

HEURISTICAS PARA PROBLEMAS DE ENRRUTAMIENTO

Modelo de Enrutamiento para la recolección de pedidos en un almacén convencional con líneas de dos bloques iguales, mediante los algoritmos heurísticos S-shape y la brecha más grande.

Leididy Aparicio Macias y Ayra Valentina Chaparro Quintero

Proyecto de grado para optar el título de Ingeniería Industrial

Director

Vlakxmir Robles Marín

Magister en Ingeniería

Universidad Industrial de Santander

Facultad de Ingenierías Físico – Mecánicas

Escuela de Estudios Industriales y Empresariales

Bucaramanga

2023

Tabla de contenido

Introducción	14
1 Generalidades del Proyecto.....	16
1.1 Planteamiento del problema.....	16
1.2 Justificación del proyecto.....	18
1.3 Objetivos	20
1.3.1 Objetivo general.....	20
1.3.2 Objetivos específicos	20
1.4 Metodología	20
1.4.1 Fase 1. Realizar una revisión bibliográfica en general sobre el problema a tratar	20
1.4.2 Fase 2. Definir las características del problema.....	20
1.4.3 Fase 3. Adaptar un modelo matemático.....	21
1.4.4 Fase 4. Definir y desarrollar los algoritmos a usar	21
1.4.5 Fase 5. Evaluar los algoritmos	21
1.4.6 Fase 6. Documentación	22
2 Revisión de literatura	22
2.1 Análisis Bibliométrico	22
2.1.1 Scopus.....	23
2.1.1.1 Indicadores básicos.	23
2.1.1.1.1 Documentos por año.	23
2.1.1.1.2 Área temática.	24
2.1.1.2.1 Autores.	25
2.1.1.2.2 Países.....	25

HEURISTICAS PARA PROBLEMAS DE ENRRUTAMIENTO

2.1.1.3 <i>Indicadores de relación.</i>	26
2.1.1.3.1 Palabras claves.	27
2.1.1.3.2 Citación de autores.	27
2.1.2 Web of Science	28
2.1.2.1 Indicadores básicos.	28
2.1.2.1.1 Documentos por año.	28
2.1.2.1.2 Área temática.	29
2.1.2.1.3 Autores.	30
2.1.2.1.4 Países.	31
2.1.2.2 Indicadores de relación.	31
2.1.2.2.1 Palabras clave.	32
2.1.2.2.2 Citación de autores.	32
2.1.3 ScienceDirect	33
2.1.3.1 Indicadores básicos.	33
2.1.3.1.1 Documentos por año.	33
2.1.3.1.2 Área temática.	34
2.1.3.2 Indicadores de relación.	35
2.1.3.2.1 Palabras claves.	35
2.2 Análisis preliminar de la literatura	35
3 Marco Teórico	42
3.1 Logística	42
3.1.1 La logística de almacenamiento	43
3.2 Almacenamiento	43
3.2.1 Análisis ABC	44
3.2.2 Métodos de solución	44

HEURISTICAS PARA PROBLEMAS DE ENRRUTAMIENTO

3.2.3.1	Proceso de Jerarquía Analítica.....	44
3.2.3.2	Optimización Lineal.....	44
3.2.3.3	Redes Neuronales.....	45
3.2.3.4	Agrupación.....	45
3.3	Preparación de pedidos	45
3.4	Optimización de rutas de “Picking”	45
3.5	Diseño del almacén	46
3.5.1	Almacenes convencionales	46
3.5.2	Almacén de dos bloques iguales	47
3.5.3	Almacenes no convencionales	47
3.5.4	Almacenes generales.....	48
3.6	Optimización combinatoria.....	49
3.6.1	Problema del agente viajero	49
3.6.2	Teoría de la complejidad computacional	50
3.7	Algoritmos de Solución	52
3.7.1	Algoritmos exactos	52
3.7.2	Heurística	52
3.7.2.1	S-Shape.....	53
3.7.2.2	La brecha más grande.....	54
3.7.2.3	Midpoint.....	55
3.7.2.4	Return.....	55
3.7.2.5	Combined.....	56
3.7.3	Metaheurística	57
3.8	Herramienta computacional Python.....	58
4	Descripción del problema y modelo matemático.....	58

HEURISTICAS PARA PROBLEMAS DE ENRRUTAMIENTO

4.1 Descripción del problema	58
4.2 Modelo Matemático	60
4.2.1 Formulación del problema	60
5 Diseño de los algoritmos.....	67
6 Descripción del algoritmo.....	69
6.1 Estructura básica	69
6.2 Paso a paso	70
6.2.1 Parametros.py	70
6.2.2 Main.py.....	70
6.2.3 Objetos.py.....	73
6.2.4 Algoritmo.py.....	75
6.2.5 Graphing.py.....	78
7 Evaluación del algoritmo	86
7.1 Ruta realizada por cada heurística.....	86
7.1.1 Graficas arrojadas de la heurística s-shape	87
7.1.2 Graficas arrojadas de la heurística la brecha más grande	90
7.2 Resultados diseño experimental.....	93
8 Resultados	101
8.1 Resultados tipo de almacenamiento y comparación entre heurísticas s-shape y la brecha más grande.....	102
8.1.1 Pequeñas instancias.....	102
8.1.2 Medianas instancias	105
8.1.3 Grandes instancias.....	108
8.1.4 Análisis de Varianza para Pequeñas Instancias	112
8.1.5 Análisis de Varianza para medianas Instancias	114

HEURISTICAS PARA PROBLEMAS DE ENRRUTAMIENTO

8.1.6 Análisis de Varianza para grandes Instancias	116
9 Conclusiones	118
10 Recomendaciones.....	120
Referencias Bibliográficas	121

HEURISTICAS PARA PROBLEMAS DE ENRRUTAMIENTO

Lista de tabla

Tabla 1 Estructura del proyecto	15
Tabla 2. Algoritmo brecha más grande	67
Tabla 3. Algoritmo S-Shape.....	68
Tabla 7. Análisis de variancia para instancias pequeñas.	93
Tabla 8. Análisis de variancia para instancias medianas.	96
Tabla 9. Análisis de variancia para instancias grandes.	99
Tabla 10. Resultados tipo de almacenamiento para un almacén pequeño con 20 órdenes...	102
Tabla 11. Resultados tipo de almacenamiento para un almacén pequeño con 40 órdenes...	103
Tabla 12. Resultados tipo de almacenamiento para un almacén pequeño con 60 órdenes...	104
Tabla 13. Resultados tipo de almacenamiento para un almacén mediano con 20 órdenes...	105
Tabla 14. Resultados tipo de almacenamiento para un almacén mediano con 40 órdenes...	106
Tabla 15. Resultados tipo de almacenamiento para un almacén mediano con 60 órdenes...	107
Tabla 16. Resultados tipo de almacenamiento para un almacén grande con 20 órdenes.	108
Tabla 17. Resultados tipo de almacenamiento para un almacén grande con 40 órdenes.	109
Tabla 18 Resultados tipo de almacenamiento para un almacén grande con 60 órdenes.	110
Tabla 19. Consolidado de análisis de varianza tipo de almacenamiento ABC/NO ABC para instancias pequeñas.....	112
Tabla 20. Consolidado de valores diagrama cajas y bigotes tipo de almacenamiento ABC/NO ABC para instancias pequeñas.....	113
Tabla 21. Consolidado análisis de varianza entre heurísticas para instancias pequeñas.	113
Tabla 22. Consolidado de valores diagrama cajas y bigotes heurísticas.	114
Tabla 23. Consolidado de análisis de varianza tipo de almacenamiento ABC/ NO ABC para instancias medianas.....	114

HEURISTICAS PARA PROBLEMAS DE ENRRUTAMIENTO

Tabla 24. Consolidado de valores diagrama cajas y bigotes tipo de almacenamiento ABC/NO ABC para instancias medianas.....	115
Tabla 25. Consolidado análisis de varianza entre heurísticas para instancias medianas.	115
Tabla 26. Consolidado de valores diagrama cajas y bigotes heurísticas.	115
Tabla 27. Consolidado de análisis de varianza tipo de almacenamiento ABC/NO ABC para instancias grandes.	116
Tabla 28. Consolidado de valores diagrama cajas y bigotes tipo de almacenamiento ABC/NO ABC para instancias grandes.	116
Tabla 29. Consolidado análisis de varianza entre heurísticas para instancias grandes.	117
Tabla 30. Consolidado de valores diagrama cajas y bigotes heurísticas.	117
Tabla 31. Número de veces que se aprecia menor distancia por almacén para cada heurística.	118

HEURISTICAS PARA PROBLEMAS DE ENRRUTAMIENTO

Lista de figuras

Figura 1. Desarrollo del tema con el pasar de los años en Scopus	24
Figura 2. Área temática en Scopus	24
Figura 3. Autores más relevantes en Scopus	25
Figura 4. Países destacados en tema de investigación en Scopus.....	26
Figura 5. Concurrencia palabras clave en Scopus	27
Figura 6. Autores más importantes teniendo en cuenta las citaciones en Scopus	28
Figura 7. Desarrollo del tema con el pasar de los años en Web of Science.....	29
Figura 8. Área temática en Web of Science.....	30
Figura 9. Autores más relevantes en Web of Science.....	30
Figura 10. Países destacados en tema de investigación en Web of Science	31
Figura 11. Concurrencia palabras clave en Web of Science.....	32
Figura 12. Autores más importantes teniendo en cuenta las citaciones en Web of Science....	33
Figura 13. Desarrollo del tema a lo largo de los años en ScienceDirect	34
Figura 14. Área temática en ScienceDirect.....	34
Figura 15. Concurrencia pablaras claves en ScienceDirect	35
Figura 16. Línea del tiempo de documentos relacionados al enrutamiento para la recolección de pedidos en un almacén convencional con líneas de dos bloques iguales.....	42
Figura 17. Diseño almacén de dos bloques iguales	47
Figura 18. Ejemplo recolección de una ruta s-shape	53
Figura 19. Ejemplo recolección de una ruta la brecha más grande	54
Figura 20. Ejemplo recolección de una ruta midpoint.....	55
Figura 21. Ejemplo recolección de una ruta return.....	55
Figura 22. Ejemplo recolección de una ruta combined.....	56

HEURISTICAS PARA PROBLEMAS DE ENRRUTAMIENTO

Figura 23. ruta de la heurística s-shape en pequeñas instancias.	87
Figura 24. ruta de la heurística s-shape en medianas instancias.	88
Figura 25. ruta de la heurística s-shape en grandes instancias.	89
Figura 26. ruta de la heurística brecha más grande en pequeñas instancias.	90
Figura 27. ruta de la heurística brecha más grande en medianas instancias.	91
Figura 28. ruta de la heurística brecha más grande en grandes instancias.	92
Figura 29. Diagrama cajas y bigotes de heurísticas para instancias pequeñas.	94
Figura 30. Diagrama de dispersión tipo de almacenamiento para instancias pequeñas.	95
Figura 31. Diagrama cajas y bigotes de heurísticas para instancias pequeñas.	95
Figura 32. Diagrama de dispersión heurísticas para instancias pequeñas.	96
Figura 33. Diagrama cajas y bigotes de tipo de almacenamiento para instancias medianas. ...	97
Figura 34. Diagrama de dispersión tipo de almacenamiento para instancias medianas.	97
Figura 35. Diagrama cajas y bigotes de heurísticas para instancias medianas.	98
Figura 36. Diagrama de dispersión heurísticas para instancias medianas.	98
Figura 37. Diagrama cajas y bigotes de tipo de almacenamiento para instancias grandes.	99
Figura 38. Diagrama de dispersión tipo de almacenamiento para instancias grandes.	100
Figura 39. Diagrama de dispersión heurísticas para instancias grandes.	100
Figura 40. Diagrama de dispersión heurísticas para instancias grandes.	101

HEURISTICAS PARA PROBLEMAS DE ENRRUTAMIENTO

Lista de apéndices

Ver apéndices en la carpeta comprimida adjunta

Apéndice A. Software modelo de enrutamiento para comparación de algoritmos heurísticos tales como la brecha más grande y S-Shape.

Apéndice B. Diferencia en las distancias entre las dos heurísticas

Apéndice C. Artículo publicable

HEURISTICAS PARA PROBLEMAS DE ENRRUTAMIENTO

Resumen

Título: Modelo de Enrutamiento para la recolección de pedidos en un almacén convencional con líneas de dos bloques iguales, mediante los algoritmos heurísticos S-shape y la brecha más grande*

Autores: Ayra Valentina Chaparro Quintero y Leididy Aparicio Macias**

Palabras claves: enrutamiento, recolección de pedidos, logística, algoritmos, S-shape y la brecha más grande.

Descripción:

La pandemia del covid-19 impactó los mercados mundiales donde el e-commerce creció y los sistemas de distribución retoman importancia dada la necesidad de mejorar los tiempos de entrega por ello, el presente documento realiza un modelo de enrutamiento con variaciones en el número de pickers, un único depósito y clasificación ABC es de tipo NP-hard mediante métodos heurísticos s-shape y la brecha más grande, los cuales dependen del problema y el resultado en la mayoría de los casos no es óptimo sin embargo, según la revisión de literatura al comparar los métodos exactos con heurísticas los documentos encontrados afirman que las soluciones factibles son mejores que las soluciones óptimas y compara los algoritmos encontrando soluciones factibles de mínimas distancias con un almacén convencional con líneas de dos bloques iguales.

Como resultado se presenta que la distancia obtenida con la heurística s-shape respecto con la brecha más grande es menor para los tres tamaños tanto de almacenes como de órdenes.

En cuanto al análisis de varianza del almacén pequeño se evidencia que el factor (tipo de almacenamiento) ABC o NO ABC, no es significativo en la distancia y para el factor heurística s-shape y la brecha más grande, es significativo en la distancia. Para el almacén mediano el factor (tipo de almacenamiento) ABC o NO ABC, no es significativo en la distancia y en cuanto al factor heurística s-shape y la brecha más grande, es significativo en la distancia, y existe una diferencia relevante entre las distancias promedio teniendo en cuenta las heurísticas. En el almacén grande el factor (tipo de almacenamiento) ABC o NO ABC, no es significativo en la distancia, pero en el factor heurística s-shape y la brecha más grande, es significativo en la distancia, y existe una diferencia relevante entre las distancias promedio.

*Trabajo de Grado

**Facultad fisicomecánica. Escuela de estudios industriales y empresariales. Director Vlaxmir Robles Marín.

Abstract

Title: Routing model for the picking of orders in a conventional warehouse with lines of two equal blocks, using the heuristic algorithms S-shape and the largest gap.

Authors: Ayra Valentina Chaparro Quintero y Leididy Aparicio Macias**

Key words: routing, order picking, logistics, algorithms, S-shape and the biggest gap

Description:

The covid-19 pandemic impacted world markets where e-commerce grew and distribution systems regain importance given the need to improve delivery times, therefore, this document makes a routing model with variations in the number of pickers , a single repository and ABC classification is of the NP-hard type by heuristic methods s-shape and the largest gap, which depend on the problem and the result in most cases is not optimal, however, according to the literature review When comparing the exact methods with heuristics, the documents found affirm that the feasible solutions are better than the optimal solutions and compare the algorithms finding feasible solutions of minimum distances with a conventional warehouse with lines of two equal blocks.

As a result, it appears that the distance obtained with the s-shape heuristic with respect to the largest gap is smaller for the three sizes of both warehouses and orders.

Regarding the analysis of variance of the small warehouse, it is evident that the factor (type of storage) ABC or NOT ABC, is not significant in the distance and for the heuristic factor s-shape and the largest gap, it is significant in the distance. For the median warehouse, the factor (type of storage) ABC or NO ABC, is not significant in the distance and as for the heuristic factor s-shape and the largest gap, it is significant in the distance, and there is a relevant difference between the average distances taking into account the heuristics. In the large warehouse, the factor (type of storage) ABC or NOT ABC, is not significant in the distance, but in the heuristic factor s-shape and the largest gap, it is significant in the distance, and there is a relevant difference between the average distances.

*Bachelor Thesis

**Facultad fisicomecánica. Escuela de estudios industriales y empresariales. Director Vlaxmir Robles Marín.

HEURISTICAS PARA PROBLEMAS DE ENRRUTAMIENTO

Introducción

La pandemia del covid-19 durante 2020 golpeó los mercados mundiales, y aunque algunos mercados se vieron afectados por el frenazo en la actividad económica, el e-commerce, por el contrario, creció vertiginosamente. De acuerdo con el más reciente informe de la Cámara Colombiana de Comercio Electrónico, el e-commerce experimentó un crecimiento acelerado en 2020 en respuesta a las medidas de confinamiento establecidas en el país, experimentando un alza de 11% semanal en los momentos más críticos de la cuarentena. (Toro,2020).

Al aumentar exponencialmente las ventas online, se generan desafíos sobre los sistemas de distribución, como el de mantener e incluso mejorar los tiempos de entrega, por consiguiente, se requieren análisis y mejoras en una serie de actividades desde la entrada del pedido como seleccionar los productos comprados, empaquetarlos adecuadamente y realizar el respectivo envío a sus clientes, este proceso se denomina picking.

Para entender el impacto financiero de esta actividad en una bodega, se ha podido establecer que, en una instalación mal proyectada, los costos del picking en el almacén pueden superar el 60% de los costos operativos totales del almacén (Tompkins et al.,2003).

Para reducir al mínimo tolerable es un objetivo cuyo cumplimiento puede significar la diferencia entre una empresa competitiva y otra que no lo es, entre permanecer en el mercado o desaparecer, por lo tanto es importante tener en cuenta los diferentes diseños de almacenes convencionales tales como: almacén de una cuadra que contiene solo dos pasillos transversales, en la parte delantera y trasera, a su vez el almacén de dos bloques tiene las mismas características anteriores mencionadas, pero además este cuenta con un pasillo transversal que pasa por el centro del depósito.

Teniendo en cuenta que el almacenamiento convencional con líneas de dos bloques iguales supone una alternativa viable para optimizar los tiempos de picking para el comercio electrónico, en esta investigación el problema a tratar es el enrutamiento teniendo en cuenta un

HEURISTICAS PARA PROBLEMAS DE ENRRUTAMIENTO

único picker, un único depósito y clasificación ABC es de tipo NP-hard (Weidinger, Boysen y Schneider, 2019) por la variedad de productos almacenados ubicados en un número variable de estantes y en posiciones distintas dentro de ellos, provocando así, múltiples rutas posibles por las que puede optar el selector, así como múltiples alternativas de depósito. Dada la trascendencia y actualidad del tema se han venido desarrollando trabajos en diferentes facultades incluyendo la Escuela de Estudios Industriales y Empresariales de la Universidad Industrial de Santander, destacándose el proyecto titulado “Un Algoritmo de Búsqueda Tabú para El Problema de Enrutamiento de un Recolector (SPRP) en un Almacén de Comercio Electrónico con Almacenamiento Disperso y Múltiples Depósitos” del cual se han desprendido ponencias tanto nacionales como internacionales por lo tanto sugiere un campo prometedor de investigación.

Tabla 1 Estructura del proyecto

Objetivos	Capítulos y subcapítulos
Realizar una revisión bibliográfica sobre el picker routing problem warehouse y sus métodos de solución.	Capítulo 2
<ul style="list-style-type: none"> • Construir la ecuación de búsqueda • Realizar el análisis bibliométrico de los resultados de la ecuación de búsqueda 	Subcapítulo 2.1
Definir las características del problema de optimización específico a trabajar.	Capítulo 3
<ul style="list-style-type: none"> • Características de los tipos de almacenes, métodos de solución, políticas de enrutamiento. 	Subcapítulo 3.2
Formular un modelo matemático para el problema de ruteo de recolección de pedidos en almacenes.	Capítulo 4, Capítulo 5

HEURISTICAS PARA PROBLEMAS DE ENRRUTAMIENTO

Definir y desarrollar los algoritmos a usar para dar solución al modelo formulado. Capítulo 6

Evaluar los algoritmos a partir de las variables, restricciones y supuestos del problema abordado. Capítulo 7

1 Generalidades del Proyecto

1.1 Planteamiento del problema

El almacén convencional de dos bloques es bastante común según la bibliografía encontrada en las bases de datos Scopus, Web of Science y ScienceDirect, dicho almacén contiene un pasillo transversal adicional a los dos pasillos transversales, en la parte delantera y trasera que pasa por el centro del almacén. Al dividir el espacio de recolección en dos partes iguales, este pasillo transversal central aumenta la cantidad de oportunidades para que los trabajadores cambien de pasillo. Cuando se visita más de una ubicación en un solo recorrido, tener pasillos transversales adicionales en un almacén puede reducir las distancias de viaje entre ubicaciones al aumentar la cantidad de opciones de ruta. Es importante rescatar de esto, ya que según el diseño que tenga el almacén puede llegar a brindar una reducción en la distancia de recorrido para la operación de preparación de pedidos, que es la operación más costosa y la más estrechamente asociada con el tiempo de entrega de los pedidos.

La preparación de pedidos es la operación del almacén que se encarga de acondicionar los productos demandados por los clientes para que sean enviados en forma de pedidos, dicho lo anterior esta es la operación que más tiempo requiere y mayores costos genera debido a las operaciones intensivas en mano de obra del sistema, la preparación de pedidos comprende alrededor del 50% de los costos operativos totales del almacén, mientras que los viajes representan casi la mitad, con el objetivo de minimizar costos y tiempo a la hora de realizar la operación correspondiente se utilizan diferentes modelos de enrutamiento los cuales tienen

HEURISTICAS PARA PROBLEMAS DE ENRRUTAMIENTO

entre otros los siguientes métodos de solución que son heurísticas, metaheurísticas y algoritmos exactos. En el caso particular de la investigación se va a abordar los métodos heurísticos S-shape y la brecha más grande.

Las heurísticas son un método usado para hacer más sencilla la solución de problemas difíciles. Debido a la existencia de algunos problemas importantes con un gran interés práctico complejos de resolver, comienzan a surgir algoritmos capaces de ofrecer posibles soluciones que, aunque no consiguen el resultado óptimo, sí se acercan en un tiempo de cálculo razonable. En esta investigación se busca comparar los algoritmos S-shape y la Brecha más grande encontrando soluciones factibles de mínimas distancias, teniendo en cuenta que en los artículos encontrados en las bases de datos no se evidencia la comparación entre estos dos.

Para que la operación de la preparación de pedidos se pueda optimizar y tener mejores resultados se considera pertinente realizar un modelo de enrutamiento que es el encargado de brindar posibles rutas teniendo en cuenta diferentes variables. En esta investigación se considera un almacén convencional con dos bloques iguales en donde se tiene en cuenta un único depósito y variaciones con el número de recolectores, en donde se transportan los pedidos demandados, por consiguiente, se quiere obtener un modelo que evidencie la ruta que genera para el recolector la mínima distancia total recorrida.

El problema reside en la programación para un conjunto de P pedidos $p = \{1, 2, \dots, p\}$, donde los productos están ubicados en un conjunto V de posiciones $V = \{1, 2, \dots, v\}$ dentro del almacén y son almacenados en un conjunto S de SKU o unidades de almacenamiento $S = \{1, 2, \dots, s\}$. Este problema cuenta con un solo depósito donde se pueden entregar los pedidos, en cada posición de secuencia se visita una posición de almacenamiento, se tiene un máximo de pedidos activos, de modo que no se exceda la capacidad que se va a manejar para el carrito. Se considera una variación en el número de recolectores, por lo tanto, se debe tener en cuenta el tamaño de los pasillos para que cuando se tenga más de un recolector no se vaya a presentar

HEURISTICAS PARA PROBLEMAS DE ENRRUTAMIENTO

ningún tipo de bloqueo entre ellos en las diferentes partes del almacén; de igual forma, se tendrá una clasificación ABC, con el fin de variar la ruta para la recolección de pedidos dependiendo su clasificación e identificar las posibles rutas factibles por las que quiere optar el selector.

1.2 Justificación del proyecto

Actualmente sin duda alguna el internet es el medio más utilizado teniendo en cuenta la importancia que ha adquirido como consecuencia a la emergencia sanitaria que se vive actualmente, se considera una necesidad para las empresas a la hora de facilitar el proceso de compra a los clientes, debido a esto, surge una etapa mucho más sólida del eCommerce; un sistema de compra y venta de productos y servicios que utiliza el internet como medio principal de intercambio, en otras palabras se trata de un comercio que gestiona los cobros y pagos a través de medios electrónicos, donde le permite a sus clientes poder acceder a sus productos o servicios desde cualquier parte del mundo. El eCommerce en Latinoamérica ha crecido a tasas del 30% en los últimos cinco años y se espera que en los próximos tres, este crecimiento sea entre el 40 y 50%, debido a la llegada de las redes 5G (Pueyrredó M, 2019) presidente del eCommerce Institute. Cabe destacar que el eCommerce en Colombia representa el 4,4% de Latinoamérica.

Por esta razón para el eCommerce es un gran reto la recolección de pedidos dado que es la actividad prioritaria en el proceso de alistamiento de órdenes, teniendo en cuenta que el diseño del área de almacenamiento en los centros de distribución, busca facilitar la optimización de los recorridos, la rapidez de la preparación de los pedidos, la precisión de los mismos y la colocación más eficiente de las existencias todos ellos en pro de conseguir ciclos de pedido más rápidos, menores costos de operación, capacidad de almacenamiento y mejor servicio al cliente, está es la manera más productiva de disminuir costos, aumentar ganancia, optimizar uso de recursos dentro del almacén; dado que en el presente el mejoramiento continuo es una necesidad.

HEURISTICAS PARA PROBLEMAS DE ENRRUTAMIENTO

En la actualidad, la recolección de pedidos es fundamental para las empresas porque son parte de la cadena de suministro y de la logística, donde hay una oportunidad para buscar mejorar el rendimiento de las operaciones que ocurren en el almacén, con metodologías para cumplir los objetivos de las empresas, pero sin la necesidad de grandes inversiones en tecnologías. Se conoce como la actividad más costosa de una bodega o centro de distribución típico, ya que, representa el 65% de los costos operativos y el 50% de la fuerza de trabajo.

El proyecto se realiza con la necesidad de mejorar la productividad de los almacenes, este se enfoca en el proceso de alistamiento de órdenes porque es la actividad que más representa costos y esfuerzos, además de esta depende en gran medida la entrega a tiempo de pedidos al cliente final y el flujo adecuado del proceso productivo.

El problema a tratar en esta investigación es el enrutamiento teniendo un único recolector para un almacén convencional con líneas de dos bloques iguales un único depósito y clasificación ABC, con el fin de variar la ruta para la recolección de los pedidos dependiendo su clasificación e identificar las posibles rutas factibles por las que puede optar el selector, por esto se propone validar un algoritmo que contribuya a la reducción de las distancias recorridas en la recolección de los pedidos que repercute directamente en el tiempo de respuesta del almacén; cabe resaltar que el sistema convencional de almacenaje es el sistema más flexible y económico, debido a que su implementación es verdaderamente sencilla y su gran capacidad de adaptabilidad facilita los cambios que las empresas necesitan en función de sus necesidades. Esto da la posibilidad a que los almacenes se distribuyan de forma totalmente lógica. Por otra parte, es importante mencionar que en la literatura especializada hacen énfasis en la búsqueda de métodos eficientes y en la necesidad de reducir los tiempos de cálculo.

HEURISTICAS PARA PROBLEMAS DE ENRRUTAMIENTO

1.3 Objetivos

1.3.1 Objetivo general

Comparar soluciones factibles de mínimas distancias para una distribución del almacén convencional con líneas de dos bloques iguales mediante el uso de algoritmos heurísticos

1.3.2 Objetivos específicos

- Realizar una revisión bibliográfica sobre el picker routing problem warehouse y sus métodos de solución.
- Definir las características del problema de optimización específico a trabajar.
- Adaptar un modelo matemático para el problema de ruteo de recolección de pedidos en almacenes.
- Definir y desarrollar los algoritmos a usar para dar solución al modelo formulado.
- Evaluar los algoritmos a partir de las variables, restricciones y supuestos del problema abordado.
- Elaborar un artículo de carácter publicable con los resultados del trabajo de investigación.

1.4 Metodología

1.4.1 Fase 1. Realizar una revisión bibliográfica en general sobre el problema a tratar

- Revisión de literatura sobre el picking routing problem warehouse
- Revisión de literatura sobre los métodos de solución para el problema de la recolección de pedidos

1.4.2 Fase 2. Definir las características del problema

- Seleccionar e identificar el diseño de almacén más utilizado para la recolección de pedidos
- Decidir la cantidad de recolectores que realizarán la recolección de pedidos
- Determinar el número de depósitos a utilizar para la recolección de pedidos

HEURISTICAS PARA PROBLEMAS DE ENRRUTAMIENTO

1.4.3 Fase 3. Adaptar un modelo matemático

- Adaptar un modelo que represente el problema de recolección de pedidos teniendo en cuenta

Función objetivo: Minimizar las distancias totales en las rutas de los recolectores.

Variables de decisión

- Número de recolectores
- Ruta para la recolección de pedidos utilizando la clasificación ABC.
- Pedidos Heterogéneos o homogéneos.
- Capacidad de los recolectores

Restricciones

- Metros cuadrados del almacén
 - Capacidad máxima del recolector
 - Único deposito
 - Ancho de los pasillos
- Establecer hipótesis y probar su veracidad con el lenguaje de programación Python
 - Hacer comparaciones de los datos obtenidos entre los algoritmos S-Shape y la brecha más grande y así poder comprobar la coherencia del modelo adaptado, teniendo en cuenta las características definidas en el proyecto
 - Si los datos no se ajustan a lo esperado se debe ir ajustando el modelo matemático

1.4.4 Fase 4. Definir y desarrollar los algoritmos a usar

- Selección de los algoritmos entre los ya existentes identificados en la revisión de literatura relacionada con el problema de recolección de pedidos

1.4.5 Fase 5. Evaluar los algoritmos

- Buscar instancias en las que se haya abordado el problema de enrutamiento para hacer contraste con el modelo adaptado en esta investigación.
- Probar el algoritmo por medio del software de programación Python.

HEURISTICAS PARA PROBLEMAS DE ENRRUTAMIENTO

- Comparar las soluciones factibles generadas en las investigaciones encontradas con las arrojadas por el modelo matemático desarrollado en el proyecto, teniendo como criterio los tiempos de cómputo más pequeños y las mínimas distancias.

1.4.6 Fase 6. Documentación

- Elaboración del documento de proyecto de grado
- Escritura del artículo de carácter publicable con la información destacada de la investigación y los resultados obtenidos.

2 Revisión de literatura

2.1 Análisis Bibliométrico

Se puede identificar el interés por el tema en especial en los años recientes debido al crecimiento exponencial de las ventas online (Aguilar, 2020) y esto tiene una estrecha relación con el surgimiento de nuevas tecnologías aptas para la comercialización de los productos en el mercado, de esta forma, se crea la necesidad de realizar investigaciones con el fin de obtener nuevas ideas para la adaptación y actualización de los cambios sociales que se van presentando.

