

## Resolución de los problemas Order Batching / Picking a través de un Algoritmo Evolutivo

Fabio M. Miguel<sup>1</sup> [0000-0002-4810-8008], Mariano Frutos<sup>2</sup> [0000-0003-2585-4195],  
Máximo Méndez<sup>3</sup> [0000-0002-7133-7108], Fernando Tohme<sup>4</sup> [0000-0003-2988-4519]

<sup>1</sup> Sede Alto Valle y Valle Medio,  
Universidad Nacional de Río Negro, Argentina.

<sup>2</sup> Departamento de Ingeniería,  
Universidad Nacional del Sur e IIESS UNS-CONICET, Argentina.

<sup>3</sup> Instituto Universitario de Sistemas Inteligentes SIANI,  
Universidad de Las Palmas de Gran Canaria ULPGC, España.

<sup>4</sup> Departamento de Economía,  
Universidad Nacional del Sur e INMABB UNS-CONICET, Argentina.

fmiguel@unrn.edu.ar, mfrutos@uns.edu.ar,  
maximo.mendez@ulpgc.es, ftohme@criba.edu.ar

**Abstract.** En este trabajo se propone un algoritmo evolutivo para resolver de manera combinada el Order Batching Problem y el Order Picking Problem. Este problema combinado consiste en seleccionar y recoger en lotes varios artículos de pedidos de distintos clientes de forma simultánea desde un área de almacenamiento y dado un plazo para la finalización del armado de cada pedido conforme a un plan de distribución. Se busca determinar el plan que minimiza el costo total del proceso de picking, el que será directamente proporcional al tiempo de desplazamiento en el área de almacenamiento, al tiempo para recoger cada artículo y descargarlo en el área de despacho y a la puntualidad en el armado de cada pedido, en el sentido de que la anticipación o la demora acarreará costos por ineficiencia en el uso del espacio o por incumplimientos del plan de distribución. Del análisis de los resultados alcanzados, al compararlos con otros resultados extraídos de la literatura, se observa un buen desempeño del algoritmo propuesto en este trabajo.

**Keywords:** Order Batching, Order Picking, Algoritmo Evolutivo.

### 1 Introducción

La optimización de procesos dentro de un centro de distribución es un factor crítico para el desempeño operativo de la logística interna y externa de la empresa. Estos procesos operativos están relacionados a los movimientos de mercaderías dentro de los centros de distribución, como el de recepción, ubicación, almacenamiento, selección y

recolección de pedidos, clasificación y despacho [1] [2]. El almacenamiento, hace referencia a la guarda física de los artículos a la espera de un pedido. La selección y recolección de estos artículos es el proceso principal en la mayoría de los depósitos. Consiste en tomar desde las ubicaciones de almacenamiento correspondientes, la cantidad correcta de los artículos correctos en base a lo requerido por los clientes, en sus pedidos, para transportarlos hacia la zona de preparación de los pedidos [3]. La clasificación de los pedidos seleccionados y recolectados consiste en la reagrupación de las unidades de cada lote en los pedidos individuales de cada cliente una vez que se completa el proceso de picking [4] [5]. En la mayoría de los casos este proceso incluye las actividades de marcado y etiquetado, de embalaje y agrupamiento en una unidad indivisible de carga. El despacho es el proceso que involucra la verificación de cada unidad de carga chequeando que los pedidos estén completos, la elaboración de los documentos necesarios para poder despachar la mercadería y finalmente la carga de la misma al medio de transporte. El presente trabajo se enfoca en la optimización del proceso de selección y recolección de pedidos.

## **2 Descripción del problema y revisión de la literatura**

Respecto a los mencionados procesos operativos relacionados al movimiento de mercaderías dentro de los depósitos, se identifican claramente tres problemas. El primero de ellos, es la asignación de los artículos ingresados a diferentes posiciones de almacenamiento. El segundo, es el agrupamiento de artículos de pedidos de clientes en lotes para su recolección. Y el tercero, es la programación de la secuencia de recolección de los artículos realizada por los equipos de picking y el traslado al área de despacho [6] [7] [8]. En este trabajo nos enfocamos en el tratamiento integrado de estos dos últimos problemas, dado que son críticos para la eficiencia de las operaciones del depósito e involucran la mayor parte de los costos de las operaciones del depósito, al comprender actividades intensivas en mano de obra [9] [10]. Para entender este problema primero debemos describir con mayor detalle el proceso de selección y recolección de los artículos. Este proceso comienza con el ingreso de una orden de preparación de una determinada cantidad de pedidos realizados por distintos clientes, en la que se detalla para cada pedido las cantidades y especificaciones de distintos artículos disponibles en el área de almacenamiento, y en la que se define la fecha o momento en el que debe estar disponible cada pedido en el área de despacho. Este momento está definido en base al programa de distribución y el tiempo requerido en el proceso de clasificación y despacho. Los artículos deben ser recogidos de las posiciones de almacenamiento en el menor tiempo posible, para lo que cada equipo de picking, debe visitar una secuencia de posiciones de almacenamiento con el fin de tomar los artículos solicitados. Al finalizar esta operación, se obtiene el tiempo de procesamiento de la orden y de cada uno de los pedidos. El problema integrado consiste en determinar el plan que minimiza el costo operativo del proceso de picking, esto es, el costo de seleccionar y recoger en lotes varios artículos de pedidos de distintos clientes de forma simultánea desde un área de almacenamiento determinada dado un plazo para la finalización del armado de cada

