

Construcción de un oleoducto de mínimo costo con algoritmos evolutivos

Andrés Bello

Facultad de Ingeniería
Montevideo, Uruguay
andresbu93@gmail.com

Juan Picca

Facultad de Ingeniería
Montevideo, Uruguay
jumapico@gmail.com

I. INTRODUCCIÓN

Los problemas de redes surgen en una gran variedad de situaciones. Las redes de transporte, eléctricas y de comunicaciones son ejemplo de las distintas situaciones que encontramos en la vida diaria. Es necesario representar a la red o redes, en diversas áreas como pueden ser: producción, distribución, planeación de proyectos, localización de instalaciones, etcétera. La representación permite tener un apoyo visual y conceptual para mostrar las relaciones entre los componentes del sistema, en ámbitos científicos, sociales y económicos.

Considerando el problema de transportar líquidos a través de sistemas de tuberías, representamos de manera gráfica la red que permite representar el problema real. Tomando como referencia un modelo real del proceso de exploración y producción de la paraestatal PEMEX, la cual se muestra en la figura 1, se puede apreciar la red de distribución básica desde el origen de la producción hasta su destino de almacenamiento dentro de alguna planta de procesamiento.

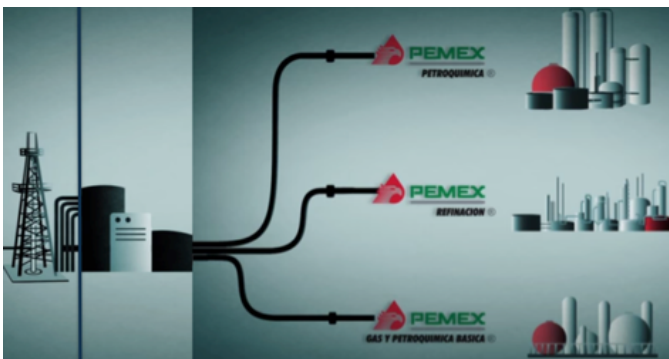


Figura 1. Red de distribución básica de PEMEX (fuente internet)

La topología de la red esta interconectada entre el origen y el destino, los cuales son conocidos, pasando por dos nodos intermedios en el trayecto para llevar a cabo distintas operaciones que deben ser aplicadas al producto que se va a transportar de un nodo a otro. La figura 2 muestra la topología de la red que vamos a tomar como base para el problema que se va a tratar.

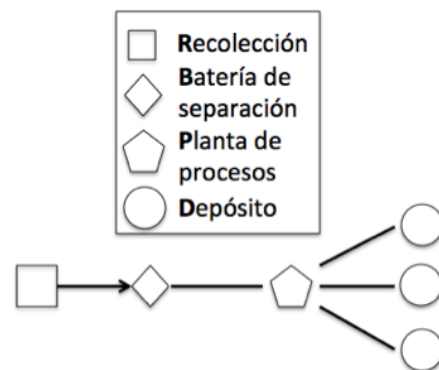


Figura 2. Topología de la red de tuberías

El objetivo de la planeación de la red, es diseñar una infraestructura con trayectorias para transportar líquidos valiosos a través de sistemas de tuberías, que permitan minimizar costos económicos considerando la presión del fluido en el diseño y los costos de operación y mantenimiento.

Para la resolución del problema planteado se podrían utilizar algoritmos deterministas, por ejemplo, algoritmos basados en metodologías Greedy. Pero aplicarlos a este tipo de problemas implica que los tiempos de computo se eleven considerablemente o no den resultados buenos. Por lo tanto se considera la utilización de algoritmos evolutivos, los cuales son útiles para la resolución de problemas de clase NP. Este tipo de algoritmo es el que se utiliza para resolver el problema planteado y se detalla en la siguiente sección.

II. ALGORITMOS EVOLUTIVOS

Los algoritmos evolutivos (AE) son estrategias utilizadas para la resolución de problemas de optimización, búsqueda y aprendizaje. Los algoritmos que se basan en el paradigma de algoritmos genéticos consisten en mantener un conjunto de individuos en una población que se mezclan y compiten entre sí para evolucionar y modificar la población inicial.

La técnica consiste en formar un conjunto de individuos (donde cada uno representa una posible solución al problema) y aplicar un conjunto de operadores probabilistas sobre ellos para emular la evolución. Inicialmente la población se forma aplicando métodos aleatorios o heurísticas determinadas al problema particular.

El AE asocia a cada individuo un valor de aptitud (que depende de la función objetivo definida para el problema), y aplica un conjunto de operadores evolutivos (cruzamiento y mutación) de forma iterativa para obtener nuevos individuos con mejor valor de aptitud.

Las condiciones de parada de las iteraciones pueden ser varias, por lo general se considera una cota de calidad de la función objetivo, una cantidad máxima de generaciones, o una cantidad máxima de generaciones en las que no mejore el valor óptimo encontrado hasta el momento.

Como resultado el AE devuelve la mejor solución encontrada en el proceso iterativo, tomando en cuenta la función objetivo.

III. DESCRIPCIÓN Y MODELADO DEL PROBLEMA

Se desea construir un oleoducto que transporte hidrocarburos en una superficie de terreno dada. Dicha construcción debe ser la que conlleve el menor costo económico de todas las posibles construcciones.

