



Universidad Autónoma del Estado de Morelos
Instituto de Investigación en Ciencias Básicas y Aplicadas
Centro de Investigación en Ingeniería y Ciencias Aplicadas

**Optimización de Trayectorias para Corredores aplicados a
Ductos para transporte de Líquidos Valiosos**

Tesis para obtener el grado de:
Doctorado en Ingeniería y Ciencias Aplicadas

presenta:

Pedro Moreno Bernal

Director: Dr. Marco Antonio Cruz Chávez

Sinodales: Dr. Sergio Alonso Serna Barquera, Dra. Margarita Tecpoyotl
Torres, Dr. David Juárez Romero, Dr. Alvaro Zamudio Lara, Dr. Martín
Heriberto Cruz Rosales, Dr. Martín Gerardo Martínez Rangel

Cuernavaca, Morelos

Mayo, 2019



UNIVERSIDAD AUTÓNOMA DEL
ESTADO DE MORELOS



Instituto de
Investigación en
Ciencias
Básicas y
Aplicadas

INSTITUTO DE INVESTIGACIÓN EN CIENCIAS BÁSICAS Y APLICADAS

Defensa de Tesis de Maestría en Ingeniería y Ciencias Aplicadas



"2019, a 100 años del asesinato del General Emiliano Zapata Salazar"

Cuernavaca, Morelos, a 04 de marzo de 2019.

DR. ROSENBERG JAVIER ROMERO DOMÍNGUEZ
COORDINADOR DEL POSGRADO
EN INGENIERÍA Y CIENCIAS APLICADAS
P R E S E N T E

Atendiendo a la solicitud para emitir DICTAMEN sobre la revisión de la TESIS "Optimización de Trayectorias para Corredores aplicados a Ductos para transporte de Líquidos Valiosos" que presenta el alumno **PEDRO MORENO BERNAL**, para obtener el título de **DOCTORADO EN INGENIERÍA Y CIENCIAS APLICADAS** con opción terminal en **TECNOLOGÍA ELÉCTRICA**.

Nos permitimos informarle que nuestro voto es:

NOMBRE	DICTAMEN	FIRMA
DR. DAVID JUÁREZ ROMERO		
DR. SERGIO ALONSO SERNA BARQUERA	APROBADO	
DR. ÁLVARO ZAMUDIO LARA		
DRA. MARGARITA TECPOYOTL TORRES	Aprobado	
DR. MARTÍN HERIBERTO CRUZ ROSALES (FCAel)	Aprobado	 27 Mayo 2019
DR. MARTÍN GERARDO MARTÍNEZ RANGEL (FCAel)		
DR. MARCO ANTONIO CRUZ CHÁVEZ	Aprobado	

PLAZO PARA LA REVISIÓN 20 DÍAS HÁBILES (A PARTIR DE LA FECHA DE RECEPCIÓN DEL DOCUMENTO)

NOTA. POR CUESTION DE REGLAMENTACIÓN LE SOLICITAMOS NO EXCEDER EL PLAZO SEÑALADO, DE LO CONTRARIO LE AGRADECEMOS SU ATENCIÓN Y NUESTRA INVITACIÓN SERÁ CANCELADA.

Resumen

El transporte de líquidos valiosos por ductos es una actividad vital para llevar la producción hacia los lugares de procesamiento, consumo o distribución. Los ductos son el medio de transporte más económico y seguro para mover grandes volúmenes a distancias considerables. Además, son eficientes y amigables con el ambiente ya que pueden reducir derrames, contaminación ambiental y acumulación de producto. Sin embargo, diseñar y construir ductos a través de largas distancia no es una tarea fácil. Los parámetros a considerar en el diseño de un ducto son: la ubicación de la trayectoria, la longitud del ducto, el diámetros interno, el tipo de material, el revestimiento, el espesor de la pared, los tramos libres, la protección catódica, entre otros.

La ubicación de la trayectoria que conecte una localidad origen con una destino implica identificar la mejor trayectoria dentro de un corredor geográfico con superficie variante. Dentro del corredor se deben evaluar múltiples criterios que hacen difícil de encontrar una solución única que sea la mejor entre múltiples alternativas. Por otra parte, el problema de encontrar la ruta más corta en un grafo no dirigido con pesos reales positivos es un problema combinatorio. Existe una gran cantidad de problemas de optimización combinatoria para los cuales no es posible garantizar con métodos clásicos de programación matemática que la solución obtenida sea la óptima. Además, estos métodos suelen ser poco eficientes para determinados problemas. Para abordar problemas más complejos de optimización, se justifica el uso de metaheurísticas debido a su capacidad de exploración y explotación del espacio de soluciones para abordar este tipo de problemas.

En este trabajo se presenta un método de optimización estocástica que explora distintas alternativas de rutas en un corredor para la mejor ubicación de la trayectoria para ductos de transporte terrestre. El método estocástico aplica una estructura de vecindad variable a partir de dos puntos aleatorios de la solución de la ruta y encuentra rutas semi-aleatorias a través de movimientos tabú en la trayectoria mientras minimiza una función de costo de distancia. Diez instancias son estudiadas a partir de un análisis espacial so-

bre dos escenarios realistas. Los escenarios se construyen con información de las cartas topográficas digitales E14B49, E14B69 y E14B79 a escala 1:50,000 del INEGI, que corresponden al estado de Veracruz, México. Los resultados experimentales indican que el algoritmo estocástico obtiene buenas soluciones respecto al algoritmo base (método ávido) con mejoras en la calidad de la solución superiores al 13% en promedio.

Abstract

The transport of valuable liquids through pipelines is a vital activity to take production to the places of processing, consumption or distribution. Pipelines are the most economical and safe system of transportation for moving large volumes over considerable distances. Also, pipelines are efficient and environmentally friendly as they can reduce spills, environmental pollution, and product accumulation. However, designing and building pipelines over long distances is not an easy task. The parameters to consider in the design of pipelines are the location of the trajectory, the length of the duct, the internal diameters, the type of material, the coating, the thickness of the wall, the free sections, the cathodic protection, between others.

The trajectory location that connects a source with a destination implies identifying the best trajectory within a geographical corridor on a variant surface. It involves factors which affect numerous ecological, social, and economic parameters when determining the best route location. On the other hand, the shortest path problem on an undirected weighted graph is a combinatorial problem. There are many combinatorial optimization problems for which it is not possible to guarantee with classical methods of mathematical programming that the solution obtained is the optimal one. These methods are usually inefficient for certain problems. To solve complex optimization problems, the use of metaheuristics is justified due to its capacity to explore and exploit the space of solutions to address this type of problem.

This work presents a heuristic optimization method by exploring GIS spatial alternatives to refine the route and to align it with the engineering design for transport pipelines in an early stage. The proposed heuristic applies a variable neighborhood structure selecting randomly two points from the path solution and finding routes by tabu movements across the path while minimizing a distance cost function. The experimental analysis compares the proposed method in two realistic problem scenarios in Veracruz, Mexico. Improvements up to 13% in the distance of the route at one of the scenarios are reported.

Índice general

Índice de figuras	VIII
Índice de tablas	XI
1. Introducción	1
1.1. Antecedentes	3
1.2. Problema de seleccionar la trayectoria (PST)	11
1.3. Objetivos de la investigación	12
1.4. Alcance de la investigación	13
1.5. Contribuciones de la tesis	14
1.6. Organización de la tesis	14
2. Planteamiento del problema	16
2.1. Diseño de ductos en México	16
2.2. Modelos de datos geográficos	21
2.3. Problema de seleccionar la ruta óptima	25
2.4. Complejidad del problema	29
2.5. Formulación matemática	38
3. Metaheurísticas	41
3.1. Métodos de búsqueda	41
3.1.1. Búsqueda Local Iterada	43
3.1.2. Métodos ávidos	45
3.1.3. Metaheurísticas	46
3.1.3.1. Tipos de Metaheurísticas	48
4. Algoritmo de Recocido Simulado para el PST	52
4.1. Recocido Simulado	52
4.2. Metodología para tratar información geográfica ASCII-TXT	57
4.3. Algoritmo ávido	65
4.4. Algoritmo de Recocido Simulado para el PST (RS-PST)	68

4.5. Estructura de vecindad	70
4.6. Análisis de sensibilidad del algoritmo RS-PST	78
4.7. Análisis de la complejidad del algoritmo RS-PST	85
5. Análisis de resultados	89
5.1. Análisis experimental	89
5.1.1. Entorno de ejecución	90
5.1.2. Instancias de prueba y escenarios	90
5.2. Parámetros sintonizados del algoritmo RS-PST	98
5.3. Resultados numéricos	102
5.3.1. Análisis de eficacia	102
5.3.2. Análisis de eficiencia	105
5.3.3. Vistas de las trayectorias en 2D y 3D en el mapa	107
6. Conclusiones y trabajo futuro	115
6.1. Conclusiones	115
6.2. Trabajo futuro	117
Nomenclatura	120
Bibliografía	122
A. Código fuente del algoritmo ávido	131
B. Código fuente del algoritmo RS-PDT	133
C. Eficiencia algorítmica	135
D. Productos obtenidos entre 2013 y 2019	138

Índice de figuras

2.1. SISTRANGAS	18
2.2. Esquema básico de transporte de hidrocarburos	20
2.3. Integración de datos en sentido horizontal	22
2.4. Concepto de capa en GIS	22
2.5. Modelo ráster	23
2.6. Modelos vectorial	23
2.7. Representación de una capa de costos acumulados y una ruta encontrada	25
2.8. Comportamiento de las funciones de complejidad algorítmica	31
2.9. Representación de un grafo rectangular	36
2.10. Representación de un ciclo hamiltoniano en el grafo rectangular	37
3.1. Comportamiento del método BLI en la resolución de problemas: (izq.) función unimodal, el método es capaz de encontrar la solución óptima del problema, (der.) función multimodal, el método puede fallar en encontrar el óptimo global y retornar un óptimo local del problema	45
3.2. Conceptos vinculados con el patrón de búsqueda de una metaheurística: (izq.) exploración del espacio de soluciones y (der.) explotación de buenas soluciones encontradas	48
4.1. Procesos de la metodología propuesta	59
4.2. Representación de una trayectoria sobre la matriz de costos acumulados	61
4.3. Representación del grafo de malla rectangular del mapa	62
4.4. Porcentaje de influencia de elevación y suelo	64
4.5. Suma de porcentajes de influencia para obtener la matriz de costos acumulados	64
4.6. Representación de la solución a través de un vector para una trayectoria 2D	67
4.7. Representación de la trayectoria inicial (amarillo) y la óptima (rosa)	70

4.8. Representación del espacio de soluciones de una estructura de vecindad	71
4.9. Solución s' (color verde) que mejora el costo de s	73
4.10. Solución s' (color verde) que no mejora el costo de s	73
4.11. Solución s_v para contar el número de visitas de cada celda	74
4.12. Solución actual s	75
4.13. Solución semi-aleatoria s_v a partir de nodos visitados	76
4.14. Solución s_v	76
4.15. Solución s'	77
4.16. Soluciones encontradas por el RS-PST con la estructura de vecindad variable	77
4.17. Costo de la solución (distancia) en función de la Temperatura (T_0)	79
4.18. Costo de la solución en función de la Temperatura (T_0)	81
4.19. Costo de la solución en función del factor de decremento (β)	82
4.20. Costo de la solución en función de la longitud de la cadena de Markov	83
4.21. Costo de la solución en función del criterio de paro (T_f)	84
5.1. Ubicación de la carta topográfica E14B49 en el estado de Veracruz	92
5.2. Conjunto topográfico E14B49	92
5.3. Ubicación de las cartas topográficas E14B69 y E14B79 en el estado de Veracruz	93
5.4. Conjunto topográfico de la carta E14B69	93
5.5. Conjunto topográfico de la carta E14B79	94
5.6. Capada de datos DEM de E14B69 y E14B79	95
5.7. Costo de la solución (distancia) en función de la Temperatura	99
5.8. Costo de la solución (distancia) en función de la Temperatura	100
5.9. Costo de la solución (distancia) en función de la Temperatura	100
5.10. Costo de la solución (distancia) en función de la Temperatura	101
5.11. Costo de la solución (distancia) en función de la Temperatura	101
5.12. Trayectorias ávida, RS-PST y oleoducto real	105
5.13. Trayectoria inicial vs óptima de la instancia 1 en 2D	108
5.14. Trayectoria inicial vs óptima de la instancia 1 en 3D	108
5.15. Trayectoria inicial vs óptima de la instancia 2 en 2D	109
5.16. Trayectoria inicial vs óptima de la instancia 2 en 3D	109
5.17. Trayectoria v de la instancia 3 en 2D	110
5.18. Trayectoria inicial vs óptima de la instancia 3 en 3D	110
5.19. Trayectoria inicial vs óptima de la instancia 4 en 2D	111
5.20. Trayectoria inicial vs óptima de la instancia 4 en 3D	111
5.21. Trayectoria inicial vs óptima de la instancia 5 en 2D	112

5.22. Trayectoria inicial vs óptima de la instancia 5 en 3D	112
5.23. Trayectoria inicial vs óptima de la instancia 6 en 2D	112
5.24. Trayectoria inicial vs óptima de la instancia 7 en 2D	113
5.25. Trayectoria inicial vs óptima de la instancia 8 en 2D	113
5.26. Trayectoria inicial vs óptima de la instancia 9 en 2D	114
5.27. Trayectoria inicial vs óptima de la instancia 10 en 2D	114
D.1. Ponencia en Jornadas Académicas	138
D.2. Estancia de investigación en el CINVESTAV	139
D.3. Participación en proyecto PROINNOVA-CONACYT	140
D.4. Participación en proyecto PROINNOVA-CONACYT	141
D.5. Estancia d investigación en la Universidad de la República, Uru- guay	142
D.6. Artículo en revista “Programación matemática”	143
D.7. Artículo de divulgación en revista Inventio-UAEM	144
D.8. Artículo de divulgación en revista Inventio-UAEM	145

Índice de tablas

4.1. Función de distribución de Boltzmann.	54
4.2. Capas de datos con sus respectivos costos de viabilidad del escenario 1	63
4.3. Capas de datos con sus respectivos costos de viabilidad del escenario 2	63
4.4. Parámetros de control del algoritmo RS-PST	79
4.5. Rangos de los parámetros de control del algoritmo RS-PST	80
4.6. Muestras definidas para cada uno de los parámetros de control del algoritmo RS-PDT	80
4.7. Análisis estadístico del comportamiento del parámetro de control T_0	82
4.8. Análisis estadístico del comportamiento del parámetro de control β	83
4.9. Análisis estadístico del comportamiento del parámetro de control L_{Markov}	84
4.10. Valores sintonizados de los parámetros de control del algoritmo RS-PST	85
5.1. Coordenadas UTM de los DEM	95
5.2. Instancias de prueba del escenario 1	96
5.3. Instancias de prueba del escenario 2	96
5.4. Instancias de prueba de los escenarios 1 y 2	97
5.5. Valores sintonizados de los parámetros de control del algoritmo RS-PST	98
5.6. Análisis estadístico de la función objetivo del algoritmo RS vs BFS	103
5.7. Porcentaje de mejora de la calidad de la solución del RS-PST vs ávido	104
5.8. Resultados comparativos de la calidad de la solución con respecto a un oleoducto del mundo real	104

5.9. Comparación de eficiencia en minutos de los algoritmos ávido
vs RS-PST 106

C.1. Medidas de eficiencia estándar 137

Capítulo 1

Introducción

El transporte de hidrocarburos y sus derivados se realiza a través de ductos, camiones, buque-tanques, ferrocarriles, entre otros medios [1]. El medio de transporte más económico y seguro para mover grandes volúmenes de hidrocarburos a distancias considerables es a través de ductos. El transporte de hidrocarburos a través de ductos requiere suficiente infraestructura de transporte en los activos de exploración y producción que opere en condiciones ideales [2]. Los ductos suelen ser más eficientes que otros medios para transportar líquidos valiosos, además de que son amigables con el ambiente debido a que pueden reducir en gran medida derrames, contaminación ambiental y acumulación de producto[3].

Con el desarrollo de la explotación y producción de hidrocarburos es necesario integrar nueva infraestructura de recolección y transporte por ductos para trasladar materia prima desde los yacimientos hasta las plantas de procesamiento, así mismo, para distribuir productos derivados hasta los centros de consumo. Sin embargo, diseñar y construir ductos a través de largas distancias no es una tarea fácil. Seleccionar la ubicación donde deben ser construidos depende en muchas ocasiones de condiciones económicas, climáticas y geográficas, en las cuales se deben considerar obstáculos topográficos, geológicos, hidrológicos, naturales y artificiales [4]. Además, el diseño de un

ducto requiere considerar distintos parámetros como son: *seleccionar la ruta, la longitud del ducto, el diámetro interno, el tipo de material, el revestimiento, el espesor de la pared, los tramos libres y la protección catódica* [5]. También, es necesario considerar las presiones de operación para determinar la cantidad de estaciones de compresión y bombeo, así como la potencia de las mismas [6].

En el proceso de planeación del ducto, seleccionar la ruta o trayectoria es un paso importante que afecta otros parámetros a considerar en el diseño. Generalmente, un ducto se compone de tuberías de acero que se extienden a través de grandes distancias, las cuales suelen enterrarse y protegerse con un recubrimiento anticorrosivo especial, pero también pueden ir superficialmente. Por este motivo, el proceso de seleccionar la ubicación del ducto es de vital importancia para el proyecto. Además, seleccionar la trayectoria del ducto en una etapa temprana permite tomar decisiones a tiempo [7]. La selección de la trayectoria debe considerar todos los factores que afectan la ruta tales como uso de suelo, topografía, elevación, entre otros. Así mismo, el terreno y los obstáculos naturales y artificiales deben ser tomados en cuenta para seleccionar la ruta en la etapa de diseño [8]. Por lo tanto, es importante definir la mejor ruta considerando factores como costos económicos, problemas legales, ambientales, derechos de vía y requerimientos del proyecto [9]. Una vez seleccionada la ruta y aprobada en campo, ésta puede impactar en la disminución de costos ambientales, sociales y económicos permitiendo un ahorro de materiales de construcción así como de costos de inversión, de capital, de operación y mantenimiento [6].

1.1. Antecedentes

El problema de seleccionar la trayectoria (PST) a través de un terreno heterogéneo es un problema complicado. El problema de selección involucra factores ecológicos, sociales y económicos que afectan la ubicación de la mejor ruta [10]. Además, determinar rutas factibles a través de un terreno implica en muchas ocasiones evitar obstáculos que son considerados como restricciones [11]. Por otra parte, el problema de encontrar el camino óptimo entre dos ubicaciones físicas es un problema muy común en aplicaciones tales como cartografía, robótica y de Sistemas de Información Geográfica (GIS, por sus siglas en inglés). El camino óptimo corresponde a la acumulación mínima de costos entre dos localidades físicas [12]. El costo no necesariamente tiene un enfoque económico, puede estar determinado por distancia geográfica, tiempo o aspectos ambientales, sociales o técnicos [13].

Los modelos de datos de GIS permiten manejar eficientemente datos del terreno de estudio. Además, las herramientas GIS permiten combinar modelos de datos de distintas fuentes y distintas escalas de mapas que pueden ser una ayuda significativa para el análisis espacial [14]. Los problemas de análisis de redes GIS han sido resueltos utilizando los algoritmos Dijkstra [15] y Bellman-Ford [16], entre otros métodos. El análisis de redes GIS basado en el problema de la ruta más corta se ha utilizado en aplicaciones de planificación de rutas tales como tuberías, ductos, autopistas, líneas de transmisión, entre otras [17]. En muchas aplicaciones de planificación de rutas, es necesario evaluar varias alternativas para la ruta más corta que se desea, en lugar de una ruta más corta [18]. Depende de agregar elementos adicionales que no necesariamente producirían la ruta más corta, sino una mejor ruta que sería una ruta ad hoc para el problema. Por lo tanto, se hace necesario explorar diferentes rutas

competitivas a través de la red GIS. Además, la búsqueda de una ruta óptima obedece a objetivos tales como la distancia o el costo que se debe minimizar. Los problemas del mundo real son más complicados de resolver porque son problemas complejos de planeación de rutas.

Los trabajos recientemente publicados sobre el problema de selección de ruta a través de redes GIS han tratado el problema como un problema de ubicación de ruta óptima. Así mismo, se ha realizado una amplia investigación en encontrar la ruta más corta en redes GIS utilizando el algoritmo Dijkstra. En muchos de los trabajos reportados, el algoritmo Dijkstra o una variante del algoritmo son implementados para abordar el problema de seleccionar la ruta óptima a través de largas distancias en aplicaciones del mundo real. Sin embargo, Dijkstra proporciona un método exacto para encontrar la distancia más corta que conecta dos puntos, pero no funciona si alguno de los bordes tiene un valor negativo. Tampoco funciona para explorar más alternativas de rutas dentro de un área determinada puesto que siempre encontrará la misma ruta.

Para evaluar múltiples alternativas de rutas, se aborda el problema de ubicación del corredor que implica identificar la mejor ruta a través de un terreno que conecta dos puntos. Este problema ha sido estudiado por varios investigadores [19, 20, 21, 11, 17]. Goodchild y Huber utilizan representación de datos ráster y Huber utiliza representación de datos vectoriales. Así mismo, Scaparra y Medrano abordan el problema de la ubicación del corredor como k-ésimo camino más corto (Kth-Shortest Path Problem) o ruta de acceso más corta (NSP, Near-Shortest Path Problem) respectivamente. El problema k-ésimo camino más corto es una generalización del problema de la ruta más corta. El algoritmo no solo encuentra la ruta más corta, sino k-1 rutas en orden de costo no decreciente. k es el número de caminos más cortos para

encontrar. El problema puede restringirse para tener la ruta más corta k sin bucles o con bucle. El algoritmo de la ruta de acceso más corta se ha identificado como eficaz para generar conjuntos de buenas alternativas de ruta para diseñar nuevas infraestructuras. Si bien el algoritmo en sí mismo es más rápido que otros enfoques incluido el problema Kth-Shortest Path, el tamaño del conjunto de soluciones y el tiempo de cálculo aumentan exponencialmente a medida que aumentan los parámetros o el tamaño del problema. Por otra parte, independientemente del método que se utilice, el análisis espacial eficiente para encontrar la ruta más corta entre dos localidades depende de la matriz de costos. Habitualmente, la matriz de costos se genera aplicando distintos procedimientos analíticos en las capas de los datos, incluyendo datos topográficos, geológicos, del terreno y ambientales [22].

En aplicaciones del mundo real, la ubicación óptima de la ruta del ducto ha sido resuelta por Baeza et al. [23], Marcoulaki et al. [24], Wei et al. [25], entre otros, mediante la aplicación de diferentes métodos de optimización. Baeza comparan el método de Dijkstra versus el método de optimización de colonias de hormigas (ACO) para el problema de la ruta óptima de la tubería de concentrado de mineral. Su enfoque consiste en encontrar la solución de ruta más corta en un terreno complejo en términos de costo y precisión. Incluye costos de longitud y pendientes de la ruta en el modelo. Marcoulaki et al. desarrollaron un framework de optimización basado en optimización estocástica para resolver los problemas de seleccionar la ruta y del diseño de equipo en un ducto para el transporte de fluidos utilizando herramientas GIS. Wei et al. proponen una “plataforma de diseño de optimización de la red de tuberías del sistema de recolección de gas” para optimizar el diseño de la red de tuberías de recolección y transmisión de gas en un bloque del campo petrolero de Daqing, China. El diseño de la red de tuberías considera parámetros como el

diseño de la topología, el diámetro de la tubería, el espesor de la pared y la potencia de trazado eléctrico, entre otros parámetros. Su propuesta de optimización puede reducir la longitud de la recolección de gas y la recolección del gasoducto, lo que ahorra un costo de inversión considerable en el diseño de la red del gasoducto.

Kang y Lee [26] optimizan el diseño de una ruta par un ducto en alta mar basándose en la ruta de menor costo para evitar obstáculos a través del algoritmo Laplacian-Smoothing para hacer la ruta más suave. El método y el algoritmo son efectivos en el diseño de rutas de tuberías offshore con presencia de obstáculos. Rodriguez et al. [27] optimizan el diseño de redes de tuberías para hidrocarburos en términos de características de transporte. Su enfoque estocástico minimiza el costo de la ruta del ducto de la distribución de productos entre campos petroleros y plantas de procesamiento, teniendo en cuenta las características geográficas. Las soluciones están representadas por vectores binarios. Cada vector de n bits corresponde a n ubicaciones posibles para los nodos de concentración (campos petroleros). Una solución generada por el algoritmo de recocido simulado modifica su orden en relación con la matriz de costos de los nodos para evaluar la calidad de la solución. El área de estudio se ubica en Santa Cruz, Argentina y se aplicó al diseño de un sistema de ductos de transporte de gas natural. Los parámetros reportados del algoritmo de recocido simulado que mejores soluciones obtuvieron son: temperatura inicial 100, longitud de cadena de Markov 3 y factor de decremento beta 0.99.

Por otra parte, algunas investigaciones han estado utilizando software comercial y software libre para encontrar la ruta más corta basada en el algoritmo de Dijkstra y sus variantes. Middleton y Bielicki [28] proponen una versión modificada del método Dijkstra para planear una red de captura y almacenamiento de CO₂. En su enfoque, consideran información geográfica de eleva-

ción, áreas urbanas, parque estatales y nacionales, cruce de ríos, caminos y vías férreas, como criterio para encontrar la mejor ubicación de la tubería. El modelo propuesto genera una eficiente red de tuberías para transportar CO₂ estimando y minimizando costos. La instancia utiliza 37 fuentes de CO₂ y 14 embalses en el estado de California, EE. UU. El modelo a optimizar es de programación lineal entera mixta con múltiples restricciones, en la que la función objetivo es minimizar el costo total de construcción de la red de transporte. El solver utilizado para resolver el problema es ILOG's CPLEX 11.0.

Brimberg et al. [29] utilizan una búsqueda Tabú (Tabu Search), una búsqueda de vecindad variable (VNS por sus siglas en inglés) y un método de ramificación y poda (Branch and Bound) para resolver el problema del diseño de un sistema de ductos de transporte de hidrocarburos que se compone de 33 nodos. El modelo a optimizar es un modelo de programación mixta (binaria-entera). El escenario de prueba es un sistema real ubicado en el sur de Gabón, Africa. El solver utilizado para resolver el problema de optimización es CPLEX 7.0. Los resultados experimentales muestran que las metaheurísticas fueron más eficientes que el método de ramificación y poda en obtener la solución óptima encontrada. El tiempo que demora el B&B en encontrar la solución óptima es de 4.2 segundos, mientras que las metaheurísticas TB y VNS demoraron menos de un segundo.

Kjaernested et al. [30] proponen dos metodologías para seleccionar la ruta para el diseño de una tubería de transporte de fluido geotérmico en Islandia, basada en el problema de la ruta más corta utilizando el paquete de algoritmos genéticos de MATLAB. Los datos de elevación (DEM) utilizados están en escala 1:25,000. El método VTDT (Variable Topography Distance Transform) utilizado en el trabajo ofrece una buena y funcional forma de diseñar rutas para tuberías con un impacto visual mínimo. La única limitante es la resolución

de la escala del DEM. El algoritmo NSGAll tiene un rendimiento peor que el VTDT, sin embargo el método es prometedor debido a la posibilidad de incluir otras funciones objetivo.

Górniak-Zimroz [31] propone un método de optimización multi-criterio para determinar la ubicación óptima para una banda transportadora de una compañía minera utilizando restricciones técnicas y ambientales de la región baja de Silesia en Polonia. El método es desarrollado y probado utilizando el software ArcGIS. El software logra identificar la mejor ubicación minimizando el impacto ambiental y social de la ruta propuesta.

Por otra parte, Soltani et al. [32] presentan una aplicación de planeación de ruta en sitios de construcción con múltiples objetivos. El trabajo compara los métodos Dijkstra, A* (pronunciado A estrella) y un algoritmo genético. Las pruebas experimentales muestran que la ruta más corta producidas por los algoritmos Dijkstra y A* son similares con costos idénticos. Sin embargo, los tres métodos producen diferentes rutas. La ruta óptima la obtiene el algoritmo Dijkstra, seguida por el algoritmo A*, pero para abordar problemas de gran escala, el algoritmo Dijkstra es ineficiente comparado con A*. El algoritmo genético obtiene soluciones cercanas al óptimo obtenida por Dijkstra en menor tiempo.

Luettinger y Clark [33] proponen un proceso de selección GIS considerando costos de construcción para determinar la ruta óptima entre dos localidades basado en el problema de la ruta de menor costo. La ruta resultante fue aceptada en el proyecto de construcción del acueducto del distrito metropolitano de Salt Lake y Sandy, Utah, EE.UU.

Devine y Haneberg [34] proponen una metodología para seleccionar la ruta de menor costo para un oleoducto submarino ubicado en el Artico. La resolución de los datos es de 5m por celda y el área de estudio es de 300 km^2

proveídos por la encuesta geológica de Canadá y C-CORE. Las restricciones para buscar la ruta son evitar pendientes mayores a 7° y evitar depresiones mayores a $5m$ de profundidad. El software utilizado es ArcGIS.

Abudu y Williams [35] proponen una metodología basada en GIS para identificar la ruta óptima y de menor costo para una tubería de hidrocarburos considerando restricciones ambientales, económicas y de seguridad asociadas al diseño de ductos de hidrocarburos. El área de estudio se encuentra en el sureste de Uganda, Africa. El modelo se aplica para diseñar un ducto que conecte una refinería con una terminal de distribución en un área de $29 km^2$. El solver utilizado para abordar el problema es ArcGIS.

A pesar del uso de metaheurísticas, en muchos casos, las estrategias ávidas como Dijkstra o A* (a-estrella) son más eficientes que otro tipo de algoritmos para instancias pequeñas, y para instancias grandes o en donde falla el algoritmo de Dijkstra, el enfoque normal es utilizar la programación dinámica o las metaheurísticas. Los trabajos relacionados muestran que existe gran interés en estudiar distintas aplicaciones del problema de planificación, diseño y/o selección de rutas, también para ubicación de corredores para buscar soluciones factibles de trayectorias aplicadas a ductos o tuberías de transporte líquidos valiosos. Así mismo, se muestra que la mayoría de las propuestas utilizan software comercial que implementan métodos exactos tales como Dijkstra, entre otros.

