



Desafío SBPO 2025 – Descripción del Problema

El Problema de la Selección de Órdenes Óptima

Enero de 2025

Índice

1. Introducción	2
2. Descripción del problema	2
2.1. Definiciones y notaciones	2
2.2. Conjuntos	4
2.3. Constantes	4
2.4. Objetivo del problema y restricciones	4
3. Ejemplo	5

1. Introducción

El recorrido de un paquete dentro de la red de entregas de MERCADO LIVRE comienza mucho antes de la fase de transporte. Una vez que una Orden es generada por un cliente, entra en un *backlog* de órdenes a preparar y se le asigna un plazo de procesamiento basado en el tiempo de entrega prometido. La siguiente etapa involucra la recolección en el almacén (warehouse) de los items que componen la orden para su posterior embalaje y transporte. Obviamente, la recolección de items ((llamado comúnmente también como proceso de picking de items) de una orden individual es menos eficiente que la recolección realizada al agrupar subconjuntos de órdenes, llamados *waves* de órdenes o, simplemente, *waves*. Esto se debe a que la recolección simultánea de órdenes permite aprovechar las sinergias entre los lugares de recolección de los items de las diferentes órdenes. El proceso de seleccionar sucesivamente del *backlog* el siguiente subconjunto de órdenes a ser recolectados, a través de una o varias rutas, se conoce como **recolección de órdenes en waves**. Sin embargo, para maximizar la productividad de la operación de recolección, la selección de las órdenes que compondrán la próxima *wave* debe hacerse de manera criteriosa. De esta manera, el enfoque de este desafío es proponer algoritmos para resolver el problema de encontrar una *wave* de órdenes óptima, maximizando la productividad de la recolección.

2. Descripción del problema

En esta sección, describiremos el problema en detalle.

2.1. Definiciones y notaciones

Comenzamos dando definiciones y notaciones básicas.

- **Elemento:** un producto que puede ser solicitado.
- **Orden:** una lista específica de elementos solicitados por un cliente junto con sus cantidades.
- **Recolección/picking:** el proceso de recuperar productos del almacén.
- **Backlog:** el conjunto de órdenes cuyos elementos aún no han sido recolectados.
- **Wave:** un subconjunto de órdenes del *backlog* seleccionados para ser procesados (o enrutados) juntos, con el objetivo de maximizar la productividad de las operaciones de recolección.
- **Pasillo:** para fines de este desafío, un almacén puede ser concebido como un único piso rectangular conteniendo filas paralelas de estanterías compuestas por múltiples capas de estantes, cada una de ellas con un *dirección*

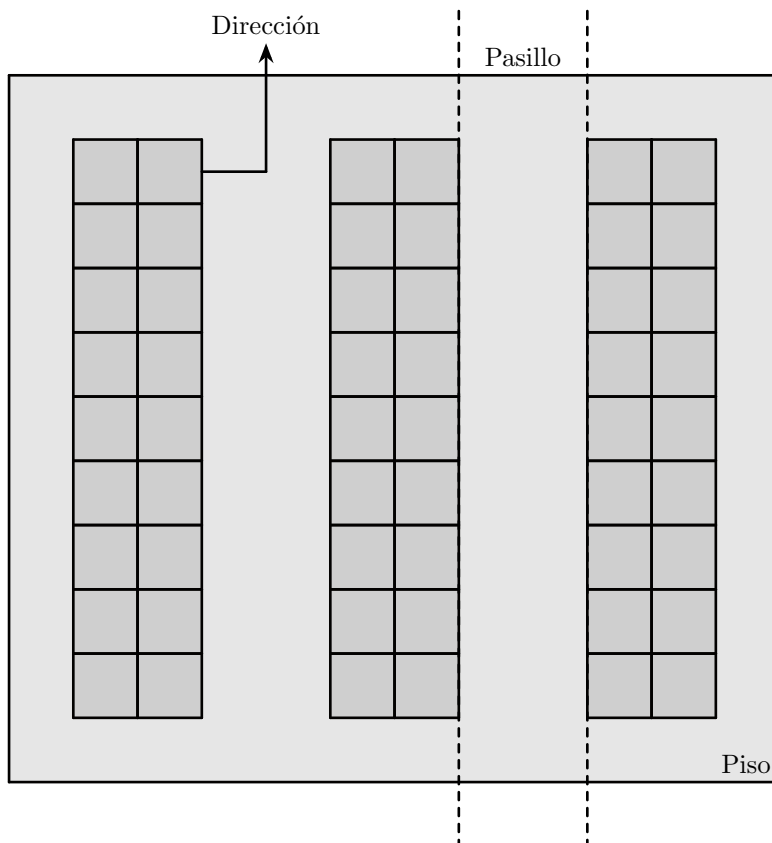


Figura 1: Diseño del piso de un almacén con sus elementos básicos.

asociada. En estos estantes se almacenan los elementos que pueden formar parte de un Órden. Los espacios libres entre dos filas de estanterías consecutivas son llamados *pasillos* y permiten la circulación de personas o equipos que realizan la recolección de los elementos presentes en un Órden. En la Figura 1 se ve un ejemplo de un diagrama representando el piso del almacén.

A continuación se presentan los conjuntos y constantes necesarias para definir el problema objetivo de este desafío.