El oleoducto debe cumplir con las siguientes características:

- Parte de un pozo
- Debe pasar por un patín de recolección
- Debe pasar por una estación de recolección
- Finaliza la trayectoria en un destino
- No se debe colocar bombas en el tramo del pozo al patín de recolección

En la superficie del terreno hay varios patines de recolección y varias estaciones de recolección, por lo que se deberá encontrar aquellos por los cuales pasar para que la elección conlleve a la solución óptima del problema.

En la construcción del oleoducto se debe tener en cuenta que el terreno puede no estar nivelado, es decir, que cuente con diferentes valores de alturas. Esto conlleva que se deba considerar las posibles variantes en la presión del flujo. Cuando el nivel de presión disminuye de forma tal que impida al flujo llegar a un destino, se deben tomar decisiones pertinentes en la construcción de forma tal que la presión tenga un aumento considerable. También se debe tomar en cuenta que la presión no disminuye o varía simplemente por una diferencia de altura, sino también se deben considerar los pliegues formados por los cambios de direcciones, el rozamiento que genera la tubería con el flujo y la trayectoria del flujo.

El oleoducto puede ser construido por tuberías de diferente diámetro, en particular para el problema se consideran dos, las cuales se diferencian en el diámetro, d_1 y d_2 donde se cumple que: $d_1 > d_2$. Por otro lado también se cuenta con la utilización de bombas para aumentar la presión.

Así como se explicitó anteriormente uno de los problemas claves es la variación de los valores de presión en la trayectoria del flujo en el oleoducto. Es claro que cuando dicho valor es menor a un cierto valor mínimo permitido se debe realizar una acción para aumentarlo, para lo cual se siguen las siguientes estrategias:

1. Utilización de bombas
2. si se está en una tubería de diámetro d_1 se cambia el tipo de tubería a una de diámetro d_2 lo que provoca un aumento en la presión.

Por lo tanto el algoritmo debe tener en cuenta la colocación de bombas y la variación del tipo de tubería en los puntos en que se detecte que la presión disminuye debajo de un mínimo aceptable.

Otro punto a considerar en la construcción del oleoducto es la representación de la superficie del terreno. Se debe tener en cuenta que el costo de construcción no es uniforme para toda la superficie ya que hay zonas en las que está prohibido construir, por ejemplo en un aeropuerto, una escuela, etc. También hay que tener en cuenta que la superficie del terreno no está nivelada, por lo que se debe tener en cuenta las alturas de los puntos por los que pasa el oleoducto.

Para simplificar la implementación del algoritmo vamos a trabajar con una representación discreta del terreno que constará de tres matrices:

1. La primera representa si está permitido o no la construcción en un lugar dado indicando con el valor 1 que se puede construir y con el valor 0 que no.
2. La segunda matriz representa el costo total de compra y construcción en una celda de terreno dada.
3. Por último esta matriz contiene la elevación promedio de una celda (utilizada para los cálculos de presión y costo de tuberías).

Las matrices anteriores se deben construir a partir de la información geográfica.

Además se necesitan un punto de origen y el largo de una parcela de terreno para realizar la correspondencia entre una parcela de terreno a una celda de la matriz. También se utilizarán las coordenadas del pozo, los patines de recolección, las estaciones de recolección y el destino, estas coordenadas se reciben como celdas de una matriz.

Por último, son necesarios parámetros que indiquen los diámetros de las tuberías, las características del fluido (densidad, viscosidad, y velocidad con la que se espera que fluya por la tubería), los rangos de presión para cada tipo de tubería, cuánta presión da una bomba, los costos, etc.

Con toda la información mencionada anteriormente se calculan los costos de construcción.

IV. FORMULACIÓN MATEMÁTICA

A continuación se presenta la formulación matemática del problema planteado.

Dados los siguientes elementos:

- Sea el punto geográfico (O) por el cual debe iniciar la construcción del oleoducto (pozo)
- Sea el punto geográfico (D) donde debe finalizar la construcción del oleoducto

- Sea la matriz de costos de las parcelas $C_{i,j}$, donde: $c_{i,j} \forall i, j: i \leq N, j \leq N$ representa costo de construcción en la parcela representada con la celda i, j de la matriz
- Sea la matriz $LP_{i,j}$, donde:
 $lp_{i,j} \in \{0,1\} \forall i, j: i \leq N, j \leq N$. Es decir,
 $lp_{i,j} = 0$ si en la parcela representada por la celda i, j no está permitido construir que con
 $lp_{i,j} = 1$ si lo está
- Sea la matriz de alturas de las parcelas $A_{i,j}$, donde: $a_{i,j} \forall i, j: i \leq N, j \leq N$ representa la altura (en metros) promedio en la parcela representada por la celda i, j
- Sea $\{p_1, p_2, \dots, p_n\}$ con
 $n = \text{cantidad de patines de recolección}$ don de $p_i \forall i: 1 \leq i \leq n$ representa un patín de recolección.
- Sea $\{e_1, e_2, \dots, e_n\}$ con
 $n = \text{cantidad de estaciones de recolección}$ donde $e_i \forall i: 1 \leq i \leq n$ representa una estación de recolección.
- Sean las listas $T_1 = \{t_0, t_1, \dots, t_k\}$,
 $T_2 = \{t_0, t_1, \dots, t_w\}$, donde:
 $T_3 = \{t_0, t_1, \dots, t_z\}$
 - T_1 representa los puntos (t) por los que pasa el oleoducto en el tramo dado por $O - p_i$
 - T_2 representa los puntos (t) por los que pasa el oleoducto en el tramo dado por $p_i - e_j$
 - T_3 representa los puntos (t) por los que pasa el oleoducto en el tramo dado por $e_i - D$
- Sea la función de costo c tal que dado dos puntos pertenecientes a un tramo determine el costo de construcción del oleoducto en el sub-tramo determinado por dichos puntos. Esta función debe tener en cuenta el costo de utilización de bombas y tipos de tuberías para aumentar la presión el los puntos que sea necesario.

