

[193003] Programación de taladros de perforación para campos petroleros en fase de desarrollo

David Leonardo Beltrán Huertas ^{1a,c}, Juan Camilo Daza Acevedo ^{2a,c}, Daniel Felipe González Rivera ^{3a,c},

Santiago Leonardo Arrubla Bernal ^{b,c}

^aEstudiante de Ingeniería Industrial

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

^cPontificia Universidad Javeriana, Bogotá, Colombia

Abstract

Well drilling planning is one of the largest scheduling activities (more than 10 years) and has high operational costs. This planning must be effective through the whole operation and consider factors that could change along the performance. This research presents a scheduling program using optimization techniques, considering operation costs and delay in drilling activities. In addition, this tool is capable of reprogramming initial planning, taking into account drilling activities that had been made, positions of drills, and variation of activities along the years. Initial planning is simulated using drill's costs, oil wells locations and sequences of activities related with a greedy algorithm. Regarding this, PSO algorithm is implemented to improve solutions and to create a solution frontier. Then, one solution is reprogramming using a dynamic algorithm, which allows to make a new scheduling every year, taking into account drilling activities that had already been performed and stochastic sequence of the activities. The schedule program was validated with different simulations and a validation program that verifies if all restrictions were respected. The experimental results demonstrate that the initial scheduling is more expensive than planning all over again, and that the proposed method provides a good solution for schedule problems.

1. Justificación y planteamiento del problema.

El sector de hidrocarburos es un sector en constante cambio y desarrollo que ha contribuido aproximadamente un 9% al PIB entre los años 2010 y 2017. En la última década, uno de cada cuatro pesos en finanzas públicas se deriva del sector de los hidrocarburos. Esta industria contribuye al progreso social en Colombia, donde encontramos mejoras, construcción de infraestructura y servicios públicos, etc. Según la Agencia Nacional de Hidrocarburos, en Colombia las reservas del crudo son de 6,2 años al cierre del 2018 por lo que se tendrá petróleo hasta el 2024, de modo que existe una necesidad de explorar nuevos campos petroleros para el sostenimiento de la industria. (Dinero, 2018).

Dada la importancia de los hidrocarburos a nivel tanto nacional como global, nos fijamos en el proceso de intervención de pozos en campos petroleros en fase de desarrollo. Este consta de diferentes actividades como la perforación, el abandono y reacondicionamientos o *workovers*, estos últimos consisten en modificar el estado de un yacimiento, teniendo como objetivo la reparación o trabajo de pozos en estos. Para la realización de estas actividades se cuentan con equipos especializados de perforación (taladros), los cuales, de acuerdo con su potencia, pueden ejecutar un mayor rango de operaciones, lo que a su vez implica un mayor costo de funcionamiento. El proceso de intervención inicia en la localización de los yacimientos, seguido por un estudio del área para determinar los límites del campo y finaliza con un análisis de impacto ambiental y social. Posteriormente, cuando el terreno ha sido preparado, se da paso a la instalación de la plataforma, donde se arma el taladro para comenzar el proceso de horadar el pozo principal, en esta primera parte del proceso se realiza un

orificio de mayor diámetro y menor profundidad, para seguidamente comenzar con la perforación del pozo hasta alcanzar la profundidad donde se encuentra el yacimiento(Freudenrich & Strickland, 2001).

La propuesta de valor de este trabajo está centrada en generar un control en la planificación de los taladros utilizados para la perforación, *workover* o abandono de los pozos petrolíferos, de manera que se ajuste a la aleatoriedad del proceso y se tenga en cuenta que para el siguiente período de tiempo se conocen los pozos que no han sido terminados, así como sus actividades realizadas y las localizaciones donde terminaron los equipos de perforación en el período anterior. De este modo se logra obtener control de los costos y de la tardanza generada por el desfase de tiempo entre el momento programado para la inicialización de cada actividad y el tiempo de inicio ejecutado teniendo en cuenta que el desfase siempre es positivo, es decir, un taladro no puede iniciar una actividad antes del momento planeado. Esta planificación la llamaremos reprogramación y el desfase lo denominaremos tardanza.

La reprogramación se puede dar por tres factores. El primero se presenta en caso de que el tiempo de perforación del pozo sea mayor a un rango estimado al inicio de la simulación, hallado con base en el análisis de datos históricos, lo cual lleva a tomar la decisión de continuar con su perforación retrasando la operación o detener el proceso en ese pozo y continuar con la programación establecida. El segundo caso se da cuando al realizar las pruebas de viabilidad (si hay o no hay crudo en el pozo), no se encuentre petróleo en ese yacimiento, generando una nueva planeación. El último caso es cuando se incorpora un nuevo punto de perforación dentro del campo. Haciendo lo anterior, y considerando que para cada reprogramación varía la ubicación inicial de los taladros, se busca dar dinamismo a la operación, siendo conscientes de los cambios que se producen a lo largo del proceso. Un ejemplo propuesto de la ruta planeada en cada período se puede observar en la figura 1, donde se muestra los puntos de inicio de la planeación y la ruta propuesta para cada período de tiempo, además, en los períodos siguientes se observan marcados con una X los yacimientos que se perforaron en los períodos anteriores o los cuales después de realizarse las pruebas fueron descartados, por lo cual, no se tienen en cuenta en la programación de periodos futuros.

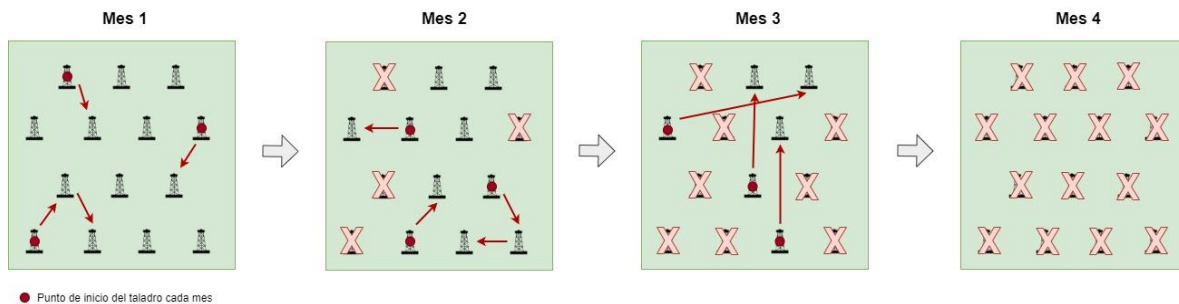


Figura 1. Ejemplo de una solución propuesta. Fuente: Autoría propia

Dentro del planteamiento del problema se tendrán en cuenta dos conjuntos principales, los cuales son el número de pozos del campo petrolífero y la cantidad de taladros por tipo que se posee. Además, se tendrán en cuenta diferentes parámetros como la distancia entre pozos, la secuencia estocástica de actividades del pozo, la tarifa por usar o no el taladro, el costo de armado y desarmado de taladro, el costo de trasladarse de un pozo a otro y las actividades que puede realizar cada tipo de taladro.

Cabe añadir que este proyecto se realizará basado en el comportamiento de campos como La Cira-Infantas y por ende los datos de estudio serán simulados, por lo que se diseñará un modelo equivalente determinístico con el cual se evaluará finalmente el rendimiento de la propuesta. Debido a que el problema depende de diferentes variables como el proceso, el campo y su estado, etc. No es posible la medición de un costo estimado con respecto a una referencia, el cual se dificulta por la incertidumbre del proyecto. Dicha medición solo se tendrá

cuando se finalice el proyecto real. Basándose en esta problemática nos planteamos la siguiente pregunta, ¿Cuál es el impacto de la reprogramación en el costo total y la tardanza?

2. Antecedentes

En la actualidad se hace más evidente la necesidad que tienen las organizaciones de optimizar los recursos, esto se puede evidenciar en la gran cantidad de publicaciones que se han desarrollado en los últimos años referente a este tema. Tan solo en la base de datos de ProQuest la palabra “optimización” cuenta con más de 750.000 artículos publicados en la última década y más de 100.000 en lo que va del presente año. Por lo anterior se denota una amplia aplicación del tema, por lo que a su vez, se observa una creciente gama de oportunidades de realizar investigaciones en este tema para mitigar impactos negativos y maximizar los beneficios.

La investigación de operaciones es una disciplina que utiliza técnicas de modelos matemáticos, análisis estadísticos y optimización matemática con el objetivo de alcanzar las soluciones óptimas, teniendo como propósito, además, apoyar el proceso de toma de decisiones. En esta ciencia aplicada existen técnicas exactas que resuelven las cuestiones concernientes a la optimización del problema, desafortunadamente, estas metodologías solo son exitosas en problemas muy acotados y de baja-media complejidad. Para problemas de alta complejidad se usa una gran variedad de técnicas denominadas heurísticas, hiperheurísticas y metaheurísticas. A continuación, se muestra una revisión literaria de los diferentes métodos de optimización.

Método	Definición y características Principales	Ventajas	Desventajas
Programación Lineal (PL) (Ingeniería Industrial online, n.d.)	Es un algoritmo basado en modelos matemáticos que resuelve problemas de la vida real. El objetivo es optimizar, es decir, minimizar o maximizar funciones lineales con restricciones lineales. Actualmente existen diferentes métodos de solución como el simplex, Punto Interior entre otros.	<ul style="list-style-type: none"> Encuentra la solución óptima del problema. Algunos problemas se pueden linealizar. Softwares con interfaces amigables como lo es el <i>solver</i> de <i>Excel</i>. 	<ul style="list-style-type: none"> Muchos modelos no se pueden linealizar. Cuando el problema es de alta complejidad, o tiene muchas variables no es factible.
Programación Dinámica (Chaer, 2015)	La programación dinámica tiene como objetivo calcular el valor esperado en el futuro, calculando en cada etapa de tiempo ese valor esperado.	<ul style="list-style-type: none"> Mejora la eficiencia en la solución de los problemas. Más fácil de abordar los problemas. Buenos resultados en términos de función objetivo. Los cálculos de cada subprograma se guardan de manera eficiente, garantizando su consulta en un análisis posterior. 	<ul style="list-style-type: none"> Descomponer los problemas puede ser muy complejo. No puede ser aplicable a todo tipo de problemas.
Colonia de Hormigas (Blum, Puchinger, Raidl, & Roli, 2012)	Se basa en la estructura de la colonia de las hormigas. Buscan el mejor recorrido desde el nido hasta el sitio en el que se encuentra su alimento. La idea es imitar este comportamiento y el rastro de la feromona que deja cada una de	<ul style="list-style-type: none"> Se logra llegar a buenas soluciones rápidamente. Ninguna ruta se hace dominante, lo que hace la exploración más diversificada. Utilizado para resolver problemas de optimización 	<ul style="list-style-type: none"> Los resultados del algoritmo inicial tienden a ser inferiores a los de otros algoritmos. No existe forma de saber qué tan cerca está de la solución óptima.

