

Gestión de Vehículos de Reparto Interno en Terminales Marítimas de Contenedores

Israel López-Plata, Christopher Expósito-Izquierdo, Belén Melián-Batista y J. Marcos Moreno-Vega

Departamento de Ingeniería Informática y de Sistemas
Universidad de La Laguna
38271 La Laguna, España
{ilopezpl, cexposit, mbmelian, jmmoreno}@ull.es

Resumen En este trabajo se introduce el Problema de Sincronización de Vehículos de Reparto Interno. Su objetivo es gestionar la flota de vehículos de reparto interno disponible en el patio de una terminal marítima de contenedores para llevar a cabo el movimiento de contenedores entre las diferentes áreas funcionales. El objetivo del problema es mover los contenedores requeridos en el menor tiempo posible. En este trabajo se propone una Búsqueda por Entornos Variables para asignar, determinar las rutas y planificar los vehículos existentes de forma eficiente. Los resultados computacionales demuestran que el algoritmo propuesto posee un alto rendimiento para un amplio conjunto de escenarios propuestos.

Palabras clave: Búsqueda por Entornos Variables, Vehículos de reparto interno, Terminal marítima de contenedores

1. Introducción

Las terminales marítimas de contenedores son instalaciones logísticas localizadas en puertos y cuyo principal cometido es facilitar el almacenamiento y transbordo de contenedores de una manera eficiente y flexible [6]. En términos generales, una terminal marítima de contenedores puede ser considerada como una interfaz logística que conecta los diferentes medios de transporte dentro de una cadena global de suministros, habitualmente los buques portacontenedores y los transportes terrestres, como camiones y trenes [4].

Una terminal marítima de contenedores suele dividirse en tres áreas funcionales: el lado del mar, que da servicio a los buques entrantes; el patio de contenedores, que proporciona un almacenamiento temporal de los contenedores manejados por la terminal; y la interfaz terrestre, que conecta la terminal con los transportes de tierra. En una terminal marítima de contenedores convencional, los contenedores son movidos entre las diferentes áreas funcionales mediante una flota de vehículos de reparto interno [2].

En términos generales, la gestión de la flota de vehículos disponibles en una terminal puede ser dividida en tres operaciones diferentes, todas con un gran impacto en el rendimiento global de toda la terminal. Primero, los movimientos

de contenedores requeridos deben ser asignados a los vehículos que los van a realizar. Una vez los vehículos conocen el conjunto de tareas que tienen que llevar a cabo, se deben definir las rutas apropiadas para realizar dichos movimientos de contenedores. Por último, es necesario determinar la planificación de las tareas a realizar por cada vehículo con el objetivo de definir cuándo los contenedores van a ser movidos. Además, se requiere la sincronización entre los diferentes vehículos para poder acceder a los contenedores a mover.

En este trabajo se describe y resuelve el Problema de Sincronización de Vehículos de Reparto Interno, PSVRI, en el contexto de una terminal marítima de contenedores. El objetivo de este problema es optimizar la utilización de los vehículos de reparto interno en términos de tiempo de trabajo, donde se requiere sincronización de cara al acceso a las diferentes localizaciones de recogida y entrega de contenedores.

El resto de este trabajo se organiza tal como sigue. El PSVRI es introducido en la Sección 2. La Sección 3 presenta una Búsqueda por Entornos Variables para resolver el problema bajo estudio. Los resultados computacionales son presentados en la Sección 4. Finalmente, la Sección 5 describe las principales conclusiones extraídas de la realización del trabajo y sugiere varias líneas de trabajo futuras.

2. Descripción del Problema

El Problema de Sincronización de Vehículos de Reparto Interno –PSVRI– persigue gestionar la flota de vehículos existente en una terminal marítima de contenedores de forma eficiente. El objetivo de optimización consiste en minimizar el tiempo requerido por los vehículos en desplazar contenedores en la terminal. Esto es, reducir el *makespan* de la planificación de los mismos.