Se desean encontrar los puntos t_{i1}, t_{j2}, t_{h3} tales que conformen los tramos T_1 , T_2 y T_3 respectivamente tal que el costo de dichos tramos sea el menor posible de todas las posibles combinaciones existentes de puntos que conformen los tramos. Es decir, se quiere hallar CT tal que:

$$CT = \sum_{i=0}^{i=k} c(t_i, t_{i+1}) + \sum_{j=0}^{j=w} c(t_j, t_{j+1}) + \sum_{h=0}^{h=z} c(t_h, t_{h+1})$$

V. DISEÑO E IMPLEMENTACIÓN DE ALGORITMOS EVOLUTIVOS

En esta sección se detalla el diseño e implementación del AE para resolver el problema planteado.

V.I. Biblioteca Watchmaker

La biblioteca Watchmaker es un framework, desarrollado en Java, que utiliza el paradigma de orientación a objetos para implementar algoritmos evolutivos.[1]

El núcleo de la librería es la clase "EvolutionEngine". Para implementar un AE debemos instanciar dicha clase e invocar el método "evolve" que se encarga de realizar la ejecución y brindar la mejor solución encontrada. Dicho método recibe parámetros fundamentales para la resolución del problema. Por ejemplo:

- **populationSize**
Indica el numero de candidatos solución presentes en la población en cualquier instante de tiempo.
- **eliteCount**
Indica el número de candidatos que se preservan mediante elitismo, esto es, un subconjunto de la población con el mayor valor de fitness se mantiene sin cambios en la siguiente generación.
- **seedCandidates**
Conjunto de candidatos iniciales. El número de candidatos no puede superar al valor populationSize.
- **conditions**
Condiciones necesarias para que el algoritmo evolutivo finalice la ejecución.

Como resultado el método retorna la mejor solución encontrada.

Luego la biblioteca provee la clase FitnessEvaluator, en la cual debemos implementar en el método GetFitness la forma en la que se calculara el valor de fitness, es decir, la forma de indicar el valor de aptitud de un candidato a solución.

La biblioteca también brinda la clase CandidateFactory para generar una nueva población de candidatos. Y por último tenemos que definir uno o mas EvolutionaryOperator, que es la encargada de realizar las transformaciones a la población original, dependiendo de los operadores que se hayan definido.

V.II. Representación de soluciones

Los individuos se van a representar mediante una estructura de la siguiente forma:

- $P_1 = \{p_0, p_1, \dots, p_k\}$ Lista de coordenada(s) desde el origen hasta el patín de recolección.
- $P_2 = \{p_0, p_1, \dots, p_w\}$: Lista de coordenada(s) desde el patín de recolección hasta la estación de recolección.
- $P_3 = \{p_0, p_1, \dots, p_z\}$: Lista de coordenada(s) desde la estación hasta el destino.

Donde cada coordenada (p_i) representa una celda en las matrices.

La coordenada de origen y de destino no se incluyen en la representación ya que son los únicos puntos por los que puede comenzar y terminar la tubería.

V.III. Salida del algoritmo

Como resultado final, el algoritmo genera cinco archivos de salida:

- lands.out: Especifica las celdas por las que pasa el oleoducto.
- pumps.out: Especifica las celdas en las que deben colocarse bombas para aumentar la presión.
- cost.out: Indica el costo de la solución encontrada, además especifica el patín de recolección, la estación de recolección utilizada y otros valores referentes a la solución como el tiempo utilizado por el algoritmo (milisegundos) o la cantidad de generaciones utilizadas para obtener la solución.
- sections-diameter-1.out: Indica las celdas en las que se utiliza la tubería de tipo 1.
- sections-diameter-2.out: Indica las celdas en las que se utiliza la tubería de tipo 2.

V.IV. Generación de la población inicial

Para la generación de las soluciones iniciales se utilizara un algoritmo que genere de forma aleatoria los distintos tramos, determinando los patines de recolección y las estaciones de recolección también de forma aleatoria.

