Facultad de Ingeniería
INGENIERÍA INDUSTRIAL
Trabajo de Grado – Primer Semestre 2019

Trabajo de grado en modalidad de aplicación

[183006] Diseño de aplicativo de ruteo con inventario entre proveedor y minoristas

Sebastián Felipe Cardozo Puentes [a,c], Juan Guillermo Moreno Hernández [a,c], Nicolás David Sanabria Bedoya [a,c], Jorge Arturo Guzmán Torres [a,c], John Leonardo Vargas Mesa [b,c].

[a]Estudiante de Ingeniería Industrial

[b]Director del Trabajo de Grado, Departamento de Ingeniería Industrial

[c]Pontificia Universidad Javeriana, Bogotá, Colombia.

Resumen de diseño en Ingeniería (En inglés)

The companies that distribute products face daily with problems associated to the design of the route management. The solution to this problem comprises the use of forecast of the demand, assignment of customers to distribution centers, determination of dates and assignment of vehicles for the distribution work.

In the mid 80's, some companies developed an inventory model where the seller was made charge of maintaining availability of products of its customers, this model es known as Vendor Management Inventory. Associated to this model of work between companies, at academic level, it developed the Inventory Routing Problem, this problem has worked with exact solution methods, heuristics, among others, with the objective of search the best way to meet the demand of the customers with stable inventory levels and low cost.

In the present work, it is shown the design of an intuitive application which supports and complements the work of routing planning and inventory management through a metaheuristic that generates quality solutions in efficient times. The application was programmed in Visual Basic for Applications of Microsoft Excel with an user interface where it is collected and administered the data provided by the company that is using which is guided from the introduction of the application, until the obtaining of results that facilities the use for the user, the comprehension of the develop and the results.

With respect to the creation of the Application, the setup was divided in four branches. Design of initial solution, design of metaheuristic, design of an interface and design of calibration, where each one of the branches are adapted for create an application of high quality.

The heuristic works through three fundamental pillars, the first pillar is the obtaining of an initial solution. This initial solution is generated randomly and allow visualize a field of feasible solutions wider. The second pillar, refers to the local search, whereby we want to improve the initial solution for later rum the third pillar that refers to a perturbation or modification of the local search within the range of feasible solution with the purpose of explore and verify different solutions and finally compare the initial solution with the local search with the perturbed solution and select the best of the three. At the end of comparison, it runs again sequentially each of the three pillars and it is comparing with the best solution selected, this cycle is run repeated as many times as necessary until it obtains the best solution of the comparison.

One of the considerations more important to keep in mind is the develop of the interface, nowadays it is not enough that an efficient software with its optimal in the best case, it needs to be friendly, understandable, and comfortable for the users. For this reason, the develop of the interface of the application is based on the practice for the customer, that was developed to collect the input data through the option *company* which allows them to download the format in the book of Excel to fill the required data, load this data

to the application and continue to run the algorithm and see the presentation of results. Additionally, the customer can protect its data through user and password that is requested in the initialization of the application.

Finally, the heuristic was compared with the results obtaining in the doctoral work of Archetti and we concluded that:

- For instances between 5 and 15 customers, we obtain an average GAP of 10,5% with a maximum of 19,1% and minimum of 0,0%.
- For instances between 20 and 35 customers, we obtain an average GAP of 17,4% with a maximum of 23,6% and minimum of 7,2%.
- For instances between 40 and 50 customers, we obtain an average GAP of 24,3% with a maximum of 34,8% and minimum of 14,0%.

In terms of execution times, it can be seen that:

- For instances between 5 and 15 customers the average of time is 258,4 seconds.
- For instances between 20 and 35 customers the average of time is 267,6 seconds.
- For instances between 40 and 50 customers the average of time is 235,5 seconds in comparison to 2733,9 seconds of the exact method.

Se obtuvo un GAP promedio de 17.7% para 400 instancias ejecutadas

1. Justificación y planteamiento del problema

Las empresas enfrentan diferentes tipos de problemas dependiendo de la labor que desarrollan a diario, para dichos problemas es necesario buscar soluciones de calidad que generen valor a la vez que mejoran el uso de los recursos disponibles. En ocasiones, se puede observar como muchas compañías poseen disyuntivas entre mantener los costos de distribución y garantizar los niveles de servicio asociados a la disponibilidad de productos para la venta, esto debido a que no cuentan con sistemas de rutas o de inventarios adecuados que les permita tomar decisiones de forma eficiente e informada.

En los años ochenta se desarrolló una estrategia de gestión de inventarios entre proveedores y clientes llamada Vendor Managed Inventory[1], (VMI) la cual busca disminuir los niveles de inventario mediante la asociación con los proveedores, de tal manera que el proveedor tenga la suficiente información para abastecer a su cliente justo cuando el stock esté en un nivel mínimo. Durante su desarrollo, VMI tuvo varios inconvenientes ya que muchas empresas se negaban a dar información acerca de sus niveles de inventario y datos relacionados, suponían que era información privilegiada y que compartiéndola podrían perder su ventaja competitiva en el mercado.

Al principio es un proceso extenuante teniendo en cuenta que las empresas sienten que están engendrando el síndrome del zorro que cuida el gallinero[2], estas empresas sentían que en cualquier momento sus proveedores podrían traicionar su confianza y vender la información al mejor postor, pero el zorro, en este caso los proveedores, aprendieron que la mejor forma de mantener al cliente es haciéndose cargo de los inventarios y aumentando la confianza para cooperar juntos a lo largo de la red de suministro. Con el paso del tiempo, surgieron casos de éxito de empresas que utilizando VMI obtuvieron diferentes beneficios. BOSE, Harley Davidson y Dell's computers son casos que implementaron esta estrategia de forma exitosa.

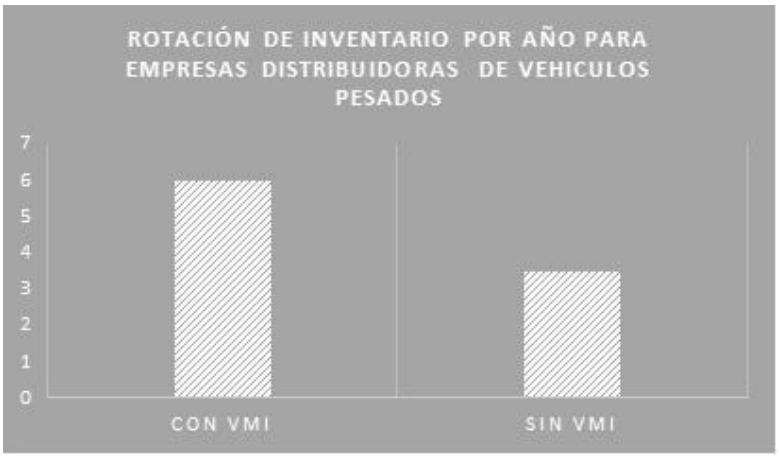
Uno de los problemas más comunes en la cadena de suministro es la incertidumbre sobre la demanda, ya que tanto el proveedor como el cliente pueden verse afectados por las altas y bajas, conocidas bajo el nombre de 'efecto látigo'[3], en donde el proveedor por su parte tiene que cumplir pedidos en tiempos cortos y con un elevado costo para poder satisfacer la demanda variante de su cliente, mientras que el minorista tiene que abastecerse a costos altos para cubrir la demanda del consumidor.

Con VMI es posible observar que los requerimientos y niveles de inventario tanto del proveedor como del minorista se establecen con el paso del tiempo [4], ya que el proveedor puede programar y calcular los tiempos de ventas y transporte de productos con mínimos costos y a su vez el minorista maneja bajo inventario y niveles de servicio altos con costos razonables, dando solución a uno de los problemas más comunes en la cadena de suministro para ambas partes (minorista y proveedor). VMI es una estrategia de negocio cuya abstracción matemática se representa a través

del problema de ruteo con inventarios conocido comúnmente como Inventory Routing Problem (IRP), en el cual se desarrolla una estrategia periódica de reaprovisionamiento entre minoristas y proveedores y se combina con la planeación eficiente de rutas de distribución para cada periodo de tiempo.

Los proveedores de Walmart fueron algunas de las primeras empresas encargadas de solucionar problemas relacionados con el ruteo de inventarios, esto debido a que Walmart fue pionera en la implementación de estrategias VMI de abastecimiento con sus proveedores[4], el resultado de este proceso se ve reflejado en la reducción del flujo de inventario y en la incertidumbre de las unidades que permanecen en stock. En 1989, la compañía registró que el 1.7% de sus costos totales eran de distribución[5], siendo esta una cifra muy baja en comparación con sus competidores directos Kmart y Sears, los cuales presentaban un porcentaje del 3% y 5% respectivamente para sus costos de transporte con respecto a sus costos totales.

Además del aumento en la rotación de inventario es posible observar otros beneficios adquiridos mediante la consolidación de estrategias de VMI en las empresas. En primer lugar, se aprecia incrementos en los informes de ventas, teniendo en cuenta que los minoristas reducen sus unidades almacenadas, estos pueden ofrecer mayor variedad de productos a sus clientes aumentando sus ventas por la cantidad de productos ofrecidos. En adición, se muestra un incremento en las ventas del 24% al cabo de un año luego de implementar estas estrategias en las compañías[6]. Por otro lado, se determinó que antes de usar esta metodología el inventario de las empresas rotaba 4.82 veces al año, mientras que, con el VMI este hacía una rotación de 3.5 veces anualmente, como se aprecia a continuación en la figura 1[5].



Gráfica 1. *Comparación de rotación de inventarios por año, vehículos pesados*[5].

Con un mayor control del inventario y la estabilización de la demanda se pueden combatir los costos asociados a la obsolescencia de productos y devoluciones por parte de los clientes, también se reducen gastos administrativos, pues al tener contacto directo entre proveedores y minoristas, se reduce el tiempo de procesamiento y manejo de órdenes, donde se evitan desajustes en los pedidos que se realizan, permitiendo que el distribuidor genere menos requerimientos de producto, puesto que el proveedor es quien se encarga de todo el reaprovisionamiento.

Uno de los principales desafíos en el modelo de ruteo con inventarios por parte de los proveedores es ajustar su logística, incluyendo dentro del despacho de mercancías el mantenimiento del inventario. Específicamente es necesario mezclar las decisiones de distribución (ruteo) con las de aprovisionamiento del cliente (inventario). A nivel teórico el 'Inventory Routing Problem'[7] aborda este nuevo desafío. El problema de ruteo con inventario (IRP) agrupa aquellos problemas relacionados con gestionar los niveles de inventario con el ruteo de los vehículos de distribución entre los diferentes clientes; El objetivo común de este modelo es minimizar los costos asociados al almacenamiento y la distribución.

Asociado a la situación anteriormente descrita muchas empresas generalmente poseen varios depósitos distribuidos a lo largo de una ciudad o de una región ubicados estratégicamente para abastecer completamente a todos los clientes y disminuir la cantidad de camiones que distribuyen la mercancía y así mismo los productos; MD o multi-depot es un aspecto adicional asociado al problema de ruteo con inventarios 'IRP' en el que además de tener decisiones asociadas al costo de inventarios mínimos y el ruteo del producto óptimo se debe tener en cuenta que las empresas proveedoras poseen múltiples depósitos, lo cual puede ayudar a mitigar los costos relacionados con faltantes y ruteo de productos

entre clientes. Esta característica adicional hace necesario la incorporación de decisiones de asignación de clientes a depósitos.

2. Antecedentes

El problema de ruteo con inventarios es un tema ampliamente tratado, existen diferentes artículos, investigaciones y revistas que hablan sobre el desarrollo de heurísticas y metaheurísticas con el fin de solventar los problemas relacionados que se presentan en diferentes actividades y operaciones referentes al ruteo y el nivel de inventario. Metaheurísticas como algoritmos genéticos, optimización por enjambre de partículas y búsqueda tabú son las más utilizadas para la solución de estos problemas. Dadas las características de este problema puede ser categorizado entre los problemas NP dado que contiene al Traveling Salesman Problem[1] cuando solo existe un depósito y un periodo de tiempo con una demanda de una unidad en cada cliente.

El doctor Nevin Aydin hacia el año 2014 escribió acerca de la implementación de un algoritmo genético para resolver el problema de ruteo con inventarios[2], en su escrito se evidencia la importancia de la planeación en el horizonte de tiempo para el modelo de inventarios conjuntamente con el ruteo del material; en el artículo, se mostró además la respectiva mejoría al implementar el algoritmo para saber la cantidad de producto a mantener en el inventario y las rutas óptimas para transportarlo desde el depósito hasta el cliente.

Algunos datos importantes obtenidos del artículo del doctor Adyin detallan que si se busca implementar un algoritmo genético se debe tener en cuenta parámetros dentro del algoritmo para mantener los individuos con cromosomas que incluyan el ruteo hacia todos los clientes y que dentro de estas rutas el cliente sea visitado una única vez. Analizar a los individuos por una puntuación que considere la cantidad de producto transportado en cada ruta, el tiempo necesario para realizar el ruteo y la cantidad de clientes abastecidos con cada una de las rutas, también es un aspecto a tener en cuenta.

Las restricciones al momento de programar un algoritmo genético son parte importante para encontrar soluciones cercanas al óptimo y que ajusten a cada modelo dependiendo del tipo de estudio o de empresa al que se le quiere buscar solución al problema de ruteo con inventarios, tal y como lo detallan Seyed Hamid, Seyed Taghi y Ali Roozbeh en su artículo acerca del control de inventarios para empresas bajo el modelo de vendor managed inventory[3]. En dicho artículo, se estructura también un algoritmo genético enfocado principalmente en dar solución para el ruteo entre un solo proveedor y un solo cliente, en los cuales se comercializan diferentes productos y con un modelo de ordenes faltantes, el modelo utilizado para el inventario del proveedor es el EOQ, por su parte, los espacios de almacenamiento tanto para el proveedor como para el minorista son mínimos y las soluciones conseguidas son eficientes en relación con el modelo propuesto por los autores ya que minimiza notablemente los costos asociados a los inventarios del proveedor.