	<p>ellas para encontrar buenas soluciones a problemas de optimización. Tiene como características:</p> <ol style="list-style-type: none"> 1) Tiempo y espacios discretos. 2) Durante cada intervalo de tiempo los individuos de la colonia se desplazan a la misma velocidad. 3) Cada individuo deposita una cantidad de feromona. 4) El criterio que dispone una hormiga para decidir el camino a tomar se determina por el mayor grado de feromona acumulado. 	<p>combinatoria de tipo estocásticos y dinámicos.</p>	
<p>Enjambre de partículas-PSO (Yao & Xu, 2007)</p>	<p>El procedimiento de búsqueda se basa en la inteligencia social, donde hay un líder que ejerce influencia sobre los miembros del grupo, donde estos pueden inferir decisiones o descubrir alternativas que permitan alcanzar el objetivo de una mejor manera que la mostrada por el líder. De esta manera, se realiza un procedimiento de búsqueda en el espacio de soluciones que explora simultáneamente varios subespacios que se encuentran próximos entre sí.</p>	<ul style="list-style-type: none"> • Su tiempo de búsqueda es bastante rápido. • Explora un gran número de soluciones y encuentra soluciones de buena calidad. • En general encuentra buenas soluciones en cualquier sistema aplicable. 	<ul style="list-style-type: none"> • Difícil de codificar en sistemas complejos con respecto a otras metaheurísticas. • No existe forma de saber qué tan cerca está de la solución óptima.
<p>Genético (Ramírez, 2018)</p>	<p>Este algoritmo metaheurístico es basado en las estrategias evolutivas, principalmente entre el cruce y la mutación. Trabaja sobre soluciones al problema que van evolucionando a través de la generación, hasta que converge al individuo que optimiza la función objetivo del problema. Su secuencia de pasos es la siguiente:</p> <ol style="list-style-type: none"> 1) Genera una población inicial, después de elegir la codificación. 2) Calcula la función objetivo de cada configuración de la población. 	<ul style="list-style-type: none"> • Opera de forma simultánea con varias soluciones. • Permite una exploración de un amplio horizonte de soluciones. • Encuentra buenas soluciones cuando el problema tiene múltiples parámetros. 	<ul style="list-style-type: none"> • Puede tardar mucho en converger dependiendo de la cantidad de parámetros que se utilicen en el tamaño de la población. • La codificación depende, de la naturaleza de las variables de decisión. Cuando estas variables son mixtas (binarias y reales) es bastante difícil realizar una codificación binaria de variables reales. • No existe forma de saber

	<p>3) Realiza selección.</p> <p>4) Realiza recombinación.</p> <p>5) Realiza mutación y genera la nueva población de la siguiente generación.</p> <p>6) Si los criterios de parada no se cumplen el proceso regresa al paso 2.</p>		<p>qué tan cerca está de la solución óptima.</p>
<p>Búsqueda Tabú (Chen, Tan, Qian, & Chen, 2018)</p>	<p>La búsqueda Tabú es un procedimiento metaheurístico utilizado para dominar un algoritmo de búsqueda local con el propósito de evitar que el proceso quede en un óptimo locales.</p> <p>Se basa en seleccionar de modo agresivo el mejor movimiento en su vecindad. Cada movimiento es guardado en una memoria de corto o largo plazo con el fin de no explorarse nuevamente. Esto fuerza a la búsqueda a explorar otros sectores del espacio de búsqueda.</p>	<ul style="list-style-type: none"> ● Sencillo de implementar, con respecto a otras metaheurísticas ● Usa la memoria para salir de los óptimos locales. ● Múltiples aplicaciones en inteligencia artificial. ● Por lo general, encuentra buenas soluciones si la solución inicial es buena. ● Lleva un registro histórico de soluciones visitadas. 	<ul style="list-style-type: none"> ● La eficiencia depende de la solución inicial. ● Gran esfuerzo computacional. ● No existe forma de saber qué tan cerca está de la solución óptima.
<p>Algoritmo Weiszfeld (Cañavate Bernal, Cobacho Tornel, & Rodríguez Gómez, 2002).</p>	<p>El algoritmo Weiszfeld de localización es un método iterativo que generalmente converge en el punto óptimo, que busca resolver el problema de Weber centrado en establecer la ubicación de una fábrica. Esta busca la mejor posición de una fábrica respecto a la prioridad que tenga con sus almacenes.</p>	<ul style="list-style-type: none"> ● Llega a la solución óptima la mayoría de las veces. ● Sencillo de implementar. ● Poco esfuerzo computacional 	<ul style="list-style-type: none"> ● La solución llega a fallar si en una iteración el punto encontrado corresponde a la localización de uno de los almacenes convergiendo a este punto.

Un estudio realizado por Elham Damghanijazi y Arash Mazidi, tiene como objetivo realizar la comparación entre la programación dinámica y cinco algoritmos metaheurísticos que incluyen Hill Climbing, Simulated Annealing, PSO, colonia de hormigas y algoritmo genético, para la solución del problema del viajero (TSP por sus siglas en inglés “travelling salesman problem”) en diez nodos. La comparación se realiza en términos de solución óptima, tiempo de ejecución y memoria. Como resultado la programación dinámica es un método preciso. Por procedimientos heurísticos se obtienen resultados aproximados en un tiempo y espacio de memoria menor. En síntesis, se reconoce que la programación dinámica y el algoritmo genético obtienen los mejores resultados (Damghanijazi & Mazidi, 2017).

Un estudio realizado por Blanca Guillén enfocado en el reparto de una cantidad de productos en diferentes tiendas, con el objetivo de encontrar las rutas que generen el menor costo, basado en el problema de rutas de vehículos VRP (vehicle routing problem) y algunas de sus variantes. El VRP consiste en determinar el conjunto óptimo de rutas que realiza un grupo de vehículos para suplir a un grupo de clientes, siendo uno de los problemas

de optimización más estudiados. Este desprende diferentes variantes, entre ellas se encuentra el problema de rutas de vehículos con ventanas de tiempo (VRPTW por sus siglas en inglés “vehicle routing problem with time windows”), consiste en que cada cliente tiene una ventana de tiempo asociada, no respetar dichas ventanas de tiempo corresponden a un sobrecosto dentro de la operación. Llegando a la conclusión de que entre más características toma en cuenta el problema más acertada es la solución encontrada (Guillén, 2017).

Adicional a lo anterior, una investigación desarrollada por Edwin Montes orientada a la resolución del VRPTW en diferentes instancias variando la cantidad de clientes (entre 25 y 100) mediante el uso de metaheurísticas tales como el algoritmo genético, colonia de hormigas, entre otros y teniendo énfasis en híbridos basados en estas. Como resultado todas la metaheurísticas tiene un buen desempeño, sin embargo, destaca el empleo de menos recursos computacionales para el algoritmo de colonia de hormigas – genético (Montes,2017).

Los autores Danesh, Ramli, Rahman, Iberahim & Qudeiri proponen un modelo detallado sobre la optimización en perforación de orificios, que tiene como objetivo encontrar una ruta óptima usando una máquina de perforación en un área determinada. Esta investigación se centra en el desarrollo del algoritmo de optimización de colonias de hormigas (ACO), para calcular el posicionamiento de la máquina a lo largo de los ejes x, y, z. En este estudio se realizaron diferentes series de simulaciones que permitieron comparar la influencia de parámetros como el peso de la heurística, la diferencia en el valor del nodo, valor inicial de la población de hormigas y número de iteraciones en cada una de las simulaciones. Teniendo como resultado una evidente relación entre cada uno de los diferentes parámetros y a su vez la ruta más corta para la perforación de 6, 10 y 12 agujeros utilizando dicha máquina (Narooei, Ramli, Rahman, Iberahim, & Qudeiri, 2014).

Por otra parte, los tiempos requeridos para perforar hasta la profundidad total del pozo petrolero, dependen de la velocidad de penetración de la broca y de evitar demoras causadas por lutitas, tuberías de perforación atascadas, pérdida en el fluido de circulación, el control de maquinaria y cualquier propuesta metodológica que minimice los tiempos de perforación, dando como resultado una disminución en los costos de operación (Caenn, Gray, & Darley, 2011). Además, un estudio realizado por Hernández y Lemus tuvo como objetivo establecer el modelo de control para un pozo de referencia, ubicado en la cuenca del Valle del Magdalena Medio en el municipio de Puerto Boyacá-Colombia. Como resultado se determinó que el modelo adecuado para controlar la tasa de perforación es un ARFIMA con un 95% de confianza (Hernández Martínez & Lemus Polanía, 2015).

Otro artículo relacionado fue realizado por Reginald, Ilker, Corrinne y Dirk (Dewil, Küçükoğlu, Luteyn, & Cattrysse, 2019), el cual busca encontrar el mejor camino para realizar un taladrado en tarjetas de manufactura. Dentro de este artículo se resaltan los diferentes métodos de optimización que se han usado para resolver esta problemática, la cual muestra, grosso modo, cuáles son los métodos más utilizados para dar una solución, en los cuales se resaltan los métodos PSO (Particle swarm optimization), GA (Genetic Algorithm) y ACO (Ant colony optimization)^[1].

Ahora bien, el artículo desarrollado por Johan Aranda y Javier Orjuela nos muestra detalles acerca de la optimización multiobjetivo, la cual es definida como un problema de optimización que presenta múltiples funciones objetivo, las que a su vez son optimizadas bajo un conjunto de soluciones factibles. Esto debido a que un problema multiobjetivo no cuenta con una única solución óptima. El problema de optimización multiobjetivo general se evidencia en la siguiente ecuación:

$$\text{Min}F(x) = (F_1(x), F_2(x), \dots, F_k(x))$$

A este conjunto de soluciones se le denomina frontera de Pareto el cual consta de los puntos en el espacio de búsqueda que corresponden a las soluciones óptimas no dominadas. Una solución no dominada corresponde a cuando no es posible encontrarse una solución que mejore uno de los objetivos sin afectar negativamente a otro. La totalidad de soluciones pertenecientes a la frontera son igualmente buenas, la selección final de la solución corresponde a la cantidad de recursos que se posean y al nivel de interés entre las diferentes funciones objetivos (Pinilla & Castro, 2015).

