



U N I V E R S I D A D
Panamericana

Alumno:

Diego Issac Cano Vizcaino
Atzin Eduardo Cruz Briones
Max Georges Sainte Guzmán

Rutas de transporte

Materia: Optimización y metaheurística

Maestro: Dr. Jonás Velasco Álvarez

Optimización de Rutas de Transporte con Algoritmos Metaheurísticos

Introducción

La eficiencia en la planificación de rutas de transporte constituye un tema de creciente importancia para diversas organizaciones, particularmente aquellas que buscan maximizar el aprovechamiento de sus recursos al atender un conjunto predefinido de paradas y destinos. El objetivo central es optimizar la logística del transporte, minimizando tanto los costos como los tiempos de desplazamiento, mientras se asegura el cumplimiento de restricciones operativas, tales como la capacidad de los vehículos. En este informe se presenta un enfoque para resolver un problema de optimización de rutas mediante tres técnicas metaheurísticas: Búsqueda Tabú (Tabu Search), Búsqueda Aleatoria Localizada (BAL), y Búsqueda Aleatoria Simple (BAS). Cada uno de estos algoritmos se evalúa con respecto a su eficiencia, precisión y capacidad para determinar la mejor ruta de transporte bajo distintas condiciones operativas.

La optimización de rutas de transporte es un problema de relevancia no solo por su impacto en la reducción de costos operativos, sino también por su influencia directa en la calidad del servicio y la disminución del impacto ambiental. Al minimizar los tiempos de recorrido y el número de recursos empleados, se contribuye significativamente a un sistema de transporte más eficiente y sostenible. Por ende, la selección e implementación de algoritmos de optimización resulta fundamental para alcanzar estos objetivos y abordar las restricciones inherentes a este tipo de problemas logísticos.

La importancia de resolver este tipo de problemas radica también en la naturaleza dinámica de los sistemas de transporte. Factores como la variabilidad de la demanda, las condiciones cambiantes del tráfico, y las restricciones de infraestructura imponen retos adicionales que requieren el uso de técnicas sofisticadas para ser abordados con éxito. La capacidad de adaptarse a estas fluctuaciones en tiempo real y de encontrar rutas óptimas o casi óptimas resulta esencial para garantizar la eficiencia del sistema.

El Problema

El problema específico consiste en organizar de manera eficiente las rutas de un autobús para un conjunto de paradas y destinos, garantizando que la capacidad del vehículo no exceda las 20 personas por viaje. Además, se establece que cada viaje debe ser realizado por un único autobús, y se consideraron dos escenarios posibles: la optimización del recorrido entre paradas y la optimización del recorrido desde las paradas hacia la planta de destino. Cada uno de estos escenarios plantea retos específicos, tales como la minimización de las distancias

recorridas, la reducción de los tiempos de espera y la adaptación a fluctuaciones inesperadas en la demanda.

La búsqueda de una solución óptima para este problema requiere una evaluación detallada de costos y eficiencia, dadas las restricciones impuestas por la capacidad del vehículo. Con este fin, se exploraron distintas estrategias de optimización que incluyen técnicas metaheurísticas como la Búsqueda Tabú, y variantes como BAL y BAS. Estos problemas de optimización son notoriamente difíciles debido a la complejidad combinatoria, la presencia de múltiples óptimos locales y la influencia de factores como la cantidad de paradas y su distribución geográfica.

Además, el problema puede extenderse al considerar múltiples autobuses y varios puntos de destino, lo cual añade una capa adicional de complejidad y hace necesario el uso de estrategias de optimización más robustas. El análisis de estos problemas debe tomar en cuenta no solo la eficiencia individual de cada viaje, sino también la eficiencia global del sistema de transporte. La coordinación de múltiples vehículos, la asignación de recursos y la planificación simultánea de rutas son aspectos que se tornan críticos para garantizar un sistema verdaderamente óptimo.

Descripción de los Algoritmos

Búsqueda Tabú

La Búsqueda Tabú es un algoritmo de búsqueda heurística diseñado para explorar el espacio de soluciones más allá de los máximos locales mediante el uso de una memoria adaptativa que evita la revisión de soluciones previamente visitadas. En este caso, se definió una función de costo que cuantifica la distancia total recorrida por el autobús, y se realizaron múltiples iteraciones para mejorar la solución inicial. Este enfoque evita caer en soluciones subóptimas, ya que la lista tabú almacena soluciones anteriores para garantizar una exploración exhaustiva del espacio de búsqueda, impidiendo ciclos y asegurando una diversificación controlada.

