



Trabajo de grado en modalidad de aplicación

Programación de carrotaques de transporte de hidrocarburos mediante una búsqueda tabú

John Alexander Cardona García^{a,c}, Oscar Fabián Garzón Albarracín^{a,c}, Daniela Osorio Morales^{a,c},

Juan Carlos García Díaz, Ph.D.^{b,c}

^aEstudiante de Ingeniería Industrial

^bProfesor, Director del Proyecto de Grado, Departamento de Ingeniería Industrial

^cPontificia Universidad Javeriana, Bogotá, Colombia

Resumen de diseño en Ingeniería

As an application project of knowledge in engineering, it was evidenced a possible process improvement opportunity in the Colombian hydrocarbons sector. Considering that in the Oil field there are locations where the crude oil is extracted in order to be transported to refineries or other places where it is distilled or sent to different locations by other transport means.

Therefore, the problem that is going to be solved is a variant of the well known multi-depot vehicle routing problem, with a homogenous fleet, under the constraints of minimum amount of tank-trucks hired from available companies. After searching for bibliographical sources regarding the observed situation, it was proceeded to define a mathematical model then, design and elaborate a hybrid Tabu Search to determine how many tank-trucks to hire and the respective routes for each one.

Finally, to be able to measure the performance of the result obtained with the application, in contrast with the current method, a simulation of the routes network for the tankers was made. The feasibility of the solution under random scenarios was set as a performance indicator.

The method proposed in this project achieved to improve the current process in an 18.77%, reducing the transportation cost by \$ 3.465.065.575.

1. Justificación y planteamiento del problema

En Colombia, el comercio en el sector de Hidrocarburos es una actividad determinante en el desarrollo y equilibrio económico del país. En este sector, se ven reflejados efectos realmente importantes. Como lo son: la actividad económica, exportaciones, inversión privada y generación de ingresos públicos (Unidad de planeación minero energética, 2013).

Hoy en día, el sector de Hidrocarburos cuenta con una infraestructura de transporte diseñada para trasladar el crudo desde su ubicación de producción hasta refinerías o puertos de exportación. Así, para transportar los principales compuestos del petróleo se emplean los medios de oleoductos, carrotanques y transporte fluvial. A pesar de que el transporte de crudos por el sistema de oleoductos abarca el 81.28% (Ecopetrol, 2009), la cobertura de la red del mismo en Colombia es insuficiente para la evacuación de la producción total, lo cual hace necesario el uso de carrotanques para poder cumplir con el 12.09% del transporte de crudo y sus derivados (Ecopetrol, 2009).

Actualmente, se ha estado implementando un modelo de planeación para el transporte en carrotanques del crudo donde se tiene en cuenta el cumplimiento de la demanda de crudo que debe ser cargada en un punto determinado de cargue y la cantidad mínima de carrotanques a contratar a cada una de las empresas prestadoras del servicio de transporte, con el objetivo de llevar el crudo a determinados puntos de descargue en un horizonte de planeación específico.

Analizando el modelo de planeación actual, diariamente, se movilizan cerca de 500 carrotanques. Cada carrotanque genera un costo de \$1'242.137 por una disponibilidad de 24 horas (Ecopetrol, 2016). Suponiendo así, que los carrotanques son empleados para recorrer rutas para las cuales se necesite tener disponibilidad de 24 horas del día, el costo total de transporte generado es de \$621'068.500 por la movilización de 5'000.000 galones diarios de producto.

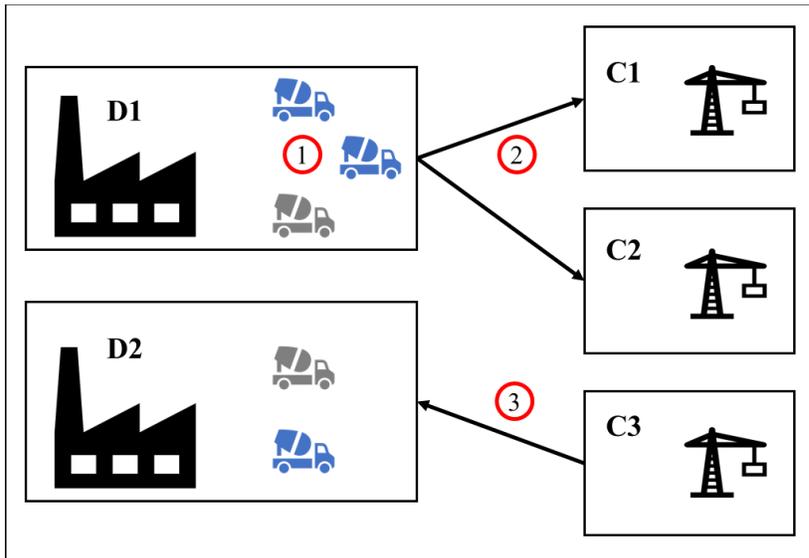
Debido a que el modelo descrito, no contempla la minimización de costos de transporte para hallar su solución, permite desarrollar una oportunidad de mejora.

Se ha evidenciado que las técnicas de planeación de la distribución en la práctica producen ahorros que van desde el 5% al 25% de los costos totales de transporte (Avella, Boccia y Sforza, 2004). De esta manera, un modelo de planeación para la distribución de crudo basado en técnicas de computer science (ciencia computacional) lograría reducir considerablemente los costos de transporte de crudo movilizados de un punto de cargue a un punto de descargue por medio de carrotanques.

En este trabajo, se propone implementar una solución, que al igual que el método actual, cumpla con la demanda de los puntos de cargue de crudo y la contratación de carrotanques requerida por cada empresa, añadiendo la reducción de costos totales de transporte por medio de la minimización de la cantidad de carrotanques contratados y los tiempos de ruta asignadas a cada uno de ellos.

Tomando a Cantu, Salazar y Voyer (2017), el clásico problema de VRP consiste en una red $n+1$ nodos donde el nodo 1 representa el centro de abastecimiento y los nodos $\{2, 3, \dots, n+1\}$ representan el conjunto de clientes. El objetivo de este problema es minimizar el tiempo total de viajes logrando cumplir con la demanda de visitas de cada cliente. El VRP es adaptado a un MDVRP en el cual existen $n+1$ centros de abastecimiento (nodos de descargue) y $n+1$ clientes (nodos de cargue), tomando las siguientes decisiones para encontrar su solución:

Ilustración 1. Decisiones del Problema



1. ¿Cuántos carrotaques, de cada empresa prestadora del servicio de transporte, deben ser contratados al inicio de la programación?
2. ¿Qué nodo de carga debe visitar el carrotaque, partiendo del nodo de descarga en el que se encuentra?
3. ¿Qué nodo de descarga debe visitar el carrotaque, partiendo del nodo de carga en el que se encuentra?

2. Antecedentes

El problema abordado en este trabajo se define como una variante del VRP, MDVRP. Para este problema se contemplan las siguientes características:

- Flota Homogénea: los carrotaques tienen la misma capacidad de transporte.
- Múltiples centros de abastecimiento: Existe $n+1$ centros de abastecimiento en la red
- Múltiples Viajes: cada carrotaque puede ejecutar más de un viaje durante todo el horizonte de planeación
- Entrega dividida: Cada cliente puede ser visitado más de una vez, con el fin de cumplir con su demanda total.
- Tercerización de vehículos de transporte y porcentaje de participación: Los vehículos son contratados con diferentes empresas prestadoras del servicio con un porcentaje mínimo de participación para cada una de ellas.

Se tomó como referencia a Cantu-Funes, Salazar-Aguilar, Boyer, (2017) quienes describen un problema que se asemeja en mayor proporción a las características del problema abordado en este trabajo. Cantu-Funes, Salazar-Aguilar, Boyer, (2017), consideran un problema MDPVRP con ventanas de tiempo y fechas de vencimiento para atender a cada cliente. El horizonte de planeación es dividido en periodos de tiempo iguales, para lograr el cumplimiento de estas restricciones. Finalmente, los viajes asignados a un vehículo son contemplados en grupos de 3 nodos, considerando así, la unión de 2 arcos de la red como 1 viaje (nodo i – nodo j – nodo i).

Por otro lado, Surjandari, Rachman, Dianawati y Wibowo (2011) abordan el problema como un Petrol Station Replenishment Problem (PRSP), en donde se plantean como objetivo optimizar la entrega del petróleo a un conjunto de estaciones de gasolina usando una flota limitada de carotanes. El problema es restringido por múltiples centros de abastecimiento, múltiples productos, entrega dividida y ventanas de tiempo para el cumplimiento de la entrega en cada estación de gasolina. Se propone como solución un TS, en donde el número de soluciones del vecindario y la lista Tabu son definidas por medio de pruebas experimentales.

Eric Prescott-Gagnon, Guy Desaulniers, Louis-Martin Rousseau, (2014) presentan un problema ODVRPTW con múltiples centros de abastecimiento y flota heterogénea de vehículos. En él proponen dos algoritmos híbridos: primero un LNS que utiliza un TS para el proceso de reconstrucción de la solución y segundo un LNS que utiliza un CG en la fase de reconstrucción. De esta manera, se analizan los resultados, donde el LNS con CG arroja mejor calidad de solución, sin embargo, el LNS con TS arroja buenos resultados en menor tiempo sin tener que invertir en un software comercial de programación lineal. De acuerdo a lo anterior, para la solución del problema se tendrá en cuenta la meta-heurística de trayectoria TS planteada en el problema, ya que tiene las restricciones de múltiples centros de abastecimiento; además, cumple con el requerimiento de desempeño de resolver el problema en un tiempo computacional reducido.

Ran Liu, Zhibin Jiang y Na Geng, (2012) proponen una hibridación para el MDOVRP donde se propone un HGA para resolver el MDOVRP. En él, el GA utiliza una heurística de búsqueda local en vez del procedimiento de mutación normal del algoritmo; por lo tanto, se tiene en cuenta para el desarrollo de la meta-heurística, el proceso de hibridación con heurísticas de búsqueda local para la resolución de sub-problemas como el reordenamiento de los elementos de cada ruta.

Mao Xiaocun, Zhang Xinxin y Li Zhenping (2016) presentan un VRP donde se propone un Scheduling para una distribución de crudo refinado utilizando carotanes con la carga completa (Full Load) para la distribución. Se propone métodos heurísticos teniendo como parámetro de solución los costos debido al tamaño de la flota desarrollando el enrutamiento de los vehículos en múltiples periodos de entrega.