Para finalizar, es necesario resaltar que no se encontraron antecedentes que resuelvan exactamente la problemática planteada en pozos petrolíferos. Además, nuestra propuesta se va a centrar en desarrollar un modelo de optimización para dar solución al ruteo de los taladros teniendo en cuenta los diferentes tipos de

taladros, las actividades, la secuencia estocástica de las actividades, entre otras. Esto con el fin de tomar las decisiones correctas en el momento adecuado.

3. Objetivos

Objetivo General

Diseñar un algoritmo de optimización para la programación de taladros en campos petroleros en fase de desarrollo, considerando el número y tipo de taladros, la secuencia estocástica de las actividades, así como sus coordenadas iniciales y localización de cada pozo objetivo o a ser intervenido.

Objetivos Específicos

1. Analizar algoritmos de optimización considerando la mejor solución para la problemática.
2. Calcular la solución del equivalente determinístico que se tiene en el ruteo de taladros para pozos petrolíferos.
3. Definir el comportamiento de la variable estocástica.
4. Evaluar y comparar el impacto económico de la solución propuesta, respecto al equivalente determinístico.

4. Metodología

La figura 2 representa la metodología diseñada para este trabajo. Se resumen las fases que están asociadas a las diferentes actividades (DMAIC), y a qué objetivo está relacionada cada fase.

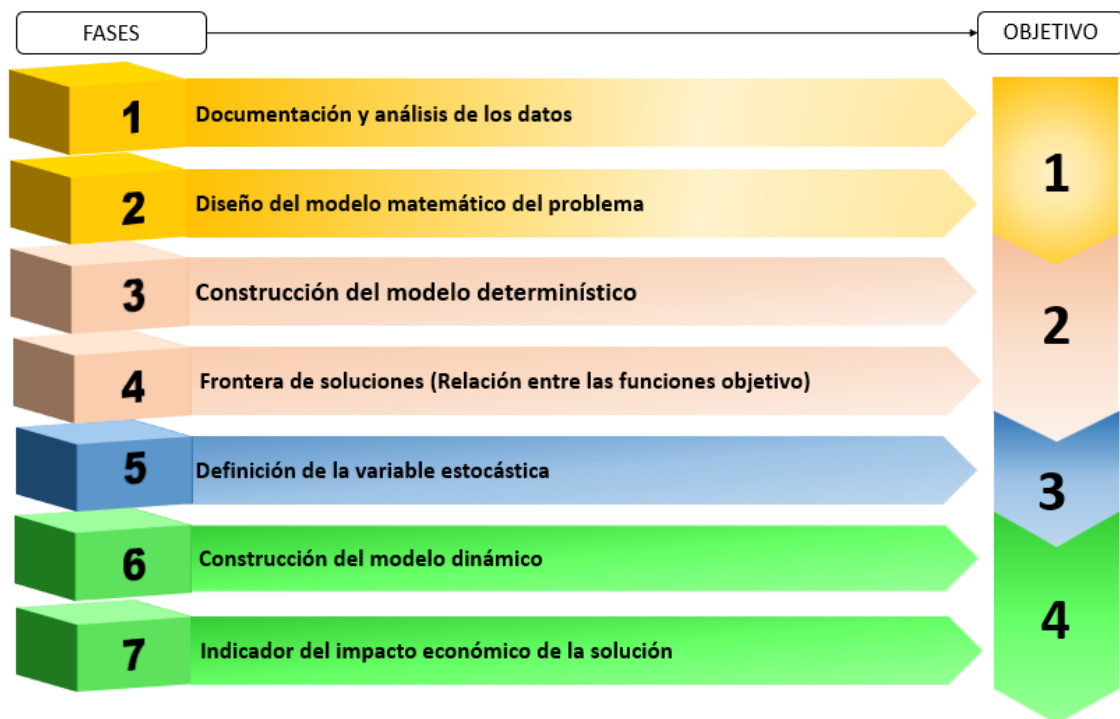


Figura 2. Metodología del trabajo de grado. Fuente: Autoría Propia

4.1 Fase 1. Documentación y análisis de los datos

Se obtuvieron por medio de nuestro director de trabajo de grados la siguiente información simulada de un campo petrolífero: El número de pozos a intervenir, indicando las coordenadas cartesianas de cada uno de ellos con una relación de distancia de 300 metros a 1; 10 tipos de actividades que se pueden realizar a los diferentes pozos según se haya establecido la operación y la duración estimada de cada una en días. Estas actividades

corresponden a: perforación, *workovers* (e.g. profundización de bomba, cambio de bomba, profundización de bomba y cambio de bomba, estimulación, fracturamiento, etc.) y abandono de pozos. Además, se tiene establecida la secuencia de actividades a realizar en los diferentes pozos, así como el tiempo, programado en días, en el que la actividad debe iniciar. Adicionalmente, se tiene información de que los pozos deben estar agrupados en plataformas según su cercanía. Estas agrupaciones conocidas como clústeres, no se encuentran establecidas en los datos iniciales.

Por otro lado, se cuenta con cuatro tipos de equipos especializados (taladros), cada tipo cuenta con información concerniente a qué clase de actividades puede realizar, su cantidad de taladros, el costo de operar [Dólar/h], el costo de armado [Dólar/h], el costo de desarmado [Dólar/h], costo de movilización [Dólar/h], costo de *stand by* [Dólar/hora], el tiempo de armado [días] y desarmado [días] al movilizarse en diferentes plataformas, el tiempo de armado [hora] y desarmado [hora] al movilizarse en la misma plataforma, y la velocidad de movilización [m/hora] o de transporte. Para la uniformidad de la información, se convirtieron las unidades de tiempo a días. Finalmente, y no menos importante, los taladros no cuentan con una localización inicial dentro del campo.

4.2 Fase 2. Diseño del modelo matemático determinístico del problema

4.2.1 Puntos iniciales

Como se comenta anteriormente, no se tiene una posición de inicio para los taladros dentro del campo petrolífero, por esta razón, y con el fin de buscar una solución óptima, se utiliza el algoritmo de Weiszfeld (Cañavate Bernal et al., 2002), adaptándolo al problema presentado y buscando un punto inicial para los taladros según su tipo. El algoritmo asigna un peso al número de actividades que puede realizar el tipo de taladro en cada pozo. Teniendo definido el peso, se ejecutó el algoritmo Weiszfeld para cada tipo de taladro, inicializando con la ecuación de centro de gravedad para la iteración 0. En esta ecuación, w_i es el peso de cada pozo, a_i y b_i las coordenadas X y Y respectivamente.

$$x_0 = \frac{\sum w_i a_i}{\sum w_i} \quad y_0 = \frac{\sum w_i b_i}{\sum w_i}$$

Posteriormente, se itera el algoritmo con las ecuaciones (1), hasta que la diferencia entre cada iteración sea menor al criterio escogido, con el fin de encontrar una posición de inicio óptima del taladro. Al obtener el punto en coordenadas cartesianas, se calculó la distancia euclidiana de cada pozo con respecto al punto encontrado y se asignó el pozo con menor distancia como el punto inicial de cada tipo de taladro. Los puntos iniciales se toman como base para la elaboración del ruteo, se espera que el ruteo se dé desde cualquier punto inicial donde se encuentren los taladros de tal forma que no se limite a un solo punto establecido.

$$g_i(x, y) = \frac{w_i}{\sqrt{(x - a_i)^2 + (y - b_i)^2}}$$

$$x = \frac{\sum a_i g_i(x, y)}{\sum g_i(x, y)} \quad y = \frac{\sum b_i g_i(x, y)}{\sum g_i(x, y)}$$

Ecuación 1. Ecuaciones algoritmo Weiszfeld

4.2.2 Plataformas (Clústeres)

Como se mencionó anteriormente, es necesario asignarle una agrupación inicial a los pozos de manera que se construyan plataformas. Estas plataformas se encargan de discernir los costos de armado y desarmado que se presentan en el ruteo de los taladros. Se implementó una metodología de optimización agrupando pozos petroleros con el vecino más cercano para una creación de clústeres iniciales, donde cada clúster es una plataforma. Posteriormente, se volvieron a evaluar los pozos para modificar su clúster en caso de que se encontraran en un clúster más cercano, como se puede observar en la figura 3.

Pseudocódigo Heurística Clústeres	
Pozos_Coordenadas	//Ubicación geográfica de los pozos
Distancia_Pozos	//Matriz de distancia de los pozos
Pozos_Clúster	//Clústeres iniciales de los pozos
Total_Pozos	//Número de pozos totales a evaluar
Distancia_C	//Distancia máxima del Clúster
For 1 to Total_Pozos	
	// Asignar un clúster al pozo si no tiene
	// Asignar el mismo clúster a los pozos cercanos que cumplan con la limitante
	If Pozos_Cluster[i] is equal to 0
	Set Clúster
	Set Pozos_Cercanos
	End if
	End For
For 1 to Total_Pozos	
	//Reasignar clústeres si un pozo se encuentra más cercano a otro
	For 1 to Total_Pozos
	If j is different to I and Distancia_Pozos[i][j] less than DistanciaC / 2
	Set Pozos_Cluster[j] = Pozos_Cluster[i]
	End if
	End For
	End For

Figura 3. Pseudocódigo y representación gráfica de los clústeres. Fuente: Autoría Propia

Una de las restricciones que se tiene en la construcción de estas plataformas, por experiencias ya planteadas de parte de nuestro director, no pueden ser más grandes que $150 \times 150 \text{ m}^2$, ya que es el tamaño usual que poseen.