Teniendo en cuenta el gran interés que se presenta por investigar este tipo de temas se evidencia un amplio volumen de información que se ha generado en su mayoría en los últimos años y además es de fácil acceso por lo tanto ha resultado atractivo para la comunidad interesada en investigar los temas en cuestión, en consecuencia se crea la disciplina conocida como bibliometría la cual emplea conocimientos estadísticos y matemáticos con la finalidad de analizar el funcionamiento científico por ello cuenta con mecanismos utilizados para evaluar los aspectos de este fenómeno llamados indicadores bibliométricos como documentos publicados por año, autores, países, área temática, palabras clave y citación de autores. de lo anterior se deriva la importancia que tiene la bibliometría en el campo de la investigación por consiguiente se han creado opciones en las bases de datos como scopus, web of science y ScienceDirect que permiten extraer y analizar la información adquirida.

HEURISTICAS PARA PROBLEMAS DE ENRRUTAMIENTO

Para conseguir un buen criterio de la investigación para la recolección de pedidos en un almacén convencional con líneas de dos bloques iguales, mediante los algoritmos heurísticos S-shape y la brecha más grande, se utilizaron dos de las principales bases de datos, Scopus, Web of Science y ScienceDirect, en segundo lugar, se hizo un análisis tipo bola de nieve de citas en los artículos más relevante de la investigación.

Considerando que cada año se registra gran variedad de artículos es necesario acudir al análisis bibliométrico de tal forma que se puedan obtener los documentos relevantes para el tema de investigación en estudio.

los indicadores básicos se tendrán en cuenta para mostrar los documentos publicados por año, autores, países, área temática, para los indicadores de relación como palabras clave y citación de autores se utilizará la herramienta VOSviewer que sirve para construir y visualizar redes bibliométricas como palabras claves y citación de autores entre otras.

A continuación, se mostrará la ecuación de búsqueda y los análisis que se realizaron en las bases de datos Scopus, Web of Science y ScienceDirect.

2.1.1 Scopus

Para realizar la búsqueda en scopus se utilizó la ecuación, ((Optimization) AND (picking) AND (heuristics) AND (warehouse) AND (orders)) al día 13/09/2021. Se limitó la información teniendo en cuenta los documentos con enfoque al proyecto propuesto.

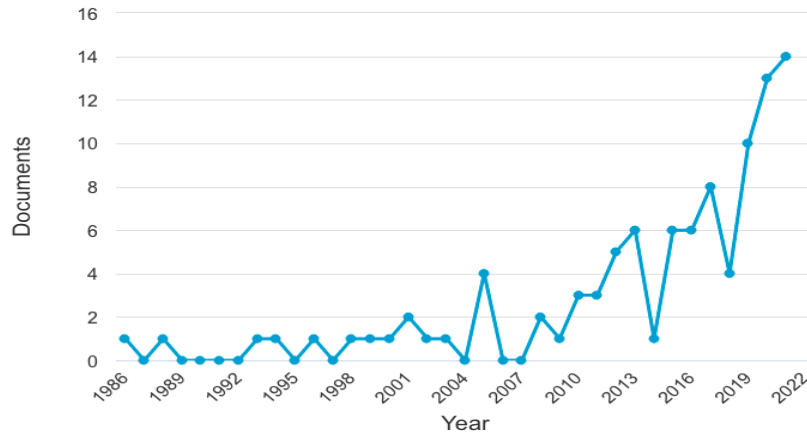
2.1.1.1 Indicadores básicos. Al analizar los resultados de la búsqueda, se obtuvieron los siguientes gráficos estadísticos.

2.1.1.1.1 Documentos por año. En la figura 1 se logra evidenciar el interés de investigar sobre este tema con el pasar de los años por ejemplo a partir del año 2010 se ve un crecimiento positivo, aun así, se coincide que en algunos años no se publicó ningún documento relacionado al tema.

HEURISTICAS PARA PROBLEMAS DE ENRRUTAMIENTO

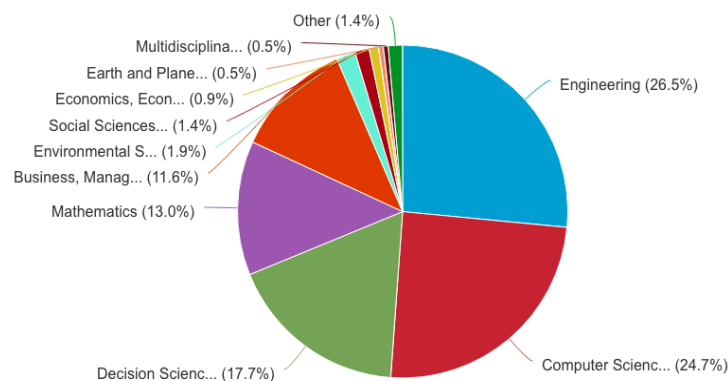
Por otro lado, llama especialmente la atención que del año 2017 al 2018 disminuyeron las investigaciones, sin embargo, a partir del año 2019 se observa que aumenta exponencialmente el número de documentos publicados según Scopus, por ende, se puede demostrar el aumento progresivo por parte de la comunidad académica al abordar este tipo de investigaciones.

Figura 1 Desarrollo del tema con el pasar de los años en Scopus



2.1.1.1.2 Área temática. En la figura 2 las áreas de investigación más relevantes en esta temática teniendo en cuenta los documentos publicados son ingeniería y ciencias de computación con un 26,5% y 24,7% respectivamente, posterior ciencias de decisión y matemáticas con un 17,7% y 13% respectivamente, cómo se puede evidenciar estas son las cuatro áreas que predominan con un 81,9% de los documentos publicados para el tema de interés en esta investigación, lo que sugiere que es un tema propio de nuestra disciplina de estudio ingeniería.

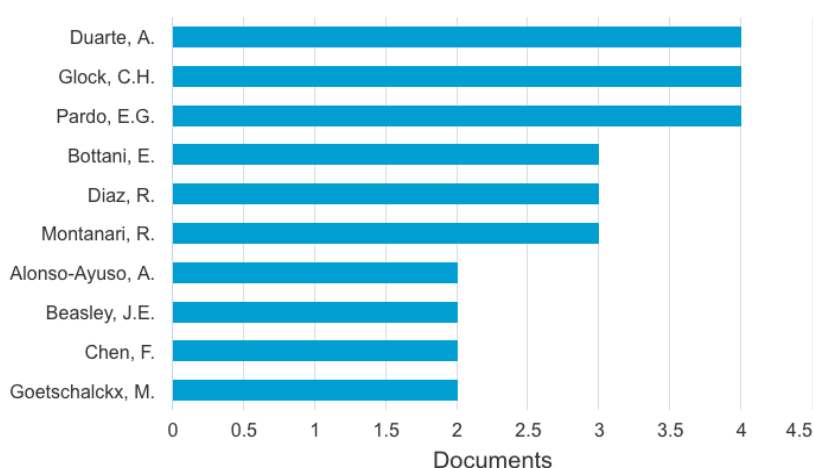
Figura 2 Área temática en Scopus



HEURISTICAS PARA PROBLEMAS DE ENRRUTAMIENTO

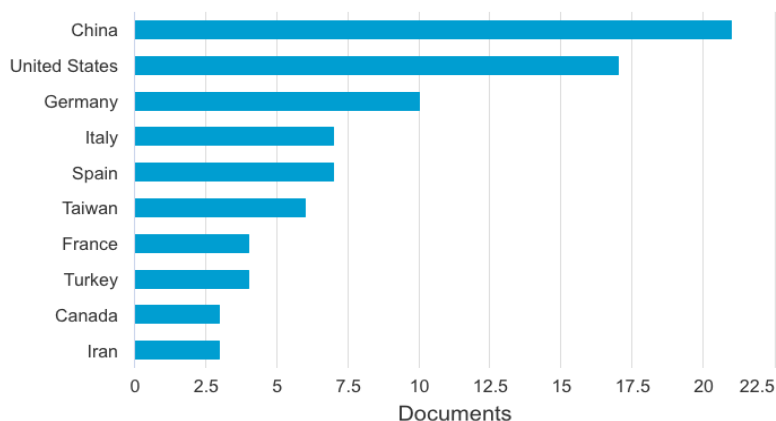
2.1.1.2.1 Autores. En la figura 3 muestra los 10 autores más relevantes para la investigación teniendo en cuenta el número de documentos publicados, cabe señalar que los autores con mayor número de publicaciones son Duarte, A., Glock, C.H., Pardo, E.G. teniendo cada uno un total de 4 documentos, luego Bottani, E., Díaz, R., Montanari, R. cada uno con 3 documentos y finalmente los cuatro últimos con un total de 2 documentos cada uno, se refleja buen material de diferentes autores que pueden servir de referencia para la investigación.

Figura 3 Autores más relevantes en Scopus



2.1.1.2.2 Países. En la Figura 4 muestra los 10 países con mayor número de documentos publicados, se puede observar que China está liderando con un total de veintiún documentos relacionados con el tema de investigación, luego está Estados Unidos con diecisiete documentos, después Alemania con diez documentos y también se puede observar que los países con menos documentos publicados relacionados con este tema de investigación son Canadá e Irán con un total de 3 documentos, con respecto a la cantidad de documentos publicados cabe mencionar que se refleja interés de los países nombrados en investigar sobre estos temas.

HEURISTICAS PARA PROBLEMAS DE ENRRUTAMIENTO

Figura 4 Países destacados en tema de investigación en Scopus

Finalmente, en la base de datos Scopus a partir de la ecuación de búsqueda utilizada genera resultados en beneficio al tema de investigación por ejemplo que en los últimos años han aumentado notoriamente los artículos publicados en relación con la temática. Además, el enfoque del plan de trabajo es congruente con el campo de la ingeniería Industrial en mayor medida según el grafico estadístico obtenido en el análisis bibliométrico anteriormente, en consecuencia, la investigación toma valor puesto que es clara la importancia de continuar indagando respecto a dicho tema en cuestión.

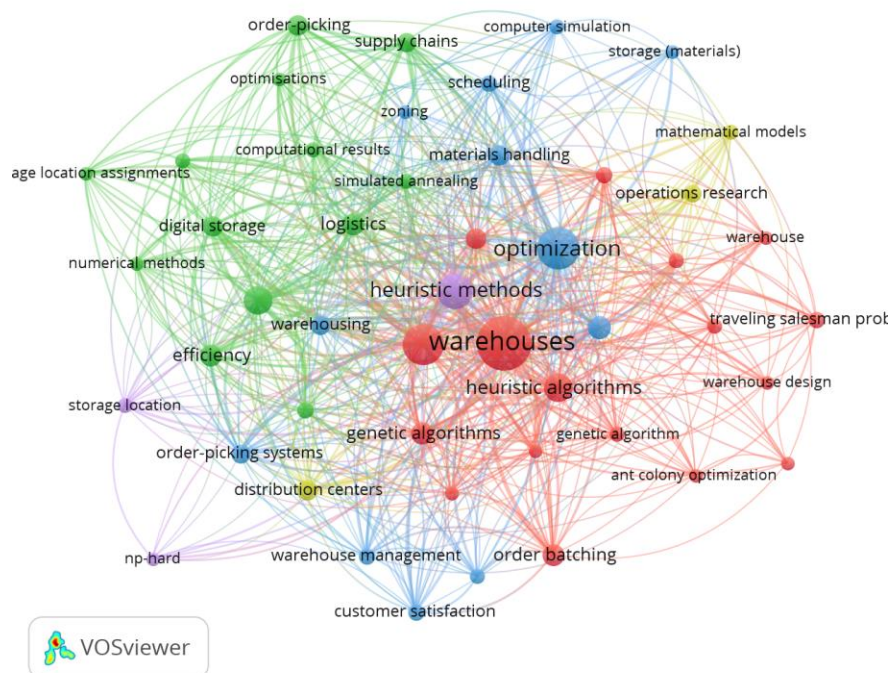
2.1.1.3 Indicadores de relación. Para la realización de los indicadores de relación se utilizó el software de VOSviewer, el cual es una herramienta de software que sirve para construir y visualizar redes bibliométricas, se pueden construir en base a citas, acoplamiento bibliográfico, co-citas o relaciones de autorías.

VOSviewer también ofrece funciones de extracción de texto que se pueden utilizar en redes de concurrencia de términos importantes extraídos de un cuerpo de literatura científica.

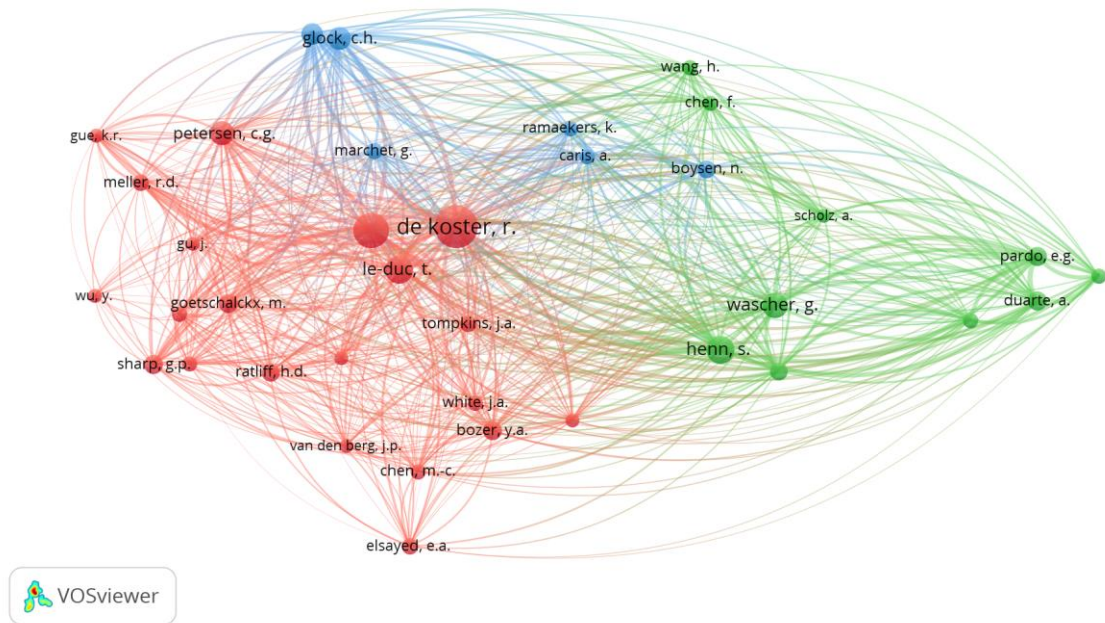
HEURISTICAS PARA PROBLEMAS DE ENRRUTAMIENTO

2.1.1.3.1 Palabras claves. En la figura 5 se observa la concurrencia de palabras claves reunidas por colores en 5 clústeres los cuales representan algoritmos de agrupamiento por temáticas relevantes. En los colores que mejor se puede evidenciar es en el rojo, azul y morado en donde la palabra más usada es warehouse, optimization, heuristics methods respectivamente. Además, es importante destacar que los círculos que tienen mayor tamaño representan más frecuencia de aparición de la palabra en los artículos.

Figura 5 Concurrencia palabras clave en Scopus



2.1.1.3.2 Citación de autores. En la figura 6 se observa 3 clústeres los cuales representan grupos constituidos por los autores más importantes, teniendo en cuenta las citaciones a lo largo del tiempo. En este caso se evidencia que los círculos con mayor tamaño es el de Koster, r; Le. Duc, t. y Petersen, cg. lo que significa que estos autores son los que más citaciones tienen al pasar los años.

Figura 6 Autores más importantes teniendo en cuenta las citaciones en Scopus

2.1.2 Web of Science

Para realizar la búsqueda en web of science se utilizó la ecuación, ((Optimization) AND (picking) AND (heuristics OR s-shape OR biggest gap) AND (warehouse) AND (orders)), obteniendo 75 documentos al día 13/09/2021. Se limitó la información teniendo en cuenta los documentos con enfoque al proyecto propuesto.

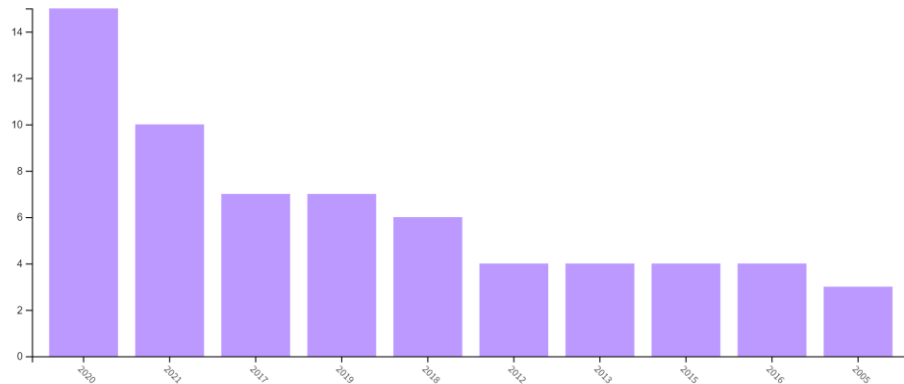
2.1.2.1 Indicadores básicos. Al analizar los resultados de la búsqueda, se obtuvieron los siguientes gráficos estadísticos.

2.1.2.1.1 Documentos por año. En la figura 7 se puede observar el aumento gradual de investigar sobre el tema de la investigación con el pasar de los años por ejemplo a partir del año 2005 hasta la actualidad se evidencia un crecimiento positivo, aun así, la excepción a lo narrado anteriormente es que en el año 2019 el número de documentos publicados es menor a los publicados en los años 2018 y 2017, para finalizar llama especialmente la atención que desde el 2006 al 2011 no aparecen registrados en la base de

HEURISTICAS PARA PROBLEMAS DE ENRRUTAMIENTO

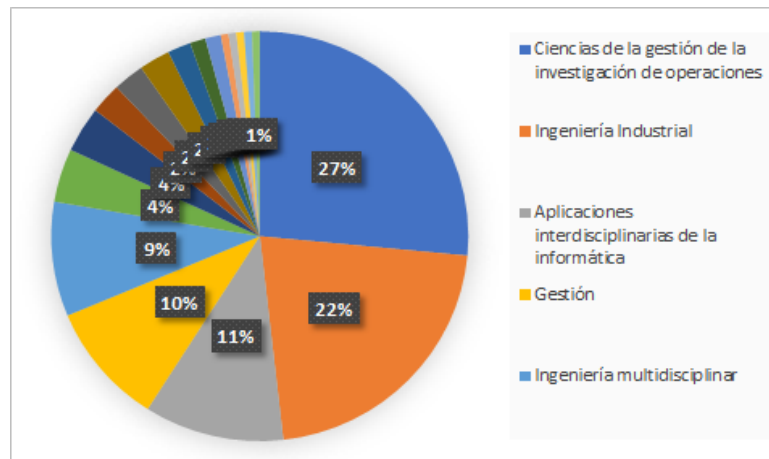
datos que se toma como fuente, al igual que en las bases de datos Scopus se logra evidenciar el crecimiento progresivo por parte de la comunidad académica por abordar este tipo de investigaciones.

Figura 7 Desarrollo del tema con el pasar de los años en Web of Science

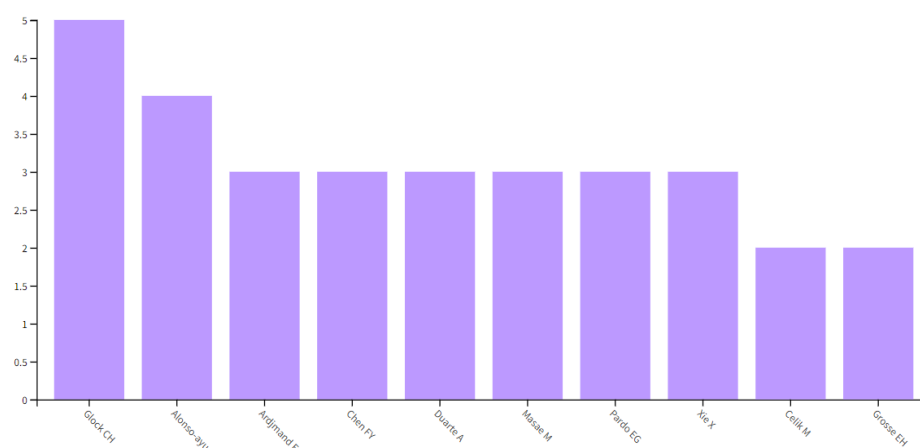


2.1.2.1.2 Área temática. En la figura 8 las áreas de investigación más relevantes en esta temática teniendo en cuenta los documentos publicados con mayor relevancia son: ciencias de la gestión de la investigación de operaciones con un 27% de publicaciones, ingeniería Industrial 22%, Aplicaciones Interdisciplinarias de la informática 11%, Gestión con 10% y, por último, pero no menos importante Ingeniería multidisciplinar con 10% de registros. En definitiva, las anteriores áreas tienen el 70% de publicaciones respecto al tema en cuestión de la investigación, al igual que en la base de datos scopus se logra percibir la fuerza de investigación en ingeniería, en este caso en particular se acerca aún más a nuestro campo de la ingeniería industrial.

HEURISTICAS PARA PROBLEMAS DE ENRRUTAMIENTO

Figura 8 Área temática en Web of Science

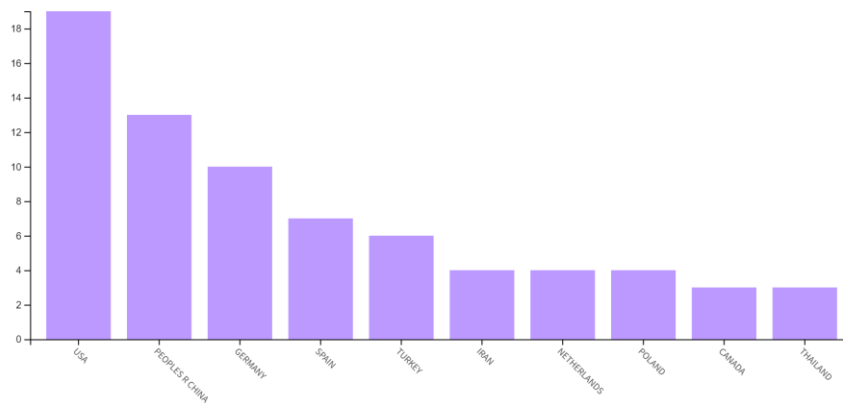
2.1.2.1.3 Autores. En la figura 9 muestra los 10 actores más relevantes para la investigación teniendo en cuenta el número de documentos publicados, cabe señalar que los autores con mayor número de publicaciones son: Glock CH con 5 publicaciones, Alonso-ayuso A con 4 registros. Luego Ardjmand E, Chen FY, Duarte A, Masae M, Pardo EG, Xie X. cada uno con 3 registros publicados y por último Celik M, Grosse EH cada uno con 2 publicaciones registradas en la base de datos Web Of Science, se refleja buen material que pueden servir de referencia para la investigación.

Figura 9 Autores más relevantes en Web of Science

HEURISTICAS PARA PROBLEMAS DE ENRRUTAMIENTO

2.1.2.1.4 Países. En la Figura 10 muestra los 10 países con mayor número de documentos publicados en la respectiva base de datos, se puede observar que Estados Unidos está liderando con un total de diecinueve documentos relacionados con el tema en cuestión de la investigación, luego está China con trece registros, después Alemania con siete publicaciones y también se puede evidenciar que los países con menos documentos publicados relacionados con este tema de investigación son Canadá y Tailandia con un total de 3 documentos asociados a la investigación, cabe resaltar que los países desarrollados tienen gran interés por investigar sobre el tema en estudio, por lo cual sugieren continuar indagando sobre la temática propuesta en la investigación.

Figura 10 Países destacados en tema de investigación en Web of Science

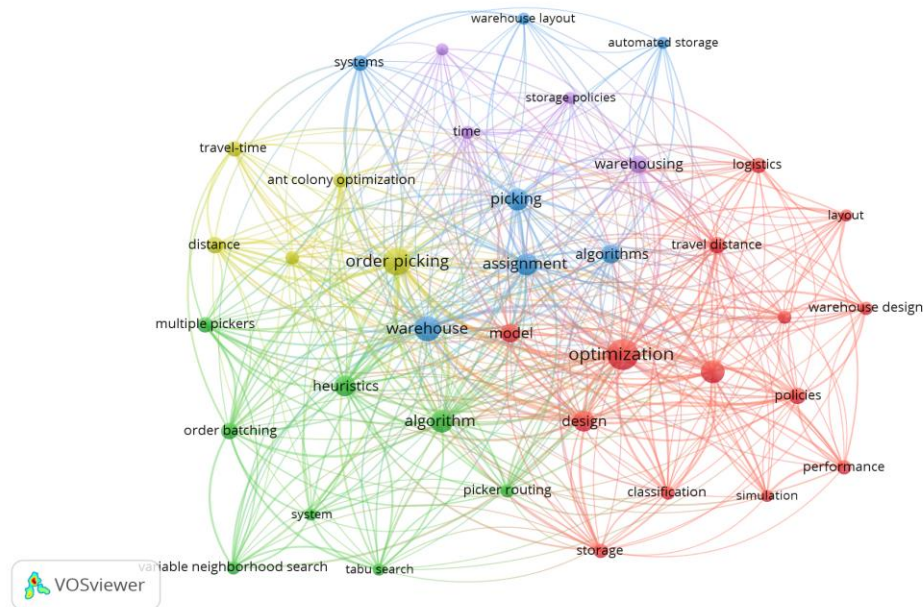


2.1.2.2 Indicadores de relación. Al igual que con Scopus, para la realización de los indicadores de relación se utilizó el software de VOSviewer.

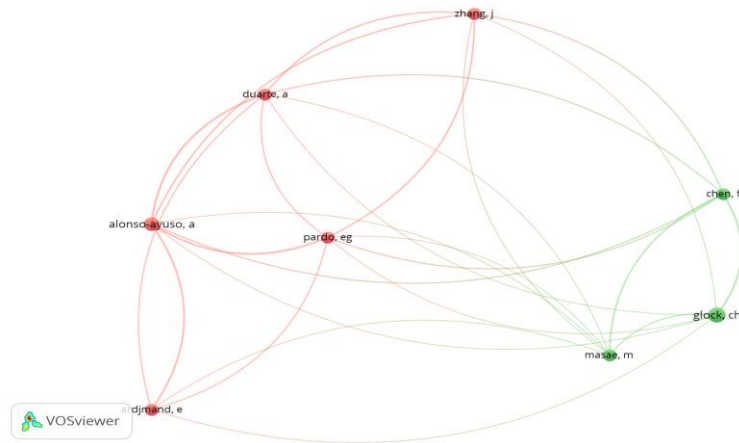
HEURISTICAS PARA PROBLEMAS DE ENRRUTAMIENTO

2.1.2.2.1 Palabras clave. Al elaborar el tesauro, con el fin de eliminar duplicidades en términos, se encontró que los resultados estaban limpios, obteniendo 38 palabras, distribuidas en 4 clústeres, los cuales representan algoritmos de agrupamiento por temáticas relevantes, se puede observar que las palabras claves sobresalientes son optimización, warehouse, picking y algorithms, por otro lado las menos destacadas son simulation, automated storage y system, teniendo en cuenta que las esferas de mayor tamaño es proporcional a la relevancia de las palabras claves

Figura 11 *Concurrencia palabras clave en Web of Science*



2.1.2.2.2 Citación de autores. En la figura 12 se observa la concurrencia de citaciones distribuidas por colores en 2 clústeres de algoritmos de agrupamiento que representa las citaciones de los autores más importantes a lo largo del tiempo. En este caso se evidencia que la esfera con mayor tamaño es la de Alonso-ayuso, lo que significa que es el autor que más relevancia tiene con el pasar del tiempo.

Figura 12 Autores más importantes teniendo en cuenta las citaciones en Web of Science

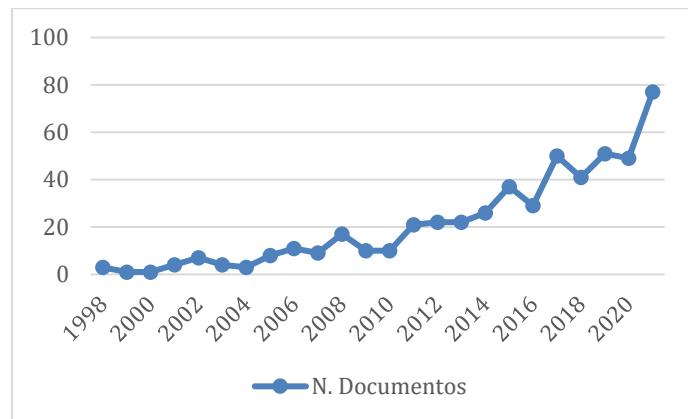
2.1.3 ScienceDirect

Para realizar la búsqueda en web of science se utilizó la ecuación, ((Optimization) AND (picking) AND (heuristics) AND (warehouse) AND (orders)), obteniendo 567 documentos al día 15/09/2021. Se limitó la información teniendo en cuenta los documentos con enfoque al proyecto propuesto.

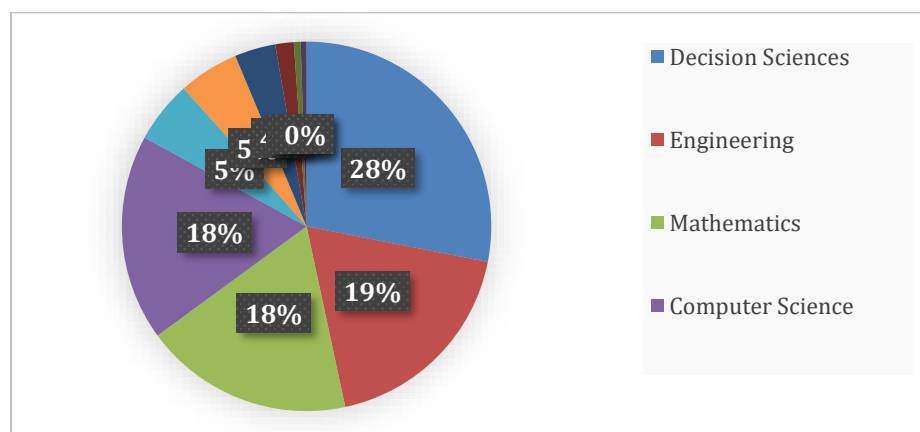
2.1.3.1 Indicadores básicos. Al analizar los resultados de la búsqueda, se obtuvieron los siguientes gráficos estadísticos.

2.1.3.1.1 Documentos por año. En la figura 13 se puede observar el crecimiento exponencial de las publicaciones y esto hace referencia al interés de los investigadores respecto al tema con el pasar de los años, por ejemplo a partir del año 20011 hasta la actualidad se evidencia un crecimiento positivo, por otra parte llama especialmente la atención que en el año 2016 disminuye el registro de los documentos publicados en la base de datos que se toma como fuente, en las tres bases de datos analizadas se logra evidenciar el crecimiento progresivo por parte de la comunidad académica por abordar este tipo de investigaciones.