Finalmente, Francois Despau, Sebastian Basterrech, (2014), proponen un SA para resolver el MTRPTW donde utilizan la heurística SI para inicializar la búsqueda del SA, cuyo resultado arroja una solución de calidad en un tiempo computacional bajo.

3. Objetivos

3.1. Objetivo General

Desarrollar una meta-heurística para generar una programación de carotanes más eficiente que la obtenida con los métodos actuales, cumpliendo con las demandas de cada refinería y reduciendo los tiempos y costos de transporte.

3.2. Objetivos Específicos

- Analizar el contexto de la problemática para la obtención de variables, parámetros y restricciones que serán empleados en el desarrollo de las metodologías y diseño de la aplicación.

- Diseñar un aplicativo que por medio de una meta-heurística arroje una solución factible de la programación de carrotaques, en donde se pueda interactuar con los parámetros y solución del modelo.
- Diseñar un modelo de simulación para contrastar el porcentaje de veces que la solución obtenida con el modelo propuesto y el método actual cumplen con las restricciones del problema.
- Evaluar la factibilidad del modelo propuesto para la solución del problema mediante análisis de los indicadores de desempeño y el impacto del proyecto en términos financieros en contraste con la solución del método actual.

4. Metodología

4.1. Análisis del contexto del problema

Se define y describe el problema, con el fin de identificar los parámetros, variables y restricciones que serán empleados en su solución. En primer lugar, el problema es definido junto con su modelamiento matemático; posteriormente, los datos reales son obtenidos de una base de datos, y finalmente, estos datos son analizados estadísticamente para obtener las distribuciones de probabilidades requeridas en la simulación.

4.1.1. Definición del problema

El problema consiste en planear las rutas de una flota de carrotaques encargados de cargar crudo en nodos de cargue y descargarlo en nodos de descargue, minimizando el costo total de transporte de crudo durante una semana (168 horas). Para esto, se asigna un número determinado de viajes a cada carrotaque con el fin de satisfacer la demanda de los nodos de cargue.

El servicio de transporte de carrotaques es contratado con 5 diferentes empresas. Cada empresa exige un porcentaje de participación mínimo del total de carrotaques contratados al final de la semana y genera un cobro por la contratación de un carrotaque y el número de horas en las que el carrotaque es empleado. Al inicio de cada semana, carrotaques de diferentes empresas se encuentran ubicados en los nodos de descargue, es decir, el nodo en donde terminó la ruta programada en la semana previa.

Los nodos de cargue de crudo de la red, cuentan con una demanda semanal de producto a extraer. La demanda es trabajada en términos de número de carrotaques necesarios para su movilización. Como se considera una flota homogénea de carrotaques, con capacidad de 200 barriles, si un nodo de carga tiene una demanda de 400 barriles, ésta será considerada como una demanda de 2 carrotaques.

Por último, las rutas disponibles entre los nodos se encuentran previamente establecidas, es decir, cada nodo de cargue tiene nodos de descargue asignados.

Los nodos de cargue y su demanda semanal en términos de carrotaques, se pueden observar en la Tabla 1. Los nodos de descargue se encuentran en la Tabla 2.

Tabla 1. Nodos de cargue

Código	Nombre	Demanda	Código	Nombre	Demanda
1	Altair	3	37	Eyparrendajoazor	24
2	Apiay	12	38	Garibayjilquero	40
3	Aullador	11	39	Guarrojo Ocelote	585
4	Labrador	59	40	Guatiquia Avispa	612
5	Leonope	11	41	La Paloma Colon Norte	13
6	Tua	808	42	Leona B Sur	14
7	Bazar	140	43	Kona	1
8	Carmentea	37	44	Adalia	49
9	Kananaskis	44	45	Begonia	56
10	Buenavistacorrales	30	46	Llanos 58	117
11	Cabiona A	34	47	Los Ocarros Las Maracas	72
12	Cabresteroakira	77	48	Magdalena Medio Monoarana	2
13	Cachicamogretaotto	45	49	Mana	36
14	Cachicamoguacharaca	11	50	Mansoya	33
15	Campos Ayacucho	179	51	Midas Aacordionero	175
16	Canaguaroanaguay	56	52	Midas Chuirra	9
17	Casanareestecurito	23	53	Moriche Mauritia Norte	54
18	Casimenamantis	89	54	Nashira	36
19	Casimenapisingo	66	55	Onca	37
20	Castillai	629	56	Oropendola	12
21	Cdentrerrios	30	57	Petromag	824
22	Ciriguelocachicamo	10	58	Platanillo	90
23	Cohembi	190	59	Quillacinga	48
24	Corcelcaruto	17	60	Quinde	42
25	Corcel C	56	61	Ramiriqui	59
26	Atarraya	24	62	Rioverde Tilodiran	25
27	Pendare	49	63	Rio Zulia	20
28	Cravoviejobastidas	19	64	Rubiales	70
29	Cravoviejocarrizales	56	65	Sabanero	40
30	Cravoviejozilote	96	66	Santana	94
31	Cubiro Careto	21	67	Sardinata	10
32	Cubiro Copa	204	68	Telba	214
33	Cubiro Petirrojo	18	69	Tibu	143
34	Cubiro Yopo	5	70	Tiple Jilguero Sur	52
35	Dorotea B	109	71	Vireo	7
36	El Eden La Casona	14			

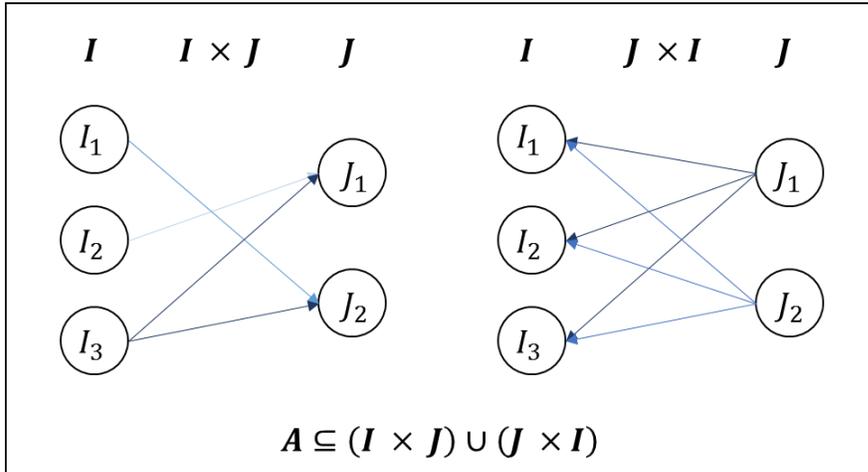
Tabla 2. Nodos de descargue

Código	Nombre	Código	Nombre
1	AraguaneyDescPesados	9	MezclaDiluPAC
2	AraguaneyDescVasconia	10	MonterreyDesc
3	AyacuchoDesc	11	Orito
4	CusianaCPFECPCDesc	12	RubialesECP
5	CusianaCPFPACDesc	13	RubialesPAC
6	EstacionBabillas	14	Santana
7	Jaguey	15	VasconiaDesc
8	LeridaDesc	16	Velazquez26

Formulación

Se define la red del problema como el conjunto de nodos $B = \{I \cup J\}$, donde I contiene el conjunto de nodos de descarga y J el conjunto de nodos de carga. Los arcos que unen los nodos, está dado por el conjunto de arcos $A \subseteq (I \times J) \cup (J \times I)$, el cual representa los arcos disponibles desde un nodo de descarga i a un nodo de carga j y los arcos disponibles desde un nodo de carga j a un nodo de descarga i .

Ilustración 1. Red del Problema



Conjuntos

$I = \{1, 2, \dots, I\}$ – Nodos de descarga

$J = \{1, 2, \dots, J\}$ – Nodos de carga

$A = \{I \times J \cup J \times I\}$ – Posibles viajes entre los nodos de descarga y nodos de carga

$B = \{I \cup J\}$ – Nodos de descarga y nodos de carga

$K = \{1, 2, \dots, kmax\}$ – Carrotaques disponibles

$E = \{1, 2, \dots, E\}$ – Empresa prestadora del servicio de carrotaques

$L = \{1, 2, \dots, lmax\}$ – Número de viajes asignados a cada carrotaque

Parámetros

mb_{ij} : Matriz binaria de viajes previamente establecidos entre el nodo i y el nodo j

d_{ij} : Matriz de tiempo entre el nodo i y el nodo j

td : Tiempo promedio de descarga/carga en un nodo

do_j : Demanda total de carrotaques en el nodo de carga j durante el horizonte de planeación

kd_{ie} : Número inicial de carrotaques disponibles de la empresa e en cada nodo de descarga i

pe_e : Porcentaje mínimo de participación de la empresa e

T : Duración del horizonte de planeación

$kmax$: Número máximo de carrotaques disponibles

$lmax$: Número máximo de viajes posibles a asignar a un carrotaque

ckf : Costo fijo por contratar un carrotaque

ck : Costo por hora de uso de un carrotaque

Variables de decisión

$$X_{ijkl} = \begin{cases} 1 & \text{Si el carrotaque } k \text{ de la empresa } e \text{ viaja de } i \text{ a } j \text{ en su viaje } l \\ 0 & \text{d.l.c.} \end{cases}$$

$$U_{ke} = \begin{cases} 1 & \text{Si el carrotaque } k \text{ de la empresa } e \text{ es contratado} \\ 0 & \text{d.l.c.} \end{cases}$$

S_{kel} : Tiempo de finalización del viaje l del carrotaque k de la empresa e

HI_{ke} : Tiempo de inicio de la ruta del carrotaque k de la empresa e

Función objetivo

$$\min Z: \sum_{(i,j) \in A} \sum_{k \in K} \sum_{e \in E} \sum_{l \in L} d_{ij} X_{ijkl} ck + \sum_{k \in K} \sum_{e \in E} ckf U_{ke} \quad (1)$$

s.a.