El PSVRI asume la existencia de un conjunto de n trabajos, denotado como J , donde cada trabajo hace referencia al movimiento de un determinado contenedor desde una posición inicial a otra final dentro de la terminal de contenedores. Las posiciones iniciales y finales de un trabajo $j \in J$ son denotadas como $s(j)$ y $t(j)$, respectivamente. El conjunto de todas las posibles posiciones asociadas con al menos un trabajo es denotado como $P = \{1, 2, \dots, p\}$. Cabe mencionar que algunos trabajos pueden compartir posiciones iniciales o finales, en cuyo caso sería necesaria la sincronización en el acceso a dicha posición de los vehículos que realizan estos trabajos. Así pues, la realización de un trabajo implica el movimiento de un vehículo hasta la posición inicial del trabajo, la recogida del contenedor, el movimiento hasta la posición final y por último el almacenamiento del contenedor que se está trasladando. Los tiempos invertidos en la recogida y colocación del contenedor para el trabajo $j \in J$ se denotan como $p(j)$ y $d(j)$, respectivamente.

Los trabajos existentes en el PSVRI son realizados por una flota de m vehículos de reparto interno, denotada como V . Cada vehículo $v \in V$ tiene una posición de partida, $i(v)$, un tiempo inicial en el que empieza a estar disponible, $a(v) \geq 0$, y una prioridad que representa la preferencia del gestor de la terminal por los trabajos realizados por este vehículo en comparación con los de otro, denota-

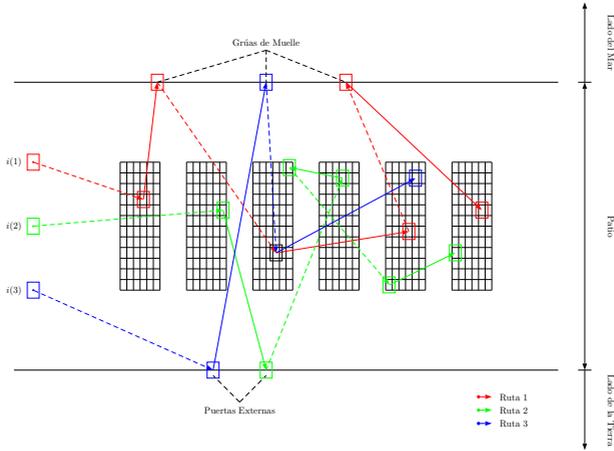


Figura 1: Ejemplo de PSVRI compuesto por $n = 8$ trabajos y $m = 3$ vehículos

da como $\delta(v)$. El tiempo requerido por un vehículo interno para moverse entre dos posiciones i y j es representado como m_{ij} y es proporcional a la distancia Euclídea entre ambas posiciones.

Para una mejor comprensión del problema, en la Figura 1 se muestra un ejemplo ilustrativo del mismo, donde $n = 8$ trabajos deben ser realizados por $m = 3$ vehículos. La ruta realizada por cada vehículo es representada de un color, donde las líneas discontinuas representan el movimiento del vehículo sin carga, mientras que las continuas indican el movimiento con contenedor desde la posición inicial hasta la final de un determinado trabajo. Si se realiza un análisis en profundidad del ejemplo mostrado, se observa que existe una posición en el tercer bloque de contenedores compartida por dos trabajos, realizados por los vehículos 1 y 3 respectivamente. Ambos vehículos no pueden acceder a esta posición al mismo tiempo debido a restricciones físicas del entorno, por lo que el vehículo de menor prioridad $\delta(v)$ debe esperar a que aquel de mayor prioridad finalice su trabajo en la posición compartida.

En contraste con el conocido Problema de Rutado de Vehículos, PRV [3], el PSVRI considera la sincronización en el acceso a las diferentes posiciones del patio. En aquellos casos en los que varios vehículos intentan acceder simultáneamente a una posición de la terminal, éstos son secuenciados, de tal forma que sólo uno de ellos puede realizar su trabajo asignado. En los casos en los que

la sincronización no es requerida, el PSVRI puede ser reducido a un PRV con recogidas y entregas [1], por lo que se puede afirmar que se trata de un problema \mathcal{NP} -duro y consecuentemente se requiere de técnicas de optimización eficientes para su resolución en entornos reales.

