser gestionados por instalaciones especializadas presentes en la mayoría de los puertos importantes, conocidas como *terminales de contenedores*. En estas se suelen identificar tres áreas: la línea de atraque, el patio de contenedores y la zona de intercambio terrestre. En la primera zona, los buques portacontenedores son atracados en el muelle y la carga y descarga es realizada por las grúas del muelle. Desde este punto, unos vehículos especializados transportan los contenedores hasta la zona del patio de contenedores, donde se apilan temporalmente. Por último, la tercera zona es el lugar donde los contenedores son recogidos por los operadores terrestres, que los transportarán por camión o ferrocarril hasta su destinatario.

* Autor de contacto. Correo-e: juan.correcher@uv.es

En la línea de atraque se han identificado distintos problemas de suma importancia para el rendimiento de una terminal. La revisión más reciente de los enfoques propuestos para afrontarlos es la realizada por [1,**?**]. Uno de estos problemas es el de asignar un tiempo y una posición de atraque a los buques que solicitan atracar en la terminal, conocido como *Berth Allocation Problem* (BAP). El objetivo es minimizar el coste total de asignación, para lo que se consideran los datos relativos a cada barco (longitud, tiempo estimado de llegada y tiempo deseado de salida) y diferentes tipos de coste, como el coste por obligar al barco a esperar hasta recibir una posición de atraque, el coste por retrasar la salida del buque o el coste que supone atracar un buque lejos de la zona del patio reservada para sus contenedores. Según los aspectos considerados, habitualmente se distinguen diferentes versiones del problema, entre las que destacan las relativas al tipo de muelle considerado (discreto, continuo o híbrido); las relativas al tiempo de llegada de los buques (dinámico, estático, etc.); o las relativas a los distintos costes (espera, retraso, desviación respecto a la posición deseada, etc.).

Un problema muy relacionado es el *Quay Crane Assignment Problem* (QCAP), en el que se busca la mejor asignación de grúas para los barcos dado un plan de atraque viable y la carga de trabajo requerida por cada barco. Según el modo de asignar las grúas, se identifican dos versiones: la que admite un número de grúas variable durante el tiempo de proceso del barco (*variable-in-time*) y la que mantiene el número de grúas fijo (*time-invariant*). Dado que esto puede influir notablemente en el tiempo de procesamiento de los buques, el QCAP se suele abordar junto con el BAP, buscando de esta forma una mejor planificación del atraque.

Hay numerosos enfoques que afrontan la integración entre diferentes versiones del BAP y el QCAP, conocidos como *Berth Allocation and quay Crane Assignment Problems* (BACAP). Siguiendo a [6], emplearemos el acrónimo BACAP para referirnos a la versión integrada en la que únicamente se determina el número de grúas asignadas a cada barco, mientras que BACASP hará referencia a la versión donde también se determina qué grúas en particular son asignadas a cada barco.

El primer estudio de integración fue presentado por [5]. En él se propone un modelo lineal entero para el BACAP continuo de asignación variable en el tiempo. Para obtener soluciones cercanas al óptimo desarrollan una relajación lagrangiana del modelo y emplean el método del subgradiente. Otro trabajo importante es el de [4], que aborda un BACAP de muelle continuo y asignación de grúas variable en el tiempo. Una de las novedades es la consideración del descenso de la productividad marginal de las grúas conforme aumenta el número de grúas asignadas a un mismo barco, fruto de las interferencias que se provocan entre ellas. Los autores proponen un modelo lineal entero mixto (MILP), implementado en CPLEX, y dos heurísticas.