A continuación se detalla el algoritmo para la generación de un tramo entre dos puntos:

1. dado un punto de origen O y un punto de destino D se traza una línea imaginaria l entre O y D y a partir de esta se toma un sector circular que va desde $-\theta$ a $+\theta$ partiendo de l y con un radio máximo dado
2.
 1. si el punto D está en el sector circular y en la línea que va desde O hasta d no hay celdas prohibidas se elige el punto D como P , en caso contrario se utiliza el siguiente punto (2.2)
 2. sino se elige aleatoriamente un punto P dentro del sector circular que determinaremos por la elección de un ángulo aleatorio en el rango ($-\theta$, $+\theta$) y un radio aleatorio; si en la línea que va de O a P no hay celdas prohibidas nos quedamos con el punto P ; en caso contrario volvemos a realizar la selección hasta un número predeterminado de veces, a partir del cual se irá aumentando el ángulo del

sector (lo que permitirá a la solución “volver hacia atrás” y eventualmente sortear las celdas prohibidas).

3. se agrega el tramo (conjunto de celdas) desde el punto O hasta el punto elegido P ; si P no es D hacemos O igual a P y volvemos al punto 1; en caso contrario finalizamos el tramo.

Para generar un candidato generaremos un tramo entre el pozo y un patín de recolección elegido al azar.

Debido a la restricción de que no puede haber bombas entre el pozo y el patín de recolección se calculará la pérdida de presión en el tramo generado y en caso de que no alcance la presión para llegar al patín se descartará el tramo y se generará uno nuevo hasta hallar uno válido.

Luego se repetirá el algoritmo entre el patín de recolección y una estación de recolección elegida al azar y finalmente entre la estación de recolección elegida y el destino.

Como forma de brindar información al algoritmo desde un comienzo se implementó un algoritmo ávido, en particular utilizando la estrategia Greedy. Este consta en que dado un punto de origen O y un destino D , determine camino ($O, p_1, p_2, \dots, p_n, D$) de costo mínimo tomando en cada paso decisiones localmente óptimas, en particular para este problema las decisiones se basan en: sea p_i la celda actual, se calcula el costo de ir a cualquier adyacente de p_i (siempre que esté permitido) y se considera como próxima celda aquella que tenga costo mínimo entre las seleccionadas anteriormente. De esta forma se va formando el camino celda a celda, donde la siguiente posición será aquella de mínimo costo entre todas las posibles posiciones adyacentes.

Como para este problema se tienen varios patines de recolección y varias estaciones de recolección por las que el oleoducto tiene que pasar, se ingresan tantos individuos “greedy” como combinaciones entre patines y estaciones hayan, es decir, se encontraran individuos tal que vayan desde el origen al patín 1 y luego a la estación 1. Luego se incorporará otro que vaya desde el patín 1 hacia la estación 2, y así sucesivamente.

V.V. Función de fitness

Se busca minimizar el costo de construcción del oleoducto, por lo que tenemos que encontrar el mínimo de la función CT planteada en la sección IV:

$$\min\left(\sum_{i=0}^{i=k} c(p_i, p_{i+1}) + \sum_{j=0}^{j=w} c(p_j, p_{j+1}) + \sum_{h=0}^{h=z} c(p_h, p_{h+1})\right)$$

V.VI. Cálculo de pérdida de presión

Para este trabajo particular se utilizan dos tipos de tuberías, los cuales difieren en diámetro. Inicialmente se comienza construyendo el oleoducto con tuberías del tipo de mayor diámetro. Si en algún momento se detecta una caída de presión se cambia el tipo de tubería por uno de menor diámetro. Este cambio conlleva un aumento significativo en la presión del

flujo, aún así puede no ser suficiente, por ejemplo en los casos en que la tubería debe pasar por superficies con cambio altura. Para estos casos se considera colocar una bomba de presión.

Dada la restricción mencionada, el AE debe contar con un algoritmo que calcule las caídas de presión. Para ello se implementó una ecuación que determina el valor de la presión entre dos puntos. Se tomó como referencia la tesis: “Desarrollo de un algoritmo y programa en MATLAB para sistematizar y automatizar un modelo digital hidrodinámico en estado estable de un oleoducto para transporte de crudo liviano.”[1]

$$P_A = P_B + \rho * g * (H_B - H_A) + f * \rho * V * (L_B - L_A) / 2 * D_i$$

Donde:

- P_A = presión máxima requerida en el punto A.
- P_B = presión de entrada requerida en el punto B.
- ρ = densidad.
- g = aceleración de la gravedad
- H_B = altura en el punto B
- H_A = altura en el punto A
- f = factor de fricción
- V = velocidad del flujo
- L = distancia entre el punto A y el punto B
- D_i = diámetro interior de la tubería

Por simplicidad se decidió considerar varias variables como constantes, por ejemplo, la densidad, la velocidad del flujo, la aceleración de la gravedad, los diámetros y la fricción. Estos valores se parametrizaron de forma tal que sean parámetros de entrada del algoritmo.