En el modelo anterior se probaron 24 instancias cada una con 10 repeticiones, estas instancias probadas estaban enfocadas en verificar el orden de realización de los pedidos y establecer la cantidad máxima de pedidos atrasados. Se pudo concluir con dicho artículo que para el presente trabajo de grado se deber considerar escenarios adicionales como ventas perdidas en lugar de pedidos atrasados, incluir múltiples depósitos, más clientes, entre otros aspectos.

Como bien se mencionó anteriormente, una de las metaheurísticas más utilizadas es la de optimización por enjambre de partículas. Seyed Taghi del departamento de Ingeniería Industrial de la universidad de tecnología de Sharif junto a Saeid Sadeghi, Javad Sadeghi diseñaron un algoritmo mejorado basado en la optimización por enjambre de partículas para optimizar un modelo hibrido de VMI y problema de ruteo con demanda difusa[4]. En su trabajo de

---

[1] Dantzig, G., Fulkerson, R., & Johnson, S. ***(1954). Solution of a Large-Scale Traveling-Salesman Problem. Journal Of The Operations Research Society Of America,*** 2(4), 393-410.

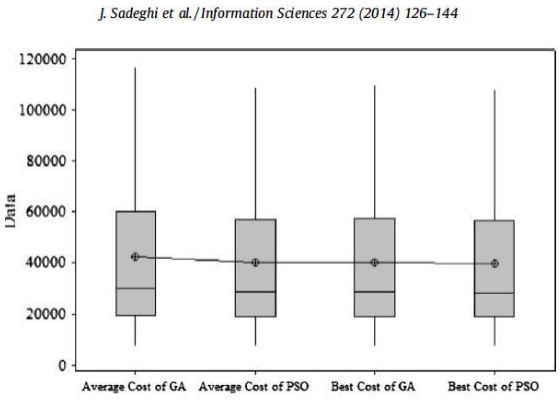
[2] Aydin, N. *A genetic algorithm on inventory routing problem. Emerging Markets Journal,* 2014.

[3] Reza. S, Akhavan.S, Roozbeh. A. *A genetic algorithm for vendor managed inventory control system of multi – product multi - constraint economic order quantity model: Elsevier Inc.* 2011.

[4] Sadeghi. J, Saeid. S, Niaki. S. *Optimizing a hybrid vendor-managed inventory and transportation problem with fuzzy demand: An improved particle swarm optimization algorithm: Elsevier Inc.* 2014

investigación buscaron la cantidad mínimas de órdenes a cumplir y las rutas óptimas para abastecer a cada uno de los clientes con el fin de disminuir los costos asociados al nivel de inventarios y transporte.

Para probar la factibilidad del algoritmo por enjambre de partículas, los autores programaron en paralelo un algoritmo genético con la misma finalidad para poder comparar y corroborar las soluciones obtenidas, adicional a lo anterior, los autores utilizaron el método Taguchi para poder optimizar el tiempo de ejecución de los algoritmos. En la figura 2 se visualiza mejor dicha comparación, se tomó uno de los gráficos de dicho artículo en el que se observa por medio de un diagrama de caja y bigotes el promedio de los costos totales y el menor costos obtenido en cada uno de los algoritmos. Cabe resaltar que existe baja variabilidad en los costos totales entre la utilización de un algoritmo por enjambre de partículas y un algoritmo genético, por tal razón, el aplicativo del presente trabajo de grado realizará la optimización mediante un algoritmo genético.



Gráfica 2. *Box plot de las funciones de costos obtenidas mediante GA y PSO*[8]

Los algoritmos genéticos en la solución del problema con inventarios parecen ser una buena alternativa, teniendo en cuenta la gran cantidad de artículos e investigaciones acerca del IRP que utilizan para su desarrollo algoritmos genéticos. El algoritmo genético es una metaheurística que además de ofrecer buenas soluciones, si se programa correctamente puede ofrecer tiempos medianamente cortos.

Además de los trabajos encontrados que desarrollaron algoritmos genéticos y optimización por enjambre de partículas es posible encontrar escritos que utilizaron por su parte la búsqueda tabú como herramienta para encontrar buenas soluciones.

Jacques Renauld, Gilbert Laporte y Fayez Boctor formularon una búsqueda tabú con el fin de encontrar solución al problema de ruteo con múltiples depósitos[5]. Los autores adaptaron la búsqueda tabú a su escenario especifico consiguiendo que las soluciones mejoraran en 20 de 23 instancias probadas. Además, programaron una versión con tiempos de ejecución más cortos y con mejores soluciones en 19 de 23 instancias probadas, demostrando que la búsqueda tabú en escenarios relacionados con el ruteo de vehículos entre múltiples depósitos ofrece soluciones favorables, sin embargo, no consideraron el problema conjunto con el nivel de inventarios.

En cuanto a la literatura del IRP con múltiples vehículos, el articulo "A hybrid Granular Tabu Search algorithm for the Multi-Depot Vehicle Routing Problem"[6] encuentra soluciones para el enrutamiento de vehículos con múltiples depósitos, es decir, se especializan esencialmente en garantizar las rutas optimas entre depósitos y clientes. Para conseguir soluciones cercanas al óptimo para dicho enrutamiento se realizó un hibrido de varios algoritmos que se complementaron entre sí. Primero se asignaron clientes a cada depósito, luego se resolvió un VRP por medio de

---

5 Renaudl, Jacques, Laporte, Gilbert and Boctor, Fayez F., 1995, *A tabu search heuristic for the multi-depot vehicle routing problem*. Elsevier Science Ltd.

6 Escobar, John Willmer, Linfati, Rodrigo, Toth, Paolo and Baldoquin, Maria G., 2014, *A hybrid Granular Tabu Search algorithm for the Multi-Depot Vehicle Routing Problem*. Springer.

algoritmos de ahorro. Finalmente, se mejoraron las soluciones obtenidas por medio de un algoritmo de búsqueda tabú modificado.

Todos los aspectos relacionados con la programación de metaheurísticas influyen directamente en la obtención de buenas soluciones. Unnikrishnan y Karoonsoontawong poseen una muy buena investigación acerca del problema de ruteo con inventarios con una solución estructurada bajo la metaheurística búsqueda local[7]. Su proyecto buscó en su momento la mejor manera de abastecer un almacén de cadena manteniendo óptimos niveles de inventario y encontrando rutas con tiempos cortos. Se basaron además en la construcción de modelos probabilísticos en los que estudiaban el comportamiento del programa al momento de incumplir con alguna restricción. Llegaron a la conclusión que la variabilidad en la demanda repercute directamente en encontrar soluciones cercanas al optimo, ya que cuando la variabilidad de la demanda era alta, las funciones objetivos obtenidas mejoraban considerablemente en la búsqueda tabú.

La búsqueda tabú en esencia es una metaheurística que ofrece buenas soluciones cuando de programar el ruteo de vehículos se trata, pero este proyecto de grado está enfocado principalmente en el problema conjunto del ruteo con inventarios por lo que no se tendrá en cuenta la programación del aplicativo con búsqueda tabú.

Existen múltiples formas de resolver el problema de ruteo con inventarios, dentro de estas formas están los algoritmos o heurísticas propias y la programación entera lineal. Muchos autores sugieren la programación mediante heuristicas con el fin de programar pensando directamente en el escenario al cual se le quiere encontrar solución y ajustando la programación de acuerdo con los parámetros y restricciones propios del caso. "Mixed integer linear programming model for multicommodity multi-depot inventory routing problem"[8] es un artículo en el cual se presenta la solución de ruteo con inventarios mediante el modelo de integración mixta de programación lineal. En este caso, el problema tiene en cuenta que los proveedores poseen múltiples depósitos para abastecer a todos sus clientes, muy similar a la propuesta presente en este trabajo.

Paralelo a lo anterior, el articulo tiene en cuenta un proveedor con diferentes productos para ofrecer a sus clientes. Cabe resaltar que el problema de ruteo con inventarios tiene una generalidad en la que hay M cantidad de fabricantes que producen cada uno un tipo distinto de producto, por lo que el problema se enfoca en abastecer cada tipo de producto de cada fabricante en los camiones y de esta manera cubrir con la demanda especificada por cada cliente en específico.

En este caso se desarrolló la integración mixta de varios modelos para la solución del problema de ruteo. Para optimizar las rutas, se realizó VRP o ruteo por proximidad, en cuanto a los inventarios se equilibraron los costos de mantener inventarios junto con el de hacer el ruteo de la flota homogénea de transporte de productos.

Otros artículos relacionados con la programación heurística para dar solución al problema de ruteo con inventarios relacionan varios productos, varias restricciones y modelos EOQ para el manejo de inventarios. Cárdenas, Treviño y Ming relacionan en uno de sus artículos la forma en como programaron un algoritmo para poder encontrar solución a dicho problema[9]. En su trabajo de investigación, ellos demuestran que el algoritmo propuesto es mejor que la implementación de un algoritmo genético en tres simples aspectos el costo total, el número de evaluaciones de la función objetivo y el tiempo computacional. Aunque realizar una heurística propia puede parecer una alternativa a la implementación del algoritmo genético, es importante tener en cuenta que las heurísticas especializadas en ciertos escenarios no consideran diferentes generalidades para su aplicación en diferentes empresas, como nuestro trabajo de grado se centra en un aplicativo que pueda ser utilizado por cualquier compañía, es necesario implementar un algoritmo que se adapte a cualquier tipo de escenario.

---

[7] Karoonsoontawong, Ampol and Unnikrishnan, Avinash, 2013, *Inventory Routing Problem with Route Duration Limits and Stochastic Inventory Capacity Constraints*.

[8] Ramkumar, N., Subramanian, P., Narendran, T. T., & Ganesh, K. *Mixed integer linear programming model for multi-commodity multi-depot inventory routing problem*. *Opsearch, 49*(4), 413-429. 2012

[9] Cárdenas Barrón, Leopoldo Eduardo, Treviño Garza, Gerardo and Ming Wee, Hui, 2012, *A simple and better algorithm to solve the vendor managed inventory control system of multi-product multi-constraint economic order quantity model*.

Varias investigaciones encontradas no solamente se centran en encontrar soluciones cercanas al óptimo para disminuir los costos y tiempos en los que incurren las compañías, algunas investigaciones vieron la necesidad de aportar también en programas de ayuda y disminuir las emisiones de gases para contribuir en la construcción de una mejor sociedad.

El artículo "Solution to the Multidepot Inventory Slack-Routing Problem at the Planning Stage" [10]presenta un enfoque diferente para el problema de ruteo con inventarios. En este trabajo se busca ayudar a los grupos de socorro y ayuda para afrontar la difícil tarea de enviar recursos médicos y alimenticios en momentos de desastres naturales, ataques terroristas, entre otros. El problema planteado tiene como parámetros los múltiples depósitos de los grupos de socorristas a lo largo del mundo, la diferente capacidad de los camiones para enviar los insumos, los tiempos mínimos que deben acatar estos grupos de ayuda para enviar dichos insumos, con el fin de obtener el ruteo de sus inventarios de la forma más eficiente.

En Tamilnadu, India trabajaron en la investigación de algoritmos que estuvieran estructurados bajo el modelo de IRP[11], pero que como función objetivo disminuyeran la cantidad de emisiones de dióxido de carbono con el fin de contribuir en gran medida en el fuerte impacto que producen los vehículos encargados de toda la logística en almacenes. El reto en esta investigación fue además de encontrar tiempos mínimos y niveles de inventarios óptimos, encontrar la manera de reducir la polución que producían los camiones que transportaban desde los depósitos hacia los almacenes, por medio de la planificación de la demanda, la cantidad de inventario para satisfacer al cliente el menor número de veces al mes y por último encontrar la manera de visitar al cliente el menor número de veces por mes.

En la Universidad Nacional de Colombia, en la sede de Medellín también se realizó una investigación similar a la que hizo en India[12]. En esta investigación se analizó el impacto que genera la implementación del modelo IRP en relación con los costos logísticos, la disminución de tiempos en rutas, los costos por mantener inventario y adicionalmente se analizó el impacto ambiental que generan las empresas cuando utilizan el modelo IRP.

Adicional a todos los escritos e investigaciones encontradas, existen varias aplicaciones en la internet que permiten realizar el ruteo de productos de uno o más depósitos hacia diferentes clientes. A continuación, se presentan las aplicaciones que actualmente se encuentran abiertas al público que le dan solución a este problema:

- La aplicación SmartMonkey.io es una herramienta para las hojas de cálculo de Google que permite encontrar soluciones cercanas al optimo mediante la integración de varios servicios de Google y la información proveniente de la empresa. Desarrolla soluciones de planeación de ruteo eficientes que permiten que las empresas mejores su eficiencia operacional, sin embargo, esta aplicación no integra el nivel de inventarios en su desarrollo.

- LINGO: Lingo (Linear Generalize Optimizer) es una herramienta que permite la formulación de problemas lineales y no lineales, resolverlos y realizar un análisis sobre los resultados obtenidos. Lingo es una plataforma que permite encontrar los mejores resultados ya sea de maximizar ganancias o minimizar costos. A continuación, en la imagen 1 se ilustra un ejemplo sobre la formulación de un problema de optimización en Lingo.

---

[10] Xianfeng, Yang y Shanjiang, Zhu, 2016, *Solution to the Multidepot Inventory Slack-Routing Problem at the Planning Stage*. American Society of Civil Engineers.

[11] Balamurugan, T, karunamoorthy, L, Arunkumar, N y Santhosh, D, 2018, O*ptimization of inventory routing problem to minimize carbon dioxide emissioN*. Anna University.

[12] Zapata Cortes, Julian Andres, Arango Serna, Martín Darío y Serna Urán, Conrado Augusto, 2018, *Comparison of three IRP-based models to reduce logistics costs and greenhouse gas emissions* •. Universidad Nacional de Colombia.