El trabajo más reciente que aborda esta versión del problema es el de [6]. Los autores proponen un MILP para un BACAP continuo con asignación de grúas invariable en el tiempo. La productividad de las grúas puede ser tenida en cuenta debido a que se recibe como dato del problema una estimación del tiempo de procesamiento de cada barco por cada número

de grúas admitido. Los autores extienden este modelo para afrontar un BACASP y adicionalmente proponen un algoritmo de planos de corte.
El resto de este artículo está organizado como sigue. En el Apartado 2 describimos el problema y los presupuestos de nuestro enfoque. En el Apartado 3, proponemos un nuevo modelo entero para el BACAP. A continuación, en el Apartado 4 explicamos nuestro algoritmo genético basado en claves aleatorias. En el Apartado 5 describimos los experimentos realizados y valoramos los resultados. Finalmente, en el Apartado 6 presentamos las conclusiones y posibles líneas futuras de investigación.

2. Descripción del problema

El problema de la asignación de atraques y grúas (BACAP) es el problema de optimización consistente en obtener el plan de atraque de mínimo coste considerando el tiempo de llegada de los buques y el número de grúas disponible para servirlos, entre otros datos. En nuestro caso, abordamos la versión del problema en la hay un muelle continuo y la asignación de las grúas es invariable en el tiempo, es decir, una vez se ha asignado un número de grúas a un barco, este número no puede cambiar durante su tiempo de procesamiento. El plan de atraque puede ser representado como un esquema espacio-tiempo en el que el eje vertical representa las posiciones de atraque en el muelle, y el eje horizontal, el tiempo. Cada barco es representado como un rectángulo cuya base es el tiempo de procesamiento y cuya altura es su longitud. Los barcos pueden ser atracados a lo largo del muelle, mientras que las grúas pueden moverse a lo largo del mismo para servir a los barcos, siempre y cuando no se crucen entre ellas.
El objetivo es asignar una posición y tiempo de atraque y un número de grúas a cada barco minimizando el coste total de asignación. Por cada barco consideramos tres tipos distintos de coste: el coste de espera del barco para atracar, el coste de retraso en su salida y el coste de desviación con respecto a su posición deseada en el muelle. Para ello, conocemos:

- Conjunto de períodos de tiempo: $T = \{1, 2, ..., H\}$, siendo H el horizonte temporal de planificación.
- Longitud del muelle: L.
- Número de grúas en el muelle: G.
- Conjunto de barcos: V; y $N = |V|$, esto es, el número de barcos.
  Para cada buque i, conocemos:
  - longitud (eslora): l_i
  - tiempo de llegada estimado: a_i
  - tiempo de salida deseado: s_i
  - número máximo de grúas asignable al barco: g_i^{max}
  - número mínimo de grúas asignable al barco: g_i^{min}
  - tiempo de procesamiento estimado considerando g grúas: u_i^g
  - posición deseada en el muelle: d_i
  - coste de espera por cada período de tiempo más allá del tiempo estimado de llegada: C_i^w
  - coste de retraso por cada período de tiempo más allá del tiempo de salida deseado: C_i^d

- coste por cada unidad de longitud de desviación respecto a la posición deseada en el muelle: C_i^p

El tiempo de espera de un barco es definido como la diferencia entre su tiempo de atraque real y su tiempo estimado de llegada. El tiempo de retraso es la diferencia entre el tiempo de salida real y el tiempo de salida deseado. Asimismo, cada barco tiene una posición deseada en el muelle, que es la posición más cercana a los contenedores objetivo situados en el patio de contenedores. Por tanto, la desviación respecto de la posición deseada es el número de unidades de longitud entre la posición ideal y la real. Nótese también que podemos tener en cuenta implícitamente la variación de la productividad de las grúas especificando los tiempos de procesamiento estimados (u_i^g) según el número de grúas que se le pueden asignar al barco.

Los presupuestos de este enfoque son los siguientes:

- El tiempo de planificación está dividido en segmentos de tiempo iguales y el horizonte temporal es el último de ellos.
- Una vez un barco ha sido atracado, no puede cambiarse su posición.
- Una vez ha comenzado el procesamiento de un barco, este no puede ser interrumpido.
- El tiempo de procesamiento de cada barco es considerado independiente de su posición de atraque.
- El número de grúas disponible en el muelle es fijo y todas las grúas tienen las mismas características.
- Todas las grúas pueden moverse a lo largo de todo el muelle, pero no pueden cruzarse entre ellas.
- Cada grúa solo puede ser asignada a un barco a lo sumo en cada período de tiempo.
- Cada barco puede tener diferente importancia relativa, lo cual se reflejaría en los distintos costes.