HEURISTICAS PARA PROBLEMAS DE ENRRUTAMIENTO

Figura 13 Desarrollo del tema a lo largo de los años en ScienceDirect

2.1.3.1.2 Área temática. En la figura 14 las áreas de investigación más relevantes en esta temática teniendo en cuenta los documentos publicados con mayor relevancia son: ciencias de la decisión 28% de publicaciones, ingeniería 19%, matemáticas y ciencias de la computación 18%. En definitiva, las anteriores áreas tienen el 83% de publicaciones respecto al tema en cuestión de la investigación, al igual que en las anteriores bases de datos mencionadas, se logra percibir la fuerza de investigación en ingeniería, por consiguiente, sugiere a investigaciones futuras sobre el tema de estudio.

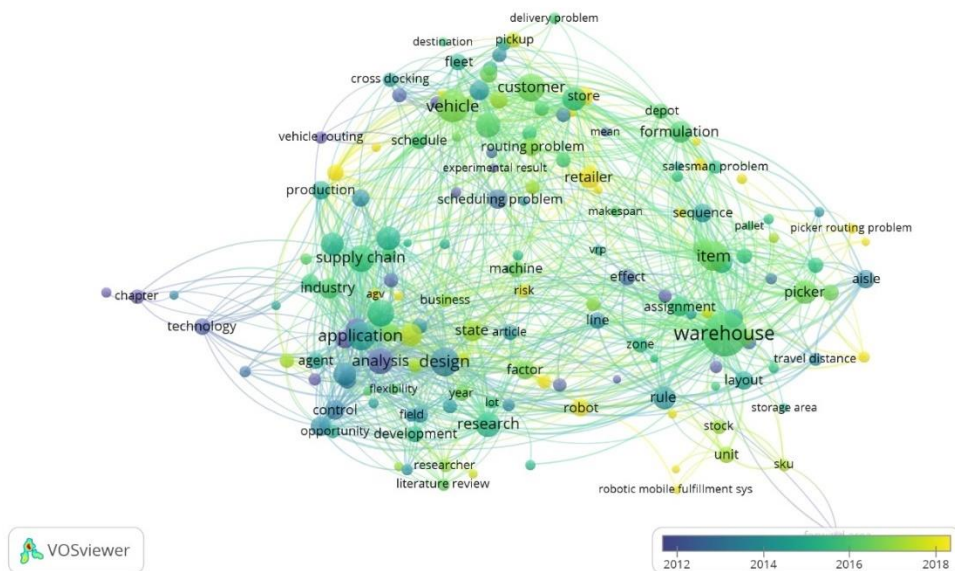
Figura 14 Área temática en ScienceDirect

HEURISTICAS PARA PROBLEMAS DE ENRRUTAMIENTO

2.1.3.2 Indicadores de relación. Al igual que con Scopus y Web of Science, para la realización de los indicadores de relación se utilizó el software de VOSviewer.

2.1.3.2.1 Palabras claves. En la figura 15 se observa la concurrencia de palabras claves reunidas en 4 clústeres por colores los cuales representan algoritmos de agrupamiento por temáticas relevantes en los respectivos años. Los colores que se pueden evidenciar son el morado, azul, verde y amarillo, en los años 2012, 2014, 2016 y 2018 respectivamente, en donde la palabra que sobresale en cada uno de los años es analysis, desing, warehouse y retailer. Además, es importante destacar que los círculos que tienen mayor tamaño representan más frecuencia de aparición de la palabra en los artículos.

Figura 15 Concurrencia pablaras claves en ScienceDirect



2.2 Análisis preliminar de la literatura

Debido a la pandemia, se tuvieron que tomar medidas como el confinamiento lo cual generó incertidumbre para los empresarios, en vista de lo anterior se vieron en la obligación de implementar y fortalecer las ventas online ya que era la única manera de llegar a los clientes, de la misma forma los clientes se vieron en la necesidad de crear la costumbre de realizar las compras por internet, esta situación que tuvo que enfrentar todo el mundo hizo que el comercio

HEURISTICAS PARA PROBLEMAS DE ENRRUTAMIENTO

electrónico evolucionara el resultado de la implementación de las ventas online en pequeñas y medianas empresas que no ofrecían ese servicio.

Pero, sin embargo, la preparación de pedidos es una operación crítica para la mayoría de las empresas en línea, por lo cual diversos estudios han revelado que la recolección de órdenes es la actividad más costosa de una bodega o centro de distribución típico, pues representan el 65% de los costos operativos y el 50% de la fuerza de trabajo (Jiménez, 2013, generalidades en los centros de distribución, párr.2), sin embargo una política de preparación de pedidos diseñada adecuadamente puede minimizar tanto los costos de picking como los de logística en general.

Para el proceso mencionado anteriormente, las empresas toman decisiones sobre el diseño y el control de los sistemas de preparación de pedidos a nivel táctico u operativo. Estas decisiones comunes son en los centros de distribución y en los almacenes que pueden ser clasificados según su relación con el flujo de producción, ubicación, el material a almacenar, localización, logística y el enrutamiento de los selector(es), los costos obligan a utilizar el espacio de almacenamiento de manera más eficiente, de modo que se han desarrollado diversas maneras de asignar los productos a sus lugares de almacenamiento en los centros de distribución, tales como: almacenamiento dedicado (De Koster y Neuteboom, 2001), almacenamiento aleatorio (Choe y Sharp, 1991; Vidal, 2009), almacenamiento abierto (De Koster, Duc y Roodbergen, 2007) y (Hausman, Schwarz y Graves 1976), almacenamiento de acuerdo con su rotación (Vidal, 2009), almacenamiento basado en clases (Vidal, 2009), familias de agrupación (Koster et al., 2007) y, operaciones de recolección de ordenes (Picking) (Ratliff y Rosenthal, 1982) (Jiménez, 2013, generalidades en los centros de distribución, párr.2), para resolver los diferentes problemas encontrados por los autores, en la literatura se han propuesto tres tipos generales de algoritmos:

HEURISTICAS PARA PROBLEMAS DE ENRRUTAMIENTO

Los algoritmos exactos siempre encuentran una óptima (es decir, la ruta más corta) a un problema de ruta del recolector de pedidos. Los ejemplos incluyen los algoritmos de RR, De Koster y Van der Poort (1998) y Roodbergen y De Koster (2001a , b) (Masae et al., 2020, característica del algoritmo, párr.2).

Las heurísticas son algoritmos dependientes del problema construidos de acuerdo con sus especificaciones, y el resultado en la mayoría de los casos no es óptimo (Sörensen, 2015). Los ejemplos incluyen el recorrido conocido como forma de S, el punto medio y la heurística de brecha más grande (Hall, 1993) (Masae et al., 2020, característica del algoritmo, párr.3).

Las metaheurísticas son algoritmos independientes del problema de alto nivel que proporcionan un conjunto de pautas o estrategias para encontrar una solución aproximada al problema (Sörensen, 2015). Los ejemplos incluyen algoritmos genéticos (GA ; Tsai et al., 2008), optimización de colonias de hormigas (ACO ; Chen et al., 2013), optimización de enjambres de partículas (PSO ; Lin et al., 2016) o búsqueda tabú (TS ; Cortés et al., 2017) (Masae et al., 2020, característica del algoritmo, párr.4).

En el análisis previo de literatura que se tomaron de las bases de datos se logra plasmar una línea de tiempo de los artículos publicados que tratan el problema de recolección de pedidos desde diferentes perspectivas y métodos de solución que tienen como principal objetivo en común de minimizar los costos de picking que resulta ser costosa para las compañías dedicadas a actividades económicas que necesitan almacenamiento de productos, de este modo a lo largo de los años se tuvieron en consideración publicaciones relevantes a la investigación como para el caso de Koster, R., y Der Poor, E., (1998), donde estudian el problema de encontrar rutas eficientes de selección de pedidos tanto para almacenes, donde los recolectores tienen un depósito central para cosecha y depositar carros y listas de selección, y modernos almacenes, donde los camiones de preparación de pedidos pueden recoger y depositar paletas en la cabecera de cada pasillo sin regresar al depósito, ampliaron el conocido

HEURISTICAS PARA PROBLEMAS DE ENRRUTAMIENTO

algoritmo polinomial de Ratliff y Rosenthal que consideró almacenes con un depósito central. En la práctica, el problema se resuelve principalmente utilizando la denominada forma de S heurístico en el que los selectores de pedidos se mueven en una curva en forma de S a lo largo de las ubicaciones de selección, Los autores concluyeron que el nuevo algoritmo ofrece una reducción del tiempo de viaje por ruta de entre un 7 y un 34% obteniendo, así como resultado que la reducción del tiempo de viaje depende en gran medida de la disposición y el funcionamiento del depósito, siguiendo en este razonamiento se analiza el texto de T. Le-Duc., y Koster, R., (2005), en concordancia proponen un modelo probabilístico que les permite estimar la distancia de viaje promedio de un recorrido de recolección. Usando la distancia de viaje promedio como función objetivo, presentan una formulación matemática para el problema de optimización de la zona de almacenamiento. Sin embargo, el enfoque exacto solo puede manejar instancias de almacén de tamaño pequeño. Para sortear este obstáculo, proponen una heurística, al realizar una serie ejemplos numéricos muestran que el método heurístico funciona muy bien en todos los casos, Después del desarrollo del procedimiento heurístico los autores lograron deducir que cuando se parte de una solución factible y simplemente se intercambian clases de almacenamiento entre pasillos, las otras condiciones se satisfacen automáticamente. En definitiva, las soluciones factibles generan mejores resultados que las soluciones óptimas obtenidas de los métodos exactos, este documento es un ejemplo claro porqué tomaron en consideración un diseño simple y para este diseño sólo fue relevante la heurística de retorno.

A continuación, en el artículo publicado por Onut et al., (2008), utilizaron un método de solución metaheurístico el cual se enfoca en el diseño del almacén y por consiguiente el tipo de distribución que es un factor importante a la hora de minimizar las rutas y costos de la operación en general que es el tema que nos ocupa en particular el estudio tiene como objetivo modelar el problema de diseñar un almacén de varios niveles considerando los costos de

HEURISTICAS PARA PROBLEMAS DE ENRRUTAMIENTO

manipulación en tres dimensiones. Una de las contribuciones del modelo propuesto es mejorar el diseño del almacén bidimensional (Bassan et al., 1980) al diseño de almacén de niveles múltiples considerando una estrategia de almacenamiento basada en clases que incluye tres tipos de productos paletizados, a saber, A, B y C. los autores exponen que la principal dificultad para resolver este tipo de problemas de diseño es luchar con la no linealidad en las variables y las limitaciones para encontrar una solución óptima, a su vez, dado que se demostró que el modelo propuesto era NP-hard, proponen cómo solución utilizar un algoritmo novedoso PSO que genera soluciones casi óptimas en poco tiempo, continuando con el tema de estudio se llevó a cabo el análisis del artículo publicado en la base de datos Web of Science por Theys et al., (2010), estos autores tratan el problema de secuenciación y distribución de los preparadores de pedidos en los sistemas de almacén de pasillos múltiples paralelos convencionales, para este problema de vendedor ambulante (TSP), los algoritmos exactos solo existen para almacenes con un máximo de tres pasillos transversales, mientras que para otros tipos de almacenes la literatura proporciona una selección de heurísticas de construcción dedicadas. Los autores informan ahorros promedio en la distancia de ruta de hasta un 47% cuando se utiliza la heurística TSP LKH (Lin – Kernighan – Helsgaun), el rendimiento de la LKH adaptada es en promedio más rápida que el algoritmo exacto y a su vez superior a las heurísticas disponibles en la literatura, cabe destacar que según lo antes interpretado se corrobora que las heurísticas generan mejores soluciones que los métodos exactos. Siguiendo el hilo conductor del tema de investigación propuesto, en el año 2010 se vuelve a aplicar el método metaheurístico antes mencionado en este caso para evitar la congestión entre recolectores que tiene como fin minimizar las rutas y por consiguiente los costos implementados en el proceso económico, en el texto de Chen et al., (2013), buscan desarrollar un método de enrutamiento para controlar la congestión del recolector que desafía la suposición tradicional con respecto al sistema de preparación de pedidos de pasillo estrecho. Proponen un nuevo algoritmo de enrutamiento

HEURISTICAS PARA PROBLEMAS DE ENRRUTAMIENTO

basado en Ant Colony Optimization (ACO) es un algoritmo metaheurístico que simula el comportamiento de las colonias de hormigas en la naturaleza, se utiliza comúnmente para resolver el problema del vendedor ambulante (TSP), para dos recolectores de pedidos (A-TOP) con consideración de congestión. También se analizan los impactos del diseño del almacén, el tamaño del pedido y la relación pick-time-walk-time en A-TOP y el rendimiento del sistema. A-TOP puede adaptarse a diferentes configuraciones de almacén. Los autores afirman que el tiempo medio de preparación y el tiempo medio de espera de dos preparadores de pedidos se seleccionan como criterios de rendimiento.

Reanudando la importancia del tipo de distribución del almacén para cumplir el objetivo común que corresponde a minimizar los costos de operación en las empresas, se encuentra un artículo de Jiaxi et al., (2016) que definen un nuevo problema de asignación de almacenamiento dinámico (DSAP) y desarrolla un mecanismo integrado con fines de optimización, basado en la clasificación ABC y la afinidad mutua de productos. Se desarrolla una heurística basada en afinidad de productos (PABH), una técnica basada en la minería de datos, para calcular las relaciones por pares entre productos. Se muestra que el DSAP analítico y multiparamétrico es un problema de asignación cuadrática (QAP) y, por lo tanto, no determinista en tiempo polinómico (NP-Hard). Debido a la problemática encontrada por los autores desarrollaron un algoritmo genético codicioso (GA) para manejar dicha complejidad computacional del DSAP. En tal sentido es importante tomar en cuenta todas las variables que genera la actividad de recolección de pedidos, por lo tanto, en el año 2018 en el artículo tienen en consideración los múltiples recolectores que serían parte fundamental puesto que aportan la mano de obra a la actividad en cuestión, en el artículo de Ardjmand et al., (2018), se desarrolló la asignación de pedidos, la preparación de pedidos por lotes y el enrutamiento de los recolectores en un gran almacén de preparación de oleadas de una importante empresa de logística de terceros de EE. UU. Desarrollaron un modelo matemático para el problema y se

HEURISTICAS PARA PROBLEMAS DE ENRRUTAMIENTO

propusieron dos algoritmos (LD-PSO y PSA-ACO) para resolver el modelo. Los métodos propuestos se probaron con los datos obtenidos del almacén y se compararon con el algoritmo MMI que se utiliza en el almacén y un VND de última generación. Se observó que el LD-PSO sugerido es mucho más efectivo en términos de tiempo de cálculo que el método exacto. Además, se encontró que PSA-ACO supera a todos los demás algoritmos en términos de calidad de la solución y es bastante confiable para problemas a gran escala observados en el almacén, se puede deducir que es reiterativo que las heurísticas proporcionan mejores soluciones que los métodos exactos.

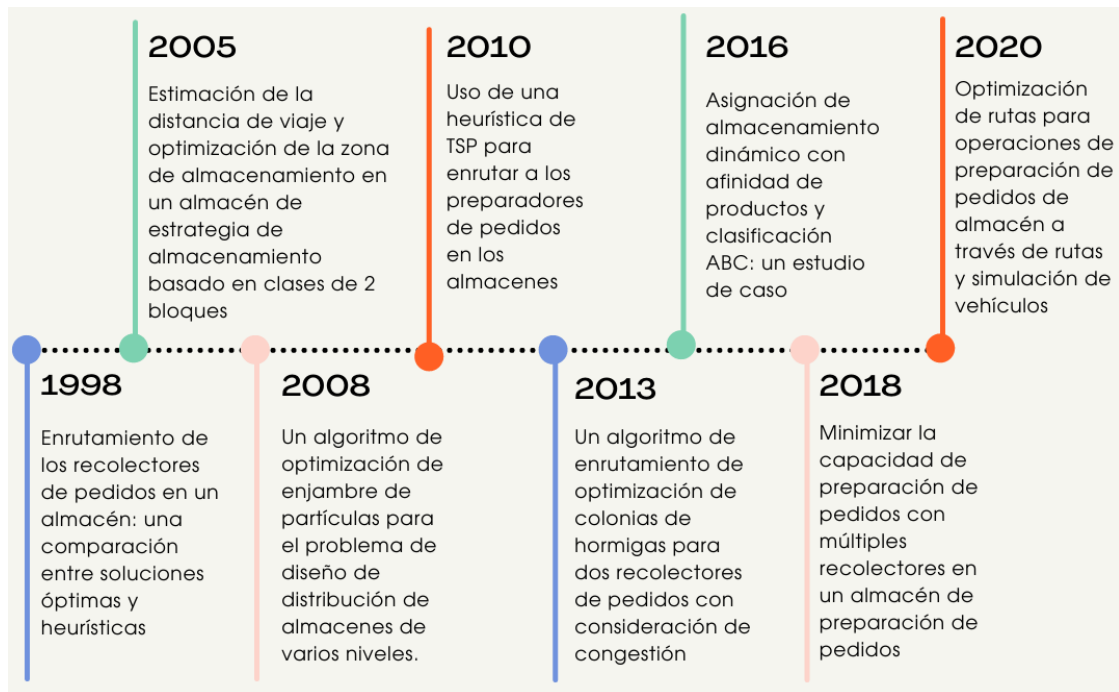
El documento publicado en el año 2020 reitera la importancia de tratar esta problemática vigente ya que al pasar los años los investigadores se ven más interesados con el tema, dejando a su vez paso a futuras investigaciones, lo antes mencionado por medio de artículos como el de Shetty, et al., (2020), proponen un enfoque simple pero efectivo que se basa en la formulación de un problema de generación de rutas de vehículos (VRP) siendo el objetivo de este estudio proponer una política de enrutamiento que minimice el tiempo / distancia total de viaje en el proceso de preparación de pedidos, lo que a su vez minimiza los costos operativos generales. Los autores lo lograron mediante optimización y simulación, seguido de análisis estadísticos y los resultados obtenidos muestran que el enfoque propuesto basado en VRP supera a los métodos de brecha más grande en forma de S, retorno, punto medio.

Por todo lo dicho, se puede concluir que en la mayoría de los artículos mencionados su principal objetivo es encontrar rutas eficientes de selección de pedidos para almacenes, y a su vez para la solución de dicha problemática, hay tres documentos que concuerdan en comparar los métodos exactos con heurísticas afirmando que las soluciones factibles son mejores que las soluciones óptimas y dichos artículos fueron publicados en los años 2005, 2010 y 2018, después de este análisis a la literatura donde se reafirma las buenas soluciones que presentan las

HEURISTICAS PARA PROBLEMAS DE ENRRUTAMIENTO

heurísticas, por esta razón se valida la investigación propuesta que está encaminada a comparar la forma de s y la brecha más grande.

Figura 16 Línea del tiempo de documentos relacionados al enrutamiento para la recolección de pedidos en un almacén convencional con líneas de dos bloques iguales.



3 Marco Teórico

El actual trabajo de investigación analiza el modelo de enrutamiento para la recolección de pedidos en un almacén convencional con líneas de dos bloques iguales y se busca solucionarlo mediante los algoritmos heurísticos S-shape y la brecha más grande. Por consiguiente, es importante definir algunos conceptos, para precisar la interpretación que tendrán estos términos en el proyecto.

3.1 Logística

Según el Instituto Colombiano de Automatización y Codificación Comercial (GS1 Colombia) es el proceso de planear, controlar y administrar la cadena de abastecimiento y distribución, desde el proveedor hasta el cliente y con un enfoque en la red de valor y colaboración entre los actores de la red logística interna y externa.

HEURISTICAS PARA PROBLEMAS DE ENRRUTAMIENTO

3.1.1 La logística de almacenamiento

Se encarga de gestionar y planificar todo lo relativo a los elementos, mercancías o materias primas que una empresa recibe para realizar su actividad (Lopez, 2021).

Es decir, no solo tratan de almacenar, también tratan de que el almacenaje sea eficiente. por ejemplo, no tendría mucho sentido colocar y guardar todo perfectamente, pero desordenado, puesto que todo sería un completo desorden. Cabe aclarar que este tipo de logística se ejecutará en el caso de que la empresa tenga almacén. Ya que existen empresas que todo lo que reciben pasa directamente a producción.

las funciones que debe cumplir la logística de almacenamiento son:

- ❖ Actualizar los inventarios
- ❖ Registro del lugar en el que se encuentran los productos almacenados
- ❖ Planificar la zona de almacenamiento según el tipo de producto
- ❖ Facilitar la incorporación de los aprovisionamientos al proceso de producción
- ❖ Indicar cómo serán transportados cada uno de los aprovisionamientos

3.2 Almacenamiento

El almacenamiento es una parte esencial de la gestión de la cadena de suministro debido a sus importantes funciones, como almacenar piezas y materiales, así como productos terminados, y proporcionar un medio unificado y simplificado para consolidar materiales de proveedores de todo el mundo. Para mantenerse al día con la creciente competencia en la industria y superar a los competidores, los almacenes deben realizar todas sus operaciones, por ejemplo, recepción, almacenamiento, preparación de pedidos (cross-docking) y envío de la manera más eficiente para garantizar el buen funcionamiento de la cadena de suministro minimizando el costo (Shetty et al., 2020, introducción, párr.1).

3.2.1 Análisis ABC

Diseñar una política de control de inventarios enfocada a cada producto resulta económicamente inviable, debido a esto se recomienda la implementación de métodos para la clasificación de productos como por ejemplo el análisis ABC. El análisis ABC permite la clasificación de los productos en tres grandes grupos: A, los más importantes; B, los importantes; y C, los menos importantes (Garza, 2018, p. 13).

De acuerdo con la literatura existen diversas estrategias para la clasificación de los productos, de tal forma que se beneficie toda la cadena de suministro. Por ejemplo, los productos se pueden clasificar basado en precio, valor, peso, demanda, rotación y disponibilidad (Garza, 2018, p. 14).

3.2.2 Métodos de solución

Existen distintos métodos para la solución de problemas del tipo análisis ABC, entre los que destacan los siguientes:

3.2.3.1 Proceso de Jerarquía Analítica. Este método es el de uso más común, se caracteriza porque puntúa cada tipo de ítem de inventario en cada criterio y luego combina las diferentes puntuaciones usando un esquema de ponderación subjetivo (Garza, 2018, p. 14).

3.2.3.2 Optimización Lineal. Este método resuelve un problema de programación lineal para cada elemento del inventario para determinar pesos que maximizan la puntuación ponderada para ese artículo sujeto a restricciones que la suma ponderada para cada elemento que utiliza este mismo conjunto de pesos es menor o igual a uno. Por lo tanto, una crítica inmediata de este modelo es que, con más de un puñado de elementos, el proceso se volverá tedioso y requiere mucho tiempo para su solución (Garza, 2018, p. 14).

HEURISTICAS PARA PROBLEMAS DE ENRRUTAMIENTO

3.2.3.3 Redes Neuronales. Para este método, se sigue un proceso similar al de los algoritmos genéticos, pero utilizan redes neuronales artificiales para resolver un problema de clasificación de inventario con cuatro criterios: precio unitario, costo de pedido, rango de demanda y tiempo de entrega. Las entradas a la red son valores de estos criterios para diferentes ítems de inventario. La salida de la red es una categorización de un conjunto de valores de criterios como A, o B, o C (Garza, 2018, p. 14).

3.2.3.4 Agrupación. Considera el análisis de ABC como un problema de agrupación en el que los elementos de inventario que tienen que ser categorizados se dividen en tres clústeres difusos, minimizando una función de agrupación apropiada. (Garza, 2018, p.14)

3.3 Preparación de pedidos

La preparación de pedidos se define comúnmente como el proceso de recuperar artículos de sus ubicaciones de almacenamiento en respuesta a los pedidos de los clientes, se considera uno de los que más tiempo y trabajo requieren. Algunos autores estimaron que representa hasta el 55% de los costos operativos totales del almacén (Tompkins et al., 2010), lo que ilustra que la preparación de pedidos es una palanca importante para aumentar la eficiencia del almacenamiento. En la práctica, la mayoría de los almacenes de preparación de pedidos se operan de acuerdo con el principio de picker-to-parts y con una alta proporción de trabajo manual (De Koster et al., 2007; Van Gils et al., 2018), principalmente porque los humanos pueden reaccionar de manera más flexible a los cambios que ocurren en el proceso de preparación de pedidos que las máquinas debido a sus habilidades cognitivas y motoras (Grosse et al., 2015, 2017)

3.4 Optimización de rutas de “Picking”

“El proceso de optimización de “picking” requiere del modelaje de un sistema geométrico adecuado, el cual corresponda a una ruta de viaje dentro de un almacén” (Garza, 2018, p. 14).

HEURISTICAS PARA PROBLEMAS DE ENRRUTAMIENTO

El propósito principal de la optimización de los procesos de “picking” es la mejora en el nivel de servicio hacia los clientes, así como la reducción de costos mediante la eliminación de actividades que no generan valor en el proceso. Las ventajas de la optimización del proceso de “picking” guardan una alta relación con el grado de complejidad de la ruta trazada, mientras que su impacto se ve mermado o beneficiado por el grado de involucramiento del “picker” sobre el proceso (Garza, 2018, p. 13).

Algunos de los objetivos principales que se buscan a la hora de implementar el sistema de recolección de pedidos son los siguientes, minimizar los tiempos de traslado promedio, maximizar el uso de los espacios en el almacén y minimizar el tiempo de procesamiento de una orden (Garza, 2018).

De igual manera las empresas toman algunas decisiones sobre el diseño que tendrá la bodega y el control de sistemas de recolección de pedidos a nivel táctico y operativo, las cuales son, diseño de distribución y sus respectivas dimensiones, asignación de productos en los estantes, asignar órdenes para recoger los pedidos y rutina para surtir la bodega (Garza, 2018).

3.5 Diseño del almacén

El diseño del almacén tiene en cuenta el tipo general de almacén considerado, el número y la ubicación de los depósitos y varias características de los pasillos. En cuanto al tipo de almacén, la literatura discutió tres variantes principales de almacén (Makusee et al., 2010, características del problema, párr.2)

3.5.1 Almacenes convencionales

Los almacenes convencionales tienen una forma rectangular con pasillos de picking paralelos que son perpendiculares a un cierto número de pasillos transversales rectos. Los almacenes convencionales con dos pasillos transversales en los extremos delantero y trasero a menudo se denominan almacenes de un solo bloque, mientras que los almacenes con más de dos pasillos transversales a menudo se denominan almacenes de bloques múltiples, donde cada

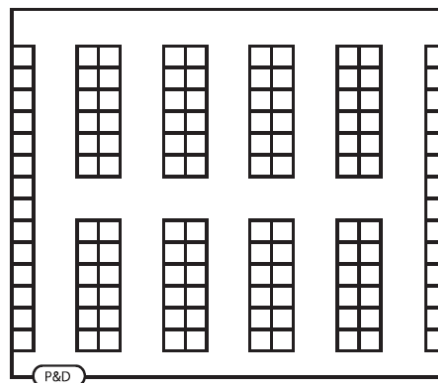
HEURISTICAS PARA PROBLEMAS DE ENRRUTAMIENTO

bloque del almacén consta de varios pasillos secundarios (Makusee et al., 2010, características del problema, párr.3).

3.5.2 Almacén de dos bloques iguales

Contiene un pasillo transversal adicional a los dos pasillos transversales, en la parte delantera y trasera que pasa por el centro del almacén. Al dividir el espacio de recolección en dos partes iguales, este pasillo transversal central aumenta la cantidad de oportunidades para que los trabajadores cambien de pasillo. Cuando se visita más de una ubicación en un solo recorrido, tener pasillos transversales adicionales en un almacén puede reducir las distancias de viaje entre ubicaciones al aumentar la cantidad de opciones de ruta (Ozturkoglu & Hoser, 2018, introducción, párr.3).

Figura 17 *Diseño almacén de dos bloques iguales*



Tomado de “a discrete cross aisle design model for order-picking warehouses” (p.413), por O. Ozturkoglu y D. Hoser, 2019, Elsevier.

3.5.3 Almacenes no convencionales

Los almacenes no convencionales no disponen todos los pasillos de picking o pasillos transversales en paralelo entre sí, sino que seleccionan un diseño diferente para facilitar el acceso a determinadas regiones del almacén o para mejorar la utilización del espacio. Los ejemplos incluyen la espina de pescado y la V voladora (Çelik y Süral, 2014) y los diseños en

HEURISTICAS PARA PROBLEMAS DE ENRRUTAMIENTO

forma de U (Glock y Grosse, 2012) (Makusee et al., 2010, características del problema, párr.4).

3.5.4 Almacenes generales

Los modelos de almacenes generales no hacen suposiciones sobre los pasillos del almacén, sino que utilizan matrices de distancias generales. Como resultado, no es posible utilizar matrices de distancia especialmente estructuradas como en el trabajo de RR, por ejemplo, lo que dificulta la solución eficiente del problema de enrutamiento de los preparadores de pedidos en estos almacenes. El problema resultante es idéntico al TSP o CVRP clásico (Makusee et al., 2010, características del problema, párr.5).

El diseño del almacén define el número y la ubicación de los depósitos, así como las características de los pasillos. En la literatura se analizaron tanto los almacenes de depósito único como los de varios depósitos con pasillos anchos y estrechos. En los almacenes con pasillos estrechos, por ejemplo, el preparador de pedidos puede recoger artículos de ambos lados del pasillo sin tener que cruzarlo, mientras que, en los almacenes de pasillos anchos, el picking de ambos lados del pasillo hace que sea necesario cruzar el pasillo, lo que lleva a un adicional. distancia de viaje. Si el almacén utiliza estanterías de almacenamiento de bajo nivel, los artículos se pueden recoger directamente de las estanterías sin necesidad de desplazamientos verticales (Scholz y Wäscher, 2017), mientras que en el caso de racks de almacenamiento de alto nivel, también pueden ser necesarios movimientos verticales. El primer almacén generalmente se conoce como un sistema de preparación de pedidos de bajo nivel, mientras que el segundo se conoce como un sistema de alto nivel (Makusee et al., 2010, características del problema, párr.6).

3.6 Optimización combinatoria

Los problemas de optimización combinatoria presentan complejidad a la hora de resolverse debido a la cantidad de soluciones factibles que se tienen (García, 2016, optimización combinatoria, párr.2).

Un problema de optimización combinatoria es un problema de optimización en el cual el espacio de soluciones posibles es discreto. La optimización combinatoria es una rama de la optimización de las matemáticas aplicadas fuertemente relacionada con la investigación operativa, la teoría algorítmica y la teoría de la complejidad computacional. Los algoritmos de optimización combinatoria se relacionan con problemas Np Hard. Estos son los modelos más complejos que se encuentran en la investigación de operaciones, debido en gran parte a su tamaño; además no existe un método que garantice soluciones óptimas, es decir, solo se logran soluciones factibles (García, 2016, optimización combinatoria, párr.2).