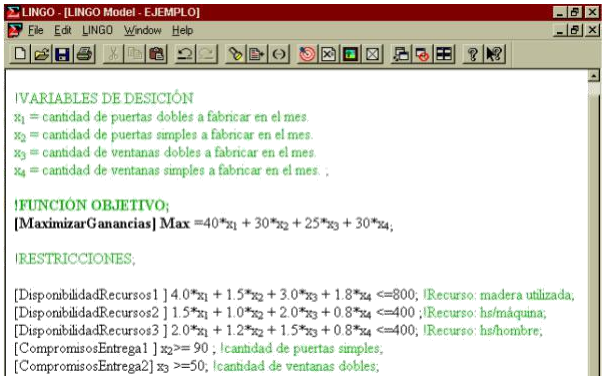


Imagen 1. *Software Para Programación Lineal. Elaborado por Canizo, E. Lucero, P, (2002).*

Lingo es una herramienta poderosa para la formulación y ejecución de diferentes problemas de optimización incluyendo algunos escenarios del IRP, sin embargo, se requieren conocimientos de programación para el desarrollo de esta herramienta lo cual restringe ampliamente la demanda del software.

- CPLEX: Es una herramienta que permite modelar problemas de negocios matemáticamente y resolverlos con diferentes algoritmos que producen soluciones precisas y lógicas. Esta herramienta permite la optimización de decisiones para aumentar la eficiencia operacional, disminuir costos y aumentar la rentabilidad. CPLEX proporciona solucionadores de programación matemática flexibles y de alto rendimiento para programación lineal, programación entera mixta, programación cuadrática y problemas con variabilidad en sus restricciones. Al igual que Lingo CPLEX requiere de conocimientos en programación avanzados para la ejecución de problemas de IRP.

- MATLAB: Permite el análisis iterativo y el desarrollo de procesos de diseño con un lenguaje de programación que expresa las matemáticas de matrices. Esta plataforma está diseñada para resolver problemas científicos y de ingeniería. Cuenta con graficas integradas las cuales facilitan el análisis de la información. Una de las características principales como se destaca en la página de la plataforma es el alto nivel de leguaje de programación que requiere el programa. A continuación, en la imagen 2 se ilustra un ejemplo de ejecución en Mathlab.

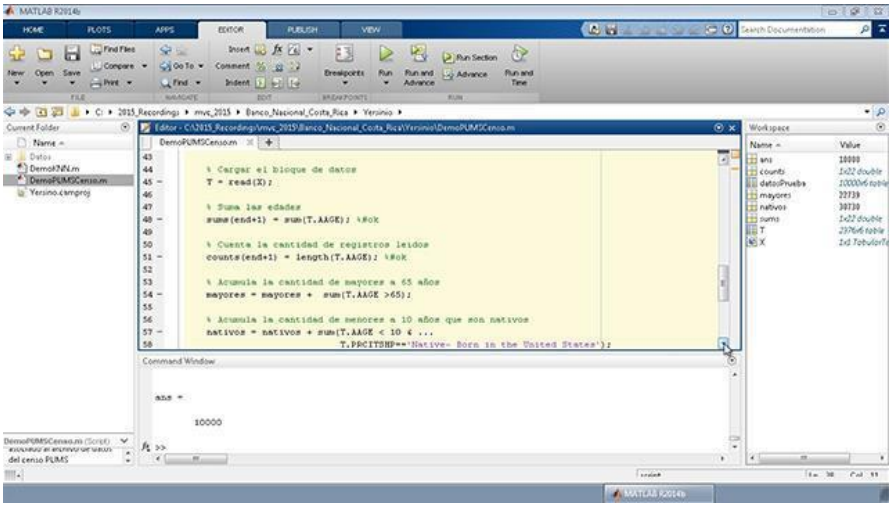


Imagen 2. *Programando con MATLAB. Elaborado por Hernández. G. (2013).*

- GAMS: Es un sistema general de modelaje algebraico el cual está diseñado con el fin de suplir principalmente dos necesidades: modelar problemas de optimización ya sean lineales, no lineales o mixtos, y realizar programación matemática. Permite construir modelos que se pueden adaptar a nuevas situaciones en diferentes sectores. Es una plataforma que está disponible en versiones para computadores personales,

estaciones de trabajo, bases de datos y super computadores. GAMS es principalmente útil para problemas de un gran tamaño y con un alto grado de complejidad. El sistema modela los problemas en una forma compacta a través de lenguaje de programación. GAMS presenta una ventaja sobre los anteriores programas ya que permite plantear un lenguaje de modelización que permite poder escribir en un editor la formulación matemática del problema y posteriormente aplicarle una serie de programas de resolución. En la imagen 3 y 4 se ilustra un ejemplo de programación en GAMS.

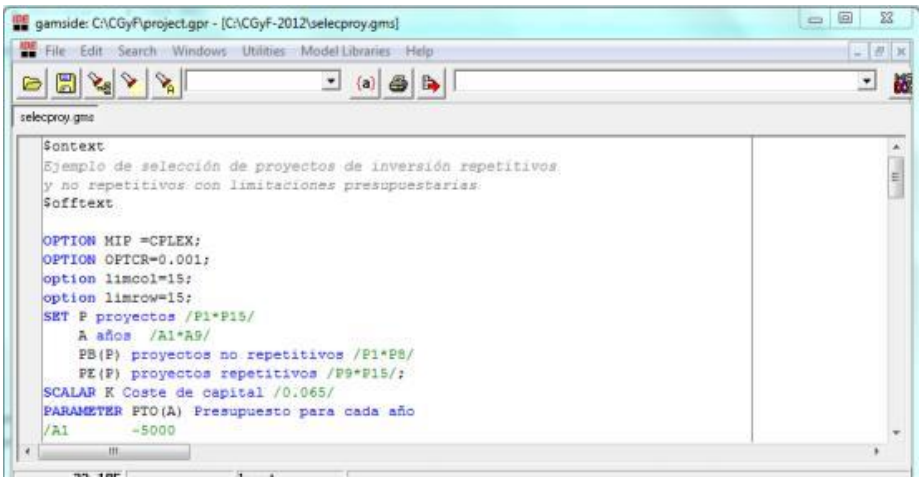


Imagen 3. *Control y gestión de finanzas. Elaborado por Mocholí. M. (2014).*

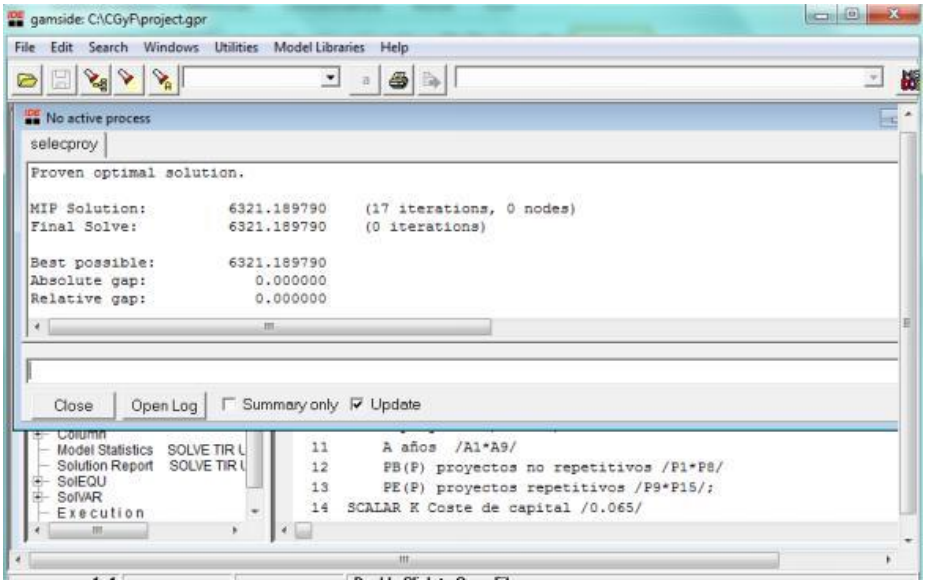


Imagen 4. *Control y gestión de finanzas. Elaborado por Mocholí. M. (2014)*

Como es evidente, en la actualidad existen varias plataformas muy poderosas en el mercado las cuales permiten obtener soluciones a diferentes problemas de optimización incluyendo el IRP, los programas anteriormente explicados son los más comunes si se habla del Inventory Routing Problem, un ejemplo de esto es el artículo llamado ”An inventory routing problem with soft time windows” Zhenping Li, Chongyiu Jiang, Lulu Jiang (2015) en donde desarrollan un modelo de programación para resolver un tipo de escenario del IRP y lo implementan en Lingo, otro ejemplo es el del artículo denominado “Blood inventory-routing problem under uncertainty” Kazemi, Rabbani, Tavakkoli, Charheza (2017) en donde ejecutan su formulación matemática en la plataforma CPLEX. Sin embargo, esta y la gran mayoría de las aplicaciones que existen actualmente en el mercado ofrecen servicios similares, en donde algunos no integran el nivel de inventarios con el ruteo de vehículos o simplemente no tienen la estructura de programación generalizada e intuitiva para que sea utilizada fácilmente por cualquier empresa.

En el presente trabajo de grado se realizará un aplicativo que implemente una metaheurística para poder disminuir los costos asociados a problemas de ruteo con inventarios, específicamente se implementará para el problema de ruteo e inventarios con múltiples depósitos. El aplicativo se diferencia de los anteriores trabajos porque estará enfocado en compañías que posean múltiples depósitos para cubrir la demanda de sus clientes, que cuenten con una flota uniforme de transporte y que operen bajo el modelo de ventas perdidas cuando no se cumpla con el pedido de un cliente.

Aunque existen diferentes escritos, investigaciones y artículos que exponen el tema del problema de ruteo con inventarios, no existen herramientas que ayuden a las empresas a solucionar fácilmente estos problemas ya que las herramientas o aplicaciones que existen actualmente en el mercado solo se enfocan en realizar el ruteo óptimo de los vehículos y no la integración de este ruteo con el manejo de los inventarios. Nuestro aplicativo es un complemento libre en Excel, fácil de descargar e instalar en cualquier computador, que pueda ser usado por cualquier empresa en búsqueda de una solución para el problema de ruteo de inventarios.

3. Objetivos

3.1 ***Objetivo General:*** *Desarrollar un módulo tipo complemento en el lenguaje VisualBasic.net para Excel MS ® que extienda las funcionalidades de este software adicionando opciones para la generación de soluciones con procedimientos meta-heurísticos a problemas relacionados con el Inventory Routing Problem.*

3.2 Objetivos Específicos:

- *Formular un modelo para el IRP capaz de adaptarse a diferentes situaciones problemáticas reales*
- *Diagnosticar el desarrollo de herramientas de software para la solución de problemas empresariales reales que presenten las características del IRP*
- *Crear una herramienta que integre métodos meta-heuristicos y el modelo del IRP dentro de las funciones de Excel MS ®*
- *Comparar los resultados obtenidos en la solución de instancias del IRP propuestas en la literatura contra las soluciones generadas por el aplicativo para medir su eficiencia.*

4. Metodología

Este proyecto de grado está basado en el modelo matemático propuesto en el año 2007 por Archetti en su artículo "A Branch-and-Cut Algorithm for a Vendor Managed Inventory Routing Problem", el cual es presentado y explicado a continuación:

Se considera una cadena de abastecimiento en donde un proveedor común debe satisfacer la demanda de un conjunto de minoristas $M=\{1,2,\ldots,n\}$ en un horizonte de tiempo H, en cada tiempo discreto $t \in T =\{1,2,\ldots,H\}$ una cantidad de producto r0t, el cual se pone a disposición del proveedor, pero es consumido por el minorista. Se definió un inventario inicial B0 y una capacidad máxima de inventario para cada minorista S.

Los proveedores poseen un inventario inicial y si se visita al minorista S en el momento t, se envía la cantidad de producto Xst de tal forma que el inventario del minorista alcance un valor máximo, es decir, se satisfaga toda la demanda del cliente, siendo esto una política determinística de nivel de pedido. Por su parte, el costo de inventario es cobrado tanto al proveedor como al minorista. Se denota H0 como el costo unitario del inventario para el proveedor y Bt el nivel de inventario del proveedor en el momento t, adicionalmente se incluyó en el cálculo de costos el momento H+1 para poder tener en cuenta las operaciones realizadas en el momento H. Para el minorista se denota hs como el costo unitario de inventario del minorista en el momento $s \in M$. El nivel de inventario al final de horizonte de tiempo puede ser diferente al nivel de inventario inicial, por tal razón, la demanda no tiene un comportamiento periódico. Los envíos entre el proveedor al minorista se pueden realizar en cada momento $t \in T$ mediante un camión de capacidad conocida C, el cual es capaz de atender a varios minoristas en un mismo viaje y cada viaje del vehículo visita a todos los minoristas que se atienden en el mismo momento. El costo de transporte entre proveedor y minoristas se denota Cij donde se el camión se traslada del vertice i al vertice j, se deja una variable binaria Ytij la cual es 1 si j sigue inmediatamente después de i en la ruta recorrida en el momento t, 0 de lo contrario. Para el problema se quiere

Determinar la cantidad de producto a enviar a cada minorista y la ruta que se debe tomar para visitar a todos los minoritas que deben abastecerse en el momento t. La función objetivo se presenta a continuación:

$$\sum_{t\in\mathcal{T}'} h_0 B_t + \sum_{s\in\mathcal{M}}\sum_{t\in\mathcal{T}'} h_s I_{st} + \sum_{i\in\mathcal{M}'}\sum_{j\in\mathcal{M}', j<i}\sum_{t\in\mathcal{T}} c_{ij} y_{ij}^t. \qquad (1)$$

Imagen 5. *Función objetivo modelo matemático Archetti.*

Se tuvieron en cuenta las siguientes restricciones:

1. **<u>Definición de nivel de inventario en el proveedor:</u>** El nivel de inventario del proveedor en el momento t está dado por el nivel de inventario del proveedor en t-1, más la cantidad de producto r0t-1 disponible en el momento t-1, menos la cantidad total enviada a los minoritas en el momento t-1.