3. Un modelo lineal entero mixto

Proponemos el siguiente modelo lineal entero mixto de programación matemática inspirado en los trabajos de [4] y [6].

Definimos las siguientes variables:

- t_i = tiempo de atraque del buque i;
- p_i = posición de atraque del buque i;
- $\sigma_{ij} = 1$, si el buque i está completamente a la izquierda del buque j, esto es, si el buque i es servido completamente antes que el buque j; 0, en otro caso;
- $\delta_{ij} = 1$, si el buque i está completamente bajo el buque j, esto es, si el buque i está atracado completamente a la derecha del buque j (mirando los buques desde el muelle); 0, en otro caso;
- h_i = retraso de salida del buque i;
- e_i = desviación del buque i respecto de su posición deseada;
- $r_{igt} = 1$, si el procesamiento del buque i con g grúas empieza en el período t; 0, en otro caso.

A partir de estas variables y parámetros proponemos siguiente modelo:

$$Min \sum_{i \in V} (C_i^w (t_i - a_i) + C_i^d h_i + C_i^p e_i) \quad (1)$$

sujeto a:

$$t_j - t_i - \sum_{t=a_i}^{H} \sum_{g=g_i^{min}}^{g_i^{max}} (r_{igt} \cdot u_i^g) - (\sigma_{ij} - 1)H \geq 0, \quad \forall i,j \in V, i \neq j \ (2)$$

$$p_j - (p_i + l_i) - (\delta_{ij} - 1)L \geq 0, \quad \forall i,j \in V, i \neq j \ (3)$$

$$\sigma_{ij} + \sigma_{ji} + \delta_{ij} + \delta_{ji} \geq 1, \quad \forall i,j \in V, i \neq j \ (4)$$

$$p_i + l_i \leq L + 1, \quad \forall i \in V \ (5)$$

$$t_i = \sum_{t=a_i}^{H} \sum_{g=g_i^{min}}^{g_i^{max}} r_{igt} \cdot t, \quad \forall i \in V \ (6)$$

$$h_i \geq t_i - s_i + \sum_{t=a_i}^{H} \sum_{g=g_i^{min}}^{g_i^{max}} (r_{igt} \cdot u_i^g) - 1, \quad \forall i \in V \ (7)$$

$$e_i \geq p_i - d_i, \quad \forall i \in V \ (8)$$

$$e_i \geq d_i - p_i, \quad \forall i \in V \ (9)$$

$$\sum_{t=a_i}^{H} \sum_{g=g_i^{min}}^{g_i^{max}} r_{igt} = 1, \quad \forall i \in V \ (10)$$

$$\sum_{i \in V} \sum_{g=g_i^{min}}^{g_i^{max}} \sum_{\tau=max(a_i, t-u_i^g+1)}^{t} r_{ig\tau} \cdot g \leq G, \quad \forall t \in T \ (11)$$

$$\sigma_{ij}, \delta_{ij}, \in \{0,1\}, \quad \forall i,j \in V, i \neq j \ (12)$$

$$r_{igt} \in \{0,1\}, \quad \forall i \in V, \forall t \in \{a_i, \ldots, H\}, \forall g \in \{g_i^{min}, \ldots, g_i^{max}\} \ (13)$$