4.2.3 Modelo Matemático

El modelo planteado toma en cuenta características del problema de ruteo de vehículos VRP, más específicamente con ventanas de tiempo (VRPTW), teniendo en cuenta que se tienen asignados tiempos de inicio para las actividades. Para ajustar el problema planteado a un VRP se considera cada una de las diferentes actividades como nodos, es decir, cada actividad que se tiene por pozo es representada como un nodo que posee la ubicación del respectivo pozo y está asociada a una ventana de tiempo definida entre el tiempo inicial de la actividad a realizar y el tiempo inicial de la siguiente actividad propuesta en la secuencia de cada pozo. Asimismo, no se tiene en cuenta una demanda asociada a los nodos y no se cuenta con un depósito, para suplir esta necesidad en el problema se toma un punto imaginario como representación del depósito el cual tiene distancia cero con cada uno de los nodos. Por lo tanto, se puede modelar el problema como un VRPTW conservando las restricciones inicialmente planteadas en la problemática. A continuación, se presenta el modelo matemático planteado:

Conjuntos	
I	Nodos de actividades en el campo petrolífero
K	Cantidad de equipos o taladros
Parámetros	
$ActTal_{k,i}$	Matriz binaria si el equipo k puede realizar la actividad del nodo i
$CO_{k,i}$	Costo de operación del taladro k , en el nodo i [\\$]
$CM_{k,i}$	Costo de movilización del taladro k [\$/día]
CE_k	Costo de operación del taladro k [\\$]
$CArm_k$	Costo de armado del taladro k [\$/día]
$CDesArm_k$	Costo de armado del taladro k [\$/día]
TA_i	Tiempo de actividad del nodo i [día]

$TArm_{k,i,j}$	Tiempo de armado del equipo k en el nodo i si viene del nodo j
$TDesarm_{k,i,j}$	Tiempo de desarmado del equipo k en el nodo i si viene del nodo j
$Velocidad_k$	Velocidad del equipo k en el nodo i si viene del nodo j [0,3 km/día]
$Dist_{i,j}$	Distancia entre el nodo i al nodo j [0,3 km]
$Tmin_i$	Tiempo mínimo de inicio del nodo i
$Tmax_i$	Tiempo máximo de inicio del nodo i
M	Gran M . Numero muy grande.
Variables de decisión	
$X_{i,j,k}$	Binaria: 1 Si el equipo k se mueve del pozo al pozo i al pozo j . 0 De lo contrario.
$Y_{i,k}$	Variable de decisión del día de inicio de actividad del equipo k en el nodo i .
T_i	Variable de decisión del día de inicio de actividad en el nodo i .
Función Objetivo	
Función objetivo 1: Costo total. $\min Costo\ Total := \sum \sum \sum_{i,j,k} X_{ijk} (CO_{k,j} + CArm_k * TArm_{k,i,j} + CDesArm_k * TDesArm_{k,i,j} + CM_k * (Dist_{i,j}/Velocidad_k) + CE_k)$	
Función Objetivo 2: Tardanza	
1. Tardanza total:	$\min Tardanza := \sum_i T_i - Tmin_i$
2. Tardanza promedio por actividad:	$\min Tardanza := (\sum_i T_i - Tmin_i)/NumAct$
Restricciones	
(1) $\sum_{i \in I} X_{i,j,k} \leq ActTal_{k,j}$	$\forall k \in K, j \in I$
(2) $X_{i,i,k} = 0$	$\forall k \in K, i \in I$
(3) $\sum_{k \in K} \sum_{j \in I} X_{i,j,k} = 1$	$\forall i \in I, i \neq 1$
(4) $\sum_{i \in I} X_{i,1,k} \leq 1$	$\forall k \in K$
(5) $\sum_{i \in I} X_{i,j,k} = \sum_{i \in I} X_{j,i,k}$	$\forall k \in K, j \in I, j \neq 1$
(6) $Y_{j,k} \geq Y_{i,k} + (TArm_{k,i,j} + TDesarm_{k,i,j} + \left(\frac{Dist_{i,j}}{Velocidad_k}\right) + TA_i) * X_{i,j,k} - M * (1 - X_{i,j,k})$	$\forall k \in K, i \in I, j \in I, j \neq i$
(7) $Y_{i,k} \geq Tmin_i * \sum_{j \in I} X_{j,i,k}$	$\forall k \in K, i \in I, i \neq 1$
(8) $Y_{i,k} \leq Tmax_i * \sum_{j \in I} X_{j,i,k}$	$\forall k \in K, i \in I, i \neq 1$
(9) $T_i = \sum_{k \in K} Y_{i,k}$	$\forall i \in I$
(10) $X_{j,i,k}$	$\in [0,1]$
(11) $x_{i,j,k}, Y_{i,k}, T_i \geq 0$	$\forall i \in I, \forall j \in I, \forall k \in K, \forall t \in T, \forall a \in A$

Tabla 1. Modelo matemático

La función objetivo de costo total se encuentra discriminada por 5 costos:

1. **Costo de operación:** este se refiere a la tarifa del equipo bajo operación, es decir, cuando se encuentra realizando la actividad en cuestión. La tarifa no varía entre tipo de actividad.
2. **Armado:** este se refiere al costo del armado del taladro, sin embargo, existen dos costos asociados, cuando está en la misma plataforma y cuando es de diferente plataforma.
3. **Desarmado:** este se refiere al costo del desarmado del taladro, sin embargo, existen dos costos asociados, cuando está en la misma plataforma y cuando es de diferente plataforma.
4. **Movilización:** es la tarifa de la movilización de todos los taladros en el horizonte de la operación.

5. **Espera/Stand by:** es el costo asociado a un equipo que no se encuentra realizando ninguna actividad. Esto ocurre cuando el equipo llegó más temprano a la localización o ese pozo está ocupado por otra máquina.

La función objetivo de Tardanza se refiere al desfase entre el momento inicial de la actividad y su momento inicial programado. Esta puede expresarse de dos maneras diferentes.

1. Tardanza total en días.
2. Tardanza promedio por actividad en días.

Con respecto a las restricciones: La primera restricción (1), define que los equipos solo pueden realizar sus actividades permitidas. La segunda restricción (2) define que no se pueden crear rutas entre el mismo nodo. La tercera restricción (3) precisa que se tienen que visitar todos los nodos. La cuarta restricción (4) concreta que solo puede haber una ruta por taladro. La quinta restricción (5) define que las rutas entran y salen del mismo nodo. La sexta restricción (6) define ventanas de tiempo para realizar las actividades de los nodos. La séptima restricción (7) establece el tiempo mínimo en que puede comenzar a trabajar el equipo. La octava restricción (8) define el tiempo máximo en que puede comenzar el equipo. La novena restricción (9) define el tiempo de inicio de actividad de cada nodo. La décima y onceava restricción (10-11) son restricciones que definen las variables y su rango de valores.

Para finalizar, el problema se puede modelar con programación lineal limitando la cantidad de nodos y enfocandose en una sola función objetivo. Por tal razón, se va a realizar una simulación con este método para una pequeña instancia. Para resolver la totalidad del problema se van a utilizar las técnicas de heurísticas y metaheurísticas, además, para modelar la variable aleatoria planteada y tener en cuenta la reprogramación, se utilizará un algoritmo dinámico, el cual tiene como objetivo simular el comportamiento de un sistema en cada estado de tiempo, teniendo en cuenta los cambios que se generen y realizando los ajustes necesarios.

4.3. Fase 3. Construcción del modelo determinístico

4.3.1. Algoritmo Voraz

Teniendo en cuenta el modelo planteado, y la magnitud de los datos proporcionados, se diseñó un algoritmo voraz (Greedy algorithm), el cual tiene como objetivo seleccionar la ruta óptima en cada iteración del ruteo, hasta llegar a una solución factible que cumpla las restricciones planteadas en la problemática. El algoritmo que se propuso se puede observar en la figura 4.

Pseudocódigo Algoritmo Voraz	
Pozos_Terminado	//Variable binaria que se pone en verdadero cuando todos los pozos ya realizaron sus actividades
Total_Taladros	//Número total de taladros programados
Pozo_Escogido	//Pozo que se le asignará al taladro en programación
While Pozos_terminados is equal to False	
	//Función corre hasta que todas las actividades de los pozos se encuentren terminadas
	//Escoge el mejor pozo al que puede ir el taladro
	//Asigna el pozo escogido a la ruta del taladro
	For 1 to Total_Taladros
	Set Pozo_Escogido = Escoger_Pozo[i] // Función que encuentra el mejor pozo al que puede ir el taladro
	If Pozo_Escogido is different to 0
	Set Asignar_Pozo[i][Pozo_Escogido] // Asigna el pozo escogido al taladro que se está programando actualmente
	End if
	End For
	End While

Figura 4. Pseudocódigo Algoritmo Voraz. Fuente: Autoría Propia

Esta metodología diseñada, encuentra iterativamente el ruteo parcial de cada taladro al ubicar el mejor nodo al que puede ir. Comienza escogiendo un taladro y evaluando desde su posición actual cuál es el nodo que disminuye el costo de operación, costo de armado y desarmado, costo de movilización y costo de espera, de manera que su siguiente posición disminuya ese costo total. Posteriormente, se escoge el siguiente taladro y se

prosigue a realizar los mismos pasos hasta que complete todo el ruteo. El ruteo final tiene un total de 3926 actividades y parte de una secuencia inicial guardada en un vector, de manera que se intervenga un pozo varias veces con el mismo o diferentes taladros que sean capaces de realizar dicha actividad.

4.3.2. Metaheurística

Con el fin de mejorar la solución inicial implementada con el algoritmo voraz, es necesario proceder con una metaheurística, para esto se debe establecer los criterios que se desean optimizar, en este caso serían el ruteo y/o la recombinación de taladros. Para el caso del ruteo, según los antecedentes las metaheurísticas PSO, TABU y Genético son las que mejor se desempeñan, sin embargo, al ser un problema de gran esfuerzo computacional intervenir en el ruteo tardaría demasiado en encontrar una solución más favorable. Por otra parte, está la recombinación del número de taladros por tipo, para este caso no existe una evidencia empírica de cuál es el mejor método por utilizar, a pesar de esto, la metaheurística PSO es recomendable porque su tiempo de búsqueda es ágil y explora un gran número de soluciones. Por tales razones se establece usar como criterio la recombinación de número de taladros por tipo utilizando el PSO, con el fin de encontrar una mejor solución al problema planteado. En la figura 5 se presenta el pseudocódigo del PSO.

Pseudocódigo PSO	
For partícula	Inicializar partícula //Inicia con un aleatorio para la cantidad de taladros por tipo
End For	
Do	
For Partícula	Calcula nuevo valor por partícula //Hace la nueva combinación de cantidad de taladros por partícula
	If nuevo valor de partícula es mejor que su mejor valor.*
	Mejor valor de partícula es igual al valor actual.
	End If
End For	
	Calcula el mejor valor de todas mejores posiciones de las partículas. * //Guarda la mejor posición encontrada
For partícula	Realiza ecuación de velocidad y actualiza la nueva posición de la partícula
End For	
While maximun iterations or maximun time.	
*Mejor Valor: el que tenga menor tardanza dado un rango de costos.	