V.VII. Operadores

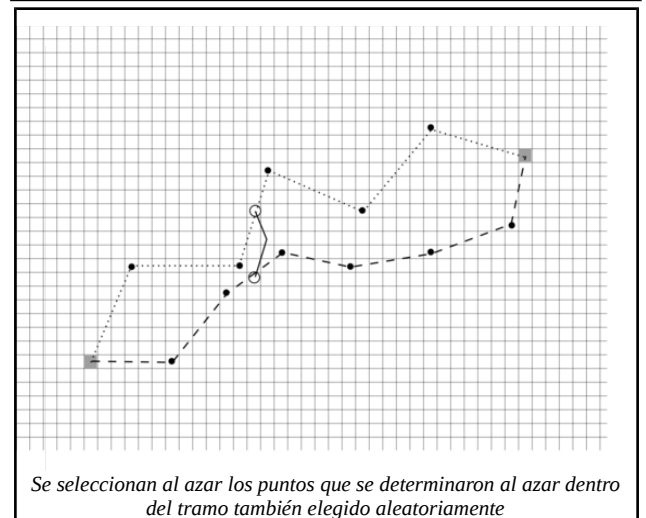
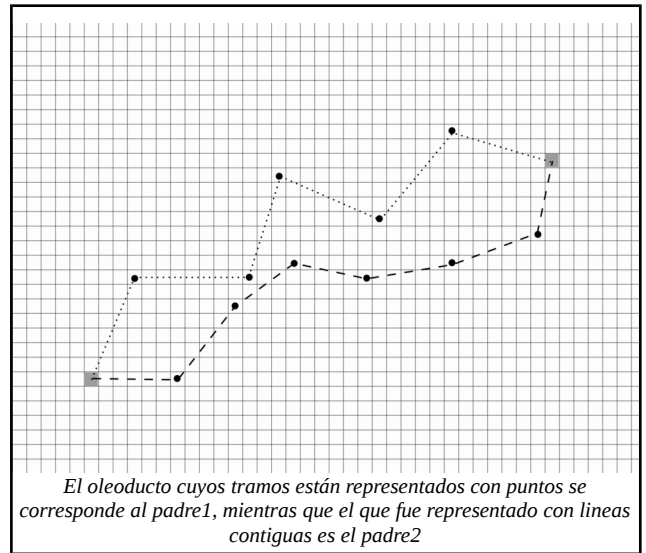
V.VII.1 Operador de cruzamiento.

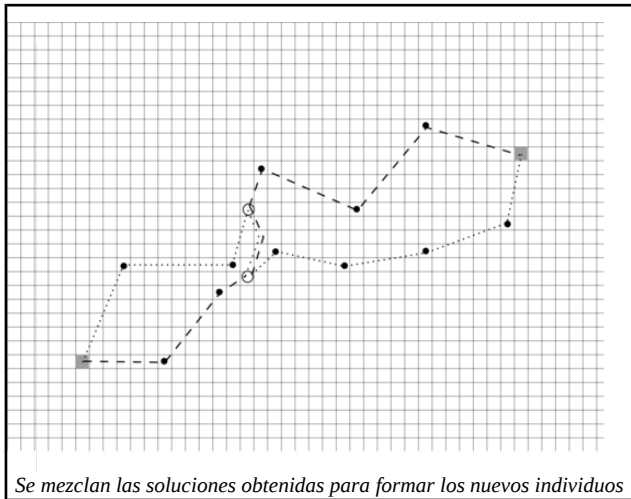
Se utilizará un operador de cruzamiento con las siguientes características:

1. se elige aleatoriamente una de las tres listas de celdas (t_1) de los que se compone una tubería
2. para el padre 1 se elige al azar una celda perteneciente a t_1 (p_1)
3. para el padre 2 se elige al azar una celda perteneciente a t_2 (p_2)
4. Se obtiene un subtramo (conjunto de celdas) con origen p_1 y destino p_2 utilizando el algoritmo Greedy comentado en la sección V.IV. En caso de que este no obtenga una solución válida se utiliza el mismo algoritmo aleatorio con el que se generan las soluciones iniciales.

5. se genera un hijo que desde el origen hasta p_1 contiene las celdas del padre 1, luego se agrega el conjunto de celdas de p_1 a p_2 generado en 4, y por último los datos desde p_2 hasta el destino se corresponden a los datos del padre 2.
6. se genera un hijo que desde el pozo hasta p_2 contiene los datos del padre 2, luego se agrega el conjunto de de p_2 a p_1 generado en 4 (se agregan los tramos en orden inverso), y por último los datos desde p_1 hasta el destino se corresponden a los datos del padre 1.

A continuación se despliega un a secuencia de imágenes ilustrando el algoritmo:



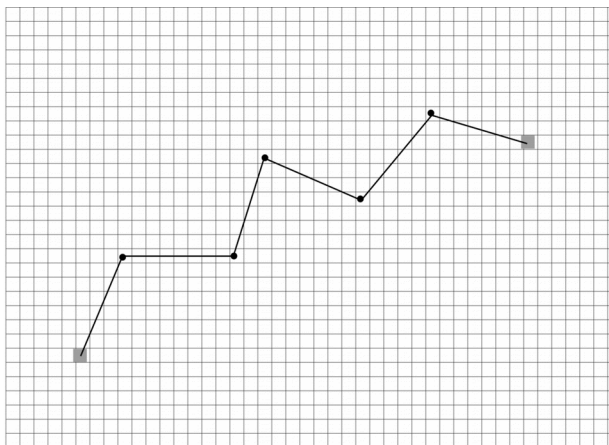


V.VII.2 Operador de mutación

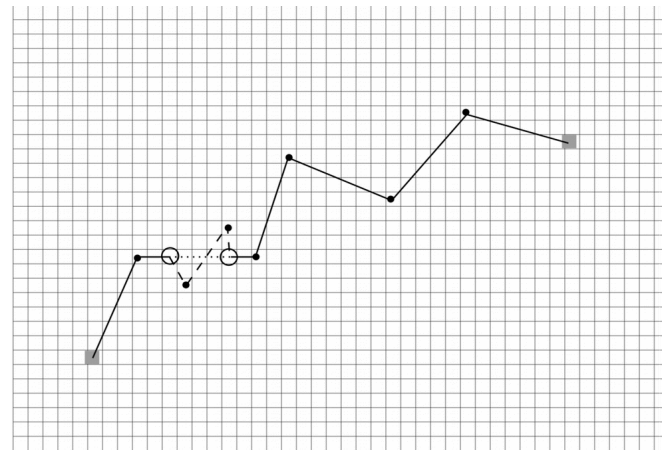
Se utilizará un operador de mutación con las siguientes características:

1. Dada s solución. Se determina al azar un tramo t_i .
2. Se determinan al azar dos puntos de t_i , p_1 y p_2 .
3. Se genera un camino de celdas desde p_1 a p_2 utilizando el algoritmo Greedy comentado en la sección V.IV. Descartando el camino actualmente construido entre dichos puntos.

A continuación se muestra una secuencia de imágenes ilustrando el funcionamiento del operador



Oleoducto original



El las líneas punteadas se representa el nuevo sub-tramo del oleoducto.

V.VIII. Estrategia de reemplazo

La estrategia de reemplazo utilizada es $(\mu+\lambda)$, la cual consiste en mantener tantos a los padres como a los hijos en la población. Por lo tanto la estrategia de selección actúa sobre todos los individuos, en particular la librería WatchMaker selecciona aquellos con mejores aptitudes para luego aplicar los operadores evolutivos. En consecuencia no es necesario tener un valor de elitismo determinado y tampoco una probabilidad de cruzamiento distinta de 1.

V.IX. Criterio de parada

Como criterio de parada se utiliza una combinación de dos condiciones:

- Cantidad de generaciones sin mejorar el valor de fitness (2000)
- Tiempo máximo de 5 minutos

VI. ANÁLISIS EXPERIMENTAL

En esta sección se presentarán los datos y valores obtenidos luego de realizar un análisis experimental sobre un conjunto de tres instancias independientes.

V.I. Plataforma de ejecución

El algoritmo evolutivo fue desarrollado con JAVA 8 y utilizando la librería WatchMaker. El PC utilizado para realizar el análisis experimental cuenta con procesador Inter Core i3-4150 3.50GHz, 8GB de RAM y Fedora 23 como sistema operativo.

V.II. Instancias del problema

Los datos de entrada para el algoritmo evolutivo son cuatro archivos:

- archivo .xml, con parámetros del problema
- archivo.cost, con los costos totales de compra y construcción en una parcela de terreno dada
- archivo.allowed, que indica si es posible construir o no en una parcela dada

- archivo.elevation, con la altura promedio para una parcela de terreno,

Para realizar el ajuste paramétrico se crearon tres instancias independientes, así como también se utilizaron otras tres instancias independientes para realizar la evaluaciones experimental. Dichos conjuntos de datos consisten en información extraída de de distintos puntos de ciudades de México[3].

Como se comentó en la sección III el algoritmo utiliza tres matrices: matriz de costo, matriz de elevación y matriz de permitidos. El procedimiento para cargar las matrices es el siguiente; se convierte el archivo .tif que contiene el territorio a formato .png para luego escalarlo al tamaño deseado (la cantidad de pixeles se corresponde con tamaño de las matrices) y luego exportarlo a formato .pgm. Dicho formato transforma la imagen en un archivo donde por cada línea tiene el valor ASCII del color correspondiente. Una rutina carga dicho archivo y lo recorre formando las diferentes matrices. La matriz de costo se forma multiplicando cada línea del .pgm por 10. En la matriz de elevación y permitidos se ingresan 0's y 1's respectivamente, de forma que no haya elevación ni lugares no permitidos, eso es a causa de que no se pudo encontrar información real. Igualmente se tuvo en cuenta la programación de dichas restricciones, por lo cual si en un futuro se quieren ingresar instancias de datos que contemplen las elevaciones y las posiciones de terreno no permitidas el algoritmo se comportará de la forma esperada, afectando la presión producto de la altura y tanto los operadores evolutivos como el algoritmo Greedy toman las decisiones pertinentes en caso de que se encuentren con posiciones de terreno no permitidas.

Además de la información de costos, elevaciones y lugares permitidos se utilizaron datos reales referentes a los valores de presión al salir del pozo, de los patín de recolección, de las estaciones de recolección, etc. Dichos valores fueron otorgados por Pedro Moreno. Estos valores son:

- Valor de presión a la salida del pozo: 17.570 Kg/cm²
- Valor de presión a la salida del patín de recolección: 36.200 Kg/cm²
- Valor de presión a la salida de la estación de recolección: 70.300 Kg/cm²
- Presión otorgada al colocar una bomba: 50 kg/cm²
- Diámetro de tubería mayor: 6" con un valor límite inferior de presión de: 21 kg/cm²
- Diámetro de tubería inferior: 4" con un valor límite inferior de presión de: 5 kg/cm²

Nota: El rango de presión de la tubería de 6" es de 21.0 kg/cm² a 40 kg/cm² se utiliza esta tubería para cualquier presión superior a 21 kg/cm².