$$B_t = B_{t-1} + r_{0t-1} - \sum_{s\in\mathcal{M}} x_{st-1} \qquad t \in \mathcal{T}', \qquad (2)$$

$$\text{where } r_{00} = 0 \text{ and } x_{s0} = 0,\ s \in \mathcal{M}.$$

Imagen 6. *Restricción inventario modelo matemático Archetti.*

2. **<u>Abastecimiento desde el proveedor hacia los minoritas:</u>** está restricción asegura que el proveedor pueda satisfacer la demanda de los minoritas en cualquier momento t.

$$B_t \geq \sum_{s\in\mathcal{M}} x_{st} \qquad t \in \mathcal{T}. \qquad (3)$$

Imagen 7. *Restricción abastecimiento modelo matemático Archetti.*

3. **<u>Nivel de inventario en los minoristas:</u>** El nivel de inventario de los minoristas está definido por el nivel de inventario de los minoritas en el momento t-1, más la cantidad recibida del proveedor en el momento t-1, menos la cantidad consumida en el momento t-1.

$$I_{st} = I_{st-1} + x_{st-1} - r_{st-1} \qquad s \in \mathcal{M} \quad t \in \mathcal{T}', \qquad (4)$$

$$\text{where } x_{s0} = r_{s0} = 0,\ s \in \mathcal{M}.$$

Imagen 8. *Restricción inventario minoristas modelo matemático Archetti.*

4. **<u>Desabastecimiento de los minoritas:</u>** Esta restricción asegura que en cada momento t los minoristas posean un nivel de inventario mayor o igual que 0.

$$I_{st} \geq 0 \qquad s \in \mathcal{M} \quad t \in \mathcal{T}'. \qquad (5)$$

Imagen 9. *Restricción desabastecimiento modelo matemático Archetti.*

5. **<u>Cantidad de pedido:</u>** Estas restricciones garantizan que para cada periodo t se debe cumplir con la demanda del minorista S.

$$x_{st} \geq U_s z_{st} - I_{st} \qquad s \in \mathcal{M} \quad t \in \mathcal{T} \tag{6}$$

$$x_{st} \leq U_s - I_{st} \qquad s \in \mathcal{M} \quad t \in \mathcal{T} \tag{7}$$

$$x_{st} \leq U_s z_{st} \qquad s \in \mathcal{M} \quad t \in \mathcal{T}. \tag{8}$$

Imagen 10. *Restricción cantidad de pedido modelo matemático Archetti.*

6. **<u>Capacidad del camión:</u>** Esta restricción asegura que la cantidad cargada en el camión o los camiones para abastecer a los minoristas en cada momento t, no exceda la capacidad máxima del vehículo.

$$\sum_{s \in \mathcal{M}} x_{st} \leq C \qquad t \in \mathcal{T}. \tag{9}$$

Imagen 11. *Restricción capacidad camión modelo matemático Archetti.*

7. **<u>Enrutamiento:</u>** Estas restricciones garantizan que para cada momento t ∈ T se determine una ruta viable para visitar a todos los minoristas atendidos en el momento t. Estas restricciones se construyen de la siguiente manera:

   a. Si se debe atender a un minorista es necesario que una variable binaria Zot se active indicando que el minorita S en el periodo t debe ser visitado.

$$\sum_{s \in \mathcal{M}} x_{st} \leq C z_{0t} \qquad t \in \mathcal{T}. \tag{10}$$

Imagen 12. *Restricción enrutamiento modelo matemático Archetti.*

   b. Si al menos un minorista debe ser atendido en el momento t, entonces se debe asegurar que la ruta pase por el minorista S.

$$\sum_{j \in \mathcal{M}', j<i} y_{ij}^t + \sum_{j \in \mathcal{M}', j>i} y_{ji}^t = 2 z_{it} \qquad i \in \mathcal{M}' \quad t \in \mathcal{T}. \tag{11}$$

Imagen 13. *Restricción segunda de enrutamiento modelo matemático Archetti.*

   c. **<u>Restricciones de eliminación de subtours</u>** (ver Fischetti, Salazar-Gonzalez y Toth, 1998, y Gendreau, Laporte y Semet, 1998):

$$\sum_{i \in \mathcal{S}} \sum_{j \in \mathcal{S}, j<i} y_{ij}^t \leq \sum_{i \in \mathcal{S}} z_{it} - z_{kt} \qquad \mathcal{S} \subseteq \mathcal{M} \quad t \in \mathcal{T} \tag{12}$$

for some $k \in \mathcal{S}$.

Imagen 14. *Restricción tercera de enrutamiento modelo matemático Archetti.*

8. **Restricciones de no negatividad:**

$$x_{st} \geq 0 \qquad s \in \mathcal{M} \quad t \in \mathcal{T}. \tag{13}$$

$$y_{ij}^{t} \in \{0,1\} \qquad i \in \mathcal{M} \quad j \in \mathcal{M}, j < i \quad t \in \mathcal{T}. \tag{14}$$

$$y_{i0}^{t} \in \{0,1,2\} \qquad i \in \mathcal{M} \quad t \in \mathcal{T}. \tag{15}$$

$$z_{it} \in \{0,1\} \qquad i \in \mathcal{M}' \qquad t \in \mathcal{T}. \tag{16}$$

Imagen 15. *Restricciones no negatividad modelo matemático Archetti.*

Teniendo en cuenta el modelo matemático descrito anteriormente, se programó una metaheurística que encontrará soluciones con niveles de inventario variables, pero adecuados para cada minorista y para cada proveedor en cada uno de los diferentes periodos, disminuyendo considerablemente los costos asociados al nivel de inventario, es decir, no se tiene en cuenta una política de inventarios dado que se busca realizar combinaciones de envíos de producto entre proveedor y minoristas de forma tal que la demanda se satisfaga completamente en el periodo exacto donde el minorista lo requiera.

La heurística que resuelve el problema de optimización de ruteo con inventarios está basada en tres métodos principales: Método de obtención de solución inicial, búsqueda local y perturbación de los resultados obtenidos. Adicionalmente, para cada uno de estos aspectos principales se ejecuta un algoritmo de ruteo por inserción para finalmente obtener una buena solución en tiempos mínimos de ejecución.

Para poder inicializar la heurística se programó el método de la obtención de la solución inicial, este método básicamente genera aleatoriamente la asignación de clientes para cada una de las bodegas disponibles, luego determina el envío de productos para cada cliente y de forma aleatoria decide en qué periodo se va a enviar los productos de forma parcial o completa. Finalmente se apoya en el método de ruteo por inserción para definir cuáles van a ser las rutas que se tomarán para solucionar el problema. Cabe resaltar que este método para iniciar de forma aleatoria el aplicativo mantiene el procedimiento dentro de las restricciones propias del problema y se ajusta dependiendo del número de clientes, bodegas y vehículos.

Procedimiento heurístico general

El algoritmo heurístico general está constituido de la siguiente forma. Para cada iteración hasta llegar a un parámetro general N1 se genera una solución inicial, a esta solución inicial se le realiza una búsqueda local y se inicializa un parámetro general j en 0, a continuación, se realiza un segundo ciclo hasta que el parámetro j llegue a ser igual a otro parámetro general N2, este ciclo comprende la perturbación de la solución predominante y luego una búsqueda local a esta solución perturbada. Si la función objetivo de la solución predominante es mayor que la función objetivo de la solución perturbada, entonces la solución perturbada pasa a ser la solución predominante y el parámetro j vuelve a ser 0 y en caso de que la función objetivo de la solución predominante sea menor a la solución perturbada se le suma una unidad al parámetro j. Al finalizar, el ciclo interno se guarda la mejor solución encontrada entre las predominantes y las soluciones perturbadas y se reinicia el ciclo general.

Para entender la manera como opera la heurística propuesta se realizó un pseudocódigo el cual se presenta en el documento Anexo 4.

Basados en la definición de Aitana Vidal Esmorís en el documento denominado ”Algoritmos heurísticos en optimización” (2013), La búsqueda local es un método metaheurístico que busca dar solución a problemas de optimización computacionalmente complejos como lo es el IRP. Los algoritmos de búsqueda local hacen uso de problemas que tratan de encontrar una solución óptima ente varias soluciones encontradas. Estos algoritmos se desplazan entre las soluciones a través de cambios locales, hasta encontrar la mejor solución en un plazo de tiempo

establecido o el punto óptimo cuando el tiempo máximo tiende a infinito. La búsqueda local es un algoritmo que tiene la posibilidad de devolver una solución factible incluso si se interrumpe en cualquier momento antes de su finalización.

En adición, la búsqueda local es un método metaheurístico que se comporta de manera eficiente para el desarrollo de solución del IRP, por este motivo existen numerosos documentos en donde diferentes autores hacen uso de esta metaheurística para darle solución a este problema. En el documento llamado ”An integrated local search metod for inventory and routing desicions” escrito por Emmanoul Zachariadis, Christos Tarantilis y Chris Kiranoudis (2009) se evidencia como los autores diseñaron dos operadores de búsqueda local innovadores para tratar eficazmente las características del IRP denominados inserción y extracción, el primer operador inserta un nuevo punto de reabastecimiento en el horario de un cliente, mientras que el segundo consiste en eliminar una visita de reposición.

Otro documento que resalta el desarrollo de la búsqueda local para dar solución al Inventory Routing Problem es el de Lei Qin, Lixin Miao, Qingfang Ruan y Ying Zhang al cual denominaron “A local search method for periodic inventory routing problem“ (2014), en donde proponen un método de búsqueda local basado en cuatro operadores(inserción, eliminación, adición y una combinación de estos) para dar solución al problema de inventario. Así mismo, En el documento “A hybrid heuristic based on iteraded local search for multivehicle inventory routing problem” elaborado por Edcarllos Santos, Luiz Satoru Luidi Simonetti y Pedro Gonzalez (2016), denotan la búsqueda local iterada, la cual es un método de optimización que explora un espacio de soluciones por perturbaciones sucesivas de óptimos locales, en donde estas soluciones se obtienen a partir de la fase de búsqueda local inicial. Los autores desarrollaron cuatro estructuras para desarrollar un análisis de decisiones de enrutamiento y de inventario de forma simultánea, ejecutan la búsqueda local a través de sucesivas aplicaciones de las características del problema. Finalmente, muestran los resultados basados en una estructura de eficiencia.

Como se ve reflejado en los documentos anteriores y en los resultados de la literatura actual, la búsqueda local iterada junto con la búsqueda local, son métodos metaheurísticos de optimización acordes para la solución del IRP (Inventory Routing Problem). Para el presente trabajo se implementó la búsqueda local iterada propuesta por William J. Guerrero Rueda en la tesis de doctorado de ingeniería “Modelos y métodos de optimización para el problema de localización y ruteo de inventarios” la cual se rige por el pseudocódigo ilustrado en la imagen 16.

Algorithm 4 MS-ILS for MD-IRP

```
1:  S* ← ∅ ;
2:  for i ← 1 to N1 do
3:      S_0 ← Initial_Sol;
4:      S_0 ← LS(S_0);
5:      j := 0;
6:      repeat
7:          S' ← Perturbation ( S_0 );
8:          S' ← LS(S');
9:          if f(S_0) > f(S') then
10:             S_0 ← S';
11:             j := 0;
12:         else
13:             j := j + 1;
14:         end if
15:     until j = N2
16:     S* ← Save_Best(S_0, S*);
17: end for
18: Return S*;
```

Imagen 16. *Pseudocódigo de ILS por William Rueda*

A partir de las contribuciones de los autores Qin, et al. (2014) y Zachariadis, et al. (2009) dentro de la búsqueda local en las soluciones de problemas de ruteo con inventarios, con respecto al operador de búsqueda local propuesto se combinan dos tipos de vecindarios, el primero en las decisiones de inventario y el segundo en las decisiones de asignación. Se exploran primero en las decisiones de asignación de clientes a bodegas asignándolo aleatoriamente, luego se evalúan las decisiones de inventario para posteriormente realizar el ruteo mediante una heurística de inserción y evaluar la solución. Dentro de la Búsqueda local en las decisiones de asignación, se exploran dos vecindarios, el primero, selecciona un minorista y lo reubica en un depósito diferente; luego, selecciona dos minoristas asignados a diferentes depósitos y los intercambia en su asignación de depósito. Esto se hace para realizar una exploración más amplia en la solución predominante. Luego, en la Búsqueda local en las decisiones de inventario se explora solamente

un vecindario, en el cual para un minorista aleatorio, es posible eliminar una visita en un periodo aleatorio. Esto requiere aumentar el nivel de existencias en la visita programada anterior y reducir el costo de ruteo en el periodo.

El operador de búsqueda local se presenta en el anexo 4.

Operador de perturbación

El objetivo de la perturbación es evitar los óptimos locales para ello se propone que un único minorista se reasigne a un depósito diferente, luego en un cliente aleatorio se programa una nueva visita en un periodo aleatorio, para posteriormente realizar el ruteo a la solución encontrada por medio de una heurística de mejor inserción.

El operador de perturbación se presenta en el Anexo 4.

Según la Oficina de Coordinación de Simulación y Modelado (2001), la cual se encarga de la administración de los asuntos concernientes a la informática dentro del departamento de defensa de los Estados Unidos, sostiene que existe un procedimiento llamado Verificación, Validación y Acreditación (VV&A) dentro de cualquier aplicación de un modelo o simulación, asegurando que los resultados obtenidos sean apropiados para su finalidad.

Por un lado, de acuerdo con esta institución, la verificación corresponde al proceso por el cual un determinado modelo o simulación representa de manera adecuada la descripción y las especificaciones del aplicativo por parte de sus desarrolladores, lo que quiere decir que el sistema ejecuta las acciones que sustenta al consumidor y no hay incongruencias entre su definición y los procedimientos ejecutados.