Figura 5. Pseudocódigo PSO. Fuente: Autoría Propia

Para realizar la decodificación del problema, es decir pasar de un problema continuo (movimiento de partículas) a un problema discreto (permutaciones de enteros / variación de los taladros) se aborda de la siguiente manera. La posición de una partícula está formada por una cantidad posible de valores enteros, más precisamente una cantidad específica de número de taladros por tipo. La velocidad está representada por una magnitud entera, para este caso significa la cantidad de taladros que se suman o restan para la siguiente posición de la partícula. Cada tipo de taladro tiene su propia velocidad y son independientes entre ellas. A continuación, se presenta un ejemplo de la decodificación.

	Pos. Inicial	Velocidad	Pos. Final
Rig / equipo 1	10	+1	11
Rig / equipo 2	10	+2	12
Rig / equipo 3	10	-5	5
Rig / equipo 4	10	-6	4

Figura 6. Ejemplo de codificación PSO. Fuente: Autoría Propia

Por último, esta utiliza los siguientes parámetros: variables aleatorias iniciales en un rango de mínimo 2 y máximo 30, esto se debe a que son los límites que se pueden tener de las cantidades de taladros. Se define 30 partículas con un peso $C1=2$, $C2=2$ y con un $w=0.7$, una velocidad máxima de 8 unidades de taladros y como criterio de parada un máximo de 3000 iteraciones o un tiempo máximo de seis horas. Para finalizar se crea una función rebote en caso de que la siguiente posición de cada partícula supere los límites establecidos.

4.4. Fase 4. Frontera de soluciones

Dado que no existe a priori un peso correspondiente a cada función objetivo se establece construir una frontera de soluciones. Esta metodología se utiliza para que el tomador de decisión pueda evaluar el costo versus la tardanza. Para construir esta frontera se deben realizar múltiples soluciones con cantidades diferentes de taladros, en este caso se utiliza un enjambre de partículas (PSO) para encontrar unos puntos de la frontera, con el fin de sensibilizar sobre ella y así obtenerla. La metodología NISE (Non Inferior Set Estimation) encuentra una frontera más exacta donde minimiza el error en la misma, sin embargo, por los alcances del proyecto, se establece dejar este método como futuros trabajos.

4.5. Fase 5. Definición de la variable estocástica

Dada la definición del problema, se evidencia que la información de los pozos se “actualiza” constantemente, es decir, los yacimientos obtienen una nueva información histórica del desempeño de cada pozo y con base a ello priorizan y asignan actividades por pozo nuevamente. La probabilidad de que un trabajo se “actualice”, depende del tipo de actividad y lo lejos que esté del horizonte de tiempo en el momento que se está haciendo la programación. Dicha probabilidad de cambio esta moldeada por la siguiente curva de superficies que se observa en la figura 7.

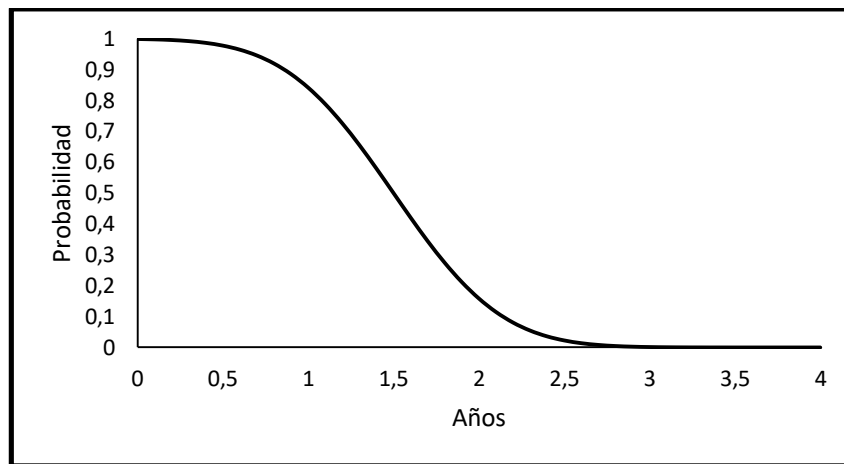


Figura 7. Función de probabilidad del cambio de actividades. Fuente: Autoría Propia

Esta simulación se construyó a partir de una función de supervivencia que está dada por $[1-F(t)]$, donde $F(t)$ es la función acumulada de la distribución de probabilidad normal con media 1,5 y desviación estándar de 0,5. El eje vertical indica el comportamiento a lo largo de los años desde el momento en que se hace una nueva reprogramación, y el eje vertical indica la probabilidad de que la actividad tenga o no un cambio. Como conclusión, toda actividad que esté programada para hacerse después de dos años y medio tiene una probabilidad cercana al cien por ciento de ser cambiada por otra actividad.

4.6. Fase 6. Construcción del modelo dinámico.

Como se mencionó anteriormente, la programación dinámica estocástica tiene como objetivo simular el comportamiento de un sistema en cada estado de tiempo, de tal manera que se pueda conocer el comportamiento de la operación mientras transcurren los años y realizar los ajustes necesarios.

El diseño de un modelo dinámico, que tenga en cuenta la variable estocástica anteriormente planteada, busca llevar la problemática a un contexto más real, donde la planeación inicial, al desarrollarse a lo largo de varios años, pueda cambiar y tener una perspectiva general del impacto que podrían tener estos cambios a lo largo del proyecto. Por consiguiente, se plantea un modelo de simulación anual donde cada año se actualiza la planeación, teniendo en cuenta los pozos y actividades ya realizadas. Posteriormente, las actividades que no han sido

realizadas se modifican dependiendo del comportamiento estocástico de estas, como se puede observar en la figura 8 y en la figura 8.1.

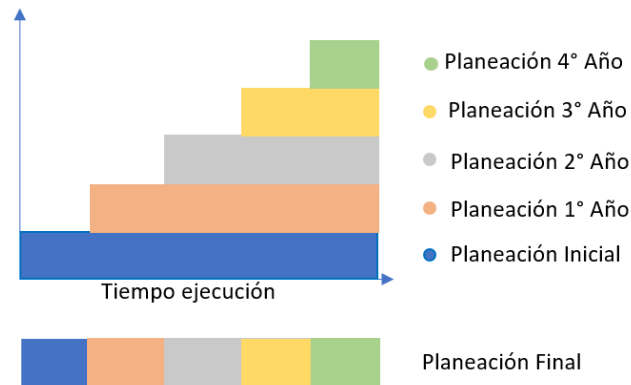


Figura 8. Ejemplo de la planeación dinámica. Fuente: Autoría Propia

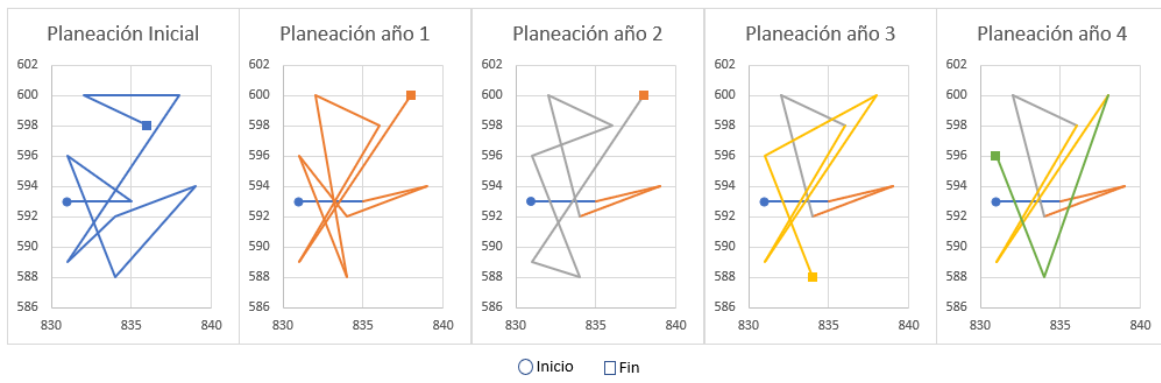


Figura 8.1. Ejemplo de la ruta de la planeación dinámica. Fuente: Autoría Propia

Estos cambios se realizan construyendo una solución inicial de planeación con las 3926 actividades de los pozos, y las posiciones iniciales de los taladros. Posteriormente, en cada año de planeación se vuelve a correr una simulación, teniendo en cuenta en cual pozo quedaron y cuales secuencias ya se habían realizado.

Pseudocódigo Programación Dinámica	
Total_Años	//Número total de años que se reprograman
Set Algoritmo_Voraz()	//Se llama la función algoritmo voraz anteriormente descrita
For 1 to Total_Años	
	//Se encuentra la posición inicial donde se encuentra el taladro al n año de programación
	//Se actualizan las actividades que no se han programado, de acuerdo con el comportamiento de su variable estocástica
	//Se realiza el ruteo nuevamente con el algoritmo voraz
	Set Posición_Inicial(Año)
	Set ActualizarActividades(Año)
	Set Algoritmo_Voraz()
End For	

Figura 9. Pseudocódigo programación dinámica. Fuente: Autoría Propia

4.7. Fase 7. Indicador del impacto económico de la solución

El impacto de la metodología propuesta se llevará a cabo a través de indicadores que midan la diferencia económica de la solución inicial respecto a la realmente desarrollada en la planeación, como se muestra en la ecuación 2 y ecuación 3. El primer indicador se encuentra relacionado al porcentaje de varianza entre el costo

inicial planeado respecto al costo desempeñado en la última planeación, de manera que se pueda medir la eficiencia de la reprogramación. El segundo indicador se refiere a la cantidad de actividades que realmente se realizaron, debido a que los pozos al reprogramarse pueden ser cerrados antes de la fecha inicial propuesta.

$$\% \text{ Costo final de la operación} = \frac{\text{Costo última planeación}}{\text{Costo inicial de planeación}} * 100\%$$

Ecuación 2. Indicador del costo final de la operación

$$\% \text{ Actividades realizadas} = \frac{\text{Cantidad de actividades realizadas en la última planeación}}{\text{Cantidad de actividades realizadas en la planeación inicial}} * 100\%$$

Ecuación 3. Indicador del costo final de la operación

5. Resultados

A continuación, se muestran los resultados obtenidos en cada fase.

5.1. Fase 2: Diseño del modelo matemático del problema.

5.1.1. Los puntos de inicio

Después de realizar el algoritmo de Weiszfeld para los cuatro tipos de taladros, se dio como resultado el punto óptimo de inicio para cada tipo, obteniendo los siguientes pozos de inicio (tabla 2). Como se observa más adelante en la sección 5.2.1 el costo de movilización representa en promedio el 0,2% del costo total, de manera que el cálculo de los puntos iniciales no afecta de manera significativa al problema, por esta razón, no se busca dentro de este trabajo la optimización de estos puntos.