3. Búsqueda por Entornos Variables

Para resolver el Problema de Sincronización de Vehículos de Reparto Interno presentado en la Sección 2 del presente trabajo, se propone un algoritmo metaheurístico basado en la conocida Búsqueda por Entornos Variables, en inglés Variable Neighbourhood Search, VNS [5], cuyo objetivo es obtener soluciones de alta calidad en instancias del problema realistas y en tiempos computacionales razonables.

Cada solución factible del problema tiene la forma, $\phi = (r_1, r_2, \dots, r_m)$, donde r_v es la ruta del vehículo $v \in V$. La ruta r_v comienza en la posición inicial de $v \in V$, $i(v)$, y visita consecutivamente la posición inicial y final del subconjunto de trabajos asignados. Esto es, $r_v = (r_v^0, r_v^1, r_v^2, \dots, r_v^{n(v)})$, $\forall v \in V$, donde $n(v)$ es el número de trabajos a realizar por $v \in V$. En este contexto, r_v^i denota el i -ésimo trabajo a realizar por el vehículo $v \in V$ a lo largo de su ruta, r_v . A modo de recordatorio, las posiciones inicial y final del trabajo r_v^i se denotan como $s(r_v^i)$ y $t(r_v^i)$, respectivamente. Además, se incluye un trabajo ficticio, denotado como 0, al comienzo de la ruta r_v para representar la salida del vehículo $v \in V$ desde su posición inicial. En este caso, $s(r_v^0) = t(r_v^0) = i(v)$, $\forall v \in V$. Debido a que 0 es un trabajo ficticio, no hay tiempos de recogida ni almacenamiento asociados. Esto es, $p(r_v^0) = d(r_v^0) = 0$, $\forall v \in V$.

La ruta r_v del vehículo $v \in V$ viene caracterizada por las siguientes cantidades:

- Tiempo invertido en mover el vehículo v desde la posición final de un trabajo hasta la posición inicial del siguiente:

$$t_{mov}(r_v) = \sum_{i=0}^{n(v)-1} m_{t(r_v^i)s(r_v^{i+1})} \quad (1)$$

- Tiempo invertido en transportar los contenedores:

$$t_{tran}(r_v) = \sum_{i=1}^{n(v)} m_{s(r_v^i)t(r_v^i)} \quad (2)$$

- Tiempo invertido en recoger y almacenar los contenedores:

$$t_{proc}(r_v) = \sum_{i=1}^{n(v)} (p(r_v^i) + d(r_v^i)) \quad (3)$$

De acuerdo a las ecuaciones anteriores, el tiempo invertido por el vehículo $v \in V$ en realizar los trabajos asignados es:

$$t(r_v) = a(v) + t_{mov}(r_v) + t_{tran}(r_v) + t_{proc}(r_v) \quad (4)$$

La función objetivo del PSVRI es minimizar el *makespan* de la planificación realizada:

$$\min_{v \in V} \max t(r_v) \quad (5)$$

El pseudocódigo de la Búsqueda por Entornos Variables propuesta se muestra en el Algoritmo 1, y su objetivo principal es, a partir de una solución inicial, explorar el espacio de soluciones de forma eficiente mediante la aplicación de varias estructuras de entorno. Así pues, el primer paso del algoritmo consiste en construir una solución inicial para el problema. Esto se realiza mediante un método constructivo que asigna los trabajos seleccionados de forma aleatoria a aquel vehículo con menor tiempo de uso (línea 3). A partir de aquí se explora el espacio de soluciones hasta alcanzar el criterio de parada definido. La búsqueda propuesta explora los k_{max} entornos disponibles, los cuales son:

- *Intercambio*. Intercambia el orden de realización de dos trabajos diferentes dentro de una misma ruta de vehículos.
- *Reinserción*. Elimina la asignación de un trabajo a un determinado vehículo y lo reasigna a otro vehículo diferente, en un orden establecido por el propio movimiento.

Antes de iniciar la exploración de los entornos disponibles, la solución actual pasa por un proceso de agitación (línea 8). Este proceso permite evitar la repetición de la misma posición inicial en la búsqueda local realizada posteriormente y basada en el entorno analizado N_k (línea 9). Si la solución obtenida en la búsqueda local mejora a la mejor solución obtenida hasta ese momento (línea 10), entonces la nueva solución es considerada como la mejor solución y la exploración de los entornos comienza de nuevo (línea 12). En otro caso, el algoritmo comienza con la exploración del siguiente entorno (línea 14). La solución con mejor valor objetivo encontrada en todas las exploraciones (línea 18) es comparada con la mejor solución encontrada en todas las repeticiones realizadas. Esta última solución es la devuelta por el algoritmo antes de finalizar su ejecución (línea 22).