Los valores de densidad, viscosidad y velocidad fueron extraídos de Viscopedia[4] y la **"Memoria-Resumen:**

"Proyecto para la instalación de una Refinería de Petróleo en Extremadura" [5]

y son:

- Velocidad = 5 m/s
- Viscosidad = 0.000107 m²/s
- Densidad = 870 kg/m³

VI.III. Ajuste paramétrico

La evaluación paramétrica se realiza para ajustar los parámetros antes de realizar la evaluación experimental, para ello se utilizan tres instancias de ciudades independientes y distintas combinaciones de parámetros. Los parámetros elegidos para ajustar son:

- Tamaño de la población, 50 y 100 individuos
- OffSpringMultiplier (valor que indica cuantos hijos crea a partir de los padres), 1 y 7.
- Probabilidad de mutación: 1% y 3%

Se ejecuto el algoritmo 30 veces para cada combinación de los parámetros anteriores y para cada una de las tres instancias. Los criterios de parados son los especificados en la sección V.IX.

Las tablas 1, 2 y 3 muestran los resultados de las ejecuciones para las tres instancias de datos respectivamente.

Se realizó el test de Kolmogorov-Smirnov para determinar si los valores obtenidos siguen una distribución normal y poder comparar las medianas de forma correcta. Como se puede ver en la tablas indicadas anteriormente, el resultado del test fue que los datos obtenidos no siguen una distribución normal para todas las instancias. Por lo tanto se realizó un test de rango para determinar aquella configuración de parámetros que obtenga los mejores valores de costo promedio. El resultado de este fue que la mejor combinación de parámetros es:

- #población = 100
- #offspring = 1
- $p_m = 0.1$

Resultados del análisis paramétrico

Instancia 1

#poblacion	#offspring	pm	min(CT)	min(\overline{CT})	SD cost	P-valor K-S	min(gen)	\overline{gen}	min(T)	min(\overline{T})	SD time
100	1	0,1	4389251.7498	4399612.7195	6034.1825539	0.85773	7937	7863.5666667	300011	299963.4	339.87914688
100	1	0,3	4393195.1481	4402541.1098	5275.8214221	0.83491	6663	6484.8666667	300011	300023.73333	11.805706783
100	7	0,1	4391315.7108	4252598.9375	5330.4643936	0	1161	1146.9655172	300102	300147.48276	82.122565495
100	7	0,3	4391249.1091	4401386.8328	5612.8622263	0.91469	938	922.76666667	300088	300189.96667	108.01579812
50	1	0,1	4390274.5853	4411413.5233	20502.634638	0.21344	12808	11439.8666667	250325	215884.73333	36125.36974
50	1	0,3	4392507.7888	4409111.6423	18046.186857	1.17E-01	10629	10743.0666667	278698	252690.76667	45530.742652
50	7	0,1	4387850.4294	4253220.3473	7958.4755839	7.79E-02	2347	2324.3103448	300083	300063.65517	38.170927477
50	7	0,3	4389877.7888	4407435.8508	17491.194995	0.077865	1868	1850.5	300039	300091.23333	44.206945416

Tabla 1 – Resultados de la primera instancia de datos

Instancia 2

#poblacion	#offspring	pm	min(CT)	min(\overline{CT})	SD cost	P-valor K-S	min(gen)	\overline{gen}	min(T)	min(\overline{T})	SD time
100	1	0,1	5299104.1303	5320217.4629	7295.2110793	0.87591	7172	7242.8	300014	300022.56667	14.085167873
100	1	0,3	5287143.9572	5324148.2623	11780.191126	0.44215	5750	5841.1666667	300060	300029.46667	18.537442693
100	7	0,1	5282806.9658	5317139.7031	11689.221195	0.78749	1029	1029.7333333	300161	300171.26667	84.346548417
100	7	0,3	5291423.9572	5323021.6721	11294.03623	0.59828	798	822.7	300278	300196.53333	108.57850595
50	1	0,1	5276039.2386	5317302.4766	13056.909942	0.23417	13671	14459.5	300014	296646.03333	13400.029746
50	1	0,3	5300886.7709	5322275.7774	8865.14	0.81556	12147	11744.133333	300026	299222.06667	4369.5222412
50	7	0,1	5273776.5979	5315366.5653	11932.982772	0.24889	2110	2095.9333333	300123	300078.46667	52.988439424
50	7	0,3	5305626.7709	5319085.6894	9420.3236566	0.42238	1732	1684.0333333	300198	300088.83333	53.727975974