	Pozo de Inicio
Rig / equipo 1	627
Rig / equipo 2	627
Rig / equipo 3	1
Rig / equipo 4	83

Tabla 2. Pozos iniciales de equipos. Fuente: Autoría Propia

5.1.2. Plataformas (Clústeres)

La metodología planteada para crear los clústeres se corrió sobre los 707 pozos propuestos, teniendo en cuenta las coordenadas cartesianas de cada uno y el tamaño máximo que puede tener una plataforma, como se puede observar en la figura 10, donde se representa por cada color el clúster al que pertenece el pozo. El algoritmo dio como resultado la creación de 134 clúster que reúnen los 707 pozos propuestos, como se puede observar en la figura 11. Además, el promedio de tamaño del clúster que se crearon es de $50 \times 55 \text{ m}^2$.

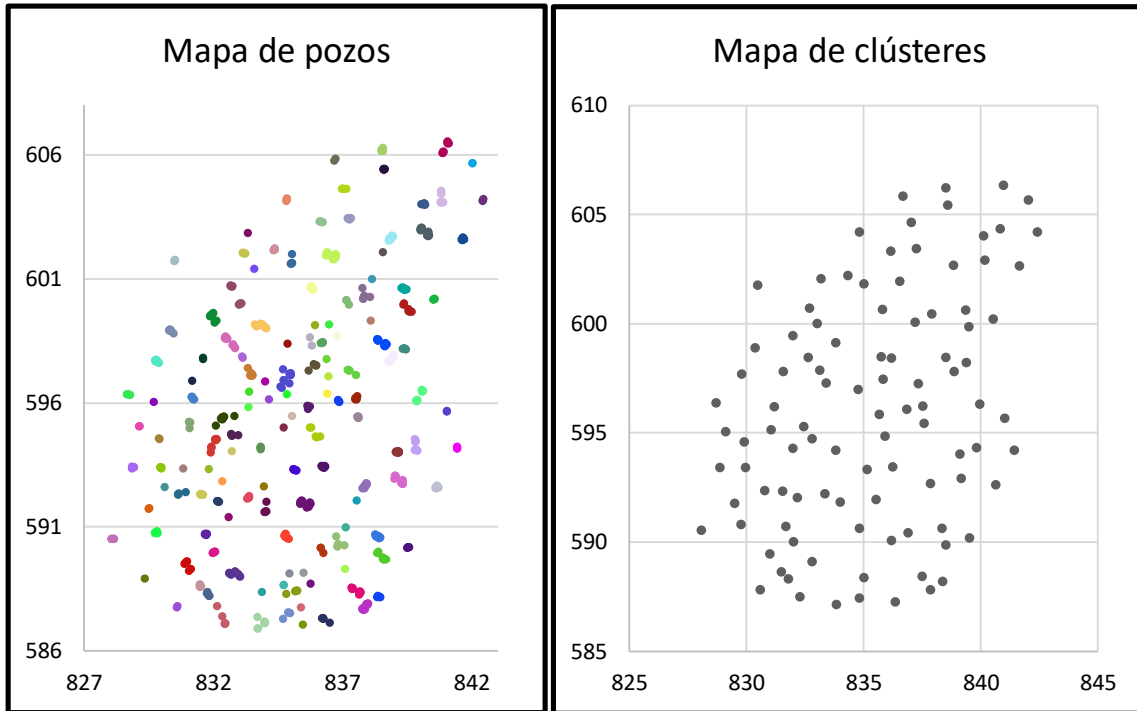


Figura 10. Pozos divididos por clúster y Figura 11. Mapa del clústeres. Fuente: Construcción Propia

5.1.3. Modelo matemático

El modelo matemático planteado fue programado y simulado con el programa *GUSEK* en una instancia pequeña con la finalidad de tener un acercamiento inicial a la solución del problema. Para poder simular el modelo, fue necesario disminuir el número de actividades totales y el máximo de taladros disponibles. Por otro lado, con el fin de linealizar el problema nos enfocamos en una sola función objetivo y la otra función la modelamos como una restricción, de manera que minimizará una e intentara buscar mínimos valores en la otra. La función objetivo que se minimizó fue la tardanza. En total se realizó la simulación con diecinueve actividades representados en cuatro pozos dando como respuesta el siguiente ruteo:

1	2	3	4	5	6	7	8
Pozo 1 - Act 3	Pozo 1 - Act 9	Pozo 1 - Act 6	Pozo 1 - Act 2	Pozo 1 - Act 5	Pozo 2 - Act 1	Pozo 2 - Act 3	Pozo 2 - Act 1
Pozo 1 - Act 6			Pozo 2 - Act 7				
Pozo 2 - Act 6							

9	10	11	12	13	14	15	16
Pozo 2 - Act 3	Pozo 3 - Act 1	Pozo 3 - Act 1	Pozo 3 - Act 2	Pozo 3 - Act 8	Pozo 3 - Act 6	Pozo 3 - Act 10	Pozo 4 - Act 6

Tardanza	0
Costo total	\$ 9.250.303

Pozo # 1	Pozo # 2	Pozo # 3	Pozo # 4
Rig/ Tipo 1	Rig/ Tipo 4		

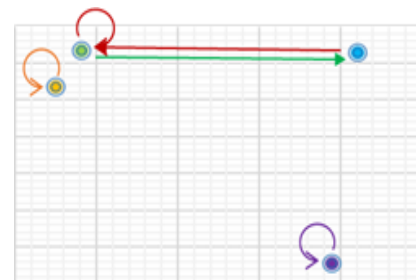


Tabla 3. Resultados Modelo matemático y Figura 12. Ruteo del modelo matemático. Fuente: Construcción Propia

Se obtuvo un total de 16 equipos para la realización de las 19 actividades, con fin de realizar el ruteo minimizando la tardanza y restringiendo la función de costos, dando como resultado que la mayoría de equipos realicen una única actividad por lo cual son muy pocos los taladros realizan ruteo entre pozos tal como se muestra en la figura 12. Después de tener una primera aproximación de la respuesta, se estrecha más el límite de la restricción de costos, de modo que se encuentre una función objetivo de costos mejor, pero dada la complejidad del problema se queda buscando un óptimo, de modo que se establece un criterio de parada de 12 horas.

Como conclusión, el problema se modeló con una instancia pequeña menor al uno por ciento del total de los datos, a causa de que el programa pueda encontrar una solución dentro de un límite de tiempo razonable, por tal razón no es representativo con la complejidad del problema. Por último, para realizar el modelamiento con el conjunto completo de datos se desarrollará con heurísticas, metaheurísticas y una programación dinámica para la variable estocástica mencionada anteriormente.

5.2. Fase 3: Construcción del modelo determinístico.

5.2.1. Algoritmo Voraz

El algoritmo voraz planteado en la metodología tiene como principal parámetro de entrada el número de taladros por tipo que se utilizan en la operación, teniendo en cuenta su posición inicial y las actividades que puede realizar cada uno. En la tabla 4 se presentan los resultados obtenidos con algoritmo voraz al variar la cantidad de unidades de cada tipo de taladro.

Unidades Tipo 1	Unidades Tipo 2	Unidades Tipo 3	Unidades Tipo 4	Tiempo de ejecución [seg]	Fo Tardanza [Días]		Fo Costo Total [\$]					
					Total	Promedio	Total	Operando [\$]	Armado [\$]	Desarmado [\$]	Movilización [\$]	Espera /Stdby [\$]
5	6	8	10	64,5	1.183.146	301,28	\$2.148.010.480	\$850.339.200	\$353.727.270	\$203.090.745	\$4.066.347	\$736.786.918
12	16	8	5	64,8	1.543	0,39	\$2.881.611.036	\$904.256.400	\$406.654.740	\$240.543.798	\$5.479.622	\$1.324.676.476
12	8	7	12	64	59.045	15,04	\$2.559.971.536	\$862.918.800	\$396.361.980	\$218.296.452	\$5.168.206	\$1.077.226.098
8	4	16	12	60	1.982.514	504,84	\$2.655.782.885	\$868.483.200	\$399.663.270	\$214.226.067	\$4.589.283	\$1.168.821.065
8	14	13	6	61,2	79.020	20,12	\$2.626.521.902	\$865.671.600	\$369.095.040	\$222.345.156	\$5.006.502	\$1.164.403.604

Tabla 4. Respuestas Algoritmo Voraz. Fuente: Autoría Propia

Como se puede apreciar en los resultados, el tiempo de ejecución del programa se encuentra cerca de los 60 segundos. La cantidad de taladros afecta significativamente los costos y la tardanza en días que tiene la problemática. Además, se puede observar que la solución con menor tiempo de tardanza y mayor costo es aquella que presenta mayor cantidad de taladros tipo 1 y tipo 2, ya que estos pueden realizar más actividades, pero sus costos de operación son mayores a los de los demás taladros. En la sección 5.5 se presenta un análisis de la frontera de soluciones generada, de manera que se puedan analizar las combinaciones de soluciones que se presentan.

5.2.2. Metaheurística

Para mejorar la solución inicial se establece implementar la técnica metaheurística PSO recombinando la cantidad de taladros por tipo. Para implementarlo, es necesario trazar nuevamente el modelo, pues la metaheurística PSO optimiza una sola función objetivo y el problema planteado tiene dos funciones objetivo. Para esto, es obligatorio pasar una función objetivo a una restricción, en este caso se estableció pasar el costo total en una restricción, por tanto, se establece un límite del más o menos siete por ciento de la solución inicial. De este modo, la función objetivo es minimizar esa tardanza teniendo en cuenta un rango de costos que no se pueden superar. Por otro lado, cada iteración tarda alrededor de un minuto, y poner un máximo de iteraciones puede requerir de mucho esfuerzo computacional, lo cual tarda demasiado. Por ello, se decide como criterio de parada un lapso de seis horas. A continuación, se presenta el punto inicial y el punto resultante en la tabla 5.

	# Tipo 1	# Tipo 2	# Tipo 3	# Tipo 4	Fo Tardanza [Días]		Fo Costo Total [\$]
					Total	Promedio	
Voraz	8	4	16	12	1.982.514	504,84	\$2.655.782.885
PSO	5	21	6	8	1.315	0,33	\$2.668.373.854
Porcentaje de variación					-99,93%		0,5%

Tabla 5. Respuesta algoritmo PSO. Fuente: Autoría Propia

Como se puede observar en la figura 12, la metaheurística PSO disminuyó en un 99,93% la tardanza con respecto al punto del algoritmo voraz. Dados los resultados, existe una clara evidencia de que recombinar el número de taladros tiene un gran impacto en las funciones objetivo, por tal motivo se selecciona esta técnica metaheurística con las mismas parametrizaciones para encontrar los puntos de la frontera de soluciones

5.3. Fase 4: Frontera de soluciones

Para hallar la frontera exacta, es necesario ejecutar una solución por cada costo dado, sin embargo, este es un método poco eficiente, ya que requiere de mucho tiempo para ejecutar todas las soluciones. Se realizó un PSO en diferentes intervalos de costos, los cuales generan unos puntos determinados de la frontera, posteriormente se ejecutó una sensibilidad sobre ellos obteniendo así la frontera.