En este caso, se sabe que el algoritmo inicializa todas las variables requeridas y debe obtener una solución inicial mediante la pre-asignación de depósitos y clientes por medio del porcentaje de participación de cada bodega, teniendo en cuenta la capacidad de cada centro de abastecimiento y la factibilidad de atender a los usuarios dependiendo de la demanda en cada periodo; para posteriormente convocar la heurística encargada de realizar la búsqueda local, que tiene como objetivo perturbar la solución inicial seleccionando una bodega y ubicándola de manera aleatoria en el orden establecido, encargándose de ejecutar los cambios bajo las restricciones dadas y buscando una mejor solución con respecto a la función objetivo. Finalmente, este proceso se efectúa tantas veces como el desarrollador lo programe y el algoritmo debe ser capaz de manejar sus rutinas dentro del tiempo establecido.

Luego de tener toda la heurística en correcto funcionamiento, se procede a revisar mediante inspecciones dentro del programa Visual Basic for Applications cada uno de los procedimientos de todos los módulos junto con los valores que adquieren las variables tanto en los inventarios de los depósitos y clientes, como en las rutas generadas por periodo; de esta manera se asegura que el aplicativo realice correctamente los procesos designados y que los resultados sean a la operación.

Por otro lado, se encuentra el procedimiento de validación, el cual bajo los estándares del Departamento de Defensa de los Estados Unidos (DOD), corresponde al proceso por el cual se determina el grado de representación del mundo real desde la perspectiva final del modelo junto con sus respectivos datos de entrada y de salida. Con lo anterior, se busca asegurar que los métodos y resultados del modelo se aproximen a la realidad. De esta manera, se compararon los resultados obtenidos en la literatura tanto de Archetti et al, (2007), como de Vargas et al, (2014), donde por medio de sus modelos MIP que aseguraban el punto óptimo se contrastaron las soluciones a las instancias que convergían en todos los programas. Con esto, se confrontaron tres funciones objetivo, donde dos fueron extraídas de la literatura y una era adquirida de nuestro Inventory Routing Solver, para cada una de las instancias que coincidían en todos los modelos las cuales son 40 instancias con múltiples depósitos, cantidad de vehículos variada, demanda y distancias previamente generadas aleatoriamente.

Finalmente se calibró el modelo, donde se evaluaban determinadas instancias varias veces variando su tiempo de procesamiento en intervalos. El proceso realizado tenía como objetivo determinar en qué límite de tiempo de procesamiento convergían las funciones objetivo, es decir, definir un tiempo máximo donde la mejora en la calidad de las soluciones no era significativa y por lo tanto a determinada duración se obtenía un resultado acorde. Con esto, se determinó que el tiempo máximo debía ser variable dada la cantidad de usuarios del problema, de esta manera no

se desperdiciaba el recurso buscando soluciones en problemas cortos y se proporcionaba más tiempo de computo a instancias que lo sí demandaban.

El complemento Inventory Routing Solver requirió para su desarrollo en primer lugar una estructura general que agrupara una metaheurística y una interfaz de usuario. El desarrollo de este proyecto se adecua a las normas y estándares de seguridad de la información definida en la norma técnica colombiana NTC-ISO/IEC 27001 y evaluación de software definida en la norma ISO 9126. Antes de iniciar con el desarrollo de la interfaz se establecieron los estándares de calidad que recomienda la norma con el fin de cumplir con todos los requisitos internacionales del software:

- Funcionalidad: Esta herramienta contiene un grupo de atributos que permiten determinar si un software cuenta con un número de funciones que satisfagan las actividades para las que fue diseñado. Por esta razón se establecieron los siguientes atributos:
- Adecuación: Permite evaluar si el software cuenta con las funciones para realizar las actividades que fueron establecidas en su definición.
- Exactitud: Evalúa el resultado final que obtiene el software y determina si tiene consistencia a lo que se espera de él.
- Interoperabilidad: Consiste en evaluar si el software puede interactuar con un sistema independiente.
- Conformidad: Determina si el software se adhiere a estándares o regulaciones que existan.
- Seguridad: Impide el acceso de personal no autorizado de forma accidental o predeterminada a la información del software.

En el desarrollo del IRS se buscó que el software cumpliera con todas las funciones necesarias para este caso específico: inventarios y ruteo, esto con el fin de garantizar resultados exactos como se ilustra en la imagen 18. Además, se estableció un plan de seguridad de la información para que los datos de los clientes se mantengan bajo confidencialidad como se muestra en el Imagen 17.

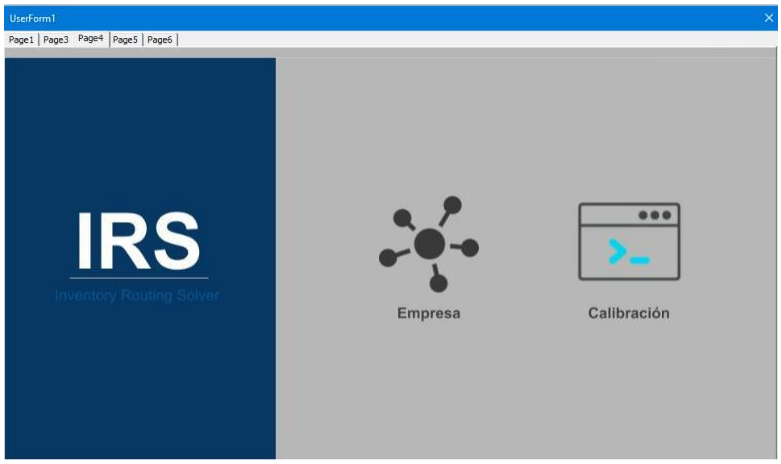


Imagen 17. *Ramas del IRS*

	CANTIDAD
BODEGAS	1
CLIENTES	4
PERIODOS	3

	VEHICULOS
CANTIDAD	1
COSTO	
CAPACIDAD	817

IRS

BODEGAS					
NOMBRE	DIRECCION	CAPACIDAD	COSTO DE MANTENER	INVENTARIO INICIAL	
	universidad javeriana bogota	3954	0,3	1318	## 1

CLIENTES							
NOMBRE	CAPACIDAD	DIRECCION	COSTO MANTENER	INVENTARIO INICIAL	DEMANDA EN EL PERIODO		
					1	2	3
	225	farmatodo cra	0,35	150	75	75	75
	54	calle 134 #9-2	0,14	36	18	18	18
	144	centro comerc	0,17	72	72	72	72
	150	Calle 185 #56-	0,36	100	50	50	50

Imagen 18. *Ramas del IRS*

- Confiabilidad: Se contemplan una serie de atributos los cuales determinan la capacidad del software de mantener el nivel de ejecución durante un tiempo establecido y bajo un conjunto de condiciones definidas. Las características que el estándar sugiere son:
- Nivel de madurez: Establece la frecuencia de falla por errores del software.
- Tolerancia a fallas: Se refiere a la capacidad de mantener el desarrollo del software en caso de una falla.
- Recuperación: Verifica si el software puede reasumir el funcionamiento después de un fallo, así como la restauración de datos perdidos.

El desarrollo del software fundamentado en la herramienta Visual Basic for Applications como se evidencia en la imagen 9, permite al usuario la estabilidad de la información que maneja, esto quiere decir que sus datos no se perderán y podrá acceder a estos cuando desee.

- Usabilidad: Está formado por un grupo de atributos de permiten evaluar el esfuerzo que debe realizar un usuario en el momento de utilizar el software. Estos atributos se dividen en:
- Comprensibilidad: Se refiere al esfuerzo ejecutado por los usuarios para reconocer la estructura lógica del software.
- Facilidad de aprender: Establece el esfuerzo que los usuarios deben hacer para aprender a utilizar el software o la aplicación.
- Operabilidad: Reúne los conceptos que evalúan la operación y el control del sistema.

El uso del software por parte del cliente es intuitivo con el fin de facilitar su aplicación. Por esta razón se realiza un interfaz, como se ilustra en las imágenes 17, 18 y 19, que tenga conexión con el programa y sobre todo que sea amigable con el cliente, en donde este sea capaz de comprender y ejecutar las tareas necesarias para la obtención de resultados.

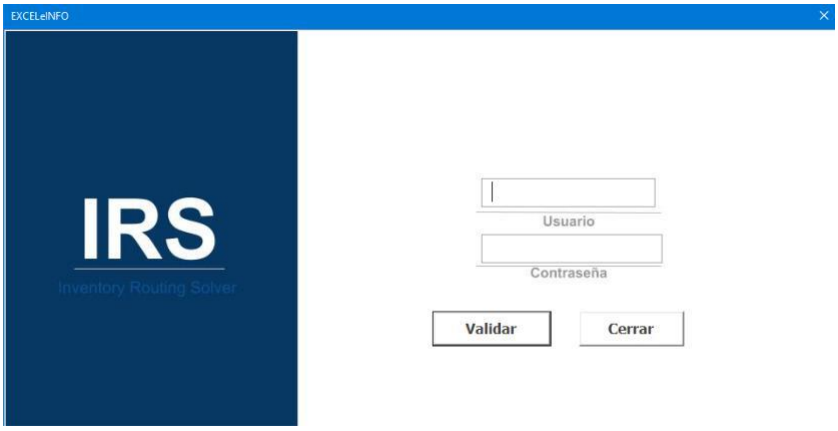


Imagen 19. *Ramas del IRS*

- Eficiencia: Esta característica permite determinar la relación entre el nivel de funcionamiento del software y el número de recursos usados. Los aspectos a evaluar son:
- Comportamiento con respecto al tiempo: Se establecen los atributos del programa relativos a los tiempos de respuesta y procesamiento de datos.
- Comportamiento con respecto a recursos humanos: Establece la cantidad de recursos usados y la duración de su uso en la relación de sus funciones.

Se ejecutó el IRS en un computador doméstico, con restricciones de memoria y procesamiento, sin embargo, como se evidencia en el numeral de resultados, se presentan soluciones cercanas al óptimo en tiempos muy inferiores a los arrojados en el método exacto. A continuación, se resalta el comportamiento de los resultados y su tiempo de ejecución en el grafico 5, 6 y 7.

- Mantenibilidad: Establece los atributos para medir el esfuerzo para realizar modificaciones al software, ya sea por corrección de posibles errores o un aumento de la funcionalidad. Se tiene los siguientes factores:

- Capacidad de análisis: determina el esfuerzo para diagnosticar deficiencias o causas de fallas, identifica las partes que deben ser corregidas.
- Capacidad de modificación: Mide el esfuerzo para realizar cambios en los diferentes aspectos del software con el fin de que funciones de forma adecuada
- Estabilidad: Permite evaluar los riesgos de efectos inesperados debido a posibles modificaciones realizadas al software
- Facilidad de prueba: Es el esfuerzo para validar el software en el momento posterior al que este ha sido modificado.

Se Realizó el software de una forma organizada y estable, en donde se efectúen diferentes subrutinas con el fin de disminuir la complejidad en un posible cambio o corrección de código. Así mismo, la plataforma Excel permite identificar las fallas que se puedan presentar en el momento de la ejecución del programa, evidenciando la parte del código que presenta el error, permitiendo una fácil identificación de fallas en el código. A continuación, se ilustra en la imagen 20 un ejemplo de la organización del código a través de subrutinas, además en la imagen 21 se evidencia la forma en que la plataforma Excel identifica posibles errores en el momento de la ejecución del programa.

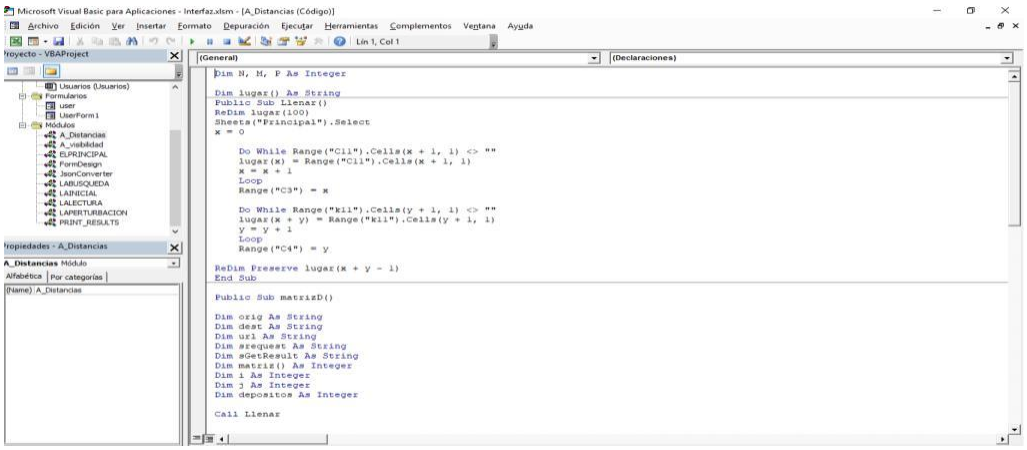


Imagen 20. *Software en subrutinas. Elaboración propia.*

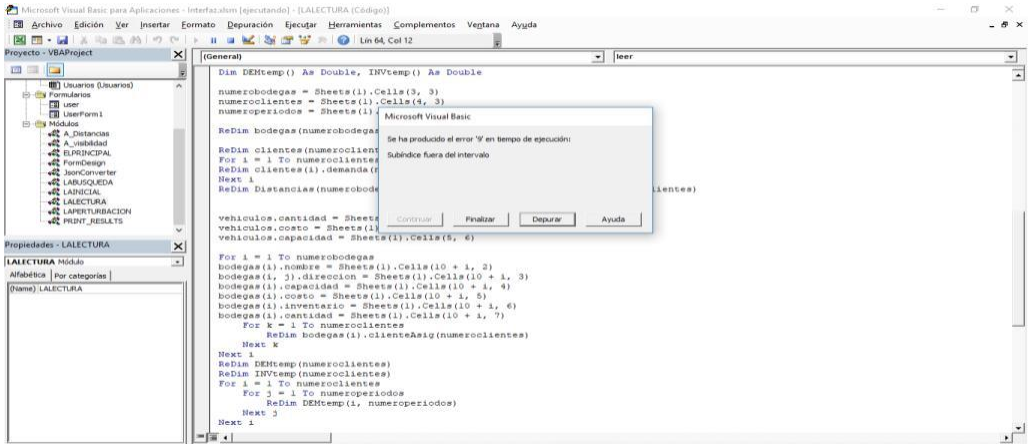


Imagen 21. *Identificación de error del software. Elaboración propia*

- Portabilidad: Se refiere a la capacidad del programa a ser transferido de un ambiente a otro. Considera los siguientes aspectos:
- Adaptabilidad: Evalúa la oportunidad para adaptar el programa a diferentes ambientes sin tener que ser modificado.