La estrategia de Búsqueda Tabú se basa en definir un entorno o vecindad de soluciones a partir de la solución actual. A través de la exploración de esta vecindad, se identifica la mejor solución posible, la cual pasa a ser la nueva solución de partida. Este proceso continúa durante un número predefinido de iteraciones, con el fin de alcanzar una mejora global. La lista tabú juega un papel clave al impedir que el algoritmo regrese a soluciones ya exploradas, evitando así los ciclos repetitivos que podrían limitar el progreso.

Ejemplo de código:

```
# Inicializar solución
solucion_inicial = generar_solucion_inicial()
costo_mejor = calcular_costo(solucion_inicial)
```

```
# Iteraciones de búsqueda Tabú
for iteracion in range(100):
    vecinos = generar_vecinos(solucion_inicial)
    mejor_vecino = seleccionar_mejor(vecinos)
    if calcular_costo(mejor_vecino) < costo_mejor:
        solucion_inicial = mejor_vecino
        costo_mejor = calcular_costo(mejor_vecino)
```

La Búsqueda Tabú es particularmente útil en problemas con múltiples máximos locales, ya que su estrategia de memoria le permite evitar estancarse en estas soluciones subóptimas y continuar explorando nuevas áreas del espacio de soluciones. No obstante, el costo computacional de mantener y actualizar la lista tabú puede incrementar significativamente la complejidad temporal del algoritmo en comparación con otras técnicas más simples.

Otra característica importante de la Búsqueda Tabú es la flexibilidad para incorporar criterios de aspiración, que permiten aceptar una solución prohibida si cumple ciertos requisitos, como ser mejor que la mejor solución encontrada hasta el momento. Esta característica permite al algoritmo escapar de áreas de baja calidad del espacio de búsqueda y favorece la diversificación, explorando regiones menos exploradas.

Búsqueda Aleatoria Simple (BAS)

La Búsqueda Aleatoria Simple (BAS) es un método basado en la generación de soluciones aleatorias y la selección de la mejor entre ellas. Aunque BAS es un enfoque conceptualmente simple, puede requerir un gran número de iteraciones para obtener una solución cercana al óptimo. Esta falta de garantía de encontrar una solución óptima se ve compensada por la rapidez de su implementación y la ausencia de requerimientos complejos de ajuste.

BAS se puede implementar de manera eficiente en sistemas donde el tiempo de desarrollo es limitado y se necesita una solución rápidamente. La simplicidad de BAS radica en la generación de posibles soluciones sin un proceso estructurado de mejora iterativa. Aunque este enfoque puede no garantizar una solución óptima, puede ser útil para establecer un punto de referencia inicial o para proporcionar una solución rápida que luego pueda ser mejorada por otros algoritmos más complejos.

Ejemplo de código:

```
# Generar soluciones aleatorias
mejor_solucion = None
costo_mejor = float('inf')
for iteracion in range(100):
    solucion = generar_solucion_aleatoria()
    costo_actual = calcular_costo(solucion)
```

```
if costo_actual < costo_mejor:
    mejor_solucion = solucion
    costo_mejor = costo_actual
```

Una de las principales ventajas de BAS es la rapidez con la que puede ser implementada, ya que no requiere una parametrización exhaustiva ni ajustes finos. Sin embargo, debido a la naturaleza inherentemente aleatoria del proceso, la eficiencia del algoritmo en términos de convergencia hacia una solución óptima es limitada, implicando un elevado costo computacional en términos de tiempo si se necesitan soluciones de alta calidad.

En la práctica, BAS puede servir como un componente de estrategias más amplias, donde actúa como una fase de exploración inicial. Esta fase puede ayudar a definir áreas prometedoras del espacio de búsqueda que luego pueden ser exploradas con mayor detalle mediante otros algoritmos más sofisticados.

Búsqueda Aleatoria Localizada (BAL)

La Búsqueda Aleatoria Localizada (BAL) mejora la estrategia de búsqueda aleatoria mediante la restricción del área de búsqueda a una vecindad próxima a la mejor solución conocida. Este enfoque permite que el algoritmo sea más eficiente al centrarse en una región del espacio de búsqueda que ya ha mostrado potencial, incrementando así la probabilidad de encontrar una solución óptima local de manera más rápida.

BAL combina la simplicidad de BAS con un enfoque más dirigido que permite mejorar la eficiencia sin requerir la complejidad de métodos como la Búsqueda Tabú. BAL se basa en generar nuevas soluciones dentro de un radio de proximidad en torno a la mejor solución actual, lo que limita la búsqueda a una región específica y aumenta la probabilidad de encontrar mejoras incrementales.