Para llevar a cabo la planeación de la trayectoria por donde debe ser construido un ducto, se requiere realizar un análisis espacial sobre la superficie del terreno. GIS es la herramienta utilizada para planear la trayectoria a partir de modelos de datos geográficos y técnicas de análisis espacial. El software comercial GIS más utilizado para realizar la selección de ruta o la ubicación de un corredor aplicado a ductos o tuberías es ArcGIS de la compañía ES-

RI [36]. Este software busca la ruta más corta utilizando el algoritmo Dijkstra y sus variantes [37]. Por otra parte, el software libre más utilizado para gestionar información geográfica, geoespacial y para realizar análisis espacial es QGIS [38], proyecto oficial de Open Source Geospatial Foundation (OSGeo).

Existen muchas herramientas de software GIS, tanto privadas como comerciales. La mayoría de las herramientas GIS soporta distintos modelos de datos y aplica diversas técnicas de análisis espacial. Por ejemplo, las herramientas de la empresa ESRI están bien documentadas y sus interfaces de usuario son amigables y fáciles de usar. Además, cuentan con una gran número de herramientas y técnicas de análisis espacial. Sin embargo el costo de licenciamiento no es económico y están diseñadas para atender tareas comunes, entre otras desventajas. Por otra parte, las herramientas GIS de software libre no son tan intuitivas para el usuario, a pesar de que permiten la interacción e improvisación sobre las aplicaciones que se desarrollan sobre múltiples plataformas. Además, incorporan técnicas para análisis estadístico, análisis espacial y geo-procesamiento que requieren de una comprensión compleja de las mismas. Una desventaja del software libre es que no suele contar con soporte técnico oficial y establecido. Otra desventaja es que suelen estar diseñadas para uso de propósitos generales.

Por tal motivo, es necesario crear herramientas y metodologías de análisis espacial que permitan considerar criterios de selección que correspondan con las metas y objetivos de diseño. Así mismo, que permitan abordar problemas complejos del mundo real en la toma de decisiones.

1.2. Problema de seleccionar la trayectoria (PST)

El problema de seleccionar la trayectoria de un ducto terrestre se puede simplificar en dos componentes espaciales, por un lado la ubicación del punto de origen y por otro, la ubicación de la trayectoria del ducto hasta su destino. Un ducto puede atravesar por distintos tipos de terreno o de suelo, lo cual involucra variables de distinta naturaleza. Por ejemplo, sobre el terreno por donde debe pasar el ducto suelen presentarse una gran cantidad de restricciones propias de la región geográfica, las cuales actúan como obstáculos en la búsqueda que deben ser evitados. Por otra parte, existen restricciones de riesgo social lo cual implica que zonas urbanas y poblados deben ser evitados, así como otros obstáculos artificiales de alto valor. Cuando el problema considera como restricción la altura, el problema se presenta en tres dimensiones.

Para abordar el problema seleccionar la trayectoria, se define un enfoque en términos de encontrar la ruta más corta entre dos puntos aplicado a un análisis espacial GIS [39]. Determinar la trayectoria más corta entre dos puntos es un problema combinatorio típico de teoría de grafos [40]. El costo de la solución del problema PST se define en términos de la distancia total de la trayectoria entre un punto de origen y un destino. Para probar el desempeño del algoritmo de solución al problema, se generaron dos escenarios realistas, uno se encuentra en las localidades de "Los Naranjos" "Joachín" del municipio de Tierra Blanca y el otro en el municipio de Veracruz, ambos escenarios en el estado de Veracruz de Ignacio de la Llave. Además, un escenario de gran dimensión es usado para probar la eficiencia y eficacia de la implementación del método propuesto.

En la vida real, el proceso completo de diseñar un ducto puede demorar meses, dado que implica realizar una investigación exhaustiva sobre gran-

des cantidades de información geográfica, estudios en campo, derechos de vías, políticas públicas, activos de infraestructura existente, hábitats sensibles, etcétera. Contar con un estudio del proceso completo ayuda a determinar la decisión final de invertir o no en la construcción del ducto [41]. Esto motiva a buscar un método para el problema de seleccionar la trayectoria óptima basado en el problema de ruta más corta. No se pretende encontrar una solución exacta sino una aproximación de una ruta adecuada para el problema abordado en este trabajo. Cabe mencionar, que ubicar la mejor trayectoria dentro de un corredor suele ser abordado con métodos para Kth-Shortest Path o Near-Shortest Path lo cual demanda gran cantidad de tiempo y recursos de cómputo. Así mismo, utilizar métodos exactos como Dijkstra siempre resultara la misma trayectoria. Este trabajo de investigación aborda el problema para seleccionar la trayectoria óptima explorando alternativas espaciales dentro de un corredor geográfico a través de la implementación de un algoritmo de recocido simulado que utiliza una estructura de vecindad variable para refinar la ruta y alinearla con el diseño de ingeniería, sin utilizar métodos para calcular K-rutas dentro del corredor. Para esta clase de problemas, no se encontró la aplicación de una estructura de vecindad variable. Sin embargo, las estructuras de búsqueda de vecindario se han aplicado a otros tipos de problemas. En muchos casos, la implementación de una estructura híbrida de vecindario a veces da mejores resultados, pero en otros no [42, 43].

1.3. Objetivos de la investigación

El objetivo general de este trabajo de investigación es contribuir con la implementación de una estructura de vecindad variable a través de un algoritmo de recocido simulado que permita explorar alternativas de rutas a partir de una

ruta inicial. Así mismo, demostrar que el enfoque de optimización propuesto valida encontrar la ruta más corta lo cual beneficiará en trabajos futuros.

Los objetivos particulares de este trabajo de investigación son:

1. Estudiar extensivamente los diferentes métodos de solución propuestos en la literatura.
2. Implementar una metodología para tratar información geográfica en formato ASCII-TXT.
3. Implementar una técnica ávida como algoritmo base para la búsqueda de la solución inicial sobre un grafo ponderado no dirigido.
4. Implementar una estructura de vecindad en el algoritmo de recocido simulado para seleccionar la trayectoria óptima entre dos puntos mediante la exploración de alternativas espaciales dentro de un corredor.
5. Sintonizar los parámetros de control del algoritmo de recocido simulado con la finalidad de mejorar su desempeño tanto en la calidad de la solución como en tiempo de cómputo.
6. Efectuar un estudio comparativo del método de optimización con respecto al algoritmo base sobre instancias realistas en localidades del estado de Veracruz.

1.4. Alcance de la investigación

Este trabajo de investigación se aborda el problema para seleccionar la trayectoria óptima entre dos puntos mediante la exploración de alternativas espaciales dentro de un corredor geográfico a través de la implementación de un algoritmo de recocido simulado con una estructura de vecindad variable.

La función objetivo a minimizar es la distancia de la trayectoria definida en kilómetros. La información geográfica de la topografía y elevación pertenece al estado de Veracruz.

1.5. Contribuciones de la tesis

Las principales contribuciones de este trabajo de investigación son:

- I. Una metodología para tratar información geográfica del área de estudio en formato ASCII-TXT distinta a otros métodos GIS que utilizan información en formato vectorial y ráster.
- II. Diseño y desarrollo de una estructura de vecindad variable con recocido simulado para optimizar el PST.
- III. Los resultados reportados que muestran la eficacia del método propuesto en el proceso de optimización de trayectorias en un corredor.

1.6. Organización de la tesis

Este trabajo de investigación se estructura del modo que se describe a continuación. En el capítulo 2 se presenta un esquema general de diseño de ductos y el problema de seleccionar la trayectoria para ductos de transporte. Así mismo, se presenta la formulación matemática del problema en términos de encontrar la ruta más corta y se discute la complejidad del problema. El capítulo 3 introduce las técnicas de optimización heurística. Así mismo, se presentan los conceptos generales asociados a la técnica ávida y a las metaheurística para resolver problemas de optimización combinatoria. El capítulo 4 presenta la implementación del algoritmo de recocido simulado con una estructura de vecindad variable para resolver el problema de seleccionar la trayectoria ópti-

ma. En el capítulo 5 se describe la evaluación experimental realizada sobre los algoritmos implementados y se discuten los resultados alcanzados. Por último, las conclusiones y las principales líneas de trabajo futuro se presentan en el capítulo 6.

Capítulo 2

Planteamiento del problema

Este capítulo presenta el problema de seleccionar la trayectoria óptima aplicado a ductos de transporte terrestre para líquidos valiosos. En la sección 2.1 se presenta un esquema general de la normatividad del diseño de ductos en México. La sección 2.2 presenta los modelos de datos geográficos y de análisis espacial en GIS. La sección 2.3 describe el problema PST en el marco del trabajo de investigación. La sección 2.4 describe las clases de problemas de la teoría de la complejidad computacional. También, describe de forma general el uso de metaheurísticas para resolver problemas combinatorios. Además, describe de forma general el problema de la ruta más corta (SPP) utilizado como referencia para el desarrollo de la metodología de solución del problema abordado. Por último, la sección 2.5 presenta el problema PST en su formulación mono-objetivo, considerando el costo total de la distancia de la trayectoria como único objetivo a optimizar.

2.1. Diseño de ductos en México

México cuenta con una red de transporte de hidrocarburos a través de PEMEX logística de más de 17,000 km. de ductos [44]. La red de ductos de PEMEX no es suficiente para abastecer a todo el territorio nacional [45]. Una

de las principales desventajas de la red es que hay grandes distancias entre las principales refinerías del país que impide una eficiente distribución de recursos energéticos disponibles [46]. Por tal motivo, los sistemas de transporte de hidrocarburos siguen teniendo un rol importante para la industria petrolera nacional y para el país, por lo que es indispensable que los sistemas de ductos se planeen, se diseñen y se construyan de forma adecuada.

La falta de infraestructura tanto de transporte como de distribución resta flexibilidad operativa y limita la capacidad de abastecer las necesidades del mercado [2]. Por ello es necesario invertir en la ampliación de sistemas de ductos, que en el mejor de los casos, se pueda proveer de infraestructura suficiente con mecanismos de redundancia de respaldo para hacer frente a contingencias diversas, tales como sabotaje y desastres naturales.

México realizó una transformación en el sector energético a partir de la aprobación de la Reforma Energética en 2013. Esto abrió un proceso de transición hacia un modelo abierto y competitivo en dicho mercado, lo cual abre un mundo de retos para el sector. La reforma busca la participación de empresas privadas nacionales e internacionales para invertir y aplicar nuevas tecnologías que permitan ampliar la infraestructura de exploración, explotación y distribución de hidrocarburos.

Por otra parte, con respecto al petróleo crudo y petrolíferos, México registró un total de reservas de 25,858 millones de barriles de petróleo crudo al 1 de enero de 2017, lo que representa un descenso del 1.1 % con respecto al 2016. Sin embargo, el uso de gas natural aumentó progresivamente en el país, principalmente para la generación de energía eléctrica[47], con una demanda de combustibles fósiles del 70 % en 2016 al sustituir el combustóleo y diésel como fuentes primarias. Para abastecer el gas natural, se implementó la estrategia integral de suministro de gas natural mediante la expansión de

la red de transporte por ducto, añadiendo 3,392 kilómetros a la red nacional de gasoductos entre diciembre de 2012 hasta agosto de 2017. Se estima que para el año 2031 la demanda de gas natural aumente 26.8 % respecto a 2016 cubriendo la mayoría de los sectores a nivel nacional [47], lo cual impulsará la expansión de la infraestructura para el transporte mediante la red de ductos con el propósito de aumentar su capacidad y eficiente distribución a través del Sistema de Transporte y Almacenamiento Nacional Integrado de Gas Natural (SISTRANGAS) gestionado por el Centro Nacional de Control del Gas Natural (CENAGAS). En la figura 2.1 [48] se muestra el plan quinquenal de expansión del SISTRANGAS 2015-2019.

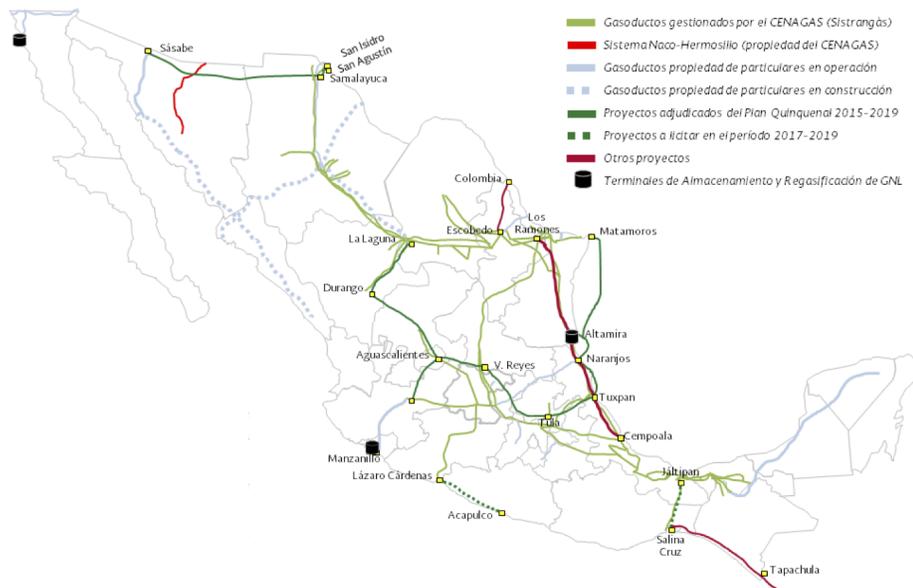


Figura 2.1: SISTRANGAS

En la fase de diseño de ductos, se deben seleccionar los componentes

que conforman el sistema de transporte con base en las condiciones de operación del mismo, así como de las características del fluido a transportar. Las tuberías de transporte pueden ser tuberías de acero para la recolección y transporte de hidrocarburos amargos (petróleo crudo, condensados, gasolinas, gases licuados, amoniaco anhídrido líquido, etcétera) o no amargos (gas natural). Las tuberías que se utilicen en el diseño de ductos terrestres para hidrocarburos amargos deben cumplir con la norma NFR-001-PEMEX-2000, y las tuberías para ductos no amargos deben cumplir la norma de referencia NFR-002-PEMEX-2001 [6]. Así mismo, los accesorios (bridas, conexiones soldables, espárragos, tuercas, empaques, entre otros) a utilizar en el sistema de transporte de hidrocarburos deben satisfacer lo indicado en la norma de referencia NFR-096-PEMEX-2004. Las válvulas deben satisfacer los requisitos de composición química, capacidad mecánica, componentes y calidad especificados en la norma ISO-14313.

Los hidrocarburos fluyen del yacimiento en estado gaseoso, líquido o mixto, lo cual requiere de sistemas de recolección para su transporte hacia un destino donde será procesado. En la figura 2.2 se muestra el enfoque general de recolección de hidrocarburos con base en la norma NRF-030-PEMEX-2009 la cual ha sido utilizada, entre otras normas nacionales e internacionales, para diseñar y construir los activos de infraestructura de Petróleos Mexicanos.

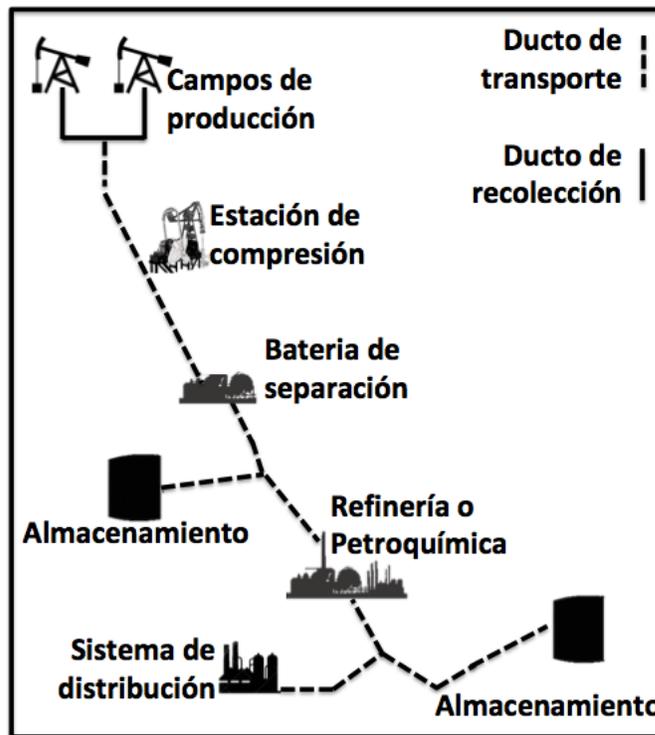


Figura 2.2: Esquema básico de transporte de hidrocarburos

El enfoque general del estándar de referencia muestra que la mezcla desde los campos de producción se transporta a través de una línea colectora hasta una estación de separación, conocida como batería de separación. En la batería de separación se realiza el proceso de separar el gas, el aceite y el agua que generalmente suelen venir juntos. Después del proceso de separación es necesario transportar a los hidrocarburos hasta una planta de procesamiento o almacenamiento para que posteriormente sean enviados hacia los centros de distribución o exportación. El transporte de los hidrocarburos se realiza por distintos medios de transporte, de los cuales los ductos son el medio de transporte más económico. La principal desventaja de los ductos es la distancia que hay entre las localidades físicas de origen y destino sobre un terreno determinado. Seleccionar la trayectoria sobre una superficie de terreno por

donde debe pasar el ducto involucra un proceso de diseño de la trayectoria a partir de datos geográficos.

2.2. Modelos de datos geográficos

La razón de ser de los GIS reside en los datos, por lo que es importante conocer la forma y propiedades de los datos que van a ser utilizados [49], de esto dependen los procesos que se pueden o no realizar sobre ellos, así como saber si se pueden o no integrar datos de distintas fuentes de procedencia y formatos.

El sistema de coordenadas geográficas permite definir la forma de la tierra a partir de un modelo. Existen distintas formas de expresar una posición geográfica a partir de una proyección cartográfica. En este trabajo se utiliza como sistema de referencia el sistema de coordenadas geográficas universal transversal de Mercator (UTM). La escala expresa la relación de una distancia medida en un mapa con la distancia que esta medida representa en la realidad [49].

La información de mapas que representan una porción de superficie terrestre suele encontrarse separada de manera horizontal en bloques y de manera vertical en capas (ver figura 2.3 y 2.4) [49]. Esta separación permite combinar distintos mapas para cubrir extensiones de mayor o menor amplitud (escala) con distintos elementos topográficos separados en capas como son curvas de nivel, carreteras, poblaciones, aeropuertos, cuerpos de agua, etcétera.



Figura 2.3: Integración de datos en sentido horizontal

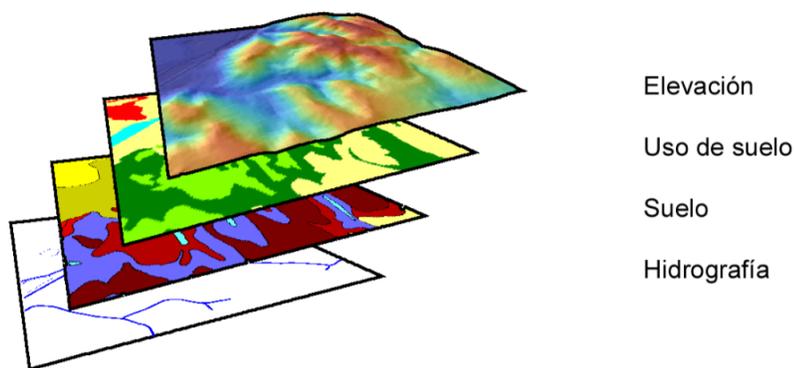


Figura 2.4: Concepto de capa en GIS

Los modelos de representación de los datos de un mapa se pueden clasificar en dos grupos: modelos de representación **ráster** y modelos de representación **vectorial**. En el modelo ráster se realiza una división de forma sistemática de acuerdo a un patrón, en una serie de unidades mínimas denominadas celdas. Cada celda contiene información que representa una propiedad en una posición (ver figura 2.5). Cabe mencionar que la información contenida en una celda puede ser aportada por una única variable o por varias variables, en caso de que exista alguna relación. Una de las ventajas de la estructura regular de una malla ráster es que permite conocer las coordenadas (x, y) de las

celdas sin necesidad de almacenarlas, así como la orientación, habitualmente suele ser norte-sur [49].

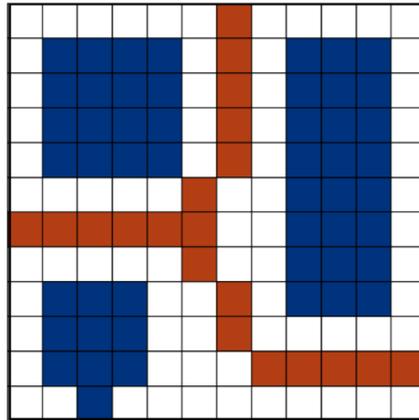


Figura 2.5: Modelo ráster

Por otro lado, en el modelo vectorial (ver figura 2.6) no existe división de la zona en celdas, sino que se cuenta con entidades geométricas cuyas características son constantes. La representación de un mapa se modela mediante una serie de primitivas geométricas que representan propiedades de una superficie terrestre a través de puntos, líneas y polígonos [49].

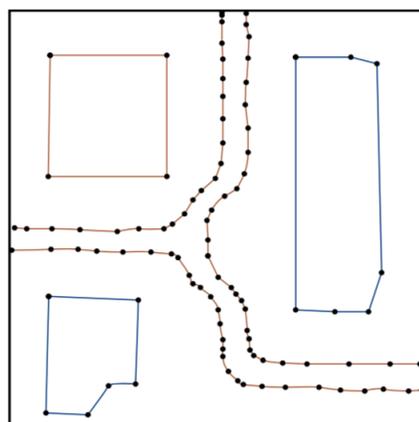


Figura 2.6: Modelos vectorial

La forma de almacenar una capa ráster es simplemente almacenando sus

valores uno a uno en una estructura similar a la de la propia capa topográfica que representa. Es decir, la forma más directa de representar una capa ráster es mediante una matriz de datos.

La ventaja de almacenar los datos en una matriz es que permite tener un formato intuitivo al cual se le pueden aplicar operaciones de cálculo matricial. Así mismo, la implementación en un lenguaje de programación se puede hacer mediante la representación de un arreglo bi-dimensional, permitiendo recorrer la matriz simplificando la implementación de todo tipo de procesos. Una desventaja considerable en la representación matricial ráster es que las celda próxima entre si tienden a tener valores similares, lo cual lleva a tener almacenamiento de datos redundante lo cual requiere de una gran cantidad de espacio de almacenamiento. Para llevar a cabo un análisis espacial es necesaria una capa ráster de **costos acumulados**. La capa de costos acumulados codifica el valor de cada celda con base en algún criterio preestablecido. Dados un punto de origen y un destino, se puede calcular el camino de costos mínimo y obtener la ruta de menor costo [49]. De tal forma que, entre mayor sea la capa o capas , mayor tiempo de ejecución requiere el algoritmo para encontrar la ruta óptima. La figura 2.7 muestra una matriz de costos acumulados creada a partir de otras capas de datos para realizar el análisis espacial con base en los criterios de diseño para determinar la mejor alternativa de la trayectoria, así como una trayectoria encontrada sobre la misma.

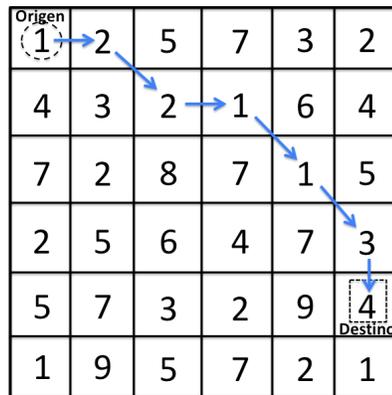


Figura 2.7: Representación de una capa de costos acumulados y una ruta encontrada

Un **corredor** calcula la suma de los costos de una matriz ráster de costos acumulados [50]. En ese contexto, una **trayectoria** es una secuencia de nodos en una red que corresponde a la ruta encontrada entre un nodo origen y uno destino. La trayectoria óptima representa la secuencia de nodos que minimizan el costo de la trayectoria de todas las posibles trayectorias en la red. A continuación se detalla el problema de diseño de trayectorias basado en seis criterios de selección.

2.3. Problema de seleccionar la ruta óptima

El diseño de una trayectoria se puede simplificar en dos componentes espaciales: la ubicación del punto de origen (por ejemplo una pozo, un centro de procesamiento, un refinería) y la ubicación del ducto de transporte hacia los lugares de demanda [51]. Los centros de procesamiento y refinерías a menudo se encuentran ubicados en lugares remotos de un pozo, por lo que el ducto debe cubrir distancias mucho mayores. Un ducto puede atravesar por ciudades, países, regiones, y puede pasar por muchos estados de un país. La construcción de un ducto requiere de permisos de derechos de vía por el

paso de terrenos públicos y privados, bajo ciertas condiciones que hacen que el trámite demore tiempo significativo. Por lo tanto, determinar la trayectoria de un ducto de transporte a través de una superficie determinada se convierte en un desafío. La toma de decisiones para el diseño y su construcción demanda considerar rutas alternas. La mayoría de los corredores tienen una ubicación inicial predefinida y una ubicación final, por lo tanto el objetivo principal es identificar la mejor trayectoria que atraviese una superficie entre dos localidades físicas [11].

En el proceso de seleccionar la trayectoria, hay un componente clave que todos los proyectos de diseño de ductos tiene en común que obedece a la pregunta ¿cómo la trayectoria inicial del ducto afectará posteriormente todas las actividades requeridas para el proyecto? [52]. Así mismo, distintos criterios son utilizados para seleccionar la trayectoria incluyendo las comunidades por donde debe pasar, las necesidades técnicas del proyecto, la construcción del ducto a lo largo de la ruta, adquisición de derechos de vía, problemas ambientales y de zonas arqueológicas, sistemas de mapeo, estudio de campo, entre otros.

Ozane [9] establece que la selección de la ruta debe ser guiada por los siguientes seis objetivos:

1. Establecer la ruta más corta para reducir costos de material y construcción.
2. Minimizar el daño e impacto ambiental.
3. Mantener caminos, autopistas, vías férreas y cruces de caminos de agua al mínimo.
4. Evitar poblaciones y áreas urbanas.

5. Utilizar derechos de vía de tuberías existentes.
6. Proveer acceso a lo largo de la trayectoria del ducto para personal y equipo que permita la correcta operación y funciones de mantenimiento.

De acuerdo a los objetivos mencionados, es de suma importancia para el diseño de un ducto establecer la **ruta más corta** entre el punto de origen hasta el punto destino considerando los objetivos del 2 al 6. Esto se determina con base en los requerimientos técnicos, ambientales y de ingeniería del proyecto. La metodología de solución propuesta para el problema PST se basa en los seis objetivos y se define en términos de encontrar una trayectoria óptima entre dos coordenadas geográfica UTM sobre una superficie determina. La región por donde la trayectoria debe ser ubicada es discretizada para formar la matriz de costos acumulados ráster. La topografía del espacio de la superficie que forma el escenario se encuentra en una estructura de datos en formato vectorial, la cual es transformada a su equivalente en ráster para poder formar los costos acumulados que requiere el análisis espacial bajo ciertas restricciones. La sección 2.5 describe el modelo matemático del problema optimizar la ruta más corta en función de la distancia entre dos puntos.

Este trabajo de investigación se enfoca únicamente en el problema de seleccionar la trayectoria óptima que puede ser aplicada a ductos de transporte de forma genérica sin considerar propiedades de fluidos, con base en los siguientes cuatro procesos:

1. **Identificar metas y objetivos de diseño** - Identificar los factores que serán considerados en el criterio de selección para determinar las mejores alternativas para las trayectorias.
2. **Recopilar datos geográficos relevantes** - Recolectar datos relevantes

que correspondan con los criterios de selección, tales como tipo de vegetación, cuerpos de agua, elevación, poblados, carreteras, entre otros.

3. **Transformar datos geográficos en una red (grafo)** - Clasificar las capas de los datos en costos por su impacto y relevancia en los criterios de selección, discretizar los datos en un ráster con una escala apropiada y conectar las celdas ráster con aristas utilizando la geometría adecuada y asignar costos de viabilidad a las aristas para su análisis en el grafo.
4. **Análisis de la ruta más corta** - Computar un conjunto de rutas alternativas que tengan en cuenta los objetivos de diseño establecidos para que puedan ser consideradas en la toma de decisiones.

El problema de encontrar la ruta más corta a través de una superficie no es tan simple como encontrar una sola alternativa en una red. Cuando los costos de las aristas del grafo tienen diferentes valores, la ruta óptima es diferente de otra encontrada con distintos valores [51]. Además, en la práctica los criterios de selección pueden incluir consideraciones implícitas que no se incluyeron en el análisis, por lo tanto es importante explorar una variedad de soluciones que cubra el rango espacial de soluciones razonables.

Cabe mencionar, que la mayoría de las herramientas GIS trata la información de las capas en formato ráster o vectorial. Sin embargo, existe información de elevación en formato ráster pero no en su formato vectorial y viceversa. Por este motivo, en este trabajo se simplifica el manejo de la información de las capas de datos en formato ASCII-TXT por su facilidad de uso en lenguajes de programación de alto nivel. La conversión de los datos ráster y vectorial a su equivalente en ASCII-TXT se realiza a través de la herramienta GDAL/OGR [53].

Con respecto a las técnicas metaheurísticas, existe sólo una biblioteca de

análisis espacial de software libre llamada SEXTANTE que implementa algoritmos genéticos para realizar análisis geo-espacial. La herramienta SEXTANTE es una extensión de otra herramienta de software libre llamada gvSIG. La mayoría de las herramientas GIS no implementa metaheurísticas para optimizar debido a su alto costo computacional. Dada la necesidad de contar con herramientas para realizar análisis espacial y geo-espacial sobre grandes cantidades de datos geográficos se propone en este trabajo de investigación la implementación de una técnica de optimización estocástica a través de un enfoque GIS.

Por otra parte, para saber qué tipo de técnica se debe emplear en el análisis espacial para explorar el mayor número de alternativas de soluciones que hay en el problema es necesario conocer su complejidad a través de la teoría de la complejidad.

2.4. Complejidad del problema

Clases de problemas. El análisis de algoritmos *a priori* permite expresar la complejidad de un algoritmo, independientemente de la arquitectura del hardware o del lenguaje de programación, a través de la notación denominada “O” (big-O, en inglés). El cálculo de la complejidad de un algoritmo corresponde al número de pasos requeridos por el algoritmo para resolver un problema [54]. Por lo tanto, a mayor complejidad, mayor será el tiempo que requiere el algoritmo para obtener una solución a un problema.