3.6.1 *Problema del agente viajero*

Quizá el problema clásico de optimización combinatoria más famoso es conocido como el problema del agente viajero. Recibió este nombre pintoresco porque puede describirse en términos de un agente de ventas que debe visitar cierta cantidad de ciudades en un solo viaje. Si comienza desde su ciudad de residencia, el agente determinará cuál ruta debe seguir para visitar cada ciudad exactamente una vez antes de regresar a su casa de manera que se minimice la longitud total del viaje (Lieberman, 2017), el Steiner TSP es un problema de optimización combinatoria NP-hard y los algoritmos exactos solo están disponibles para almacenes que constan de dos y tres bloques. En un artículo fundamental de Ratliff y Rosenthal (1983) proponen un algoritmo exacto para la configuración de bloque único, basado en subgrafos de recorrido parcial y siete clases de equivalencia. Roodbergen y De Koster (2001) extienden este algoritmo exacto a la configuración del almacén con tres pasillos transversales. Para los almacenes con tres o más bloques, no se dispone de un algoritmo exacto

HEURISTICAS PARA PROBLEMAS DE ENRRUTAMIENTO

hasta la fecha, por lo que la atención científica se ha dedicado en gran medida a encontrar heurísticas eficientes para determinar la ruta más adecuada para el preparador de pedidos. El problema del agente viajero se complica con rapidez a medida que aumenta el número de ciudades (Theys et al., 2010, descripción el problema y revisión de literatura, párr.3). En el caso de un problema con n ciudades y una ligadura entre cada par de ellas, el número de rutas factibles que debe considerarse es $(n - 1)!/2$. El denominador de 2 surge porque cada ruta tiene una vía inversa equivalente con la misma distancia. Así, mientras un problema del agente viajero con 10 ciudades tiene menos de 200.000 soluciones factibles que deben ser consideradas, y un problema con 15 ciudades tiene alrededor de 10^{10} soluciones factibles; se puede concluir que al aumentar el número de ciudades mayor serán las soluciones factibles que se generan, teniendo en cuenta lo anterior se desencadena la enorme dificultad para resolver grandes problemas de este tipo por lo tanto, los métodos heurísticos guiados por metaheurísticas son una forma popular de enfrentar tales problemas (Lieberman, 2017).

3.6.2 Teoría de la complejidad computacional

La optimización computacional incluye las disciplinas de investigación operativa para modelar el sistema, matemáticas para formular el modelo, ciencias de la computación para el diseño y análisis de algoritmos e ingeniería del software para implementar el modelo. Matemáticamente, los problemas pueden caracterizarse atendiendo a la dificultad que entraña su resolución por un ordenador. (Niño, Vargas, 2018, p.35).

Esta teoría se centra en clasificar los problemas computacionales acorde a su dificultad, y sus clases según Sipser (2013) son:

- Clase P: básicamente comprende a todos los problemas que se pueden resolver por un programa razonablemente rápido, problemas como multiplicación, laberintos y ordenación. La P proviene de que el tiempo en que son resueltos estos problemas en una máquina de Turing determinista es de comportamiento polinomial (*Polinomial*

HEURISTICAS PARA PROBLEMAS DE ENRRUTAMIENTO

time). Esta clase se conoce como una clase robusta que no es afectada por las particularidades del modelo computacional que se aborde y cuyos problemas se solucionan de forma realista en computador (Rivera & Jaramillo, 2020, p.30).

- Clase NP: Esta clase es de mayor complejidad puesto que contiene todos los problemas de la clase P, pero también abarca muchos otros importantes como, por ejemplo: ruteo vehiculos, planificación de trabajo, diseño de circuitos y plegamiento de proteínas. Estos problemas tienen la característica de que, a pesar de tomar una gran cantidad de tiempo para encontrar una solución al problema, esta se puede corroborar en un tiempo razonable, es decir, estos problemas tienen verificadores de tiempo polinomial. NP proviene de que el tiempo en el que son resueltos estos problemas en una máquina de Turing no determinística es polinomial (*Nondeterministic Polynomial time*) (Rivera & Jaramillo, 2020, p.30).
- Clase NP-Complete: Es un subgrupo de NP que abarca a todos los problemas esencialmente iguales, con algunas complicaciones en sus tiempos polinomiales lo cual los hace más difíciles que los NP. Algunos problemas ubicados acá son el sudoku, el plegamiento de las proteínas, el emparejamiento 3D, el knapsack problem y el satisfiability problem (Rivera & Jaramillo, 2020, p.31).
- Clase NP-Hard: Son problemas al menos tan difíciles como los problemas en NP y su alta complejidad radica en que en los problemas NP-Hard, a diferencia de los NP, los tiempos de solución como los tiempos de verificación crecen exponencialmente con el tamaño del problema y también en que algunos de estos problemas son de complejidad desconocida. Algunos ejemplos de estos problemas son: el ajedrez, el TSP y el problema de la suma de subconjuntos. (Rivera & Jaramillo, 2020, p.31).

3.7 Algoritmos de Solución

3.7.1 Algoritmos exactos

Los algoritmos consisten en un conjunto ordenado de pasos que aseguran la solución correcta a un problema, permiten describir una serie de instrucciones que debe realizar un computador para lograr un resultado. En el caso de los algoritmos exactos o de búsqueda exhaustiva se caracterizan por ser propios de cierto grupo de problemas, no presentan un alto grado de complejidad y proporcionan soluciones óptimas dentro de un conjunto de soluciones potenciales. Un ejemplo de este tipo de método se presenta en De Koster y Van der Poort (1998); donde construyen un algoritmo de programación dinámica con el propósito de calcular recorridos mínimos de selección de pedidos en almacenes con hasta tres pasillos transversales (Merlano y Castellanos, 2014, p.62).

Otro ejemplo de este tipo de método es la búsqueda con retroceso o backtracking, que trabaja tratando continuamente de extender una solución parcial. En cada etapa de la búsqueda, si una extensión de la solución parcial actual no es posible, se va hacia atrás para una solución parcial corta y se trata nuevamente, (Merlano y Castellanos, 2014, p.62).

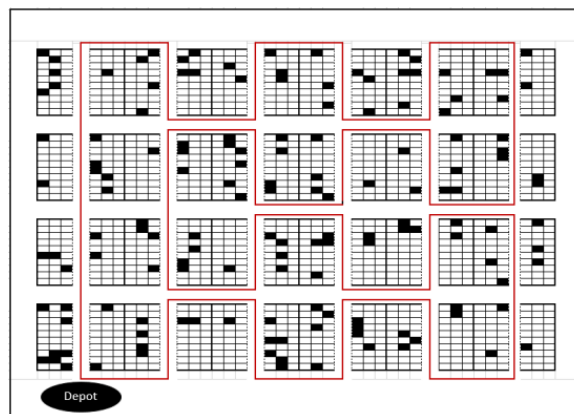
3.7.2 Heurística

Se denomina heurística al arte de inventar. En programación se dice que un algoritmo es heurístico cuando la solución no se determina en forma directa, sino mediante ensayos, pruebas y reensayos. El método consiste en generar candidatos de soluciones posibles de acuerdo con un patrón dado; luego los candidatos son sometidos a pruebas de acuerdo con un criterio que caracteriza a la solución. Si un candidato no es aceptado, se genera otro; y los pasos dados con el candidato anterior no se consideran. Es decir, existe inherentemente una vuelta atrás, para comenzar a generar un nuevo candidato; por esta razón, este tipo de algoritmo también se denomina "con vuelta atrás" (Chavarria y Zuñiga, 2019, p.25).

HEURISTICAS PARA PROBLEMAS DE ENRRUTAMIENTO

3.7.2.1 S-Shape. Se puede considerar como la heurística de enrutamiento más básica. Consiste en que todos los subpasillos con producto se deben atravesar completamente, iniciando cada uno desde el pasillo de cruce donde finalizó el anterior, tomando así la forma de “S”. Los subpasillos sin producto a recoger se omiten. El recolector debe iniciar por recorrer completamente el pasillo con producto que esté en el extremo izquierdo o derecho, según sea el más cercano al punto de inicio hasta llegar al pasillo de cruce posterior del almacén. A partir de ahí se visitan uno a uno los subpasillos pertenecientes al bloque más lejano hasta terminar en el extremo contrario del almacén. El único caso en donde no es necesario atravesar completamente un subpasillo es cuando éste es el último del bloque, en este caso se recoge el último producto que contiene y se regresa al pasillo de cruce de donde se partió. Al cambiar de bloque se debe iniciar con el subpasillo más cercano al último del bloque anterior y se explora siempre en dirección descendente. Al finalizar con todos los subpasillos con producto el recolector debe regresar al punto de inicio. (Arriola y Ramos, 2019, p.34).

Figura 18 *Ejemplo recolección de una ruta s-shape*

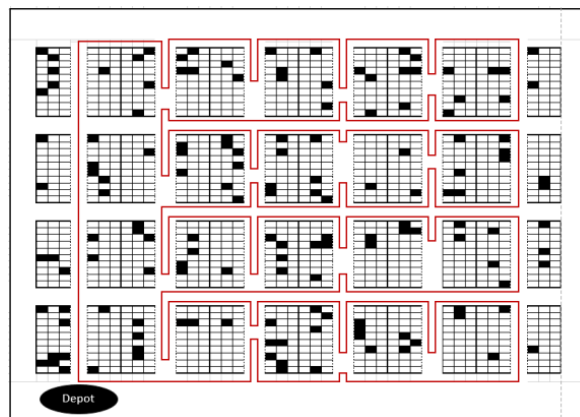


Tomado de “Framework basado en heurísticas que optimizan la ruta para recolectar productos en almacenes” (p.35), por R. Arriola y F. Ramos, 2019.

HEURISTICAS PARA PROBLEMAS DE ENRRUTAMIENTO

3.7.2.2 La brecha más grande. Esta heurística consiste en identificar cual es la brecha más larga en cada subpasillo y evitar recorrerla. La brecha puede ser ya sea entre el pasillo de cruce superior y el primer producto a recoger, entre productos adyacentes o del último producto del subpasillo al pasillo de cruce inferior. Todos los productos que se encuentren por encima de la brecha más larga serán tomados por el pasillo de cruce superior y el recolector deberá regresar al pasillo de cruce por donde ingresó y debe hacer lo mismo hasta explorar todos los subpasillos del bloque, para luego tomar todos los productos restantes desde el pasillo de cruce inferior. Los únicos subpasillos que se atraviesan completamente son los pertenecientes al primer pasillo que se encuentra al extremo izquierdo o derecho que se esté más cerca del punto de inicio, y el ultimo de cada bloque (de esta forma se pasa de un pasillo de cruce a otro). Al terminar de recoger todos los productos se debe regresar al depot. (Arriola y Ramos, 2019, p.35).

Figura 19 *Ejemplo recolección de una ruta la brecha más grande*

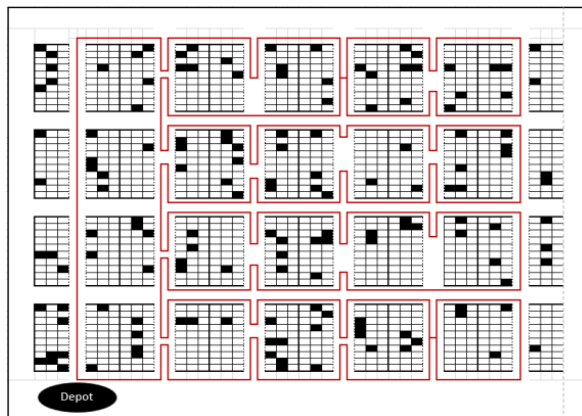


Tomado de “Framework basado en heurísticas que optimizan la ruta para recolectar productos en almacenes” (p.36), por R. Arriola y F. Ramos, 2019.

HEURISTICAS PARA PROBLEMAS DE ENRRUTAMIENTO

3.7.2.3 Midpoint. Es una política de enrutamiento similar a Largest Gap con la diferencia de que aquí se identifica el centro de cada subpasillo con producto y el recolector en primera instancia toma únicamente los productos que se encuentran en la mitad superior para luego recolectar los productos restantes desde el pasillo de cruce inferior. En caso de que el producto se encuentre exactamente en el centro se toma desde cualquiera de los dos pasillos de cruce. Al terminar de recolectar todos los productos se debe regresar al depot. (Arriola y Ramos, 2019, p.36).

Figura 20 Ejemplo recolección de una ruta midpoint

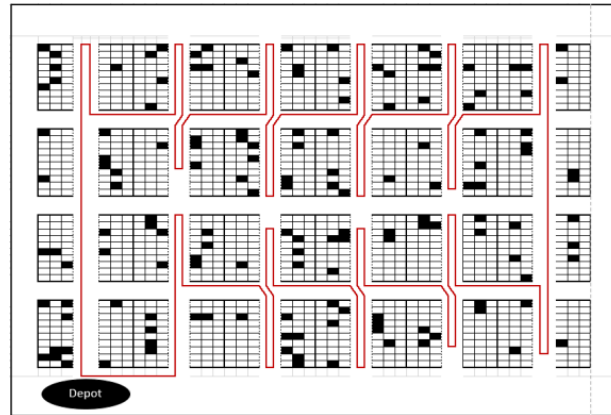


Tomado de “Framework basado en heurísticas que optimizan la ruta para recolectar productos en almacenes” (p.37), por R. Arriola y F. Ramos, 2019.

3.7.2.4 Return. Esta política de enrutamiento consiste en que al explorar los subpasillos, el recolector entra y sale por el mismo pasillo de cruce, es decir, al tomar el último elemento regresa para luego continuar con el próximo subpasillo. En caso de que la configuración del almacén contenga más de un bloque, el recolector visita los subpasillos de los dos bloques adyacentes al pasillo de cruce alternadamente. Si el número de bloques es superior a tres, al finalizar con ambos se debe trasladar al pasillo de cruce el cual sea adyacente a los siguientes bloques sin explorar. Al finalizar se debe regresar al depot. (Arriola y Ramos, 2019, p.37).

Figura 21 Ejemplo recolección de una ruta return

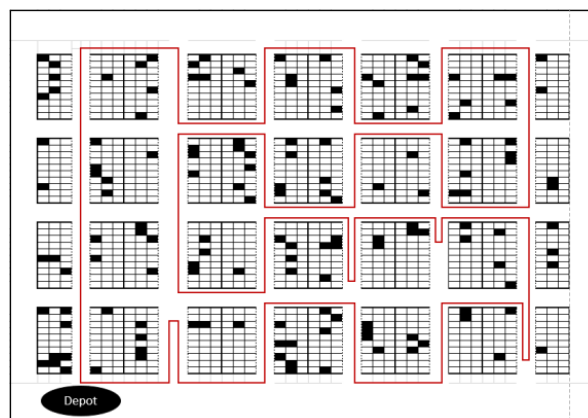
HEURISTICAS PARA PROBLEMAS DE ENRRUTAMIENTO



Tomado de “Framework basado en heurísticas que optimizan la ruta para recolectar productos en almacenes” (p.38), por R. Arriola y F. Ramos, 2019.

3.7.2.5 Combined. En esta heurística, el recolector siempre recoge todos los elementos contenidos en el subpasillo, solo que, al recoger el último debe decidir entre (1) continuar por el pasillo de cruce frontal, o (2) regresar al pasillo posterior. Esta decisión se toma según la distancia más corta al siguiente nodo con producto dentro del mismo bloque. El recolector debe iniciar por recorrer completamente el primer pasillo de recolección hasta llegar al pasillo de cruce posterior del almacén. A partir de ese punto se recorren los subpasillos por bloque hasta llegar al pasillo de cruce frontal y de esta forma regresas al depot. (Arriola y Ramos, 2019, p.38).

Figura 22 Ejemplo recolección de una ruta combined



Tomado de “Framework basado en heurísticas que optimizan la ruta para recolectar productos en almacenes” (p.39), por R. Arriola y F. Ramos, 2019.

HEURISTICAS PARA PROBLEMAS DE ENRRUTAMIENTO

3.7.3 *Metaheurística*

El término metaheurística se obtiene de anteponer a heurística el sufijo “meta” que significa “más allá” o “a un nivel superior”. Los conceptos actuales de lo que es una metaheurística están basados en las diferentes interpretaciones de lo que es una forma inteligente de resolver un problema. Las metaheurísticas son: estrategias inteligentes para diseñar o mejorar procedimientos heurísticos muy generales con un alto rendimiento (Belen et al., s.f)

- **Algoritmo genético:** Se encuentran dentro de los algoritmos evolutivos e imitan aspectos del comportamiento humano, tales como el aprendizaje, la percepción, la evolución y la adaptación (Merlano y Castellanos, 2014, p.66).
- **Optimización de enjambre de partículas:** Es usado para resolver problemas no lineales y está orientado a encontrar mínimos o máximos globales (Merlano y Castellanos, 2014, p.66)
- **Optimización por colonia de hormigas:** Es un algoritmo de tipo poblacional y constructivo, se inspiran directamente en el comportamiento de las colonias reales de hormigas para solucionar problemas de optimización combinatoria (Merlano y Castellanos, 2014, p.67).
- **Recocido simulado:** Ha sido empleado para resolver problemas combinatorios. La idea surge del proceso físico conocido como recocido, en el cual se eleva la temperatura de un sólido hasta el punto de que se vuelve líquido, a continuación, la temperatura se disminuye de forma paulatina para obtener una estructura cristalina sin defectos (estado basal) (Merlano y Castellanos, 2014, p.68).
- **Búsqueda tabú:** Se caracteriza por el uso de memoria adaptativa y estrategias especiales de resolución de problemas. Ya que este tipo de metaheurística es destacado por resolver problemas de optimización duros (NP- Hard), se ha incrementado el

HEURISTICAS PARA PROBLEMAS DE ENRRUTAMIENTO

número de aplicaciones de búsqueda Tabú durante los últimos años (Merlano y Castellanos, 2014, p.68).

3.8 Herramienta computacional Python

Es un lenguaje de programación multiplataforma y de código abierto que puede utilizarse tanto para desarrollo web, creación de software y procesamiento de datos, entre muchos otros propósitos. Esta versatilidad y facilidad para aprenderlo (es ampliamente considerado el lenguaje más sencillo de aprender) lo han convertido en el lenguaje de programación más popular del mundo, Según el índice PYPL.

Python se basa en los lenguajes C y C++ y tiene sus raíces en el sistema operativo UNIX. Python existe desde hace años, pero no fue hasta mediados de la década de 2000 cuando se afianzó en el mundo del desarrollo web, con el auge de sitios como WordPress.

4 Descripción del problema y modelo matemático

4.1 Descripción del problema

Se considera un almacén convencional de dos bloques iguales con un único depósito, donde se van a hacer variaciones en el número de recolectores, los cuales transportan los artículos demandados por medio de carros teniendo en cuenta la capacidad máxima, con respecto al tipo de almacenamiento se lleva a cabo la clasificación ABC, según la empresa Mecalux (2020) afirma que la clasificación ABC consiste en organizar la distribución de las distintas mercancías dentro del almacén a partir de su relevancia para la empresa, de su valor y de su rotación. Con este sistema se prioriza la adquisición y colocación de los productos no por su volumen o cantidad, sino por el aporte económico que suponen para las empresas. Las referencias se clasifican en tres niveles:

- **Artículos con rotación A:**

En cantidad, suelen ocupar el 20% de los inventarios, pero son los que más rotación experimentan y, por tanto, tienen una importancia estratégica, las referencias A son los

HEURISTICAS PARA PROBLEMAS DE ENRRUTAMIENTO

productos en los que la empresa tiene invertido más presupuesto y generan el 80% de los ingresos, a la hora de ubicar estas referencias en el almacén, tienden a situarse en zonas bajas, de acceso directo y fácil para el operario.

- **Artículos con rotación B:**

Comprenden la franja de rotación media y suelen representar, en cantidad, el 30% de los inventarios, estos artículos se renuevan con menos velocidad, por lo que su valor y relevancia es menor frente a los productos A, en este caso, hay que prestar atención a la evolución de las referencias clasificadas como B por si pueden dar el salto a la rotación A o, en cambio, convertirse en productos C, en el almacén, se ubican en zonas de altura intermedia cuyo acceso no es tan directo como en las posiciones que ocupan los productos A, pero tampoco resultan ser las más inaccesibles.

- **Artículos con rotación C:**

En su conjunto, los productos C son los más numerosos, llegando a suponer el 50% de las referencias almacenadas. Sin embargo, también son los menos demandados por parte de los clientes, en la instalación de almacenaje, como se necesita acceder a ellos de manera esporádica, ocupan las zonas más altas o menos accesibles.

Teniendo en cuenta las anteriores consideraciones se quiere llegar a una solución que defina la ruta de los recolectores en el almacén con un tiempo mínimo de cómputo y la mínima distancia recorrida.

El problema reside en la programación para un conjunto de P pedidos $p = \{1, 2, \dots, p\}$, donde los productos están ubicados en un conjunto V de posiciones $V = \{1, 2, \dots, v\}$ dentro del almacén y son almacenados en un conjunto S de SKU o unidades de almacenamiento $S = \{1, 2, \dots, s\}$.

Considerando el impacto del costo en tiempo de desplazamiento en la operación de un almacén y el costo de encontrar la solución más factible, se han desarrollado técnicas

HEURISTICAS PARA PROBLEMAS DE ENRRUTAMIENTO

heurísticas que permiten organizar la ruta para la recolección de pedidos de una orden que ayudan a disminuir el tiempo de recorrido (Scholz, Schubert, & Wäsher, 2017). Cada una de estas estrategias intuitivas tienen distinto costo en tiempo de recorrido y en general no obtienen el tiempo óptimo. Sin embargo, son muy usuales en la práctica debido al bajo costo de implementación y tiempo requerido para la planificación (Vargas et al., 2020).

Las dos heurísticas que se utilizan en esta investigación son s-shape y la brecha más grande.

4.2 Modelo Matemático

4.2.1 Formulación del problema

- **Índices**

Los índices del modelo son:

K: conjunto de SKU

O: número de órdenes de pedido

s: depósito

i: posición actual del recorrido

v: posición inicial de alistamiento del pedido

- **Parámetros**

P: conjunto de items existentes en el almacén

L: conjunto de localizaciones

Pos: conjunto de posiciones de almacenamiento

B: capacidad de carro de recolección

γ_v : número de elementos almacenados en la posición de almacenamiento v

a_{os} : cantidad de artículos de SKU s exigido por orden O.

D_{ij} : Distancia entre las posiciones i y j

T: Recolector

HEURISTICAS PARA PROBLEMAS DE ENRRUTAMIENTO

- **Consideraciones**

- Se establece un carro para recolectar todas las órdenes.
- La orden puede estar compuesta por varios ítems; pero en ningún caso puede superar la capacidad del carro (Pascual,2019).
- Se considera un almacén convencional con dos bloques iguales compuesto de pasillos rectos con estanterías de almacenaje que se apilan a su vez verticalmente en distintas alturas, considerando un almacenamiento (ABC) (Pascual,2019).
- Cada recolector se le asigna un único carro, es decir que recolector es igual a carro (Pascual,2019).
- El carro tendrá una capacidad fija, independientemente de la forma o tamaño de los ítems (Pascual,2019).
- Se considera dato todo aquello que define el problema que se pretende abordar. Para el problema trabajado se definen los siguientes datos y conjuntos de datos:
 - Conjunto P: hace referencia al conjunto de productos existentes en el almacén. Todos ellos tienen un espacio de almacenamiento asignado y conocido (Pascual,2019, Conjunto de datos del problema, párr.3).
 - Conjunto L: hace referencia al conjunto de localizaciones (posiciones en el centro del pasillo) desde las cuales el recolector puede alcanzar una serie determinada de productos ubicados a ambos lados (Pascual,2019, Conjunto de datos del problema, párr.4).
 - En el grafo que define el problema, los nodos serán un conjunto compuesto de dicho conjunto de localizaciones L además de:

El nodo origen, representado como $\{s\}$, que será punto de partida y de retorno para todos los carros utilizados (Pascual,2019, Conjunto de datos del problema, párr.6).

Una serie de localizaciones artificiales LA, que conectan los nodos del conjunto L para formar el layout del almacén, pero desde los cuales no se alcanza ningún producto.

HEURISTICAS PARA PROBLEMAS DE ENRRUTAMIENTO

Representa los cruces de pasillos y/o subpasillos (Pascual,2019, Conjunto de datos del problema, párr.7).

-Los nodos, también denominados vértices, que componen el grafo son los siguientes (Pascual,2019, Conjunto de datos del problema, párr.8).

$$V = \{L, LA, s\}$$

- Conjunto Pos: hace referencia al conjunto de posiciones de almacenamiento disponibles (ABC), dónde se almacenan, o no, productos. Desde cada localización L se podrán alcanzar un subconjunto de posiciones. Como se ha mencionado en la definición del conjunto P, cada producto tiene asignado su posición de almacenaje y son previamente conocidas (Pascual,2019, Conjunto de datos del problema, párr.9).

- Conjunto O: hace referencia al conjunto de órdenes que se reciben por parte de los clientes y de las cuales deben recogerse todos los productos que las componen. Se representará mediante el índice 'o', es decir, PO será el subconjunto de productos P que componen el pedido o, y a su vez LO representará el subconjunto de localizaciones L desde las que se alcanzan todos los PO (Pascual,2019, Conjunto de datos del problema, párr.10).

- Dij: hace referencia a la distancia que hay entre el nodo i y el nodo j del grafo, es decir, distancias entre posiciones consecutivas del almacén. Esta matriz de datos es la que define el Layout del almacén. Se trata de una matriz simétrica, ya que para cada arco i-j existe otro arco j-i tal que $d_{ij} \equiv d_{ji}$. Todas las distancias han de ser positivas o cero en caso de que no exista conexión entre los nodos implicados (Pascual,2019, Conjunto de datos del problema, párr.11).

- Parámetro B: Capacidad de los carros, es decir, el peso máximo que pueden almacenar en un mismo recorrido, carro o recolector. Es conocido y fijado al inicio del problema (Pascual,2019, Conjunto de datos del problema, párr.12).

HEURISTICAS PARA PROBLEMAS DE ENRRUTAMIENTO

- T: hace referencia al único recolector. Se representa mediante el índice 't'. En nuestro caso se define de la siguiente manera.

$$T = t$$

Además, de los conjuntos previamente definidos se obtiene una serie de datos adicionales como son los siguientes:

- Número de pedidos a recoger.
- El código de identificación de cada producto demandado dentro de cada pedido.
- Las cantidades que se demanda de cada producto dentro de cada pedido

(Pascual,2019, Conjunto de datos del problema, párr.14).

- **Variables**

Las variables involucradas en el problema que se estudia son las que se definen a continuación (Pascual,2019, Variables que intervienen en el problema, párr.2).

- Zot: se trata de una variable binaria. Su valor en la solución determina el carro que recoge el pedido. Si esta variable Zot vale 1, es decir, se activa, indicará que el pedido 'o' es asignado al carro 't'. Si por el contrario el valor que toma la variable es 0, es decir, no se activa, el pedido 'o' no será asignado al carro 't' (Pascual,2019, Variables que intervienen en el problema, párr.3).

- Xtij: se trata de una variable binaria. Su valor en la solución determina la ruta que realiza cada carro. Si esta variable Xtij vale 1, es decir, se activa, indicará que el carro 't' recorre el arco descrito por los nodos consecutivos i-j, donde 'i' representa el nodo de inicio y 'j' el nodo de salida. Si por el contrario el valor que toma la variable es 0, es decir, no se activa, el arco i-j no formará parte del recorrido del carro 't' (Pascual,2019, Variables que intervienen en el problema, párr.4).

- Yti: se trata de una variable binaria. Su valor en la solución determina los nodos visitados por cada carro. Si esta variable Yti vale 1, es decir, se activa, indicará que el carro 't'

HEURISTICAS PARA PROBLEMAS DE ENRRUTAMIENTO

visita el nodo 'i'. Si por el contrario el valor que toma la variable es 0, es decir, no se activa, el nodo 'i' no será ninguno de los nodos visitados por el carro 't' (Pascual,2019, Variables que intervienen en el problema, párr.5).

- E_{tj} : Se trata de una variable binaria. Se ha definido de manera auxiliar para indicar que un carro recorre el camino entre dos nodos consecutivos i-j, sin tener en cuenta en qué dirección se realice el movimiento. Si esta variable E_{tj} vale 1, es decir, se activa, indicará que el carro 't' recorre el arco i-j y/o el arco j-i. El orden de los subíndices de la variable no indica en este caso el orden de visita (Pascual,2019, Variables que intervienen en el problema, párr.6).

- G_{ti} : se trata de una variable entera. Su valor en la solución determina el grado de cada nodo para el recorrido de cada carro. El grado de un nodo o vértice, tal y como se muestra en la siguiente imagen, indica el número de arcos parten de él. El valor entero que tome la variable G_{ti} indicará el número de arcos que recorre el carro 't' con nodo de inicio 'i'. Por ejemplo, si G_{12} toma el valor 3, el carro 1 recorrerá tres arcos que parten del nodo 2 hacia otros nodos (Pascual,2019, Variables que intervienen en el problema, párr.7).

- **Restricciones**

- Para los tres almacenes se consideró un único depósito, un recolector de manera general y a su vez se tuvo en cuenta para $n=20$, $n=40$ y $n=60$ productos.

- Como se ha explicado en apartados anteriores, el problema que se estudia se representa como un grafo $D = \{V, A\}$. En éste, los nodos representan las localizaciones del almacén desde las cuales se pueden alcanzar una serie de productos almacenados o los cruces entre pasillos y los arcos la unión entre estas localizaciones, que estarán definidos como 0 o positivos según el layout del almacén. Lo que se quiere asegurar con esta restricción es que si un pedido está asignado a un carro (Z_{ot} activa) entonces cada nodo que contiene productos de ese pedido deberá ser visitado al menos una vez por el carro 't'. Esta relación se expresa matemáticamente de la siguiente manera (Pascual,2019, Restricciones, párr.3).

HEURISTICAS PARA PROBLEMAS DE ENRRUTAMIENTO

$$\sum_{(ij) \in i^+} X_{tij} \geq z_{ot} \quad \forall o \in O, \quad t \in T, \quad i: l(i) \in L_o$$

Se ha definido anteriormente que para cada arco ij existe un arco ji , por tanto, aunque se alcance un nodo ' i ' desde el nodo ' $i-1$ ' y se pretenda recorrer el camino inverso, es decir, regresar al nodo ' $i-1$ ', se recorrerán arcos diferentes. De este modo, un carro no recorrerá un mismo arco más de una vez. Para regularizar la continuidad de flujo en el grafo, se establece la siguiente restricción que asegura que cada carro que alcance un nodo i por alguno de sus arcos contenidos en i^- , debe abandonar el mismo por alguno de los arcos pertenecientes a i^+ . Esta relación se expresa matemáticamente de la siguiente manera (Pascual,2019, Restricciones, párr.4).

$$\sum_{(ij) \in i^+} X_{tij} = \sum_{(j,i) \in i^-} X_{tij} \quad \forall i \in V, \quad t \in T$$

Uno de los requisitos que deben cumplirse a la hora de la recogida de pedidos, es que los recolectores deben comenzar su ruta desde la posición origen, definida como $\{s\}$ y deberán terminar su ruta en la misma posición. Por tanto, esta restricción indica que, si un carro ' t ' recoge un determinado pedido ' o ', es decir Z_{ot} se activa, se tiene que activar un arco de salida del origen s para este carro. Esta relación se expresa matemáticamente de la siguiente manera (Pascual,2019, Restricciones, párr.5).

$$\sum_{(s,j) \in s^+} X_{tsj} \geq z_{ot} \quad \forall t \in T, \quad o \in O$$

Otra de las relaciones entre variables del problema que debe establecerse es que, si un carro recorre un arco, por fuerza ese carro ha debido visitar el nodo conjunto de carros o recolectores disponibles del que parte. Es decir, si la variable X_{tij} se activa, deberá activarse también Y_{ti} ya que implica que el carro ' t ' visita el nodo ' i ' y en caso contrario sería imposible que X_{tij} se activase. Esta relación se expresa matemáticamente de la siguiente manera (Pascual,2019, Restricciones, párr.7).