$$\sum_{(i,j) \in A} \sum_{k \in K} \sum_{e \in E} \sum_{l \in L} X_{ijkl} \geq do_j, \quad \forall j \in J, \quad (2)$$

$$\sum_{(i,j) \in A} \sum_{k \in K} X_{ijke1} \leq kd_{ie}, \quad \forall i \in I, \forall e \in E, \quad (3)$$

$$S_{ke,l-1} + \sum_{(i,j) \in A} X_{ijkl} (d_{ij} + td) = S_{kel}, \quad \forall k \in K, \forall e \in E, \forall l \in L \setminus \{1\}, \quad (4)$$

$$HI_{ke} + \sum_{(i,j) \in A} X_{ijke,1} (d_{ij} + td) = S_{ke1}, \quad \forall k \in K, \forall e \in E, \quad (5)$$

$$X_{ijkl} \leq U_{ke}, \quad \forall (i,j) \in A, \forall k \in K, \forall e \in E, \forall l \in L, \quad (6)$$

$$\sum_{k \in K} U_{ke} \geq \sum_{k \in K} \sum_{n \in E} U_{kn} p_{e'}, \quad \forall e \in E, \quad (7)$$

$$S_{kel} + \sum_{(i,j) \in A} X_{ijkl} d_{ij} \leq T, \quad \forall k \in K, \forall e \in E, \forall l \in L, \quad (8)$$

$$\sum_{(i,j) \in A} X_{ijkl} \leq 1, \quad \forall k \in K, \forall e \in E, \forall l \in L, \quad (9)$$

$$X_{ijkl} \leq mb_{ij}, \quad \forall (i,j) \in A, \forall k \in K, \forall e \in E, \forall l \in L, \quad (10)$$

$$\sum_{(i,j) \in A} X_{ijkl} \geq \sum_{(i,j) \in A} X_{jike,l+1}, \quad \forall j \in B, \forall k \in K, \forall e \in E, \forall l \in L \setminus \{lmax\}, \quad (11)$$

$$\sum_{(i,j) \in A} \sum_{k \in K} \sum_{e \in E} X_{jike1} \leq 0, \quad \forall j \in J, \quad (12)$$

$$X_{ijkl} \in \{0,1\}, \quad \forall (i,j) \in A, \forall k \in K, \forall e \in E, \forall l \in L, \quad (13)$$

$$U_{ke} \in \{0,1\}, \quad \forall k \in K, \forall e \in E, \quad (14)$$

$$S_{kel} \geq 0, \quad \forall k \in K, \forall e \in E, \forall l \in L \quad (15)$$

El objetivo **(1)** es minimizar el costo total de transporte, donde se incluye el costo variable de que el carrotanque se desplace de un nodo a otro junto con el costo fijo de contratar un carrotanque. La restricción **(2)** garantiza que la demanda de carrotanques que deben visitar cada nodo de cargue se cumpla. La restricción **(3)** evita que la cantidad inicial de carrotanques de una empresa determinada en cada punto de descargue al inicio de la semana se exceda. Las restricciones **(4)** y **(5)** imponen consistencia en el tiempo entre viajes consecutivos de un mismo carrotanque, es decir, el carrotanque no puede iniciar un nuevo viaje antes de terminar con el viaje anterior. Las restricciones **(6)** y **(7)** garantizan que se contrate como mínimo un porcentaje de carrotanques de cada empresa sobre el número total de carrotanques contratados en el horizonte de planeación. La restricción **(8)** garantiza que el tiempo total trabajado por un carrotanque no exceda la duración del horizonte de planeación. La restricción **(9)** asegura que un carrotanque solo puede recorrer un arco de la red $\{(i,j), (j,i)\}$ por viaje. La restricción **(10)** garantiza que un carrotanque solo puede recorrer las rutas previamente establecidas entre los nodos de cargue y nodos de descargue. La restricción **(11)** impone consistencia en el nodo de finalización e inicio entre viajes consecutivos de un mismo carrotanque, es decir, el carrotanque no puede iniciar un nuevo viaje en un nodo diferente al nodo en donde finalizó el viaje inmediatamente previo. La restricción **(12)** asegura que ningún carrotanque inicie su ruta en un punto de cargue. Las restricciones **(13)**, **(14)** y **(15)** definen la naturaleza de las variables de decisión.

4.1.2. Análisis de Datos

El tiempo en que un carrotanque tarda en ir de un nodo a otro dentro de las rutas definidas, fue obtenido de una base de datos, en la cual se encontraba el tiempo de diferentes viajes para los arcos establecidos en la red.

Para el modelo matemático, la heurística y meta-heurística se considera el tiempo en que un carrotanque tarda para ir de un nodo a otro como un parámetro determinístico, siendo este la media de los tiempos obtenidos para cada arco.

Por otro lado, para la simulación, el tiempo es considerado variable. A cada una de las variables se le realizó:

- Estadística descriptiva para tener un alcance de su comportamiento
- Análisis de independencia de los datos
- Evaluación y selección de la distribución de probabilidad del tiempo
- Elaboración de gráfica de P-P Plot con el fin de observar la asimetría con la distribución seleccionada

El análisis estadístico de entrada realizado al tiempo en que un carro tanque tarda de ir entre las rutas establecidas de un nodo de cargue a un nodo de descargue se encuentra en el *Anexo 6 Análisis de datos de entrada*.

4.2. **Aplicativo**

El diseño del aplicativo tuvo como primera instancia una revisión bibliográfica de artículos que describen modelos con objetivos, parámetros, variables y restricciones similares al abordado en este trabajo. Se eligió diseñar una heurística eficiente en tiempo de ejecución para poder ser utilizada como esclava en la hibridación con la meta-heurística.

Heurística

La estructura de la heurística se diseñó en relación con las restricciones, parámetros y variables definidas en el contexto del problema. Se utilizó el lenguaje de programación de Visual Basic para realizar el aplicativo, debido al alto conocimiento y el fácil acceso de la herramienta. El diagrama de flujo de la heurística se puede observar en la Ilustración 3 y su pseudocódigo se presenta a continuación:

Tabla 3. Pseudocódigo de la heurística

Heurística	<i>Voraz vecino más cercano</i>
(1)	Ordenar semillas
Hacer	
(2)	Crear ruta
(3)	Asignar ruta
Mientras la demanda insatisfecha sea mayor a 0	

(1) Ordenar semillas

Los nodos de cargue son ordenados de mayor a menor demanda. El orden de los índices es guardado en el vector semillas, de tal manera que cuando se vaya a crear una ruta, el primer punto de cargue a ser visitado es seleccionado a partir de este vector.

Se puede observar un ejemplo de nodos de cargue en la Tabla 4. En la Tabla 5 los nodos de cargue son ordenados de acuerdo a su demanda y finalmente, en la Tabla 6, la semilla es generada.

Tabla 4. Ejemplo nodos de cargue

Nodo de Cargue	Demanda
1	15
2	20
3	5
4	10
5	50

Tabla 5. Ejemplo nodos de cargue ordenados

Nodo de Cargue	Demanda
5	50
2	20
1	15
4	10
3	5

Tabla 6. Ejemplo vector semillas

Vector Semillas
5
2
1
4
3

(2) Crear ruta

El vector ruta se define como una variable de tipo estructura, es decir ruta(i) almacena 3 valores en una posición:

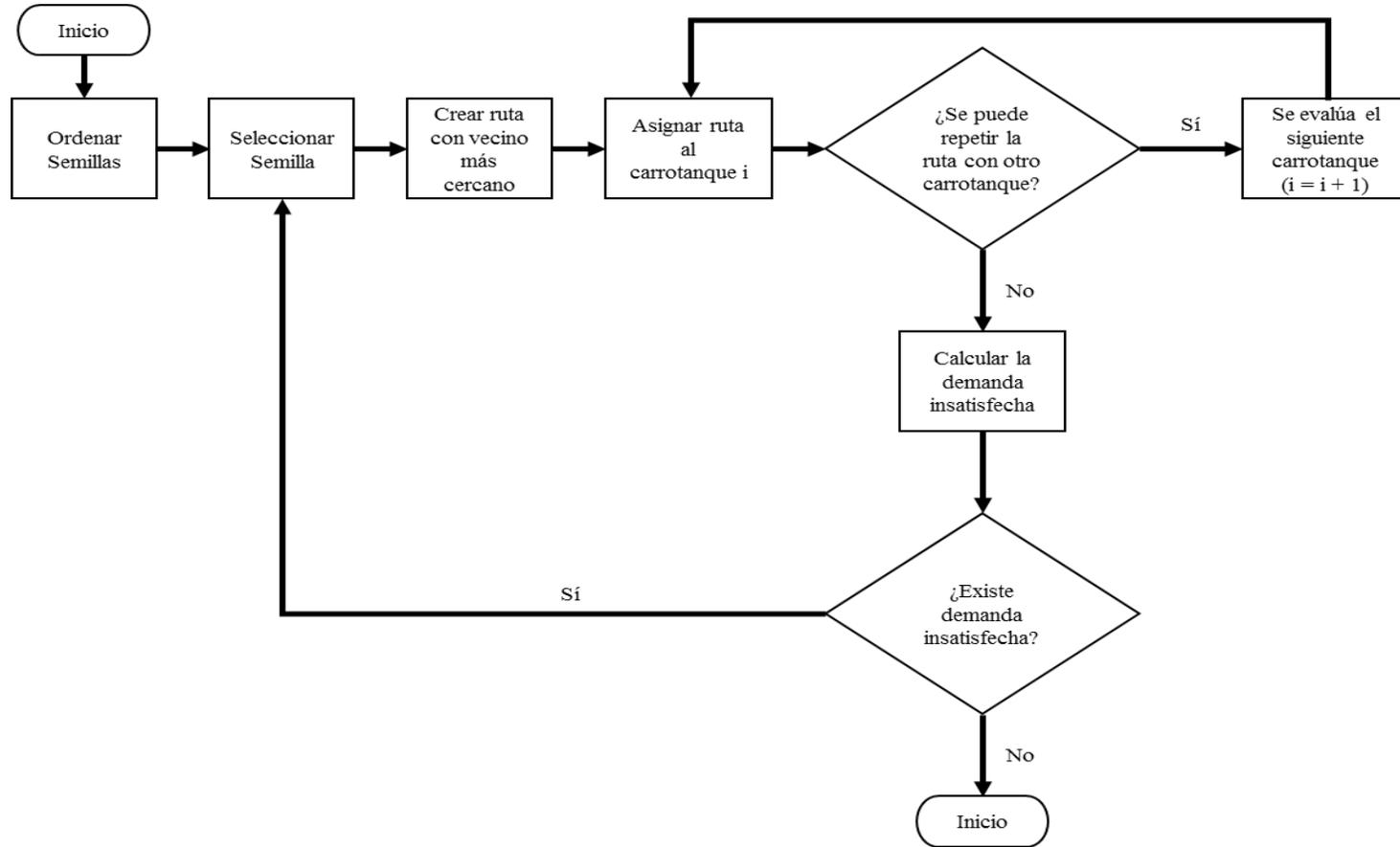
- ruta(i).Origen_Descargue: el nodo de descargue en donde el carrotanque inicia la ruta
- ruta(i).Destino_Cargue: el nodo de descargue al cual el carrotanque se dirige a cargar en el viaje
- ruta(i).Destino_Descargue: el nodo de descargue al cual el carrotanque se dirige a descargar para finalizar el viaje

En la Tabla 7 se puede observar un ejemplo del vector ruta.