La complejidad temporal [55] $T(n)$ permite expresar matemáticamente la relación entre la cantidad de datos de entrada y el tiempo de ejecución del algoritmo. Un algoritmo eficiente es simple y hace un uso adecuado de los recursos de cómputo [56]. Para medir el tiempo de ejecución $T(n)$ se conside-

ran las siguientes operaciones elementales: operación aritmética, asignación a una variable, llamada a una función, retorno de una función, comparaciones lógicas y acceso a una estructura de datos. Por lo tanto, el tiempo de ejecución no es el tiempo físico, sino el número de operaciones elementales que realiza el algoritmo [57].

La notación big-O permite identificar el orden de complejidad de un algoritmo. Un algoritmo es más eficiente con respecto de otro mientras menor sea su orden de complejidad para resolver el mismo problema. Existen distintos tipos de complejidad. Los tipos de complejidad más comunes se listan a continuación [57]:

1. $O(1)$ **Complejidad constante** se presenta en la ejecución de una operación elemental sin repeticiones (ciclos)
2. $O(\log n)$ **Complejidad logarítmica** se presenta en algoritmos iterativos y recursivos
3. $O(n)$ **Complejidad lineal** se presenta en un ciclo, cuando las operaciones al interior son constantes
4. $O(n \log n)$ **Complejidad $n \log n$** se presenta en algunos algoritmos de ordenamiento
5. $O(n^2)$ **Complejidad cuadrática** se presenta en ciclos anidados o en recursiones dobles
6. $O(n^3)$ **Complejidad cúbica** se presenta cuando hay tres ciclos anidados o en recursiones triples
7. $O(n^k)$ **Complejidad polinomial** se presenta para $k \in \mathbb{N} : k > 3$. Entre mayor sea el valor de k , menos eficiente es el algoritmo

8. $O(k^n)$ **Complejidad exponencial**, cuando n es muy grande, el algoritmo puede ser excesivamente costoso. Tal es el caso de algunos problemas combinatorios.

De acuerdo a la clasificación, el algoritmo más eficiente es aquel que presenta una complejidad constante. El comportamiento de un algoritmo con complejidad constante se manifiesta en que a pesar de que el tamaño de la entrada de datos aumente, el tiempo de ejecución del algoritmo se mantiene constante. En el caso de la complejidad logarítmica, si el tamaño de la entrada se incrementa 100 veces, el tiempo de ejecución del algoritmo únicamente se duplica, por lo que se considera eficiente. Por el contrario, si un algoritmo tiene complejidad exponencial, este se considera ineficiente debido a que su tiempo de ejecución aumenta exponencialmente conforme se incrementa el tamaño de la entrada [56]. La figura 2.8 muestra gráficamente el comportamiento de las funciones de la complejidad algorítmica.

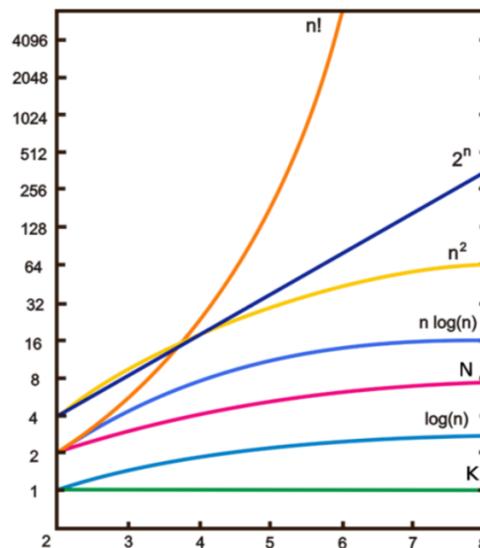


Figura 2.8: Comportamiento de las funciones de complejidad algorítmica

Algunos algoritmos conocidos y sus respectivas complejidades se listan a

continuación[58]:

1. Buscar un elemento en una lista no ordenada tiene una complejidad de $O(n)$
2. Buscar un elemento en una lista ordenada tiene una complejidad de $O(\log n)$
3. Quicksort tiene una complejidad de $O(n \log n)$
4. Calcular el determinante de una matriz tiene una complejidad de $O(n^3)$
5. Ordenamiento por el método de la burbuja (Bubble Sort) tiene una complejidad de $O(n^2)$

La teoría de la complejidad estudia y clasifica los problemas de acuerdo a su complejidad de resolución [59]. Los problemas cuya complejidad está acotada por un polinomio, que se encuentre dentro de los primeros seis órdenes de magnitud de la jerarquía de complejidad listada anteriormente, son los denominados problemas **P**. [60] Un problema pertenece a la clase **P** si puede ser resuelto en tiempo polinomial en una computadora determinista. [58]

El término **determinista** significa el paso siguiente que realizará el algoritmo es determinado por los pasos anteriores. Ejemplos de problemas que pertenecen a esta clase son: Ordenamiento, búsqueda binaria, operaciones matriciales.

Un problema que pertenece a la clase **NP** puede ser resuelto en tiempo polinomial pero usando una computadora **no determinista**. Una computadora no determinista puede “adivinar” la opción correcta de entre varias opciones, siempre y cuando ésta exista. Además, una máquina no determinista no hace elecciones incorrectas que le hagan regresar a un estado previo. Las computadoras no deterministas no existen en el mundo real. El no determinismo es

una abstracción mental que hace que los problemas difíciles de resolver parezcan triviales.

La máquina no determinista se emula a través de un algoritmo. Es posible convertir un algoritmo no determinista a uno determinista, a un costo computacional muy elevado [58]. Para comparar la clase P de la clase NP, la clase P contiene problemas que pueden resolverse rápidamente, mientras que la clase NP contiene problemas cuya solución puede verificarse fácilmente, pero es difícil hallarla. Sin embargo, existe otra clase de problemas denominada **NP-Completos**. Un problema pertenece a la clase NP-Completo si todos los algoritmos requeridos para su resolución requieren tiempo exponencial en el peor de los casos. Este tipo de problemas son muy difíciles de resolver en tiempo razonable. En esta clase de problemas difíciles de resolver se encuentran el problema del agente viajero (TSP, por sus siglas en inglés), el problema de satisfactibilidad booleana (también llamado SAT), los ciclos hamiltonianos en grafos, entre otros problemas.

Problema de la ruta más corta y de menor costo. Existe una distinción entre los problemas de la ruta más corta y la de menor costo. El problema de la ruta más corta es un caso especial del problema de la ruta de menor costo, en donde la penalización de atravesar una región se compone únicamente de la distancia recorrida en cada paso a través de la región. Cuando una red representa un espacio del mundo real, las penalizaciones pueden surgir de muchos otros factores. Por ejemplo, en una red que representa una carretera no sólo es de interés la ruta más corta de un punto a otro, también puede ser de interés el tiempo de recorrer la ruta, por lo tanto, factores como la velocidad promedio del tráfico, el promedio de la velocidad de cambio de las luces de los semáforos, el número de vueltas a la derecha y a la izquierda, entre otros fac-

tores que pueden tener efectos en el resultado para obtener la ruta de menor costo en el tiempo de recorrido de un punto a otro en comparación con la ruta más corta.

Al determinar la ruta de menor costo para la construcción de ductos sobre un tipo de terreno o suelo, puede existir factores que impacten en los costos de construcción sobre un determina terreno. Factores como la pendiente, tipo de suelo, tipo de vegetación, uso de suelo, áreas protegidas, cuerpos de agua, y factores de mantenimiento como poder acceder a carreteras principales, entre otros que impactan en la toma de decisiones para determinar la trayectoria del ducto [51]. Para resolver problemas de la ruta más corta y de menor costo se utilizan los mismos algoritmos, sólo hay que considerar las restricciones espaciales a optimizar.

El problema del camino más corto (SPP, por sus siglas en inglés) busca el camino de distancia más corta o de menor costo sobre un enorme número de alternativas. Los métodos matemáticos para encontrar la ruta de menor costo en una red aparecen a principios de los años 50s, a diferencia de otros problemas similares como el problema del agente viajero (TSP) y el problema del árbol de expansión mínima (MST, por sus siglas en inglés) [51]. En el problema TSP, a partir de un conjunto de puntos y sus respectivas distancias entre pares, se debe encontrar la ruta más corta posible que visite cada punto exactamente un vez y que finalice en el punto de inicio. Karl Menger [61] definió matemáticamente el TSP abordando el problema a través de un enfoque de fuerza bruta, y observó la no optimalidad de la heurística del vecino más cercano. En 1972 Richard M. Karp [62] demostró que un ciclo hamiltoniano es un problema NP-Completo, por lo que el problema TSP se considera un problema NP-Duro.

En general, un problema NP-Completo es un problema de decisión, es

decir, debe decidir si algo es verdadero o falso. El problema MST es similar al TSP, en el sentido de que todos los puntos deben ser alcanzados en la menor distancia posible a través de una red que conecta una serie de puntos. Los métodos Kruskal[63] y Prim [64] son los más utilizados en nuestros días para abordar problemas relacionados con el MST. Estos desarrollos sentaron las bases de la teoría y métodos para resolver problemas de optimización en redes, incluido el problema de la ruta más corta.

El problema de encontrar un camino simple con el menor número de nodos (vértices) en una red, es uno de los problemas más importantes en la teoría de grafos. El análisis de costos a través de una red, implica encontrar la ruta de menor costo entre dos nodos de la red. El problema de encontrar la ruta de menor costo tiene numerosas aplicaciones incluso dentro del ámbito geográfico, a través de modelos de datos ráster, vectorial o ambos. El análisis de costos permite modelar los costos a través de bordes (aristas) que conectan a los nodos (vértices), permitiendo establecer restricciones de desplazamiento por los bordes, en una dirección o en ambas direcciones. La representación de los modelos de datos así como los casos particulares de análisis espacial son distintos[49].

La implementación de algoritmos de cálculo de rutas de menor costo es en cierta medida similar, dado que los algoritmos trabajan sobre estructuras denominadas grafos, utilizados para representar una red. El algoritmo más popular dentro de la teoría de grafos para calcular rutas óptimas es el propuesto por Edsger Dijkstra [15]. La forma más directa y simple de resolver este tipo de problemas es la de probar todas las posibles combinaciones de rutas y elegir la menor costo. Sin embargo, dado el crecimiento exponencial del número de posibilidades, a medida que se incrementa el tamaño del problema, no es viable abordar este tipo de problemas con técnicas exactas, sino con métodos

de aproximación, heurísticas o metaheurísticas.

Un camino en un grafo es una secuencia de aristas que conectan una secuencia de vértices y un ciclo es el camino tal que el vértice inicial y final son los mismo. En un camino no aparece ningún vértice dos veces y en un circuito el vértice inicial y final aparecen dos veces. Un grafo que contiene un ciclo hamiltoniano es llamado grafo hamiltoniano. Una descomposición hamiltoniana es una descomposición de bordes (arcos) de un grafo en circuitos hamiltonianos[65]. Por lo tanto, cada ciclo del grafo es hamiltoniano, y cada ciclo hamiltoniano puede ser convertido en un camino removiendo uno de sus arcos pero un camino hamiltoniano puede ser extendido a un ciclo hamiltoniano sólo si sus dos puntos finales (origen y destino) son adyacentes[65].

El problema del camino hamiltoniano es un problema NP-Completo [66] [59]. Decidir si existe un camino que visite un sólo vértice en un grafo sólo una vez, es un caso especial del problema de la ruta más corta. Un grafo de malla rectangular $R(n, m)$ es un subgrafo de G_∞ (grafo infinito) inducido por $V(m, n) = v : 1 \leq v_x \leq m, 1 \leq v_y \leq n$, donde v_x y v_y son las coordenadas x, y de v (ver figura 2.9 [67]. $R(n, m)$ tiene un ciclo hamiltoniano si y solo si es de tamaño uniforme y $m, n > 1$ [68].

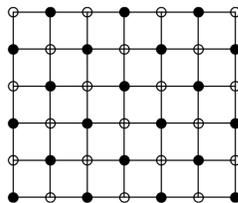


Figura 2.9: Representación de un grafo rectangular

La figura 2.10 muestra un ciclo hamiltoniano para el grafo de malla rectangular de la figura 2.9 de tamaño uniforme.

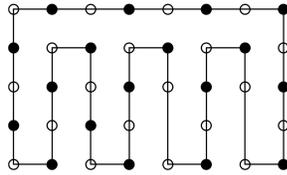


Figura 2.10: Representación de un ciclo hamiltoniano en el grafo rectangular

El problema de encontrar el camino más largo entre dos nodos en un grafo es un problema NP-Duro [69] [59]. El problema de encontrar el camino más corto entre dos nodos en un grafo es un problema que pertenece a la clase P [59]. Sin embargo, el problema de encontrar el camino más corto en una red que involucra n capas de datos de información geográfica puede ser un problema NP. Este trabajo de investigación no se centra en comprobar si el problema de encontrar el camino más corto en una red que involucra n capas de datos es NP, sin embargo queda la pregunta abierta. Comprobar a qué clase de problemas pertenece el problema planteado está fuera del alcance de este trabajo y queda como trabajo futuro.

Los problemas complejos que requieren de explorar el mayor número de permutaciones o combinaciones se abordan con técnicas de optimización combinatoria.

Optimización combinatoria La optimización combinatoria es una parte de la Programación Matemática, en la que se estudian problemas del tipo $\min\{f(x) : x \in S\}$, donde S es el conjunto de soluciones y $S < \infty$. En optimización combinatoria se desarrollan algoritmos para resolver problemas caracterizados por tener un número finito de soluciones factibles. Un algoritmo efectivo permite determinar alguna solución al problema en tiempo razonable, donde razonable es el tiempo que se está dispuesto a esperar [70].

El tiempo y consumo de recursos computacionales aumenta de manera

exponencial al tratar problemas combinatorios NP-Completo con métodos exactos [71]. Cuando la aplicación de métodos de búsqueda exhaustiva o enumerativa y las técnicas de programación matemática no son aplicables para la resolución de un problema, las técnicas heurísticas aparecen como la única alternativa viable para abordar problemas NP-Completo con espacios de soluciones de dimensión alta. A diferencia de las técnicas exactas, las técnicas heurísticas no pueden garantizar *a priori* la obtención de la solución óptima de un problema. En muchos casos, tampoco es posible garantizar que la solución obtenida tenga un cierto margen de error respecto de la solución óptima. No obstante, en la práctica, numerosas técnicas heurísticas llegan a tener un comportamiento mucho más eficiente que las técnicas exactas en la resolución de problemas complejos de optimización combinatoria. Las técnicas heurísticas obtienen buenas soluciones aproximadas en tiempos razonables [71]. Por otra parte, las técnicas metaheurísticas proporcionan un enfoque genérico para la resolución de problemas complejos a partir de heurísticas subordinadas. En el capítulo 3 se describe con mayor detalle a las técnicas metaheurística. En el capítulo 4 se describe la implementación de la técnica metaheurística utilizada para abordar el problema tratado en este trabajo de investigación.

2.5. Formulación matemática

El problema de localizar la trayectoria del corredor de un ducto de transporte es un problema de optimización, el cual consiste en determinar la ruta óptima entre dos localidades físicas en una red. El problema PST se formula como una variante del problema del camino más corto entre dos nodos en un grafo que representa un escenario GIS.

En este trabajo, la red es representada por el grafo no dirigido $G = (N, A)$,

donde $N = u_1, u_2, \dots, u_n$ es el conjunto de los nodos del grafo y $A = (u_1, v_1), (u_2, v_2), \dots, (u_m, v_m)$ el conjunto de aristas, y $n = |N|$ y $m = |A|$; cada arista $(u, v) \in A$ está asociada a un costo real positivo c_{uv} .

En el problema del mundo real que se aborda, un vértice representa el centro de una celda ráster del mapa y una arista el costo de viabilidad de pasar entre dos vértices de la red.

Los costos de viabilidad se determinan a partir de los valores de la escala $0 : 10$, donde el valor 10 representa un costo de viabilidad muy alto, obligando a la búsqueda de otras alternativas de menor costo y el valor 0 representa un costo muy bajo por lo que no existe restricción alguna de paso. Para formar la matriz de costos acumulados a partir de n capas de datos, a cada capa de datos se le asigna un valor dentro de la escala de costos de viabilidad según su relevancia en el diseño de la trayectoria. Una vez definido el valor de cada capa de datos, se suman y se ponderan los valores de las distintas capas que coincidan dentro de la misma celda (i, j) en la matriz de costos acumulados. La ponderación de los valores determina el costo de viabilidad de paso entre dos puntos del mapa.

El costo del recorrido no necesariamente debe ser simétrico. El objetivo del problema de la ruta de menor costo de una origen a un destino es encontrar la trayectoria de costo mínimo desde un nodo origen $s \in N$ hasta un nodo destino $t \in N$. Cada arista tiene asociada una variable de decisión binaria x_{uv} que es igual a 1 si se encuentra en la ruta más corta, y 0 en otro caso. Así mismo, cada arista tiene asociada otra variable de decisión binaria p_u que es igual a 1 si el paso por el nodo u esta permitido y 0 si está prohibido. La formulación matemática del problema de encontrar el camino de menor costo se describe en la ecuación 2.1.

$$\begin{aligned}
 \text{mín } z(x) &= \sum_{(u,v) \in A} c_{uv} \cdot x_{uv} \cdot p_u \\
 \text{s.a. } \sum_{(u,v) \in A} x_{uv} - \sum_{(v,u) \in A} x_{vu} &= \begin{cases} 1 & \text{if } u = s \\ -1 & \text{if } u = t, \\ 0 & \text{if } u \neq s, t. \end{cases} \\
 x_{u,v} &= \{0, 1\} \forall (u, v) \in A \\
 p_u &= \{0, 1\} \forall u_i \in N
 \end{aligned} \tag{2.1}$$

La trayectoria entre los nodos $s = 0$ y $t = k$ es una secuencia finita de nodos $p = \{u_0, \dots, u_k\}$ tal que para calcular el costo de la distancia en kilómetros se considera la escala del mapa y se determina por la ecuación 2.2. La escala del mapa utilizada en este trabajo es de 1:50,000.

$$\text{distancia}(s, t) = \frac{p \cdot 50}{1000} \tag{2.2}$$

Capítulo 3

Metaheurísticas

El presente capítulo describe la técnica de búsqueda utilizada para resolver el problema de optimización de trayectorias para corredores aplicados a ductos de transporte con un enfoque mono-objetivo. En la sección 3.1 se introducen los métodos de búsqueda heurísticas como técnicas de búsqueda local para resolver problemas combinatorios. En la sección 3.1.1 se introducen los métodos constructivos o ávidos (término *greedy* en inglés) los cuales son usados para construir la solución inicial a partir de información disponible de forma inmediata. En la sección 3.1.2 se describen las metaheurísticas como técnicas para resolver problemas orientados a mejorar el desempeño y calidad de búsqueda proporcionados por las heurísticas. En la sección 3.1.2.1 se describen los tipos de metaheurísticas reportados en la literatura. Por último, la sección 3.2 presenta la metaheurística Recocido Simulado utilizada para optimizar la solución obtenida por el algoritmo greedy.

3.1. Métodos de búsqueda

Un método de búsqueda es un procedimiento sistemático para hallar una solución a un determinado problema. Cuando existen múltiples soluciones candidatas, el procedimiento implica explorar el conjunto de posibles solucio-

nes con el fin de identificar aquella(s) que proporcione(n) las mejores funcionalidades para resolver el problema en cuestión. Formalmente, el problema se puede formular mediante uno o varios objetivos, que son tomados en cuenta para evaluar las funcionalidades provistas por cada solución candidata, con el objetivo de hallar una que resuelva apropiadamente el problema. Un mecanismo efectivo para resolver un determinado problema consiste en explorar todas y cada una de las soluciones candidatas. Se implementa de este modo una búsqueda exhaustiva del espacio de soluciones del problema. La búsqueda parte de un estado inicial caracterizado por una solución candidata y define un procedimiento sistemático para analizar y comparar todas las posibles soluciones. El procedimiento exhaustivo garantiza encontrar la solución de un problema (en caso de que exista) o determinar la inexistencia de una solución.

Sin embargo, este tipo de métodos tienen un serio inconveniente que limita su aplicación en la práctica en problemas realistas: el número de pasos requeridos y por lo tanto también el costo computacional, en términos de tiempo de cómputo, para realizar la exploración completa del espacio de soluciones que puede ser muy grande, en especial al considerar problemas complejos y/o escenarios realistas de gran dimensión.

Para afrontar el problema del elevado costo computacional de los métodos de búsqueda exhaustiva se han propuesto métodos alternativos basados en realizar una exploración sistemática y más eficiente del espacio de soluciones. Los métodos denominados de búsqueda local se basan en un procedimiento que parte de una solución candidata inicial y que aplica una serie de pasos en los que iterativamente se explora un conjunto de soluciones vecinas, que difieren poco de la solución candidata que se considera en cada paso.

Formalmente, a partir de una solución s , se define una estructura de vecindad $N(s)$ como el conjunto que incluye todas las soluciones que pueden

alcanzarse desde s mediante modificaciones simples o movimientos. Al explorar la vecindad de una solución s se compara cada elemento de la vecindad con s y se determina la solución que mejor resuelve el problema en cuestión, que pasa a ser la solución candidata a considerar en el siguiente paso del método.

El criterio utilizado para construir la vecindad determina las propiedades del método de búsqueda local. Es claro que un criterio demasiado exhaustivo puede conducir a una búsqueda poco eficiente computacionalmente, que se comportará como una búsqueda exhaustiva. En este contexto, los métodos heurísticos proporcionan criterios eficaces para definir búsquedas eficientes. La idea consiste en seleccionar para la comparación solamente aquellas soluciones promisorias, de la misma manera en que un experto aplicaría su intuición y experiencia, restringiendo la amplitud de la búsqueda para resolver un problema determinado. El procedimiento resultante es eficiente y efectivo para calcular soluciones de buena calidad, aunque no garantiza encontrar la solución óptima del problema [72]. Reeves [73] describe la técnica heurística considerándola como una técnica de búsqueda y optimización como “una técnica que busca soluciones buenas, a un costo computacional razonable, aunque sin garantizar factibilidad u optimalidad de las mismas”. En algunos casos, ni siquiera se puede determinar qué tan cerca del óptimo se encuentra una solución factible.

3.1.1. Búsqueda Local Iterada

Considérese que se requiere encontrar la mejor solución de un problema, cuya calidad se puede evaluar mediante una función $f(x)$, siendo x un conjunto de variables que definen o caracterizan a la solución s . En lugar de explorar todas las soluciones del espacio de búsqueda, es posible aplicar un procedi-

miento sistemático como el definido por el método de búsqueda local iterada (BLI) [74].

Como se explicó previamente, el método BLI parte de una solución inicial, que puede ser determinada aleatoriamente o seleccionada tomando en cuenta conocimiento específico del problema. A partir de una solución candidata, BLI aplica una búsqueda iterativa que consiste en reemplazar la solución actual por una mejor solución existente en su vecindad $N(s)$, evaluada aplicando la función $f(x)$. De este modo, la calidad de la solución mejora en cada iteración y el método es capaz de proporcionar una solución de alta calidad, inclusive puede alcanzar la solución óptima del problema.

Dado que el método no necesariamente explora exhaustivamente todo el espacio de soluciones del problema, su desempeño (evaluado en uso de recursos computacionales y tiempo de ejecución) es muy superior a una búsqueda exhaustiva. Por este motivo, BLI constituye una alternativa eficiente para la resolución de problemas. El procedimiento definido por el método BLI permite resolver exitosamente problemas con un único óptimo global, ya que la búsqueda iterada permite ‘subir la colina’ del óptimo y retornar esa solución. Sin embargo, el método no garantiza encontrar la mejor solución para problemas con óptimos locales, ya que la búsqueda iterada puede conducir a escalar la colina incorrecta (la del óptimo local) en lugar de la colina del óptimo global. Este comportamiento se ejemplifica en el diagrama presentado en la Figura 3.1 el método BLI permite resolver problemas con un único óptimo (problemas unimodales) pero puede fallar para problemas con más de un óptimo local (problemas multimodales).

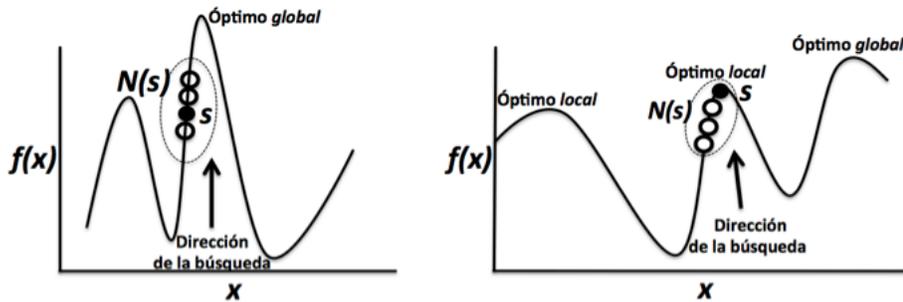


Figura 3.1: Comportamiento del método BLI en la resolución de problemas: (izq.) función unimodal, el método es capaz de encontrar la solución óptima del problema, (der.) función multimodal, el método puede fallar en encontrar el óptimo global y retornar un óptimo local del problema

3.1.2. Métodos ávidos

Un algoritmo ávido (o greedy) es un método constructivo utilizado principalmente para resolver problemas de optimización. Esta técnica se basa en tomar decisiones a partir de información disponible de forma inmediata, sin considerar los efectos que pudieran tener en el futuro [57]. El método greedy construye soluciones a partir de tomar decisiones localmente óptimas en cada paso, en busca de una solución global del problema [75]. Los algoritmos voraces buscan optimizar una función objetivo que permite medir la calidad de la solución obtenida. Los problemas que son resueltos con métodos greedy incluyen encontrar el mejor orden para ejecutar un conjunto de tareas en una computadora, encontrar la ruta más corta en un grafo, entre otros problemas [76].

La forma en que un algoritmo greedy construye una solución paso a paso es la siguiente [77]:

- Inicialmente el conjunto de candidatos elegidos esta vacío
- Luego, en cada paso se agregan al conjunto los mejores candidatos, a través de una función de selección

- La función de selección depende del problema que se está tratando
- Si el conjunto de candidatos deja de ser factible, se elimina el candidato que se acaba de agregar; el candidato eliminado no será considerado nuevamente
- Si el conjunto de candidatos aún es factible, el candidato que se acaba de agregar permanecerá en el conjunto de candidatos seleccionados en ese momento
- Cada vez que se amplía el conjunto de candidatos elegidos, se verifica si el conjunto constituye una solución al problema

Los algoritmos greedy construyen la solución en etapas sucesivas, considerando la mejor decisión en cada etapa. Cuando el algoritmo greedy funciona correctamente, la primera solución encontrada siempre es óptima. La función de selección suele estar basada en la función objetivo, incluso puede ser la misma. Las técnicas greedy son muy populares y simples de diseñar. En general, los métodos greedy tienen una complejidad menor que los métodos iterativos [76].

3.1.3. Metaheurísticas

La palabra metaheurística se compone de dos términos griegos, “meta” que significa más allá y “heurística”, que significa encontrar o descubrir. La palabra se utiliza para designar a procedimientos de resolución de problemas orientados a mejorar el desempeño y calidad de búsqueda proporcionados por las heurísticas. La primera mención al término ‘metaheurística’ fue realizada por Glover [78] en el trabajo que definió las bases de este tipo de técnicas.

Las metaheurísticas se definen como procedimientos de búsqueda de alto nivel que aplican una (o varias) reglas heurísticas, que permite explorar el

espacio de búsqueda de manera más eficiente que una búsqueda exhaustiva (en términos de tiempo y recursos computacionales requeridos) y de manera más eficaz que heurísticas simples (en términos de la calidad de las soluciones calculadas). Para mejorar el procedimiento de búsqueda de los métodos heurísticos, las metaheurísticas proponen combinar diversos esquemas provistos por una serie de heurísticas subordinadas. De este modo, una metaheurística combina una o varias heurísticas con un procedimiento general de alto nivel que determina cómo coordinar la ejecución de las heurísticas subordinadas y proporciona mejoras al patrón de búsqueda.

Dos ejemplos relevantes de metaheurísticas que combinan una heurística de búsqueda local con un procedimiento de alto nivel para mejorar las capacidades de calcular buenas soluciones son Recocido Simulado [79] (RS) y Búsqueda Tabú [78] (BT). RS se basa en potenciar a un método de búsqueda local iterada con un procedimiento que permite aceptar soluciones de peor calidad que la solución actual, con el propósito de escapar de óptimos locales. BT propone combinar una búsqueda local iterada con una memoria de corto plazo que guía la búsqueda; las soluciones examinadas en el pasado se incluyen en una lista tabú, prohibiendo que sean visitadas en los siguientes pasos de la iteración para evitar ciclos y escapar de óptimos locales.

Los dos casos presentados previamente ejemplifican claramente los objetivos concretos de las metaheurísticas: encontrar de manera eficiente soluciones de buena calidad y analizar el espacio de búsqueda, evitando quedar atrapadas en zonas específicas [80]. Dos conceptos relevantes se asocian con los objetivos mencionados: la exploración (o diversificación) del espacio de búsqueda y la explotación (o intensificación) de buenas soluciones encontradas previamente. La Figura 2 ilustra estos conceptos, presentando los patrones exploratorios y de explotación que deben formar parte de la propuesta

conceptual de toda metaheurística.

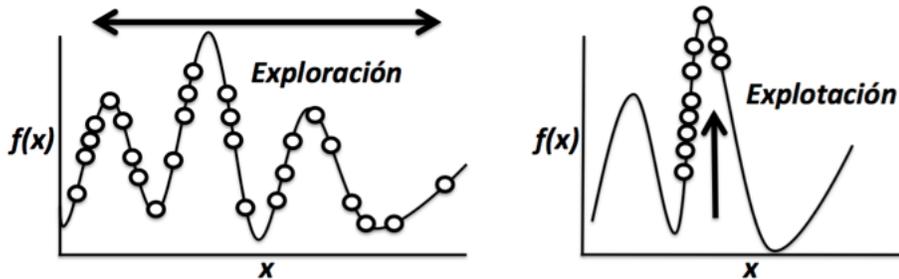


Figura 3.2: Conceptos vinculados con el patrón de búsqueda de una metaheurística: (izq.) exploración del espacio de soluciones y (der.) explotación de buenas soluciones encontradas

Los procedimientos que definen a las metaheurísticas suelen basarse en la emulación de conceptos y procesos bien conocidos y de validez intuitiva. De este modo se han propuesto metaheurísticas basadas en procesos naturales como el enfriamiento de los metales (Recocido Simulado), la evolución natural (Algoritmos Evolutivos), el comportamiento social de insectos y aves (Optimización por Colonia de Hormigas y Optimización por Enjambre de Partículas) y otros fenómenos.