2.2. Conjuntos

- \mathcal{O} Conjunto de órdenes en *backlog*.
- \mathcal{I}_o Subconjunto de elementos solicitados por el Orden $o \in \mathcal{O}$.
- \mathcal{I} Conjunto de elementos, donde $\mathcal{I} = \bigcup_{o \in \mathcal{O}} \mathcal{I}_o$.
- \mathcal{A}_i Subconjunto de pasillos conteniendo al menos una unidad del elemento i .
- \mathcal{A} Conjunto de pasillos, donde $\mathcal{A} = \bigcup_{i \in \mathcal{I}} \mathcal{A}_i$.

2.3. Constantes

- u_{oi} Número de unidades del elemento $i \in \mathcal{I}$ solicitado por el Orden $o \in \mathcal{O}$.
- u_{ai} Número de unidades del elemento $i \in \mathcal{I}$ disponibles en el pasillo $a \in \mathcal{A}$.
- LB / UB Límite inferior / superior del *tamaño* de la *wave* (definido a continuación).

2.4. Objetivo del problema y restricciones

El objetivo del problema de recolección de órdenes en *waves* es seleccionar un subconjunto de órdenes del *backlog*, es decir, la *wave*, y un subconjunto de pasillos (lugares de recolección) que maximicen la productividad del proceso de recolección. Esto se consigue al crear condiciones favorables para la generación de rutas de recolección eficientes. Para este desafío, vamos a explorar una de estas condiciones, que es concentrar los elementos de la *wave* en la menor cantidad posible de pasillos, respetando las restricciones descritas a continuación.

De manera más formal, sea $O' \subset \mathcal{O}$ el subconjunto de órdenes en la *wave*, y $A' \subset \mathcal{A}$ el subconjunto de pasillos que serán visitados para recolectar todos los elementos en O' . El objetivo es maximizar el número de elementos recolectados por pasillo visitado, matemáticamente representado por:

$$\text{máx} \frac{\sum_{o \in O'} \sum_{i \in \mathcal{I}_o} u_{oi}}{|A'|} \quad (1)$$

Existen esencialmente tres familias de restricciones. Las dos primeras establecen límites operacionales en el número total de unidades que necesitan ser

recolectadas para las órdenes incluidas en la *wave*. Se expresan como:

$$\sum_{o \in O'} \sum_{i \in I_o} u_{oi} \geq \text{LB}, \quad (2)$$

$$\sum_{o \in O'} \sum_{i \in I_o} u_{oi} \leq \text{UB}. \quad (3)$$

La tercera familia de restricciones asegura que los pasillos seleccionados tengan almacenamiento suficiente para cada uno de los elementos incluidos en las órdenes dentro de la *wave*. Esto se traduce en:

$$\sum_{o \in O'} u_{oi} \leq \sum_{a \in A'} u_{ai}, \quad \forall i \in I_o \text{ con } o \in O'. \quad (4)$$

Cualquier par (O', A') de $\mathcal{O} \times \mathcal{A}$ que satisfaga las restricciones (2), (3) y (4) constituye una *solución factible* para el problema de recolección de órdenes en *wave*. En este contexto, la *wave* O' también se considera *factible*.

Sin embargo, observe que puede haber múltiples candidatos para el subconjunto A' de pasillos seleccionados que satisfagan la restricción (4) para la *wave* dada. Naturalmente, el objetivo es identificar entre estos subconjuntos uno que maximice la función objetivo (1).

3. Ejemplo

Considere el siguiente ejemplo con 5 órdenes, 5 elementos diferentes y 5 pasillos. El número de unidades de cada elemento solicitado por cada Orden en el *backlog* se presenta en la Tabla 1, mientras que la disponibilidad de elementos en cada pasillo se presenta en la Tabla 2. Los límites inferior y superior para el tamaño de la *wave* son 5 y 12, respectivamente.

Orden	Elemento 0	Elemento 1	Elemento 2	Elemento 3	Elemento 4
0	3	0	1	0	0
1	0	1	0	1	0
2	0	0	1	0	2
3	1	0	2	1	1
4	0	1	0	0	0

Cuadro 1: Número de unidades de cada elemento solicitado por cada Orden.

Pasillo	Elemento 0	Elemento 1	Elemento 2	Elemento 3	Elemento 4
0	2	1	1	0	1
1	2	1	2	0	1
2	0	2	0	1	2
3	2	1	0	1	1
4	0	1	2	1	2

Cuadro 2: Número de unidades de cada elemento disponible en cada pasillo.

Algunos ejemplos de *waves* factibles, y sus asignaciones de pasillos, son:

- Órdenes 0 y 4, usando pasillos 0 y 1.
 - Total de unidades: 5;
 - Número de pasillos: 2;
 - Valor objetivo: 2.5.
- Órdenes 0, 2 y 3, usando pasillos 1, 3 y 4.
 - Total de unidades: 12;
 - Número de pasillos: 3;
 - Valor objetivo: 4.
- Órdenes 0, 1, 2 y 4, usando pasillos 1 y 3.
 - Total de unidades: 10;
 - Número de pasillos: 2;
 - Valor objetivo: 5.

En la última solución, el número promedio de elementos recolectados por pasillo (el valor objetivo) es 5, mayor que los de las otras soluciones mostradas, haciendo que la última solución sea preferible en comparación con las demás. De hecho, se puede mostrar que esta solución es óptima para la instancia dada.