Tabla 2 – Resultados de la segunda instancia de datos

Instancia 3

#poblacion	#offspring	pm	min(CT)	min(\overline{CT})	SD cost	P-valor K-S	min(gen)	\overline{gen}	min(T)	min(\overline{T})	SD time
100	1	0,1	3251271.1665	3262321.8332	7188.5238681	0.043785	11407	11211.7	300030	300020.43333	7.5552372031
100	1	0,3	3253011.1665	3263483.5446	6669.3199287	0.1465	9494	9498.0666667	300018	300018.73333	8.5739514623
100	7	0,1	3252241.1665	3262969.7452	6776.2140288	0.0088137	1598	1663.7	300049	300090.13333	49.87690595
100	7	0,3	3253391.1665	3262632.1673	6477.6800218	0.022179	1354	1335.6333333	300042	300148.2	66.54860811
50	1	0,1	3253341.1665	3265195.5893	6571.1798539	0.034158	18205	16641.566667	238758	213672.03333	37295.534486
50	1	0,3	3251841.1665	3265231.9667	8156.0796764	0.11915	18827	16075.466667	300015	261575.93333	33992.575414
50	7	0,1	3250361.1665	3260978.1665	6848.1445511	0.023057	3367	3384.6	300067	300044.73333	22.65777378
50	7	0,3	3251501.1665	3261476.23	7229.6121225	0.096679	2670	2709.2666667	300102	300067.4	30.922149187

Tabla 3 – Resultado de la tercer instancia de datos

Resultados de la comparación contra un algoritmo ávido**Instancia 1**

algoritmo	min(CT)	min(\overline{CT})	min(T)	min(\overline{T})	gen	P-valor MW	mejora sobre alg. Ávido	mejora promediosobre alg. Ávido
AE	4386338.5028	4393890.0547	211008	161094.2	7298	0.00E+00	30.28%	30.16%
Greedy	6291548.6348	6291548.6348	58	58	-	-	-	-

Instancia 2

algoritmo	min(CT)	min(\overline{CT})	min(T)	min(\overline{T})	gen	P-valor MW	mejora sobre alg. Ávido	mejora promediosobre alg. Ávido
AE	4959762.5074	4976512.1488	300010	296669.03333	6860	1.97E-11	29.25%	29.02%
Greedy	7010695.6031	7010695.6031	59	59	-	-	-	-

Instancia 3

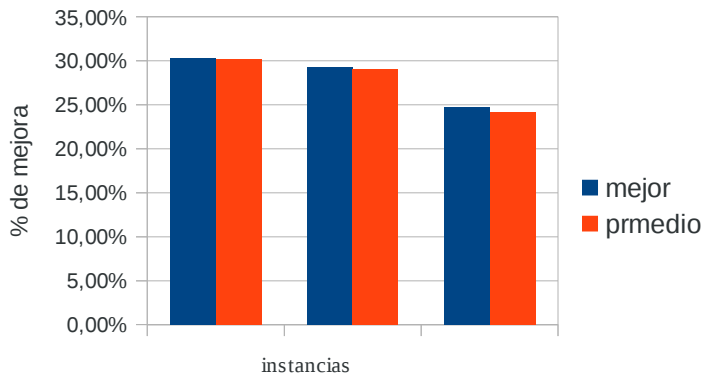
algoritmo	min(CT)	min(\overline{CT})	min(T)	min(\overline{T})	gen	P-valor MW	mejora sobre alg. Ávido	mejora promediosobre alg. Ávido
AE	5798790.0398	5838711.3731	300041	300029.1	7390	1.97E-11	24.68%	24.16%
Greedy	7698995.1058	7698995.1058	58	58	-	-	-	-

VI.IV. Comparación con un algoritmo ávido

Como forma de evaluar la eficiencia computacional y la calidad de las soluciones obtenidas mediante el algoritmo evolutivo se realiza una comparación de resultados entre el AE y un algoritmo ávido sobre instancias independientes a las utilizadas para la evaluación paramétrica.

En este caso se utilizó el algoritmo Greedy detallado en la sección V.IV obteniendo los resultados que se muestran en “Resultados de la comparación contra un algoritmo ávido”.

Como forma de obtener información estadística sobre si hay una diferencia significativa entre los resultados de ambos algoritmos se realizó el test de Mann-Whitney-Wilcoxon, obteniendo que realmente existe una diferencia considerable. Como es de notar en la tablas el algoritmo evolutivo mejora considerablemente frente al algoritmo Greedy, si bien el AE tiene una demora significativa, en las primeras generaciones obtiene mejores soluciones que el algoritmo Greedy. En la gráfica 1 se pueden apreciar los porcentajes de mejora para el mejor costo y costo promedio.



Gráfica 1 – Porcentaje de mejora frente al algoritmo Greedy

VII. BIBLIOGRAFÍA CONSULTADA

[1]. Librería WatchMaker : <http://watchmaker.uncommons.org/>

[2]. “Desarrollo de un algoritmo...”. R.Reyes, P.Augusto-
<http://bibdigital.epn.edu.ec/handle/15000/7878>

[3]. Página web con mapa de México: http://mapserver.inegi.org.mx/CEM3_mapa/

[4]. Página web viscopedia, <http://www.viscopedia.com/viscosity-tables/substances/crude-oil/>

[5]. Memoria-Resumen: “Proyecto para la instalación de una Refinería de Petróleo en Extremadura”, <http://www.grupoag.es/refineriabalboa/medioambiente/pdf/oleoductodecrudo.pdf>