pedido, donde cada pedido está conformado por distintas cantidades de diferentes artículos y cada lote está conformado por una variedad de artículos de diferentes pedidos [11] [12] [13]. Por un lado, si se excediera el momento en que finaliza el armado del pedido respecto al momento en que dicho pedido debía estar listo, se incurriría en costos por incumplimientos del programa de distribución. Y, por otro lado, si dicho momento se anticipara se incurriría en un costo por la utilización ineficiente del área de despacho, al ocuparse con pedidos que todavía no deben ser despachados obstaculizando los flujos dentro del área aumentando sus tiempos de procesamiento [11]. Formalmente, el problema integrado está compuesto por el Order Batching Problem (OBP) y el Order Picking Problem (OPP) [14] [15] [16]. El OBP consiste en encontrar la cantidad y el tamaño óptimo de los lotes de artículos, para que cada lote sea recogido por un equipo de picking [13]. Para esto se debe considerar la capacidad del equipo de picking y el momento requerido en que debe estar cada artículo en el área de despacho para la finalización del armado de cada pedido [17] [18] [19]. El OPP consiste en identificar secuencias óptimas de visitas de las posiciones de almacenamiento en las que se encuentran cada uno de los artículos incluidos en un lote determinado [20] [21] [22]. En adelante se identifica este problema integrado como OBP+OPP. En [2] se realiza una revisión de la literatura sobre el OBP y el OPP. En [1] [21] [23] se presenta una revisión de heurísticas para el OPP. En [8] se presentan dos formas de resolución al OPP, mediante Ant Colony Optimization y mediante Iterated Local Search. En [15] se presentan otras meta-heurísticas aplicadas a los problemas de ruteo subyacentes. En [4], se utiliza un enfoque basado en el clustering para resolver el OBP teniendo en cuenta patrones de demanda en lugar de la distancia recorrida en cada secuencia de visitas. En [7] se propusieron varias heurísticas para el OBP. Luego, en [13] se propuso un modelo de programación entera para el OBP donde las distancias de cada secuencia de visita son estimadas y el problema se resuelve con una heurística basada en lógica difusa. En [11] se utiliza un algoritmo genético múltiple para resolver el OBP+OPP. Este trabajo es novedoso por la utilización de ventanas temporales flexibles para el tiempo de entrega de cada pedido. En el presente trabajo se sigue la metodología utilizada en esta última publicación, pero utilizando para la resolución del problema un algoritmo evolutivo con un cromosoma específico que facilita la utilización de diferentes números de lotes. Mediante este algoritmo se busca obtener mejoras en el desempeño comparando los resultados obtenidos sobre las instancias y configuración del lay-out propuestos en la publicación mencionada.

### 3 Definición de parámetros y variables

$\mathcal{P} = \{1, \dots, nArt\}$  es el conjunto de artículos, hay  $nArt$  tipos de artículos diferentes y cada uno de estos tipos de artículos tiene un peso unitario determinado. El conjunto de pesos unitarios de los  $nArt$  tipos de artículos se representa mediante  $\mathcal{W} = \{w_1, \dots, w_p, \dots, w_{nArt}\}$ .  $\mathcal{P}_i$  es el subconjunto de artículos pedidos por el cliente  $i$ , es decir  $\mathcal{P}_i \subset \mathcal{P}$ . Además se supone que cada cliente realiza un solo pedido de varios artículos diferentes con distintas cantidades de cada uno de ellos, por lo que la cantidad de clientes,  $nC$ , es igual a la cantidad de pedidos,  $nPed$ , quedando el conjunto de