4. Resultados Computacionales

Esta sección tiene como objetivo evaluar el rendimiento del algoritmo propuesto en la Sección 3 para resolver el Problema de Sincronización de Vehículos de Reparto Interno. La Búsqueda por Entornos Variables descrita en la sección previa se ha implementado en Java Standard Edition 8, mientras que las ejecuciones se han realizado en un ordenador equipado con un Intel i7-1.80 GHz y 8 GBs de RAM. Los objetivos específicos de esta sección son los siguientes:

Algoritmo 1 Búsqueda por Entornos Variables para la resolución del Problema de Sincronización de Vehículos de Reparto Interno

```
1:  $s_{mejor} \leftarrow \emptyset$ 
2: for (número de repeticiones) do
3:    $s \leftarrow \text{MétodoConstructivo}()$ 
4:    $svNS \leftarrow s$ 
5:   while (no se alcanza el criterio de parada) do
6:      $k \leftarrow 0$ 
7:     while ( $k < k_{max}$ ) do
8:        $s_{agitada} \leftarrow \text{Agitar}(s, N_k(s))$ 
9:        $s_{local} \leftarrow \text{BúsquedaLocal}(s_{agitada}, N_k(s))$ 
10:      if ( $f(s_{local}) < f(s)$ ) then
11:         $s \leftarrow s_{local}$ 
12:         $k \leftarrow 0$ 
13:      else
14:         $k \leftarrow k + 1$ 
15:      end if
16:    end while
17:    if ( $f(s) < f(svNS)$ ) then
18:       $svNS \leftarrow s$ 
19:    end if
20:  end while
21:  if ( $s_{mejor} = \emptyset$  or  $f(svNS) < f(s_{mejor})$ ) then
22:     $s_{mejor} \leftarrow svNS$ 
23:  end if
24: end for
25: Devolver  $s_{mejor}$ 
```

- Evaluar el rendimiento relativo del algoritmo explicado en la Sección 3 en comparación con otros algoritmos conocidos.
- Analizar la influencia en términos de calidad de la solución y tiempo de ejecución de cada uno de los factores que caracterizan al PSVRI.

Los experimentos computacionales se han llevado a cabo sobre un conjunto de instancias realistas creadas para el desarrollo de esta experimentación. Su objetivo es abarcar el mayor rango posible de entornos potenciales encontrados en una terminal marítima de contenedores. Las instancias generadas difieren en unas de las otras en base a las siguientes características:

- *Número de trabajos*, n . Indica el número de trabajos a realizar por los vehículos en la terminal. Los valores de este parámetro son 20, 40 y 60 trabajos.
- *Número de vehículos de reparto interno*, m . Representa el número de vehículos de reparto interno disponibles en la terminal para realizar los trabajos requeridos. Sus valores son 2, 5 y 10.
- *Número de posiciones*, p . Indica el número de posiciones dentro de la terminal y donde los vehículos pueden situarse. El valor de este parámetro es 50, 100 y 200 posiciones.

Para cada combinación de valores se han generado 5 instancias, por lo que en total se disponen de 90 instancias diferentes para la realización de los experimentos computacionales así como para investigaciones futuras.

4.1. Evaluación del algoritmo propuesto

Los experimentos computacionales realizados persiguen evaluar el rendimiento del algoritmo propuesto en la Sección 3 en términos de valor de función objetivo y de tiempo computacional. Con el objetivo de conocer la eficacia de la Búsqueda por Entornos Variables propuesta, ésta es comparada con otras propuestas algorítmicas conocidas en la literatura, las cuales son las siguientes:

- *BKS*. Búsqueda multiarranque con múltiples entornos, la cual cambia el entorno utilizado si la solución no puede ser mejorada. El objetivo de este algoritmo es representar la solución mas próxima a la óptima, para que el resto de algoritmos puedan compararse con esta. Es por ello por lo que este algoritmo es ejecutado 100 veces más que el resto de algoritmos.
- *VNS*. Algoritmo basado en la Búsqueda por Entornos Variables propuesto en la Sección 3.
- *Local*. Búsqueda local utilizando sólo uno de los entornos disponibles. La solución inicial es obtenida con la misma función constructiva que la utilizada en el algoritmo *VNS*.
- *Random*. Solución obtenida de forma completamente aleatoria.