$$h_i, e_i \geq 0, \quad \forall i \in V \ (14)$$

$$p_i \geq 1, \quad \forall i \in V \ (15)$$

La función objetivo (1) es la minimización del coste total de planificación, que consiste en la suma del coste de espera, el coste de retraso y el coste de desviación respecto de la posición deseada para cada barco. Las restricciones (2–4) impiden solapamientos entre las representaciones de los barcos en tiempo y espacio. Las restricciones (5) determinan las posiciones válidas en el muelle para cada buque. Las restricciones (6) vinculan las variables t_i a las variables r_{igt} para impedir inconsistencias. Las restricciones (7) definen el retraso de cada barco y las (8-9) la desviación respecto de la posición deseada. El procesamiento de cada barco comienza una sola vez y con un número determinado de grúas (10). Más aún, en cada período de tiempo el número de grúas asignadas a los barcos no puede exceder el número total de grúas existentes en el muelle (11). Finalmente, las restricciones (12–15) definen los dominios de las

variables. Nótese en (15) que la primera posición disponible en el muelle es 1 y no 0.
Este modelo es evaluado en diferentes experimentos computacionales sobre problemas generados aleatoriamente. En el Apartado 5 describiremos dichos experimentos y comentaremos los resultados.

4. Un algoritmo genético de claves aleatorias sesgado

El modelo propuesto es efectivo para encontrar una solución óptima en problemas con un número de barcos reducido, pero para obtener buenas soluciones en problemas de mayor tamaño debemos recurrir a heurísticas. Así pues, proponemos un algoritmo genético de claves aleatorias sesgado (*Biased Random-Key Genetic Algorithm*).
Este tipo de algoritmos genéticos funcionan asignando una clave aleatoria (un número entre 0 y 1) a cada gen, de modo que la secuencia de elementos del problema que representa el cromosoma quedaría conformada por los genes reordenados de menor a mayor clave. Las claves son especialmente relevantes en cuanto al cruzamiento, pues este consiste en decidir aleatoriamente, para cada gen, cuál es el que pasará a formar parte del cromosoma hijo de entre los correspondientes genes de los padres. Esta selección se suele realizar con un sesgo en favor de uno de los padres, de ahí el calificativo de *sesgado*. Una vez se han establecido todos los genes del cromosoma hijo, la secuencia correspondiente de elementos del problema se determina reordenando de nuevo los genes según las claves que traían asociadas. El esquema de este tipo de algoritmo genético que seguimos aquí está inspirado en el trabajo de [3].
En nuestro caso, los cromosomas están formados por dos listas de tantos elementos como buques hay en el problema. La primera lista está formada por claves aleatorias, de forma que, por ejemplo, la clave situada en la primera posición (el primer gen) será la correspondiente al buque con identificador 1. A partir de esta lista, reordenando las claves de menor a mayor, se obtiene una secuencia de barcos que servirá para construir una solución. La segunda lista determina el número de grúas asignado a cada barco, de modo que, por ejemplo, el número existente en la primera posición corresponderá al número de grúas asignadas al barco con identificador 1 en el problema. Este número, obviamente, debe cumplir la condición de ser un número de grúas válido para tal barco. Estas dos listas son utilizadas por un algoritmo heurístico constructivo de gran rapidez que crea una solución insertando los barcos de uno en uno en el plan de atraque. La función de fitness del algoritmo genético será la inversa de la función objetivo, pues se trata de un problema de minimización.
El cruzamiento sobre la lista de claves es el indicado arriba, aplicando un sesgo $CProb \in [0, 1]$ en favor del primer cromosoma padre. Sobre la lista de grúas se aplica el mismo proceso, incluido el sesgo, pero con un nuevo número aleatorio por cada gen. En cuanto a la mutación, incluimos nuevos individuos generados aleatoriamente.