- Facilidad de instalación: Determina el esfuerzo para instalar el programa o el software en un ambiente determinado.
- Conformidad: Permite evaluar si el software se adhiere a estándares o convenciones relativas a la portabilidad.
- Capacidad de reemplazo: Es el esfuerzo usado para sustituir el programa por otro producto con funciones similares.

El software desarrollado es adaptable a cualquier empresa o compañía ya sea pequeña, mediana o grande, que tenga un problema de tipo IRP ("Inventory Routing Problem") y cumpla con las características del problema. Así mismo, se realizó el IRS en la plataforma Excel la cual facilita el intercambio de datos y soluciones a otras plataformas en caso de que se requiera.

Con respecto a la Interfaz de Usuario Gráfica GUI, la cual según Luna (2004), corresponde a todos los elementos gráficos que ayudan a comunicar al hombre con un sistema o máquina, se empleó la función de Excel User Forms para ambientar la herramienta que introdujera sus funcionalidades al cliente, se dividiera en dos apartados principales, obtuviera los resultados arrojados por la metaheurística y estos sean mostrados de manera comprensible.

Posteriormente, el cliente cargará el archivo con toda la información requerida a la plataforma interactiva y se ejecutarán todos los subprocesos para inicializar la metaheurística. Con el fin de obtener una aproximación a la realidad con respecto a las distancias entre el cliente y el proveedor, se insertaron las funcionalidades de Google Maps al algoritmo en Microsoft Excel para garantizar la georreferenciación de las localizaciones entre nodos y de esta manera adquirir las distancias (ruta más corta entre los puntos), lo que significa que al momento de ejecutar esta acción, el programa obtendrá los datos del momento, teniendo en cuenta el tráfico actual, las retenciones y los posibles retrasos en la vía, dando como resultado información actualizada y no información histórica, la cual podría afectar significativamente la operación.

Por otro lado, se creó el apartado calibración, el cual tiene como función ejecutar de manera sistemática total o parcialmente las instancias obtenidas de la literatura por parte de Archetti et al, (2007); en esta división en la interfaz de usuario se definen el número de perturbaciones de la metaheurística, el tiempo de procesamiento de cada instancia y el límite de tiempo de procesamiento. Con esto, se obtienen todos los resultados de los archivos contemplados y se evalúa su comportamiento con el fin de ajustar los parámetros del algoritmo en función de las soluciones arrojadas. Es importante mencionar que las instancias consideradas para la calibración del programa fueron halladas en archivos planos con extensión (.dat), por lo que previo al desarrollo de la interfaz de usuario se modificaron estos archivos de manera metódica para traducir la información en la plantilla destinada para la ejecución del algoritmo, siendo esta un registro con extensión editable en Microsoft Excel (.xlsx), mismo patrón utilizado para el apartado empresa.

5. Diseño de la interfaz

Una de las consideraciones más importantes a tener en cuenta es el desarrollo de la interfaz, como ya se mencionó anteriormente, en la actualidad no basta con tener un programa o software que haga lo que se pide de una forma adecuada, es necesario que este sea amigable, entendible y cómodo para los usuarios. Por tal razón, el desarrollo de la interfaz para el aplicativo está basada en la practicidad ofrecida al cliente por medio de Visual Basic for Applications. Esta interfaz cuenta con dos ramas o líneas de ejecución:

La primera rama recopila los datos de entrada a través de la opción Empresa, la cual permite por medio del diligenciamiento de un formato en un libro de Excel el ingreso de los datos, para posteriormente cargarlos al aplicativo y darle continuidad a la ejecución y presentación de resultados; la otra rama o apartado de la interfaz, está encargada de realizar la calibración de las distintas instancias que se tomaron de la literatura tanto de Archetti et al, (2007), como de Vargas et al, (2014).

Inicialmente se diseñaron los formularios en Visual Basic Excel de una forma gráfica en donde el usuario cuente con indicaciones claras y pueda manejar la interfaz intuitivamente sin ningún problema. La interfaz inicia el software como se observa en la imagen 19 con un sistema de seguridad de la información el cual requiere un usuario y

contraseña para acceder al programa. Para fines académicos se podrá acceder al Inventory Routing Solver mediante el Usuario: admin y la contraseña: admin en los campos requeridos.

Una vez el programa ha validado los datos de usuario y contraseña del cliente o del administrador, este lo dirige a un formato en donde dependiendo del usuario le permitirá escoger una de las dos opciones de ejecución del programa, como lo son empresa y calibración, ilustrado por la imagen 17. Dando cumplimiento a la norma técnica colombiana NTC-ISO/IEC 27001.

Posterior a esto, en caso de que el usuario sea un cliente, la interfaz lo dirigirá a un formato en donde este podrá diligenciar una plantilla que se encargará de recopilar los datos de entrada como lo son: número de vehículos, número de depósitos, número de clientes y sus datos específicos como direcciones de cada punto de interés y los costos asociados como se muestra en la imagen 18.

Una vez terminado el diligenciamiento de la plantilla con todos los parámetros necesarios, se procede a obtener la matriz de distancias por medio de la Interfaz de Programación de Aplicaciones API de Google, con el fin de extraer datos en tiempo real sobre las distancias que serán recorridas por los vehículos asignados. Este módulo funciona a través de una clave API la cual establece una conexión con Google Maps en donde esta plataforma devuelve los datos de distancia y tiempo de recorrido reales teniendo en cuenta diferentes variables como lo son tráfico, accidentes y retrasos. Esto le permite al cliente contar con datos actualizados y no históricos que puedan alterar la eficiencia del programa. Con lo anterior, el cliente debe encargarse de ingresar correctamente las direcciones de origen y destino con el fin de lograr mejores resultados que se aproximen a la realidad y debe validar si la información recibida es correcta.

Como se mencionó anteriormente, todos los formatos de la interfaz se diseñaron de una forma en la cual el cliente cuente con instrucciones de desarrollo, lo cual le permita el uso fácil y práctico de esta. En adición a esto, la interfaz cuenta con un módulo interno para el desarrollo de la matriz de distancias.

Este módulo que se desarrolló está encargado de ejecutar la matriz de distancias entre los depósitos y los clientes recopilados en la plantilla que fue cargada por el usuario. Una vez la interfaz ha recopilado los datos de entrada y ha ejecutado la matriz de distancias, esta procede a enviar la información al libro de Excel donde se ejecutará el software; con los parámetros obtenidos anteriormente en la pantalla principal del aplicativo se espera que se desarrolle la metaheurística que minimice los costos asociados al transporte y manejo de inventarios.

Luego de un tiempo determinado, en la hoja la de *Resultados* se podrá visualizar la información acerca de la mejor ruta de distribución, los niveles de inventarios propios necesarios, tiempos estimados para la entrega y la función objetivo total y dividida entre costos de inventario y ruteo tanto de los centros de distribución como de los clientes.

Acto seguido, como lo indica la imagen 22 la rama de *Calibración* se encarga de ejecutar sistemáticamente las instancias obtenidas de la literatura por medio de un parámetro establecido por el desarrollador. Esto quiere decir que el usuario tiene la posibilidad de evaluar diferentes escenarios dada la naturaleza de las instancias ya sea por cantidad de vehículos, número de depósitos o la suma de los clientes. Por ejemplo, si se especifica “1abs”, el programa únicamente tendrá en cuenta 47 de las 711 instancias obtenidas en la literatura que corresponden a las instancias de un solo centro de abastecimiento

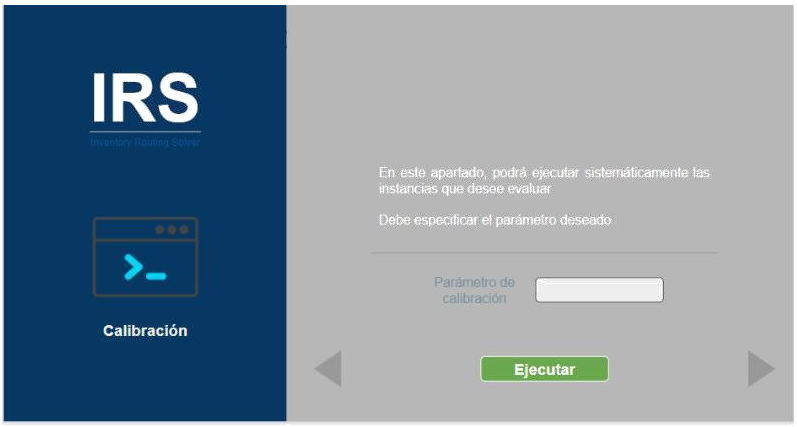


Imagen 22. *Apartado Calibración del IRS*

5.1 Características detalladas del problema

El Inventory Routing Solver está basado en el modelo matemático descrito por Archetti en el 2007, por tal razón posee las siguientes características específicas del problema:

- El horizonte de planeación del ruteo con inventarios es finito y discreto.
- Las bodegas tienen capacidad limitada en cada periodo.
- La demanda de los minoristas es conocida en cada periodo.
- Los camiones tienen capacidad limitada.
- La flota de camiones tiene un tamaño fijo.
- Los niveles de inventario iniciales para los minoristas y el proveedor son conocidos.
- El nivel de inventario inicial del minorista es diferente al nivel de inventario final, por tal razón, la demanda del minorista no posee un comportamiento periódico.
- Para los minoristas y el proveedor se cobra un costo por mantener inventario, el cual es un valor fijo durante todo el horizonte de planeación.
- Se poseen en esencia 3 periodos de planeación, pero se tuvo un en cuenta un cuarto periodo con el fin de calcular los costos asociados a las operaciones realizadas al final del periodo 3.
- Los camiones pueden visitar varios minoristas en un solo viaje, en cada periodo.
- Los camiones deben tener la capacidad para distribuir la cantidad total de producto de todos los minoristas que se planean visitar en un solo viaje, en cada periodo.
- El problema consiste en calcular la cantidad de producto a enviar a cada minorista y definir la combinación de minoristas a visitar para cada periodo dentro del horizonte de planeación, satisfaciendo el total de la demanda de cada minorista.

El presente trabajo de grado propone el diseño y construcción de un aplicativo con interfaz gráfica que permita el fácil entendimiento para cualquier persona que desee programar las órdenes y el ruteo de inventarios entre los almacenes y el cliente minorista. El diseño del aplicativo consiste en una pantalla de inicio en el que se especifique aspectos de interés para el modelamiento de ruteo y niveles de inventario como:

- Tipo de cliente minorista
- Nombre del cliente minorista
- Cantidad de stock disponible en el almacén del depósito
- Cantidad de stock disponible en el almacén del minorista
- Nivel mínimo de inventarios en el almacén del minorista
- Cantidad actual de inventario en el almacén propio
- Cantidad demandada por el cliente en periodos de tiempo
- Cantidad de producto a enviar a la bodega en cada periodo
- Cantidad y capacidad de los vehículos

Posteriormente la solución se ilustrará en la hoja de Excel Resultados mediante tablas y gráficos donde se podrá visualizar la información acerca de la mejor ruta de distribución, los niveles de inventarios propios necesarios, tiempos estimados para la entrega y la función objetivo total y dividida entre costos de inventario y ruteo tanto de los centros de distribución como de los clientes.

5.2 Requerimientos de diseño

El desarrollo del aplicativo cumple con los siguientes indicadores y requerimientos con el fin de que pueda ser utilizado de manera comercial y que a su vez se dé a conocer para solucionar y disminuir los problemas relacionados con el ruteo y manejo de inventarios. A continuación, se presentan los requerimientos e indicadores esperados:

- Tiempo de ejecución máximo: El aplicativo recolecta los datos y parámetros enunciados en el anterior punto y posteriormente ejecuta la metaheurística en un tiempo determinado, con el fin de que se logre solucionar el problema en el menor tiempo posible.

- Interfaz de fácil uso: El diseño del aplicativo con ayudas visuales con el fin de que cualquier persona en cualquier momento sepa usarlo y lo use de manera adecuada.

- Flexibilidad a cualquier compañía: Se espera que el aplicativo pueda ser utilizado por cualquier empresa que desee resolver un problema de IRP.

- Soluciones de calidad y configurables: Se espera que las soluciones aportadas por el aplicativo estén dentro de los rangos y restricciones para cada una de las empresas que lo usen y que además en dado caso de necesitar configuración adicional, el software sea intuitivo para que otra persona pueda reprogramarlo.

- Configuración de tiempo y adiciones: Se espera que el software permita al usuario programar varios periodos basándose en los pronósticos realizados por la empresa.

5.3 Restricciones de diseño

Cabe destacar que las complicaciones actuales para el diseño del aplicativo son la confiabilidad de las empresas para compartir sus datos y poder utilizarlos dentro del aplicativo, por lo que esta restricción tiene un enfoque en el pensamiento social actual que a medida que pase el tiempo y haya más confianza para realizar uniones estratégicas puede lograrse mejorar este aspecto. Por otra parte, es importante mencionar que el dinero y el tiempo son factores que impiden que el aplicativo sea un software de primera mano; si como grupo de trabajo se tuviera más tiempo y dinero se podría mejorar el desempeño de la metaheurística y la presentación en el diseño de la aplicación con el fin de vender el aplicativo a gran escala.

Actualmente, es posible establecer que los aplicativos diseñados en Colombia con base en inventarios y ruteo de productos son pocos y por tal razón no hay valores ni personas de referencia para tomar esta información y utilizarla como ayuda para mejorar y ofrecer un mejor aplicativo a las organizaciones.