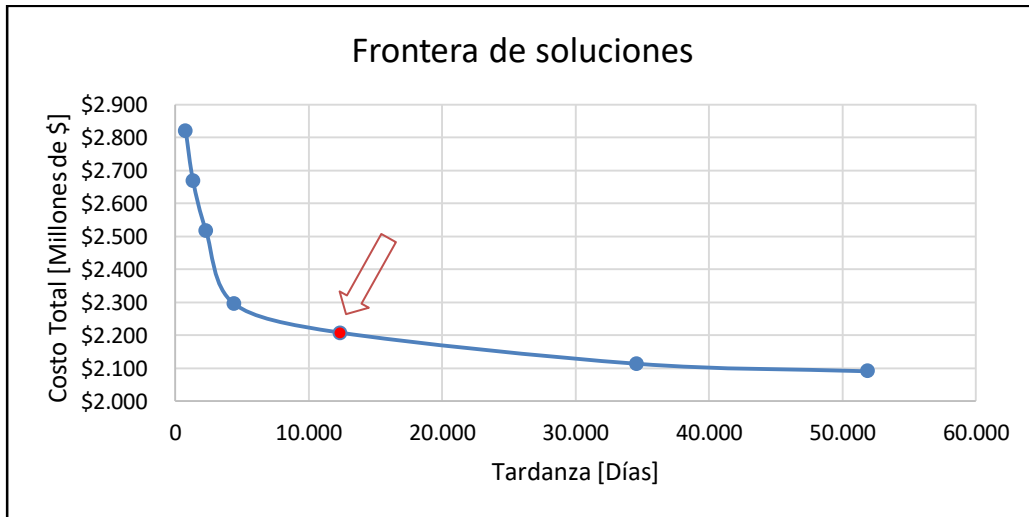


Figura 13. Frontera de soluciones. Fuente: Autoría Propia

Como se observa en la figura 13, el comportamiento de la frontera está dado por una función potencial negativa, esto quiere decir que tiene una curva hiperbólica, la cual tiene una asíntota vertical cuando la tardanza tiende a cero y una asíntota horizontal cuando el costo total tiende a cero. De igual forma, nunca corta a los ejes de coordenadas. Cada punto de la frontera corresponde a una combinación de taladros, en la siguiente tabla se presenta dicha combinación con sus respectivas funciones objetivo.

Frontera de soluciones - PSO				FO Tardanza [Días]		FO Costo Total [\$]
# Taladros 1	# Taladros 2	# Taladros 3	# Taladros 4	Total	Promedio x Act	
5	11	7	6	51.860	13,2	\$2.091.333.086
5	12	6	5	34.542	8,8	\$2.114.178.361
5	15	6	5	12.331	3,1	\$2.208.411.188
4	16	6	7	4.410	1,1	\$2.295.153.450
6	17	7	7	2.287	0,6	\$2.516.966.690
5	21	6	8	1.315	0,3	\$2.668.373.854
3	24	10	10	765	0,2	\$2.819.959.251

Tabla 6. Frontera de soluciones con funciones objetivo.

Como se observa en la tabla 6, el taladro tipo 2 es el que predomina en su cantidad, esto se debe a que este tipo de taladro puede realizar el 90% de las actividades y su costo es menor comparado con el taladro tipo 1. Para los taladros tipo 3 y tipo 4 sus costos asociados son menores que la del tipo 1 y 2, sin embargo, estos solo pueden realizar el 20% de las actividades. Para la construcción de la programación dinámica se toma un punto de la frontera descrita, este punto puede ser cualquiera dado que el tomador de decisiones selecciona dicho punto según sus intereses, para efectos de este trabajo de grado y poder realizar el modelamiento dinámico se selecciona cualquier punto, en la figura 13 se presenta el punto seleccionado, el cual se configura con 5, 15, 6 y 5 taladros del tipo 1, 2, 3 y 4 respectivamente.

5.4. Fase 6: Construcción del modelo dinámico

La programación dinámica planteada tiene como objetivo reprogramar una solución inicial mientras avanza el tiempo de ejecución, teniendo en cuenta el comportamiento de las actividades y cómo afecta su desarrollo a la planeación. El punto que se escogió para esta simulación se encuentra resaltado en la tabla 6. Esta programación, como se observa en la figura 14, afecta el costo total y la tardanza y, además, disminuye un 14% la cantidad de actividades realizadas. En la figura 14 se representa cómo se genera el ruteo de un taladro en específico.

Rig / Equipo tipo	Unidades disponibles	Planeación Inicial			Planeación Final		
		Fo Tardanza [Días]	Fo Costo [\$]	Actividades Realizadas	Fo Tardanzas [Días]	Fo Costo [\$]	Actividades Realizadas
1	5	10544	\$ 2.284.039.234	3926	5595	\$ 2.170.432.385	3416
2	15						
3	6						
4	5						

Tabla 7. Resultados respuesta dinámica. Fuente: Autoría Propia

Año 8	685	407	155	640	587	650	482	4	594	661	58	704	653	156	32	552	586	52	645	165	212	450	11	653	494	95
Año 7	685	407	155	640	587	650	482	4	594	661	58	704	653	156	32	552	586	52	645	165	212	450	11	653	494	95
Año 6	685	407	155	640	587	650	482	4	594	661	58	704	653	156	32	552	586	52	645	165	212	450	11	335	212	682
Año 5	685	407	155	640	587	650	482	4	594	661	58	704	653	156	32	552	586	52	645	165	212	89	62	619	405	605
Año 4	685	407	155	640	587	650	482	4	594	661	58	704	653	156	32	552	586	52	645	546	549	156	249	351	435	315
Año 3	685	407	155	640	587	650	482	4	594	661	58	704	653	156	104	182	218	485	60	433	312	114	253	459	162	552
Año 2	685	407	155	640	587	650	482	4	594	661	58	232	269	223	396	273	182	550	699	14	563	414	168	619	405	670
Año 1	685	407	155	640	587	650	605	482	154	306	490	369	448	577	459	574	694	201	251	495	440	12	376	89	690	188
Determinístico	685	407	155	382	697	366	35	529	653	631	562	35	203	293	266	345	629	83	361	542	14	212	143	481	54	582

Figura 14. Ruta de un taladro - dinámico. Fuente: Autoría Propia

5.5. Fase 7: Indicador del impacto económico de la solución

5.5.1. Análisis de resultados

Teniendo en cuenta la solución planteada en el modelo lineal y la heurística, se realizaron las comparaciones entre ellas con las mismas instancias como se puede observar en la tabla 8, donde se tuvo un gap de costos del 20% entre las dos soluciones y ninguna de las dos presentó una tardanza asociada. Por otro lado, el tiempo de ejecución en la programación lineal es de sesenta segundos, sin embargo, al buscar mejores soluciones teniendo en cuenta la función objetivo de costos, el tiempo de simulación aumenta considerablemente, por lo que se empleó un criterio máximo de parada de doce horas. Para finalizar se resalta que la programación lineal es bastante buena para problemas o instancias pequeñas, sin embargo, al ser este un problema de alta complejidad y multiobjetivo no es factible realizarlo con este método usando la totalidad de los datos.

	Heurística	Modelo matemático
Costo Total	\$ 11.609.335	\$ 9.250.303
operando [\$/hora]	\$ 6.405.600	\$ 6.405.600
Armado [\$/hora]	\$ 3.488.400	\$ 2.708.100
Desarmado [\$/hora]	\$ 1.705.440	\$ 132.396
Movilización [\$/hora]	\$ 9.895	\$ 4.207
Tardanza	0	0
Tiempo ejecución	< 3(seg)	60 (seg)

Tabla 8. Comparación de PL vs Heurística. Fuente: Autoría Propia

Por otro lado, una vez obtenido el resultado tanto determinístico como dinámico, se obtuvo que la programación dinámica disminuyó un 4,97% del costo total con respecto a la programación determinística, equivalente a \$113'606,848 de dólares. Este resultado, como se puede observar en la figura 15, indica que seguir la planeación inicial sin tener en cuenta la variación del proceso genera un sobre costo en su ejecución y una mayor tardanza. Adicionalmente, se aprecia que las planeaciones dinámicas oscilan en un rango de costos, puesto que a medida que se cumplen las planeaciones la probabilidad de que haya un cambio en las actividades futuras disminuye.

Ahora bien, la cantidad de tiempo requerido para realizar la totalidad de actividades durante la planeación inicial es de 9,25 años. Una vez realizada la planeación estocástica, el tiempo obtenido es de 9,19 años, es decir, una reducción del 0,62%, además, se obtuvo una disminución promedio del 14% del número total de actividades realizadas. En consecuencia, se puede evidenciar que una disminución significativa de actividades no afecta el tiempo total de ejecución, esto se debe principalmente a que el tiempo de inicio de actividades no varió en cada reprogramación. Por otra parte, la disminución del total de actividades contribuyó a la reducción de la función objetivo de la tardanza, a causa de que al haber menor cantidad de actividades se cuenta con mayor disponibilidad de los taladros y se reduce el desfase entre el tiempo inicial planeado y el momento en que se realizó la actividad.