HEURISTICAS PARA PROBLEMAS DE ENRRUTAMIENTO

$$Y_{ti} \geq X_{tij} \quad \forall (i,j) \in A, \quad t \in T$$

- **Función Objetivo**

Esta busca minimizar las distancias totales recorridas asociadas a realizar la ruta correspondiente teniendo en cuenta los recorridos de las heurísticas S-shape y la Brecha más grande.

$$MIN \sum_{i=0}^n \sum_{j=0}^m Dij * Xij \quad i \neq j$$

$i = 1,2,3, 4, \dots, n$; $n =$ Subíndice que representa los materiales

$j = 1,2,3, 4, \dots, m$; $m =$ subíndice que representa las ubicaciones

- **Notación del modelo matemático**

F.O

$$MIN \sum_{i=0}^n \sum_{j=0}^m Dij * Xij \quad i \neq j$$

s.a:

$$\sum_{(ij) \in i^+} X_{tij} \geq z_{ot} \quad \forall o \in O, \quad t \in T, \quad i: l(i) \in l_o \quad (1)$$

$$\sum_{(ij) \in i^+} X_{tij} = \sum_{(j,i) \in i^-} X_{tji} \quad \forall i \in V, \quad t \in T \quad (2)$$

$$\sum_{(s,j) \in s^+} X_{tsj} \geq z_{ot} \quad \forall t \in T, \quad o \in O \quad (3)$$

$$Y_{ti} \geq X_{tij} \quad \forall (i,j) \in A, \quad t \in T \quad (4)$$

5 Diseño de los algoritmos

Tabla 2. Algoritmo brecha más grande

Algoritmo: la brecha más grande
input: una lista de selecciones
output: Secuencia de nodos a visitar
<ol style="list-style-type: none"> 1. Recuperar todos los subpasillos con al menos 1 selección 2. Agrupe todo estos subpasillo (denote un subpasillo sa) con selecciones en bloque y clasifique los bloques en distancia decreciente desde el depósito para obtener B_s 3. Determinar el subpasillo más cercano (más a la izquierda o a la derecha) con selección en el bloque más alejado del depósito 4. Atraviese la ruta más corta P desde el depósito hasta el frente de este subpasillo y luego hasta la selección más alejada en el subpasillo 5. Elimine todas las selecciones de todos los bloques que se encuentran en esta ruta más corta para obtener los bloques resultantes $B_s^{(1)}$ 6. Elimine todos los subpasillos $B_s^{(1)}$ en que ya no tengan selecciones para obtener los bloques resultantes $B_s^{(2)}$ 7. # Divida cada subpasillo en $B_s^{(2)}$ en subpasillos delantero y trasero, luego combine todos los sa traseros no vacíos (respectivamente delanteros) para obtener $B_s^{(3)}$, una lista de medios bloques $B_s^{(3)} \leftarrow []$ 8. for b in B_s do 9. Frente_b $\leftarrow []$ 10. Atrás_b $\leftarrow []$ 11. for sa in b do 12. $d_1 \leftarrow$ distancia total al recoger todas las selecciones desde el frente de sa 13. $d_2 \leftarrow$ distancia total al recoger todas las selecciones desde atrás de sa 14. if length(sa) > 1 then 15. Calcule la brecha entre todos los pares de nodos adyacentes en sa 16. $d_3 \leftarrow$ max (longitud de subpasillo - brecha entre 2 nodos adyacentes) 17. end if 18. $min_d = \min \{ d_1, d_2, d_3 \}$ 19. if $min_d = d_1$ then 20. frente_b \leftarrow frente_b + sa 21. else if $min_d = d_2$ then 22. atrás_b \leftarrow atrás_b + sa 23. else 24. frente_b \leftarrow frente_b + frente_ sa 25. atrás_b \leftarrow atrás_b + inversa (atrás_ sa) 26. end if 27. end for 28. $B_s^{(3)} \leftarrow B_s^{(3)} + [atrás_b, frente_b]$ 29. end for 30. nodo_secuencia P

HEURISTICAS PARA PROBLEMAS DE ENRRUTAMIENTO

Tabla 3. Algoritmo S-Shape

Algoritmo: S-Shape
input: una lista de selecciones
output: secuencia de nodos a visitar
<ol style="list-style-type: none"> 1. Recuperar todos los subpasillos con al menos 1 selección 2. Agrupe todo estos subpasillo (denote un subpasillo sa) con selecciones en bloque y clasifique los bloques en distancia decreciente desde el depósito para obtener B_s 3. Determinar el subpasillo más cercano (más a la izquierda o a la derecha) con selección en el bloque más alejado del depósito 4. Atraviese la ruta más corta P desde el depósito hasta el frente de este subpasillo y luego hasta la selección más alejada en el subpasillo 5. Elimine todas las selecciones de todos los bloques que se encuentran en esta ruta más corta para obtener los bloques resultantes $B_s^{(1)}$ 6. Elimine todos los subpasillos $B_s^{(1)}$ en que ya no tengan selecciones para obtener los bloques resultantes $B_s^{(2)}$ 7. $nodo_secuencia \leftarrow P$ 8. for b in $B_s^{(2)}$ do 9. Determine el sa más cercano (más a la izquierda o a la derecha) en el siguiente bloque desde la posición actual 10. if closest sa is rightmost sa in next block then 11. Invertir el orden de b 12. end if 13. # invertir el orden de los nodos en cada sa impar en cada bloque 14. for sa in b do 15. if sa is even-numbered and not last subaisle then 16. $nodo_secuencia \leftarrow nodo_secuencia + sa$ 17. else if sa is even-numbered and last subaisle then 18. $nodo_secuencia \leftarrow nodo_secuencia + camino$ a la última elección en sa 19. else 20. $nodo_secuencia \leftarrow nodo_secuencia + inversa(sa)$ 21. end if 22. end for 23. end for 24. Atraviesa todos los nodos en $nodo_secuencia$ 25. Regreso al depósito

6 Descripción del algoritmo

6.1 Estructura básica

El programa se compone de 5 archivos principales

1. **Algoritmos.py**: Contiene las funciones de los métodos de búsqueda de camino más corto
2. **Graphing.py**: Contiene el código usado para generar los grafos y mostrar de manera grafica los caminos usados por cada algoritmo, este se realizó usando la librería `igraph`
3. **Objetos.py**: Contiene clases que representan elementos básicos de la bodega (bloque y subpasillo) junto a otras claves auxiliares (grafo)
4. **Parámetros.py**: Contiene variables a usar para todas las simulaciones (como la cantidad de pasillos o las dimensiones de los estantes)
5. **Main.py**: Ejecuta las simulaciones e imprime resultados

Cuando el código se ejecuta, el orden a grandes rasgos es el siguiente:

1. Se ejecuta `main.py`, el cual genera 1000 iteraciones de ambos algoritmos usados (`s-shape` y la brecha más grande) mediante la función `probar_algoritmo`, esta función recibe el algoritmo a usar y una lista de productos que el carrito debe recoger.
2. Al llamar `probar_algoritmo`, se va al archivo `graphing.py`, el cual toma datos de `parámetros.py` para determinar las dimensiones de los elementos de la bodega y realizar diferentes operaciones para generar el grafo e instanciar las clases almacenadas en `objetos.py`, una vez se han posicionado todos los nodos relevantes, se llama el algoritmo generado (sea `s-shape` o brecha más grande), el cual recibe
 - a. El grafo: Para tener los nodos con los que generar el camino
 - b. Los bloques de la bodega: Para tener los pasillos con productos a recoger
 - c. El nodo inicial: Para saber dónde inicia el carrito

HEURISTICAS PARA PROBLEMAS DE ENRRUTAMIENTO

3. El camino retornado por el algoritmo se gráfica y se retorna la distancia que este atravesó.
4. De vuelta en main.py, se almacenan estas distancias en listas, una para cada tipo de algoritmo, al finalizar el algoritmo se promedian los valores de cada lista y se realiza una última ejecución de cada algoritmo, las cuales exportan imágenes para mostrar la estrategia que cada uno usa.

6.2 Paso a paso

6.2.1 Parametros.py

```
1 # Dimensiones de imagen de salida
2 dims = 1000
```

La imagen es un cuadrado con 1000x1000 pixeles de acuerdo con este parámetro

```
1 # Margen alrededor
2 warehouse_margin = 1
```

Se agrega un margen de imagen para facilitar la visualización de los caminos

```
1 # Tamano de las repisas
2 shelf_width = 3
3 shelf_height = 10
```

Ancho y alto de los estantes

```
1 # Cantidad de pasillos horizontales y verticales
2 h_aisles = 2
3 v_aisles = 3
```

Cantidad de pasillos horizontales y verticales en el almacén, estos determinan de forma indirecta la cantidad de estantes

6.2.2 Main.py

```
1 import random
2 from statistics import mean
3 from graphing import probar_algoritmo, p_amount
4 from algoritmos import s_shape, brecha_mas_grande
```

HEURISTICAS PARA PROBLEMAS DE ENRRUTAMIENTO

Primero se importan las librerías de random (para generar la lista de productos que el carro debe recoger) y statistics (para obtener el promedio), del mismo modo se usan los archivos graphing.py para realizar las pruebas de algoritmos y el archivo algoritmos.py para obtener los algoritmos a probar.

```

1 # Productos seleccionados
2 iteraciones = 1000
3 cantidad_productos_min = 5
4 cantidad_productos_max = 50

```

Se establecen parámetros para la simulación, en este caso se especifican 1000 iteraciones, donde cada una genera una solicitud de productos entre 5 y 50, distribuidos de manera aleatoria.

```

1 # Colocar en True para exportar caminos en formato csv (esto para usar
   elverificador con matlab)
2 exportar_caminos = True
3 csv_s_shape = None
4 csv_brecha_mas_grande = None

```

También se establecen parámetros para determinar si la simulación debe generar archivos.csv que sean leídos por otros programas, en caso de necesitarse, se encontrarán en la carpeta csv. Además, se inicializan variables por defecto para la posición de estos caminos.

Es necesario tener en cuenta que la generación de estos archivos ralentiza la simulación de manera notable y solo debe usarse cuando sea necesario validar los caminos generados.

```

1 # Esto genera los productos aleatorios
2 productos_prueba = [list(set(random.randint(0,p_amount-1) for n in
   range(random.randint(cantidad_productos_min, cantidad_productos_max)))) for n in
   range(iteraciones)]
3 distancias_s = []
4 distancias_brecha = []

```

HEURISTICAS PARA PROBLEMAS DE ENRRUTAMIENTO

Con los parámetros anteriores genero una lista que contiene el conjunto de los productos solicitados para el carrito para cada simulación, como se especifican 1000 iteraciones, esta lista contiene 1000 conjuntos de productos. Además, inicializo listas para almacenar la distancia recorrida por el carrito por cada algoritmo.

```

1 # Contador de iteraciones
2 i = 0
3 for productos in productos_prueba:
4     if exportar_caminos:
5         csv_s_shape=f'ss_{i}'
6         csv_brecha_mas_grande = f'bmg_{i}'
7
8         d_s = probar_algoritmo(s_shape, productos, salida_csv=csv_s_shape)
9         d_brecha = probar_algoritmo(brecha_mas_grande, productos,
10            salida_csv=csv_brecha_mas_grande)
11
12        distancias_s.append(d_s)
13        distancias_brecha.append(d_brecha)
14
15    i += 1

```

Por cada conjunto de productos en la lista, se realiza una iteración, se corre el algoritmo S shape y la brecha más grande, las distancias resultantes se insertan en las listas correspondientes. Si además se especificó que deben generarse archivos csv, entonces se generan los nombres de los archivos de acuerdo con el número de iteración.

```

1 # Promedios
2 promedio_s = mean(distancias_s)
3 promedio_brecha = mean(distancias_brecha)
4 print (f'El promedio de distancia de la simulacin S shape fue de: {promedio_s}')
5 print (f'El promedio de distancia de la simulacin Brecha mas grande fue de:
6     {promedio_brecha}')

```

Una vez terminadas las simulaciones, se obtiene el promedio de cada lista y se imprimen

```

1 # Crear imágenes de algoritmos
2 d_s = probar_algoritmo(s_shape, productos_prueba[0], salida_imagen='s_shape.png')
3 d_brecha = probar_algoritmo(brecha_mas_grande, productos_prueba[0],
4    salida_imagen='brecha.png')

```

HEURISTICAS PARA PROBLEMAS DE ENRRUTAMIENTO

Para ver el comportamiento de cada algoritmo, se llama la función de probar algoritmo y se especifica que tengan salida de imagen, esto para mostrar visualmente los caminos tomados por cada algoritmo ante los mismos productos solicitados.

6.2.3 *Objetos.py*

```
1 from igraph import Graph, Vertex
```

Para las clases a crear, se toma la clase grafo (pues más adelante se le van a agregar funcionalidades) y la clase Vértice (que tiene un papel importante en el manejo de los pasillos).

```
1 class Subpasillo:
2     def __init__(self, bloque: Bloque, x: int) -> None:
3         self.bloque = bloque
4         self.x      =      x
5         self.vertices_L = []
6
7     def tiene_items(self) -> bool:
8         return bool([v for v in self.vertices_L if not v['visited']])
9
10    # Esta funcion retorna los vertices que limitan en subpasillo, en orden de
11    yascendente
12    def vertices(self) -> Vertex:
13        vertices=list(self.bloque.grafo.vs.select(x_eq=self.x, y_in=
14                [self.bloque.y_sup, self.bloque.y_inf]))
```

Primero se presenta la clase subpasillos, se llaman así por ser trozos de pasillos verticales que tienen al menos un estante a su alcance, estos también representan puntos donde el carrito pasa para recoger productos, de estos interesa saber si un subpasillo tiene productos por recoger, y los vértices que lo limitan (LA).

Para saber si tiene productos por recoger, se mira si los vértices L (que son puntos donde los productos se recogen) ya han sido visitados, si ya todos han sido visitados, no hay productos por recoger.

HEURISTICAS PARA PROBLEMAS DE ENRRUTAMIENTO

```

1 class Bloque:
2     def __init__(self, y_sup: int, y_inf: int, grafo: Graph) -> None:
3         self.y_sup = y_sup
4         self.y_inf = y_inf
5         self.grafo = grafo
6
7         self.subpasillos: 'list[Subpasillo]' = []
8
9     def tiene_items(self) -> bool:
10        return (any([s.tiene_items() for s in self.subpasillos]))
11
12    def subpasillos_items(self) -> 'list[Subpasillo]':
13        return [s for s in self.subpasillos if s.tiene_items()]

```

Un bloque es el conjunto de estantes que se encuentran entre 2 pasillos horizontales seguidos, cada bloque tiene una lista de subpasillos donde se encuentran los nodos L por los que el carrito debe pasar para recoger los productos, en ambos algoritmos los productos se recogen bloque por bloque (normalmente desde el bloque superior hasta el inferior).

```

1 class Grafo(Graph):
2     def __init__(self, *args, **kwargs):
3         super().__init__(*args, **kwargs)
4         self.bloques = []

```

Finalmente se tiene la clase Grafo, que es la clase Graph que arroja la librería igraph pero con funciones agregadas, la función mostrada arriba puede ignorarse, lo único importante es que el grafo ahora tiene una lista de bloques (self.bloques) además de tener vértices y aristas.

```

1 def vertice_nodos(self, origen, destino):
2     return self.es.find(_between=((origen.index,), (destino.index,)))

```

Esta función recibe 2 vértices y devuelve la arista que los conecta, para ello usa la función find de la librería igraph.

HEURISTICAS PARA PROBLEMAS DE ENRRUTAMIENTO

```

1 def camino_nodos(self, origen, destino) -> 'list[Vertex]':
2     camino = list(self.vs.select(self.get_shortest_paths(origen, destino,
3         weights=self.es["weight"])[0]))[1:]
4
5     for v in camino:
6         v['visited'] = True
7
8     return camino

```

Esta función retorna una lista de vértices con el camino más corto entre dos nodos, para esto usa la función get shortest paths, además itera sobre cada nodo para marcarlo como visitado (esto es importante para saber que subpasillos tienen productos por recoger).

```

1 def camino_a_aristas(self, camino):
2     aristas = []
3     for v1, v2 in zip(camino[:-1], camino[1:]):
4         arista = self.vertice_nodos(v1, v2)
5         arista['visitas'] += 1
6         aristas.append(arista)
7
8     return aristas

```

Esta función recibe un camino como el de la función de arriba, pero en vez de retornar vértices, retorna aristas, para esto itera sobre cada par de nodos y encuentra las aristas que los conectan.

6.2.4 Algoritmo.py

```

1 from objetos import Bloque, Grafo

```

Para los algoritmos se usa las clases definidas arriba, la clase subpasillo ya se encuentra usada por la clase Bloque, por lo que no se debe importar directamente

```

1 def s_shape(grafo, bloques, s):

```

Este será el algoritmo S Shape

HEURISTICAS PARA PROBLEMAS DE ENRRUTAMIENTO

```

1  # Obtengo los bloques con items
2  bloques = [b for b in bloques if b.tiene_items()]
3  camino = [s]
4
5  # Bloque mas lejano
6  bloque = bloques[0]
7
8  # Nodo mas cercano del bloque mas lejano
9  nodolej = bloque.subpasillos[0].vertices()[1]
10
11 camino.extend(grafo.camino_nodos(s, nodolej))
12
13 # Vuelvo a filtrar los bloques
14 bloques = [b for b in bloques if b.tiene_items()]

```

Primero, se necesita saber que bloques deben recorrer de acuerdo con si tienen productos o no.

```

1  for n, bloque in enumerate(bloques):
2
3      subpasillos = bloque.subpasillos_items()
4      i_ultimo_subpasillo = len(subpasillos) - 1
5
6      if n%2:
7
8          subpasillos = list(reversed(subpasillos))
9          bloque.subpasillos = subpasillos

```

Ahora, por cada bloque, se alterna entre empezar desde arriba y desde abajo del bloque, el $n \% 2$ se encarga de invertir los nodos a visitar de acuerdo con esto.

```

1  # Voy a la parte inferior del primer pasillo del bloque que voy a
2  recorrer
3  if camino[-1] != subpasillos[0].vertices()[1]:
4      camino.extend(grafo.camino_nodos(camino[-1],
5      subpasillos[0].vertices()[0]))
6      camino.extend(grafo.camino_nodos(camino[-1],
7      subpasillos[0].vertices()[1]))
8
9  subpasillos=bloque.subpasillos_items()
10 for i, sa in enumerate(subpasillos):

```

HEURISTICAS PARA PROBLEMAS DE ENRRUTAMIENTO

Esta parte hace que empiece en una esquina del bloque a recorrer y luego se itera sobre cada subpasillo del bloque

```

1  if not i%2 and i != i_ultimo_subpasillo:
2      # Atravieso el subpasillo desde abajo a la parte de arriba del
3      siguiente
4      camino.extend(grafo.camino_nodos(camino[-1], sa.vertices()[1]))
      camino.extend(grafo.camino_nodos(camino[-1], sa.vertices_L[0]))

```

Si el subpasillo es par y no es el último, entonces recorro el pasillo de abajo a arriba

```

1  elif not i%2 and i == i_ultimo_subpasillo:
2      # Obtengo el camino a los items del ultimo subpasillo
3      ultimo_item = sa.vertices_L[0]
4
5      camino.extend(grafo.camino_nodos(camino[-1], ultimo_item))
      camino.extend(grafo.camino_nodos(ultimo_item, camino[-1]))

```

Si el subpasillo es par, pero es el último, no debo atravesarlo del todo, solo recojo los productos del subpasillo y me devuelvo al inicio del subpasillo

```

1  else:
2      # Atravieso el subpasillo desde arriba a la parte de abajo del
3      siguiente
4      camino.extend(grafo.camino_nodos(camino[-1], sa.vertices()[0]))
5      camino.extend(grafo.camino_nodos(camino[-1], sa.vertices_L[-1]))

```

Si el subpasillo es impar, lo recorro de arriba abajo

```

1  # Finalmente regreso al deposito
2  camino.extend(grafo.camino_nodos(camino[-1],s))
3  return camino

```

Una vez se termina con los bloques, se va al punto de partida y se termina el algoritmo retornando los vértices que componen el camino

HEURISTICAS PARA PROBLEMAS DE ENRRUTAMIENTO

6.2.5 *Graphing.py*

```

1 import os
2 from typing import Callable
4 import igraph as ig
5 from objetos import Bloque, Subpasillo, Grafo
6 from parametros import *

```

Para el grafo se importan las librerías igrapg y typing

```

1 # V = {L, LA, s}
2 # s es el nodo inicial
3 # L es el conjunto de posiciones entre pasillos que se encuentran al
  alcance de un producto
4 # LA son el conjunto de nodos que conectan el layout del almacén
5 shelf_dimensions = (
6     shelf_width*2,
7     shelf_height
8 )
9 warehouse_dimensions = (
10    shelf_dimensions[0]*(v_aisles+1) + v_aisles,
11    shelf_dimensions[1]*(h_aisles+1) + h_aisles
12 )
13 p_amount = (v_aisles+1) * (h_aisles+1) * shelf_dimensions[0] *
14    shelf_dimensions[1]

```

Primero se calcula la representación de los estantes en el grafo con los parámetros, además se calcula la cantidad de estantes de acuerdo con la cantidad de pasillos, finalmente se calcula la cantidad total de productos que tendría la bodega de acuerdo con los parámetros

```

1 def probar_algoritmo(algoritmo: Callable, productos: 'list[int]',
    salida_imagen=None, salida_csv=None):

```

HEURISTICAS PARA PROBLEMAS DE ENRRUTAMIENTO

Esta es la función principal del archivo, y recibe una función (Callable), los productos a recoger (representados por índices de 0 a la cantidad de productos) y parámetros que determinan si debe exportar archivos csv o imagen.

```

1 # Creacion del grafo
2   g = Grafo()
3
4 # Poblar LA
5   la_amount = (2+v_aisles) * (2+h_aisles)
6   la_vertices = [g.add_vertex() for i in range(la_amount)]
7
8   for i, v in enumerate(la_vertices):
9       v['color'] = 'white'
10      v['type'] = 'LA'
11      la_coord = (
12          (i%(v_aisles+2)) * (shelf_dimensions[0] + 1),
13          (i//(v_aisles+2)) * (shelf_dimensions[1] + 1)
14      )
15
16      v['x'] = la_coord[0]
17      v['y'] = la_coord[1]

```

Acá se van poblando los nodos que delimitan los subpasillos (LA), para ello se calcula la cantidad de nodos a generar, enumerate permite hacer un for con 2 variables, donde la i es el número de iteración y la v el vértice actual

Se asigna colores y coordenadas a los vértices, además de marcarlos como nodos LA, las coordenadas se calculan como puntos (x,y)

```

1 # Poblar productos y L
2   p_vertices = [g.add_vertex() for i in range(p_amount)]
3
4   l_vertices = []
5   l_coords = []

```

Acá se generan los nodos que corresponden a puntos donde se recogen productos (L), para esto primero se generan todos los productos de antemano.

HEURISTICAS PARA PROBLEMAS DE ENRRUTAMIENTO

```

1  for i, v in enumerate(p_vertices):
2      v['size'] = 10
3      v['color'] = 'black'
4      v['shape'] = 'rectangle'
5      v['type'] = 'p'
6      v['pid'] = i
7      product_coord = (
8          ((i % (warehouse_dimensions[0] - v_aisles)) + ((i //
9              shelf_dimensions[0]) % (v_aisles+1))) + warehouse_margin,
10         ((i // (warehouse_dimensions[0] - v_aisles))) + (i // (p_amount //
11             (h_aisles+1))) + warehouse_margin
12     )
13
14     v['x'] = product_coord[0]
15     v['y'] = product_coord[1]

```

Para cada producto se asigna propiedades de color y forma, coordenadas e id de producto

```

1  if i in productos:
2      v['shape'] = 'hidden'
3      v['label'] = i
4      v['type'] = 'sp'
5
6      # # Verifica si esta a la derecha o izquierda de la repisa
7      x_delta = (shelf_width*2) - (i%(shelf_width*2))
8
9      if x_delta > shelf_width:
10         x_delta = x_delta - (shelf_width*2 + 1)
11
12         l_coord = (
13             product_coord[0] + x_delta,
14             product_coord[1]
15         )
16
17         if l_coord not in l_coords:
18             l_vertex = g.add_vertex()
19             l_vertex['color'] = 'black'
20             l_vertex['x'] = l_coord[0]
21             l_vertex['y'] = l_coord[1]
22             l_vertex['type'] = 'L'
23             l_vertex['visited'] = False
24
25             l_vertices.append(l_vertex)
26             l_coords.append(l_coord)

```

HEURISTICAS PARA PROBLEMAS DE ENRRUTAMIENTO

Por cada producto generado se mira cual producto esta para recoger, se busca el subpasillo más cercano y se genera un vértice L de manera acorde.

```

1  # Poblar s
2  s = g.add_vertex('Inicio')
3  s['shape'] = 'triangle-up'
4  s['color'] = 'black'
5  s['type'] = 's' s['x'] = 0
6  s['y'] = warehouse_dimensions[1] + 2

```

Para S basta con tomar el vértice en la esquina inferior izquierda, ese será el punto de partida

```

1  # Ahora con los nodos creados, genero la estructura de los datos
2  # que van a facilitar mi algoritmo
3
4  # Primero saco las coordenadas x de los pasillos verticales
5  v_aisle_coords = sorted([(shelf_width*2)+1)*i for i in range(v_aisles+2)])
6
7  # Ahora las coordenadas y de los pasillos horizontales
8  h_aisle_coords = sorted([(shelf_height+1)*i for i in range(h_aisles+2)])
9
10 # Conecto los vertices entre pasillos para crear los caminos que puede tomar
    el carro

```

Para sacar las coordenadas de los pasillos, se inicia desde 1 y se va sumando el ancho (o alto) de los pasillos, estas listas contienen todas las coordenadas posibles donde puede haber un nodo LA.

HEURISTICAS PARA PROBLEMAS DE ENRRUTAMIENTO

```

1  # Primero hago las conexiones verticales
2  for x in v_aisle_coords:
3      vs = g.vs.select(x_eq=x)
4      vs = sorted(list(vs), key=lambda v: v['y'])
5
6      # Creo las aristas
7      for v1, v2 in zip(vs[:-1], vs[1:]):
8          ar = g.add_edge(v1, v2)
9          ar['weight'] = v2['y'] - v1['y']
10         ar['color'] = 'grey'
11         ar['visitas'] = 0
12
13     # Ahora las horizontales
14     for y in h_aisle_coords:
15         vs = g.vs.select(y_eq=y)
16         vs = sorted(list(vs), key=lambda v: v['x'])
17
18         # Creo las aristas
19         for v1, v2 in zip(vs[:-1], vs[1:]):
20             ar = g.add_edge(v1, v2)
21             ar['weight'] = v2['x'] - v1['x']
22             ar['color'] = 'grey'
23             ar['visitas'] = 0

```

Con las coordenadas de los pasillos, se conectan los vértices en dichas coordenadas

para generar aristas (caminos) por los que el carrito va a pasarse

```

1  # Creo los objetos de bloques y subpasillos
2  bloques: 'list[Bloque]' = []
3  for y_sup, y_inf in zip(h_aisle_coords[1:], h_aisle_coords[:-1]):
4      bloque = Bloque(y_sup, y_inf, g)
5
6      for x in v_aisle_coords:
7          sa = Subpasillo(bloque, x)
8          sa.vertices_L.extend(
9              list(g.vs.select(type_eq='L', x_eq=x, y_gt=y_inf, y_lt=y_sup))
10             )
11
12         sa.vertices_L = sorted(sa.vertices_L, key=lambda x: x['y'])
13
14         bloque.subpasillos.append(sa)
15
16     bloques.append(bloque)

```

HEURISTICAS PARA PROBLEMAS DE ENRRUTAMIENTO

Para generar los bloques, se necesitan subpasillos, para generar subpasillos, se requiere las coordenadas de los nodos LA que lo delimitan.

```
1 # Corro el algoritmo insertado
2 camino = algoritmo(g, bloques, s)
```

Acá se corre el algoritmo insertado en la función, este retorna la lista de nodos que conforman el camino

```
1 # Si se especifica una salida de archivo csv
2 if salida_csv:
3     os.makedirs('csv', exist_ok=True)
4     # Defino el archivo de salida dentro de una carpeta
5     archivo_salida = f'csv/{salida_csv}.csv'
6     # Abro el archivo en modo escritura
7     with open(archivo_salida, 'w') as f:
8         # Por cada vertice en el camino generado, escribir sus coordenadas
9         for vertice in camino:
10            f.write(f' {vertice["x"]}, {vertice["y"]}\n')
```

Si se deben exportar los caminos en csv, basta con crear una carpeta, y un archivo csv que tenga las coordenadas (x, y) de cada vértice recorrido

```
1 # Con el camino dado por el algoritmo, dibujo el camino y obtendo
2   la distancia recorrida
3 distancia_total = 0
4 for arista in g.camino_a_aristas(camino):
5     arista['width'] = 2 *
6     arista['visitas']
7     arista['color'] = 'black'
8     distancia_total += arista['weight']
```

Además, con el camino se puede colorear las aristas para luego poder visualizar

```
1 # Si se especifica una salida, mostrar una imagen del camino
2 if salida_imagen:
3     layout = ig.Layout([(v['x'], v['y']) for v in g.vs]) #type: ignore
4     ig.plot(g, salida_imagen, layout=layout, bbox=(dims, dims)) #type: ignore
```

HEURISTICAS PARA PROBLEMAS DE ENRRUTAMIENTO

Si se especifica que se debe obtener una imagen, la genera, acá es notable que el layout se genera extrayendo las coordenadas (x, y) de cada vértice y poniéndolos en una lista

```
1 return distancia_total
```

El procedimiento retorna la distancia total recorrida por el algoritmo



```
parametros.py - C:\Users\UIS\Downloads\2022_06_20 (1)\parametros.py (3.10.6)
File Edit Format Run Options Window Help
# Dimensiones de imagen de salida
dims = 1000

# Margen alrededor
warehouse_margin = 1

# Tamaño de las repisas
shelf_width = 3
shelf_height = 10

# Cantidad de pasillos horizontales y verticales
h_aisles = 2
v_aisles = 3
```

Para correr el algoritmo se inicia realizando variaciones en `parámetros.py` en donde se puede definir el tamaño del almacén, se varia `h_aisles` y `v_aisles` para definir cuantos pasillos horizontales y verticales va a tener el almacén dependiendo del tamaño en donde se quiera analizar las distancias.