El Cuadro 1 muestra los resultados computacionales devueltos por la ejecución de los algoritmos mostrados anteriormente en las instancias creadas. Las columnas n , p y m representan el número de trabajos, posiciones y vehículos respectivamente. La columna f_{BKS} es el valor medio de la función objetivo devuelto por el algoritmo *BKS*, el cual es utilizado como referencia. Igualmente, las columnas Gap_{VNS} , Gap_{Local} y Gap_{Random} representan la diferencia en porcentaje de el valor medio de la función objetivo de sus algoritmos en comparación con el valor de f_{BKS} . Los tiempos computacionales requeridos por los algoritmos para resolver el problema son mostrados en las columnas $t_{VNS}(s)$, $t_{Local}(s)$ y $t_{Random}(s)$. En la última fila de la tabla se muestra el valor medio de cada una de las columnas, con el objetivo de poseer una perspectiva general de los resultados mostrados en la tabla.

De los resultados expuestos en el Cuadro 1 se pueden extraer las siguientes conclusiones:

- El algoritmo propuesto basado en Búsquedas por Entornos Variables posee una desviación negativa en todas sus ejecuciones. Esto significa que este algoritmo puede obtener soluciones mejores que las alcanzadas por el algoritmo *BKS*, alcanzando en algunos casos una desviación de hasta un -22.5%. Estas mejores soluciones son obtenidas con 100 veces menos ejecuciones, lo que implica un mejor rendimiento computacional.

n	p	m	f_{BFS}	t_{VNS} (s.)	Gap_{VNS}	t_{Local}	Gap_{Local}	t_{Random} (s.)	Gap_{Random}
20	50	2	1778.2	0.641	-0.292	0.038	21.021	0.006	25.205
		5	800.0	1.526	-1.700	0.108	32.975	0.007	39.600
		10	494.4	1.783	-2.306	0.225	41.950	0.002	59.385
	100	2	1859.8	0.336	-1.194	0.044	13.249	0.004	15.389
		5	822.0	1.183	-1.460	0.102	21.290	0.007	28.686
		10	477.0	1.601	-3.732	0.176	37.610	0.004	43.354
40	50	2	3531.2	5.815	-6.417	0.090	68.668	0.011	65.808
		5	1835.2	13.693	-16.434	0.493	77.278	0.011	93.025
		10	1101.2	16.817	-19.887	0.716	61.733	0.014	78.641
	100	2	3441.6	5.294	-2.476	0.123	36.239	0.011	43.119
		5	1565.8	9.720	-7.511	0.526	49.495	0.011	75.131
		10	899.2	17.305	-8.786	1.174	67.905	0.012	79.471
60	100	2	5286.6	22.245	-5.474	0.186	67.355	0.018	67.253
		5	2700.6	47.398	-19.633	1.184	80.241	0.030	94.883
		10	1577.6	63.603	-21.450	3.458	67.748	0.028	91.899
	200	2	5054.6	19.471	-3.047	0.356	40.442	0.028	47.240
		5	2374.6	42.976	-8.355	2.333	57.857	0.030	68.559
		10	1387.4	61.890	-12.457	5.093	71.362	0.022	90.339
-	-	-	2052.1	18.527	-7.923	0.924	50.800	0.014	61.499

Cuadro 1: Resultados computacionales devueltos por los algoritmos propuestos cuando solucionan el problema propuesto

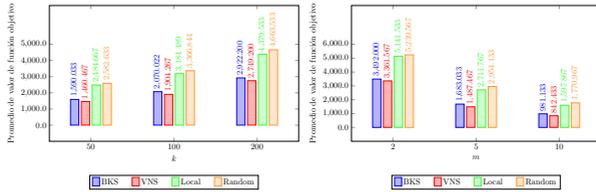


Figura 2: Comparación de valores medios de función objetivo para un número diferente de posiciones y vehículos

- Los algoritmos *Local* y *Random* obtienen soluciones con un tiempo promedio menor a 1 segundo, en comparación con los 18 segundos invertidos por el algoritmo *VNS* de media. Sin embargo, la calidad de las soluciones devueltas por el algoritmo *VNS* –con una desviación de un -8%– es mucho mejor que las retornadas por los algoritmos *Local* y *Random*, con una desviación del 50,8% y 61,5% respectivamente. Como conclusión, el algoritmo *VNS* puede obtener soluciones de calidad en tiempos computacionales pequeños.