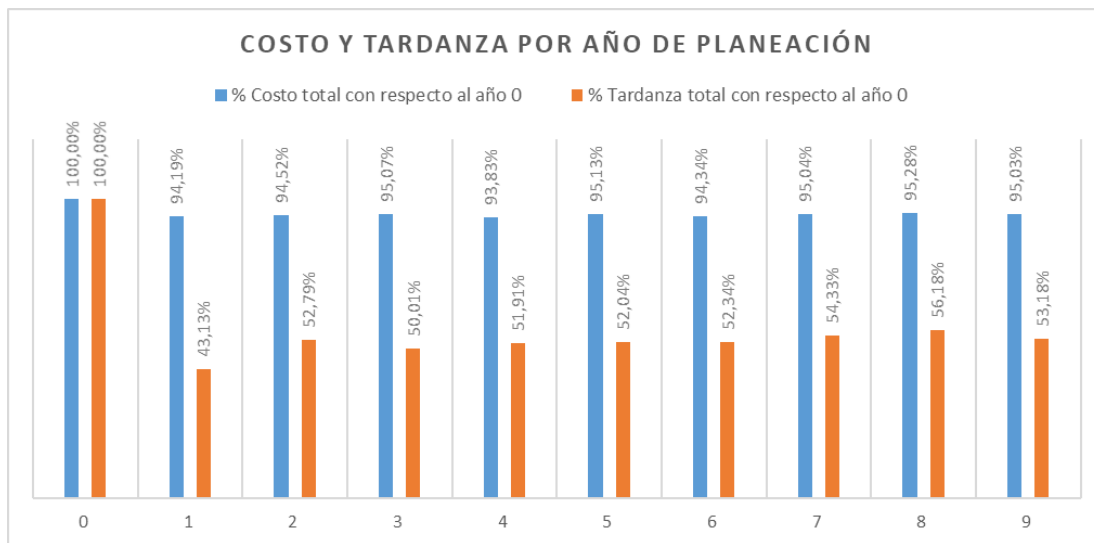


Figura 15. Costo y tardanza por año de planeación. Fuente: Autoría Propia

6. Limitaciones, conclusiones y recomendaciones.

El presente trabajo de grado diseñó la programación de rutas para taladros en campos petroleros. Su metodología consistió en la recolección de datos del proceso realizado en los campos y el desarrollo de las limitantes que afectan el problema planteado, para establecer una solución determinística que se alinee con un enfoque multiobjetivo y finalizando en la construcción de una respuesta estocástica. De este modo, se puede ofrecer un rango de soluciones donde la persona encargada de la decisión, al momento de invertir recursos en la operación, pueda definir de forma concisa cómo distribuir sus recursos y un resultado estimado del costo de la decisión que ha tomado.

Ahora bien, se realizó un modelo en programación lineal para una pequeña instancia, sin embargo, dada la complejidad y la no linealidad del problema no es posible resolver la totalidad del problema con este método. Por otro lado, el algoritmo voraz resolvió la totalidad del problema minimizando la función objetivo de costos, la cual sirvió como base de desarrollo para la optimización de la función multiobjetivo. Posteriormente, con la implementación de la metaheurística PSO, se obtuvo la cantidad por tipos de taladro que ofrecen la mejor combinatoria para la reducción de la función de tardanza, con esto se muestra que el método metaheurístico es efectivo para realizar la búsqueda de los puntos dentro de la frontera de soluciones garantizando un acercamiento al resultado óptimo.

Dado que los yacimientos obtienen nueva información histórica del desempeño de cada pozo y con base a ello priorizan y asignan nuevas actividades por pozo, se logra hacer una programación dinámica que realice estos cambios futuros de las actividades. Esta programación se hace anualmente, es decir, cada año que pase en la ejecución del ruteo se reprograman las actividades futuras conllevando a un nuevo ruteo. Como resultado, la programación dinámica obtuvo una reducción en costos del 4,97% con respecto al ruteo determinístico, esto radica en una disminución del 14% de las actividades finalmente ejecutadas contra las inicialmente planeadas. Para concluir, en problemas de temporalidades como este, siempre existe una incertidumbre sobre lo que pueda

pasar en un futuro, para este caso la incertidumbre reside en el desempeño del pozo. Por tal razón, realizar un cambio dinámico tiene como finalidad efectuar un pronóstico un poco más controlado sobre dicha incertidumbre y poder predecir mejor el comportamiento de un pozo.

Para finalizar, cabe resaltar que el proyecto tiene alcances limitados, lo cual impide su implementación, puesto que es recomendable tener en cuenta variables para una posible ejecución, por ejemplo, variables de tiempo estocásticas, usar una matriz de distancias entre nodos más exactas, ingresos generados a lo largo del tiempo, costos de mano de obra, entre otros. *Asimismo, se recomienda analizar el efecto de la tardanza de manera más detallada dentro del problema y, si es posible, buscando la manifestación de este efecto dentro del análisis de costos.* Para efectos de este proyecto de grado se abordó una parte esencial en la programación de los taladros, con lo cual se hace posible tener una proximidad al comportamiento de una variable dinámica a través del tiempo y saber cómo se puede intervenir en ella. Como futuro trabajo se podría realizar una programación estocástica con la variable de los tiempos de perforación, adicionalmente se puede implementar la metodología NISE (Non Inferior Set Estimation) para valorar mejor la frontera de soluciones.

7. Anexos

Todos los anexos aquí presentados se encuentran en el siguiente enlace de OneDrive: https://livejaverianaedu-my.sharepoint.com/:f/g/personal/da-gonzalez_javeriana_edu_co/EnPJst_uoUhAsxO39dlSACMBMqWF5GgicpBNYm4KngtrJw?e=fmrFDt

Anexo 1. Aplicativo de planeación – Excel y VBA

Acá se muestra el ruteo final tanto determinístico como dinámico, en él pueden variar la cantidad de taladros y ejecutar el programa. Así mismo, se encuentran los datos iniciales, la construcción de la variable estocástica, la frontera de soluciones y las plataformas. Este anexo tiene las siguientes hojas de Excel:

- Hoja 1 – Introducción.
- Hoja 2 - Datos iniciales.
- Hoja 3 - Clúster, Plataformas.
- Hoja 4 - Respuesta determinística.
- Hoja 5 - Frontera de soluciones.
- Hoja 6 - Construcción de variable estocástica.
- Hoja 7 - Respuesta Dinámica.
- Hoja 8 - Secuencia Dinámica.

Adicionalmente, se encuentran cuatro módulos en Visual Basic donde se realizaron los códigos necesarios para el desarrollo del trabajo de grado.

Anexo 2. Validador de planeación – Excel y VBA

Para confirmar la factibilidad del ruteo se creó un validador que verifique las restricciones establecidas. Cuando el ruteo se presenta de forma atemporal es necesario realizar el cálculo del tiempo de inicio de la actividad en cada taladro, el cual es recopilado y ordenado de menor a mayor junto con el pozo y la actividad, lo cual, genera la secuencia en la que se realizaron las actividades para cada pozo. Posteriormente, se comparan los resultados con la secuencia establecida inicialmente, y que el inicio de cada actividad sea mayor o igual al tiempo base establecido. Este anexo tiene las siguientes hojas de Excel:

- Hoja 1 - Introducción.
- Hoja 2 - Datos.
- Hoja 3 - Resultado.
- Hoja 4 - Distancias.

Adicionalmente, se encuentra un módulo en Visual Basic donde se realizaron los códigos necesarios para la validación del resultado.

Anexo 3. Resultado algoritmo Weiszfeld – Excel

Aquí se muestra la construcción de los puntos iniciales de cada tipo de taladro. Este anexo tiene las siguientes hojas de Excel:

- Hoja 1 - Introducción.
- Hoja 2 - Equipo1.
- Hoja 3 - Equipo2.
- Hoja 4 - Equipo3.
- Hoja 5 - Equipo4.

Anexo 4. Modelo matemático GUSEK–CSV's

Acá se presenta el modelo matemático implementado con programación lineal bajo el entorno de *GUSEK*, así mismo, se presentan las tablas usadas para el desarrollo de este.

- *GUSEK* - Modelo matemático
- *CSV's* – Tablas necesarias para ejecutar el modelo.

Referencias

- Blum, C., Puchinger, J., Raidl, Günther r., & Roli, A. (2012). Hybrid metaheuristics in combinatorial optimization: A survey. *ELSEVIER, 7505 LNCS*(6).
- Caenn, R., Gray, G. R., & Darley, H. C. H. (2011). *Composition and Properties of Drilling and Completion Fluids* (Vol. 6th ed). Waltham, MA: Gulf Professional Publishing.
- Cañavate Bernal, R. J., Cobacho Tornel, M. B., & Rodríguez Gómez, J. M. (2002). El algoritmo de Weiszfeld para la resolución del problema económico de Weber.
- Chaer, R. (2015). Operación óptima de sistemas dinámicos con Aversión al Riesgo. *SimSEE*.
- Chen, H., Tan, G., Qian, G., & Chen, R. (2018). Ant Colony Optimization with Tabu Table to Solve TSP Problem. *Chinese Control Conference, CCC, 2018-July, 2523–2527*.
- Damghanijazi, E., & Mazidi, A. (2017). Meta-Heuristic Approaches for Solving Travelling Salesman Problem. *International Journal of Advanced Research in Computer Science and Software Engineering, 8*(5), 18–23.
- Dewil, R., Küçükoğlu, İ., Luteyn, C., & Cattrysse, D. (2019). A Critical Review of Multi-hole Drilling Path Optimization. *Archives of Computational Methods in Engineering, 26*(2), 449–459.
- Dinero. (2018). ¿Qué nos ha dejado el petróleo luego de 100 años de desarrollo?
- Freudenrich, C., & Strickland, J. (2001). How Oil Drilling Works. Retrieved October 2, 2019.
- Hernández Martínez, H. D., & Lemus Polanía, D. F. (2015). Modelo en series de tiempo para la tasa de penetración de un pozo de petróleo de referencia: Caso Puerto Boyacá - Colombia. *Ingeniería y Ciencia, 11*(22), 151–172.
- Ingenieria Industrial online. (n.d.). PROGRAMACIÓN LINEAL. Retrieved January 2, 2020.
- Narooei, K. D., Ramli, R., Rahman, M. N. A., Iberahim, F., & Qudeiri, J. A. (2014). Tool Routing Path

Optimization for Multi-Hole Drilling Based on Ant Colony Optimization. *World Applied Sciences Journal*, 32(9), 1894–1898.

Pinilla, J. A. A., & Castro, J. A. O. (2015). Optimización multiobjetivo en la gestión de cadenas de suministro de biocombustibles. Una revisión de la literatura, 20, No 1, 21–47.

Ramírez, C. (2018). *Comparación entre las técnicas metaheurísticas y los algoritmos de estimación de distribución en procesos estocásticos*.

Yao, Y. Z., & Xu, Y. R. (2007). Parameter analysis of particle swarm optimization algorithm. *Harbin Gongcheng Daxue Xuebao/Journal of Harbin Engineering University*, 28(11), 1242–1246.

Guillén, B. (2017). Vehicle Routing Problems in a real case study : optimization of distribution routes for a shoe company in the Canary Islands.

Montes, E. (2017). Metaheurísticas para el problema de ruteo de vehículos con ventanas de tiempo (VRP-TW)