HEURISTICAS PARA PROBLEMAS DE ENRRUTAMIENTO

```

main.py - C:\Users\UIS\Downloads\2022_06_20 (1)\main.py (3.10.6)
File Edit Format Run Options Window Help
import random
from statistics import mean

from graphing import probar_algoritmo, p_amount
from algoritmos import s_shape, brecha_mas_grande

# Productos seleccionados

iteraciones = 1
cantidad_productos_min = 60
cantidad_productos_max = 60

# Esto genera los productos aleatorios
productos_prueba = [list(set(random.randint(0, p_amount-1) for n in range(random.randint(cantidad_
distancias_s = []
distancias_brecha = []

for productos in productos_prueba:
    d_s = probar_algoritmo(s_shape, productos)
    d_brecha = probar_algoritmo(brecha_mas_grande, productos)

    distancias_s.append(d_s)
    distancias_brecha.append(d_brecha)

# Promedios
promedio_s = mean(distancias_s)
promedio_brecha = mean(distancias_brecha)

print(f'El promedio de distancia de la simulación S shape fue de: {promedio_s}')
print(f'El promedio de distancia de la simulación Brecha mas grande fue de: {promedio_brecha}')

# Crear imagenes de algoritmos
d_s = probar_algoritmo(s_shape, productos_prueba[0], 's_shape.png')
d_brecha = probar_algoritmo(brecha_mas_grande, productos_prueba[0], 'brecha.png')

```

En main.py se realiza la variación de los productos que se quieren recolectar en # de productos seleccionados, se puede definir un rango colocando la mínima y máxima cantidad de productos a recolectar.

```

C:\Windows\System32\cmd.exe - "C:\Users\UIS\Downloads\2022_06_20 (1)\correr_programa.cmd"
Microsoft Windows [Versión 10.0.19044.2251]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\UIS\Downloads\2022_06_20 (1)>"C:\Users\UIS\Downloads\2022_06_20 (1)\correr_programa.cmd"

C:\Users\UIS\Downloads\2022_06_20 (1)>python main.py
El promedio de distancia de la simulación S shape fue de: 248
El promedio de distancia de la simulación Brecha mas grande fue de: 380

C:\Users\UIS\Downloads\2022_06_20 (1)>pause
Presione una tecla para continuar . . . █

```

Después de realizar las variaciones nombradas anteriormente se procede a correr el programa y este es el resultado que se obtiene después de cada corrida.

7 Evaluación del algoritmo

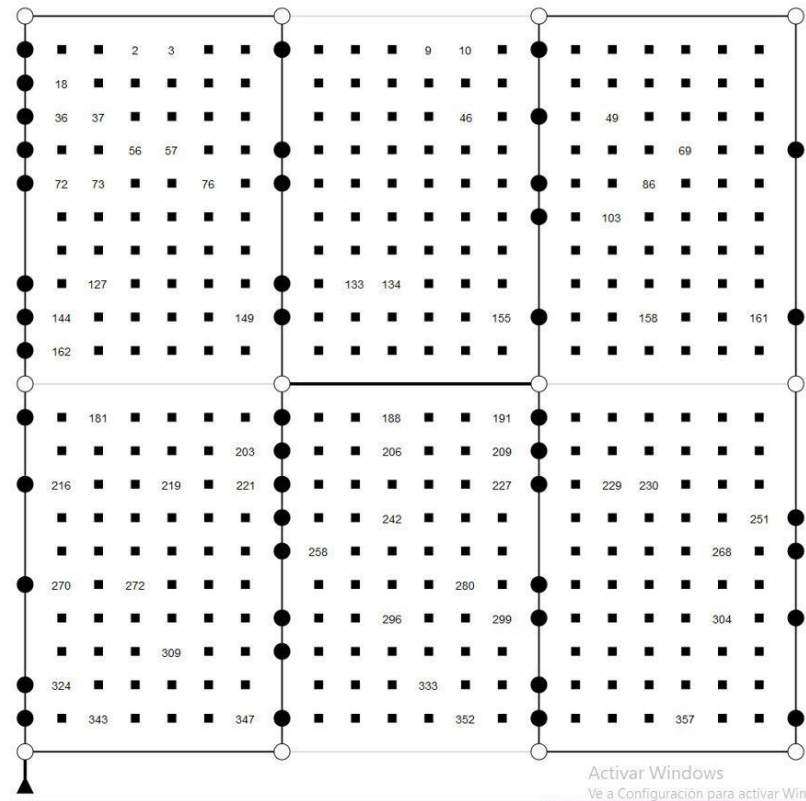
Para hacer el análisis del comportamiento de los algoritmos S-shape y la brecha más grande en el problema de Enrutamiento para la recolección de pedidos en un almacén convencional con líneas de dos bloques iguales se realizó un diseño de experimentos 2^k . los algoritmos se desarrollaron en Python versión 3.9. y el diseño factorial se realizó en el software estadístico statgraphics. Las características que se tuvieron en cuenta para el conjunto pequeño fue de 2 bloques con 1 pasillo horizontal, 2 pasillos verticales y cantidad SKU de 360, para el conjunto que se consideró mediano se tuvo en cuenta 2 bloques con 1 pasillo horizontal, 3 pasillos verticales y cantidad SKU de 480, para el conjunto que se consideró grande se tuvo en cuenta 3 bloques con 2 pasillos horizontales, 3 pasillos verticales y cantidad SKU 720, para los tres almacenes se hace variación en el número de ordenes donde se toma la decisión de realizar 3 tamaños por cada almacén clasificándolas en pequeña, mediana y grande que serían 20, 40 y 60 números de ordenes respectivamente.

7.1 Ruta realizada por cada heurística

El algoritmo realizado genera graficas donde nos permite verificar que el programa realizado está ejecutando las heurísticas de enrutamiento planteadas en esta investigación, para validar la ruta de cada heurística se revisará de forma conceptual y cualitativa la ejecución de cada heurística por medio de tres graficas tanto para la heurística s-shape como para la brecha más grande arrojadas por el programa Python.

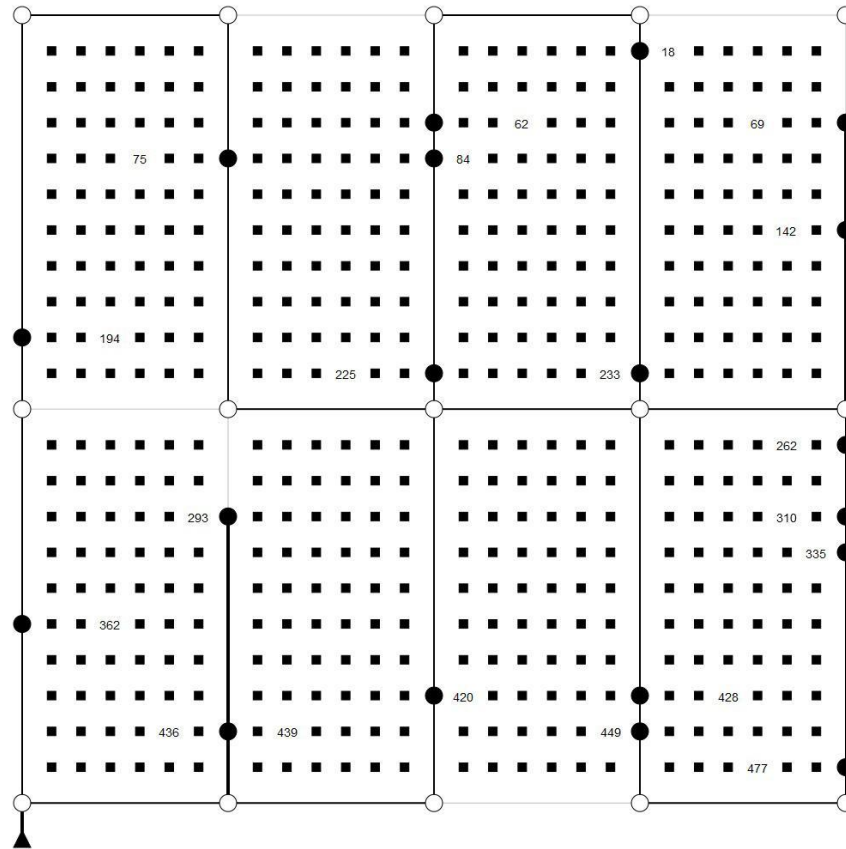
HEURISTICAS PARA PROBLEMAS DE ENRRUTAMIENTO

7.1.1 Graficas arrojadas de la heurística s-shape

Figura 23. ruta de la heurística s-shape en pequeñas instancias.

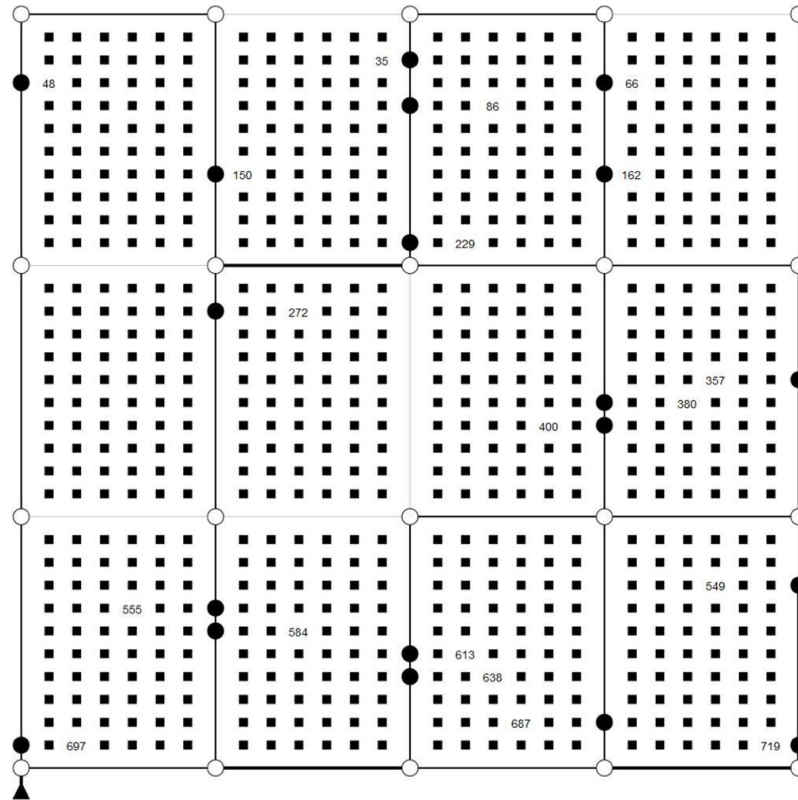
Se da inicio a la ruta desde el vértice en la esquina inferior izquierda en donde se atraviesa el pasillo en su totalidad al llegar al pasillo de cruce posterior del almacén, se visitan uno a uno los subpasillos pertenecientes al segundo bloque hasta terminar en el extremo contrario del almacén al cambiar de bloque se inicia con el subpasillo más cercano al último del segundo bloque y se explora en dirección descendente, al terminar de recorrer todos los subpasillos con producto el recolector regresa al punto de inicio.

HEURISTICAS PARA PROBLEMAS DE ENRRUTAMIENTO

Figura 24. ruta de la heurística s-shape en medianas instancias.

Se da inicio a la ruta desde el vértice en la esquina inferior izquierda en donde se atraviesa el pasillo en su totalidad al llegar al pasillo de cruce posterior del almacén, se visitan uno a uno los subpasillos pertenecientes al segundo bloque, en el cuarto subpasillo del bloque dos no es necesario recorrer completamente el subpasillo ya que es el último del bloque, por esta razón se recoge el último producto y se regresa al pasillo de cruce de donde se partió al cambiar de bloque se inicia con el subpasillo más cercano al último del segundo bloque y se explora en dirección descendente, al terminar de recorrer todos los subpasillos con producto el recolector regresa al punto de inicio.

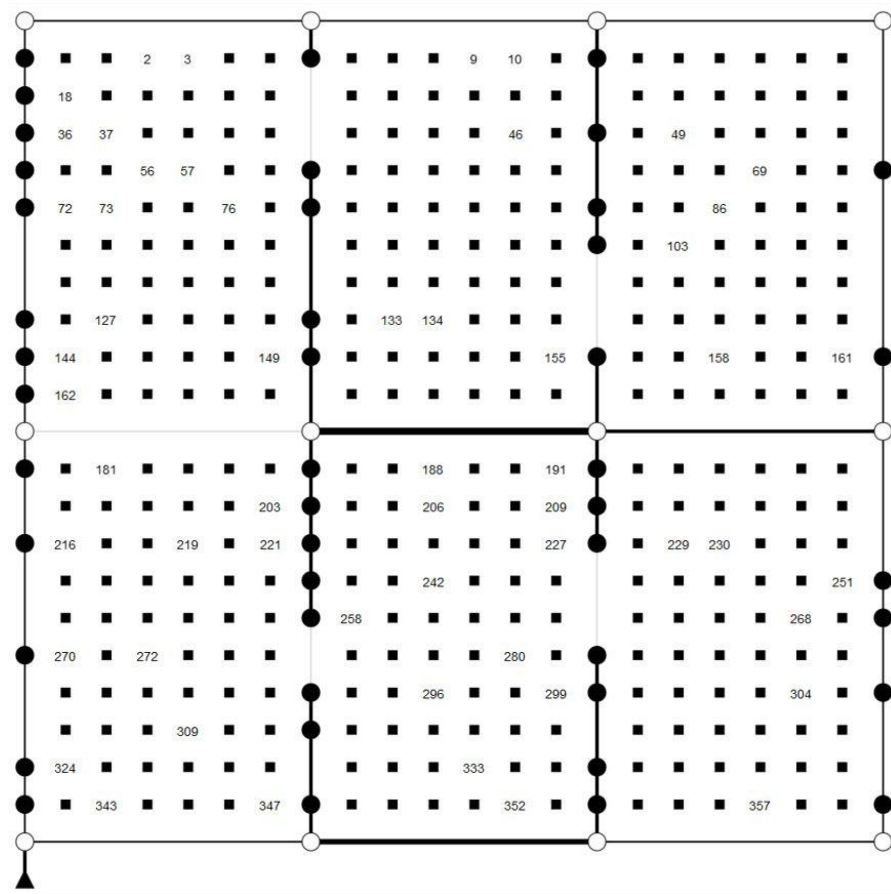
HEURISTICAS PARA PROBLEMAS DE ENRRUTAMIENTO

Figura 25. ruta de la heurística s-shape en grandes instancias.

Se da inicio a la ruta desde el vértice en la esquina inferior izquierda en donde se atraviesa el pasillo en su totalidad al llegar al pasillo de cruce posterior del almacén, se visitan uno a uno los subpasillos pertenecientes al tercer bloque, el cuarto subpasillo del bloque tres se omite debido a que no tiene productos por recolectar por lo tanto se regresa al pasillo de cruce de donde se partió, al cambiar de bloque se inicia con el subpasillo más cercano al último del tercer bloque y se explora en dirección descendente, al terminar de recorrer todos los subpasillos con producto el recolector regresa al punto de inicio, tener en cuenta que hay subpasillos que no se recorren puesto que no se tienen productos por recolectar de igual forma en el caso de los últimos bloques no es necesario recorrerlo completamente, se recoge el último producto que se tiene por recolectar y se regresa al pasillo de cruce donde se partió.

Dicho lo anterior se valida la ruta de la heurística s-shape realizada en el programa Python.

HEURISTICAS PARA PROBLEMAS DE ENRRUTAMIENTO

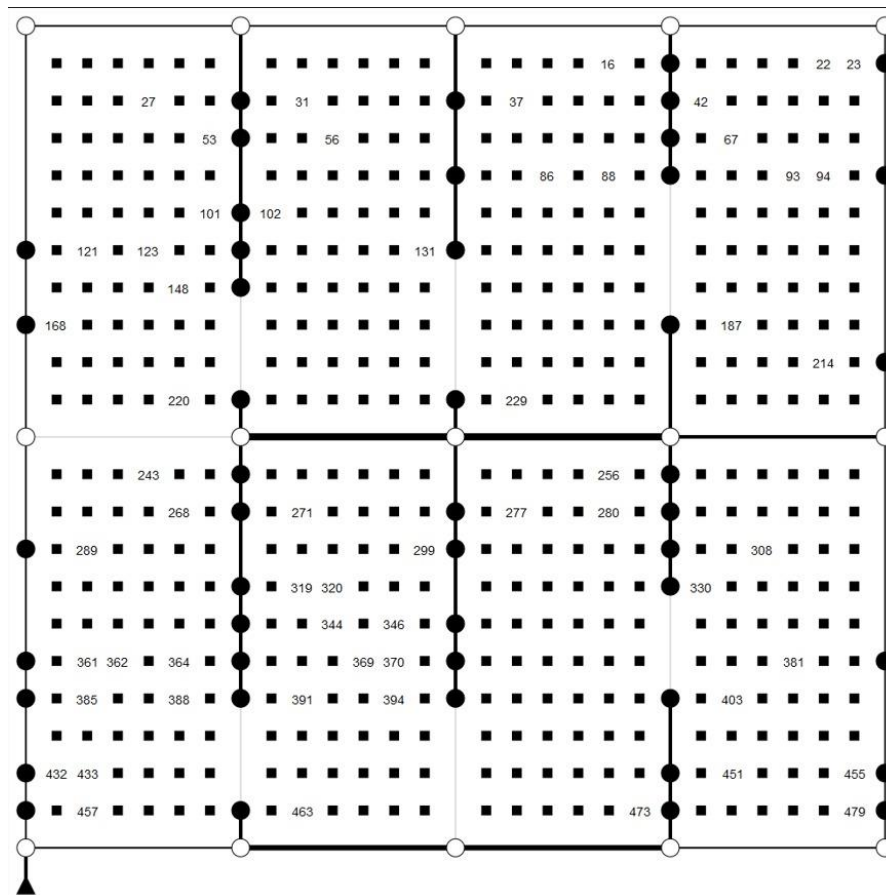
7.1.2 *Graficas arrojadas de la heurística la brecha más grande***Figura 26.** ruta de la heurística brecha más grande en pequeñas instancias.

Se da inicio a la ruta desde el vértice en la esquina inferior izquierda en donde se atraviesa el pasillo en su totalidad hasta llegar al pasillo de cruce posterior del almacén, en cada subpasillo se calcula cual es la brecha más larga para no recorrerla y la brecha puede ser entre el pasillo de cruce superior y el primer producto a recoger, entre productos adyacentes o del último producto del subpasillo al pasillo de cruce inferior, los productos que se encuentren por encima de la brecha más larga serán tomados por el pasillo de cruce superior y el recolector deberá regresar al pasillo de cruce por donde ingresó y hace lo mismo hasta explorar todos los subpasillos del bloque, para luego tomar todos los productos restantes desde el pasillo de cruce inferior, cabe aclarar que los únicos subpasillos que se atraviesan completamente son los

HEURISTICAS PARA PROBLEMAS DE ENRRUTAMIENTO

pertenecientes al primer pasillo que se encuentra al extremo y el ultimo de cada, al terminar de recoger todos los productos regresa al depósito.

Figura 27. ruta de la heurística brecha más grande en medianas instancias.

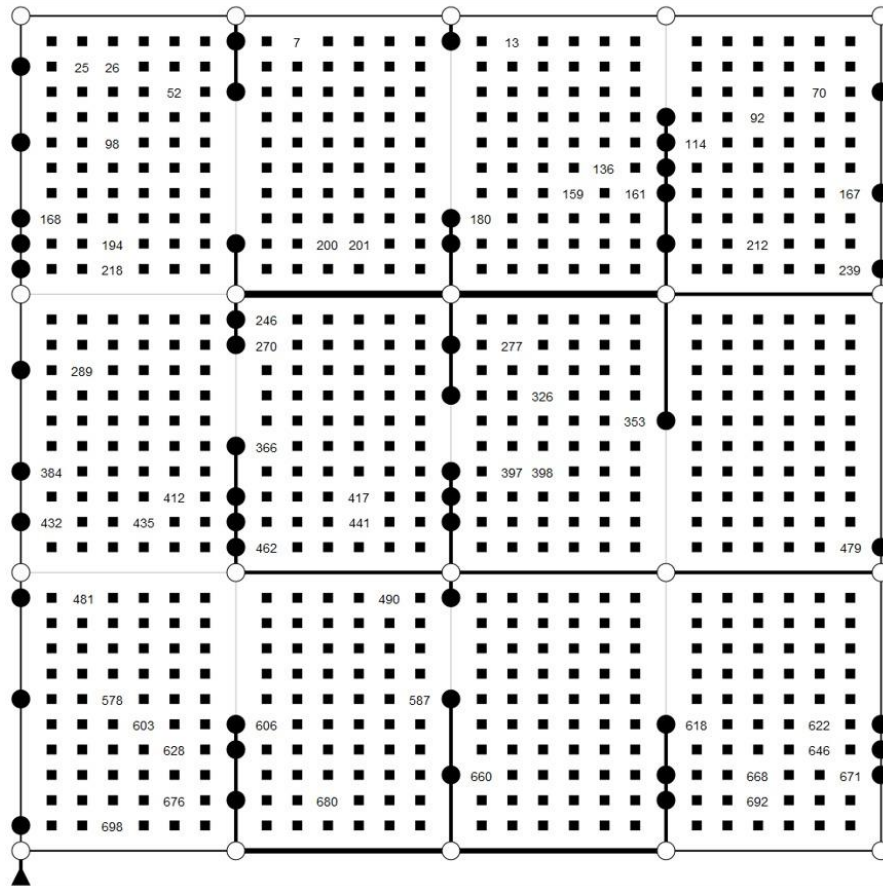


Se da inicio a la ruta desde el vértice en la esquina inferior izquierda en donde se atraviesa el pasillo en su totalidad hasta llegar al pasillo de cruce posterior del almacén, en cada subpasillo se calcula cual es la brecha más larga para no recorrerla y la brecha puede ser entre el pasillo de cruce superior y el primer producto a recoger, entre productos adyacentes o del último producto del subpasillo al pasillo de cruce inferior, los productos que se encuentren por encima de la brecha más larga serán tomados por el pasillo de cruce superior y el recolector deberá regresar al pasillo de cruce por donde ingresó y hace lo mismo hasta explorar todos los subpasillos del bloque, para luego tomar todos los productos restantes desde el pasillo de cruce inferior, cabe aclarar que los únicos subpasillos que se atraviesan completamente son los

HEURISTICAS PARA PROBLEMAS DE ENRRUTAMIENTO

pertenecientes al primer pasillo que se encuentra al extremo y el ultimo de cada, al terminar de recoger todos los productos regresa al depósito.

Figura 28. ruta de la heurística brecha más grande en grandes instancias.



Se da inicio a la ruta desde el vértice en la esquina inferior izquierda en donde se atraviesa el pasillo en su totalidad hasta llegar al pasillo de cruce posterior del almacén, en cada subpasillo se calcula cual es la brecha más larga para no recorrerla y la brecha puede ser entre el pasillo de cruce superior y el primer producto a recoger, entre productos adyacentes o del último producto del subpasillo al pasillo de cruce inferior, los productos que se encuentren por encima de la brecha más larga serán tomados por el pasillo de cruce superior y el recolector deberá regresar al pasillo de cruce por donde ingresó y hace lo mismo hasta explorar todos los subpasillos del bloque, para luego tomar todos los productos restantes desde el pasillo de cruce inferior, cabe aclarar que los únicos subpasillos que se atraviesan completamente son los

HEURISTICAS PARA PROBLEMAS DE ENRRUTAMIENTO

pertenecientes al primer pasillo que se encuentra al extremo y el ultimo de cada, al terminar de recoger todos los productos regresa al depósito.

7.2 Resultados diseño experimental

El diseño experimental es 2^2 , los factores X (tipo de ordenamiento) y Y (heurísticas). Los niveles de Y son S-shape y la brecha más grande, para X es ordenamiento ABC o NO ABC teniendo en cuenta en cada corrida que tipo de ordenamiento se identifica si realmente se está cumpliendo con el ordenamiento ABC o no, se analizará que factor (X, Y o XY) influye mayormente en la distancia. para el almacén pequeño con 20 órdenes, se tienen en cuenta 25 corridas para el tipo de almacenamiento ABC y 35 para el no ABC, para el almacén pequeño con 40 órdenes, se tienen en cuenta 32 corridas para el tipo de almacenamiento ABC y 28 para el no ABC, para el almacén pequeño con 60 órdenes, se tienen en cuenta 40 corridas para el tipo de almacenamiento ABC y 20 para el no ABC, para el almacén mediano con 20 órdenes, se tienen en cuenta 23 corridas para el tipo de almacenamiento ABC y 37 para el no ABC, para el almacén pequeño con 40 órdenes, se tienen en cuenta 26 corridas para el tipo de almacenamiento ABC y 34 para el no ABC, para el almacén pequeño con 60 órdenes, se tienen en cuenta 32 corridas para el tipo de almacenamiento ABC y 28 para el no ABC, y para el almacén grande, con 20 órdenes, se tienen en cuenta 7 corridas para el tipo de almacenamiento ABC y 45 para el no ABC, para el almacén pequeño con 40 órdenes, se tienen en cuenta 11 corridas para el tipo de almacenamiento ABC y 49 para el no ABC, para el almacén pequeño con 60 órdenes, se tienen en cuenta 7 corridas para el tipo de almacenamiento ABC y 53 para el no ABC.

Tabla 7. Análisis de variancia para instancias pequeñas.

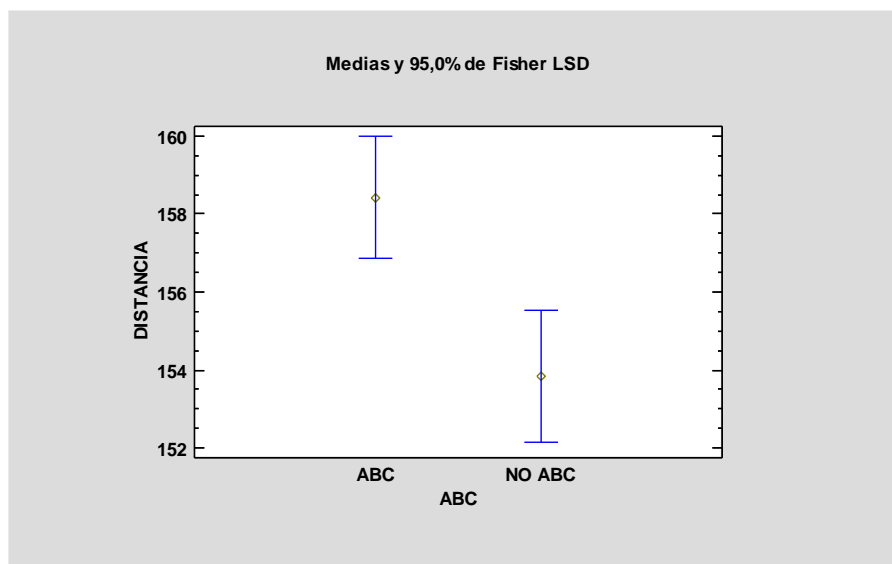
Fuente	Suma de Cuadrados	Gl	Cuadrado Medio	Razón-F	Valor-P
EFECTOS PRINCIPALES					
X:ABC	1885,77	1	1885,77	7,66	0,0059
Y: HEURISTICA	229446	1	229446	932,43	0
INTERACCIONES					
XY	1325,84	1	1325,84	5,39	0,0208

HEURISTICAS PARA PROBLEMAS DE ENRRUTAMIENTO

RESIDUOS	87602,5	356	246,074
TOTAL (CORREGIDO)	324394	359	

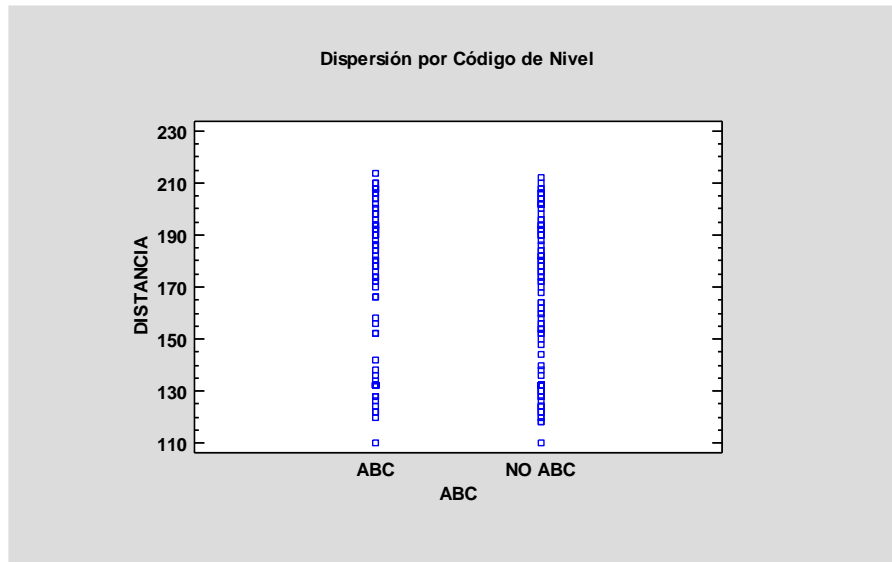
Los resultados obtenidos para el análisis de varianza del almacén pequeño permiten evidenciar que el factor (tipo de almacenamiento) ABC o NO ABC, las heurísticas y la iteración entre los dos factores tiene un efecto significativo en la distancia, ya que el valor p es menor a 0,05.

Figura 29 Diagrama cajas y bigotes de heurísticas para instancias pequeñas.

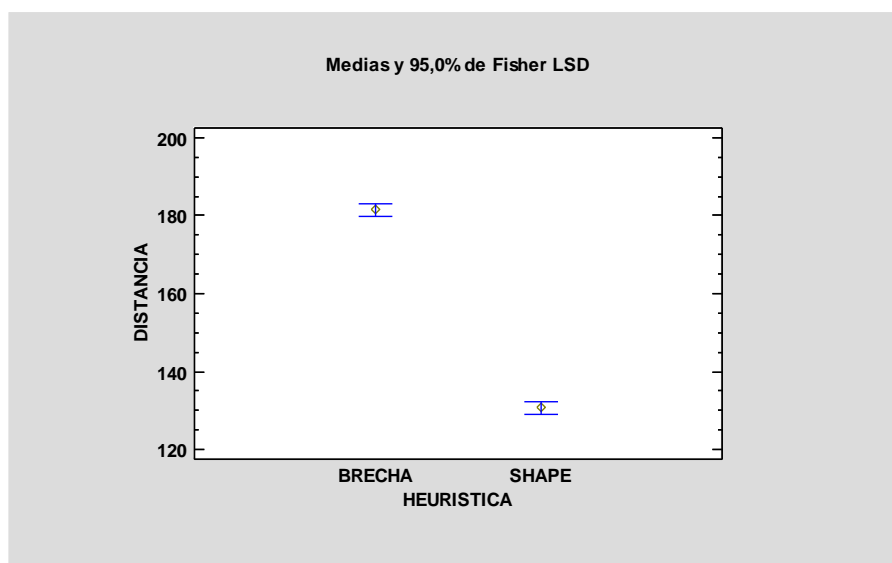


Al proyectar las líneas medias de los dos grupos (ABC y NO ABC) se evidencia que dichas proyecciones no quedan dentro de las cajas entonces se dice que ambos grupos son diferentes por lo tanto en el almacén pequeño es significativo en la distancia el tipo de almacenamiento.