clientes determinado por  $\mathcal{J} = \{1, \dots, i, \dots, nC\}$ . Cada pedido tiene un plazo determinado para finalizar con su armado, los plazos de finalización de todos los pedidos están representados en el conjunto  $\mathcal{T} = \{t_1, \dots, t_i, \dots, t_{nPed}\}$ .  $\mathcal{P}_r$  es el subconjunto de artículos diferentes presentes en el lote  $r$ . Es decir  $\mathcal{P}_r \supset \mathcal{P}$ . Los artículos en  $\mathcal{P}_r$  pueden provenir de pedidos de diferentes clientes o no.  $\mathcal{L} = \{\ell_0, \ell_1, \dots, \ell_p, \dots, \ell_{nArt}\}$  son las posiciones de almacenamiento de cada uno de los tipos de artículos, más  $\ell_0$  que es la localización del área de despacho dentro del almacén. Por ejemplo, para el artículo  $p \in \mathcal{P}$ , la posición de almacenamiento  $\ell_p$  está dada por sus coordenadas dentro del depósito, es decir,  $\ell_p = (x_p, y_p)$ .  $\mathcal{R} = \{1, \dots, r, \dots, |\mathcal{R}|\}$  es el conjunto de lotes a recoger por los operadores de los equipos de picking. Donde  $|\mathcal{R}|$  es la cantidad de lotes a recoger.  $\mathcal{S}_r = \langle s_1, \dots, s_u, \dots, s_{|\mathcal{S}_r|} \rangle$  es la ruta o sucesión de posiciones de almacenamiento a visitar en el lote  $r$ . Es decir,  $r \in \mathcal{R}$  y  $\mathcal{S}_r \supset \mathcal{L}$ . Y donde,  $s_u$  es la  $u$ -ésima posición de almacenamiento a visitar en el lote  $r$  y  $|\mathcal{S}_r|$  es la cantidad de artículos diferentes del lote  $r$ .  $\mathcal{Q}$  es el conjunto de las cantidades de cada artículos pedidos por todos los clientes, donde,  $q_{i,p} \in \mathcal{Q}$  es la cantidad total pedida por el cliente  $i$  del artículo  $p$ . Luego  $Q_i = \sum_{p \in \mathcal{P}_i} q_{i,p}$  es la cantidad total de artículos solicitada por el cliente  $i$ .  $Q_p = \sum_{i \in \mathcal{J}} q_{i,p}$  es la cantidad pedida por todos los clientes del artículo  $p$ . Y  $Q_r \supset \mathcal{Q}$  es el conjunto de cantidades de distintos artículos agrupados en el lote  $r$ .  $Q_r = \{q_1, \dots, q_p, \dots, q_{|\mathcal{S}_r|}\}$ .  $\mathcal{K} = \{1, \dots, |\mathcal{K}|\}$  es el conjunto de equipos de picking con capacidad  $Cap$ . Luego, el problema integrado OBP+OPP, se define en un grafo no dirigido  $\mathcal{G} = (\mathcal{V}, \mathcal{A})$ , donde:  $\mathcal{V}$  es el conjunto de nodos que representan todas las posiciones de almacenamiento (lugares físicos) de cada uno de los artículos  $p \in \mathcal{P}$ , más dos copias (0 y  $n + 1$ ) del nodo inicial (ubicación del área de despacho).  $\mathcal{A}$  representa el conjunto de todos los arcos factibles entre todos los nodos del conjunto  $\mathcal{V}$ . Cada arco  $(h, l) \in \mathcal{A}$  tiene asociado un tiempo  $t_{hl}$  dado por la distancia entre los nodos  $h$  y  $l$  dividida por la velocidad del equipo de picking  $v$  (esto es  $t_{hl} = D_{h,l}/v$ ) y un costo operativo por unidad de tiempo  $\zeta$ . Luego,  $t_{pick}$  es el tiempo medio para recoger un artículo, una vez que el operador ha llegado al punto de almacenamiento de un artículo determinado.  $Q_i$  representa la cantidad total de artículos pedidos por el cliente  $i$ , donde  $Q_i = \sum_{p \in \mathcal{P}} q_{i,p}$ . Esto significa que la cantidad total de artículos pedidos por el cliente  $i$ ,  $q_i$ , está formada por la suma de las cantidades solicitadas, en ese pedido, de cada uno de los  $h$  artículos solicitados por el cliente  $i$ .  $x_{hlkr} = 1$  si y solo si el artículo  $h$  es recogido inmediatamente antes del artículo  $l$  por el equipo de picking  $k$  en la secuencia del lote  $r$ , donde  $h, l \in \mathcal{V}$ ,  $k \in \mathcal{K}$  y  $r \in \mathcal{R}$ . Es decir, es igual a 1 si el vehículo  $k$  al recoger el lote  $r$  atraviesa el arco  $(h, l)$ .  $y_{hkr} = 1$  si y solo si el equipo de picking  $k$  recoge el artículo  $h$  en el lote  $r$ , donde  $h \in \mathcal{V}$ ,  $k \in \mathcal{K}$  y  $r \in \mathcal{R}$ . El modelo para el OBP+OPP es el siguiente [3] [11]:

$$\min C_{Total} : \left[ \frac{\sum_{h \in \mathcal{V}} \sum_{l \in \mathcal{V}} D_{h,l} \cdot \sum_{k \in \mathcal{K}} \sum_{r \in \mathcal{R}} x_{hlkr}}{v} + \sum_{\substack{p \in \mathcal{P} \\ q \in \mathcal{Q}}} q_p \cdot t_{pick} \right] \cdot \zeta + \sum_{i \in \mathcal{J}} (\alpha \cdot E_i + \beta \cdot T_i) \quad (1)$$

s.a.:

$$\sum_{p \in \mathcal{P}_r} (q_p \cdot w_p) \cdot y_{pkr} \leq Cap, \forall k \in \mathcal{K}, r \in \mathcal{R} \quad (2)$$

$$\sum_{r \in \mathcal{R}} y_{hkr} = 1, \forall h \in \mathcal{P}, \forall k \in \mathcal{K} \quad (3)$$