La población inicial es de N_{indiv} individuos generados aleatoriamente. A continuación, se construye una solución por cada individuo, según su cromosoma, aplicando el algoritmo constructivo que indicamos más adelante. Se reordenan los individuos según su fitness y se marcan los $E\,\%$ mejores como la élite. La siguiente generación estará formada por esta élite, otros $100\,\% - E\,\% - M\,\%$ individuos fruto de cruzar un individuo aleatorio de la élite (que será favorecido por el sesgo) con uno aleatorio del resto de la población, y $M\,\%$ nuevos individuos generados aleatoriamente (inmigración). Este proceso se repite hasta alcanzar un número $GensNoImprov$ de generaciones sin haber conseguido una solución mejor que la alcanzada hasta el momento.
El algoritmo constructivo utilizado parte de una lista ordenada de los buques y los toma de uno en uno. El orden de la lista se obtiene ordenando de menor a mayor las claves de la primera lista del cromosoma. El número de grúas asignado a cada barco queda determinado por la segunda lista del cromosoma. Este algoritmo trata de encontrar la mejor asignación de un buque construyendo y explorando un conjunto de asignaciones candidatas ordenado por coste no decreciente. Los detalles de este procedimiento pueden consultarse en nuestro trabajo previo [2].

5. Experimentos computacionales

En este apartado, en primer lugar evaluamos el modelo propuesto. Para ello lo comparamos con el modelo desarrollado recientemente por [6] para la misma versión del problema. Ambos modelos fueron implementados en C++ y CPLEX 12.6 y ejecutados en un computador Intel Core i7 2600 a 3,4 GHz con 15,7 GiB de RAM sobre diferentes conjuntos de problemas generados según ciertos criterios. Seguidamente, también aplicamos nuestro algoritmo genético, implementado en C++ y OpenMP (multihilo), sobre los mismos conjuntos de problemas y lo comparamos con nuestro modelo.
Hemos generado dos conjuntos de problemas basados en los criterios aplicados por [4]. La única diferencia entre ambos es que en el primero (*GenMB-10m*) consideramos las longitudes del muelle y de los barcos discretizadas en unidades que representan 10 m cada una, mientras que en el segundo (*GenMB-50m*) estas representan 50 m. Sin embargo, estos conjuntos se generan aleatoriamente por separado. Cada uno consta de 50 problemas, 10 por cada número de buques considerado: $N \in \{20, 30, 40, 50, 60\}$. La unidad temporal es 1 h, el horizonte de planificación es de 210 h, el muelle es de 1000 m y hay 10 grúas. Los buques pueden ser de tres tipos: Feeder, Medium y Jumbo. En cada problema, el 60 %, 30 % y 10 % de los barcos corresponde a cada clase, respectivamente. Los tiempos de llegada de los barcos se generan aleatoriamente mediante una distribución uniforme $U[0, 168]$. La posición deseada en el muelle para un barco i es generada mediante $U[1, L + 1 - l_i]$, redondeando hacia el entero mayor más cercano. El tiempo de procesamiento de cada buque resulta de dividir su carga de trabajo en horas-grúa por cada número de grúas admisible para dicho barco, redondeando hacia el entero mayor más cercano. El tiempo de salida deseado de cada barco

i es $1{,}5 \cdot min(u_i^g)$. El resto de datos se generan según las distribuciones indicadas en la Tabla 3 del citado artículo. Finalmente, los costes son los mismos para todos los barcos: $C^w = 1000$, $C^d = 2000$, y $C^p = 200$ en el conjunto *GenMB-10m*; y $C^w = 1000$, $C^d = 2000$, y $C^p = 1000$ en *GenMB-50m*.

5.1. Evaluación del modelo

Hemos llevado a cabo distintos experimentos para evaluar el modelo, uno por cada conjunto de problemas. Además, hemos hecho esto mismo con el modelo propuesto por [6]. Para cada problema, se estableció un tiempo límite de 1 hora y se registró el tiempo invertido y la distancia (*gap*) entre la mejor cota inferior y la mejor cota superior en el tiempo límite. Los resultados para cada conjunto se muestran en sendos cuadros. El Cuadro 1 muestra, para cada grupo de problemas en *GenMB-10m* con el mismo número de buques, el número de problemas para los que se ha alcanzado el óptimo, el tiempo de computación medio, la distancia media en porcentaje y la distancia máxima en porcentaje. El Cuadro 2 muestra los resultados sobre *GenMB-50m*.

Cuadro 1: Comparación del modelo con el de Türkoğulları et al. sobre *GenMB-10m*.