Para reforzar las conclusiones expuestas anteriormente, se realiza un análisis de los diferentes valores medios de la función objetivo para todos los algoritmos incluidos en la experimentación. La Figura 2 muestra una comparación para un diferente número de posiciones así como para un número distinto de vehículos de reparto interno.

5. Conclusiones y Líneas de Trabajo Futuras

En este trabajo se presenta el Problema de Sincronización de Vehículos de Reparto Interno, el cual persigue gestionar la flota de vehículos de reparto interno disponible en una terminal de contenedores, minimizando el tiempo requerido por dichos vehículos en realizar un conjunto de trabajos a realizar y minimizando el makespan de su planificación. Para resolverlo, se desarrolla un algoritmo aproximado basado en una Búsqueda por Entornos Variables, el cual es comparado con otras estrategias bien conocidas para resolver este tipo de problemas, como son Búsquedas Locales y Búsquedas Aleatorias.

Para evaluar el rendimiento del algoritmo tanto en valor medio de función objetivo como en tiempo, se realiza una experimentación sobre un gran número de escenarios realistas, los cuales varían en número de vehículos, posiciones y trabajos. Los resultados de la experimentación demuestran que el algoritmo propuesto devuelve soluciones de alta calidad en tiempos computacionales cortos, lo que permite abordar escenarios de gran tamaño y en entornos prácticos.

Como líneas de trabajo futuras, se propone la realización de un modelo de optimización el cual permita resolver de forma óptima el problema bajo análisis. Otro objetivo futuro es el desarrollo de un gestor de toma de decisiones que permita a los gestores de terminales portuarias evaluar las diferentes alternativas de gestión de vehículos internos de una manera efectiva. Como última línea de trabajo futura se propone la inclusión del dinamismo para este problema, que permita gestionar cambios en el entorno en tiempos de ejecución, esto es, una vez los vehículos se encuentran en movimiento. Los cambios a realizar pueden ser tan diversos como la adición o eliminación de trabajos, así como el cambio de prioridad de los vehículos de reparto interno.

Agradecimientos

Este trabajo ha sido parcialmente financiado por el Ministerio de Economía y Competitividad de España (proyectos TIN2012-32608 y TIN2015-70226-R). Christopher Expósito-Izquierdo agradece al Gobierno de Canarias el apoyo financiero que recibe a través de su beca de doctorado.

Referencias

1. G. Berbeglia, J.F. Cordeau, I. Gribkovskaia, and G. Laporte. Static pickup and delivery problems: a classification scheme and survey. *TOP*, 15(1):1–31, 2007.
2. E.K. Bish, F.Y. Chen, Y.T. Leong, B.L. Nelson, J.W.C. Ng, and D. Simchi-Levi. Dispatching vehicles in a mega container terminal. *OR Spectrum*, 27(4):491–506, 2005.
3. G.B. Dantzig and J.H. Ramser. The truck dispatching problem. *Management Science*, 6(1):80–91, 1959.
4. J.C. Fransoo and C.Y. Lee. The critical role of ocean container transport in global supply chain performance. *Production and Operations Management*, 22(2):253–268, 2013. cited By 12.
5. P. Hansen, N. Mladenovi, J. Brimberg, and J.A. Moreno-Prez. Variable neighborhood search. In M. Gendreau and J.Y. Potvin, editors, *Handbook of Metaheuristics*, volume 146 of *International Series in Operations Research & Management Science*, pages 61–86. Springer US, 2010.
6. I.F.A. Vis and R. de Koster. Transshipment of containers at a container terminal: An overview. *European Journal of Operational Research*, 147(1):1 – 16, 2003.