3.1.3.1. Tipos de Metaheurísticas

Existen diversos criterios para clasificar a las técnicas metaheurísticas. Uno de los criterios más utilizados por la comunidad científica toma en cuenta el número de soluciones candidatas exploradas en cada paso de iteración. De acuerdo a este criterio, las metaheurísticas se clasifican en basadas en trayectoria, que manejan una única solución en cada paso de iteración, y basadas en población, que manejan un conjunto de soluciones candidatas en cada paso [76]. Las metaheurísticas basadas en trayectoria reemplazan la solución candidata en cada paso, definiendo de esta manera una trayectoria en

el espacio de búsqueda. Ejemplos conocidos de esta categoría son la Búsqueda Tabú, Recocido Simulado, Búsqueda Ávida Aleatoria Adaptativa (GRASP por sus siglas en inglés), entre otras. Estos métodos son eficientes, al manejar solamente una solución en cada paso, pero pueden tener problemas para escapar de óptimos locales fuertemente atractores.

Las metaheurísticas basadas en población utilizan un conjunto de soluciones candidatas en cada paso (la población). Un número determinado de soluciones se reemplazan en cada iteración por nuevas soluciones que se pueden construir por recombinación de otras soluciones en la población o por recombinación es un mecanismo exclusivo de las metaheurísticas basadas en población y consiste en construir nuevas soluciones a partir de la combinación de características de soluciones seleccionadas de acuerdo a su capacidad de resolver el problema en cuestión. La mayoría de las metaheurísticas basadas en población son bio-inspiradas. Los mecanismos para seleccionar y recombinar soluciones se basan en conceptos biológicos como la evolución natural, la sinergia e inteligencia colectiva de grupos de animales (hormigas, aves, abejas, murciélagos, etc.) u otros fenómenos similares. Ejemplos conocidos de esta categoría de metaheurísticas son Algoritmos Evolutivos, Optimización por Colonia de Hormigas, Optimización por Enjambre de Partículas, entre otras.

Los Algoritmos Evolutivos engloban a un conjunto de metaheurísticas muy populares y difundidas en los últimos treinta años, por su versatilidad para resolver un gran número de problemas. Los Algoritmos Evolutivos basan su funcionamiento en una emulación del procesos de evolución natural de los seres vivos, aplicando los conceptos neo-darwinistas de selección natural, supervivencia de los individuos más aptos y diversidad genética para resolver problemas de búsqueda, optimización y aprendizaje [81]. Dentro de la categoría de los Algoritmos Evolutivos se engloban técnicas como los Algoritmos

Genéticos, la Programación Genética, las Estrategias de Evolución, la Evolución Diferencial y otras. El esquema de un Algoritmo Evolutivo (AE) se describe en Algoritmo 1. El AE aplica una búsqueda iterativa (cada iteración se denomina generación) sobre un conjunto de individuos (la población P). Cada individuo en la población representa una solución tentativa al problema a resolver. La calidad de cada solución representada se evalúa mediante una función de aptitud que determina qué tan adecuada es la solución para resolver el problema, de acuerdo a la(s) función(es) objetivo considerada(s).

Inicialmente la población se genera de forma aleatoria o aplicando una heurística específica (y simple) para resolver el problema. En cada generación, el AE aplica probabilísticamente operadores de variación, como la recombinación de dos individuos y la aplicación de cambios aleatorios (mutación) en su contenido. La utilización de una técnica de selección de soluciones que emula a la selección natural, dando mayor posibilidad de supervivencia a los individuos más aptos (de acuerdo a sus valores de aptitud), conduce a la población del AE a soluciones de mejor calidad para el problema.

Algoritmo 1: Esquema de un AE

Inicialización de la población P ;

mientras *no se cumpla el criterio de parada* **hacer**

 Evaluación de función de aptitud de P ;

 Selección de los individuos más aptos;

 Recombinación de individuo;

 Mutación de individuos;

 Reemplazo de individuos, generando la nueva población;

fin

Retorno del mejor individuo encontrado;

El criterio de parada del AE usualmente involucra un número determinado de generaciones o un tiempo límite de ejecución, una cota de calidad en los valores de aptitud, o la detección de una condición de convergencia. Estrategias específicas se utilizan para seleccionar los individuos a recombinar (el operador de selección) y para determinar qué individuos se insertan en la población luego de aplicar los operadores evolutivos (el operador de reemplazo). Finalmente, el AE retorna el mejor individuo (solución) encontrado en el proceso, tomando en cuenta la función de aptitud considerada. El mecanismo de búsqueda definido por un AE presenta un comportamiento robusto, tomando en cuenta la información presente en la población en cada generación y su capacidad de muestrear apropiadamente el espacio de búsqueda. Las múltiples variantes de operadores de recombinación y mutación existentes permiten definir diferentes balances entre la exploración y explotación del AE. Complementariamente, la capacidad de incluir información concreta sobre el problema a resolver, a través de representaciones u operadores específicamente diseñados proporcionan a los AE una gran versatilidad para la resolución de una amplia gama de problemas.

Las metaheurísticas han sido aplicadas exitosamente en múltiples áreas de aplicación en problemas académicos y del mundo real. Entre las áreas de aplicación de mayor relevancia en ciencia, industria y comercio pueden destacarse [82]: sistemas automáticos y de control, bioinformáticas, de ingeniería, procesamiento de información, manufactura e industria, planificación y calendarización, ruteo y logística, telecomunicaciones, entre otras aplicaciones.

Capítulo 4

Algoritmo de Recocido Simulado para el PST

El presente capítulo se organiza del modo que se describe a continuación. La sección 4.1 describe de forma general el algoritmo de recocido simulado. La sección 4.2 presenta la metodología para tratar información geográfica en formato ASCII-TXT para el problema PST. La sección 4.3 describe el enfoque ávido para crear la primera solución que representa una trayectoria entre un punto origen y un destino a partir de la ruta más corta. La sección 4.4 describe el algoritmo de recocido simulado aplicado a la optimización de trayectorias para el problema PST. La sección 4.5 presenta la estructura de vecindad variable aplicada en el algoritmo de recocido simulado. La sección 4.6 presenta el análisis de sensibilidad del algoritmo de recocido simulado aplicado al problema PST. Por último, la sección 4.7 presenta el análisis general de la complejidad del algoritmo de recocido simulado para el PST.

4.1. Recocido Simulado

Los algoritmos de búsqueda local trabajan con una solución inicial, la cual se va alterando a través de una búsqueda de vecindad que permite acceder

a otras soluciones que pueden ser mejores conforme avanza la búsqueda. Las alteraciones son pequeñas perturbaciones sobre la solución actual, y si la solución perturbada es mejor que la solución actual, la solución actual se sustituye por la nueva solución encontrada. Este proceso continua hasta que no se encuentre ninguna mejor solución o hasta que se cumpla con un criterio de paro. La solución encontrada no necesariamente es la optima global, debido a que como se muestra en la figura 3.1, la búsqueda local es susceptible de quedar atrapada con facilidad en óptimos locales. Para evitar que la búsqueda local quede atrapada, se emplean movimientos que permiten que la solución empeore permitiendo explorar el espacio de soluciones.

El algoritmo de **Recocido Simulado** (RS) conocido en inglés como *Simulated Annealing*, es una técnica basada en el recocido de sólidos que fue propuesta por primera vez por Kirkpatrick en 1983 [79]. El RS basa su analogía en el proceso de recocido de sólidos, cuando un material es sometido a altas temperaturas al punto de fusión, posteriormente es enfriado gradualmente, esto hace que las moléculas sean acomodadas de tal forma que la energía potencial de la configuración de las moléculas sea mínima, alcanzando una mejor resistencia.

El RS es una técnica como solución a problemas de optimización de gran tamaño. Ha sido aplicado al problema del agente viajero TSP [79]. También se ha empleado con éxito para muy diversos propósitos: diseño automático de circuitos integrados [79], supresión automática de ruido en imágenes digitales al objeto de restaurarlas [83], entre otras aplicaciones.

El RS de un objeto para mejorar alguna de sus propiedades consiste en producir al azar cambios en su descripción. Si es posible definir una temperatura en relación a tal propiedad, el proceso comienza a altas temperaturas en la que todos los cambios aleatorios son aceptados, en consecuencia el

aspecto del objeto inicial cambia continuamente, y continua con un paulatino y lento descenso de la temperatura, de manera que cada vez el número de cambios aceptados disminuye más y más, y de una forma particular del objeto acaba por emerger. Finalmente cuando la temperatura es tan baja que ningún cambio se acepta, el algoritmo se detiene con el objeto inicial profundamente alterado, objeto que probablemente pueda encontrarse para la propiedad de interés, dicha propiedad es la función objetivo.

La probabilidad de aceptación puede adoptar diferentes métodos estocásticos como son el método Monte Carlo cuando la distribución de probabilidad es la de Boltzmann. El algoritmo básico es el de Metrópolis [84], que permite simular el comportamiento de un sistema a temperatura constante. El fundamento matemático es posible a través de la teoría de probabilidades, en particular, a través de cadenas de Markov. El algoritmo RS como una secuencia de algoritmos de metrópolis a temperatura decreciente en forma monótona, legitima el uso de la temperatura en un sentido muy amplio del cual derivan las ecuaciones fundamentales del algoritmo a partir de la teoría de la información.

El RS es una búsqueda local iterativa guiada a través de un proceso estocástico, donde un estado j es aceptado con una probabilidad dada por la función de distribución de Boltzmann. El algoritmo de Metrópolis simula el cambio de energía en el proceso de enfriamiento de un sistema físico. Las leyes de la termodinámica establecen que a una temperatura T , la probabilidad de un aumento de energía de magnitud ΔE viene dada por la expresión de la tabla 4.1, donde k es la constante de Boltzmann.

Tabla 4.1: Función de distribución de Boltzmann.

$$P(\Delta E) = e^{-\frac{\Delta E}{k \cdot T}}$$

La simulación de enfriamiento físico en la transición de líquido a sólido se apoya en el algoritmo de Metrópolis [84] de la siguiente manera: dado un estado actual i con su temperatura T , generar un estado j mediante una perturbación, si la diferencia de energías ΔE entre el estado actual y el perturbado es menor o igual que cero, el estado j es aceptado como actual, si $\Delta E \geq 0$ el estado es aceptado con probabilidad dada por la función de distribución de Boltzmann; el algoritmo 2 muestra el algoritmo de Metrópolis con un número α aleatorio generado uniformemente distribuido entre $(0, 1)$, si $\alpha \leq P$, donde P es la *probabilidad de aceptación*, el estado j se acepta como el actual, si $\alpha \geq P$, el estado j se rechaza hasta alcanzar el equilibrio térmico para lo cual es necesario que la distribución de probabilidad de las configuraciones alcance el equilibrio estocástico para la distribución de Boltzmann [85].

Algoritmo 2: Algoritmo de Metrópolis

Dado un estado actual i con su temperatura T ;

mientras *no se alcance el equilibrio térmico* **hacer**

 Generar un estado j mediante una perturbación;

si $\Delta E \leq 0$ **entonces**

 | el estado j es aceptado como actual;

fin

si $\Delta E > 0$ **entonces**

 | el estado es aceptado con probabilidad $P(\Delta E)$;

fin

 Generar un número α aleatorio uniformemente distribuido entre

$(0, 1)$;

si $\alpha < P(\Delta E)$ **entonces**

 | el estado j se acepta como el actual;

fin

si $\alpha \geq P(\Delta E)$ **entonces**

 | el estado j se rechaza;

fin

fin

A continuación se describe la metodología para tratar la información geográfica del problema.

4.2. Metodología para tratar información geográfica ASCII-TXT

Para abordar el problema PDT es necesario facilitar el manejo de la información geográfica a través de un formato único. La mayoría de las herramientas GIS maneja la información de las capas de datos en formato ráster o vectorial. Sin embargo, un capa de datos puede estar en un formato pero no en el otro, es decir, se puede disponer de información de la elevación en formato ráster pero no en su formato vectorial y viceversa. Por este motivo, se simplifica el manejo de la información geográfica de las capas de datos extrayendo la información geográfica de los archivos con extensión .bil y .shp a formato ASCII-TXT a través de la herramienta GDAL/OGR.

Posteriormente, se obtienen los límites del área de estudio para cada escenario a partir del número de filas y columnas de los datos del DEM. Una vez obtenidos los límites del escenario, se construye una matriz de adyacencia para representar el grafo de malla rectangular. Los índices i, j de la matriz de adyacencia representan las coordenadas geográficas del terreno a partir de las coordenadas UTM de las esquinas noroeste y sureste de cada escenario.

La red de nodos es representada por un grafo no dirigido. Un vértice o nodo del grafo representa el centro de una celda ráster del mapa y una arista el costo de viabilidad de pasar entre dos vértices de la red. Las aristas representan el costo de viabilidad de cada celda ráster.

Los costos de viabilidad se determinan a partir de los valores de la escala $0 : 10$, donde el valor 10 representa un costo de viabilidad muy alto, obligando a la búsqueda de otras alternativas de menor costo y el valor 0 representa un costo muy bajo por lo que no existe restricción alguna de paso. Para formar la matriz de costos acumulados a partir de n capas de datos, a cada capa de

4.2 Metodología para tratar información geográfica ASCII-TXT

datos se le asigna un valor dentro de la escala de costos de viabilidad según su relevancia en el diseño de la trayectoria. Una vez definido el valor de cada capa de datos, se suman y se ponderan los valores de las distintas capas que coincidan dentro de la misma celda (i, j) en la matriz de costos acumulados. La ponderación de los valores determina el costo de viabilidad de paso entre dos puntos del mapa.

Después de generar el modelo del grafo a través de la matriz de costos acumulados, se evalúa una función de costos que permita hacer una búsqueda a través del grafo para encontrar una trayectoria factible. Si el punto origen y destino son validos, se procede a realizar la búsqueda de la trayectoria. Un algoritmo ávido basado en búsqueda en amplitud es utilizado para generar la primera solución a partir de la función de costos. A partir de la solución inicial, se realiza una búsqueda local sobre una vecindad de soluciones candidatas utilizando la metaheurística de recocido simulado. La figura 4.1 muestra el flujo de los procedimientos de la metodología propuesta.

4.2 Metodología para tratar información geográfica ASCII-TXT

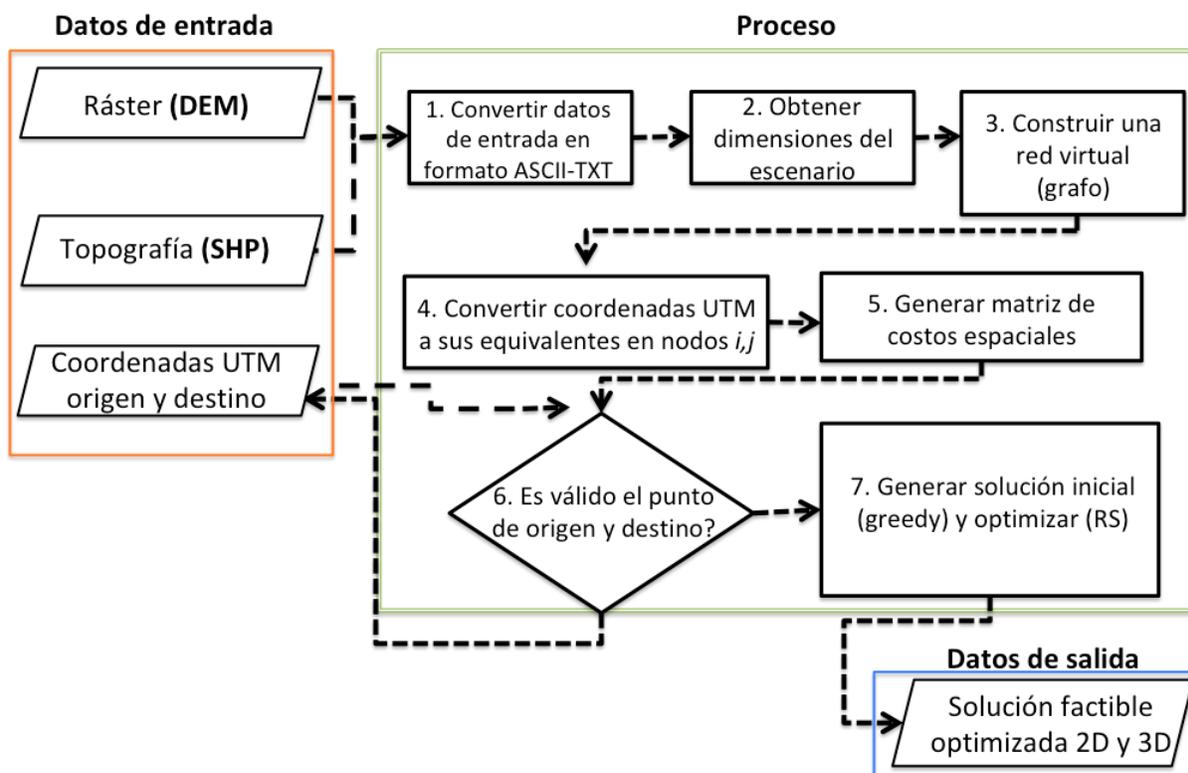


Figura 4.1: Procesos de la metodología propuesta

La metodología se puede resumir en los siguientes pasos a seguir:

1. Convertir datos ráster (DEM) y datos vectoriales (SHP) en formato ASCII-TXT
2. Obtener las dimensiones del escenario a partir de las esquinas noroeste y sureste del DEM
3. Clasificar las capas de los datos en costos de viabilidad para construir una red virtual (grafo)
4. Convertir las coordenadas UTM a nodos i, j en el grafo
5. Generar la matriz de costos acumulados para el análisis espacial

6. Revisar que las coordenadas de los puntos origen y destino sean válidas
7. Generar la solución inicial y minimizar el costo de la función objetivo con el RS

Extracción de datos ráster y vectoriales a ASCII-TXT. La metodología para tratar información geográfica del área de estudio en formato ASCII-TXT a partir de datos vectoriales y ráster se obtiene utilizando GDAL/OGR. Los datos digitales de elevación se encuentran en formato .bil y a través de la herramienta “gdal2xyz.py” se extrae la información a formato CSV. Posteriormente, se concatena el archivo CSV a un archivo TXT con la herramienta “cat” desde la línea de comandos. Por otra parte, para extraer la información geográfica de la topografía que se encuentra en formato vectorial se implementó un script en lenguaje de programación *python* que hace uso de la librería “osgeo.ogr”. De la misma forma que con el formato ráster, la información que se extrae se concatena a un archivo TXT. De esta manera, se obtiene la información en formato ASCII-TXT para posteriormente tratarla en memoria desde el lenguaje de programación C++.

Los detalles de la matriz de costos acumulados o asociados a los costos de viabilidad se describen a continuación.

Generación de la matriz de costos. Luego de obtener las coordenadas de las esquinas noroeste y sureste del escenario, así como los puntos de origen y destino de las instancias de prueba del problema, es necesario crear una matriz de costos acumulados de viabilidad entre nodos del grafo. La viabilidad es la condición del camino o vía por donde se puede transitar [86]. El costo de viabilidad de una capa de datos se determina con base en el criterio de selección de la trayectoria. Por ejemplo, si se va a permitir el paso por cuerpos

4.2 Metodología para tratar información geográfica ASCII-TXT

de agua, el costo de viabilidad de los datos que representan los cuerpos de agua en el mapa debe ser asignado con un valor cercano a 0 o 0. Por otra parte, si en el área por donde puede pasar el ducto hay áreas protegidas, como reservas naturales o zonas arqueológicas, el costo de viabilidad que debe asignarse a los datos que representan tales áreas debe ser cercano a 10 o si se debe prohibir se asigna un valor de 10.

Cabe mencionar, que la matriz de costos acumulados representa una zona específica de un terreno sobre el cual pueden existir más de una capa de datos. Por ejemplo, se puede tener una capa de datos del PH del suelo y a su vez una capa del tipo de vegetación o de vías de comunicación que comparten el mismo espacio. Por tal motivo, los costos de viabilidad deben ser ponderados para que con base en la escala de viabilidad se considere el criterio de selección de la trayectoria en el análisis espacial a partir de la función objetivo.

La figura 4.2 muestra una representación de como el algoritmo ávido busca una trayectoria entre un origen y un destino, tomando en cuenta el criterio de selección a través de la matriz de costos acumulados.

Origen 1	2	5	7	3	2
4	3	2	1	6	4
7	2	8	7	1	5
2	5	6	4	7	3
5	7	3	2	9	4 Destino
1	9	5	7	2	1

Figura 4.2: Representación de una trayectoria sobre la matriz de costos acumulados

La ecuación 4.1 muestra la operación de ponderación de cada celda ráster

de la matriz de costos acumulados.

$$C_{i,j} = \frac{C_{i,j} + P_{capa}}{N_{capas}} \quad (4.1)$$

donde, $C_{i,j}$ representa una celda en la matriz de costos acumulados, P_{capa} es el costo de viabilidad asignado a la capa de datos y N_{capas} es el número considerado de capas de datos para el criterio de selección. Cabe mencionar, que los valores de pesos asignados a cada capa de datos dependerá de los criterios de selección del proyecto que correspondan con las metas y objetivos del mismo. La figura 4.3 muestra una representación del grafo de malla rectangular de un mapa. Los nodos representan el centro de una coordenada UTM i, j y las aristas representan los costos de viabilidad.

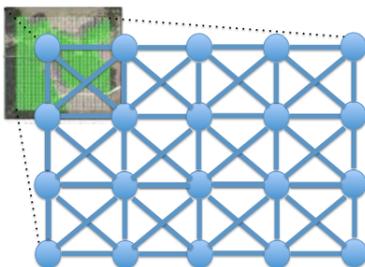


Figura 4.3: Representación del grafo de malla rectangular del mapa

Para crear los escenarios con sus respectivas instancias se tomaron distintos criterios de selección y clasificación de las capas de datos. La tabla 4.2 muestra las capas de datos utilizadas para crear el escenarios 1 con sus respectivos costos de viabilidad asignados de forma arbitraria para probar la metodología propuesta. El escenario 1 se compone por los conjuntos de datos topográficos y de elevación de la carta topográfica E14B49.

4.2 Metodología para tratar información geográfica ASCII-TXT

Tabla 4.2: Capas de datos con sus respectivos costos de viabilidad del escenario 1

Capa de datos	Costo de viabilidad	Capa de datos	Costo de viabilidad
Acueducto	5	Pista carretera	6
Calles	5	Puentes	10
Canal de agua	8	Sub. est. eléctrica	9
Caminos	5	vías férreas	9
Carreteras	7	Aeropuerto	10
Cementerios	8	Áreas de cultivo	1
Ductos	5	Áreas urbanas	8
Edificaciones	8	Áreas verdes	2
Inst. de comunicación	7	Cuerpos de agua	8
Inst. portuarias	9	Ins. industriales	10
Líneas de comunicación	9	Estanques de agua	8
Localidades	6	Inst. deportivas	9

La tabla 4.3 muestra las capas de datos utilizadas para crear el escenarios 2 con sus respectivos costos de viabilidad asignados de forma arbitraria para probar la metodología propuesta. El escenario 2 se compone por los conjuntos de datos topográficos y de elevación de las cartas topográficas E14B69 y E14B79.

Tabla 4.3: Capas de datos con sus respectivos costos de viabilidad del escenario 2

Capa de datos	Costo de viabilidad	Capa de datos	Costo de viabilidad
Acueducto	7	Vía férrea	9
Calles	8	Sub. est. eléctrica	8
Cementerio	9	Áreas verdes	0
Ductos	0	Cuerpos de agua	8
Edificaciones	9	Pobladors	9
Vías de comunicación	10	Inst. deportivas	7
Pista de aviación	9	Inst. diversas	5
Localidades	8	Inst. industriales	10
Puentes	7	Áreas de cultivo	0

4.2 Metodología para tratar información geográfica ASCII-TXT

Por otra parte, si se requiere incluir como criterio de selección de la trayectoria la capa de datos de elevación del terreno, se debe aplicar un porcentaje de influencia que permita evitar pendientes. Por ejemplo, para evitar pendientes muy empinadas, dando una importancia dos veces mayor que el tipo de suelo, es necesario dar un porcentaje de 66% al conjunto de datos de elevación en la matriz de costos acumulada. El porcentaje de influencia restante del total, se asigna a la capa de datos del suelo. La metodología para incluir el porcentaje de influencia en la capa de costos acumulados es la que se aplica para la creación de superficies de coste con varios criterios implementada por ESRI en su software ArcGIS [87].

La figura 4.4 muestra la inclusión de la capa de datos de elevación en la matriz de costos acumulados a través de un porcentaje de influencia y la figura 4.5 muestra la combinación de las capas de datos de elevación y del suelo (topografía) para formar la matriz final de costos acumulados.

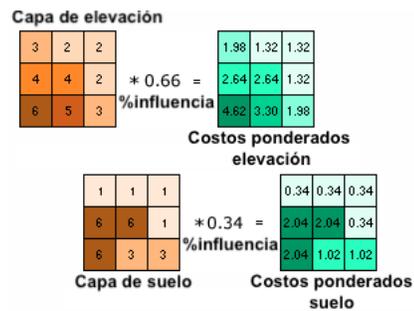


Figura 4.4: Porcentaje de influencia de elevación y suelo

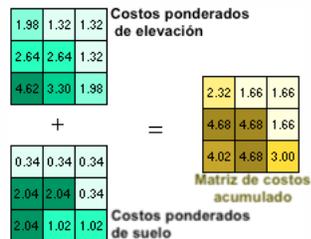


Figura 4.5: Suma de porcentajes de influencia para obtener la matriz de costos acumulados

Una vez generada la matriz de costos acumulados, el análisis espacial permite al algoritmo RS computar un conjunto de rutas alternativas que tengan en cuenta los criterios de selección de diseño establecidos para ser considerados en la toma de decisiones.

4.3. Algoritmo ávido

Un algoritmo ávido (greedy, en inglés) es un método de construcción de soluciones que toma una decisión localmente óptima en cada paso, con la idea de llegar a un óptimo global del problema. El algoritmo ávido implementado en este trabajo se basa en la estrategia de búsqueda en anchura en el grafo “Breadth-First Search-[88][89] (BFS, por sus siglas en inglés). BFS fue utilizado por primera vez para encontrar la ruta más corta en un laberinto. BFS es un algoritmo para atravesar o buscar en grafos y árboles.

El algoritmo BFS puede ser muy apropiado al explorar espacio de búsqueda donde hay una solución esperada que toma un número relativamente pequeño de pasos[90]. Sin embargo, BFS tiene el inconveniente de consumir demasiados recursos espaciales en memoria. El algoritmo BFS comienza en un nodo raíz (nodo origen) y explora todos sus vecinos antes de moverse al siguiente nivel. El algoritmo ávido implementado para generar la primera solución detalla en el Algoritmo 3. Generalmente, el algoritmo ávido comienza de un nodo raíz, y a partir de ese nodo raíz considera cualquiera de sus vecinos adyacentes como el siguiente nodo, hasta llegar al nodo destino. En el proceso de exploración de los nodos vecinos, el algoritmo ávido explora ciegamente las rutas posibles sin considerar ninguna función de costo cuando la búsqueda es no informada. Para que el algoritmo ávido realice una búsqueda informada debe utilizar una función de evaluación que le permita decidir qué

vecino adyacente es mejor con respecto al resto, y de esa forma explorar la red hasta llegar al nodo destino.

Este trabajo implementa un enfoque ávido basado en BFS para generar la primera solución del problema PST, el cual utiliza una cola de prioridad para ir almacenando los costos de los nodos, de tal forma que la búsqueda sea informada.

Algoritmo 3: Algoritmo ávido para generar la primera solución

Datos: Grafo ponderado, nodo s (origen) y nodo t (destino)

Resultado: VectorX, VectorY

1) Crear una Cola de prioridad vacía pq ;

2) Insertar s en pq ;

3) **mientras** pq no esté vacía **hacer**

$u = pq$.EliminaMin ;

si $u = t$ **entonces**

 Finalizar búsqueda ;

en otro caso

 Para cada vecino v de u **si** v no ha sido visitada **entonces**

 Marcar v como visitada;

pq .Insertar(v);

 Etiquetar u como examinado;

fin

fin

fin

La complejidad del método ávido en el peor de los casos es del orden $O(n \cdot \log n)$ donde n corresponde al número de nodos del grafo. Las operaciones de entrada y salida de la cola de prioridad tienen una complejidad del orden $O(\log n)$, y depende de si se busca minimizar o maximizar. El rendimiento del

algoritmo depende de que tan bien informada esté la búsqueda.

Generar primera solución. Una solución inicial se genera a partir de la búsqueda de una trayectoria entre un nodo origen y uno destino en el grafo a partir de una búsqueda informada utilizando la función objetivo descrita en la sección 2.5.

La figura 4.6 muestra la codificación de la solución a través de un vector solución entre un nodo origen y un destino.

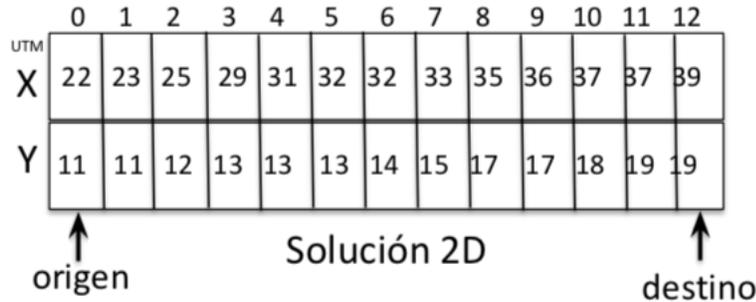


Figura 4.6: Representación de la solución a través de un vector para una trayectoria 2D

Los valores contenidos dentro del vector representan los índices i, j de los nodos del grafo. Para convertir los valores de las coordenadas (x, y) UTM a sus valores (i, j) del grafo se utiliza la ecuación 4.2 y para realizar la operación inversa la ecuación 4.3.

$$\begin{aligned}
 i &= \frac{UTM_x - MIN_x^{Box}}{escala_{mapa}} \\
 j &= \frac{UTM_y - MIN_y^{Box}}{escala_{mapa}}
 \end{aligned}
 \tag{4.2}$$

$$\begin{aligned}
 UTM_x &= (i \cdot escala_{mapa}) + MIN_x^{Box} \\
 UTM_y &= (j \cdot escala_{mapa}) + MIN_y^{Box}
 \end{aligned}
 \tag{4.3}$$

4.4. Algoritmo de Recocido Simulado para el PST (RS-PST)

Este trabajo de investigación busca minimizar el costo de la función objetivo. La función objetivo se describe en el capítulo 2, plantea minimizar el costo de la distancia en kilómetros para encontrar la trayectoria más corta entre el nodo origen y el destino. El algoritmo RS para el problema PST se detalla en el Algoritmo 4.