$$\sum_{k \in \mathcal{K}} y_{hkr} = |\mathcal{K}|, \forall h \in \{0, n+1\}, r \in \mathcal{R} \quad (4)$$

$$\sum_{h \in \mathcal{V}} x_{hlkr} = y_{lkr}, \forall l \in \mathcal{V} \setminus \{0\}, k \in \mathcal{K}, r \in \mathcal{R} \quad (5)$$

$$\sum_{l \in \mathcal{V}} x_{hlkr} = y_{hkr}, \forall h \in \mathcal{V} \setminus \{n+1\}, k \in \mathcal{K}, r \in \mathcal{R} \quad (6)$$

$$\sum_{i \in \mathcal{J}} \sum_{k \in \mathcal{K}} \sum_{r \in \mathcal{R}} q_{i,p} \cdot y_{pkr} = Q_p, \forall p \in \mathcal{P} \quad (7)$$

$$\sum_{p \in \mathcal{P}} \sum_{k \in \mathcal{K}} \sum_{r \in \mathcal{R}} q_{i,p} \cdot y_{pkr} = Q_i, \forall i \in \mathcal{J} \quad (8)$$

$$x_{hlkr} \in \{0,1\}, \forall h, l \in \mathcal{V}, k \in \mathcal{K}, r \in \mathcal{R} \quad (9)$$

$$y_{hkr} \in \{0,1\}, \forall h \in \mathcal{V}, k \in \mathcal{K}, r \in \mathcal{R} \quad (10)$$

La función objetivo (1) representa el costo total expresado en unidades monetarias por unidad de tiempo consumida en la recolección de pedidos en lotes, más una penalización por incumplimiento del plazo para finalizar el armado del pedido. El primer término de esta función, considera el tiempo de desplazamiento desde el área de preparación hasta y entre las posiciones de almacenamiento, más el tiempo de recolección, que es el tiempo requerido para recoger cada artículo una vez que se ha llegado a la posición de almacenamiento. El tiempo de desplazamiento surge del cociente entre la distancia recorrida y la velocidad promedio de desplazamiento del equipo de picking. Respecto a la función de penalización por anticipación o demora en el cumplimiento del plazo para finalizar el armado de cada pedido,  $\alpha$  es la penalización por unidad de tiempo en que se anticipa el armado de un pedido.  $\beta$  es la penalización por unidad de tiempo en que se retrasa el armado de un pedido.  $E_i$  es la cantidad de tiempo en que se anticipa al plazo el armado del pedido  $i$ . Y  $T_i$  es la cantidad de tiempo en que se retrasa el armado del pedido  $i$ . Donde  $E_i = \max\{0, t_i - c_i\}$  y  $T_i = \max\{0, c_i - t_i\}$ . Y siendo  $c_i$  el tiempo de finalización efectiva para el pedido  $i$ , que se define como el tiempo en el que todos los artículos del pedido  $i$  son recogidos y devueltos al área de despacho. El conjunto de restricciones (2) no permite que el peso de los artículos de un lote determinado exceda la capacidad de cada equipo de picking. El conjunto de restricciones (3) indica que cada posición de almacenamiento no sea visitada más de una vez en cada lote  $r$ . El conjunto de restricciones (4) asegura que todos los equipos de picking salgan y regresen al área de preparación para despacho. El conjunto de restricciones (5) y (6) preservan el flujo. Si el equipo de picking  $k$  recoge el artículo  $l$  en el lote  $r$ , antes tiene que haber recogido al artículo  $h$ . Y si el equipo de picking  $k$  recoge el artículo  $h$  en el lote  $r$ , antes tiene que haber recogido al artículo  $l$ . La restricción (7) indica que se cumpla con la cantidad pedida por todos los clientes del artículo  $p$ . La restricción (8) indica

que se cumpla con la cantidad pedida de todos los artículos, por el cliente  $i$ . Las restricciones (9) y (10) imponen las condiciones sobre los valores de las variables.

#### 4 Método de resolución

La teoría computacional clasifica a estos problemas con una complejidad NP-Hard. Es decir, que se pueden resolver de forma exacta solo en instancias muy pequeñas. Dada la intratabilidad de este problema, se requiere de algoritmos no deterministas para resolverlo en tiempos polinomiales. Por esta razón se propone un algoritmo evolutivo para resolver este problema. Se utiliza una representación del tipo permutación de enteros, muy común en problemas de optimización combinatoria como este. Se usa una representación con un cromosoma compuesto por dos genomas. Para explicar la representación veamos mediante un ejemplo muy simple, cómo se codifica una solución a partir de tres pedidos de entre los que se identifican solicitudes de cuatro artículos diferentes. En la Tabla 1, se muestra la información respecto a los pedidos ingresados. Puede verse que el pedido 1 está constituido por 3 artículos (una unidad del tipo A y dos unidades del tipo C), el pedido 2 por 5 artículos (una unidad de A, una de B, dos de C y una de D) y el pedido 3 por 4 artículos (dos unidades de B y dos de D).