5.4 Normas y estándares

El instituto colombiano de normas y certificación ICONTEC es el organismo nacional de normalización y creación de estándares que brinda soporte al productor y protección a los consumidores. En el año 2006 la norma técnica colombiana NTC-ISO/IEC 27001 fue ratificada por el consejo directivo 2006-03-22, la cual fue elaborada con el fin de brindar un modelo para el establecimiento, implementación, operación, seguimiento, revisión, mantenimiento y mejora de un sistema de gestión de la información (SGSI)[13]. La norma NTC-ISO/IEC 27001 adopta el modelo de procesos" planificar-hacer-verificar-actuar"(PHVA), permitiéndole establecer los requisitos de entrada de información, adaptando sus procesos a estos, con el fin de garantizar la seguridad de la información en todo momento, es decir, especifica los requisitos de información para implementar controles de seguridad.

[13] Fernández, C. M. La norma ISO 27001 del sistema de gestión de la seguridad de la información. (2012)

Por otro lado, se encuentra la norma ISO 9126 la cual es un estándar internacional para la ingeniería y evaluación de software, posee cuatro pilares dirigidos hacia modelos de calidad, métricas tanto internas como externas y calidad en el uso de las mismas; dentro de la primera parte del estándar se categoriza la calidad del software, donde se determinan sus características para clasificarlo según su funcionalidad, fiabilidad, usabilidad, eficiencia, mantenibilidad y portabilidad; dentro de su segunda y tercera parte se definen las métricas externas e internas respectivamente para la medición cuantitativa de la calidad del software, donde de manera externa se mide el comportamiento del programa y de forma interna se valora el software por sí mismo, para finalmente determinar la calidad en el uso de las métricas por medio de atributos medibles[14].

6. Resultados

Previo a la presentación de resultados, es importante exponer *la calibración* del modelo, ya que de esta se deriva el rendimiento de este junto con todas las soluciones obtenidas. Como primera medida se determinó que el tiempo máximo debía ser variable dada la cantidad de usuarios del problema, ya que de esta manera no se desperdiciaba el recurso buscando soluciones en problemas cortos y se proporcionaba más tiempo de computo a instancias que lo sí requerían. A partir de lo anterior, se evaluaron cuatro instancias diferentes donde el tiempo de ejecución máximo estaba dado por un factor multiplicador, lo que quiere decir que se buscaba un factor de tiempo en segundos, que multiplicada la cantidad de clientes determinaba el tiempo máximo de procesamiento. Se realizó la evaluación para los factores de 1, 5, 10 y 20 segundos donde el tiempo mínimo estaría entre los 5 segundos y el máximo 1000 segundos. Es importante resaltar que este tiempo máximo fue pensado dentro del ámbito empresarial donde un usuario estuviese dispuesto a esperar hasta 20 minutos para obtener una solución. De esta manera, se muestra el gráfico 3 para ilustrar los datos obtenidos. No obstante, en el Anexo 3 se muestran las pruebas con sus respectivas gráficas de desempeño.

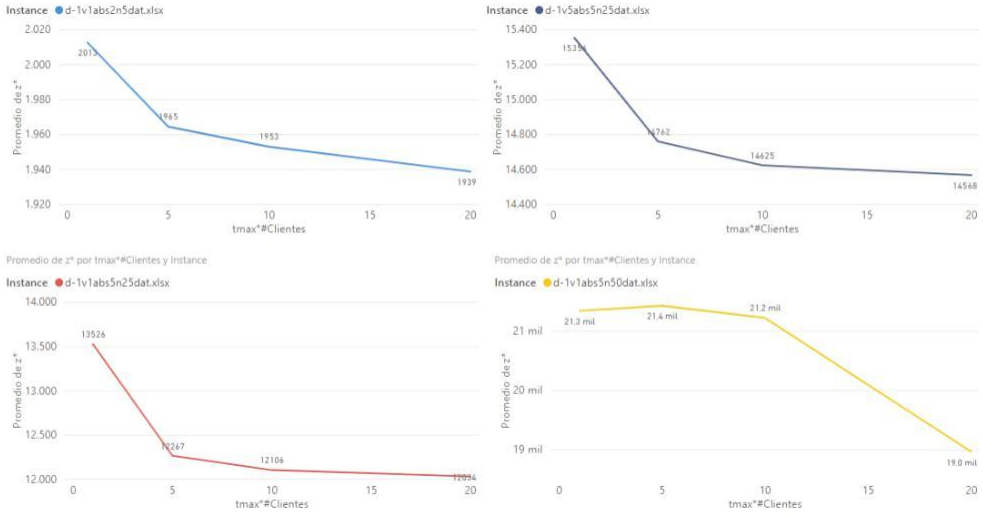


Gráfico 3. *FO para cada multiplicador*

Con lo anterior, tomando como ejemplo la instancia 1v1abs5n25, donde el tiempo inicial para 25 clientes era de 25 segundos y el tiempo final de 500 segundos, se evidencia una mejoría del 11% del total de los costos con respecto a la función objetivo. Sin embargo, la reducción de costos entre el multiplicador de 10s y 20s es de solo 0.59%, lo que quiere decir que a partir del factor 10s la función objetivo se estabiliza. Con esto, se define que el tiempo de ejecución máximo está dado por la siguiente fórmula:

$$=20 * \#$$

[14] ISO/IEC TR 9126. Software Engineering Product quality: Internal metrics. (2003)

Con el tiempo máximo de ejecución, se procede a exponer los resultados obtenidos comparados con las soluciones ofrecidas por la literatura, donde se demuestra que el algoritmo cumple con los requerimientos de desempeño, proporcionando cotas superiores o *Upper bounds* a los problemas en cuestión. Esta aclaración se destaca ya que si se hubiesen extraído funciones objetivo menores a las alcanzadas por los métodos exactos de referencia, se estaría diciendo que se encontraron resultados inferiores a los óptimos, lo cual no tiene sentido ya que estos modelos lineales exploran todo el espectro de posibilidades para encontrar la mejor solución, diferente al Inventory Routing Solver, que mediante la búsqueda local iterada mejora las soluciones encontradas acercándose al punto óptimo sin garantizarlo.

Con el fin de evaluar la efectividad de cómputo y la eficiencia del algoritmo dentro de esta investigación se empleó el programa Microsoft Excel y el lenguaje VBA (Visual Basic for Applications) para ejecutar la metaheurística expuesta en la sección de metodología. El algoritmo fue ejecutado en un procesador Intel (R) Core i5, CPU de 2.30 GHz y 8 GB de RAM. Sin embargo, es importante mencionar que Coelho et al. (2013) obtuvo resultados de las instancias evaluadas en el documento a partir un Procesador Intel Xeon™ de 2.66 GHz, y memoria RAM superior a 48 GB de RAM, lo que supone el uso de un HPC (High Performance Computing). Con lo anterior, se deduce que la calidad de las soluciones se encuentra influenciada por el tipo de máquina empleado.

En el Gráfico 4 Se muestran los resultados obtenidos tanto de Coelho et al. (2013) como los del IRS y el respectivo Gap o brecha entre ambas soluciones:

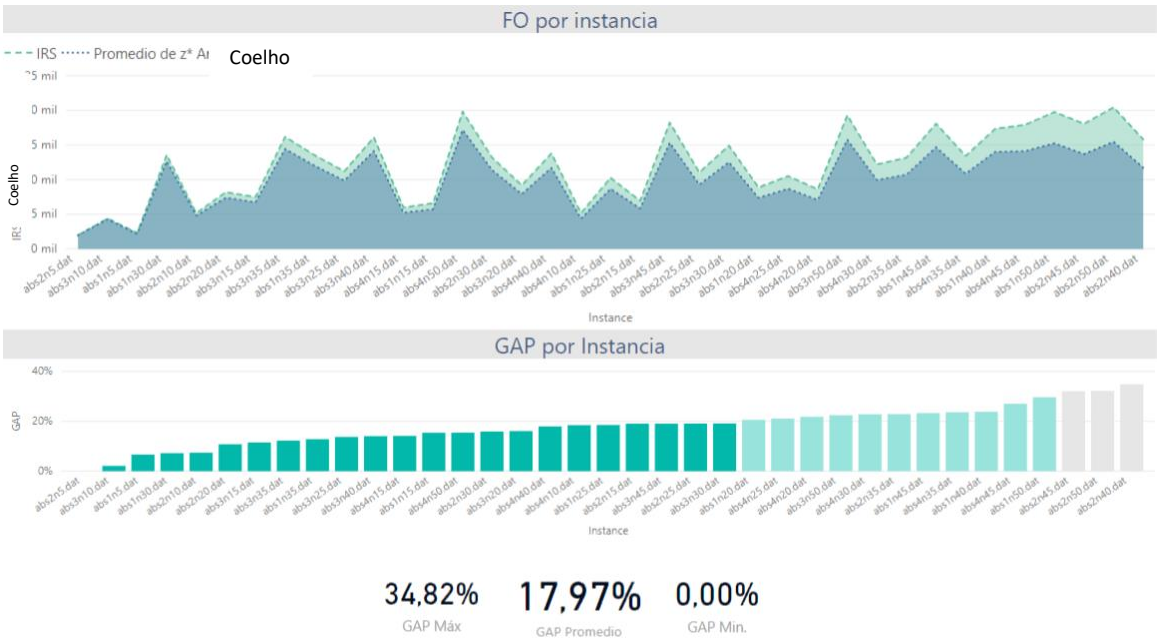


Gráfico 4. *Contraste de resultados*

Como se puede observar en la gráfica 6, se expone la función objetivo alcanzada por Archetti (2007) y por el IRS para cada una de las instancias, donde se obtuvo una diferencia porcentual media (GAP) del 17.97% con respecto a las soluciones arrojadas por el método exacto del modelo de programación lineal.

Sin embargo, con el fin de profundizar en los resultados obtenidos, se dividieron las instancias en cinco grupos principales dada la cantidad de clientes (5, 10, 15, 20, 25, 30, 35, 40, 45, 50) ya que este es un factor que torna el objeto de estudio más complicado a medida que aumenta su cantidad.

Grupo	Cantidad instancias	Mejor GAP	GAP / Grupo
5-10	2	0,00%	3,33%
15-20	7	2,13%	12,59%
25-30	8	13,73%	17,71%
35-40	8	7,23%	17,08%
45-50	12	14,04%	24,31%

Gráfico 5. Resultados IRS por grupo

En consecuencia, por medio de la tabla del gráfico 5 se determinó que el mejor GAP dentro de todos los grupos no supera el 15% de las funciones objetivo óptimas y que para instancias con una cantidad moderada de usuarios, la calidad de las soluciones se acercaba al punto óptimo, en comparación con instancias más grandes donde el límite superior se alejaba con respecto al incremento del número de clientes que supone el problema en un 24% en promedio.

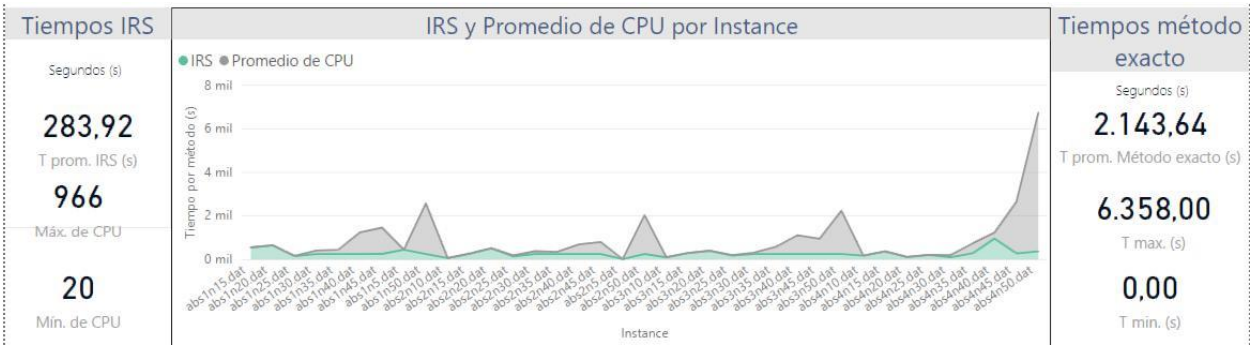


Gráfico 6. *Contraste de tiempos de ejecución*

En adición, con respecto a las soluciones obtenidas por el IRS, se encontró que el tiempo promedio de ejecución corresponde al 13% del método exacto, lo que quiere decir que se pasó de obtener soluciones óptimas en 2143 segundos, a extraer resultados 18% por encima de la mejor función objetivo por medio de un modelo metaheurístico, en un promedio de 283 segundos. Por otro lado, se evidenció que el 35% del total de las instancias evaluadas por el modelo de programación lineal tardó 1300 segundos más en promedio para obtener una solución, en comparación al Inventory Routing Solver. Esto supone una mejoría significativa en tiempos de ejecución contrastando el algoritmo desarrollado junto con un modelo de programación lineal.

Finalmente, en el Anexo 2. se presentan otros resultados obtenidos por medio del Inventory Routing Solver que proporcionarán a la literatura cotas superiores o Upper Bounds.

Actualmente en el mercado existen diversas aplicaciones computacionales encargadas de operaciones logísticas en el ámbito empresarial, por ejemplo Cloudinn ® es una herramienta que administra el flujo de compras e inventario, también existe el software RouteStar ® que ejecuta el seguimiento de tareas, gestión de contactos y optimización del ruteo de vehículos o ABC Inventory ® que administra el inventario de múltiples depósitos; como se puede observar existen diferentes herramientas organizacionales. Sin embargo, en la búsqueda realizada no hay evidencia de un aplicativo que integre tanto el ruteo de vehículos como el inventario en los depósitos de clientes y proveedores tal y como se presenta en el IRS propuesto.