Modelo	#Barcos	#óptimos	T. medio (s)	Dist. media (%)	Dist. máx. (%)
Modelo de Türkogullari	20	10	270.7	0	0
	30	10	679.3	0	0
	40	3	3575.1	59.21	100
	50	0	4379.2	98.97	100
	60	-	-	-	-
Nuestro modelo	20	10	0.6	0	0
	30	10	5.5	0	0
	40	10	13	0	0
	50	6	2252.2	2.01	6.71
	60	0	3603.9	18.56	40.02

Los cuadros muestran que nuestro modelo obtiene mejores resultados que el modelo presentado por [6] en el conjunto de problemas *GenMB-10m* (Cuadro 1). Podemos resolver óptimamente problemas de hasta 40 barcos en muy poco tiempo y muchos problemas de 50 barcos. Además, las distancias obtenidas en los casos donde no se consigue el óptimo en el tiempo límite son muy reducidas. Cabe señalar que el tiempo invertido por el modelo de [6] excede frecuentemente el tiempo límite cuando resuelve problemas de más de 40 buques. La razón es que la fase de pre-procesamiento realizada por CPLEX no considera dicho tiempo límite,

Cuadro 2: Comparación del modelo con el de Türkoğulları et al. sobre *GenMB-50m*.

Modelo	#Barcos	#óptimos	T. medio (s)	Dist. media (%)	Dist. máx. (%)
Modelo de	20	10	11.6	0	0
Türkogullari	30	10	21.5	0	0
	40	10	48.8	0	0
	50	6	1507	4.56	16.01
	60	2	3032.5	21.98	84.67
Nuestro	20	10	0.7	0	0
modelo	30	10	1.5	0	0
	40	10	24.8	0	0
	50	4	2388.5	8.37	26.39
	60	0	3602.7	29.17	51.83

por lo que no se detiene hasta que empieza la fase de ramificación y corte y detecta que este límite ha sido superado. En los problemas de 60 barcos, su modelo no consigue ninguna solución debido a que CPLEX no es capaz de gestionarlo.

Con respecto del conjunto de problemas *GenMB-50m*, nuestro modelo no consigue mejorar los resultados del modelo de [6], pero se acerca a su distancia media y, para los problemas de 60 barcos, mejora incluso la distancia máxima. Su modelo consigue varios óptimos en problemas de 50 y 60 barcos. El rendimiento de nuestro modelo sobre este conjunto de problemas es similar al alcanzado sobre el conjunto de problemas anterior. El diferente rendimiento observable en el modelo de [6] entre estos dos conjuntos puede explicarse por su fuerte dependencia del factor de discretización aplicado a las longitudes del muelle y de los buques. Ellos consideran, para cada barco, una variable distinta por cada posición disponible en el muelle, así que, conforme la discretización es más fina, el número de variables de posición consideradas aumenta geométricamente. Esto también explica las limitaciones de memoria que enfrentamos cuando aplicamos este modelo.

Todo esto nos lleva a concluir que nuestro modelo parece una buena propuesta independientemente del factor de discretización aplicado a las longitudes, pues este no provoca un aumento del número de variables. Nuestras variables de posición son continuas.

5.2. Evaluación del algoritmo genético

Hemos aplicado el algoritmo genético sobre los dos conjuntos de problemas a fin de evaluar preliminarmente su rendimiento. Para ello, hemos establecido un tamaño poblacional de $N_{indiv} = 5N$ dependiente del número de buques; $CProb = 0{,}7$; $E = 20\,\%$; $M = 15\,\%$; y $GensNoImprov = 10000$. Los resultados se muestran en los Cuadros 3 y 4. La columna de distancia media indica la distancia media entre el valor de la mejor solución alcanzada por el genético y el de la mejor solución alcanzada por el modelo en 1 h, tomando esta última como referencia. La columna de distancia máxima indica la distancia máxima en cada grupo de problemas.

Cuadro 3: Comparación del modelo y el algoritmo genético sobre *GenMB-10m*.