**Tabla 1.** Ejemplo para el OBP+OPP

Pedidos	Artículos				Total
	A	B	C	D	
1	1	0	2	0	3
2	1	1	2	1	5
3	0	2	0	2	4
<b>Total</b>	2	3	4	3	12

Para codificar esta información, la ordenamos en dos filas, una para los artículos (y se identifica el pedido al que corresponde cada artículo) y otra para asociar un número entero a cada uno de los artículos (Tabla 2).

**Tabla 2.** Codificación

Artículos (Pedido al que pertenece)											
A(1)	A(2)	B(2)	B(3)	B(3)	C(1)	C(1)	C(2)	C(2)	D(2)	D(3)	D(3)
↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑
1	2	3	4	5	6	7	8	9	10	11	12

Luego, se diseña un cromosoma con dos genomas (Tabla 3) que tiene en cuenta la codificación de la Tabla 2.

Cromosoma: {[G1] [G2]}

Genomas:

[G1]: Secuencia en que se recogerán los artículos en el almacén.

[G2]: Secuencia de posiciones en el [G1] de finalización de cada lote.

Dimensiones:

[G1]: Número total de artículos  $\sum_{i \in J} Q_i$ .

[G2]: Número total de artículos  $\sum_{i \in J} Q_i$  (el límite corresponde a lotes de un solo artículo).

**Tabla 3.** Cromosoma = Genoma 1 + Genoma 2

Genoma 1												
Posiciones →	1	2	3	4	5	6	7	8	9	10	11	12
[G1] →	6	7	9	10	11	1	3	2	12	4	8	5
	lote 1				lote 2		lote 3					
Genoma 2												
[G2] →	5	7	12	-	-	-	-	-	-	-	-	-

La secuencia en que se recogerán los artículos correspondientes a cada lote es la siguiente, lote 1: {6→7→9→10→11}, lote 2: {1→3} y lote 3: {2→12→4→8→5}. Esto indica, por ejemplo, que para el lote 2, el equipo de picking debe ir a la posición de almacenamiento del artículo 1 (A(1)), luego se debe ir a la posición de almacenamiento del artículo 3 (B(2)) y finalmente ir al área de despacho y colocar cada artículo en el lugar del área de almacenamiento donde se están armando los pedidos. En la figura 1 se muestran los recorridos que realizarán los equipos de picking. Este tipo de representación permite lograr mayores niveles de eficiencia en este tipo de problemas y da la posibilidad de incorporar de manera directa conocimiento específico del problema, lo que da mayor flexibilidad a la representación y facilita la consideración de restricciones en el espacio de búsqueda. Por otro lado, la factibilidad de los individuos respecto a los conjuntos de restricciones (2) - (8) están garantizados por la representación. En la inicialización, se busca la mayor variedad posible generando la población inicial de forma aleatoria y se utiliza un criterio de terminación enfocado al costo, que limita el número máximo de iteraciones. El proceso de búsqueda es dirigido mediante el criterio de selección favoreciendo los individuos más aptos. Para ello se utiliza la técnica de selección por torneo en la que cada individuo se toma eligiendo el mejor de un conjunto de  $k$  individuos tomados al azar de la población actual [24]. Esto se repite iterativamente hasta completar la muestra. Los operadores genéticos son aplicados al Genoma 1. Se utiliza un operador de cruce Edge Recombination Operator [25] y un operador de mutación de inserción [26]. El Genoma 2 se completa de acuerdo a la capacidad del equipo de Picking. El pseudo-código del algoritmo propuesto se presenta en la figura 2.