Tabla 7. Ejemplo vector ruta

Ruta(i)	Origen Descargue	Destino Cargue	Destino Descargue
1	5	45	11
2	11	33	13
3	13	43	2
4	2	65	5

Ilustración 2. Diagrama de flujo Heurística



A continuación, se presenta el pseudocódigo de la función (2) *Crear ruta*.

Tabla 8. Pseudocódigo (2) *Crear ruta*

Heurística (2) <i>Crear ruta</i>
ruta(1) = buscar_Semilla()
Hacer
i = i + 1
ruta(i) = nearestN(ruta(i - 1))
Mientras ruta(i) > 0

Para generar una ruta se define la primera posición de la ruta por medio de la función *buscar_Semilla*. Esta toma del vector de semillas el primer nodo de cargue que aún tenga demanda insatisfecha y lo guarda en *Destino_Cargue*. A partir de este nodo de cargue, se define el nodo *Origen_Descargue* como el nodo de descargue con carrotaques disponibles más cercano al *Destino_Cargue* y el nodo *Destino_Descargue* como el nodo de descargue más cercano al nodo seleccionado en *Destino_Cargue*.

Ya definida la primera posición de la ruta, se continúa generando la ruta haciendo uso de la función *nearestN*. La función *nearestN* recibe como parámetro la última posición de la ruta y a partir del último *Destino_Descargue* busca el *Destino_Cargue* más cercano que sea factible, es decir, si no es posible completar la ruta hasta el *Destino_Descargue* antes del tiempo límite, la función devuelve 0, de lo contrario, devuelve una posición de la ruta completa.

(3) Asignar Ruta

Ya definida una ruta, es momento de asignarla a uno o varios carrotaques si es posible. Debido a que existen nodos de cargue que deben ser visitados más de una vez para poder satisfacer su demanda, significa que, si en una ruta todos los nodos de cargue tienen una demanda mayor a 1, esta ruta podría ser asignada a 2 o más carrotaques, es en esta parte donde se minimiza el número de carrotaques necesarios.

El proceso de asignación se realiza de la siguiente forma: se toma la ruta y se asigna a un carrotaque de una empresa que se encuentre disponible teniendo en cuenta la restricción de participación mínima. Al ser asignada la ruta a este carrotaque, se resta 1 unidad a la demanda de cada nodo de cargue por cada vez que se encuentre en la ruta y se evalúa la siguiente condición de parada. Si al menos un nodo de cargue dentro de la ruta tiene una demanda insatisfecha igual a 0, se debe terminar el proceso de asignación, de lo contrario se debe asignar nuevamente la ruta a otro carrotaque disponible y repetir el proceso hasta que la condición de parada se cumpla.

Metaheurística

Para el diseño de la meta-heurística se consideró un algoritmo de trayectoria, el Tabu Search. Este algoritmo permite mejorar la solución inicial arrojada por el voraz. El diagrama de flujo de la metaheurística se puede observar en la Ilustración 4 y su pseudocódigo se presenta a continuación:

Tabla 9. Pseudocódigo de la metaheurística

Metaheurística *Tabu Search*

solucionVigente = llenarSolucion(solucionHeuristica.vector_semillas)
 MejorSolucion = solucionVigente

Hacer

- (1) llenar_Matriz_DeltaF
- (2) Escoger_Mejor_Cambio
- (3) Actualizar_Solucion_Vigente
- (4) Actualizar_Matriz_Tabu

Si solucionVigente.FO < bestSolution.FO **entonces**
 MejorSolucion = solucionVigente
 iteracionessinmejorar = 0

Sino
 iteracionessinmejorar = iteracionessinmejorar + 1

Fin del condicional

Mientras iteracionessinmejorar < MaxIteraciones

El código del tabú comienza con la inicialización de la solución vigente con la solución encontrada por la heurística haciendo uso de la función *llenarSolucion()*. Esta función recibe como parámetro un vector de semillas y a partir de este, calcula la función objetivo del problema. Dentro de esta función se encuentra programada la heurística vecino más cercano explicada previamente.

(1) llenar_Matriz_DeltaF

A continuación, se presenta el pseudocódigo de la función (1) *llenar_Matriz_DeltaF*.

Tabla 10. Pseudocódigo de la función (1) *llenar_Matriz_DeltaF*

Metaheurística (1) *llenar_Matriz_DeltaF*

Para todo nodo de cargue en i

Para todo nodo de cargue en j

Si nodo de cargue en i <> nodo de cargue en j

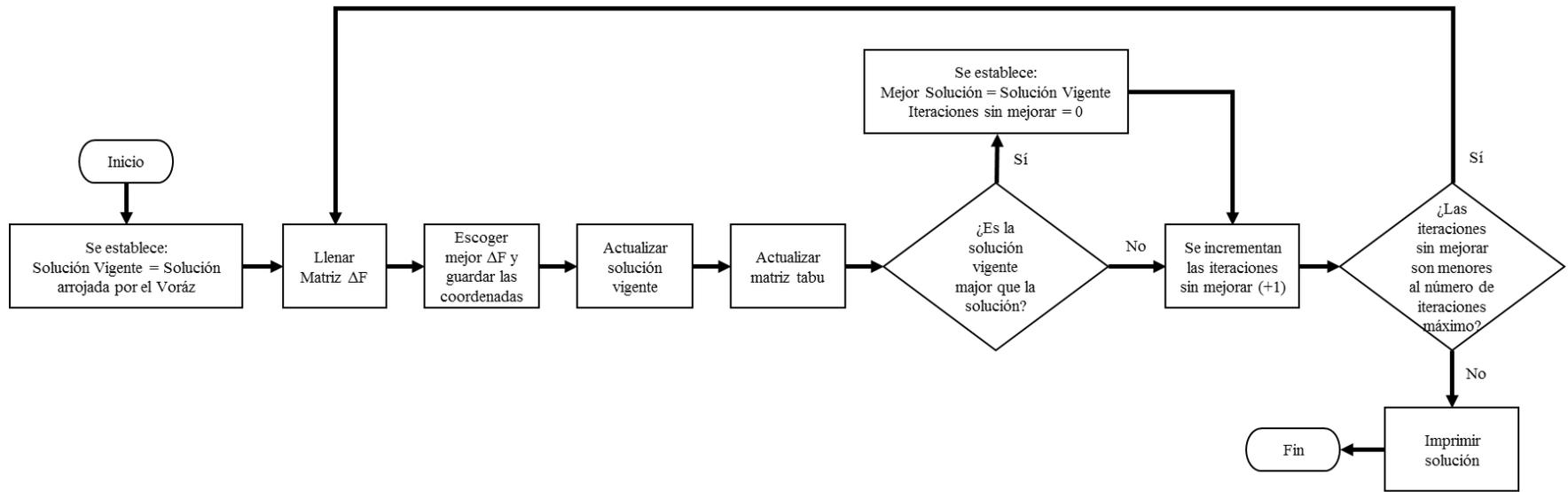
MatrizDeltaF(i,j) = (1.1) calcularDF(i,j)

Fin del condicional

Fin del ciclo

Fin del ciclo

Ilustración 3. Diagrama de flujo metaheurística



A continuación, se presenta el pseudocódigo de la función (1.1) *calcularDF*.

Tabla 11. Pseudocódigo de la función (1.1) *calcularDF*

Metaheurística - (1) llenar_Matriz_DeltaF (1.1) *calcularDF*(i,j)

```
solucionFinal = solucionVigente
aux = solucionFinal.vector_semillas(i)
solucionFinal.vectorsemillas(i) = solucionFinal.vector_semillas(j)
solucionFinal.vector_semillas(j) = aux
solucionFinal = llenarSolucion(solucionFinal.vector_semillas)
deltaF = solucionFinal.FO - solucionVigente.FO
calcularDF = deltaF
```

La función *calcularDF* se encarga de calcular el delta f entre la función objetivo vigente y una función objetivo asociada a un nuevo vector de semillas.

A manera de ilustración, se supone que el vector de semillas de la solución vigente es (5,3,1,4,2) con una función objetivo de \$1.000 y se desea calcular el delta f para $i=1$ y $j=2$. El nuevo vector de semillas sería (3,5,1,4,2) y llamando a la función *llenarSolucion*, la cual contiene el código de la heurística, se calcula la nueva función objetivo tomando como parámetro el nuevo vector. Suponiendo, para este ejemplo, que la nueva función objetivo es \$800, el delta f para $i=1$ y $j=2$ en la matriz delta f será -\$200.

El método de llamar a la heurística dentro del tabú se conoce como hibridación y permite a la meta-heurística resolver parcialmente el problema, mientras que la heurística esclava resuelve el resto del problema. Para este caso, el Tabu Search se encarga de encontrar el orden del vector de semillas que le permita a la heurística encontrar una mejor solución que la inicialmente obtenida.

(2) Escoger_Mejor_Cambio:

A continuación, se presenta el pseudocódigo de la función (2) *Escoger_Mejor_Cambio*.

Tabla 12. Pseudocódigo de la función (2) *Escoger_Mejor_Cambio*

Metaheurística (2) *Escoger_Mejor_Cambio*

```
menor = 1000000
Para todo nodo de cargue en i
  Para todo nodo de cargue en j
    Si MatrizDeltaF(i,j) < menor entonces
      Si matrizTabu(i,j) < 0 entonces
        menor = MatrizDeltaF(i,j)
        posicionI = i
        posicionJ = j
      Fin del condicional
    Fin del condicional
  Fin del ciclo
Fin del ciclo
```

La función (2) *Escoger_Mejor_Cambio* se encarga de encontrar la posición (i, j) de la matriz delta f donde se encuentre el mejor delta f. El mejor delta f es aquel que permita reducir los costos totales en mayor cantidad y no se encuentre marcado como “prohibido” en la matriz tabú. El concepto de ser marcado como “prohibido” en la matriz tabú, será explicado más adelante.