HEURISTICAS PARA PROBLEMAS DE ENRRUTAMIENTO

Figura 30 Diagrama de dispersión tipo de almacenamiento para instancias pequeñas.

Según la gráfica de dispersión presentada en la figura, para instancias pequeñas se analiza que el patrón de dispersión es un poco diferente debido a que en el nivel ABC se ve mayor concentración hacia arriba por el contrario en el nivel NO ABC no se evidencia muy bien hacia donde se concentran los datos, sin embargo, el rango de dispersión de las dos muestras es muy parecido por lo tanto no existe diferencia entre los tipos de almacenamiento.

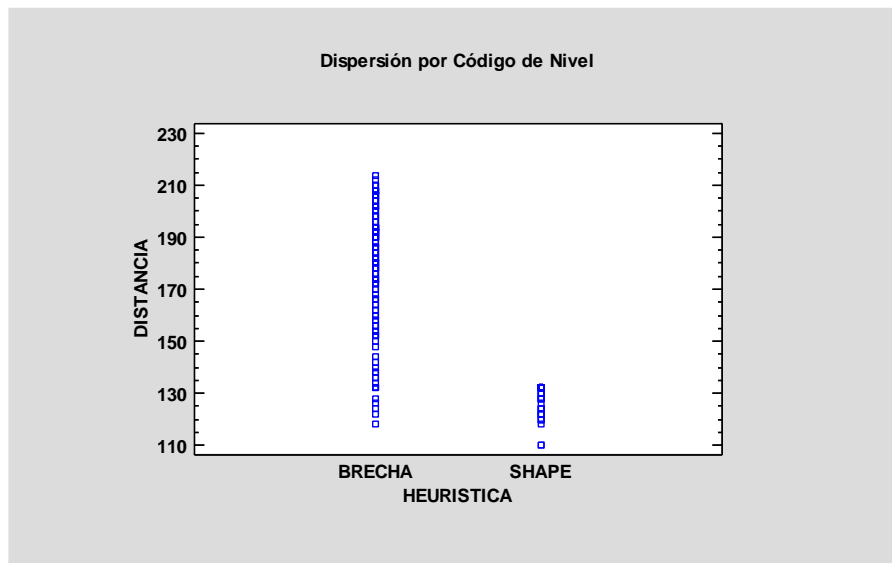
Figura 31 Diagrama cajas y bigotes de heurísticas para instancias pequeñas

Al proyectar las líneas medias de los dos grupos (s-shape y la brecha más grande) se evidencia que dichas proyecciones no quedan dentro de las cajas entonces se dice que ambos

HEURISTICAS PARA PROBLEMAS DE ENRRUTAMIENTO

grupos son diferentes por lo tanto en el almacén pequeño es significativo en la distancia las heurísticas.

Figura 32 Diagrama de dispersión heurísticas para instancias pequeñas



Según la gráfica de dispersión presentada en la figura, para instancias pequeñas se analiza que el patrón de dispersión es diferente debido a que el rango de la heurística s-shape es más pequeña respecto al rango de la heurística la brecha más grande, por ende, los resultados de la heurística la brecha más grande tiene mayor dispersión que la heurística s-shape.

Tabla 8. Análisis de variancia para instancias medianas.

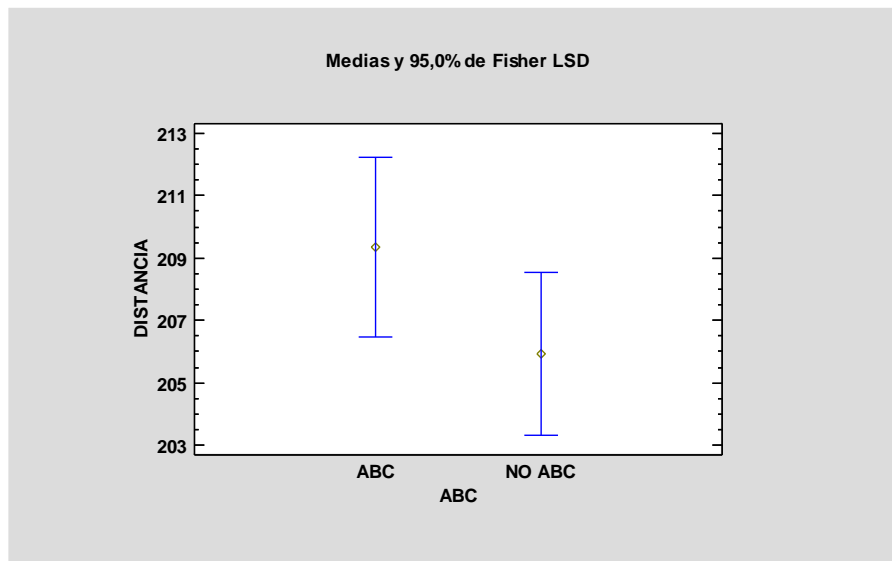
Fuente	Suma de Cuadrados	Gl	Cuadrado Medio	Razón-F	Valor-P
EFECTOS PRINCIPALES					
X:ABC	1033,81	1	1033,81	1,49	0,2231
Y: HEURISTICA	470241	1	470241	677,57	0
INTERACCIONES					
XY	596,299	1	596,299	0,86	0,3546
RESIDUOS	247068	356	694,011		
TOTAL (CORREGIDO)	720312	359			

Los resultados obtenidos para el análisis de variancia del almacén mediano permiten evidenciar que el factor (tipo de almacenamiento) ABC o NO ABC y la iteración entre los dos factores no tiene un efecto significativo en la distancia, ya que el valor p es mayor a 0,05, por

HEURISTICAS PARA PROBLEMAS DE ENRRUTAMIENTO

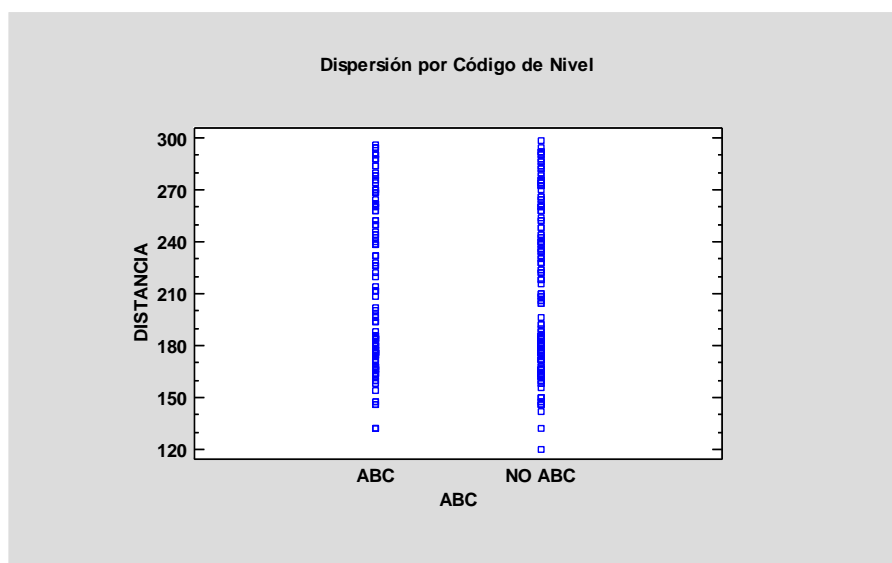
el contrario, el factor de las heurísticas si tiene un efecto significativo en las distancias debido a que el valor p es menor a 0,05.

Figura 33 Diagrama cajas y bigotes de tipo de almacenamiento para instancias medianas.



Al proyectar las líneas medias de los dos grupos (ABC y NO ABC) se evidencia que dichas proyecciones quedan dentro de las cajas entonces se dice que ambos grupos son iguales por lo tanto en el almacén mediano es significativo en la distancia el tipo de almacenamiento.

Figura 34 Diagrama de dispersión tipo de almacenamiento para instancias medianas

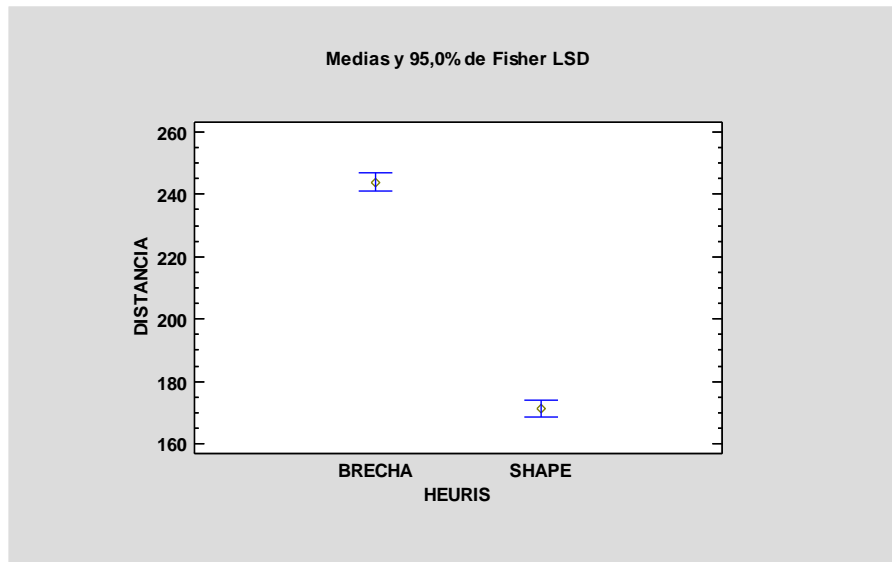


Según la gráfica de dispersión presentada en la figura, para instancias medianas se analiza que el nivel ABC presenta un menor rango de dispersión comparado con el nivel NO

HEURISTICAS PARA PROBLEMAS DE ENRRUTAMIENTO

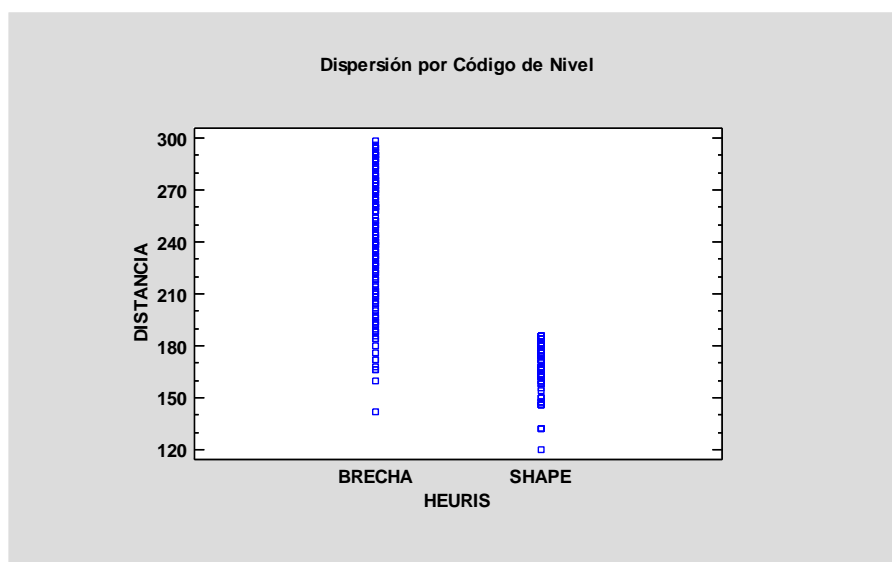
ABC, sin embargo, el rango de dispersión de las dos muestras es parecido por lo tanto no existe diferencia entre los tipos de almacenamiento.

Figura 35 Diagrama cajas y bigotes de heurísticas para instancias medianas.



Al proyectar las líneas medias de los dos grupos (s-shape y la brecha más grande) se evidencia que dichas proyecciones no quedan dentro de las cajas entonces se dice que ambos grupos son diferentes por lo tanto en el almacén mediano es significativo en la distancia las heurísticas.

Figura 36 Diagrama de dispersión heurísticas para instancias medianas



HEURÍSTICAS PARA PROBLEMAS DE ENRRUTAMIENTO

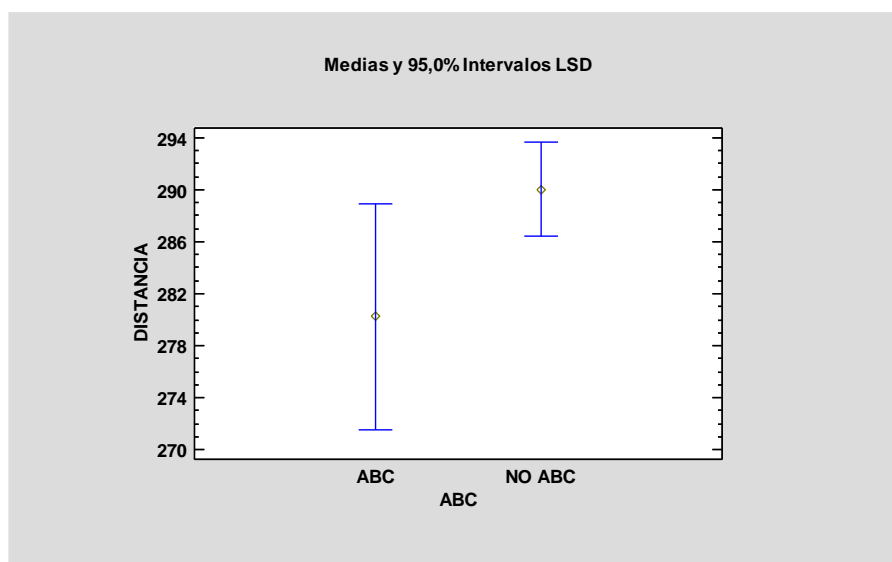
Según la gráfica de dispersión presentada en la figura, para instancias medianas se analiza que el patrón de dispersión es diferente debido a que el rango de la heurística s-shape es más pequeña respecto al rango de la heurística la brecha más grande, por ende, los resultados de la heurística la brecha más grande tiene mayor dispersión que la heurística s-shape.

Tabla 9. Análisis de variancia para instancias grandes.

Fuente	Suma de Cuadrados	Gl	Cuadrado Medio	Razón-F	Valor-P
EFECTOS PRINCIPALES					
X:ABC	4081,96	1	4081,96	2,07	0,1512
Y: HEURISTICA	213131	1	213131	108,03	0
INTERACCIONES					
XY	123,122	1	123,122	0,06	0,8029
RESIDUOS	670757	340	1972,81		
TOTAL (CORREGIDO)	1089390	343			

Los resultados obtenidos para el análisis de varianza del almacén grande permiten evidenciar que el factor (tipo de almacenamiento) ABC o NO ABC y la iteración entre los dos factores no tiene un efecto significativo en la distancia, ya que el valor p es mayor a 0,05, por el contrario, el factor de las heurísticas si tiene un efecto significativo en las distancias debido a que el valor p es menor a 0,05.

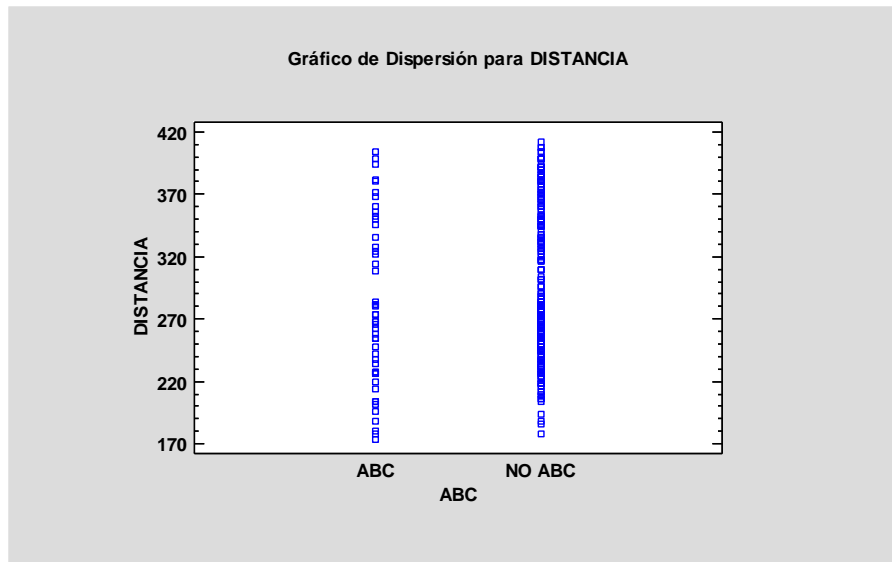
Figura 37 Diagrama cajas y bigotes de tipo de almacenamiento para instancias grandes.



HEURISTICAS PARA PROBLEMAS DE ENRRUTAMIENTO

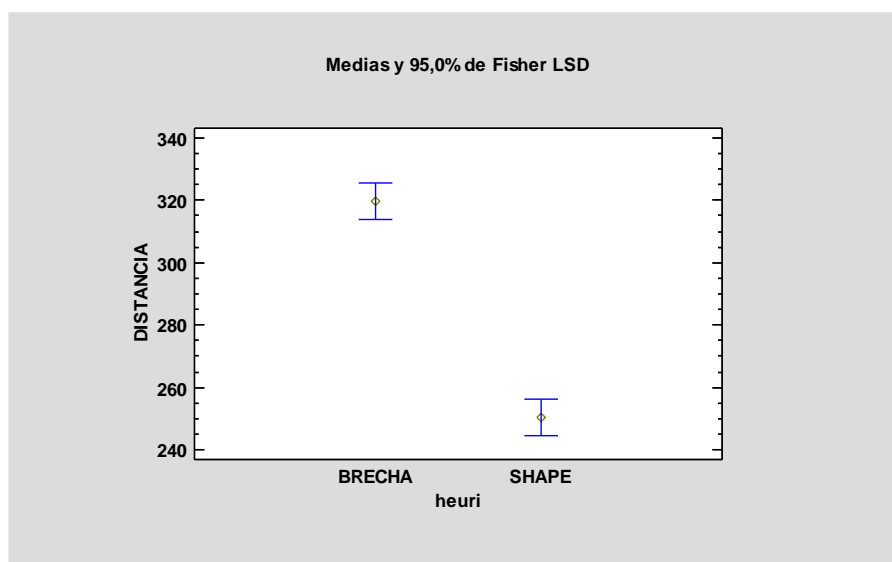
Al proyectar las líneas medias de los dos grupos (ABC y NO ABC) se evidencia que dichas proyecciones quedan dentro de las cajas entonces se dice que ambos grupos son iguales por lo tanto en el almacén grande es significativo en la distancia el tipo de almacenamiento.

Figura 38 Diagrama de dispersión tipo de almacenamiento para instancias grandes



Según la gráfica de dispersión presentada en la figura, para instancias grandes se analiza que el nivel NO ABC tiene mayor densidad debido a que hay más muestras de dicho grupo, sin embargo, el rango de dispersión de las dos muestras es muy parecido por lo tanto no existe diferencia entre los tipos de almacenamiento.

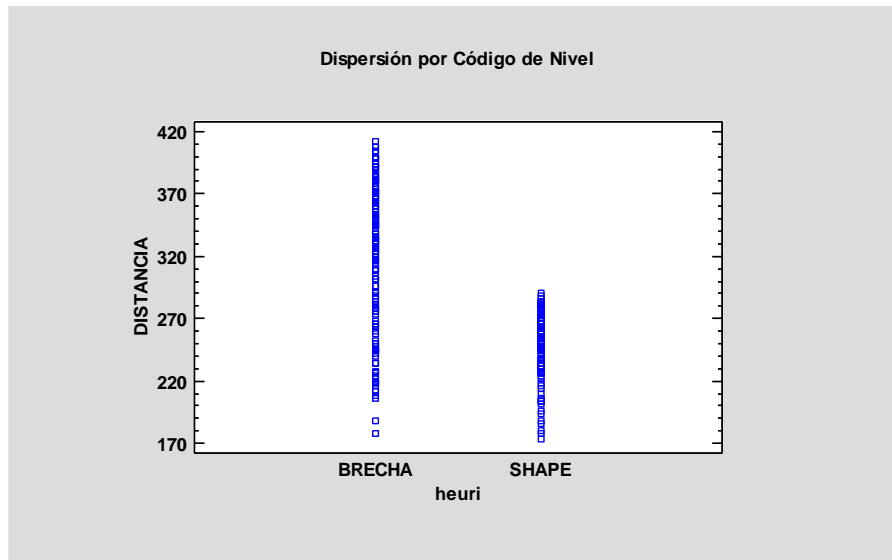
Figura 39 Diagrama de dispersión heurísticas para instancias grandes



HEURISTICAS PARA PROBLEMAS DE ENRRUTAMIENTO

Al proyectar las líneas medias de los dos grupos (s-shape y la brecha más grande) se evidencia que dichas proyecciones no quedan dentro de las cajas entonces se dice que ambos grupos son diferentes por lo tanto en el almacén grande es significativo en la distancia las heurísticas.

Figura 40 Diagrama de dispersión heurísticas para instancias grandes



Según la gráfica de dispersión presentada en la figura, para instancias grandes se analiza que el patrón de dispersión es diferente debido a que el rango de la heurística s-shape es más pequeña respecto al rango de la heurística la brecha más grande, por ende, los resultados de la heurística la brecha más grande tiene mayor dispersión que la heurística s-shape.

8 Resultados

Para la obtención de los resultados se correrá el programa en un computador con las siguientes especificaciones, Intel - Core i7, sistema operativo de 64 bits, procesador x64, memoria RAM de 8GB. Se tuvieron en cuenta tres tamaños de almacenes para obtener los resultados; almacén grande, mediano y pequeño con un único deposito, hay que tener en cuenta que el almacén grande cuenta con los parámetros de entrada como lo son 3 bloques con 2 pasillos horizontales, 3 pasillos verticales y cantidad SKU 720, para el almacén mediano se cuenta con parámetros de entrada como lo son 2 bloques con 1 pasillo horizontal,

HEURISTICAS PARA PROBLEMAS DE ENRRUTAMIENTO

3 pasillos verticales y cantidad SKU de 480, para el almacén pequeño se cuenta con parámetros de entrada como lo son 2 bloques con 1 pasillo horizontal, 2 pasillos verticales y cantidad SKU de 360.

De este modo se explica que los parámetros de entrada que se varían para cada almacén es el número de órdenes y tipo de almacenamiento, se realizarán 3 corridas, con 60 iteraciones y se tomará el resultado de la distancia de cada heurística, contando el número de órdenes a recoger en cada segmento (A, B o C) para realizar el respectivo análisis del tipo de almacenamiento ABC, en cada almacén se tendrían en cuenta tres tamaños de pedidos, grande con 60 órdenes, mediano con 40 órdenes y pequeño con 20 órdenes.

8.1 Resultados tipo de almacenamiento y comparación entre heurísticas s-shape y la brecha más grande

Se presentan los datos obtenidos para cada tamaño de almacén con los algoritmos de las heurísticas s-shape y la brecha más grande que fue modelado en el software Python, los resultados se ilustran a continuación

8.1.1 Pequeñas instancias

Tabla 10. Resultados tipo de almacenamiento para un almacén pequeño con 20 órdenes.

20 ORDENES	S-Shape		Brecha	
	ABC	NO ABC	ABC	NO ABC
Distancia 1	132	132	190	164
Distancia 2	122	132	126	172
Distancia 3	132	132	158	184
Distancia 4	132	128	172	132
Distancia 5	132	132	166	182
Distancia 6	132	132	190	152
Distancia 7	122	120	134	160
Distancia 8	128	132	132	162
Distancia 9	132	110	152	118
Distancia 10	122	132	138	180
Distancia 11	132	130	194	136
Distancia 12	120	132	128	176
Distancia 13	132	124	188	152
Distancia 14	124	126	152	154

HEURISTICAS PARA PROBLEMAS DE ENRRUTAMIENTO

Distancia 15	132	124	180	122
Distancia 16	132	132	166	170
Distancia 17	132	120	184	138
Distancia 18	132	132	178	158
Distancia 19	132	122	178	148
Distancia 20	110	132	142	156
Distancia 21	132	132	170	162
Distancia 22	128	132	136	184
Distancia 23	132	130	186	154
Distancia 24	132	132	156	174
Distancia 25	132	132	174	164
Distancia 26		128		132
Distancia 27		118		124
Distancia 28		132		174
Distancia 29		132		172
Distancia 30		130		132
Distancia 31		132		156
Distancia 32		132		182
Distancia 33		132		154
Distancia 34		132		150
Distancia 35		122		160

Tabla 11. Resultados tipo de almacenamiento para un almacén pequeño con 40 órdenes.

40 ORDENES	S-Shape		Brecha	
	ABC	NO ABC	ABC	NO ABC
Distancia 1	132	132	186	204
Distancia 2	132	124	194	140
Distancia 3	132	132	174	182
Distancia 4	132	132	208	188
Distancia 5	132	132	202	190
Distancia 6	132	132	192	202
Distancia 7	132	128	184	144
Distancia 8	132	132	192	180
Distancia 9	132	132	200	178
Distancia 10	132	132	174	190
Distancia 11	132	132	208	190
Distancia 12	132	132	186	192
Distancia 13	132	132	192	178
Distancia 14	132	132	172	198
Distancia 15	132	132	206	202
Distancia 16	132	132	192	180
Distancia 17	132	132	190	200
Distancia 18	132	132	194	208

HEURISTICAS PARA PROBLEMAS DE ENRRUTAMIENTO

Distancia 19	132	132	204	190
Distancia 20	132	132	200	202
Distancia 21	132	132	178	196
Distancia 22	132	132	198	204
Distancia 23	132	132	194	194
Distancia 24	132	132	180	202
Distancia 25	132	130	176	168
Distancia 26	132	132	186	176
Distancia 27	132	132	204	186
Distancia 28	132	132	174	202
Distancia 29	132		192	
Distancia 30	132		210	
Distancia 31	132		180	
Distancia 32	132		166	

Tabla 12. Resultados tipo de almacenamiento para un almacén pequeño con 60 órdenes.

60 ORDENES	S-Shape		Brecha	
	ABC	NO ABC	ABC	NO ABC
Distancia 1	132	132	190	192
Distancia 2	132	132	194	206
Distancia 3	132	132	180	206
Distancia 4	132	132	192	206
Distancia 5	132	132	184	204
Distancia 6	132	132	196	196
Distancia 7	132	132	180	178
Distancia 8	132	132	192	182
Distancia 9	132	132	210	176
Distancia 10	132	132	208	212
Distancia 11	132	132	204	194
Distancia 12	132	132	198	194
Distancia 13	132	132	208	192
Distancia 14	132	132	192	202
Distancia 15	132	132	208	192
Distancia 16	132	132	178	194
Distancia 17	132	132	206	204
Distancia 18	132	132	178	206
Distancia 19	132	132	192	208
Distancia 20	132	132	206	210
Distancia 21	132		198	
Distancia 22	132		190	
Distancia 23	132		208	
Distancia 24	132		210	
Distancia 25	132		196	

HEURISTICAS PARA PROBLEMAS DE ENRRUTAMIENTO

Distancia 26	132		192	
Distancia 27	132		194	
Distancia 28	132		214	
Distancia 29	132		182	
Distancia 30	132		206	
Distancia 31	132		194	
Distancia 32	132		194	
Distancia 33	132		194	
Distancia 34	132		208	
Distancia 35	132		208	
Distancia 36	132		200	
Distancia 37	132		190	
Distancia 38	132		190	
Distancia 39	132		198	
Distancia 40	132		190	

8.1.2 Medianas instancias

Tabla 13. Resultados tipo de almacenamiento para un almacén mediano con 20 órdenes.

20 ORDENES	S-Shape		Brecha	
	ABC	NO ABC	ABC	NO ABC
Distancia 1	178	166	232	208
Distancia 2	148	176	212	224
Distancia 3	132	150	160	192
Distancia 4	158	160	196	184
Distancia 5	132	164	166	218
Distancia 6	164	164	232	210
Distancia 7	164	166	202	210
Distancia 8	146	146	188	184
Distancia 9	162	150	194	196
Distancia 10	166	132	180	172
Distancia 11	176	166	226	176
Distancia 12	178	172	270	190
Distancia 13	164	160	198	188
Distancia 14	174	146	212	218
Distancia 15	168	166	222	224
Distancia 16	148	162	208	172
Distancia 17	164	162	212	206
Distancia 18	146	174	194	238
Distancia 19	168	162	212	204
Distancia 20	172	178	220	232
Distancia 21	164	162	194	208
Distancia 22	132	172	166	248

HEURISTICAS PARA PROBLEMAS DE ENRRUTAMIENTO

Distancia 23	154	146	200	204
Distancia 24		146		168
Distancia 25		156		186
Distancia 26		168		240
Distancia 27		184		260
Distancia 28		158		192
Distancia 29		182		228
Distancia 30		162		188
Distancia 31		148		206
Distancia 32		178		228
Distancia 33		164		218
Distancia 34		186		234
Distancia 35		120		142
Distancia 36		164		238
Distancia 37		172		216

Tabla 14. Resultados tipo de almacenamiento para un almacén mediano con 40 órdenes.

40 ORDENES	S-Shape		Brecha	
	ABC	NO ABC	ABC	NO ABC
Distancia 1	168	178	262	240
Distancia 2	182	172	262	264
Distancia 3	176	164	242	222
Distancia 4	184	182	250	244
Distancia 5	166	174	238	260
Distancia 6	176	176	252	264
Distancia 7	184	184	238	252
Distancia 8	164	180	226	272
Distancia 9	180	148	260	208
Distancia 10	178	164	268	244
Distancia 11	168	180	280	274
Distancia 12	170	166	240	242
Distancia 13	176	178	238	248
Distancia 14	180	162	244	230
Distancia 15	166	182	252	284
Distancia 16	174	176	250	240
Distancia 17	176	166	260	236
Distancia 18	166	166	262	228
Distancia 19	184	176	270	278
Distancia 20	166	150	244	234
Distancia 21	176	176	274	242
Distancia 22	178	162	276	222
Distancia 23	168	178	250	276
Distancia 24	178	178	246	260

HEURISTICAS PARA PROBLEMAS DE ENRRUTAMIENTO

Distancia 25	186	178	238	270
Distancia 26	160	176	226	262
Distancia 27		164		230
Distancia 28		178		286
Distancia 29		184		240
Distancia 30		180		282
Distancia 31		170		266
Distancia 32		180		260
Distancia 33		186		272
Distancia 34		184		234

Tabla 15. Resultados tipo de almacenamiento para un almacén mediano con 60 órdenes.