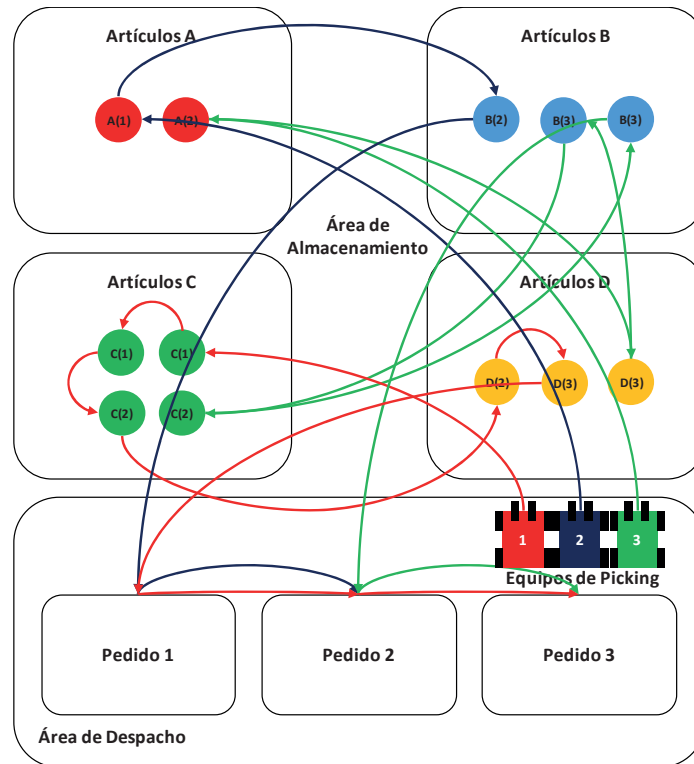


Fig. 1. Recorridos de los equipos de picking

---

```

1: Load Input      % información de Pedidos, lay-out y parámetros del algoritmo.
2: nLot ← nLotMin
3: while nLot < nLotMax
4: t ← 0;
5: P(t) ← InitPop(Entrada);
6: FitP(t) ← EvalPop(P(t));
7: For t ← 1 a MaxNumGen
8:   Q(t) ← SelecBreeders (P(t), FitP(t));
9:   Q(t) ← Crossover(Q(t));
10:  Q(t) ← Mutation(Q(t));
11:  FitQ(t) ← EvalPop(Q(t));
12:  P(t) ← SelSurviv(P(t), Q(t), FitP(t), FitQ(t));
13:  FitP(t) ← EvalPop(P(t));
14:  if TermCond(P(t), FitP(t))
15:    break;
16: end
17: end
18: nLot ← nLot + 1;
19: end
    
```

---

Fig. 2. Seudo-código del algoritmo propuesto



## 5 Experimento computacional

Con el propósito de evaluar la calidad de las soluciones y el rendimiento del algoritmo propuesto se utilizaron las instancias propuestas en [11]. Se resolvieron tres instancias. Una instancia mediana (DS1/M1), una grande (DS2/M1) y una muy grande (DS3/M1) (Tabla 4).

**Tabla 4.** Características de las instancias [11]

	DS1/M1	DS2/M1	DS3/M1
Número de pedidos	40	80	200
Número de artículos diferentes	80	160	300
Peso total medio (kg.)	13.704	37.152	158.784
Capacidad del equipo de picking (kg.)	10.000	10.000	20.000

Para las instancias, se suponen las siguientes distribuciones de probabilidad. La cantidad del artículo  $p$  en el pedido del cliente  $i$  sigue una distribución uniforme entre 1 y 10, es decir,  $q_{i,p} \sim U(1, \dots, 10)$ . El número de artículos diferentes en el pedido de un cliente  $i$  sigue una distribución normal, con media 10 y desviación estándar 5, es decir  $|\mathcal{P}_i| \sim N(10, 5)$ . El plazo en el que debe estar finalizado el armado de un pedido  $i$ , sigue una distribución uniforme, estos momentos se consideran discretizados en segundos y están entre las 10:00 am y las 18:00 pm, esto es,  $t_i \sim U(36000, \dots, 64800)$ . El peso unitario de cada artículo  $p$ , sigue una distribución uniforme entre 8 y 24 kg., es decir,  $w_p \sim U(8, \dots, 24)$ . Respecto a los parámetros de los equipos de picking, se supone una velocidad promedio de desplazamiento de  $v = 2 \text{ m/s}$ , un tiempo medio para la recolección de cada artículo de  $t_{pick} = 15 \text{ seg.}$ , un costo de desplazamiento por unidad de tiempo de  $\zeta = 0,05 \text{ u.m.}$  y una capacidad de carga ( $Cap$ ) que depende del tipo de instancia que se esté considerando. En cuanto a los parámetros de la función objetivo, se considera una penalización por unidad de tiempo de anticipación en el plazo para el armado de un pedido de  $\alpha = 0.5$  y una penalización por unidad de tiempo de demora en el cumplimiento del plazo para la finalización en el armado de un pedido de  $\beta = 1$ . Se utilizará la estrategia para acotar el espacio de búsqueda propuesta en [11] con el objetivo de que el desempeño del algoritmo sea comparable con los resultados obtenidos en esta publicación. Esto es, determinar un límite inferior,  $|\mathcal{R}|_{min}$ , y un límite superior,  $|\mathcal{R}|_{max}$ , para la cantidad de lotes en los que se agrupan los artículos a recoger, es decir,  $|\mathcal{R}|_{max} \leq |\mathcal{R}| \leq |\mathcal{R}|_{min}$ . Estos límites están dados por  $|\mathcal{R}|_{min} = (\varphi_1 \cdot \sum_{p \in \mathcal{P}} w_p) / Cap$  y  $|\mathcal{R}|_{max} = (\varphi_2 \cdot \sum_{p \in \mathcal{P}} w_p) / Cap$ . Donde  $\varphi_1$  y  $\varphi_2$  son constantes decididas por los usuarios, y  $\varphi_2 \geq \varphi_1$ . El número máximo de iteraciones se fijó 500, el tamaño de la población en 150, el tamaño del torneo en 2, los parámetros de los límites del espacio de búsqueda  $\varphi_1 = 2$  y  $\varphi_2 = 4$ , la probabilidad de cruce en 0.9, la probabilidad de mutación 0.15, la cantidad de individuos que pertenecen a la elite es el 5% de la población. Se utilizará una PC con un procesador Intel Core i7 3.00 GHz y 8 GB de RAM.