4.4 Algoritmo de Recocido Simulado para el PST (RS-PST)

Algoritmo 4: Algoritmo RS para el problema PST

Datos: *Grafoponderado*, *Vector* $X_{inicial}$, *Vector* $Y_{inicial}$

Resultado: *Vector* X_{optimo} , *Vector* Y_{optimo}

$s = s_{inicial}$;

mientras *no se cumpla el criterio de paro* $T_0 \leq T_f$ **hacer**

mientras $M \leq L_{Markov}$ **hacer**

 Seleccionar aleatoriamente 2 puntos de s ;

 Crear s' generando un camino aleatorio entre los dos puntos en

$N(s)$;

si $f(s') < f(s)$ **entonces**

$s \leftarrow s'$;

fin

 generar un número aleatorio α en $(0, 1)$;

si $\alpha \leq T$ **entonces**

$s \leftarrow s'$;

fin

fin

$\epsilon \in (0, 1)$ tasa de enfriamiento;

$T_0 \leftarrow \epsilon \cdot T_0$;

fin

Donde *Vector* $X_{inicial}$ y *Vector* $Y_{inicial}$ corresponden a los vectores de la solución generada por el algoritmo BFS, es decir, a la solución inicial $s_{inicial}$. Los vectores *Vector* X_{optimo} y *Vector* Y_{optimo} corresponden a la mejor solución encontrada por el algoritmo de optimización. T_0 y T_f corresponden a los parámetros de control del RS temperatura inicial y temperatura final respectivamente, y L_{Markov} corresponde al parámetro de la longitud de la cadena de Markov utilizado por el algoritmo Metrópolis.

La figura 4.7 muestra en color rosa la trayectoria encontrada por el algoritmo recocido simulado y la trayectoria de color amarillo es la encontrada por el algoritmo ávido.

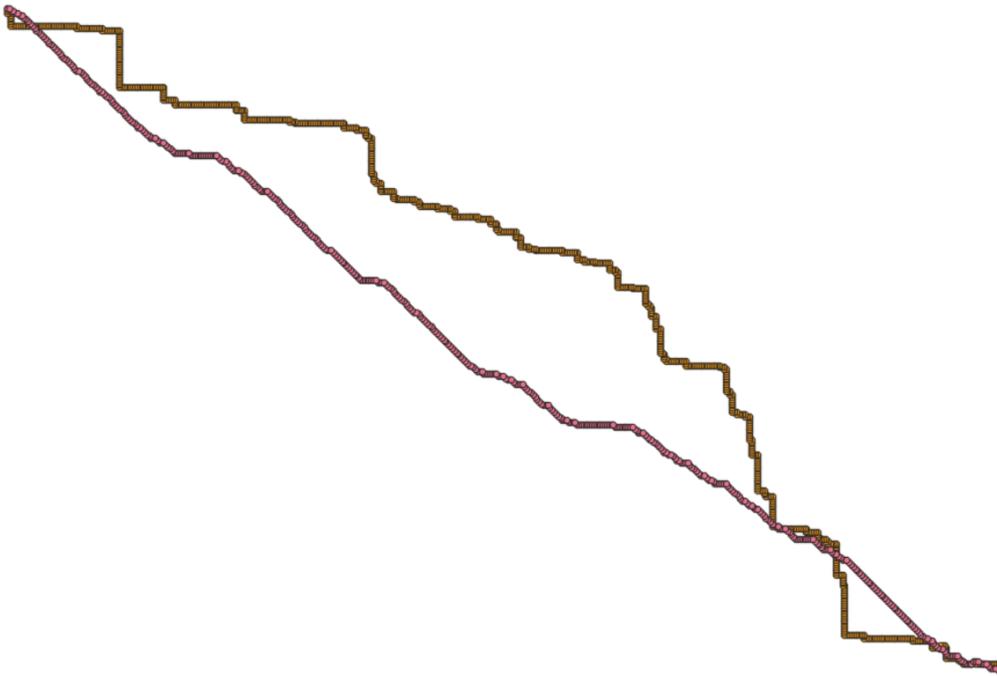


Figura 4.7: Representación de la trayectoria inicial (amarillo) y la óptima (rosa)

A continuación se describe la estructura de vecindad variable implementada en el algoritmo de recocido simulado.

4.5. Estructura de vecindad

Para mejorar la calidad de la solución se hace uso de estructuras de vecindad aplicadas al algoritmo RS, debido a que muchos problemas de optimización combinatoria requieren de búsquedas de soluciones aproximadas y ajustadas a la complejidad y naturaleza del problema [42]. Una vecindad es el conjunto de soluciones las cuales se pueden alcanzar a partir de una solución

s por medio de un movimiento σ/s [60], que puede ser un intercambio, inserción o eliminación entre los elementos de una solución s . Tomando en cuenta la definición anterior, una vecindad es definida como el conjunto de soluciones cercanas de una solución inicial, $s \in S$ en una instancia del problema, tal que, el conjunto $N(s)$ es factible en un punto cercano a s . El conjunto $N(s)$ llamado vecindad de s , indica que cada solución $s' \in N(s)$ puede ser alcanzado directamente desde el valor actual de la función objetivo en un sólo paso (ver figura 4.8). De acuerdo con esto, la vecindad defina por la función $N : S \rightarrow 2^S$ [91]. Para mejorar la solución s , es necesario movernos paso por paso desde la solución inicial factible hacia una solución que proporciona el mínimo valor de la función objetivo, la cual usualmente implica el costo. Para el problema PDT la función objetivo consiste en disminuir la distancia de la trayectoria entre un origen y un destino.

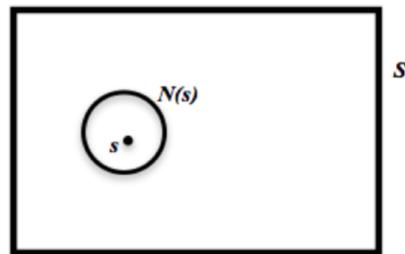


Figura 4.8: Representación del espacio de soluciones de una estructura de vecindad

El procedimiento de búsqueda local comienza de un punto s en $N(s)$, en donde se elige a s' a través de una perturbación determinada por un procedimiento estocástico. Por lo tanto, si $f(s') < f(s)$, la solución s' reemplaza a s y pasa a ser la solución actual del problema. Los detalles del algoritmo de búsqueda por vecindad se detalla en el Algoritmo 5.

Algoritmo 5: Algoritmo general de búsqueda por vecindad

```

Generar solución inicial  $s$  ;
mientras la solución continúe mejorando hacer
     $s' =$  solución inicial perturbada ;
    si  $f(s') < f(s)$  entonces
         $s \leftarrow s'$ ;
    fin
fin

```

Los movimientos realizados sobre s' por medio de la perturbación sobre s , permiten elegir la estructura de vecindad que mejora la calidad de la solución, siempre y cuando se cumpla el criterio de selección. Este proceso se realiza de forma iterativa hasta alcanzar el número de iteraciones determinadas en el algoritmo. Si la solución encontrada no mejora, significa que hemos encontrado un óptimo local. El tamaño de la vecindad se determina por la longitud de la cadena de Markov. Para el problema PDT se realizó un análisis paramétrico que permita definir el valor de L_{Markov} , en donde se tomaron seis valores arbitrarios entre el rango de $(5, 100)$. En el capítulo 6 se muestran los detalles del análisis paramétrico del RS.

Generalmente las estructuras de vecindad intercambian el orden de los elementos de la solución aleatoriamente, sin embargo, intercambiar o agregar elementos aleatoriamente no resulta eficiente en grafos con densidad media. Un enfoque que permita garantizar intercambios válidos en caminos sobre grafos es tomar dos puntos válidos aleatoriamente de la solución y reemplazar con un nuevo camino creado aleatoriamente entre esos puntos nodos.

Por ejemplo, un vector solución que contiene los valores $s = \{51, 33, 41, 7, 12, 91, 60, 32, 41\}$, en donde los puntos seleccionados aleatoriamente son 41

y 60, el sub-camino $\{41, 7, 12, 91, 60\}$ es reemplazado por otro sub-camino generado aleatoriamente, dando como resultado una nueva trayectoria como vector solución $s' = \{51, 33, 41, 29, 71, 89, 72, 60, 32, 41\}$. La figura 4.9 muestra que el movimiento aplicado a la solución s genera una nueva solución s' que puede mejorar el costo de la función objetivo. Sin embargo, el algoritmo RS durante su proceso admite soluciones s' de peor calidad que la solución actual s con la finalidad de explorar el espacio de soluciones (ver figura 4.10).



Figura 4.9: Solución s' (color verde) que mejora el costo de s



Figura 4.10: Solución s' (color verde) que no mejora el costo de s

Estructura de vecindad para el RS-PDT. A partir de una solución s se crea una solución de visitados s_v que va incrementando el número de veces que cada nodo (i, j) ha sido visitado. La figura 4.11 muestra la forma de la matriz de visitados en donde cada celda de la matriz lleva un contador (número color rojo).

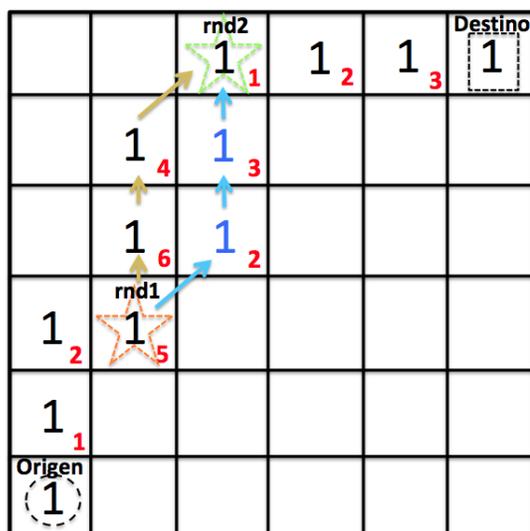


Figura 4.11: Solución s_v para contar el número de visitas de cada celda

En la figura se muestra una solución s que va del punto origen al punto destino. La solución s está representada con el número “1” en color negro. Cada celda que contiene un elemento de la solución s cuantifica el número de veces que la celda ha almacenado un elemento de la solución y de soluciones anteriores.

Para generar un sub-camino, se deben seleccionar dos puntos de la solución s de forma aleatoria. Una vez seleccionados los puntos $rnd1$ y $rnd2$ se validan. Una vez validados los puntos se procede a crear un camino aleatorio entre esos dos puntos. El sub-camino que se genera aleatoriamente se apoya considerando nodos adyacentes que han sido poco visitados. Así mismo, el proceso de generar una ruta aleatoria se apoya a través de una función que calcula la distancia entre $rnd1$ y $rnd2$. En la figura 4.11 se muestra el sub-camino color azul que fue creado entre los puntos $rnd1$ y $rnd2$ generado por un camino aleatorio guiado. Si el número de visitados de una celda es bajo, significa que esa celda ha sido visitada en pocas ocasiones. De esta forma se realiza una exploración amplia sobre el espacio de soluciones.

Para generar el sub-camino entre los punto aleatorios, se crea una matriz $n \cdot m$ generada a partir del punto $rnd1$ como origen y el punto $rnd2$ como destino. Pocas veces la matriz es simétrica, generando una matriz $n \cdot n$. El proceso de generar el sub-camino se realiza solo del punto $rnd1$ al punto $rnd2$ porque el escenario es muy grande.

A continuación se muestra el comportamiento de la estructura de vecindad variable a partir de los puntos aleatorios $rnd1$ y $rnd2$. La matriz de nodos visitados acumula el número de veces que un nodo ha sido considerado en un camino. De esta forma, se da preferencia a nodos menos visitados para seleccionar aleatoriamente el sub-camino. La figura 4.12 muestra la solución actual s en color azul. Los círculos color rojo muestran los nodos origen y destino. Así mismo, los puntos aleatorios $rnd1$ y $rnd2$ se muestran dentro de un cuadrado color verde.

1	2 →	5 →	7 →	3	2
4	↑	2	1	6	4
7	↑	8	7	1	5
2	↑	6	4	7	3
5	→	7	3	2	9
1	9	5	7	2	1

(Note: In the original image, the path 1-2-5-7-3-2 is highlighted in blue with arrows. The nodes 1 and 2 are circled in red and labeled 'Origen' and 'Destino' respectively. The nodes 7 and 3 are highlighted in green.)

Figura 4.12: Solución actual s

La figura 4.13 muestra la solución de la trayectoria s_v en color naranja generada de manera semi-aleatoria a partir de la matriz de nodos visitados.

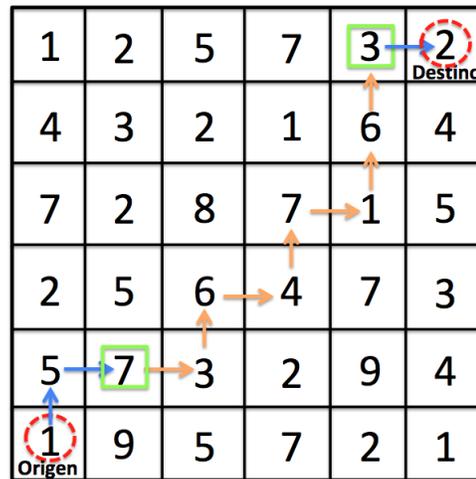


Figura 4.13: Solución semi-aleatoria s_v a partir de nodos visitados

La figura 4.14 muestra la manera en que se van a permutar los nodos para crear la solución de la trayectoria s' , a partir de encontrar la ruta (color rojo) entre dos nodos de las soluciones s y s_v con la menor distancia $dist(s, s_v)$ que haya entre ellos, para generar la solución s' .

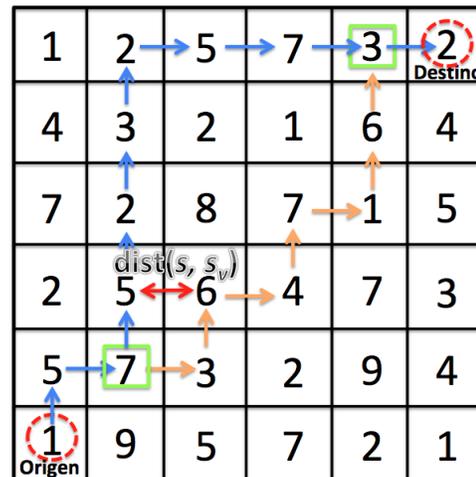


Figura 4.14: Solución s_v

Finalmente, la figura 4.15 muestra como se enlazan las trayectorias s y s_v para crear la solución permutada s' . Una vez creada la solución s' , se evalúa

el costo de la solución para que el algoritmo recocido simulado pueda mejorar la calidad de la solución durante su ejecución.

1	2	5	7	3	2
4	3	2	1	6	4
7	2	8	7	1	5
2	5	6	4	7	3
5	7	3	2	9	4
1	9	5	7	2	1

Origin (1,1) and Destino (1,6) are marked with red circles. Arrows indicate a path from (1,1) to (1,6) through (1,2), (2,2), (2,3), (3,3), (3,4), (4,4), (4,5), (5,5), (5,6).

Figura 4.15: Solución s'

La figura 4.16 muestra algunas soluciones que se obtuvieron aplicando la estructura de vecindad variable entre dos puntos que corresponden a la instancia de prueba #3 descrita más adelante.

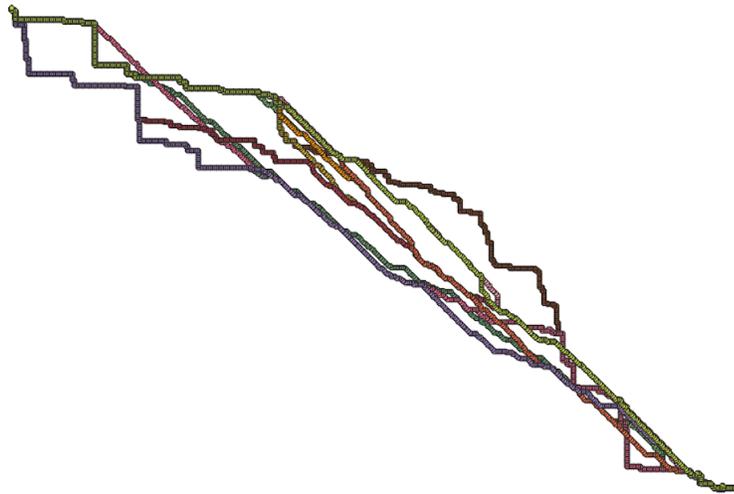


Figura 4.16: Soluciones encontradas por el RS-PST con la estructura de vecindad variable

A continuación se muestra el análisis de sensibilidad del algoritmo RS-PST para seleccionar los parámetros que mejoren el rendimiento del algoritmo.

4.6. Análisis de sensibilidad del algoritmo RS-PST

Debido a la cualidad estocástica de las metaheurísticas, distintas ejecuciones sobre una misma instancia pueden alcanzar resultados diferentes. Con el fin de obtener resultados estadísticamente significativos, se realizaron 30 ejecuciones independientes del algoritmo RS sobre una instancia de prueba. Las ejecuciones independientes se logran inicializando el generador de números pseudo aleatorios del RS con una semilla diferente en cada ejecución. Los resultados obtenidos por el algoritmo RS sobre la instancia de prueba #3 descrita en la tabla 5.2, se reportan indicando el mejor y peor valor, el valor promedio y la desviación estándar de los valores alcanzados en las 30 ejecuciones realizadas.

La configuración paramétrica o análisis de sensibilidad, es un estudio experimental que se realiza sobre los parámetros de control del algoritmo, con la finalidad de encontrar los valores que estadísticamente mejoren el comportamiento tanto en términos de calidad de la solución como de la eficiencia computacional.

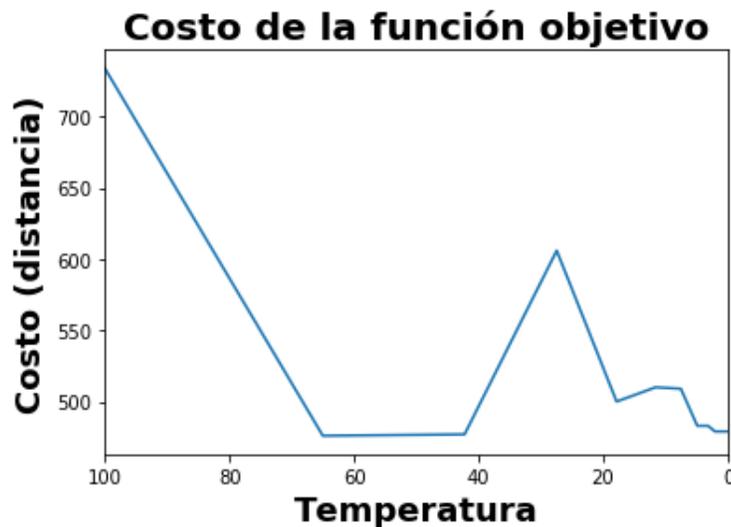
Dado un conjunto de parámetros $P = \{p_1, p_2, \dots, p_n\}$ que requieren ser sintonizados, donde cada parámetro $p \in P$ se compone de un conjunto de valores $V = \{v_1, v_2, \dots, v_n\}$ se procede a realizar un análisis que permita mejorar el comportamiento del algoritmo.

Identificar los parámetros de control. Los parámetros de control del algoritmo RS-PST a sintonizar se muestran en la tabla 4.4.

Tabla 4.4: Parámetros de control del algoritmo RS-PST

Parámetro	Descripción
T_0	Temperatura inicial
T_f	Temperatura final (criterio de paro)
β	Coefficiente de control de T_0
L_{Markov}	Longitud de la cadena de Markov

Los rangos fueron establecidos a partir de una análisis previo utilizando valores arbitrarios con la finalidad de estudiar el comportamiento del algoritmo. La figura 4.17 muestra que el parámetro de control de la temperatura tiene un comportamiento similar en un rango de valores entre 100 y cercano a cero. Por este motivo, los rangos para las muestras del parámetro control de la temperatura son asignados en un rango entre 2 y 0.01.

**Figura 4.17:** Costo de la solución (distancia) en función de la Temperatura (T_0)

Establecer rangos. Los rangos de los valores asignados a cada parámetro de control se definen a partir de resultados previos obtenidos con los cuales la calidad de la solución continuó mejorando. Para el algoritmo RS-PST, loa

rangos definidos para los parámetros de control se muestran en la tabla 4.5.

Tabla 4.5: Rangos de los parámetros de control del algoritmo RS-PST

Parámetro de control	Límite superior	Límite inferior
T_0	2.0	0.01
T_f	0.01	0.0001
β	0.990	0.955
L_{Markov}	100	5

Tamaño de las muestras. Con base en los rangos establecidos en la tabla 4.5, se establece un número de muestras para cada rango que permitan evaluar el comportamiento del algoritmo. La tabla 4.6 muestra los valores de las muestras a utilizar en el análisis de sensibilidad.

Tabla 4.6: Muestras definidas para cada uno de los parámetros de control del algoritmo RS-PST

Parámetro de control	Valores de las muestras a utilizar
T_0	2.0, 1.3, 0.9, 0.3, 0.01
T_f	0.1, 0.01, 0.001, 0.0001
β	0.99, 0.98, 0.97, 0.96, 0.95
L_{Markov}	100, 50, 20, 10, 5

Proceso de sintonización de parámetros. El objetivo de la configuración paramétrica es evaluar cada uno de los valores v definidos para uno de los parámetros de control p con base en la función objetivo. La calidad de la solución permite fijar los valores de los parámetros de control con lo que se obtienen los mejores resultados en promedio.

El procedimiento general de sintonización consiste en evaluar las permutaciones sobre los valores de un parámetro p , mientras se mantiene fijo el primer valor de los parámetros restantes. Por ejemplo, para sintonizar T_0 , se fija el valor inicial de la muestra para T_0, T_f, β y L_{Markov} y se realizan 30 ejecuciones

independientes del algoritmo RS para una instancia. Después de finalizar las 30 ejecuciones, se toma el segundo valor de la muestra para T_0 , se sustituye y se mantienen fijos los mismos valores para T_f, β y L_{Markov} y se realizan 30 ejecuciones independientes, así hasta cubrir el total de permutaciones de un parámetro p . El valor del parámetro T_0 que permita obtener el mejor resultado sobre la media de las 30 ejecuciones, determina el valor a fijar del parámetro de control.

Una vez definido el valor del parámetro de control T_0 , se repite el mismo proceso sobre el siguiente parámetro de control. De tal modo que, los parámetros T_f, β y L_{Markov} deben quedar sintonizados utilizando el mismo procedimiento. Los valores obtenidos en la configuración paramétrica permiten que el algoritmo tenga un comportamiento eficiente y eficaz, es decir, que obtiene buenas soluciones. La figura 4.18 muestra el comportamiento de T_0 . Con base en la gráfica que se muestra en la figura, el valor sintonizado de la temperatura inicial que mejores resultados en términos de calidad de la función es $T_0 = 0.9$.

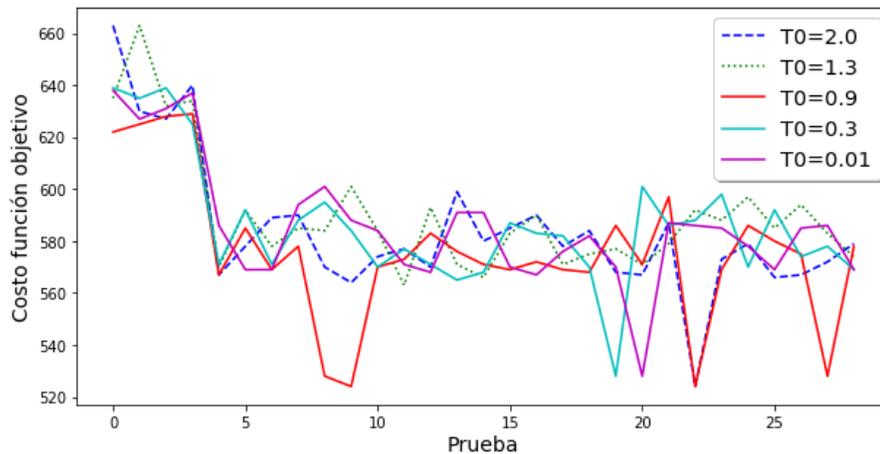


Figura 4.18: Costo de la solución en función de la Temperatura (T_0)

La tabla 4.7 muestra los resultados del comportamiento de las distintas

4.6 Análisis de sensibilidad del algoritmo RS-PST

muestras del parámetro de control $T0$ sobre la calidad de la solución, indicando el mejor y el peor valor, el valor promedio y la desviación estándar (σ) de los valores alcanzados en las 30 ejecuciones realizadas. El parámetro de control $T0$ muestra tener un mejor comportamiento en promedio con un valor de 0.9.

Tabla 4.7: Análisis estadístico del comportamiento del parámetro de control $T0$

Valor de la muestra	Mejor valor	Peor valor	Promedio	σ
2.0	524	663	585.46	27.44
1.3	563	663	591.23	23.85
0.9	524	629	577.56	28.66
0.3	528	639	587.43	24.82
0.01	528	638	587	24.30

La figura 4.19 muestra el comportamiento del costo de la función objetivo en función de los parámetros del coeficiente de control β . Con base en la gráfica que se muestra en la figura, el valor sintonizado de β que mejores resultados en términos de calidad de la función es $\beta = 0.98$.

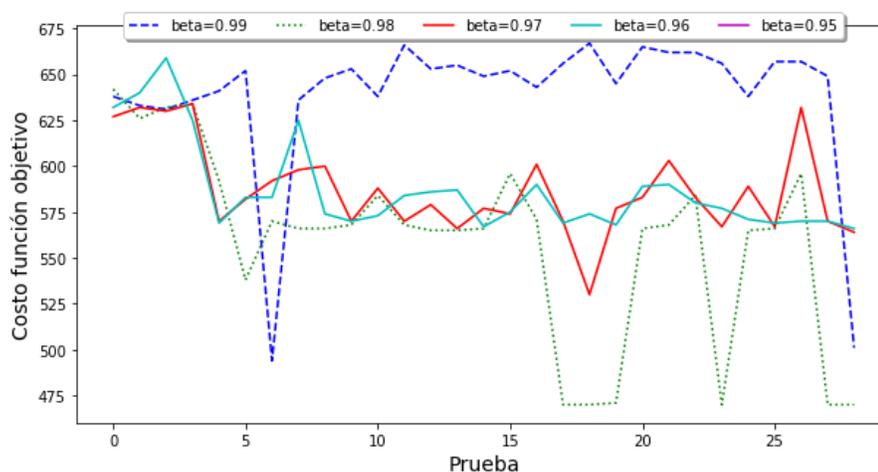


Figura 4.19: Costo de la solución en función del factor de decremento ($beta$)

La tabla 4.8 muestra los resultados del comportamiento de las distintas muestras del parámetro de control β sobre la calidad de la solución, indicando

el mejor y el peor valor, el valor promedio y la desviación estándar (σ) de los valores alcanzados en las 30 ejecuciones realizadas. El parámetro de control β muestra tener un mejor comportamiento en promedio con un valor de 0.98.

Tabla 4.8: Análisis estadístico del comportamiento del parámetro de control β

Valor de la muestra	Mejor valor	Peor valor	Promedio	σ
0.99	494	667	638.66	39.88
0.98	470	642	561.4	52.69
0.97	530	634	588.4	25.57
0.96	566	659	588.06	25.36
0.95	528	638	582.13	19.31

La figura 4.20 muestra el comportamiento del costo de la función objetivo en función de la longitud de la cadena de Markov. Con base en la gráfica que se muestra en la figura, el valor sintonizado de L_{Markov} que mejores resultados en términos de calidad de la función es 10.

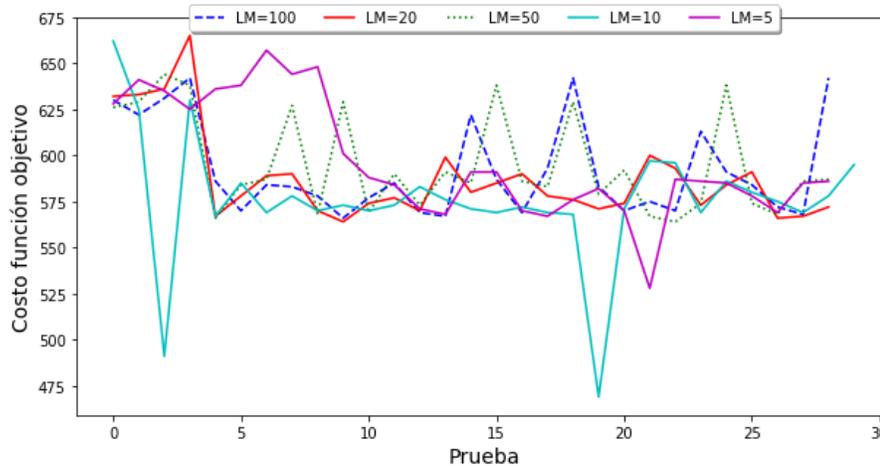


Figura 4.20: Costo de la solución en función de la longitud de la cadena de Markov

La tabla 4.9 muestra los resultados del comportamiento de las distintas muestras del parámetro de control L_{Markov} sobre la calidad de la solución,

indicando el mejor y el peor valor, el valor promedio y la desviación estándar (σ) de los valores alcanzados en las 30 ejecuciones realizadas. El parámetro de control L_{Markov} muestra tener un mejor comportamiento en promedio con un valor de 10.