(3) Actualizar_Solucion_Vigente

A continuación, se presenta el pseudocódigo de la función (3) *Actualizar_Solucion_Vigente*.

Tabla 33. Pseudocódigo de la función (3) *Actualizar_Solucion_Vigente*

Metaheurística	(3) <i>Actualizar_Solucion_Vigente</i>
	aux = solucionVigente.vector_semillas(posicionI)
	solucionVigente.vectorsemillas(posicionI) = solucionVigente.vector_semillas(posicionJ)
	solucionVigente.vector_semillas(posicionJ) = aux
	solucionVigente.FO = solucionVigente.FO + MatrizDeltaF(posicionI,posicionJ)

Ya seleccionado el mejor delta f con sus respectivas coordenadas i y j, es necesario actualizar la solución vigente. Para actualizar la solución vigente se realiza el cambio de la posición i con la j en el vector de semillas y se suma el mejor delta f a la función objetivo vigente.

(4) Actualizar_Matriz_Tabu

A continuación, se presenta el pseudocódigo de la función (4) *Actualizar_Matriz_Tabu*.

Tabla 14. Pseudocódigo de la función (4) *Actualizar_Matriz_Tabu*

Metaheurística	(4) <i>Actualizar_Matriz_Tabu</i>
	Para todo nodo de cargue en i
	Para todo nodo de cargue en j
	matrizTabu(i,j) = Max(0,matrizTabu(i,j) - 1)
	Fin del ciclo
	Fin del ciclo
	matrizTabu(posicionI, posicionJ) = tamTabu

Después de realizar la actualización de la solución vigente, es necesario restar una unidad a cada posición dentro de la matriz delta f. Se debe tener en cuenta que no puede haber valores negativos para las posiciones, ya que la matriz tabú representa el número de iteraciones en el que un cambio con posición (i, j) va a estar prohibido. A la posición, en donde se encuentra el mejor delta f, se le debe asignar el número de iteraciones prohibidas definidas por el usuario. El número de iteraciones prohibidas es almacenado en tamTabu, en la misma posición en donde se encontró el mejor delta f.

Por último, se evalúa si la nueva solución hallada por los cambios que realiza el tabú en la heurística es mejor que la mejor solución encontrada hasta el momento. En caso de ser mejor, se actualiza la mejor solución con la solución vigente y se reinicia el acumulador de iteraciones sin mejorar; de lo contrario, se aumenta en una unidad la cuenta de iteraciones sin mejorar y se evalúa si el máximo número de iteraciones ha sido alcanzado, sino vuelve a repetirse todo el procedimiento.

Parametrización

Para lograr resolver el problema de programación de rutas de carro tanques de manera eficiente, en términos de tiempo computacional y el costo total en el que se incurre para la solución del problema, es de gran importancia establecer los valores de algunos parámetros controlables. Los parámetros controlables considerados en el aplicativo elaborado son: el número máximo de iteraciones sin mejorar y el tamaño de la lista tabú. Para establecer los parámetros controlables se realizaron pruebas experimentales para tamaños de lista tabú entre 5-30 y para el número máximo de iteraciones sin mejorar entre 10-100. Así mismo se redujo la complejidad del problema seleccionando 30 nodos de cargue de los 71.

Tabla 15. Pruebas experimentales tamaño de lista tabú

No.	Tamaño lista tabú	Tiempo computacional (segundos)	Costo total (Millones de pesos)
1	5	839	349,56
2	10	1270	341,44
3	15	853	349,56
4	20	844	349,56
5	25	849	349,56
6	30	848	349,56

Ilustración 4. Pruebas experimentales tamaño de lista tabú

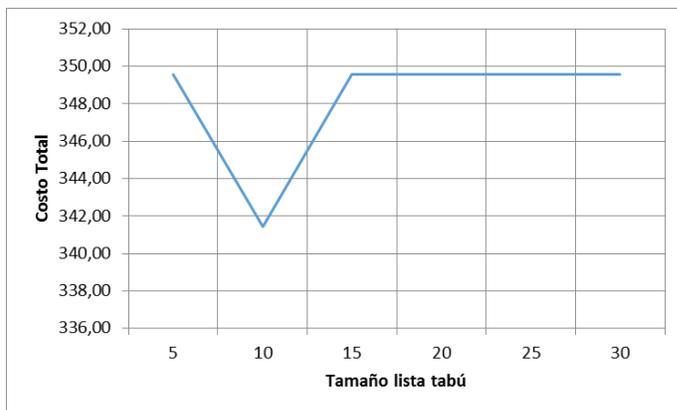
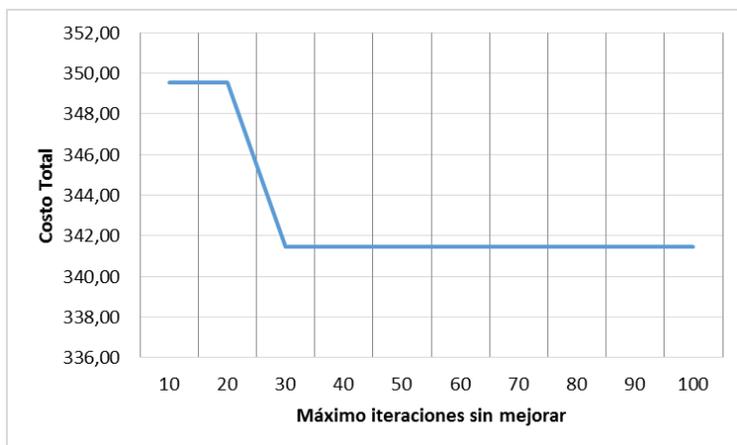


Tabla 16. Pruebas experimentales número máximo de iteraciones sin mejorar

No.	Número máximo de iteraciones sin mejorar	Tiempo computacional (segundos)	Costo total (Millones de pesos)
1	10	231	349,56
2	20	373	349,56
3	30	1499	341,44
4	40	1786	341,44
5	50	2947	341,44
6	60	3400	341,44
7	70	4962	341,44
8	80	5496	341,44
9	90	7751	341,44
10	100	9069	341,44

Ilustración 5. Pruebas experimentales número máximo de iteraciones sin mejorar



Los resultados obtenidos muestran que la eficiencia computacional en términos de costo y tiempo se alcanza con un tamaño de lista tabú de 10 y un máximo de iteraciones sin mejorar de 30.

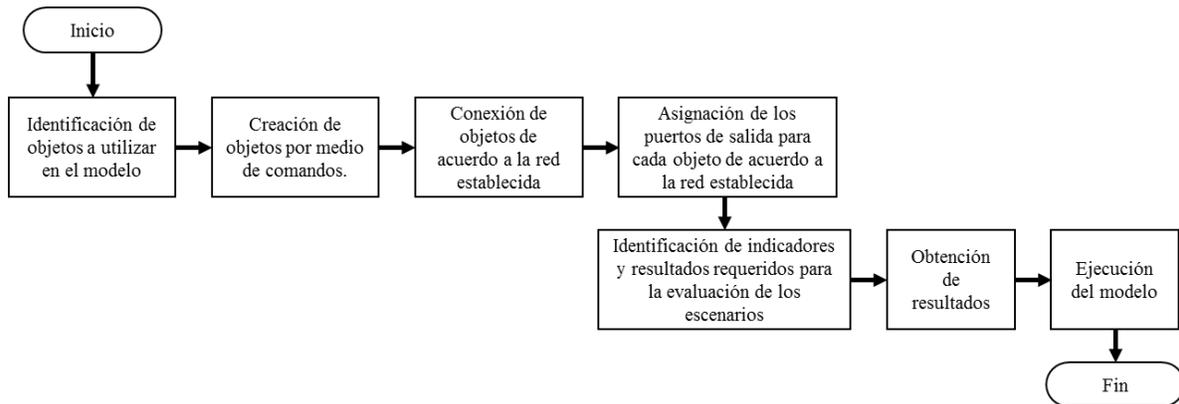
La ejecución del aplicativo se realizó en una PC con procesador Intel®Core™i7 2760QM (2,4GHz) con una memoria interna de 8GB. El código fuente fue escrito usando VBA en Microsoft Excel 2013 usando lenguaje de programación básico.

4.3. Modelo de Simulación

Existen factores externos que afectan el tiempo total de una ruta asignada a un carrotanque en la solución. Estos factores se pueden identificar como congestión en las vías causadas por accidentes o derrumbes. Con el fin de determinar el efecto de estos factores en la solución generada por el aplicativo, se elabora un modelo de simulación que permite identificar el porcentaje en que las rutas de los carrotanques terminan antes del horizonte de planeación estipulado (168 horas).

A continuación, se describe la metodología utilizada para realizar la simulación con el fin de evaluar el rendimiento de los escenarios obtenidos para la solución del problema. Se puede observar la metodología implementada en la realización del modelo de simulación en la Ilustración 7.

Ilustración 6. Metodología modelo de simulación



El desarrollo del modelo fue realizado por programación debido al tamaño del alcance del problema. Para tener un acercamiento a los conceptos del programa empleado y sus componentes, se sugiere consultar el Anexo 2 Descripción FlexSim.

Los componentes empleados se describen a continuación.

➤ Definición de los Componentes

a. Sistema:

- **Sources:** Se emplearon 17 Sources representando cada nodo de descarga en donde iniciará el trayecto de cada carrotanque.
- **Queues:** Se emplearon 88 Queues representando la cola de espera en los nodos de cargue y descarga.
- **Processor:** Se emplearon 88 Processors representando los nodos de cargue y descarga. De igual manera, para lograr representar el tiempo entre los nodos de cargue y descarga, junto con la implementación de las variables estocásticas analizadas estadísticamente, se emplearon 1136 Processors.
- **Sinks:** Se emplearon 17 Sinks representando cada nodo de descarga en donde terminará la ruta de cada carrotanque.

b. Entidades:

- Carrotanques

c. Variable:

- Tiempo de transporte entre un par de nodos.

d. Parámetros:

- Ubicación inicial de los carrotanques.
- Ruta asignada para cada carrotanque.

e. Indicadores de Desempeño:

- Porcentaje de cumplimiento: número de veces en que un carrotanque termina su ruta en un tiempo menor o igual al horizonte de planeación sobre todas las rutas realizadas.

f. Supuestos:

- Se analiza un escenario en donde los carrotanques se encuentran inicialmente en un nodo de descarga, en donde habrán terminado la ruta programada en la semana anterior.
- Los tiempos de cargue y descargue para cada nodo respectivo es tomado como un valor constante.