## 6 Resultados

En esta sección se presentan los resultados obtenidos con el algoritmo propuesto al resolver las instancias DS1/M1, DS2/M1 y DS3/M1. Los mismos se compararon con los resultados extraídos de [11] (Tabla 5,  $D_{Total}$ : Distancia total (metros);  $n_{Lot}$ : Número óptimo de lotes (cantidad);  $D_{Promedio}$ : Distancia promedio (metros);  $D_{\sigma}$ : Desvío estándar (metros);  $T_{Demora}$ : Tiempo de demora en el armado de los pedidos (segundos);  $C_{Total}$ : Costo total (unidades monetarias);  $T_{CPU\ promedio}$ : Tiempo de corrida en CPU promedio (segundos). Se realizaron 30 corridas para cada una de las instancias.

Tabla 5. Resultados

	Resultados extraídos de [11]			Algoritmo Evolutivo		
	DS1/M1	DS2/M1	DS3/M1	DS1/M1	DS2/M1	DS3/M1
$D_{Total}$	1304.00	3569.00	16945.00	1266.00	3215.00	14548.00
$n_{Lot}$	8.00	11.00	27.00	8.00	12.00	27.00
$D_{Media}$	163.00	324.46	627.59	158.25	267.91	538.81
$D_{\sigma}$	3.70	25.17	18.69	2.90	23.41	21.75
$T_{Demora}$	1181.00	4704.00	15481.00	1158.70	4801.91	15621.31
$C_{Total}$	1092.60	3104.83	9207.13	1090.78	2961.89	8898.01
$T_{CPU\ promedio}$	753.60	2629.90	5785.50	605.70	2121.90	4474.20

La  $D_{Total}$  mejora en todas las instancias (DS1/M1, DS2/M1 y DS3/M1, 2.91%, 9.92% y 14.15%, respectivamente). Para las instancias DS1/M1 y DS3/M1, el  $n_{Lot}$  fue el mismo que en el caso de los resultados extraídos de [11], mientras que en la instancia DS2/M1 el  $n_{Lot}$  fue mayor. Para la instancia DS1/M1, el  $T_{Demora}$  mejora en un 1.89%, mientras que en las instancias DS2/M1 y DS3/M1 empeora en 2.08% y 0.91% respectivamente. Finalmente, el  $T_{CPU\ promedio}$  en todas las instancias fue menor. En resumen, en la figura 3, se muestran las mejoras porcentuales obtenidas con el algoritmo propuesto en este trabajo con respecto a los resultados extraídos de [11].

## 7 Conclusiones

En este trabajo se presentó un algoritmo evolutivo para resolver de manera combinada el Order Batching Problem y el Order Picking Problem. El algoritmo propuesto para resolver el problema, presentó un novedoso criterio de representación con dos genomas, que permite la incorporación directa de conocimiento específico del problema, dando mayor flexibilidad y facilitando la consideración de restricciones en el espacio de búsqueda. El algoritmo fue testeado en instancias simuladas de distintos tamaños. Se observó que el algoritmo propuesto mejoró los resultados extraídos de [11]. A partir de este trabajo, se presenta como línea futura de investigación, la consideración de este problema bajo una plataforma cross-dock y se le dará un abordaje multi-objetivo.

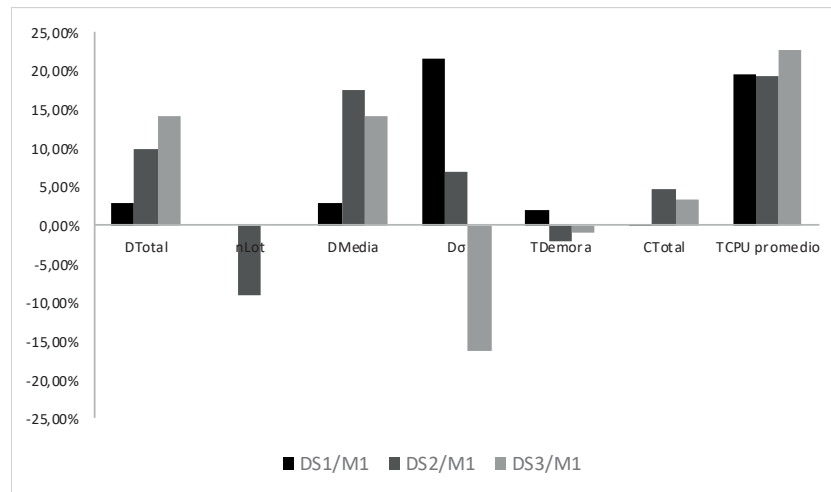


Fig. 3. Mejoras % con respecto a los resultados extraídos de [11]