Tabla 4.9: Análisis estadístico del comportamiento del parámetro de control L_{Markov}

Valor de la muestra	Mejor valor	Peor valor	Promedio	σ
100	566	642	593.26	26.11
50	564	644	596.60	27.16
20	564	665	589.03	25.11
10	469	662	577.12	36.54
5	528	657	598.06	31.63

La figura 4.21 muestra el comportamiento de la función objetivo en función del criterio de parada, a través de la temperatura final (T_f). Con base en la gráfica que se muestra en la figura 4.21, el valor sintonizado de T_f que mejores resultados obtiene en términos de calidad de la función es 0.0001

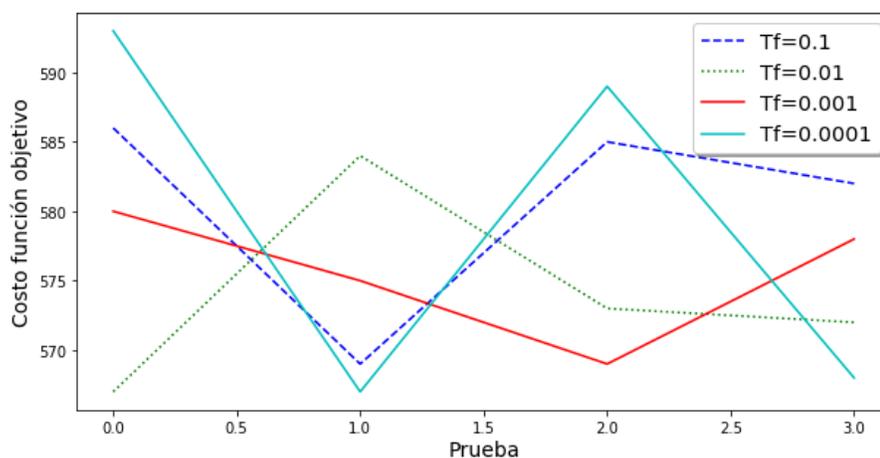


Figura 4.21: Costo de la solución en función del criterio de paro (T_f)

Con base en el análisis del comportamiento del algoritmo sobre la instancia de prueba #3 utilizada para la configuración paramétrica, los parámetros de

control que mejoran en promedio la calidad de la solución del algoritmo se muestran en la tabla 4.10.

Tabla 4.10: Valores sintonizados de los parámetros de control del algoritmo RS-PST

Parámetro	Valor
T_0	0.9
T_f	0.0001
β	0.98
L_{Markov}	10

A continuación se describe la complejidad del algoritmo RS-PST propuesto en este trabajo de investigación.

4.7. Análisis de la complejidad del algoritmo RS-PST

Para calcular la complejidad del algoritmo RS propuesto en este trabajo de investigación se evalúa la cantidad de instrucciones que realiza el algoritmo. La complejidad temporal $T(n)$ representa el número de unidades de tiempo tomadas por un algoritmo de cualquier entrada de tamaño n . Por ejemplo, un algoritmo puede tener un tiempo de ejecución $T(n) = c \cdot n$, donde c es una constante. El tiempo de ejecución de un algoritmo es linealmente proporcional al tamaño de la entrada [92]. El tiempo medio de ejecución suele ser la medida más realista. El tiempo medio de ejecución implica que todas las entradas de tamaño n son igualmente probables en una situación dada.

En el caso del problema PST la variable n corresponde al tamaños del vector solución. Su tamaño depende del número de puntos que hay entre el origen y destino. El número de puntos está en función de la escala de los datos y la

longitud de la distancia. Por ejemplo, para una solución de longitud de 40 km. y datos a escala 1:50,000, obtenemos el tamaño aproximado del vector solución a través de la ecuación $\frac{40 \cdot 1000}{50}$ dando como resultado el valor de 800. Esto quiere decir que el tamaño del vector solución requiere aproximadamente de 800 elementos para representar los puntos de la trayectoria. Para una distancia más larga, por ejemplo, una trayectoria de 400 km y datos con la misma escala, el tamaño del vector solución estaría en el orden de 8,000 elementos. Pero si modificamos la escala de los datos a una aproximación de 5m, la solución de 40 km. requiere un vector que almacene cerca de 20,000 puntos y para el segundo ejemplo cerca de 80,000 puntos. Por este motivo, la estructura de vecindad juega un papel importante dada la complejidad computacional de permutar soluciones de tamaño n . En el anexo C se describe la complejidad requerida por las sentencias de una función en la notación big-O descrita en el capítulo 2.

El tiempo de ejecución de una función se puede determinar como el tiempo requerido para cada una de las sentencias (instrucciones), multiplicadas por el número de veces que se ejecuta (si se encuentra dentro de un bucle). Por lo tanto, el tiempo de ejecución de la función del algoritmo RS-PST se determina en la ecuación 4.4. Al inicio de cada sentencia en el Algoritmo 6 se identifica el tiempo t_i que requiere la sentencia.

Algoritmo 6: Pseudo-código del algoritmo RS-PST

```

 $t_1 \leftarrow s = s_{inicial};$ 
 $t_2 \leftarrow$  mientras no se cumpla el criterio de paro  $T_0 \leq T_f$  y  $T_0 \leftarrow \epsilon \cdot T_0$ 
hacer
   $t_3 \leftarrow$  mientras  $l \leq L_{Markov}$  hacer
     $t_4 \leftarrow$  Crear  $s'$  aplicando  $H_n$  entre los dos puntos en  $N(s)$  ;
     $t_5 \leftarrow$  si  $f(s') < f(s)$  entonces
       $s \leftarrow s'$ ;
    fin
     $t_6 \leftarrow$  generar un número aleatorio  $\alpha$  en  $(0, 1)$ ;
     $t_7 \leftarrow$  si  $\alpha \leq T_0$  entonces
       $s \leftarrow s'$ ;
    fin
  fin
fin

```

donde t_1 realiza sólo una operación de asignación, por lo tanto $t_1 = 1$. El tiempo t_2 del bucle realiza dos operaciones de evaluación, la operación de comparación $T_0 \leq T_f$ la realiza n veces y la operación de decremento $T_0 = \beta \cdot T_0$ la realiza $\log n$ veces. El tiempo t_3 del segundo bucle realizar la comparación $l \leq L_{Markov}$ la realiza n veces. Los tiempos t_4, t_5, t_6 y t_7 de las operaciones internas del segundo bucle las realiza n veces. Por lo tanto, simplificando el número de operaciones relevantes del tiempo de ejecución de la función del algoritmo RS-PST tenemos que:

$$T(n) = n + \log n * n + 4n = 5n^2 + \log n \quad (4.4)$$

Para una computadora en particular se calcula el tiempo de ejecución con

respecto a los valores de t_i . Es evidente que el tiempo crece según crece el tamaño de n , por tal motivo se expresa el tiempo de ejecución de tal forma que indique el comportamiento que va a tener la función con respecto al valor de n [92]. Con la notación big-O se expresa una aproximación de la relación entre el tamaño del problema y la cantidad de pasos necesarios para resolverlos, por lo que la ecuación 4.4 se puede expresar en notación big-O de la siguiente manera:

$$f(n) = O(n^2) + O(\log n) \quad (4.5)$$

La ecuación 4.5 muestra que la complejidad del algoritmo RS-PST está en el orden de $O(n^2)$. La figura 2.8 muestra el comportamiento que tendría el algoritmo en función del tamaño de la entrada.

Capítulo 5

Análisis de resultados

El presente capítulo se organiza del modo que se describe a continuación. La sección 5.1 presenta los detalles del entorno de ejecución así como las instancias de prueba y la construcción de los escenarios realistas del estado de Veracruz. La sección 5.2 presenta los parámetros del análisis de sensibilidad que permiten que el algoritmo RS-PST tenga un mejor rendimiento, así como los detalles de convergencia. Es relevante comparar la calidad de la solución obtenida por el algoritmo RS-PST frente a la estrategia ávida. Cabe mencionar que el algoritmo RS-PST sólo valida que es capaz de encontrar la ruta de menor costo lo cual beneficiará en trabajos futuros. Por último, la sección 5.3 presenta los resultados numéricos del algoritmo propuesto. También se muestran las trayectorias optimizadas para el conjunto de instancias de prueba en 2D y 3D sobre el área de estudio y se presenta la complejidad del algoritmo propuesto desde la perspectiva de eficiencia algorítmica.

5.1. Análisis experimental

En esta sección se describen los detalles de la plataforma de ejecución utilizada para realizar el análisis experimental. Así mismo, se resumen las instancias de prueba sobre dos escenarios realistas utilizadas en la evaluación

experimental del algoritmo de Recocido Simulado que resuelve el problema de optimización del diseño de la trayectoria de ductos de transporte.

5.1.1. Entorno de ejecución

La evaluación experimental del algoritmo RS-PST fue realizada en un equipo que utiliza un microprocesador Intel Xeon X5675 a 3.07 Ghz con 12 MB de cache L3 y 24 GB de memoria RAM disponibles. El equipo utilizado pertenece a la plataforma GridMorelos. La GridMorelos es una infraestructura de alto rendimiento que surgió con la necesidad de contar con suficiente infraestructura de cómputo paralela y distribuida para resolver problemas complejos. La infraestructura GridMorelos se encuentra en el Centro de Investigación en Ingeniería y Ciencias Aplicadas (CIICAp) de la Universidad Autónoma del Estado de Morelos (UAEM).

5.1.2. Instancias de prueba y escenarios

Generalmente, el diseño de la trayectoria de un ducto comienza revisando información topográfica y de elevación de la superficie del terreno. La información que se encuentra disponible puede ser del dominio público (Google Maps) o puede ser privada. En México, el Instituto Nacional de Estadística y Geografía (INEGI) [93] es un organismo público autónomo responsable de captar y difundir información de México en cuanto a sus territorios, recursos, poblaciones y economía. Este trabajo hace uso de información geográfica pública del INEGI para desarrollar la metodología propuesta.

Para determinar que datos son relevantes para el análisis espacial es necesario identificar y delimitar el área de estudio. En el año 2017, el presidente de la república mexicana anunció el descubrimiento de un depósito importante de gas y petróleo de excelente calidad ubicado en los campos de producción

Ixachi-1, localizados en la cuenca de Veracruz [94]. Por tal motivo, este trabajo de investigación establece como área de estudio tres localidades del estado de Veracruz de Ignacio de la Llave. El área de estudio comprende la información disponible en los conjuntos de datos de las cartas topográficas con claves E14B49, E14B69 y E14B79 edición 2015, facilitadas por el INEGI.¹

A continuación se describe el enfoque metodológico utilizado en el proceso de generación de instancias realistas a partir de las fuentes de datos consultadas y las herramientas utilizadas.

Instancias de prueba Con el fin de realizar la evaluación experimental del algoritmo RS-PST, se generan un conjunto de instancias realistas del problema sobre dos escenarios ubicados en el estado de Veracruz de Ignacio de la Llave.

El escenario número 1 se compone de las capas de datos correspondientes a la carta topográfica E14B49. Dentro del área de dicha carta topográfica se encuentra el municipio de Veracruz, siendo la ciudad más grande e importante del estado, con una población de 552,156 habitantes en 2010. Su clima es tropical cálido con una temperatura media anual de 25.3° C. El municipio de Veracruz se ubica a 10 m.s.n.m en promedio, sin embargo, la altura máxima de la superficie del terreno de la carta topográfica es de 120 m.s.n.m. La figura 5.1 muestra en margen color rojo la ubicación de la carta topográfica E14B49 del estado de Veracruz de Ignacio de la Llave y la figura 5.2 muestra detalles de la topografía del área que cubre la carta topográfica.

¹El uso de los datos geográficos en este trabajo de investigación se rige bajo los términos de uso libre establecidos por el INEGI en la URL: <http://www.beta.inegi.org.mx/inegi/terminos.html>



Figura 5.1: Ubicación de la carta topográfica E14B49 en el estado de Veracruz

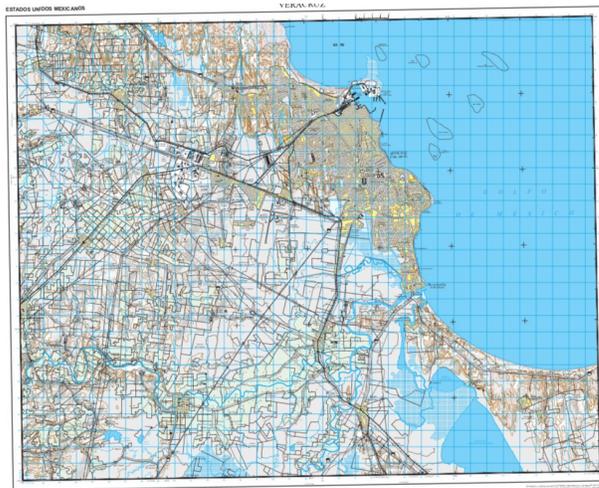


Figura 5.2: Conjunto topográfico E14B49

El conjunto de datos topográficos se encuentra en formato vectorial (SHP) a escala 1:50,000 serie III. El conjunto de datos contiene información sobre los diversos datos espaciales como son curvas de nivel, hidrografía, vías de comunicación, localidades, entre otros. Los datos son representados digitalmente por los componentes geométricos: puntos, líneas y polígonos. La información de localidades, vías de comunicación y cuerpos de agua está actualizada a la fecha de la edición del conjunto de datos vectoriales, dada la naturaleza dinámica con respecto al tiempo.

El escenario número 2 se compone de las capas de datos correspondien-

tes a las cartas topográficas E14B69 y E14B79. Las localidades más importantes dentro del área son “Los Naranjos” por un lado y “Joachín” por otro, dentro del municipio de Tierra Blanca. La altura máxima en el área que comprende a las dos localidades del escenario 2 es de 80 m.s.n.m. La figura 5.3 muestra en margen color rojo la ubicación de las cartas topográficas E14B69 y E14B79 en el estado de Veracruz y las figuras 5.4 y 5.5 muestran detalles de la topografía del área que cubre cada una de las cartas topográficas.

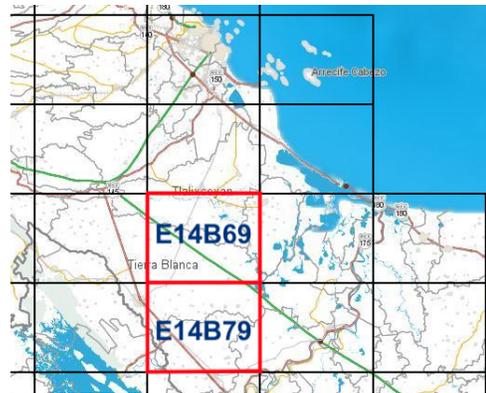


Figura 5.3: Ubicación de las cartas topográficas E14B69 y E14B79 en el estado de Veracruz

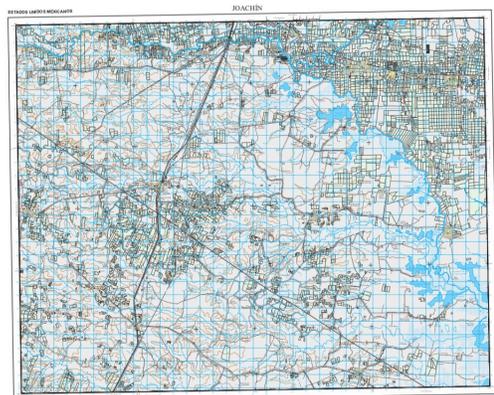


Figura 5.4: Conjunto topográfico de la carta E14B69

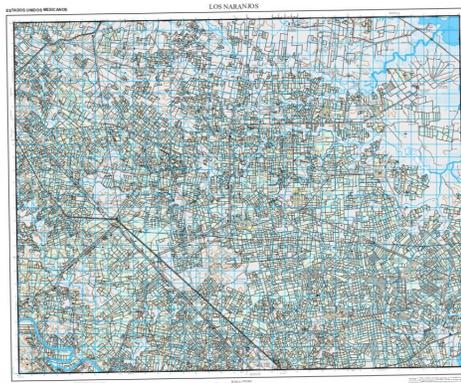


Figura 5.5: Conjunto topográfico de la carta E14B79

Por otra parte, el conjunto de datos del terreno permite gestionar e integrar información de elevación. El modelo de elevación digital (DEM, por sus siglas en inglés) se representa por el valor z asociado a un punto (x, y) del terreno. El DEM provee escalabilidad, eficiencia de almacenamiento de datos y una rápida visualización para realizar análisis del terreno [95]. El DEM en GIS suele estar representado en formato ráster. La figura 5.6 muestra la representación del DEM del escenario 2. El DEM del escenario 1 se compone de una malla ráster de 732 columnas y 586 filas con un intervalo entre elevaciones de 50 metros. La tabla 5.1 muestra las coordenadas UTM de la esquina noroeste y sureste. El DEM del escenario 2 se compone de una malla ráster de 735 columnas y 1171 filas con un intervalo entre elevaciones de 50 metros. La tabla 5.1 muestra las coordenadas UTM de la esquina noroeste y sureste de E14B69 y E14B79.

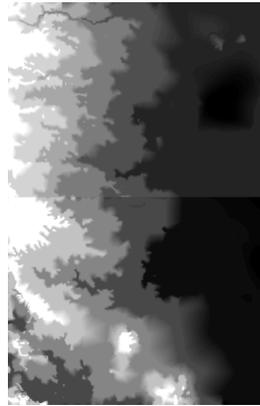


Figura 5.6: Capada de datos DEM de E14B69 y E14B79

El ráster DEM de la figura 5.6 se muestra en una escala en grises. Una celda ráster representa un valor entre la altura mínima y máxima de elevación en cada punto de la superficie del terreno de estudio. Los valores del DEM suelen ser valores reales para representar la elevación en cada punto del terreno en una escala determinada. En este trabajo se discretizaron los valores del DEM para facilitar el análisis espacial.

Tabla 5.1: Coordenadas UTM de los DEM

Carta	Esquina	Este	Norte
E14B49	Noroeste	779775	2131825
	Sureste	816375	2102525
E14B69	Noroeste	780625	2076425
	Sureste	817325	2047125
E14B79	Noroeste	781025	2048725
	Sureste	817775	2019475

Para obtener instancias aplicables al PST, fue necesario determinar las coordenadas origen y destino para múltiples trayectorias sobre un determinado terreno. La tabla 5.2 muestra las trayectorias que deben ser optimizadas para el escenario 1 y la tabla 5.3 muestra las trayectorias para el escenario 2.

Tabla 5.2: Instancias de prueba del escenario 1

No.	Origen	Destino
#1	803135, 2109906	783556, 2125940
#2	781492, 2124670	801283, 2115092
#3	805410, 2105090	782021, 2120701
#4	794615, 2109482	791176, 2128427
#5	790676, 2110898	794142, 2123386

Tabla 5.3: Instancias de prueba del escenario 2

No.	Origen	Destino
#6	805038, 2028689	798519, 2058351
#7	788220, 2027500	803800, 2074200
#8	781867, 2019821	809000, 2069000
#9	793850, 2020745	786300, 2073055
#10	794500, 2037100	801250, 2048600

No se tiene conocimiento de la existencia de benchmarks públicos para el problema de diseño de trayectorias de ductos en el país. Por este motivo, se determinaron de forma arbitraria las coordenadas origen y destino de las múltiples trayectorias descritas con anterioridad. Para probar la viabilidad del método propuesto, se toman como referencia las coordenadas origen y destino de un proyecto del diseño de un oleoducto reportado en el sitio web oficial de la SEMARNAT [96]. Las coordenadas UTM origen y destino del oleoducto reportadas son (785014, 2115995) y (790819, 2124972), la cuales están dentro del área del escenario 2.

Para la evaluación experimental de los algoritmos que resuelven el problema del diseño de trayectorias, se generaron un total de 10 instancias de prueba sobre dos escenarios realistas. La sección 5.1.2 del capítulo 5 describe los detalles utilizados para generar las instancias a partir de los escenarios de prueba sobre tres localidades del estado de Veracruz de Ignacio de la Lla-

ve. Las instancias de prueba del problema están identificadas siguiendo la nomenclatura #X. A modo de ejemplo, la instancia #1 indica que la instancia número 1 tiene las coordenadas UTM (803135, 2109906) como punto de origen y (783556, 2125940) como destino. La tabla 5.4 muestra las coordenadas UTM de los puntos origen y destino correspondientes a cada instancia. La tabla está dividida en dos grupos, el primer grupo contiene las instancias de la uno a la cinco que pertenecen al escenario 1 y el segundo grupo contiene las instancias de la seis a la diez que pertenecen al escenario 2.

Tabla 5.4: Instancias de prueba de los escenarios 1 y 2

No.	Origen	Destino
#1	803135, 2109906	783556, 2125940
#2	781492, 2124670	801283, 2115092
#3	805410, 2105090	782021, 2120701
#4	794615, 2109482	791176, 2128427
#5	790676, 2110898	794142, 2123386
#6	805038, 2028689	798519, 2058351
#7	788220, 2027500	803800, 2074200
#8	781867, 2019821	809000, 2069000
#9	793850, 2020745	786300, 2073055
#10	794500, 2037100	801250, 2048600

La solución inicial para cada instancia de prueba se genera de manera constructiva a través de un método ávido. El algoritmo ávido recibe como datos de entrada la información del grafo y los puntos origen y destino. Una vez obtenida una solución inicial, se procede a guardar la información de la solución a un archivo con extensión CSV. La finalidad de crear el archivo CSV es para visualizar posteriormente la trayectoria a través de una herramienta GIS.

5.2. Parámetros sintonizados del algoritmo RS-PST

Con base en el análisis del comportamiento del algoritmo descrito en 4.6, sobre la instancia de prueba #3 utilizada para la configuración paramétrica los parámetros de control que mejoran en promedio la calidad de la solución del algoritmo se muestran en la tabla 5.5.

Tabla 5.5: Valores sintonizados de los parámetros de control del algoritmo RS-PST

Parámetro	Valor
T_0	0.9
T_f	0.0001
β	0.98
L_{Markov}	10

Los parámetros de control sintonizados del algoritmo son utilizados para realizar las pruebas de convergencia sobre cinco instancias de prueba. Las instancias utilizadas para analizar la convergencia son: #1, #2, #3, #4 y #5 que pertenecen al escenario 1.

Convergencia del algoritmo RS-PST. La convergencia es una propiedad que permite identificar en qué punto un algoritmo alcanza estabilidad a través de una progresión dirigida hacia la uniformidad en los resultados obtenidos. Las pruebas de convergencia se realizaron aplicando los valores sintonizados de los parámetros de control que mejores resultados obtuvieron. Para cinco instancia se realizaron 30 ejecuciones, la cual es considerada como la muestra mínima aceptable para este tipo de análisis estadístico. El parámetro de control relevante para la evaluación de la convergencia es la temperatura. Los

5.2 Parámetros sintonizados del algoritmo RS-PST

resultados de convergencia para las instancias muestran la mejora continua de la solución hasta donde donde el valor de la función objetivo alcanza la convergencia.

La figura 5.7 muestra el comportamiento de la función objetivo con respecto al parámetro de control de la temperatura para la instancia #1.

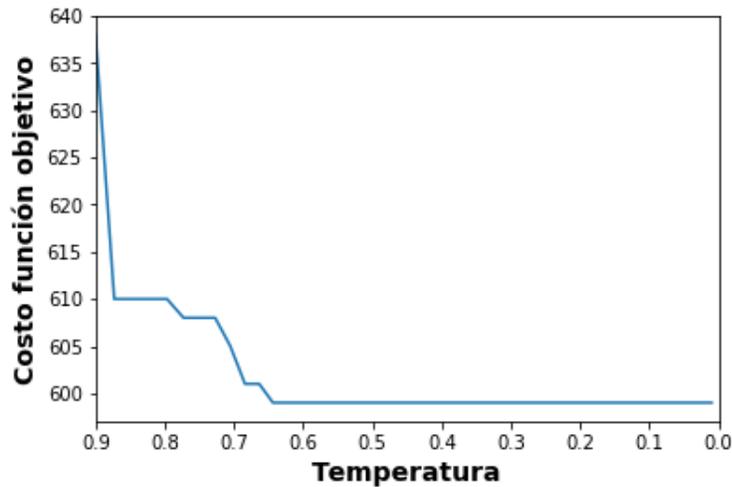


Figura 5.7: Costo de la solución (distancia) en función de la Temperatura

La figura 5.8 muestra el comportamiento de la función objetivo con respecto al parámetro de control de la temperatura para la instancia #2.

5.2 Parámetros sintonizados del algoritmo RS-PST

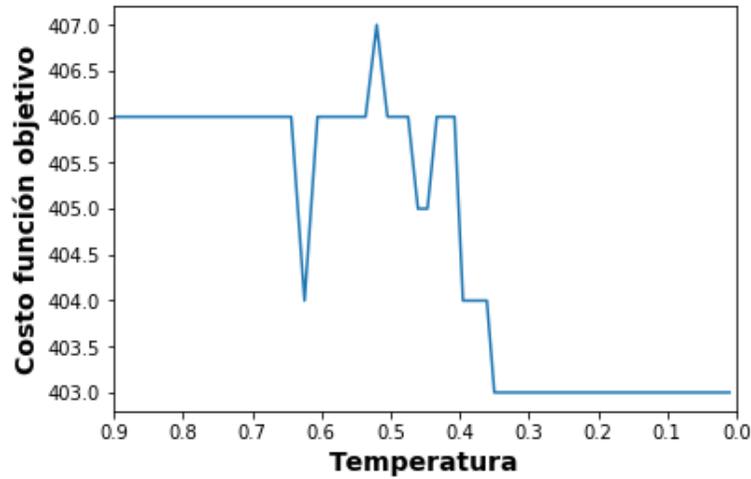


Figura 5.8: Costo de la solución (distancia) en función de la Temperatura

La figura 5.9 muestra el comportamiento de la función objetivo con respecto al parámetro de control de la temperatura para la instancia #3.

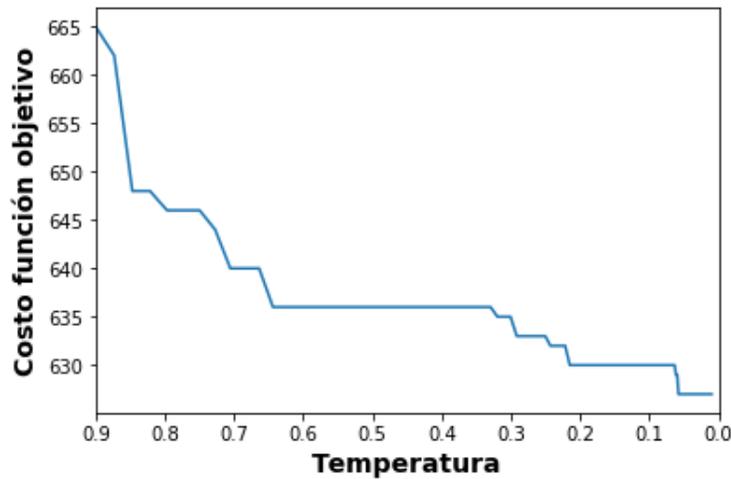


Figura 5.9: Costo de la solución (distancia) en función de la Temperatura

La figura 5.10 muestra el comportamiento de la función objetivo con respecto al parámetro de control de la temperatura para la instancia #4.

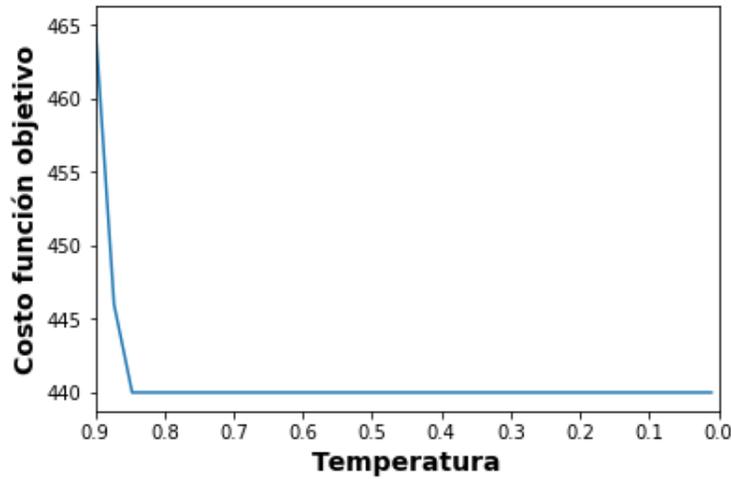


Figura 5.10: Costo de la solución (distancia) en función de la Temperatura

La figura 5.11 muestra el comportamiento de la función objetivo con respecto al parámetro de control de la temperatura para la instancia #5.

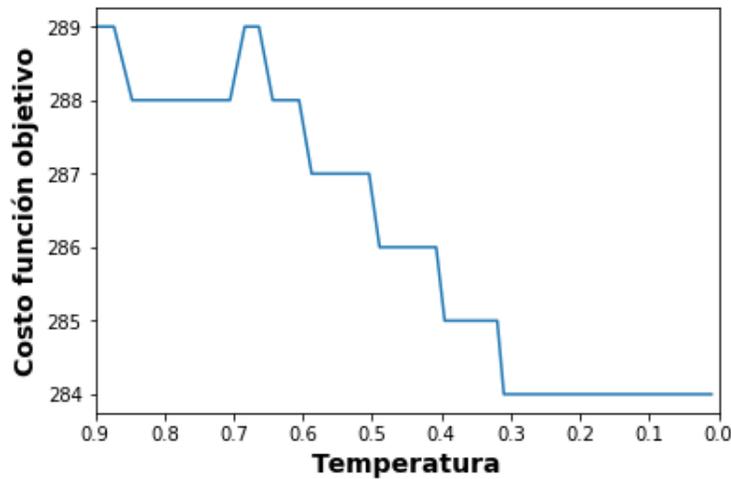


Figura 5.11: Costo de la solución (distancia) en función de la Temperatura

Para conocer el comportamiento del algoritmo, es necesario recurrir al cálculo de parámetros estadísticos como son la media y desviación estándar. La media permite obtener el promedio de las soluciones encontradas para cada instancia. La desviación estándar permite conocer que tan dispersas están

las soluciones con respecto a la media aritmética. En la siguiente sección se muestran los resultados obtenidos por el algoritmo RS sobre las diez instancias de prueba indicando el mejor valor, el valor promedio y la desviación estándar de los valores alcanzados en las 30 ejecuciones realizadas.

5.3. Resultados numéricos

El análisis estadístico permite observar el comportamiento del algoritmo con respecto a las instancias evaluadas. A través del análisis estadístico, se pueden identificar puntos críticos que afectan o benefician el desempeño del algoritmo. La comparación de resultados entre el algoritmo de Recocido Simulado y el algoritmo ávido implementados es con respecto a la calidad de la solución y la eficiencia computacional. La calidad de las soluciones se define de acuerdo a los valores alcanzados en la función objetivo. Para evaluar la eficiencia computacional, se reportan los tiempos de ejecución alcanzados por el algoritmo ávido y el RS implementados para resolver el problema.

5.3.1. Análisis de eficacia

A partir de los valores de los parámetros de control sintonizados, se procede a realizar la evaluación de desempeño del algoritmos sobre las instancias de prueba. La eficacia de un algoritmo esta dada por la calidad de las soluciones. Para el problema RS-PST la calidad de las soluciones del RS se miden con respecto a la obtenida por el algoritmo ávido.