Ejemplo de código:

```
# Búsqueda localizada
mejor_solucion = generar_solucion_inicial()
costo_mejor = calcular_costo(mejor_solucion)
for iteracion in range(100):
    solucion_vecina = generar_vecino_cercano(mejor_solucion)
    costo_actual = calcular_costo(solucion_vecina)
    if costo_actual < costo_mejor:
        mejor_solucion = solucion_vecina
        costo_mejor = costo_actual
```

BAL resulta especialmente útil en situaciones donde se ha identificado una solución prometedora y se busca mejorarla ligeramente. Al limitar el área de búsqueda, se reducen los

tiempos de cálculo y se incrementa la eficiencia, permitiendo al algoritmo enfocarse en refinar la solución existente. Sin embargo, esta limitación también conlleva el riesgo de no escapar de óptimos locales, lo cual podría resultar en soluciones subóptimas en comparación con una búsqueda más global.

En algunos casos, BAL se puede combinar con estrategias de perturbación que permitan introducir cambios más amplios cuando el algoritmo se encuentra atrapado en un óptimo local. Estas perturbaciones permiten escapar de soluciones de baja calidad y explorar otras áreas del espacio de búsqueda que podrían tener un mayor potencial.

Conclusión de los Datos

Los resultados experimentales demostraron que la Búsqueda Tabú, si bien más precisa, presenta tiempos de cómputo considerablemente mayores en comparación con BAS y BAL. La Búsqueda Tabú logró identificar rutas óptimas con menores costos totales, pero su aplicabilidad está limitada a escenarios donde se dispone de suficiente capacidad computacional y tiempo de ejecución. Por otro lado, BAL y BAS se posicionaron como métodos más efectivos para aplicaciones donde la rapidez es crucial, aunque con una precisión moderadamente menor.

Las pruebas adicionales revelaron cómo cada enfoque se adapta a distintas necesidades operativas. En escenarios donde el tiempo de cálculo es un factor crítico, BAL y BAS constituyen alternativas viables debido a su habilidad para identificar soluciones aceptables en menor tiempo. Por el contrario, la Búsqueda Tabú es adecuada cuando se persigue la máxima precisión posible y se dispone de los recursos necesarios para realizar una búsqueda exhaustiva. En última instancia, la elección del algoritmo más adecuado depende de las restricciones y prioridades específicas del problema en cuestión.

Es importante señalar que la combinación de diferentes enfoques también puede proporcionar beneficios adicionales. Por ejemplo, se puede utilizar BAS para explorar rápidamente un conjunto amplio de soluciones y luego aplicar Búsqueda Tabú para mejorar la mejor solución encontrada. Este enfoque híbrido permite aprovechar las fortalezas de cada técnica y mitigar sus debilidades, logrando una solución más robusta y eficiente.

Conclusión General

El problema de optimización de rutas de transporte se beneficia considerablemente del uso de algoritmos metaheurísticos. La Búsqueda Tabú destaca por su capacidad de hallar soluciones más precisas a través de una exploración profunda del espacio de búsqueda, mientras que BAL y BAS presentan una alternativa valiosa para escenarios donde la eficiencia temporal es un requerimiento fundamental. La selección del enfoque óptimo depende de los objetivos y

restricciones inherentes a cada situación, debiendo balancear la necesidad de precisión con la disponibilidad de recursos computacionales y tiempo de cálculo.

La planificación óptima de rutas tiene un impacto directo no solo en la reducción de costos y la mejora en la satisfacción del cliente, sino también en la minimización del impacto ambiental asociado al uso excesivo de recursos. La implementación de técnicas como la Búsqueda Tabú, BAS y BAL permite adaptar la solución a las condiciones y restricciones particulares de cada problema. Mientras que la Búsqueda Tabú proporciona un enfoque más exhaustivo y detallado, los métodos aleatorios como BAS y BAL ofrecen una flexibilidad y rapidez superiores, adecuándose mejor a contextos donde la toma de decisiones rápida es crítica.

En resumen, cada uno de los algoritmos discutidos posee ventajas y desventajas que los hacen adecuados para distintos tipos de problemas de optimización. La capacidad de seleccionar el algoritmo correcto, o incluso combinar varios enfoques, permitirá a las organizaciones enfrentar los desafíos de planificación y optimización de manera más eficiente y efectiva, logrando no solo objetivos operativos, sino también económicos y ambientales en el largo plazo.

La capacidad de combinar métodos y adaptar las estrategias de búsqueda a las características particulares del problema es esencial en el contexto de la optimización de rutas de transporte. Cada técnica presentada tiene un conjunto de fortalezas que, cuando se integran de manera adecuada, permiten resolver problemas complejos con un alto grado de eficiencia. En el futuro, la incorporación de técnicas de aprendizaje automático para ajustar dinámicamente los parámetros de los algoritmos metaheurísticos podría representar un avance significativo, mejorando aún más la capacidad de optimización y adaptabilidad de estas soluciones.