60 ORDENES	S-Shape		Brecha	
	ABC	NO ABC	ABC	NO ABC
Distancia 1	184	180	268	266
Distancia 2	178	178	278	292
Distancia 3	174	180	258	282
Distancia 4	174	182	294	274
Distancia 5	176	186	260	292
Distancia 6	180	182	290	290
Distancia 7	176	180	276	276
Distancia 8	176	180	228	258
Distancia 9	180	178	232	272
Distancia 10	186	184	288	280
Distancia 11	182	176	262	272
Distancia 12	176	178	288	258
Distancia 13	182	182	288	298
Distancia 14	184	182	290	260
Distancia 15	176	186	270	288
Distancia 16	184	184	264	292
Distancia 17	184	186	284	276
Distancia 18	174	182	260	272
Distancia 19	178	184	296	290
Distancia 20	180	166	294	236
Distancia 21	176	158	258	222
Distancia 22	168	178	270	286
Distancia 23	166	184	214	276
Distancia 24	176	176	268	254
Distancia 25	172	176	278	258
Distancia 26	176	184	268	274
Distancia 27	186	180	290	294
Distancia 28	184	180	284	252
Distancia 29	184		294	

HEURISTICAS PARA PROBLEMAS DE ENRRUTAMIENTO

Distancia 30	180		288	
Distancia 31	186		290	
Distancia 32	184		290	

8.1.3 *Grandes instancias***Tabla 16.** Resultados tipo de almacenamiento para un almacén grande con 20 órdenes.

20 ORDENES	S-Shape		Brecha	
	ABC	NO ABC	ABC	NO ABC
Distancia 1	214	252	242	286
Distancia 2	180	234	178	212
Distancia 3	202	236	226	244
Distancia 4	204	206	228	248
Distancia 5	196	254	188	276
Distancia 6	228	216	220	258
Distancia 7	204	258	234	310
Distancia 8		232		250
Distancia 9		258		280
Distancia 10		246		264
Distancia 11		210		240
Distancia 12		248		218
Distancia 13		246		244
Distancia 14		204		228
Distancia 15		250		276
Distancia 16		242		302
Distancia 17		238		246
Distancia 18		238		246
Distancia 19		232		264
Distancia 20		252		274
Distancia 21		226		234
Distancia 22		188		206
Distancia 23		254		278
Distancia 24		222		218
Distancia 25		252		244
Distancia 26		194		208
Distancia 27		230		222
Distancia 28		256		244
Distancia 29		218		252
Distancia 30		226		218
Distancia 31		232		246
Distancia 32		226		250
Distancia 33		224		266
Distancia 34		232		268
Distancia 35		232		302
Distancia 36		250		260
Distancia 37		244		296
Distancia 38		246		304

HEURISTICAS PARA PROBLEMAS DE ENRRUTAMIENTO

Distancia 39		248		290
Distancia 40		178		216
Distancia 41		234		224
Distancia 42		226		292
Distancia 43		264		266
Distancia 44		186		212
Distancia 45		228		284

Tabla 17. Resultados tipo de almacenamiento para un almacén grande con 40 órdenes.

40 ORDENES	S-Shape		Brecha	
	ABC	NO ABC	ABC	NO ABC
Distancia 1	254	250	314	318
Distancia 2	274	274	336	344
Distancia 3	238	252	324	332
Distancia 4	268	256	322	316
Distancia 5	268	240	356	302
Distancia 6	262	258	372	348
Distancia 7	174	274	360	344
Distancia 8	254	264	368	368
Distancia 9	248	244	328	346
Distancia 10	258	266	346	380
Distancia 11	266	246	350	310
Distancia 12		270		374
Distancia 13		278		362
Distancia 14		254		326
Distancia 15		260		366
Distancia 16		252		278
Distancia 17		272		364
Distancia 18		230		278
Distancia 19		268		320
Distancia 20		262		326
Distancia 21		230		288
Distancia 22		252		310
Distancia 23		250		332
Distancia 24		254		332
Distancia 25		238		316
Distancia 26		256		330
Distancia 27		264		338
Distancia 28		280		348
Distancia 29		242		344
Distancia 30		268		328
Distancia 31		236		282
Distancia 32		264		344
Distancia 33		252		350
Distancia 34		284		352
Distancia 35		274		330
Distancia 36		270		336

HEURISTICAS PARA PROBLEMAS DE ENRRUTAMIENTO

Distancia 37		258		342
Distancia 38		238		296
Distancia 39		268		332
Distancia 40		260		334
Distancia 41		244		318
Distancia 42		276		336
Distancia 43		278		360
Distancia 44		274		380
Distancia 45		268		336
Distancia 46		276		350
Distancia 47		276		318
Distancia 48		256		296
Distancia 49		276		350

Tabla 18 Resultados tipo de almacenamiento para un almacén grande con 60 órdenes.

60 ORDENES	S-Shape		Brecha	
	ABC	NO ABC	ABC	NO ABC
Distancia 1	282	272	380	354
Distancia 2	280	252	394	322
Distancia 3	282	268	404	338
Distancia 4	284	284	352	408
Distancia 5	274	264	382	370
Distancia 6	226	284	308	390
Distancia 7	282	282	398	382
Distancia 8		286		392
Distancia 9		290		392
Distancia 10		272		372
Distancia 11		278		380
Distancia 12		274		394
Distancia 13		270		404
Distancia 14		278		392
Distancia 15		280		386
Distancia 16		254		324
Distancia 17		264		370
Distancia 18		274		412
Distancia 19		282		358
Distancia 20		268		366
Distancia 21		258		376
Distancia 22		268		378
Distancia 23		270		398
Distancia 24		282		398
Distancia 25		284		344
Distancia 26		250		356
Distancia 27		256		360
Distancia 28		270		392
Distancia 29		278		404
Distancia 30		278		400

HEURISTICAS PARA PROBLEMAS DE ENRRUTAMIENTO

Distancia 31		260		346
Distancia 32		246		366
Distancia 33		282		384
Distancia 34		272		376
Distancia 35		268		374
Distancia 36		270		352
Distancia 37		274		408
Distancia 38		288		392
Distancia 39		262		350
Distancia 40		282		380
Distancia 41		278		360
Distancia 42		274		398
Distancia 43		268		380
Distancia 44		272		386
Distancia 45		282		390
Distancia 46		280		368
Distancia 47		274		398
Distancia 48		252		386
Distancia 49		274		384
Distancia 50		264		372
Distancia 51		268		384
Distancia 52		246		324
Distancia 53		260		352

Las heurísticas son un método que no consiguen resultados óptimos, pero permiten obtener soluciones factibles con un tiempo de cálculo razonable y esto hace más sencilla la solución de problemas difíciles, es por esto que en la presente investigación se quiere comparar las distancias entre las dos heurísticas desarrolladas en el software Python, en las tablas mostradas anteriormente se evidencia la diferencia entre las heurísticas y se puede observar que la distancia obtenida con la heurística s-shape respecto con la heurística la brecha más grande es menor para los tres tamaños tanto de almacenes como de órdenes.

En esta investigación se tienen instancias pequeñas medianas y grandes, para las instancias pequeñas se observa que la distancia de la heurística s-shape para el tamaño de ordenes pequeñas el 65% arroja la misma distancia y la distancia del 35% restante no supera al valor más repetido, para el tamaño de ordenes medianas el 68% arroja la misma distancia y la distancia del 32% restante no supera el valor más repetido, para el tamaño de ordenes grande

HEURISTICAS PARA PROBLEMAS DE ENRRUTAMIENTO

el 100% arroja la misma distancia por lo tanto distancia tiende a ser la misma entre más grande sea el número de ordenes por recolectar debido a que al ser un almacén pequeño con muchas ordenes se va a tener que recorrer todo el almacén realizando siempre la ruta en forma de S, cabe resaltar que las distancias en la heurística la brecha más grande no varía mucho al aumentar el tamaño de órdenes a recolectar; para las instancias medianas con tamaño de ordenes pequeñas la diferencia entre las dos distancias en ninguna iteración es mayor a 100, para el tamaño de ordenes mediano la diferencia entre las dos distancias en un 8% es mayor a 100 y para el tamaño de ordenes grande la diferencia entre las dos distancias en un 48% es mayor a 100 se observa que la diferencia entre las distancias de las heurísticas va aumentando respectivamente al número de órdenes cabe resaltar que las distancias en la heurística s-shape no varían mucho a diferencia de las distancias de la heurística la brecha más grande; para las instancias grandes con tamaño de ordenes pequeñas el 21% de las distancias son menores en la heurística la brecha más grande, para el tamaño de órdenes mediano la diferencia entre las dos distancias en un 16% es mayor a 100, para el tamaño de ordenes grande la diferencia entre las dos distancias en un 66% es mayor a 100 se observa que la diferencia entre las distancias de las heurísticas va aumentando respectivamente al número de órdenes, cabe resaltar que en este tamaño de almacén fue el único en donde se presentaron mejores resultados en la heurística la brecha más grande para tamaños de ordenes pequeñas de igual forma las distancias en la heurística s-shape no varían mucho a diferencia de las distancias de la heurística la brecha más grande.

8.1.4 Análisis de Varianza para Pequeñas Instancias

A continuación, se presenta el análisis de varianza para la distancia en las instancias que se propusieron

Tabla 19. *Consolidado de análisis de varianza tipo de almacenamiento ABC/NO ABC para instancias pequeñas.*

HEURISTICAS PARA PROBLEMAS DE ENRRUTAMIENTO

Tamaño de almacén	Tamaño de muestra (n)	Heurística	F	Probabilidad	Valor crítico para F
Pequeño	20	S-shape	0,16	0,69	4,01
Pequeño	20	Brecha más grande	0,01	0,91	4,01
Pequeño	40	S-shape	2,81	0,10	4,01
Pequeño	40	Brecha más grande	0,35	0,56	4,01
Pequeño	60	S-shape	65535	#¡DIV/0!	4,01
Pequeño	60	Brecha más grande	0,26	0,61	4,01

Los resultados obtenidos para el análisis de varianza del almacén pequeño permiten evidenciar que el factor (tipo de almacenamiento) ABC o NO ABC, no es significativo en la distancia en cinco de seis resultados, ya que el valor f es menor al f crítico, por esta razón se acepta la hipótesis nula de que todos los promedios de los grupos son iguales.

Tabla 20. Consolidado de valores diagrama cajas y bigotes tipo de almacenamiento ABC/NO ABC para instancias pequeñas.

Tamaño de almacén	Tamaño de muestra (n)	Heurística	Tipo de almacenamiento	Distancia promedio
Pequeño	20	S-shape	NO ABC	129
Pequeño	20	S-shape	ABC	129
Pequeño	20	Brecha más grande	NO ABC	160
Pequeño	20	Brecha más grande	ABC	159
Pequeño	40	S-shape	NO ABC	132
Pequeño	40	S-shape	ABC	132
Pequeño	40	Brecha más grande	NO ABC	188
Pequeño	40	Brecha más grande	ABC	190
Pequeño	60	S-shape	NO ABC	132
Pequeño	60	S-shape	ABC	132
Pequeño	60	Brecha más grande	NO ABC	198
Pequeño	60	Brecha más grande	ABC	196

En los resultados obtenidos se evidencia que no existe una diferencia relevante entre las distancias promedio teniendo en cuenta el tipo de almacenamiento por lo tanto se puede decir que no es significativo en la distancia el tipo de almacenamiento.

Tabla 21. Consolidado análisis de varianza entre heurísticas para instancias pequeñas.

Tamaño de almacén	Tamaño de muestra(n)	F	Probabilidad	Valor crítico para F
Pequeño	20	159,81	0,00	3,92

HEURISTICAS PARA PROBLEMAS DE ENRRUTAMIENTO

Pequeño	40	971,5	0,00	3,92
Pequeño	60	2535,4	0,00	3,92

Los resultados obtenidos para el análisis de varianza del almacén pequeño permiten evidenciar que el factor heurística s-shape y la brecha más grande, es significativo en la distancia, ya que el valor f es mayor al f crítico, por esta razón se rechaza la hipótesis nula de que todos los promedios de los grupos son iguales.

Tabla 22. Consolidado de valores diagrama cajas y bigotes heurísticas.

Tamaño de almacén	Tamaño de muestra (n)	Heurística	Distancia promedio
Pequeño	20	S-shape	129
Pequeño	20	Brecha más grande	160
Pequeño	40	S-shape	132
Pequeño	40	Brecha más grande	189
Pequeño	60	S-shape	132
Pequeño	60	Brecha más grande	197

En los resultados obtenidos se evidencia que existe una diferencia relevante entre las distancias promedio teniendo en cuenta las heurísticas por lo tanto se puede decir que es significativo en la distancia la heurística.

8.1.5 Análisis de Varianza para medianas Instancias

Tabla 23. Consolidado de análisis de varianza tipo de almacenamiento ABC/ NO ABC para instancias medianas.

Tamaño de almacén	Tamaño de muestra(n)	Heurística	F	Probabilidad	Valor crítico para F
Mediano	20	S-shape	0,58	0,45	4,01
Mediano	20	Brecha más grande	0,15	0,7	4,01
Mediano	40	S-shape	0,12	0,73	4,01
Mediano	40	Brecha más grande	0	0,98	4,01
Mediano	60	S-shape	0,4	0,53	4,01
Mediano	60	Brecha más grande	0,03	0,86	4,01

Los resultados obtenidos para el análisis de varianza del almacén mediano permiten evidenciar que el factor (tipo de almacenamiento) ABC o NO ABC, no es significativo en la

HEURISTICAS PARA PROBLEMAS DE ENRRUTAMIENTO

distancia, ya que el valor f es menor al f crítico, por esta razón se acepta la hipótesis nula de que todos los promedios de los grupos son iguales.

Tabla 24. Consolidado de valores diagrama cajas y bigotes tipo de almacenamiento ABC/NO ABC para instancias medianas.

Tamaño de almacén	Tamaño de muestra (n)	Heurística	Tipo de almacenamiento	Distancia promedio
Mediano	20	S-shape	NO ABC	162
Mediano	20	S-shape	ABC	159
Mediano	20	Brecha más grande	NO ABC	207
Mediano	20	Brecha más grande	ABC	204
Mediano	40	S-shape	NO ABC	173
Mediano	40	S-shape	ABC	174
Mediano	40	Brecha más grande	NO ABC	252
Mediano	40	Brecha más grande	ABC	252
Mediano	60	S-shape	NO ABC	180
Mediano	60	S-shape	ABC	179
Mediano	60	Brecha más grande	NO ABC	273
Mediano	60	Brecha más grande	ABC	274

En los resultados obtenidos se evidencia que no existe una diferencia relevante entre las distancias promedio teniendo en cuenta el tipo de almacenamiento por lo tanto se puede decir que no es significativo en la distancia el tipo de almacenamiento.

Tabla 25. Consolidado análisis de varianza entre heurísticas para instancias medianas.

Tamaño de almacén	Tamaño de muestra(n)	F	Probabilidad	Valor crítico para F
Mediano	20	148,87	0,00	3,92
Mediano	40	927,16	0,00	3,92
Mediano	60	1333,8	0,00	3,92

Los resultados obtenidos para el análisis de varianza del almacén mediano permiten evidenciar que el factor heurística s-shape y la brecha más grande, es significativo en la distancia, ya que el valor f es mayor al f crítico, por esta razón se rechaza la hipótesis nula de que todos los promedios de los grupos son iguales.

Tabla 26. Consolidado de valores diagrama cajas y bigotes heurísticas.

HEURISTICAS PARA PROBLEMAS DE ENRRUTAMIENTO

Tamaño de almacén	Tamaño de muestra (n)	Heurística	Distancia promedio
Mediano	20	S-shape	161
Mediano	20	Brecha más grande	206
Mediano	40	S-shape	174
Mediano	40	Brecha más grande	252
Mediano	60	S-shape	179
Mediano	60	Brecha más grande	273

En los resultados obtenidos se evidencia que existe una diferencia relevante entre las distancias promedio teniendo en cuenta las heurísticas por lo tanto se puede decir que es significativo en la distancia la heurística.

8.1.6 Análisis de Varianza para grandes Instancias

Tabla 27. Consolidado de análisis de varianza tipo de almacenamiento ABC/NO ABC para instancias grandes.

Tamaño de almacén	Tamaño de muestra(n)	Heurística	F	Probabilidad	Valor crítico para F
Grande	20	S-shape	0,62	0,43	3,93
Grande	20	Brecha más grande	0,81	0,37	3,93
Grande	40	S-shape	2,11	0,15	4,01
Grande	40	Brecha más grande	1,65	0,2	4,01
Grande	60	S-shape	0,15	0,7	4,01
Grande	60	Brecha más grande	0,03	0,87	4,01

Los resultados obtenidos para el análisis de varianza del almacén grande permiten evidenciar que el factor (tipo de almacenamiento) ABC o NO ABC, no es significativo en la distancia, ya que el valor f es menor al f crítico, por esta razón se acepta la hipótesis nula de que todos los promedios de los grupos son iguales.

Tabla 28. Consolidado de valores diagrama cajas y bigotes tipo de almacenamiento ABC/NO ABC para instancias grandes.

Tamaño de almacén	Tamaño de muestra (n)	Heurística	Tipo de almacenamiento	Distancia promedio
Grande	20	S-shape	NO ABC	228
Grande	20	S-shape	ABC	223

HEURISTICAS PARA PROBLEMAS DE ENRRUTAMIENTO

Grande	20	Brecha más grande	NO ABC	250
Grande	20	Brecha más grande	ABC	243
Grande	40	S-shape	NO ABC	260
Grande	40	S-shape	ABC	251
Grande	40	Brecha más grande	NO ABC	333
Grande	40	Brecha más grande	ABC	343
Grande	60	S-shape	NO ABC	271
Grande	60	S-shape	ABC	273
Grande	60	Brecha más grande	NO ABC	376
Grande	60	Brecha más grande	ABC	374

En los resultados obtenidos se evidencia que no existe una diferencia relevante entre las distancias promedio teniendo en cuenta el tipo de almacenamiento por lo tanto se puede decir que no es significativo en la distancia el tipo de almacenamiento.

Tabla 29. Consolidado análisis de varianza entre heurísticas para instancias grandes.

Tamaño de almacén	Tamaño de muestra(n)	F	Probabilidad	Valor crítico para F
Grande	20	18,039	0,00	3,92
Grande	40	390,56	0,00	3,92
Grande	60	934,48	0,00	3,92

Los resultados obtenidos para el análisis de varianza del almacén grande permiten evidenciar que el factor heurística s-shape y la brecha más grande, es significativo en la distancia, ya que el valor f es mayor al f crítico, por esta razón se rechaza la hipótesis nula de que todos los promedios de los grupos son iguales.

Tabla 30. Consolidado de valores diagrama cajas y bigotes heurísticas.

Tamaño de almacén	Tamaño de muestra (n)	Heurística	Distancia promedio
Grande	20	S-shape	225
Grande	20	Brecha más grande	248
Grande	40	S-shape	258
Grande	40	Brecha más grande	335
Grande	60	S-shape	271
Grande	60	Brecha más grande	376

HEURISTICAS PARA PROBLEMAS DE ENRRUTAMIENTO

En los resultados obtenidos se evidencia que existe una diferencia relevante entre las distancias promedio teniendo en cuenta las heurísticas por lo tanto se puede decir que es significativo en la distancia la heurística.

Tabla 31. *Número de veces que se aprecia menor distancia por almacén para cada heurística.*

Almacén	N° de veces menor distancia		
	S-Shape	Brecha más grande	Total
Pequeño	180	0	180
Mediano	180	0	180
Grande	167	13	180
Total	527	13	540
%	97,6%	2,4%	100,0%

En el apéndice diferencia en la distancia entre las dos heurísticas se aprecia que las heurísticas propuestas en esta investigación ofrecen buenas soluciones, por lo que se logró cumplir lo planteado inicialmente donde se pretendía proponer un algoritmo que nos permitiera realizar la comparación entre las distancias de las heurísticas s-shape y la brecha más grande y brindar soluciones factibles, en este caso las menores distancias las obtuvo la heurística s-shape en un 97,6% a comparación de la heurística la brecha más grande ya que dicha heurística obtuvo el 2,4%.

9 Conclusiones

Posterior a la revisión de la literatura se logra concluir que con el pasar de los años se evidencia un crecimiento exponencial en las publicaciones donde se observa el interés de investigar sobre el picking routing problem puesto que se considera una operación crítica para la mayoría de las empresas dado que la recolección de ordenes es la actividad más costosa en un almacén debido a que representan gran parte de los costos operativos.

Como resultado de los análisis de variancias del diseño de experimentos 2^2 aplicado a la heurística y al tipo de almacenamiento, se encontró que el factor heurística es el que tiene mayor impacto sobre el valor distancia, también se observa que entre más grande sea el almacén

HEURISTICAS PARA PROBLEMAS DE ENRRUTAMIENTO

el factor tipo de almacenamiento no presentan un gran impacto sobre la distancia que se debe recorrer para recolectar las ordenes en su totalidad.

Tras el análisis, podemos deducir que para la heurística s-shape en instancias pequeñas la distancia tiende a ser la misma entre más grande sea el número de ordenes por recolectar, en instancias medianas y grandes entre más grande sea el número de ordenes por recolectar el rango de distancias entre la máxima distancia y la mínima distancia tiende a disminuir.

Tras el análisis, podemos deducir que para la heurística la brecha más grande en instancias pequeñas entre más grande sea el número de ordenes por recolectar el rango de las distancias entre la máxima distancia y la mínima distancia tiende a disminuir, para las instancias medianas y grandes el rango mínimo que se obtiene entre la máxima distancia y la mínima distancia es cuando el número de órdenes por recolectar es el tamaño mediano.

Teniendo en cuenta los resultados totales que se obtuvieron en la instancia pequeña, mediana y grande pudimos evidenciar que la heurística la brecha más grande obtuvo mejores resultados en un 2,4% en cambio la heurística s-shape obtuvo mejores resultados en un 97,6%, al entrar en detalles con los tres tamaños de ordenes por recolectar que se tuvieron en cuenta, se puede evidenciar que la heurística la brecha más grande obtuvo mejores resultados que la heurística s-shape solamente en instancias grandes y tamaño pequeño de órdenes a recolectar, sin embargo solo representan el 21% de la iteraciones, el 79% restante corresponde a mínima distancia de la heurística s-shape.

El tipo de almacenamiento ABC, es un método de clasificación de inventarios que permite asociar a cada SKU, con una clase o categoría de productos, con respecto a su importancia para la empresa, al ser un tipo de inventario que ayuda a las organizaciones a identificar que productos son críticos para el éxito y rentabilidad del negocio, en esta investigación se abordó este tipo de almacenamiento y se quiso identificar si era significativo

HEURISTICAS PARA PROBLEMAS DE ENRRUTAMIENTO

en la distancia, después de realizar los análisis de varianzas correspondientes se logró concluir que en esta investigación este tipo de almacenamiento no fue significativo.

10 Recomendaciones

Se recomienda a los futuros investigadores continuar trabajando en el tema de investigación picking routing problema en diferentes aspectos, en especial lo relacionado al ecommerce debido a que en la actualidad la tendencia cotidiana es realizar sus pedidos de forma online, teniendo en cuenta los métodos heurísticos presentados en la investigación que facilitan la recolección de ítems teniendo en consideración diferentes parámetros y restricciones. Con el fin de contribuir a la disminución de los costos operativos y a su vez encontrar las rutas más factibles con menores tiempos de corridas.

Para futuras investigaciones se puede considerar abordar este problema con otro tipo de almacenamiento verificando ambas heurísticas para analizar si se presentan cambios en los resultados de las distancias.

Al realizar los análisis de variancia tipo de almacenamiento ABC/NO ABC en las instancias pequeñas queda la ambigüedad de la significancia en el tipo de almacenamiento respecto a la distancia, dado que al analizarla a detalle por tamaño de ordenes los resultados no fueron contundentes que no había relación, por lo tanto, se invita hacer énfasis en revisar dicha situación.

Referencias Bibliográficas

- Arboleda, J. y Castillo, J.A. (2016). Modelo integrado de clasificación abcmulticriterio, aplicado en el área de picking de un centro de distribución de repuestos. Colección Académica de Ciencias. *Estratégicas*, 3(2), 15-34.
- Arriola, R., y Ramos, F. (2019). *Framework basado en heurísticas que optimizan la ruta para recolectar productos en almacenes*. (Tesis de pregrado). Universidad autónoma de ciudad Juárez, Ciudad Juárez, México.
- Camacho, S., y Ramírez, G. (2020). *Un modelo multiobjetivo para el problema de ruteo de vehículos para la distribución de ayudas humanitarias para un cismo en la ciudad de Bucaramanga*. (Tesis de pregrado). Universidad industrial de Santander, Santander, Colombia.
- CHOE, Kyung. SHARP, Gunter (1991). Small parts order picking: design and operation. Recuperado el 11 de noviembre de 2021 de <http://www2.isye.gatech.edu/~mgoetsch/cali/Logistics%20Tutorial/order/article.htm>.
- DE KOSTER, René. DUC, Le. ROODBERGEN, Kees. Design and control of warehouse order picking: A literature review. En: *European Journal of Operational Research*. 182(2), (2007); p. 481-501.
- García, J. (2006). *Modelado mediante optimización combinatoria*. [Archivo pdf]. Personales upv. Recuperado el 16 de noviembre de 2021 de <http://personales.upv.es/jpgarcia/LinkedDocuments/MCOIOptimizacionCombinatoria.pdf>.
- Garza, G. (2018). *Metodología para optimización de proceso de preparación de pedidos en rutas de distribución secundaria mediante rediseño de almacén de vehículo de carga*.

HEURISTICAS PARA PROBLEMAS DE ENRRUTAMIENTO

(Tesis de maestría). Instituto Tecnológico y de Estudios Superiores de Monterrey, Monterrey, México.

Hernández, B. (22 de diciembre de 2018). *Software vosviewer bibliométria*. Investiga y educa. Recuperado el 30 de septiembre de 2021 de <https://www.investigayeduca.com/software-vosviewer-bibliometria/#:~:text=El%20software%20VOSviewer%20versi%C3%B3n%201.6,citas%20o%20relaciones%20de%20autor%C3%ADas.>

Hillier, F., y Lieberman G. (2017). *Introducción a la investigación de operaciones*. McGrawHill. https://dudasytareas.files.wordpress.com/2017/05/hillier_lieberman.pdf.

Jiménez, V., Manotas, D. y Villota, R. (2013). *Configuración de pasillos en centros de distribución basada en modelos no tradicionales: modelo espina de pescado*. SciELO. Recuperado el día 10 de noviembre del 2021 de http://www.scielo.org.co/scielo.php?script=sci_arttext&pid=S1900-38032013000100014

Juarez, J. (06 de septiembre de 2013). *Método heurístico*. Slideshare. Recuperado el 13 de noviembre de 2021 de <https://es.slideshare.net/profjavierjuarez/metodo-heurstico-1>.

López, J. F. (19 de febrero de 2019). *logística de almacenamiento*. Economipedia.com. Recuperado el 13 de noviembre de 2021 de <https://economipedia.com/definiciones/logistica-de-almacenamiento.html.z>

Makusee Masae, Christoph H, Glock a & Eric H. Grosse.(2020).Order picker routing in warehouses: A systematic literature review. Recuperado el 21 de noviembre de <https://www.sciencedirect.com/science/article/abs/pii/S0925527319304050>

Masae M., Glock C. & Grosse E. (2020). Order picker routing in warehouses: A systematic literature review. *Revista internacional de economía de la producción*. Recuperado el 12 de noviembre de 2021, de la base de datos ScienceDirect

HEURISTICAS PARA PROBLEMAS DE ENRRUTAMIENTO

Mecalux. (11 de noviembre 2020). *Las ventajas del método ABC para la clasificación de inventarios en el almacén*. Mecalux. Recuperado el 20 de noviembre de 2021, de <https://www.mecalux.com.co/blog/metodo-abc-clasificacion-almacen>.

Melian, B., Moreno, J. y Moreno, J. (2003). Inteligencia artificial. *Iberoamericana de inteligencia artificial*.
<https://sci2s.ugr.es/sites/default/files/files/Teaching/GraduatesCourses/Metaheuristica/s/Bibliography/metaheuristicas-vision-global.pdf>

Merlano, K., y Castellano K. (2021). *Un Algoritmo de Búsqueda Tabú para El Problema de Enrutamiento de un Recolector (SPRP) en un Almacén de Comercio Electrónico con Almacenamiento Disperso y Múltiples Depósitos*. (Tesis de pregrado). Universidad industrial de Santander, Santander, Colombia.

Múnera Cardona. (2020). Optimización de la estrategia de Slotting en un centro de distribución logístico de autopartes mediante técnicas heurísticas basadas en computación evolutiva. Recuperado el 11 de noviembre de 2021 de <https://repositorio.unal.edu.co/bitstream/handle/unal/78033/1216714316.2020.pdf?sequence=4&isAllowed=y>

Niño, A., y Vargas, J. (2018). *Algoritmo memético para el problema de localización y ruteo de vehículos con ventanas de tiempo en el proceso de recolección de residuos sólidos en el centro de la ciudad de Bucaramanga de la empresa de aseo de Bucaramanga-EMAB*. (Tesis de pregrado). Universidad industrial de Santander, Santander, Colombia.

Ömer, Ö. & Deniz H., (2018). A discrete cross aisle desing model for order-picking warehouses. *Elsevier*. Recuperado el 13 de noviembre de 2021, de la base de datos ScienceDirect

Pascual Montero. (2019). Modelado y resolución del problema de rutas de preparación de pedidos conjuntos. Recuperado el 20 de noviembre de 2021 de

HEURISTICAS PARA PROBLEMAS DE ENRRUTAMIENTO

<https://idus.us.es/bitstream/handle/11441/94779/TFM-9624-PASCUAL%20MONTERO.pdf?sequence=1&isAllowed=y>

Platziteam. (01 de marzo de 2021). *Que es Python y por qué empezar a programar con este lenguaje*. Platzi. Recuperado el día 15 de noviembre del 2021 de https://platzi.com/blog/python/?utm_source=google&utm_medium=cpc&utm_campaign=12915366154&utm_adgroup=&utm_content=&gclid=EAJalQobChMI6L2E2KO b9AIVAo-GCh1vpgseEAAYAiAAEgI_A_D_BwE&gclsrc=aw.ds.

Rivera, P., y Jaramillo, O. (2020). *Solución del problema de ruteo de vehículos capacitado (CVRP) usando el algoritmo de aprendizaje reforzado Q-Learning*. (Tesis de pregrados). Universidad industrial de Santander, Santander, Colombia.

Sadia Quader, Krystel K & Castillo Villa.(2018).Design of an enhanced multi-aisle order-picking system considering storage assignments and routing heuristics.Recuperado el 20 de noviembre de 2021 de <https://www.sciencedirect.com/science/article/abs/pii/S0736584516300072>

Sáenz, E. (2020). *Desarrollo de un modelo matemático para el problema de ruteo de inventario con múltiples depósitos (MDIRP)*. (Tesis de pregrado). Universidad industrial de Santander, Santander, Colombia.

Shetty, N., Sah, B. & Hoon, S. (2020). Route optimization for warehouse order picking operations via vehicle routing and simulation. *SpringerLink*. Recuperado el 10 de noviembre de 2021, de <https://bibliotecavirtual.uis.edu.co:2142/article/10.1007/s42452-020-2076-x>.