## Referencias

- [1] De Koster, R., Van der Poort, E. S., & Wolters, M. (1999). Efficient order batching methods in warehouses. *International Journal of Production Research*, 37(7), 1479-1504.
- [2] De Koster, R., Le-Duc, T., & Roodbergen, K. J. (2007). *Design and control of warehouse order picking: A literature review*. *European Journal of Operational Research*, 182(2), 481-501.
- [3] Miguel, F., Frutos, M., Tohmé, F., & Rossit, D. A. (2019). A memetic algorithm for the integral OBP/OPP problem in a logistics distribution center. *Uncertain Supply Chain Management*, 7(2019), 203-214.
- [4] Chen, M. C., & Wu, H. P. (2005). An association-based clustering approach to order batching considering customer demand patterns. *Omega*, 33(4), 333-343.
- [5] Henn, S., & Koch, S. Y. (2012). Order batching in order picking warehouses: a survey of solution approaches. *Warehousing in the Global Supply Chain*, 105-137
- [6] Henn, S., & Schmid, V. (2013). Metaheuristics for order batching and sequencing in manual order picking systems. *Computers & Industrial Engineering*, 66(2), 338-351.
- [7] Henn, S., & Wäscher, G. (2012). Tabu search heuristics for the order batching problem in manual order picking systems. *European Journal of Operational Research*, 222(3), 484-494.
- [8] Henn, S., Koch, S., Doerner, K., Strauss, C., & Wäscher, G. (2010). Metaheuristics for the order batching problem in manual order picking systems. *Business Research*, 3(1), 82-105.
- [9] Hwang, H., & Kim, D. (2005). Order-batching heuristics based on cluster analysis in a low-level picker-to-part warehousing system. *European Journal of Operational Research*, 43(17), 3657-3670.
- [10] Rana, K. (1991). Order-picking in narrow-aisle warehouse. *International Journal of Physical Distribution and Logistics*, 20(2), 9-15.

- [11] Tsai, C. Y., Liou, J. J., & Huang, T. M. (2008). Using a multiple-GA method to solve the batch picking problem: considering travel distance and order due time. *International Journal of Production Research*, 46(22), 6533-6555.
- [12] Grosse, E. H., & Glock, C. H. (2015). The effect of worker learning on manual order picking processes. *International Journal of Production Economics*, 170(C), 882-890.
- [13] Lam, C. H., Choy, K. L., Ho, G. T., & Lee, C. K. (2014). An order-picking operations system for managing the batching activities in a warehouse. *International Journal of Systems Science*, 45(6), 1283-1295.
- [14] Van Gils, T., Ramaekers, K., Braekers, K., Depaire, B., & Carisa, A. (2018). Increasing order picking efficiency by integrating storage, batching, zone picking, and routing policy decisions. *International Journal of Production Economics*, 197, 243-261.
- [15] Ho, Y. C., & Tseng, Y. Y. (2006). A study on order-batching methods of order-picking in a distribution centre with two cross-aisles. *International Journal of Production Research*, 44(17), 3391-3471.
- [16] Scholz, A., Schubert, D., & Wäscher, G. (2017). Order picking with multiple pickers and due dates - Simultaneous solution of Order Batching, Batch Assignment and Sequencing, and Picker Routing Problems. *European Journal of Operational Research*, 263(2), 461-478.
- [17] Ardjmand, E., Shakeri, H., Singh, M., & Bajgiran, O. S. (2018). Minimizing order picking makespan with multiple pickers in a wave picking warehouse. *International Journal of Production Economics*, 206, 169-183.
- [18] Öztürkoğlu, Ö., & Hoser, D. (2018). A discrete cross aisle design model for order-picking warehouses. *European Journal of Operational Research*, 275(2), 411-430.
- [19] Tappia, E., Roy, D., Melacini, M., & De Koster, R. (2018). Integrated storage-order picking systems: Technology, performance models, and design insights. *European Journal of Operational Research*, 274(3), 947-965.
- [20] Lu, W., McFarlane, D., Giannikas, V., & Zhang, Q. (2016). An algorithm for dynamic order-picking in warehouse operations. *European Journal of Operational Research*, 248(1), 107-122.
- [21] Petersen, C. G. (1997). An evaluation of order picking routing policies. *International Journal of Operations & Production Management*, 17(11), 1098-1111.
- [22] Žulj, I., Glock, C. H., Grosse, E. H., & Schneider, M. (2018). Picker routing and storage-assignment strategies for precedence-constrained order picking. *Computers & Industrial Engineering*, 123, 338-347.
- [23] Theys, C., Bräysy, O., Dullaert, W., & Raa, B. Using a TSP heuristic for routing order pickers in warehouses. *European Journal of Operational Research*, 755-763, 2010.
- [24] Wetzel, A. Evaluation of the effectiveness of genetic algorithms in combinatorial optimization. Pittsburgh, Philadelphia, USA: University of Pittsburgh, 1983.
- [25] Whitley, L. D., Starkweather, T., & Fuquay, D. (1989). Scheduling problems and traveling salesmen: The genetic edge recombination operator. Proceedings of the 3rd International Conference on Genetic Algorithms, San Francisco, Morgan Kaufmann Publishers Inc., 133-140.
- [26] Fogel, D. B. (1988). An evolutionary approach to the traveling salesman problem. *Biological Cybernetics*, 60(2), 139-144.