Número de barcos	T. medio modelo (s)	T. medio genético (s)	Distancia media modelo-genético (%)	Distancia máxima modelo-genético (%)
20	0.6	13.1	5.31	40.48
30	5.5	39.6	4.65	12.14
40	13	79.4	7.29	15.38
50	2252.2	134.4	11.25	20.99
60	3603.9	303.3	21.56	47.33

Cuadro 4: Comparación del modelo y el algoritmo genético sobre *GenMB-50m*.

Número de barcos	T. medio modelo (s)	T. medio genético (s)	Distancia media modelo-genético (%)	Distancia máxima modelo-genético (%)
20	0.7	13.7	0.78	5.88
30	1.5	31.2	2.92	15.15
40	24.8	76	5.34	10.75
50	2388.5	140.4	9.15	21.47
60	3602.7	306.8	10.14	28.19

Los cuadros muestran que las soluciones alcanzadas por el genético son peores que las del modelo, aunque este último invierte mucho más tiempo conforme aumenta el número de barcos en el problema. Así, una de las ventajas del genético es que obtiene soluciones aceptables en muy poco tiempo. Además, la distancia media de las soluciones obtenidas por el genético no supera el 22 % desde la mejor solución obtenida por el modelo.

6. Conclusiones

En conclusión, hemos propuesto un nuevo modelo lineal entero mixto para el problema de la asignación de atraques y grúas a los barcos en

una terminal de contenedores. Concretamente, abordamos la versión del problema con muelle continuo y asignación de grúas invariable en el tiempo. Además, proponemos un algoritmo genético de claves aleatorias sesgado para obtener buenas soluciones en problemas con muchos barcos. Hemos realizado diversos experimentos sobre conjuntos de problemas generados con el objetivo de evaluar preliminarmente ambos enfoques. El modelo propuesto ha sido comparado con el modelo más reciente encontrado en la literatura. Los resultados indican que nuestro modelo es mejor en tanto en cuanto no depende del factor de discretización aplicado a las longitudes. Además, consigue la solución óptima en problemas de hasta 40 barcos y en algunos de 50 barcos en tiempos inferiores a la media hora.
El algoritmo genético propuesto consigue soluciones aceptables en muy poco tiempo, aunque parece requerir mejoras, como la aplicación de una búsqueda local o el ajuste de sus parámetros. Además, cabría contrastar sus resultados con respecto al modelo sobre conjuntos de problemas con muchos más barcos. Así pues, queda esto como trabajo futuro.

Agradecimientos. Este estudio ha sido financiado parcialmente por el Ministerio de Economía y Competitividad, DPI2011-24977 y por la Generalitat Valenciana, PROMETEO/2013/049, contrato CPI-13-351.

Referencias

1. Bierwirth, C., Meisel, F.: A survey of berth allocation and quay crane scheduling problems in container terminals. European Journal of Operational Research 202(3), 615–627 (2010)
2. Frojan, P., Correcher, J.F., Alvarez-Valdes, R., Koulouris, G., Tamarit, J.M.: The continuous Berth Allocation Problem in a container terminal with multiple quays. Expert Systems with Applications 42(21), 7356–7366 (2015)
3. Gonçalves, J.F., Resende, M.G.C.: A parallel multi-population biased random-key genetic algorithm for a container loading problem. Computers and Operations Research 39(2), 179–190 (2012)
4. Meisel, F., Bierwirth, C.: Heuristics for the integration of crane productivity in the berth allocation problem. Transportation Research Part E: Logistics and Transportation Review 45(1), 196–209 (2009)
5. Park, Y., Kim, K.: A scheduling method for berth and quay cranes. OR Spectrum (25), 1–23 (2003)
6. Türkoğulları, Y.B., Taşkın, Z.C., Aras, N., Altınel, I.K.: Optimal berth allocation and time-invariant quay crane assignment in container terminals. European Journal of Operational Research 235(1), 88–101 (2014)