Los resultados obtenidos por el algoritmo RS-PST sobre cada una de las diez instancias de prueba se reportan indicando el mejor valor, el valor promedio y la desviación estándar. La tabla 5.6 muestra que para todas las instancias hubo una mejora en la calidad de la solución con respecto a la solución inicial.

5.3 Resultados numéricos

Tabla 5.6: Análisis estadístico de la función objetivo del algoritmo RS vs BFS

No.	Instancia		Ávido		RS-PST	
	Origen	Destino	Ávido(Km.)	Mejor(Km.)	Promedio	σ
#1	803135, 2109906	783556, 2125940	35.75	29.95	29.99	0.193196
#2	781492, 2124670	801283, 2115092	20.25	20.15	20.17	0.056929
#3	805410, 2105090	782021, 2120701	39.15	31.35	31.56	0.302852
#4	794615, 2109482	791176, 2128427	25.15	22	22.01	0.105503
#5	790676, 2110898	794142, 2123386	16.60	14.20	14.23	0.069305
#6	805038, 2028689	798519, 2058351	36.35	30.75	32.15	1.14175
#7	788220, 2027500	803800, 2074200	62.35	50.1	53.20	1.861871
#8	781867, 2019821	809000, 2069000	76.40	51.85	61.71	5.4664
#9	793850, 2020745	786300, 2073055	60.50	53.1	55.79	1.3733
#10	794500, 2037100	801250, 2048600	18.30	11.65	14.54	1.6021

El análisis de la tabla 5.6 muestra que se obtuvieron mejores resultados en las 10 instancias de prueba con el algoritmo RS-PST con respecto al algoritmo ávido. Así mismo, en la tabla 5.6 se muestran las mejores y peores soluciones obtenidas por el algoritmo RS-PST para cada una de las instancias. El promedio y la desviación estándar de las 30 ejecuciones de cada instancia se muestran en la tabla, donde se aprecia que el grado de dispersión (σ) de las soluciones obtenidas se encuentra con una dispersión mínima de 0.056929 y máxima de 5.4664, indicando que los valores obtenidos se encuentran cercanos al promedio de las soluciones, el cual es cercano a la mejor solución obtenida. Los resultados del análisis estadístico sobre las distintas instancias de prueba demuestran que el algoritmo RS-PST es eficaz con respecto a la calidad de la solución encontrada por el método ávido.

Las instancias presentan un porcentaje de ganancia de la calidad de las soluciones obtenidas por el RS-PST sobre el algoritmo ávido. La tabla 5.7 presenta los porcentajes de mejora de las 10 instancias de prueba. La comparación para determinar el porcentaje de ganancia se realiza sobre el promedio de las

soluciones obtenidas por el RS-PST con respecto a la solución inicial obtenida por el método ávido para cada instancia de prueba.

Tabla 5.7: Porcentaje de mejora de la calidad de la solución del RS-PST vs ávido

Instancia	% Mejora
# 1	16.1
# 2	0.4
# 3	19.4
# 4	12.5
# 5	14.3
# 6	11.6
# 7	14.7
# 8	19.2
# 9	7.8
# 10	20.5

Con base en la tabla 5.7 la eficacia del RS-PST con respecto al método ávido es de **13.65 %** en promedio. Por otra parte, se realizó una prueba de la eficacia del algoritmo RS-PST utilizando las coordenadas UTM de un oleoducto construido por PEMEX reportado por SEMARNAT [96]. La tabla 5.8 muestra los resultados obtenidos con respecto al método ávido y la mejor solución obtenida por el RS-PST.

Tabla 5.8: Resultados comparativos de la calidad de la solución con respecto a un oleoducto del mundo real

Coordenadas UTM		Ávido	PEMEX	RS-PST
Origen	Destino	Ávido(km)	Oleoducto(km)	Mejor Sol. (km)
785014, 2115995	790819, 2124972	18.5	13.5	12.15

La calidad de la solución encontrada por el algoritmo RS-PST resultó ser mejor que el diseño del ducto real por una diferencia de 1 km. de distancia entre las dos soluciones. Las capas de datos utilizadas para realizar el es-

cenario similar al caso real fueron tomadas con base en las reportadas en el documento del proyecto en el portal web de la SEMARNAT. La figura 5.12 muestra las trayectorias encontradas por el algoritmo ávido (color negro) y el RS-PST (color rojo), así como la trayectoria del oleoducto real (color púrpura).

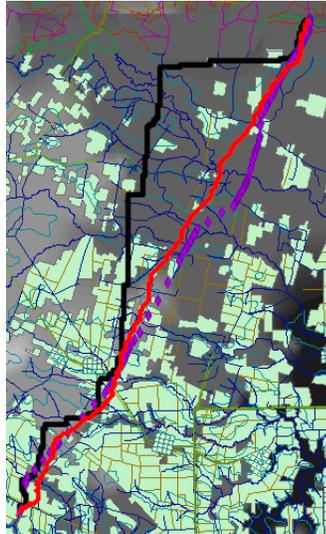


Figura 5.12: Trayectorias ávida, RS-PST y oleoducto real

Por otra parte, es importante saber qué tan eficiente es un algoritmo con respecto a otro para resolver un problema determinado. La eficiencia permite conocer la cantidad de recursos de cómputo necesarios para su ejecución. La siguiente sección describe más detalles respecto a análisis de eficiencia de los algoritmos implementados.

5.3.2. Análisis de eficiencia

El análisis de eficiencia de un algoritmo permite identificar la cantidad de recursos requeridos para su ejecución en función del tamaño de los datos de entrada. El análisis de eficiencia realizado se centra en el tiempo requerido por el algoritmo para ejecutarse en una computadora. Si el algoritmo requiere

una gran cantidad de recursos se dice que no es eficiente, por el contrario si el algoritmo es eficiente permite encontrar soluciones en un tiempo razonable.

El tiempo computacional es de suma importancia en el diseño de un algoritmo. La eficiencia del algoritmo nos permite identificar si el algoritmo es aplicable o no a un determinado problema. Por este motivo, es vital diseñar nuevos algoritmos que sean eficientes para resolver problemas complejos.

El tiempo de ejecución requerido por el algoritmo RS-PST se muestran en la tabla 5.9. Los tiempos reportados en la tabla corresponden al promedio de las 30 ejecuciones por cada una de las instancias.

Tabla 5.9: Comparación de eficiencia en minutos de los algoritmos ávido vs RS-PST

Instancia	Ávida	Prom. RS-PST
# 1	0.0118	28.603
# 2	0.009	12.767
# 3	0.014	32.177
# 4	0.0106	26.497
# 5	0.0105	16.235
# 6	0.0451	34.365
# 7	0.045	52.876
# 8	0.064	54.953
# 9	0.052	57.652
# 10	0.034	14.763

Los tiempos obtenidos por el RS-PST se comparan con los tiempos del algoritmo ávido. El tiempo que demora el algoritmo ávido es menor debido a que sólo genera una solución, la cual es tomada como la solución inicial y que es generada a través de la técnica constructiva. El algoritmo RS-PST realiza una búsqueda local sobre una vecindad $N(s)$ a través de una estructura de vecindad descrita en el capítulo 5, lo cual demanda mayor tiempo de cómputo debido a la naturaleza estocástica de la búsqueda.

5.3.3. Vistas de las trayectorias en 2D y 3D en el mapa

En esta sección se muestran las trayectorias encontradas por los algoritmos implementados a través de una herramienta GIS. Las herramientas GIS permiten gestionar distintos modelos de datos, varios tipos de análisis espacial y proveen la capacidad de visualizar la información sobre la representación de un mapa. Las herramientas GIS cuentan con una interface de usuario que permite de manera gráfica realizar distintas operaciones sobre los datos geográficos. A continuación se muestran las trayectorias sobre el terrenos de estudio de cada una de las instancias de pruebas en 2D y 3D.

La figura 5.13 muestra en 2D la solución inicial de la instancia 1 encontrada por el método ávido (color rojo) y la mejor solución encontrada por el algoritmo RS-PST (color verde) sobre el terreno que se compone por el escenario 1. La figura 5.14 muestra en 3D las soluciones de la instancia 1.

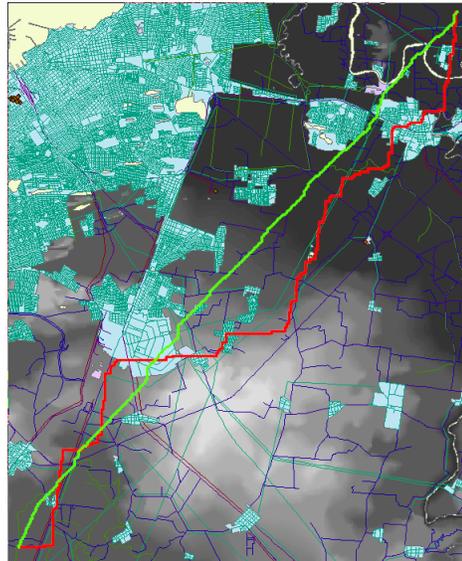


Figura 5.13: Trayectoria inicial vs óptima de la instancia 1 en 2D

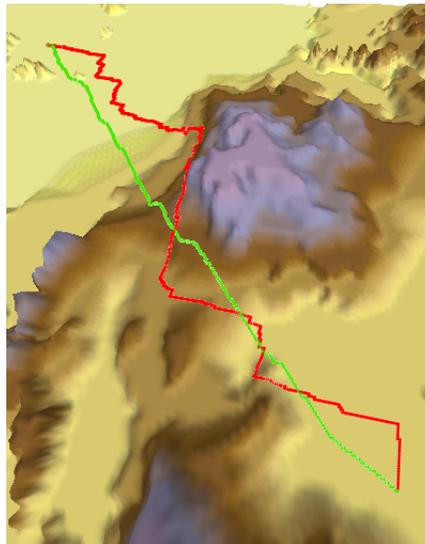


Figura 5.14: Trayectoria inicial vs óptima de la instancia 1 en 3D

La figura 5.15 muestra en 2D la solución inicial de la instancia 2 encontrada por el método ávido (color rojo) y la mejor solución encontrada por el algoritmo RS-PST (color verde) sobre el terreno que se compone por el escenario 1. La figura 5.16 muestra en 3D las soluciones de la instancia 2.

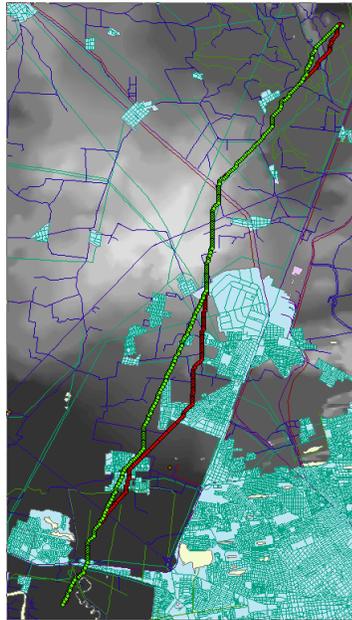


Figura 5.15: Trayectoria inicial vs óptima de la instancia 2 en 2D

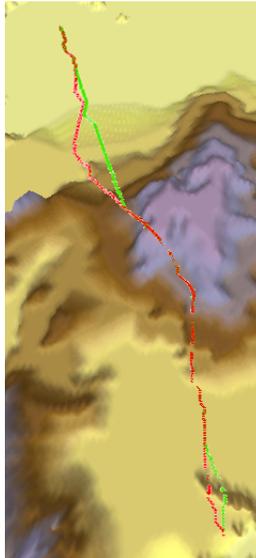


Figura 5.16: Trayectoria inicial vs óptima de la instancia 2 en 3D

La figura 5.17 muestra en 2D la solución inicial de la instancia 3 encontrada por el método ávido (color rojo) y la mejor solución encontrada por el algoritmo RS-PST (color verde) sobre el terreno que se compone por el escenario 1. La figura 5.18 muestra en 3D las soluciones de la instancia 3.



Figura 5.17: Trayectoria v de la instancia 3 en 2D

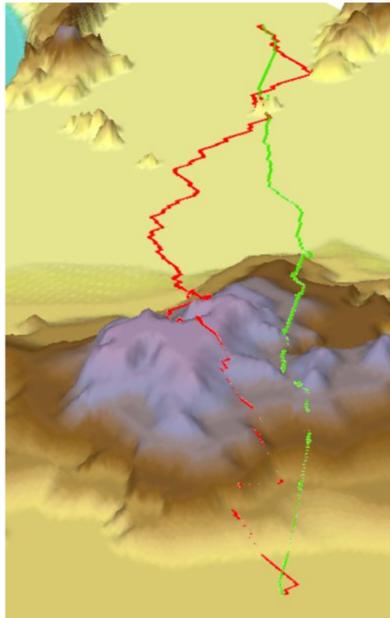


Figura 5.18: Trayectoria inicial vs óptima de la instancia 3 en 3D

La figura 5.19 muestra en 2D la solución inicial de la instancia 4 encontrada por el método ávido (color rojo) y la mejor solución encontrada por el algoritmo RS-PST (color verde) sobre el terreno que se compone por el escenario 1. La

figura 5.20 muestra en 3D las soluciones de la instancia 4.



Figura 5.19: Trayectoria inicial vs óptima de la instancia 4 en 2D

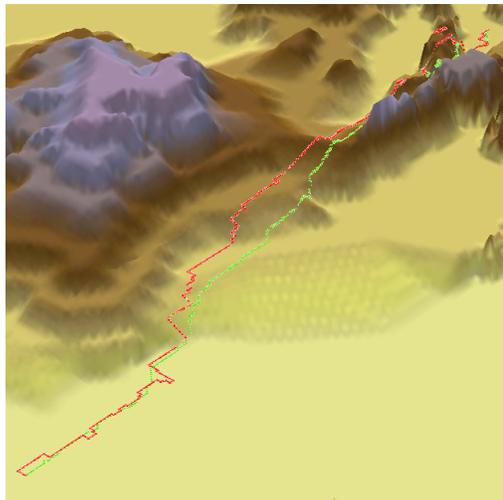


Figura 5.20: Trayectoria inicial vs óptima de la instancia 4 en 3D

La figura 5.21 muestra en 2D la solución inicial de la instancia 5 encontrada por el método ávido (color rojo) y la mejor solución encontrada por el algoritmo RS-PST (color verde) sobre el terreno que se compone por el escenario 1. La figura 5.22 muestra en 3D las soluciones de la instancia 5.

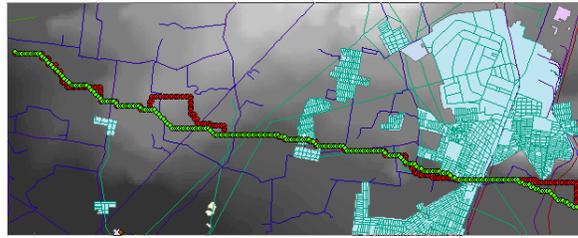


Figura 5.21: Trayectoria inicial vs óptima de la instancia 5 en 2D

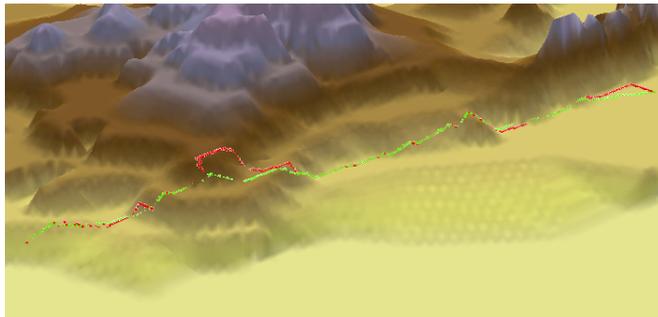


Figura 5.22: Trayectoria inicial vs óptima de la instancia 5 en 3D

Para el escenario 2 se muestran las soluciones inicial y la óptima en 2D. La figura 5.23 muestra en 2D la solución inicial de la instancia 6 encontrada por el método ávido (color rojo) y la mejor solución encontrada por el algoritmo RS-PST (color verde) sobre el terreno que se compone por el escenario 2.



Figura 5.23: Trayectoria inicial vs óptima de la instancia 6 en 2D

La figura 5.24 muestra en 2D la solución inicial de la instancia 7 encontrada por el método ávido (color rojo) y la mejor solución encontrada por el algoritmo RS-PST (color verde) sobre el terreno que se compone por el escenario 2.



Figura 5.24: Trayectoria inicial vs óptima de la instancia 7 en 2D

La figura 5.25 muestra en 2D la solución inicial de la instancia 8 encontrada por el método ávido (color rojo) y la mejor solución encontrada por el algoritmo RS-PST (color verde) sobre el terreno que se compone por el escenario 2.

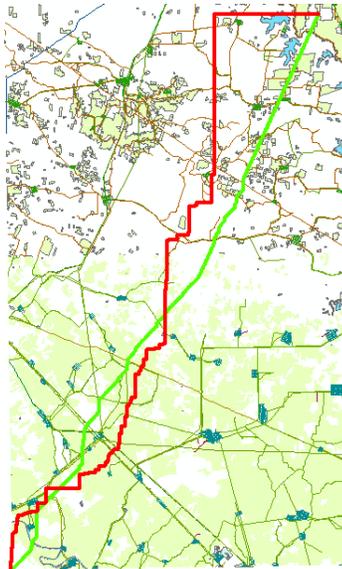


Figura 5.25: Trayectoria inicial vs óptima de la instancia 8 en 2D

La figura 5.26 muestra en 2D la solución inicial de la instancia 9 encontrada por el método ávido (color rojo) y la mejor solución encontrada por el algoritmo RS-PST (color verde) sobre el terreno que se compone por el escenario 2.

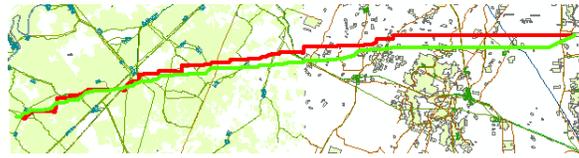


Figura 5.26: Trayectoria inicial vs óptima de la instancia 9 en 2D

La figura 5.27 muestra en 2D la solución inicial de la instancia 10 encontrada por el método ávido (color rojo) y la mejor solución encontrada por el algoritmo RS-PST (color verde) sobre el terreno que se compone por el escenario 2.



Figura 5.27: Trayectoria inicial vs óptima de la instancia 10 en 2D

En las figuras anteriores se aprecia la mejora significativa de la optimización de las trayectorias sobre las diez instancias de prueba las cuales pueden ser aplicadas a ductos de transporte sobre distintos tipos de terrenos. Cabe mencionar que, la clave para realizar el análisis espacial para determinar las mejores alternativas para la trayectorias esta en seleccionar y clasificar las capas de datos relevantes para cumplir los objetivos y metas de diseño establecidas para la toma de desiciones y en la matriz de costos acumulados.

Capítulo 6

Conclusiones y trabajo futuro

Esta sección presenta las conclusiones del trabajo realizado y las principales líneas de trabajo futuro.

6.1. Conclusiones

Este proyecto de investigación presentó una alternativa de solución al problema de seleccionar la trayectoria óptima en un corredor, la cual puede ser aplicada para diseñar ductos de transporte de líquidos valiosos (petróleo crudo, condensados, gasolinas, gases licuados, amoníaco anhídrido líquido, gas natural, etcétera).

Se aplicó una metodología para tratar información geográfica ASCII-TXT a partir de modelos de datos vectoriales y ráster a través de la herramienta GDAL/OGR. A partir de la información de elevación y topográfica del área de estudio, se generó la red basada en teoría de grafos utilizando las coordenadas geográficas del terreno. Así mismo, las capas de datos GIS fueron clasificadas de acuerdo a un factor de impacto sobre la viabilidad de paso para realizar el análisis espacial sobre la red.

Se presentaron dos enfoques, uno para generar la primera solución y el segundo para optimizar la solución. Para generar la primera solución, se im-

plemento un algoritmo ávido basado en BFS para el análisis en el grafo. Para optimizar la solución de la trayectoria inicial, se implementó un algoritmo de recocido simulado con una estructura de vecindad variable mediante la exploración de alternativas de rutas espaciales, para refinar la trayectoria y alinearla al diseño de ingeniería para ductos de transporte. La estructura de vecindad implementada selecciona aleatoriamente dos puntos de la solución de la trayectoria actual y encuentra alternativas de ruta mediante movimientos tabú que acumulan el número de ocasiones que han sido visitados los nodos a través de rutas semi-aleatorias. Esto permite minimizar la función de costos de distancia de las alternativas de ruta exploradas. Cabe resaltar, que la comparación entre la calidad de la solución entre el algoritmo ávido y el recocido simulado permite validar si el algoritmo de RS funciona para calcular la ruta más corta aplicada al problema tratado, con la finalidad de aplicar el método en trabajos futuros.

Por otra parte, se estudió el problema y se mencionaron los principales trabajos relacionados de la literatura, incluyendo trabajos que resuelven problemas similares, métodos exactos y metaheurísticas que resuelven distintos tipos de problemas del mundo real. Además, se realizó un análisis de sensibilidad del comportamiento del algoritmo sobre la instancia de prueba #3. Los parámetros de control que mejoran en promedio la calidad de la solución del algoritmo se muestran en la tabla 5.5. El análisis experimental fue realizado sobre un conjunto de 10 instancias de prueba distribuidas en dos escenarios realistas ubicados en el estado de Veracruz, utilizando información GIS obtenida del INEGI.

El algoritmo RS-PST fue comparado frente al algoritmo base (BFS) en términos de calidad de la solución y en eficiencia. Los resultados experimentales sugieren que el algoritmo RS-PST fue capaz de mejorar la calidad de

la solución obtenida por el algoritmo ávido hasta en un 13.65% en promedio sobre las 10 instancias de prueba. Con respecto a la eficiencia, un recocido simulado requiere en promedio 103.7% más tiempo que el método ávido al generar la solución inicial. Sin embargo, en problemas grandes y complejos el algoritmo ávido es menos eficiente que el algoritmo de optimización.

6.2. Trabajo futuro

A continuación, se detallan las principales líneas de trabajo futuro que surgen a partir del trabajo realizado en el marco de esta investigación.

Una técnica estocástica fue implementada para seleccionar la trayectoria óptima en un corredor aplicado a ductos de transporte, mejorando progresivamente tanto la calidad de la solución como el desempeño computacional. En particular, para el problema de optimización de trayectorias PST aplicado a ductos de transporte se implementó la metaheurística conocida como recocido simulado, la cual fue capaz de alcanzar mejores resultados respecto a los resultados obtenidos por el algoritmo base gracias a la naturaleza estocástica de la técnica para explorar el espacio de soluciones. Por esta razón, se plantean como líneas de trabajo futuro la implementación de otras metaheurísticas basadas en población como son los algoritmos evolutivos, de forma de obtener buenos resultados en calidad y tiempos de ejecución. Así mismo, se plantea ampliar la estructura de vecindad aplicando operadores distintos al utilizado en este trabajo.

El problema abordado considera información geográfica de tres cartas topográficas ubicadas en el estado de Veracruz, con sus respectivas capas de datos topográficas y de elevación. La incorporación de más cartas topográficas que permitan extender el área geográfica de estudio, hace que le com-

plejidad del problema incrementalmente. Por tal motivo, se plantea como parte de las líneas de investigación paralelizar los procesos de tratamiento de información geográfica a través de la creación de otros escenarios con más cartas topográficas, así como distintos tipos de capas de datos por cada carta topográfica con diferentes escalas.

Por último, es de interés estudiar las variantes del problema aplicando más de una función objetivo a optimizar, abriendo la posibilidad de abordar el problema con algoritmos evolutivos multiobjetivo. Así mismo, considerar propiedades y características de distintos fluidos.

Nomenclatura

AE Algoritmo Evolutivo

BFS Breadth First Search

BLI Búsqueda Local Iterada

BT Búsqueda Tabú

CENAGAS Centro Nacional de Control del Gas Natural

DEM Digital Elevation Model

ESRI Environmental Systems Research Institute

GDAL/OGR Geospatial Data Abstraction Library

GIS Geographic Information System

m.s.n.m metros sobre el nivel del mar

MST Minimum Spanning Tree

PEMEX Petróleos Mexicanos

PST Problema de Seleccionar la Trayectoria

RS Recocido Simulado

RS – PST Recocido Simulado para el Problema de Seleccionar la Trayectoria

s.a. Sujeto a

SEMARNAT Secretaría de Medio Ambiente y Recursos Naturales

SHP Shapefile

SIG Sistema de Información Geográfica

SISTRANGAS Sistema de Transporte y Almacenamiento Nacional Integrado de Gas Natural

SPP Shortest Path Problem

SSPP Single Shortest Path Problem

TSP Traveling Salesman Problem

UTM Universal Transverse Mercator

Bibliografía

- [1] Rodrigue, J.P., Comtois C. y Slack B.: *The Geography of Transport Systems*. Routledge, New York, 4ª edición, 2017, ISBN 978-1138669574. 1
- [2] Bautista-Mendoza, C.I.: *Diseño, Construcción y Mantenimiento del Poliducto Tula-Querétaro-San Miguel-Guanajuato*, 2010. 1, 17
- [3] Dubey, R.P.: *A Remote Sensing and GIS based least cost routing of pipelines*. <https://www.geospatialworld.net/article/a-remote-sensing-and-gis-based-least-cost-routing-of-pipelines/>, 2005. 1
- [4] Oseredko, Y.S., Larichev O.I. y Mechitov A.I.: *Main Gas Pipeline Route Selection Problems, Taking into Consideration Risk and Uncertainty Factors*. En Kunreuther, H.C. y Ley E.V. (editores): *The Risk Analysis Controversy*, páginas 91–101, Berlin, Heidelberg, 1982. Springer Berlin Heidelberg, ISBN 978-3-642-81940-7. 1
- [5] Nogueira, A.C. y Mckeehan D.S.: *Chapter 11 - Design and Construction of Offshore Pipelines*. En Chakrabarti, S.K. (editor): *Handbook of Offshore Engineering*, páginas 891 – 937. Elsevier, London, 2005, ISBN 978-0-08-044381-2. 2
- [6] Higa-Ramírez, K., Intriago Barron I., Luna Campos M.G., Mendoza Palacios M. y Quirino Rojas N.: *Diseño, Construcción y Mantenimiento de un Oleoducto Poza Rica-Querétaro*, 2010. 2, 19
- [7] Li, B. y Y. Chen: *Research on the Optimal Selection of a Pipeline Route Program Based on AHP*, páginas 600–608. 2014. 2
- [8] Zhang, H., Y. Liang, W. Zhang, B. Wang, X. Yan y Q. Liao: *A unified MILP model for topological structure of production well gathering pipeline network*. *Journal of Petroleum Science and Engineering*, 152:284 – 293, 2017, ISSN 0920-4105. 2

- [9] Ozanne, H.S.: *Chapter 2 - Route Selection*. En Menon, E. Shashi (editor): *Pipeline Planning and Construction Field Manual*, páginas 43 – 56. Gulf Professional Publishing, Boston, 2011, ISBN 978-0-12-383867-4. [2](#), [26](#)
- [10] Medrano, F.A. y R.L. Church: *Corridor Location for Infrastructure Development: A Fast Bi-objective Shortest Path Method for Approximating the Pareto Frontier*. *International Regional Science Review*, 37(2):129–148, 2014. [3](#)
- [11] Scaparra, M.P., Church R.L. y Medrano F.A.: *Corridor location: the multi-gateway shortest path model*. *Journal of Geographical Systems*, 16(3):287–309, 2014, ISSN 1435-5949. [3](#), [4](#), [26](#)
- [12] Stefanakis, E. y Marinos K.: *On the determination of the optimum path in space*. En Andrew, F. y Werner K. (editores): *Spatial Information Theory A Theoretical Basis for GIS*, páginas 241–257. Springer Berlin Heidelberg, 1995, ISBN 978-3-540-45519-6. [3](#)
- [13] Bagli, S., Geneletti D. y Orsi F.: *Routeing of power lines through least-cost path analysis and multicriteria evaluation to minimise environmental impacts*. *Environmental Impact Assessment Review*, 31(3):234 – 239, 2011, ISSN 0195-9255. [3](#)
- [14] Church, R.L.: *Geographical Information Systems and Location Science*. *Comput. Oper. Res.*, 29(6):541–562, 2002, ISSN 0305-0548. [3](#)
- [15] Dijkstra, E.W.: *A Note on Two Problems in Connexion with Graphs*. *Numer. Math.*, 1(1):269–271, Diciembre 1959, ISSN 0029-599X. [3](#), [35](#)
- [16] Bellman, R.: *On a routing problem*. *Quarterly of Applied Mathematics*, 16(1):87–90, 1958. [3](#)
- [17] Medrano, F.A. y R.L. Church: *A Parallel Algorithm to Solve Near-Shortest Path Problems on Raster Graphs*, páginas 83–94. Springer US, 2013, ISBN 978-1-4614-8745-6. [3](#), [4](#)
- [18] Vanhove, S. y V. Fack: *An effective heuristic for computing many shortest path alternatives in road networks*. *International Journal of Geographical Information Science*, 26(6):1031–1050, 2012. [3](#)
- [19] Goodchild, M.F.: *An Evaluation of Lattice Solutions to the Problem of Corridor Location*. *Environment and Planning A: Economy and Space*, 9(7):727–738, 1977. [4](#)