6.1 Medición del impacto

Se entiende como medición de impacto al proceso de investigación, planificación y control de posibles consecuencias y efectos a mediano y largo plazo en la implementación de un proyecto o un modelo establecido (CEPAL-ILPES, 2005). Esta investigación puede ser cuantitativa o cualitativa dependiendo del caso y del tipo de trabajo o proyecto que se realizó. Básicamente la evaluación de impacto busca responder a preguntas sobre causa y efecto, en donde se determina las consecuencias positivas y negativas que se generaron.

Este proceso tiene como objetivo determinar los efectos causados por la intervención de un proyecto o acción en un área o sector determinado. Adicionalmente, permite examinar posibles consecuencias previstas o no previstas por los ejecutores del proyecto, estos impactos pueden ser de varios tipos: sociales, económicos, financieros, ambientales, operacionales, institucionales, entre otros. Es importante aclarar que esta medición no debe ser realizada una vez el trabajo ha concluido, se requiere un tiempo mínimo para que los efectos y consecuencias del proyecto se desarrollen y puedan ser medidos de una forma clara.

Sin embargo, durante las últimas fases de ejecución de un proyecto o inmediatamente después de su finalización, es posible realizar una medición de posibles impactos a mediano y largo plazo, este es el caso del presente trabajo. A continuación, se presentan los impactos previstos del IRS (Inventory Routing Solver):

- Impacto económico

El IRS como se ha mencionado anteriormente, es una heurística que resuelve el problema de optimización de ruteo con inventarios, por tal motivo, las empresas que ejecuten este software se verán impactadas positivamente en su eficiencia operacional, es decir, disminuirán costos de ruta ya que estas serán significativamente menores y reducirán costos de inventarios ya que optimizarán este proceso.

- Impacto Social-Ambiental

Otro aspecto a resaltar es el impacto ambiental y social que el IRS puede causar, debido a la reducción de distancias de recorrido entre clientes, depósitos y bodegas, es claro que las emisiones de dióxido de carbono disminuirán. Por otro lado, el tiempo en que los camiones se encuentran en la maya vial también se reducirá, aportando así un grano de arena a descongestión del tráfico de la ciudad en donde se implemente el software.

- Impacto científico

En la literatura actual se evidencia la forma en como a través del MIP se encuentran soluciones optimas o cercanas a este para un listado de instancias, Vargas et al, (2014), así mismo es evidente como para otro listado de instancias este MIP después de un tiempo determinado (3600 segundos) finaliza su ejecución, dejando así este listado de instancias sin solución. Por esta razón, el presente IRS genera un impacto científico positivo aportando literatura valiosa, presentando soluciones no encontradas anteriormente en problemas específicos.

7. Conclusiones y recomendaciones

- Se realizó el IRS (Inventory Routing Solver) que permite a los usuarios y a las empresas que lo implementen aumentar su eficiencia operacional, disminuyendo costos de ruteo e inventarios, en donde cualquier persona puede usarlo sin necesidad de cursos o manejos especiales de programas digitales. Sin embargo, en un futuro se desea que se pueda mejorar el rendimiento del programa en otro lenguaje de programación junto con herramientas y recursos más apropiados para el desarrollo de este.

- Se diseñó una interfaz gráfica con un sistema de seguridad de la información bajo los estándares de normatividad de software que acompaña el IRS permitiendo a los usuarios y clientes del programa manejarlo de una forma fácil e intuitiva, cumpliendo así con los objetivos principales del proyecto.

- Se obtuvo una diferencia porcentual media (GAP) del 17.97% con respecto a las soluciones arrojadas por el método exacto del modelo de programación lineal. No obstante, al momento de realizar grupos para cada conjunto de instancias, se determinó que para instancias entre 5 a 15 clientes se logró un GAP promedio de 10,5% con un máximo de 19,1% y un mínimo de 0,0%; con respecto a las instancias entre 20 a 35 clientes se obtuvo un GAP promedio de 17,4% con un máximo de 23,6% y un mínimo de 7,2% y para instancias entre 40 a 50 clientes se consiguió un GAP promedio de 24,3% con un máximo de 34,8% y un mínimo de 14,04%.

- A partir de las soluciones obtenidas por el IRS, se encontró que el tiempo promedio de ejecución corresponde al 13% del método exacto, lo que quiere decir que se pasó de obtener soluciones óptimas en 2143 segundos, a extraer resultados 18% por encima de la mejor función objetivo por medio de un modelo metaheurístico, en un promedio de 283 segundos.

- El presente proyecto IRS arrojó soluciones eficientes con respecto a la literatura tanto Archetti et al, (2007) y Vargas et al, (2014) como se evidencia en el indicador de resultados GAP (diferencia porcentual media). Sin embargo, como se mencionó en las restricciones de diseño y ejecución del software, el programa fue ejecutado en un computador de uso doméstico con limitaciones de memoria y procesador, por esta razón se recomienda potenciar herramientas tecnológicas como HPC (High Performance Computing) que permitan obtener mejores soluciones en tiempos menores.

- Se establecieron posibles impactos a mediano y largo plazo que el IRS podría presentar en diferentes sectores: económico, social, ambiental y científico, en donde cada uno repercute de forma significativa en su área, ya sea aumentando la eficiencia operacional de una empresa, aportando a la descongestión del tráfico de la ciudad donde es implementado, contribuyendo con la disminución de contaminación por la combustión que generan los vehículos o la inclusión de nuevos estudios a la literatura científica. Por lo cual, se recomienda realizar una nueva medición de impacto una vez se haya cumplido un tiempo adecuado, en donde todas las variables e impactos mencionados anteriormente se desarrollen y puedan ser medidas de una forma cuantitativa.

- Se recomienda buscar nuevas líneas de financiamiento para el fortalecimiento y aplicación de herramientas tecnologías al presente proyecto, para así encaminarse al desarrollo de un Spin-off permitiendo el ingreso del software al mercado.

8. Glosario

Problema de ruteo con inventarios (IRP): Conocido en inglés como 'Inventory Routing Problem' se refiere a todos aquellos problemas relacionados con el manejo del inventario y conjuntamente el ruteo entre el almacén con los clientes o depósitos donde se trasladará dichos productos almacenados.

Inventario controlado por el proveedor (VMI): Conocido en inglés como 'Vendor Managed Inventory' se refiere a la estrategia que adoptan algunas compañías con sus proveedores, en donde dichos proveedores son los encargados de manejar el nivel de inventario de las estanterías y reabastecerlas conociendo información de primera mano y en tiempo real acerca del minorista.

Efecto látigo: Se refiere a una tendencia de cambios en inventario cada vez más grande dentro de cada eslabón de la cadena de suministro cuando la demanda del cliente final cambia con respecto al tiempo.

Proveedor: Empresa que abastece a otra de los recursos que son solicitados o necesarios para un fin determinado.

Minorista: Empresa dedicada al comercio de productos directamente con el consumidor final.

Depósito: Almacén o lugar donde se almacenan recursos para ser luego utilizados o distribuidos con un fin determinado.

Inventario: En inglés 'Stock' se refiere al producto terminado almacenado, ordenado y enlistado con el fin de poseer información de las unidades existentes en momentos de tiempo específicos.

Nivel de inventario: Cantidad de recursos que se mantienen en el almacén en un momento específico de tiempo.

Nivel de servicio: Se define como el porcentaje o la cantidad de pedidos que son realizados y cumplidos con respecto al total de pedidos solicitados.

Rotación de inventario: Se refiere al número de veces que los productos son vendidos por lo que salen del lugar donde son almacenados.

Demanda difusa: Se refiere al tipo de demanda que carece de claridad o precisión, es decir, que hace falta investigación o estructuración para tener más información sobre ella.

EOQ: Es la cantidad de pedido de compra para el reabastecimiento que a su vez minimiza los costes de inventario totales.

Heurística: En ingeniería consiste en el método basado en conocimientos y experiencias utilizado para la resolución de problemas específicos.

Metaheurística: Es un método heurístico para resolver un tipo de problema de manera general solicitando los parámetros dados por el usuario y esperando obtener eficientemente soluciones óptimas.

Algoritmo genético: Es un método adaptivo que suele usarse para resolver problemas de búsqueda y optimización. Está basado en el proceso de genético de organismos vivos.

Función fitness: Dentro del algoritmo genético es la función que permite evaluar cada individuo (solución) para luego definir si se tomará o se desechará dentro del conjunto de individuos obtenidos.

Optimización por enjambre de partículas (PSO): Es un método adaptivo que suele usarse para resolver problemas de búsqueda y optimización. Está basado en el comportamiento de las partículas o abejas en la naturaleza.

Referencias

Archetti, C., Bertazzi, L., Laporte, G., Speranza, M. G. (2007) . *A branch-and-cut algorithm for a vendor-managed inventory-routing problem*. 4382–391.

Aydin, N. (2014). *A genetic algorithm on inventory routing problem*. *Emerging Markets Journal, 3*(3), 59-66.

Bertazzi, L., Paletta, G., Speranza, M.G. (2002). *Deterministic order-up-to level policies in an inventory routing problem. Transportation Science 36(1). 119 - 132*

Balamurugan, T., Karunamoorthy, L., Arunkumar, N., & Santhosh, D. (2018). O*ptimization of inventory routing problem to minimize carbon dioxide emission*. Anna University.

Bocanegra. C, Vázquez. M. (2015). Subordinación de la cadena de suministro global comercio minorista: Wal-Mart Stores, Inc: Revista Nicolaita de Estudios Económicos.

Business Results of Vendor Managed Inventory. (2018). Tomado de: https://www.datalliance.com/writable/resources/Datalliance-Business-Results-of-VMI.pdf

Canizo, E. Lucero, P, (2002*).* Linear Generalize Optimizer. Software computacional

Cárdenas. L, Garza. G, Wee. H. (2012). A simple and better algorithm to solve the vendor managed inventory control system of multi-product multi-constraint economic order quantity model: Elsevier Inc.

Coelho, L.C., Cordeau, J.F., Laporte, G.: Consistency in multi-vehicle inventory-routing. Transportation Research Part C: Emerging Technologies 24. (2012) 270 – 287

DMSO, (2001). Defense Modeling and Simulation Office. Tomado de: http://acqnotes.com/acqnote/tasks/verification-validation-and-accreditation

Escobar, J., Linfati, R., Toth, P., & Baldoquin, M. (2014). *A hybrid Granular Tabu Search algorithm for the Multi-Depot Vehicle Routing Problem*. Springer.

Fernández, C. M. (2012). La norma ISO 27001 del sistema de gestión de la seguridad de la información.

Frazelle. E. (2002). Supply Management: McGraw-Hill Professional, AccessEngineering.

Fusing Supply Chain and Big Data for Continual Improvement. (2016). Tomado de https://flashglobal.com/blog/supply-chain-and-big-data/

ISO/IEC TR 9126. (2003) *Software Engineering Product quality.*

Karoonsoontawong, A., & Unnikrishnan, A. (2013). *Inventory Routing Problem with Route Duration Limits and Stochastic Inventory Capacity Constraints*.

González, L. L. (2004). *El diseño de interfaz gráfica de usuario para publicaciones digitales*. Revista digital universitaria

Guerrero, W,. (2014). Modelos y métodos de optimización para el problema de localización y ruteo de inventarios

Hernández. G. (2013*) MATLAB. MATrix LABoratory Software* computacional

Mocholí. M. (2014*). GAMS. Control y gestión de finanzas*. Software computacional

Montenegro. M, Pulido. J, Palacio. O. (2010). Coordinación de existencias mediante la administración de inventarios por parte del proveedor – VMI: Universidad Militar Nueva Granada, Facultad de ingeniería.

Moreno. J (2004). Metaheurísticas: Concepto y propiedades: Universidad de la laguna, departamento de estadística, I.O y computación.

Myrson. P. (2012). Beyond the Four Walls: I Can See Clearly Now: McGraw-Hill Professional, AccessEngineering.

Myrson, P. (2012). JIT in Supply Chain and Logistics: This JIT Is Good: McGraw-Hill Professional, AccessEngineering.

Ramkumar, N., Subramanian, P., Narendran, T. T., & Ganesh, K. (2012). Mixed integer linear programming model for multi-commodity multi-depot inventory routing problem. *Opsearch*, *49*(4), 413-429.

Ruiz. Xavier, Martinez. Pablo. (2019) smartmonkey.io. Software computacional.

Renaudl, J., Laporte, G., & Boctor, F. (1995). *A tabu search heuristic for the multi-depot vehicle routing problem*. Elsevier Science Ltd.

Reza. S, Akhavan.S, Roozbeh. A. (2011). A genetic algorithm for vendor managed invenoty control system of multi – producto multi - constraint economic order quantity model: Elsevier Inc.

Robinson, A. (2015). Walmart: Keys to Successful Supply Chain Management. Tomado de: https://cerasis.com/2015/05/13/supply-chain-management/

Qin, L., Miao, L., Ruan, Q., (2014) Zhang, Y A local search method for periodic inventory routing problem

Sadeghi. J, Saeid. S, Niaki. S. (2014). Optimizing a hybrid vendor-managed inventory and transportation problem with fuzzy demand: An improved particle swarm optimization algorithm: Elsevier Inc.

Waller. M, Johnson. E, Davis. T. (2001). Vendor – Managed Inventory in the retail supply chain: Journal of Business Logistics.

Xianfeng, Y., & Shanjiang, Z. (2016). *Solution to the Multidepot Inventory Slack-Routing Problem at the Planning Stage*. American Society of Civil Engineers.

Yang, X., & Zhu, S. (2016). *Solution to the Multidepot Inventory Slack-Routing Problem at the Planning Stage*.

Yao. Y, Dong. Y, Dresener. M (2004). Managing Supply Chain Backorders under Vendor Managed Inventory: A Principal-Agent Approach and Empirical Analysis: University of South Carolina.

Zapata-Cortes, J.A., Arango-Serna, M.D. and Serna-Urán, A. (2018), Comparison of three IRP-based models to reduce logistics costs and greenhouse gas emissions. DYNA, 85(205), pp. 199-204.

Tabla de contenido