➤ **Modelamiento**

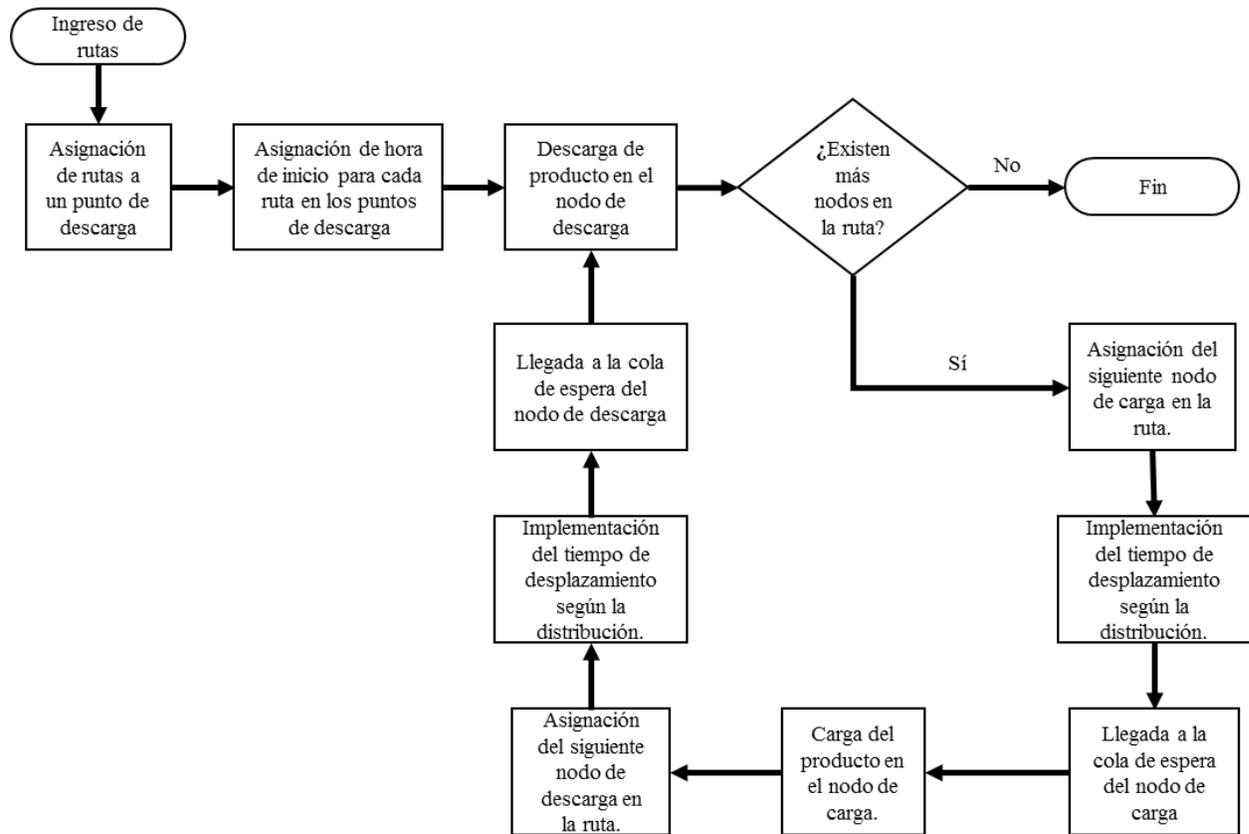
El modelo de simulación requiere de la solución generada por el aplicativo elaborado. Al tener las rutas por cada carrotanque, se realiza la asignación de la cantidad de carrotanques que inician en los nodos de descargue, junto con su hora de inicio. Esta cantidad es asignada a los nodos de descarga iniciales.

Cada carrotanque inicia la ruta en el nodo de descargue correspondiente. Seguido de esto, los carrotanques son enviados al objeto asociado con el tiempo de desplazamiento entre el nodo de descargue actual y el nodo de cargue que continúa en la ruta de la solución. Después de esto, los carrotanques llegan a la cola de espera del nodo de cargue que continúa en la ruta.

Este procedimiento será realizado de nodo de cargue a nodo de descargue, hasta que la ruta del carrotanque termine. En este caso, el carrotanque será arrojado al objeto asociado con el puerto de descargue en el que finaliza la ruta. Todo el flujo del modelo puede ser observado en la ilustración 8.

En el *Anexo 5 Modelamiento de la simulación* se encuentran los pseudocódigos de la programación del modelo de simulación en el programa FlexSim.

Ilustración 7. Flujo de la simulación



5. Componente de Diseño en ingeniería

5.1. Declaración de diseño

Se realizó el diseño de un aplicativo en Excel, el cual propone contratar un número necesario de carrotaques, asignando rutas entre nodos de cargue y descargue, para lograr transportar el crudo extraído de nodos de cargue a nodos de descargue durante un horizonte de planeación de una semana. El aplicativo busca minimizar el costo de transporte incorporando una meta-heurística (Tabú Search) y una heurística (Nearest Neighbor).

5.2. Proceso de Diseño

Para lograr justificar la elaboración del aplicativo, se ejecuta el modelo matemático con una instancia pequeña y una instancia un poco más grande en el solver de optimización Gusek.

Inicialmente se ejecuta el modelo matemático, en una instancia de 3 nodos de cargue, 2 nodos de descargue, 2 empresas prestadoras del servicio de transporte y un horizonte de planeación de 72 horas. Esta instancia tarda en generar una solución óptima factible en 24,6 segundos utilizando una memoria de 4,5Mb.

Finalmente, se ejecuta el modelo matemático con una instancia de 10 nodos de carga, 5 nodos de descarga, 2 empresas prestadoras del servicio de transporte y un horizonte de planeación de 72 horas. Esta instancia

genera una solución factible, NO óptima en 7235.5 segundos utilizando una memoria de 90.7Mb.

El problema solucionado tiene 71 nodos de cargue, 16 nodos de descargue, 5 empresas prestadoras del servicio de transporte y un horizonte de planeación de 168 horas. Se observa que en la segunda instancia ejecutada, el solver no logra encontrar la solución óptima en menos de un tiempo considerable (7200 segundos), la utilización de métodos heurísticos para la solución de este problema, es justificable.

El diseño del aplicativo se realizó en 3 fases:

- a. Definición del problema de optimización
- b. Generación de la solución

a. Definición del problema de optimización:

Los parámetros, variables, restricciones y el objetivo para lograr la solución fueron definidos. A continuación, son listados cada uno de aspectos necesarios para definir el problema.

➤ Parámetros

- Cantidad de nodos de descargue y cargue
- Cantidad de crudo a extraer por semana de los nodos de cargue
- Número de empresas prestadoras del servicio de transporte
- Horizonte de planeación
- Tiempo de duración necesario para ir de un nodo de descargue a un nodo de cargue
- Conexiones permitidas entre los nodos de cargue y los nodos de descargue
- Tiempo de duración necesario para cargar y descargar el crudo
- Cantidad de carrotaques disponibles por empresa prestadora del servicio de transporte en cada nodo de descargue
- Costo fijo de contratación de un carrotaque
- Costos de emplear un carrotaque por hora

➤ Variables

- Número de carrotaques a contratar de cada empresa prestadora del servicio de transporte
- Ruta asignada a cada carrotaque contratado
- Hora de inicio de las rutas asignadas a cada carrotaque contratado

➤ Función Objetivo

- Minimizar el costo de transportar crudo por medio de carrotaques, asociado a la cantidad de carrotaques contratados y el tiempo total de las rutas asignadas a cada carrotaque.

➤ Restricciones

- Las rutas asignadas a los carrotaques deben empezar y terminar en un nodo de descargue
- La hora de inicio de las rutas no puede ser un número negativo
- La hora en la que las rutas terminan debe ser menor al horizonte de planeación (168 horas).

- La cantidad de camiones demandados para extraer el crudo de cada nodo de cargue debe ser cumplida
- Debe cumplirse con el porcentaje de participación mínimo exigido por cada empresa prestadora del servicio de transporte
- El porcentaje esperado de cumplimiento de que las rutas lleguen en un tiempo menor o igual al horizonte de planeación considerando tiempos aleatorios debe ser mayor al 95%.

b. Generación de la solución

Los pasos para lograr la generación de la solución se encuentran resumidos en la *Ilustración 9*.

Ilustración 8. Pasos para generar la solución



5.3. Requerimientos de desempeño

Se consideraron los siguientes requerimientos de desempeño para el diseño del aplicativo:

- El aplicativo debe arrojar una solución más eficiente comparada con la metodología actual.
- El aplicativo debe contar con una interfaz amigable para el usuario, permitiendo la modificación de los parámetros de entrada empleados para su ejecución.
- El aplicativo debe arrojar una solución factible.

5.4. Rendimiento

Se realizaron las siguientes pruebas para verificar el rendimiento del aplicativo:

- Para cada ruta se imprime el orden en el que se recorre cada nodo (Cargue/Descargue) para así garantizar que empiece y termine en un nodo de descargue.
- Se imprime la hora de inicio para cada una de las rutas con el fin de demostrar que esta es no negativa.
- Para cada una de las rutas se imprime la hora fin para contrastar con el horizonte de planeación y ver que no lo supera.
- Al imprimirse el orden en que se recorre cada nodo de cargue y descargue se hace la sumatoria de las veces que se visita cada nodo de cargue con el fin de asegurar que sea el número de veces de su demanda.
- Se imprime la empresa encargada de realizar cada ruta. Se calcula el porcentaje de participación de cada una de las empresas con la suma de las rutas asignadas a ella sobre el total de las rutas y se evalúa que este sea mayor al porcentaje mínimo requerido.

5.5. Restricciones

Las restricciones contempladas en el problema, son las siguientes:

- El porcentaje de participación es una de las restricciones que restringe el espacio de solución ya que puede obligar a ciertas rutas a empezar en un nodo de cargue más lejano por cumplir este requerimiento.
- Debido a que se debe respetar el modelo de planeación que la industria utiliza para asignar nodos de descargue a nodos de cargue, se limita el espacio de búsqueda de mejores soluciones al prohibir la conexión de ciertos nodos de cargue con algunos de descargue.