- [20] Huber, D.L. y L.R. Church: *Transmission Corridor Location Modeling*. Journal of Transportation Engineering, 111(2):114–130, 1985. 4
- [21] Smith, T.R., G. Peng y P. Gahinet: *Asynchronous, Iterative, and Parallel Procedures for Solving the Weighted-Region Least Cost Path Problem*. Geographical Analysis, 21(2):147–166. 4
- [22] Collischonn, W. y Pilar J.V.: *A direction dependent least-cost-path algorithm for roads and canals*. International Journal of Geographical Information Science, 14(4):397–406, 2000. 5
- [23] Baeza, D., Ihle C.F. y Ortiz J.M.: *Comparison between ACO and Dijkstra algorithms for optimal ore concentrate pipeline routing*. Journal of cleaner production, 144:149–160, 2017. 5
- [24] Marcoulaki, E.C., Papazoglou I.A. y Pixopoulou N.: *Integrated framework for the design of pipeline systems using stochastic optimisation and GIS tools*. Chemical Engineering Research and Design, 90(12):2209 – 2222, 2012, ISSN 0263-8762. 5
- [25] Wei, L., H. Dong, J. Zhao y G. Zhou: *Optimization model establishment and optimization software development of gas field gathering and transmission pipeline network system*. Journal of Intelligent & Fuzzy Systems, 31:2375–2382, Septiembre 2016. 5
- [26] Kang, J.Y. y Lee B.S.: *Optimisation of pipeline route in the presence of obstacles based on a least cost path algorithm and laplacian smoothing*. International Journal of Naval Architecture and Ocean Engineering, 9(5):492 – 498, 2017, ISSN 2092-6782. 6
- [27] Rodríguez, D.A., Oteiza P.P. y Brignole N.B.: *Simulated Annealing Optimization for Hydrocarbon Pipeline Networks*. Industrial & Engineering Chemistry Research, 52(25):8579–8588, 2013. 6
- [28] Middleton, R.S. y Bielicki J.M.: *A scalable infrastructure model for carbon capture and storage: SimCCS*. Energy Policy, 37(3):1052 – 1060, 2009, ISSN 0301-4215. 6
- [29] Brimberg, J., Hansen P., Lih K., Mladenović N. y Breton M.: *An Oil Pipeline Design Problem*. Operations Research, 51(2):228–239, 2003, ISSN 0030364X. 7
- [30] Kjaernested, S.N., Jonsson M.Th. y Palsson H.: *Methodology for Pipeline Route Selection Using the NSGA II and Distance Transform Algorithms*. Volumen 5, páginas 543–552, 2011. 7

- [31] Górniak-Zimroz, J.G: *Spatial Planning Using GIS Technology for Optimal Multi-Criteria Location of Belt Conveyor Route*. Volumen 683 de *Applied Mechanics and Materials*, páginas 202–207. Trans Tech Publications, Diciembre 2014. [8](#)
- [32] Soltani, A.R., Tawfik H., Goulermas J.Y. y Fernando T.: *Path planning in construction sites: performance evaluation of the Dijkstra, A*, and GA search algorithms*. *Advanced Engineering Informatics*, 16(4):291–303, 2002. [8](#)
- [33] Luettinger, J. y Clark T.: *Geographic Information System-based Pipeline Route Selection Process*. *Journal of Water Resources Planning and Management*, 131(3):193–200, 2005. [8](#)
- [34] Devine, C.A. y Haneberg W.: *Optimization Methods for Arctic Pipeline Route Selection*. *Offshore Technology Conference*, Enero 2016. [8](#)
- [35] Abudu, D. y Williams M.: *GIS-based Optimal Route Selection for Oil and Gas Pipelines in Uganda*. *Advances in Computer Science*, 4:93–104, Julio 2015. [9](#)
- [36] <https://www.esri.com/en-us/arcgis/about-arcgis/overview>. [10](#)
- [37] <http://desktop.arcgis.com/en/arcmap/10.3/guide-books/extensions/network-analyst/algorithms-used-by-network-analyst.htm>. [10](#)
- [38] <https://www.qgis.org/es/site/about/index.html>. [10](#)
- [39] Hillier, F.S. y Lieberman G.J.: *Introducción a la Investigación de Operaciones*. McGraw-Hill, New York, 9ª edición, 2010, ISBN 978-607-15-0308-4. [11](#)
- [40] Abeysundara, S., Giritharan B. y Kodithuwakku S.: *A Genetic Algorithm Approach to Solve the Shortest Path Problem for Road Maps*. En *Proceedings of the International Conference on Information and Automation*, volumen 1, páginas 15–18, 2005. [11](#)
- [41] Atlas, M.A.: *Optimización del tendido de un ducto aplicando inteligencia de enjambre*, 2009. [12](#)
- [42] Cruz-Chavez, M.A., A. Martinez-Oropeza y S.A. Serna-Barquera: *Neighborhood Hybrid Structure for Discrete Optimization Problems*. En *2010 IEEE Electronics, Robotics and Automotive Mechanics Conference*, páginas 108–113, 2010. [12](#), [70](#)

- [43] Cruz-Chavez, M.A., J.C. Martinez-Oropeza, A. and Peralta-Abarca, M. Cruz-Rosales y M. Martinez-Rangel: *Variable Neighborhood Search for Non-deterministic Problems*. En Rutkowski, L., M. Korytkowski, R. Scherer, R. Tadeusiewicz, L.A. Zadeh y J.M. Zurada (editores): *Artificial Intelligence and Soft Computing*, páginas 468–478. Springer International Publishing, 2014, ISBN 978-3-319-07176-3. 12
- [44] Logística, PEMEX: *Transporte por ducto*. <http://www.pemex.com/nuestro-negocio/logistica/ductos/Paginas/default.aspx>, 2018. 16
- [45] Leyva-Trujillo, M. y Salazar Arguello S.I.: *Emisiones, fugas y derrames en el transporte de hidrocarburos por ductos*, 2017. 16
- [46] Godoy-Simón, A.: *Control de la corrosión exterior para redes de ductos enterrados asistidos por SIG y software especializado*. Tesis de doctorado, Universidad Autónoma del Estado de Morelos, 2011. 17
- [47] Energía, Secretaria de: *Prospectivas del sector energético*. <https://www.gob.mx/sener/documentos/prospectivas-del-sector-energetico>, 2018. 17, 18
- [48] CENEGAS: *Centro Nacional de Control de Gas Natural*. <https://www.gob.mx/cenagas>, 2018. 18
- [49] Olaya, V.: *Sistemas de Información Geográfica*. <http://volaya.github.io/libro-sig/>, 2011. 21, 23, 24, 35
- [50] ESRI: *Corredor*. <http://desktop.arcgis.com/es/arcmap/10.3/tools/spatial-analyst-toolbox/corridor.htm>. 25
- [51] Medrano, A.F.: *Corridor Location: Generating Competitive and Efficient Route Alternatives*. PhD. Thesis, University of California, 2014. 25, 28, 34
- [52] Nussbaum, M.: *Pipeline Route Selection*, 2012. https://www.woodgroup.com/__data/assets/pdf_file/0019/4465/2012-01_Pipeline-Route-Selection-Right-of-Way_Mustang.pdf, Last retrieval at June 19, 2018. 26
- [53] GDAL/OGR contributors: *GDAL/OGR Geospatial Data Abstraction software Library*. Open Source Geospatial Foundation, 2018. 28
- [54] Hartmanis, J. y J.E. Hopcroft: *An Overview of the Theory of Computational Complexity*. Journal of the ACM, 18:444–475, Julio 1971. 29

- [55] Sipser, M.: *Introduction to the Theory of Computation*. Course Technology, second edición, 2006, ISBN 9787111173274. 29
- [56] Martínez-Oropeza, A.: *Algoritmo genético cooperativo paralelizado en ambiente grid para el problema de ruteo vehicular con ventanas de tiempo*. Tesis de Doctorado, UAEM, 2015. 29, 31
- [57] Gómez-Fuentes, M.C.: *Introducción al Análisis y al Diseño de Algoritmos*. Publidisa Mexicana, México, 1ª edición, 2014, ISBN 978-607-28-0225-4. 30, 45
- [58] Coello, C.A.: *Introducción a la Computación Evolutiva (Notas de curso)*. <http://delta.cs.cinvestav.mx/~ccoello/compevol/apuntes.pdf>, 2016. 32, 33
- [59] Garey, M.R. y Johnson D.S.: *Computers and Intractability: A Guide to the Theory of NP-Completeness*. Freeman, San Francisco, CA, USA, 1979. 32, 36, 37
- [60] Papadimitriou, C.H. y Steiglitz K.: *Combinatorial Optimization: Algorithms and Complexity*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1982, ISBN 0-13-152462-3. 32, 71
- [61] Menger, K.: *Some Applications of Point-Set Methods*. Annals of Mathematics, 32(4):739–760, 1931, ISSN 0003486X. 34
- [62] Karp, R.M.: *Reducibility among Combinatorial Problems*, páginas 85–103. Springer US, Boston, MA, 1972, ISBN 978-1-4684-2001-2. 34
- [63] Kruskal, J.B.: *On the Shortest Spanning Subtree of a Graph and the Traveling Salesman Problem*. Volumen 7, páginas 48–50, 1956. 35
- [64] Prim, R.C.: *Shortest connection networks and some generalizations*. The Bell System Technical Journal, 36(6):1389–1401, 1957, ISSN 0005-8580. 35
- [65] Rajesh, R., Rakesh K.K., Shazad A.M. y Manjula D.: *An Optimization Algorithm for Finding Graph Circuits and Shortest Cyclic Paths in Directed / Undirected Graphs*. International Journal of Scientific & Engineering Research, 5(1), 2014, ISSN 2229-5518. 36
- [66] Diestel, R.: *Graph Theory*. Springer, Berlin, 2ª edición, February 2000. 36

-
- [67] Itai, A., Papadimitriou C.H. y Szwarcfiter J.L.: *Hamilton Paths in Grid Graphs*. SIAM Journal on Computing, 11:676–686, 1982. [36](#)
- [68] ShaoDong, C., Hong S. y Rodney T.: *An efficient algorithm for constructing Hamiltonian paths in meshes*. Parallel Computing, 28(9):1293 – 1305, 2002, ISSN 0167-8191. [36](#)
- [69] Keshavarz-Kohjerdi, F., Bagheri A. y Asgharian Sardroud A.: *A linear-time algorithm for the longest path problem in rectangular grid graphs*. Discrete Applied Mathematics, 160(3):210 – 217, 2012, ISSN 0166-218X. [37](#)
- [70] Salazar, J.J.: *Programación Matemática*. Ed. Diaz de Santos, 2001, ISBN 84-7978-504-7. [37](#)
- [71] Nesmachnow, S.: *Algoritmos genéticos paralelos y su aplicación al diseño de redes de comunicaciones confiables*. Tesis de Licenciatura, Universidad de la República, Uruguay, 2004. [38](#)
- [72] Coello-Coello, C.A., Van Veldhuizen D.A. y Lamont G.B.: *Evolutionary Algorithms for Solving Multi-Objective Problems*. Springer US, 2ª edición, 2007, ISBN 978-0-387-36797-2. [43](#)
- [73] Reeves, C. (editor): *Modern Heuristic Techniques for Combinatorial Problems*. John Wiley & Sons, Inc., New York, NY, USA, 1993, ISBN 0-470-22079-1. [43](#)
- [74] Lourenço, H.R., Martin O.C. y Stützle T.: *Iterated Local Search*. En Glover, F. y Kochenberger G.A. (editores): *Handbook of Metaheuristics*, páginas 320–353. Springer US, Boston, MA, 2003, ISBN 978-0-306-48056-0. [44](#)
- [75] Massobrio, R. y Fagúndez de los Reyes G.: *Optimización de viajes compartidos en taxis utilizando algoritmos evolutivos*, 2015. [45](#)
- [76] Talbi, E.: *Metaheuristics: From Design to Implementation*. Wiley Publishing, 2009, ISBN 0470278587, 9780470278581. [45](#), [46](#), [48](#)
- [77] Basu, S.K.: *Design Methods and Analysis of Algorithms*. PHI Learning, 2ª edición, 2013, ISBN 978-81-203-4746-5. [45](#)
- [78] Glover, F.: *Future paths for integer programming and links to artificial intelligence*. Computers and Operations Research, 13(5):533 – 549, 1986, ISSN 0305-0548. [46](#), [47](#)
- [79] Kirkpatrick, S., Gelatt C.D. y Vecchi M.P.: *Optimization by Simulated Annealing*. Science, 220(4598):671–680, 1983, ISSN 0036-8075. [47](#), [53](#)

- [80] Nesmachnow, S.: *An Overview of Metaheuristics: Accurate and Efficient Methods for Optimisation*. Int. J. Metaheuristics, 3(4):320–347, 2014, ISSN 1755-2176. [47](#)
- [81] Goldberg, D.E.: *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1ª edición, 1989, ISBN 0201157675. [49](#)
- [82] Yang, X.: *Engineering Optimization: An Introduction with Metaheuristic Applications*. Wiley Publishing, 1ª edición, 2010, ISBN 0470582464, 9780470582466. [51](#)
- [83] Geman, S. y Geman D.: *Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images*. IEEE Trans. Pattern Anal. Mach. Intell., 6(6):721–741, Noviembre 1984, ISSN 0162-8828. [53](#)
- [84] Metropolis, N., Rosenbluth A., Marshall N., Teller A. y Teller E.: *Equation of State Calculations by Fast Computing Machines*. The Journal of Chemical Physics, 21(6):1087–1092, 1953. [54](#), [55](#)
- [85] Laarhoven, P.J.M. y Aarts E.H. (editores): *Simulated Annealing: Theory and Applications*. Springer Netherlands, 1987, ISBN 978-94-015-7744-1. [55](#)
- [86] RAE. <https://dle.rae.es>. [60](#)
- [87] ArcGIS-ESRI. <http://desktop.arcgis.com/es/arcmap/10.3/tools/spatial-analyst-toolbox/creating-a-cost-surface-raster.htm>. [64](#)
- [88] Moore, E.F.: *The shortest path through a maze*. página 285–292. In Proceedings of the International Symposium on the Theory of Switching, 1959. [65](#)
- [89] Lee, C.Y.: *An algorithm for path connection and its applications*. IRE Transactions on Electronic Computers, 10(3):346–365, Julio 1961. [65](#)
- [90] Chang, W.: *Computer Science and Artificial Intelligence*. World Scientific, 2017. [65](#)
- [91] Martínez, A.: *Solución al Problema de Máquinas en Paralelo Relacionadas mediante un Algoritmo de Colonia de hormigas*, 2010. [71](#)
- [92] Joyanes-Aguilar, L.: *Algoritmos y estructuras de datos : una perspectiva en C*. McGraw-Hill, 2004, ISBN 9788448140779. [85](#), [88](#), [135](#)

- [93] Geografía, Instituto Nacional de Estadística y. <https://www.inegi.org.mx>. 90
- [94] PEMEX. http://www.pemex.com/saladeprensa/boletines_nacionales/Paginas/2017-093-nacional.aspx. 91
- [95] Datasets, Terrain. <http://www.esri.com/news/arcuser/0311/terrain-datasets.html>. 94
- [96] <http://sinat.semarnat.gob.mx/dgiraDocs/documentos/ver/estudios/2006/30VE2006X0016.pdf>. 96, 104
- [97] Gilberg, R.F. y Forouzan B.A.: *Data Structures: A Pseudocode Approach with C*. Cengage Learning, 2^a edición, 2004, ISBN 9781285225036. 135, 137

Apéndice A

Código fuente del algoritmo ávido

Código fuente del algoritmo ávido (BFS) con una estructura cola de prioridad para generar la primera solución.

```
int BFSS(int **mat, Point src, Point dest, int ROW, int COL, \
        int scale, int minX, int minY, int **mdet, int hmax, \
        float **mcost, float percent, int N, int & cost, vector<int> & init
        vector<int> & init_solY, vector<int> & init_solMDE, vector<int> & in
        vector<int> & init_solYMDE)
{
    if (!mat[src.x][src.y] || !mat[dest.x][dest.y])
    {
        return INT_MAX;
    }
    bool visited[ROW][COL];
    memset(visited, false, sizeof visited);

    visited[src.x][src.y] = true;
    queue<vector<queueNode>> q;
    vector<queueNode> path;
    queueNode s = {src, 0};
    path.push_back(s);
    q.push(path);
    while (!q.empty())
    {
        path = q.front();
        q.pop();
        queueNode last = path[path.size() - 1];
        if (last.pt.x == dest.x && last.pt.y == dest.y)
        {
```

```

    savePrintPath(path, scale, minX, minY, init_solX, init_solY);
}
for (int i = 0; i < N; i++)
{
    int row = last.pt.x + rowNum8[i];
    int col = last.pt.y + colNum8[i];
    if (isValid(row, col, ROW, COL) && mat[row][col]==1 && \
        !visited[row][col] && (mdet[row][col] >= 0 && mdet[row][c
        && (mcostt[row][col] >= 0.0 && mcostt[row][col] <= percen
    {
        visited[row][col] = true;
        vector<queueNode> newpath(path);
        queueNode Adjcell = { {row, col}, last.dist + 1 };
        newpath.push_back(Adjcell);
        q.push(newpath);
    } //fin if
} //fin for

} //fin while
return path.size();
}

```

Apéndice B

Código fuente del algoritmo RS-PDT

Código fuente del algoritmo de Recocido Simulado para optimizar la solución del problema PDT:

```
//T0 es la temperatura inicial
//tf es el criterio de paro
//beta es el decremento de la temperatura
//generar semilla
std::random_device rd;
std::mt19937 gen(rd());
// s = curr_sol = init_sol
curr_solX = init_solX;
curr_solY = init_solY;
for(k = T0; k > tf; k*=beta)// Bucle para RS
{
    for(l=0; l <N;l++)//Bucle para Metropolis (Markov-Chain Length)
    {
        rnd1 = dis(gen); //distrib. integer (0, curr_solX.size
        rnd2 = dis(gen); //distrib. integer (0, curr_solY.size
        validarPuntos(rnd1, rnd2);
        //s' = trial_sol
        perturbar(curr_solX, curr_solY, trial_solX, trial_solY, r
        alfa = disf(gen); //distrib. float(0,1)
        delta = trial_solX.size() - curr_solX.size();
        aceptacion = std::exp(-((double)delta/k)); //k = T0
        if (trial_solX.size() <= curr_solX.size()) // si s' < s
        {
            curr_solX = trial_solX;
            curr_solY = trial_solY;
        }else if ((alfa <= aceptacion) || (delta <= 0)) //sino, s
```

```
    {
      curr_solX = trial_solX;
      curr_solY = trial_solY;
    }
  }
```

Apéndice C

Eficiencia algorítmica

Generalmente, se hace referencia a la eficiencia de un algoritmo en función del número de elementos a ser procesados. Los conceptos básicos se mencionan a continuación[92] [97]:

Bucles iterativos lineales: en un bucle lineal, los incrementos o decrementos de cada iteración se realiza a través de una operación de suma o resta, según sea el caso.

```
for (i=0; i < 1000; i++) { sentencias; }
```

El código anterior ejecuta 1000 veces las sentencias internas. El número de iteraciones es directamente proporcional al factor del bucle. Por este motivo, la eficiencia del bucle está determinada por una complejidad $f(n) = n$.

```
for (i=0; i < 1000; i+=2) { sentencias; }
```

El ejemplo de arriba, el bucle tiene un incremento de 2 unidades, por lo tanto la eficiencia del bucle es proporcional a la mitad del factor determinada por $f(n) = \frac{n}{2}$.

Bucles logarítmicos: en un bucle logarítmico, la variable de control de las iteraciones es multiplicada o dividida en cada iteración.

```
for (i=0; i < 1000; i*=2) { sentencias; }
```

```
for (i=0; i < 1000; i/=2) { sentencias; }
```

En el ejemplo de código de arriba, el número de iteraciones que se realizarán es de 10 en cada uno de los ejemplos. Generalizando el análisis, el bucle que multiplica o divide esta determinado por una complejidad $f(n) = \log n$.

Bucles anidados: un bucle que contiene a otro bucle son conocidos como bucles anidados. El total de iteraciones que realizan dos bucles anidados esta determinado por el producto del numero de iteraciones del bucle interno y el numero de iteraciones del bucle externo.

```
for (i = 0; i < 1000; i++)
{
    for (j = 0; j < 1000; j++)
    {
        sentencias;
    }
}
```

el ejemplo anterior muestra que en número de iteraciones del bucle interno y externo se determina por el producto 1000 x 1000, por tal motivo la complejidad se expresa como $f(n) = n^2$

```
for (i=0; i < 10; i++)
{
    for (j=0; j < 10; j*=2)
    {
        sentencias;
    }
}
```

En el ejemplo de código anterior se aprecia que el bucle interno realizará \log_{10} iteraciones, mientras que el bucle externo realizar la n iteraciones, la cual tiene un valor de 10. La ecuación que determina el numero de iteraciones es $f(n) = 10 \cdot \log_{10}$ por lo tanto la complejidad se expresa como $f(n) = n \cdot \log n$.

```
for (i=0; i < 10; i++)
{
    for (j=0; j < i; j++)
    {
        sentencias;
    }
}
```

En el ejemplo anterior se aprecia que el número de iteraciones del bucle interno depende del valor de la variable de control del bucle externo. El número de iteraciones en el bucle interno se calcula de la siguiente forma: $1 + 2 + 3 + \dots + 9 + 10 = 55$. Si se calcula el promedio se obtiene que $\frac{55}{10} = 5.5$, que equivale a multiplicar y dividir $\frac{10 \cdot 1}{2}$. Matemáticamente esta generalización se determina por $f(n) = \frac{n+1}{2}$. Multiplicando el número de iteraciones del bucle externo, la dependencia cuadrática tiene una complejidad $f(n) = n \frac{n+1}{2}$.

La tabla C.1 muestra en orden de decremento de la eficiencia las medidas de eficiencia estándar[97].

Tabla C.1: Medidas de eficiencia estándar

Eficiencia	big-O	Iteraciones	Tiempo estimado
Logarítmica	$O(\log n)$	14	microsegundos
Lineal	$O(n)$	10,000	segundos
Lineal logarítmica	$O(n(\log n))$	140,000	segundos
Cuadrática	$O(n^2)$	10,000 ²	minutos
Polinomial	$O(n^k)$	10,000 ^k	horas
Exponencial	$O(c^n)$	2 ^{10,000}	intratable
Factorial	$O(n!)$	10,000!	intratable

Apéndice D

Productos obtenidos entre 2013 y 2019



Figura D.1: Ponencia en Jornadas Académicas



**CENTRO DE INVESTIGACIÓN Y DE ESTUDIOS AVANZADOS
DEL INSTITUTO POLITÉCNICO NACIONAL**

México, D.F., a 20 de septiembre de 2013

A quien corresponda:

Por medio de la presente, hago constar que el C. **Pedro Moreno Bernal** con número de matrícula **1720100102**, quien actualmente cursa estudios de posgrado en el Centro de Investigación en Ingeniería y Ciencias Aplicadas (CIICAp) de la Universidad Autónoma del Estado de Morelos, ha realizado una estancia de investigación en el Departamento de Computación del Centro de Investigación y de Estudios Avanzados del Instituto Politécnico Nacional (CINVESTAV-IPN), unidad Zacatenco, del **20 de mayo al 20 de septiembre** del año en curso.

Durante su estancia en el CINVESTAV-IPN, Pedro estará trabajando con miembros de mi grupo de investigación en computación evolutiva (particularmente, en aspectos relacionados con la paralelización de algoritmos bio-inspirados), bajo mi supervisión.

Sin más qué agregar, me pongo a sus órdenes, para cualquier aclaración relacionada con este asunto.

Atentamente,

Dr. Carlos A. Coello Coello
Investigador Cinvestav 3F
CINVESTAV-IPN
Departamento de Computación
Av. IPN No. 2508
Col. San Pedro Zacatenco
México, D.F. 07360
Tel. (55) 5747 3800 x 6564
Fax (55) 5747 3757
email: ccoello@cs.cinvestav.mx

Av. Instituto Politécnico Nacional # 2508 Col. San Pedro Zacatenco México, D.F. C.P. 07360
Tel.: 5747-3800 Fax: 5747-7002

Figura D.2: Estancia de investigación en el CINVESTAV

NUMERO DE PROYECTO: 199685

EMPRESA BENEFICIADA: Corrosión y Protección S.A. de C.V.

TÍTULO DEL PROYECTO: *Desarrollo de un sistema para la valoración de trayectorias y viabilidades de sistemas de ductos para líquidos valiosos*

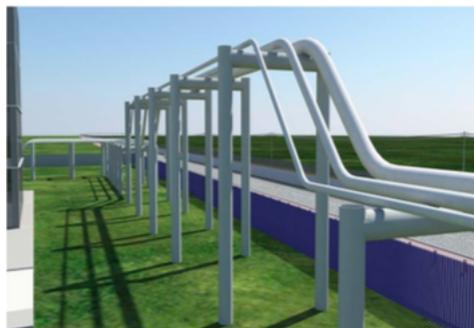


Figura D.3: Participación en proyecto PROINNOVA-CONACYT

RESULTADOS DEL PROYECTO:

- Documento que describe el modelo matemático desarrollado, que puede predecir rutas económicas de distribución en tuberías o ductos y una metodología de aplicación del modelo.
- Un modelo matemático compuesto de un modelo de optimización y un modelo de satisfacción de restricciones, que puede predecir una de las mejores rutas económica de distribución en tuberías o ductos, para grandes urbes o establecimientos de gente. Y la metodología de aplicación del modelo.
- Formación de dos doctores en Ingeniería y Ciencias Aplicadas, en un posgrado PNP:
- M.I. Pedro Moreno Bernal y M.C. Ariadna Ortiz Huerta

IMPACTOS DEL PROYECTO:

Los impactos del proyecto son del tipo:

- Económico. Al optimizar costos en instalación y mantenimiento de ductos.
- Tecnológico. Dar un mejor servicio a los clientes ya existentes así como potencializar el uso de los ductos o tubería para el transporte de líquidos vitales. De la misma manera pueden emplearse mejores condiciones de mantenimiento y mejores instalaciones de protección catódica, evitando deterioro por corrosión de los mismos ductos.
- Científico. Al contar con una estructura de experimentos en planta para emular los procesos que retroalimenten al modelo matemático y extraiga el conocimiento a perspectivas de transferencia de tecnología.
- Ambiental. Al reducir su riesgo de falla mediante detección de áreas que requieran mayor protección catódica así como el aumento de la integridad del ducto y no derramar al medio su contenido o líquidos vitales.

Figura D.4: Participación en proyecto PROINNOVA-CONACYT



UNIVERSIDAD
DE LA REPÚBLICA
URUGUAY



Montevideo, Uruguay, 5 abril de 2017

A quien corresponda:

Esta nota certifica que **Pedro Moreno Bernal**, quien cursa estudios de posgrado en el Centro de Investigación en Ingeniería y Ciencias Aplicadas (CIICAp) de la Universidad Autónoma del Estado de Morelos con número de matrícula **1720100102**, ha realizado una estancia de investigación en el Instituto de Computación de la Facultad de Ingeniería, de la Universidad de la República, Uruguay, del **6 de marzo al 5 de abril** del año en curso.

Durante su estancia de investigación, Pedro Moreno, estuvo trabajando en el grupo de investigación Centro del Cálculo en las temáticas de computación de alto desempeño y optimización utilizando algoritmos evolutivos.

Quedo a sus órdenes para cualquier aclaración relacionada con la estancia de investigación de Pedro Moreno.

Atentamente

Dr. Sergio Neschachnow
Profesor Titular
Investigador Honorario, Grado 4, PEDECIBA Informática
Investigador Activo Nivel II, ANII
Universidad de la República
Herrera y Reissig 565, Montevideo, Uruguay
Teléfono: 27114244 Int. 1052
E-mail: sergion@fing.edu.uy

UNIVERSIDAD DE LA REPÚBLICA • FACULTAD DE INGENIERÍA - Julio Herrera y Reissig 565 - C.P. 11.300 Tels.: (598) 2711 06 98
- 2711 07 98 - 2711 08 98 - Fax: 2711 54 56 - INTERNET: www.fing.edu.uy - MONTEVIDEO - URUGUAY

Figura D.5: Estancia de investigación en la Universidad de la República, Uruguay

Interfaz Gráfica para Extraer Coordenadas xy de Datos Vectoriales en Formato Shapefile

Graphical User Interface to extract xy coordinates from vector shapefiles

¹Marco Antonio Fuentes-Jiménez, ²Pedro Moreno, ²Alfonso D'Granda-Trejo, ²Jesús del Carmen Peralta-Abarca, ²Juana Enríquez-Urbano

¹FCAel-UAEM, ²Postgrado en Ingeniería y Ciencias Aplicadas CIICAp-UAEM
Av. Universidad 1001 Col. Chamilpa, Cuernavaca, Morelos, México. C.P. 62209

E-mail: markfuentes2991@gmail.com, pmoreno@uaem.mx, dgranda@uaem.mx, juana.enriquez@uaem.mx, carmen.peralta@gmail.com

PALABRAS CLAVE:

SIG, shapefile, ASCII (text),
GDAL, VectorMapConverter,
VectorMapView

RESUMEN

El tratamiento de información geográfica a partir de un conjunto de valores, para su análisis numérico, es un problema importante, debido a que la mayoría de las herramientas de Sistemas de Información Geográfica, tratan la información en formatos de imagen digital. En este trabajo presentamos una Interfaz Gráfica de Usuario, que a partir de una metodología propuesta, extrae coordenadas xy de los vértices de datos vectoriales en formato shapefile, a un archivo de texto ASCII. Así mismo, permite visualizar los datos espaciales extraídos.

Figura D.6: Artículo en revista "Programación matemática"

inventio

La génesis de la cultura universitaria en Morelos

*PENSAMIENTO UNIVERSITARIO
CIENCIA Y TECNOLOGÍA
CRÍTICA Y ARTIFICIOS
SIGNIFICAR CON TEXTOS
GALERÍA / ELISA CANO
POESÍA / ÁNGEL CUEVAS*

UNIVERSIDAD AUTÓNOMA DEL ESTADO DE MORELOS

ISSN: 2007-1760 (impreso)
ISSN: 2448-9026 (digital)
Año 14 Núm 34
noviembre 2018 - febrero 2019



Figura D.7: Artículo de divulgación en revista Inventio-UAEM

inventio

La *inventio* es la argumentación racional de un tema [*Inventio*, del latín *inveniere*, encontrar el tema del que se va a hablar].

Inventio, la génesis de la cultura universitaria en Morelos

Año 14 • Número 34

noviembre 2018 - febrero 2019

Editorial	
1	La UAEM, una de las mejores universidades públicas estatales del país José Mario Ordóñez Palacios
Pensamiento universitario	
5	El equinoccio en la zona arqueológica de Teopanzolco Francisco Granados
Ciencia y tecnología	
17	Percepción de la imagen corporal en adolescentes de secundaria en Huitzilac Pablo Fernando Aparicio García Paola Adanari Ortega Ceballos María Alejandra Terrazas Meraz
25	Metaheurísticas Jesús del Carmen Peralta-Abarca Pedro Moreno-Bernal Sergio Nesmachnow
Crítica y artificios	
45	El cambio de identidad sexo-genérica Joan Vendrell Ferré
57	Producción de arte digital en México desde un entorno complejo Cynthia Patricia Villagómez Oviedo
Galería	
65	Elisa Cano, obra plástica
Poesía	
69	Frutos de sal Ángel Cuevas
Significar con textos	

Figura D.8: Artículo de divulgación en revista Inventio-UAEM