5.6. Cumplimiento del estándar:

Se garantiza el cumplimiento del estándar ISO 9126 – Evaluación de calidad de software, asegurando las siguientes características en los aplicativos diseñados:

- **Funcionalidad:** El modelo realiza las tareas para las cuales fue desarrollado, observando los resultados de los indicadores y la animación generada.
- **Fiabilidad:** El modelo cumple con los requerimientos de desempeño establecidos. Estos requerimientos fueron comprobados en el apartado de prueba de desempeño.
- **Usabilidad:** Debido a que el modelo de simulación no fue diseñado para la manipulación del usuario, esta característica no es considerada en el cumplimiento del estándar.
- **Mantenibilidad:** Debido a que el modelo de simulación fue realizado por medio de comandos de código, posee un alto nivel de adaptabilidad en cuanto a nuevas especificaciones y requisitos que sean introducidos al planteamiento del problema.

6. Resultados

Aunque el modelo matemático no fue utilizado para encontrar la solución a todo el problema, se plantearon 10 escenarios en donde pequeñas instancias del modelo fueron corridas en Gusek y el aplicativo. En la Tabla 16 se pueden observar los escenarios evaluados.

Tabla 16. Escenarios del problema

Escenario	Nodos de Descargue	Nodos de Cargue	Empresas Transportadora	Horizonte de Planeación (Horas)
1	2	2	2	72
2	2	3	2	72
3	2	4	2	72
4	2	5	2	72
5	3	5	2	72
6	3	6	2	72
7	3	7	2	72
8	4	7	2	72
9	4	8	2	72
10	3	9	2	72

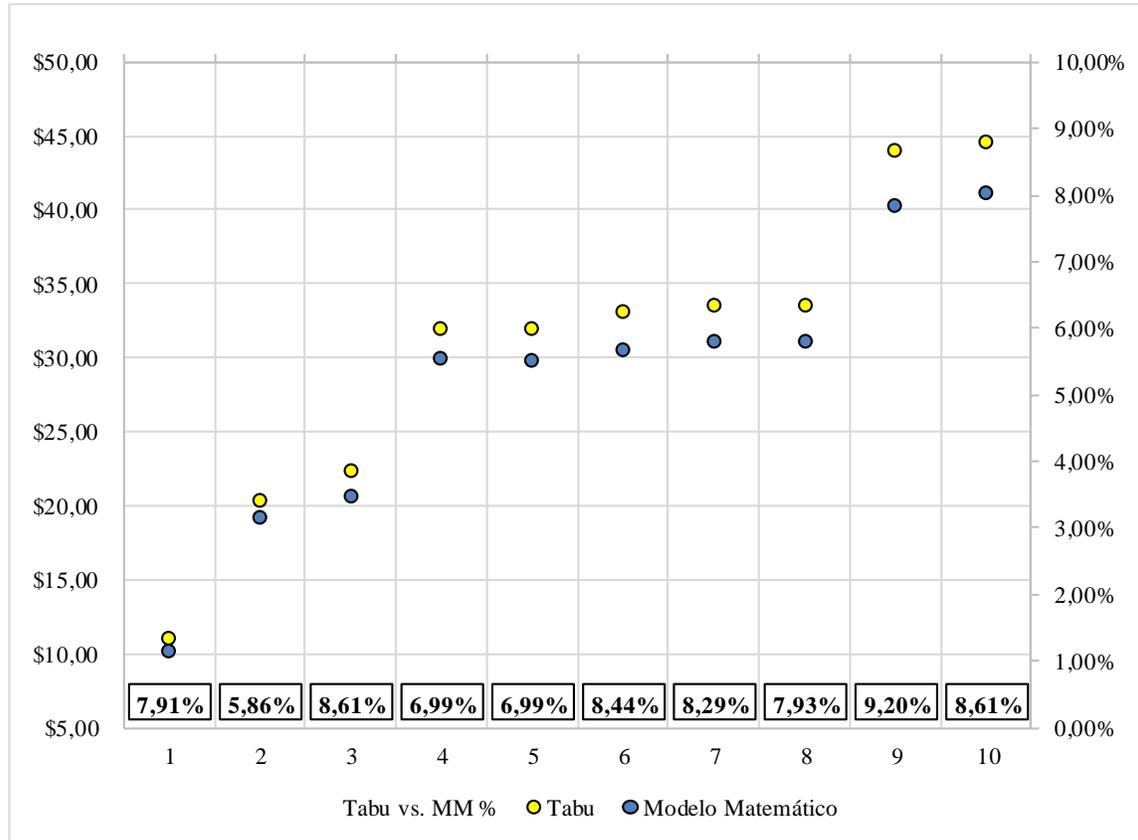
En la Tabla 17 se observan los resultados obtenidos al correr los escenarios en Gusek y el aplicativo, haciendo uso de la heurística del vecino más cercano y la metaheurística Tabu Search.

Tabla 17. Resultados obtenidos de los escenarios

Escenario	Vecino más cercano	Tabu	Modelo Matemático	Vecino más cercano vs. Modelo Matemático	Tabu vs. Modelo Matemático
1	\$ 10,952,757.56	\$ 10,942,343.36	\$ 10,140,449.96	8.01%	7.91%
2	\$ 20,322,556.72	\$ 20,277,428.52	\$ 19,154,777.76	6.10%	5.86%
3	\$ 22,362,698.50	\$ 22,317,570.30	\$ 20,548,892.00	8.83%	8.61%
4	\$ 31,952,584.42	\$ 31,902,943.40	\$ 29,818,020.56	7.16%	6.99%
5	\$ 31,944,947.34	\$ 31,891,487.78	\$ 29,806,564.94	7.17%	6.99%
6	\$ 32,994,351.56	\$ 32,980,813.10	\$ 30,414,059.94	8.48%	8.44%
7	\$ 33,539,014.22	\$ 33,536,931.38	\$ 30,970,178.22	8.29%	8.29%
8	\$ 33,539,014.22	\$ 33,536,931.38	\$ 31,072,584.52	7.94%	7.93%
9	\$ 43,907,882.30	\$ 43,881,152.52	\$ 40,184,458.66	9.27%	9.20%
10	\$ 44,574,043.96	\$ 44,551,827.00	\$ 41,021,009.14	8.66%	8.61%

Finalmente se observa en la ilustración 9 que el gap entre las soluciones obtenidas por la metaheurística contra la solución obtenida en Gusek, no supera el 10% de aumento en la función objetivo. Es importante resaltar que solo los escenarios 1, 2 y 3 lograron encontrar la solución óptima. Para los escenarios 4 – 10 se tomó la solución que Gusek encontró en 2 horas.

Ilustración 9. Tabu Search vs. Modelo matemático



El método actual, la heurística del vecino más cercano y la metaheurística Tabú Search son analizados con el fin de identificar la solución que logre minimizar en mayor cantidad los costos de transporte de crudo.

En la Tabla 18 se observan los resultados obtenidos del aplicativo elaborado. Se obtuvieron los costos de transporte de crudo asociados a las soluciones generadas por la heurística del vecino más cercano y la

metaheurística Tabú Search

Por medio de un análisis estadístico de salida (*Anexo 15 Análisis estadístico de salida*) sobre los resultados del modelo de simulación se obtiene el porcentaje esperado de las rutas que cumplen con la restricción de finalización de ruta dentro del horizonte de planeación y se muestran en la tabla 16.

Tabla 18. Resultados

Resultados	Método actual	Heurística	Metaheurística
Función objetivo	\$ 18.456'164.308	\$ 15.531'090.309	\$ 14.991'098.732
Costo total de horas	\$ 5.674'574.577	\$ 5.731'871.516	\$ 5.661'407.725
Costo total de contratación	\$ 12.781'589.730	\$ 9.799'218.793	\$ 9.329'691.007
Porcentaje de cumplimiento	100.00%	100.00%	100.00%
Porcentaje de cumplimiento (Efectos externos)	99.89%	95.78%	95.14%

Medición del Impacto

Se obtiene el indicador de variación de costos totales, con el fin de medir el impacto de los métodos enunciados.

$CT = \text{Costo Total}$

$X = \text{Cantidad de carrotaques contratados en la solución del método}$

$CC = \text{Costo de contratación de un carrotaque}$

$CH = \text{Costo por hora de recorrido}$

$TR = \text{Tiempo total recorrido de los carrotaques contratados en la solución del método}$

$Var = \text{Variación porcentual entre los métodos analizados}$

$$CT_{\text{Método}} = (X_{\text{Método}} * CC) + (TR_{\text{Método}} * CH)$$

$$Var = \frac{CT_{\text{Método 1}}}{CT_{\text{Método 2}}} - 1$$

$$CT_{\text{Heurística}} = \$ 15.531'090.309$$

$$CT_{\text{Metaheurística}} = \$ 14.991'098.732$$

$$CT_{\text{Actual}} = \$ 18.456'164.308$$

$$Var_{Heurística\ vs.\ Actual} = -15.85\%$$

$$Var_{Metaheurística\ vs.\ Actual} = -18.77\%$$

Se obtiene un costo de \$15.531'090.309 en la solución generada con la heurística de vecino más cercano. Frente al método actual se obtiene un ahorro del 15.85%, es decir una disminución del costo de \$2.925'073.998.

Con relación a la solución obtenida con el Tabú Search, se obtiene un costo de \$14.991'098.732. Frente al modelo actual se logra un ahorro del 18.77%, es decir una disminución de \$ 3.465.065.575 frente al costo generado por el método actual.

Analizando este indicador de varianza de los costos, podemos ver el cumplimiento del objetivo dado que se logra un ahorro de costos de transporte del 18.77% superando el objetivo de 5%.

7. Conclusiones y recomendaciones

Con el método seleccionado, se identifican falencias en el sistema de enrutamiento actual. Según esto evidenciamos una mejora de pasar de 2075 vehículos utilizados a pasar a 1075 vehículos, disminuyendo los costos fijos y mejorando los tiempos de ocupación de los carrotanques.

Comparado con el modelo matemático de optimización, las soluciones generadas por el aplicativo no aumentan en más de un 10% la función objetivo para los 10 escenarios evaluados. Se considera que la diferencia máxima entre la solución del aplicativo y solución óptima no debe ser mayor a un 10% para considerar el método heurístico como uno aceptable, es por esto, que con los resultados obtenidos concluimos que la metaheurística no se encuentra tan lejos de lo que podría ser la solución óptima del problema.

Analizando el desarrollo del sistema actual, cada vehículo en promedio visita tres puntos de carga por semana. Bajo el procedimiento propuesto, cada vehículo tendría menores tiempos improductivos (TO), dado que pasa a atender en promedio 6 puntos de carga.

Para el mejor aprovechamiento de los recursos, se recomienda a la industria de hidrocarburos colombiana implementar estrategias como la propuesta, logrando un ahorro de costos de 18.77% además de tener una planeación semanal de sus transportes terrestres de producto.

Frente a la meta propuesta de encontrar una alternativa la cual brindara una mejor solución por lo menos en un 5% frente a la actual, se puede concluir que, gracias al planteamiento de la heurística, se alcanza un ahorro de \$2.925'073.998 en los costos, además se llega a una mejor solución integrando la heurística dentro de una metaheurística de Tabú Search con la cual se ahorra un 2.92% adicional, superando el objetivo en un 13.77%.

Se propone una metodología para resolver el problema Np-hard MDVRP, en un tiempo computacional inferior a 5 minutos, utilizando la heurística Nearest Neighbors, para luego ser mejorada por una búsqueda tabú, de esta manera se evalúa la calidad de la solución propuesta mediante una simulación y se realizan las respectivas pruebas de asociación estadística para identificar las diferencias significativas entre la solución propuesta actual.

Finalmente se mejora la solución actual más de un cinco por ciento, cumpliendo con las expectativas del proyecto.

Se recomienda realizar un modelo implicando semi-heuristic para enriquecer este proyecto.

8. Glosario

Términos

- **Carrotanque:** Camión que sirve tanto para el transporte de líquidos como para su mantenimiento por un tiempo prolongado según sus características.
- **Crudo:** Conocido como petróleo crudo, es una mezcla de varios hidrocarburos líquidos debido a la descomposición de restos orgánicos (Ecopetrol, 2014).
- **Nafta:** Hidrocarburo líquido que se obtiene por destilación del petróleo y se utiliza como disolvente industrial (Ecopetrol, 2014).
- **Heurísticas:** métodos específicos que están diseñados para resolver un problema o una instancia de un problema.
- **Modelo matemático:** sistema de ecuaciones y expresiones matemáticas relacionadas que describen la esencia de un problema.
- **Meta-heurísticas:** métodos heurísticos para resolver un tipo de problema computacional general, usando parámetros sobre unos procedimientos genéricos y abstractos de una manera que se espere eficiente.
- **IDE:** Integrated Development Environment, es una aplicación informática que proporciona servicios integrales para facilitar el desarrollo de un software.

Abreviaciones

- **MVRPTW:** Multiple Vehicle Routing Problem with Time Windows.
- **MDVRP:** Multiple Depot Vehicle Routing Problem
- **MDPVRP:** Multiple Depot Periodic Vehicle Routing Problem
- **VRP:** Vehicle Routing Problem.
- **MVRP:** Multiple Vehicle Routing Problem.
- **ODVRPTW:** Oil Delivery Vehicle Routing Problem with Time Windows.
- **LNS:** Large Neighborhood Search.

- **TS**: Tabú Search.
- **CG**: Column Generation.
- **HGA**: Hybrid Genetic Algorithm.
- **MDOVRP**: Multiple-Depot Open Vehicle Routing Problem.
- **SA**: Simulating Annealing.
- **MTVRPTW**: Multiple-Trip Vehicle Routing Problem with Time Windows.
- **SI**: Solomon Insertion.

9. Tabla de Anexos

No. Anexo	Nombre	Desarrollo	Tipo de archivo	Enlace Corto (https://goo.gl/)	Relevancia para el documento (1-5)
1	Registro análisis de datos	Propio	Excel	https://goo.gl/euK47g	2
2	Descripción Flexsim	Propio	PDF	https://goo.gl/JU4qE4	3
3	Matriz asignación de rutas	Propio	Excel	https://goo.gl/BUo4BZ	3
4	Matriz asignación de tiempos	Propio	Excel	https://goo.gl/PtHX7V	3
5	Modelamiento de la simulación	Propio	PDF	https://goo.gl/mhMSsv	4
6	Análisis de datos de entrada	Propio	PDF	https://goo.gl/viLm1	4
7	Modelo de simulación	Propio	FSM	https://goo.gl/PGyPm7	5
8	Vídeo simulación modelo actual	Propio	MP4	https://goo.gl/MbMY5W	3
9	Vídeo simulación modelo propuesto	Propio	MP4	https://goo.gl/eSuGMq	3
10	Gusek Modelo Matemático	Propio	MOD	https://goo.gl/yN6dSV	4
11	Aplicativo método actual	Propio	Excel	https://goo.gl/rV4xiQ	1
12	Aplicativo heurística	Propio	Excel	https://goo.gl/aPma84	1
13	Aplicativo Meta-heurística	Propio	Excel	https://goo.gl/bxBHMJ	1
14	Análisis de Salida	Propio	Excel	https://goo.gl/aXyHY2	3
15	Análisis estadístico de salida	Propio	Word	https://goo.gl/aBWxeP	4

Referencias

- Cantu-Funes, Roberto., Salazar-Aguilar Angélica., y Boyer, Vincent (2017). Multi-Depot Periodic Vehicle Routing Problem with Due Dates and Time Windows. Journal of the Operational Research Society.
- Salah, Mohammed., Soeanu, Andrei., Rau, Sujoy., y Debbabi, Mourad (2012). Mechanism Design for Decentralized Vehicle Routing Problem. Concordia University.
- Cornillier, Fabien., Boctor, Fayeze., y Renaud, Jacques (2012). Heuristics for the multi-depot petrol station replenishment problem with time Windows. ELSEVIER

- Benantar, Abdelaziz., Ouafi, Rachid., y Boukachour, Jaouad (2016). A petrol station replenishment problem: new variant and formulation. Springerlink.
- Cornillier, F., Boctor, FF., Laporte, G., y Renaud, J. (2008). An exact algorithm for the petrol station replenishment problem. Journal of the Operational Research Society.
- Oudiuar, Fadoua., y El-Fellahi, Abdellah. (2017). Solving the location-routing problems Using Clustering Method. National School of Applied Sciences Abdelmalek Essaadi University.
- Banks, Jerry (1999). INTRODUCTION TO SIMULATION. AutoSimularions.
- Jiang, Jun., Ming-Ng, Kien., Leng-Poh, Kim., y Meng Teo, Kwong. (2014). Vehicle routing problem with a heterogeneous fleet and time windows. ELSEVIER.
- Amador-Fontalvo J. E. , Paternina-Arboleda C. D. y Montoya-Torres J. R. (2014). Solving the heterogeneous vehicle routing problema with time windows and multiple products via a bacterial meta-heuristic. Int. J. Advanced Operations Management, Vol. 6, No. 1.
- Angel, A. J. , Grasman, S.E. , Faulin J., Riera D. , Méndez C.A. y Ruiz B. (2009). Applying Simulation and Reliability to Vehicle Routing Problems with Stochastic Demands. 16th International RCRA workshop.
- Camiones US. (2016) Que es un camión cisterna. Recuperado de http://www.camiones.us/transportes/carga/seguro/que_es_un_camion_cisterna/Cuatro_rios_tecnologías.
- Cuatro rios tecnologías (2016), Norma ISO-9126 para análisis de software. Recuperado de http://www.cuatrorios.org/index.php?option=com_content&view=article&id=163:norma-is-o-9126-para-analisis-de-software&catid=39:blogs-feeds
- Dayariana I., Crainic T. G. y Gendreau M., Rei W. (2014). A column generation approach for a multiattribute vehicle routing problem. ELSEVIER.
- Despau F. y Basterrech S. (2014). A Study of the Multi-Trip Vehicle Routing Problem with Time Windows and Heterogeneous Fleet. International Conference on Intelligent Systems Design and Applications.
- Ecopetrol (2014). Ecopetrol. Recuperado de <http://www.ecopetrol.com.co>
- Ecopetrol (2016). Tarifas carro tanques. Recuperado de <http://www.ecopetrol.com.co/documentos/TarifasCarrotanques.pdf>
- Ecured (2016). IDE de Programación. Recuperado de http://www.ecured.cu/IDE_de_Programaci%C3%B3n
- Figueira G. y Almada-Lobo B. (2014). Hybrid simulation – optimization methods: A taxonomy and discussion. ELSEVIER.
- Hernández F, Feillet D. , Giroudeau R. y Naud O. (2015). Branch-and-price algorithms for the solution of the multi-trip vehicle routing problem with time Windows. ELSEVIER.
- Liu R., Jiang Z. y Geng N. (2012). A hybrid genetic algorithm for the multi-depot open vehicle routing problema. Springer-Verlag.
- Prescott-Gagnon E. y Desaulniers G., Rouseau L. (2012). Heuristics for an oil delivery vehicle routing problem. Springer Science+Business Media New York.
- Takes F. (2010). Applying Monte Carlo Techniques to the Capacitated Vehicle Routing Problem. Leiden University, The Netherlands.
- Unidad de planeación minero energética. (2013). Cadena del petróleo 2013. Recuperado de http://www1.upme.gov.co/sites/default/files/news/3086/files/cadena_del_petroleo_2013.pdf
- Tatsushi Nishi y Tsukasa Izuno, (2014). Column generation heuristics for ship routing and scheduling problems in crude oil transportation with split deliveries. ELSEVIER
- Ren, Yingtao., Dessousky, Maged y Ordoñez, fernando. (2010). The multi-shift vehicle routing problem with overtime. ELSEVIER.

Cornillier, Fabien., Boctor, Fayezy Renaud, Jacques (2012). Heuristics for the multi-depot petrol station replenishment problem with time windows. ELSEVIER.

Weyland, Dennis., Salani, Matteo., Montemanni, Roberto y Luca Maria Gambardella. (2013). Vehicle Routing for Exhausted Oil Collection. DalleMolle Institute for Artificial Intelligence (IDSIA, USI-SUPSI), University of Applied Science of Southern Switzerland.

Prescott-Gagnon, Eric., Desaulniers, Guy., y Rousseau, Louis -Martin. (2014). Heuristics for an oil delivery vehicle routing problem. Springer.

Koskosidis, Yiannis, A., Powell, Warren, B., y Salomon, Marius, M. (1992). An Optimization-Based Heuristic for Vehicle Routing and Scheduling with Soft Time Window Constraints. Transportation science.

Drexl, Michael (2014) A Generic Heuristic for Vehicle Routing Problems with Multiple Synchronization Constraints. Johannes Gutenberg University Mainz.

Xiaocun Mao, Xinxin Zhang, Zhenping Li (2016). Full Load Distribution Scheduling Problem for Refined Oil. Science publishing group.