

Capítulo 1

Resultados

En este capítulo se presentan los resultados que se obtienen al explorar las vecindades de reinserción de subrutas, intercambio de subrutas y reinserción de dos clientes, utilizando los modelos de optimización creados en los capítulos anteriores. Para ello se utilizan tres instancias del Problema de Enrutamiento de Vehículos con restricción de Capacidad (CVRP) generadas a partir de la metodología propuesta en [7].

En la sección 1.1 se muestran los resultados de los modelos aplicados al CVRP y se analizan en términos de calidad de las soluciones y tiempo computacional, en comparación con una búsqueda exhaustiva. En la sección 1.2 se presentan los beneficios de las estrategias implementadas para la generación y estructuración de datos, destacando su impacto en la exploración de vecindades, así como un análisis comparativo entre los modelos de optimización y una búsqueda simple basada en los datos generados, evaluando el desempeño relativo en diferentes escenarios.

1.1. Resultados de los modelos de optimización

Para evaluar el desempeño de los modelos de optimización, se llevaron a cabo experimentos utilizando una computadora con procesador Intel Core i5 de octava generación y 8 GB de memoria RAM. Las pruebas incluyeron tanto una búsqueda exhaustiva como la resolución de los modelos de optimización que se presentan en el capítulo ??.

La búsqueda exhaustiva se implementó en el lenguaje *Common Lisp*, aprovechando los trabajos previos desarrollados en la facultad [4, 2, 6, 1, 5]. Este enfoque permite evaluar todas las posibles soluciones del problema y establecer una base comparativa para los resultados obtenidos con los modelos de optimización.

Por otro lado los modelos matemáticos se diseñaron e implementaron en el lenguaje de programación C++, utilizando grafos dirigidos para representar las soluciones del VRP. Debido a la integración de la biblioteca GLPK (*GNU Linear Programming Kit*) con el lenguaje C++, las funciones *glp_add_cols*, *glp_set_obj_coef* y *glp_set_row_name*, nativas de GLPK se pudieron utilizar en C++, permitiendo la creación de las variables, la función objetivo y las restricciones del modelo.

Los problemas utilizados para los experimentos fueron generados aleatoriamente con diferentes cantidades de clientes: 35, 62 y 150 clientes. Las distancias entre los clientes fueron números aleatorios en el rango de 1 a 50, mientras que las demandas de los clientes se generaron aleatoriamente entre 1 y 20. La capacidad de los vehículos fue calculada según una fórmula previamente descrita en [7]. Se corrieron un total de cincuenta experimentos para cada algoritmo con cada uno de los grupos de clientes (35, 62 y 150 clientes).

A continuación, se muestran los resultados para los criterios de vecindad de reinserción de subrutas e intercambio de subrutas separados en dos epígrafes.

1.1.1. Resultados para la reinserción de subrutas

La tabla 1.1 muestra las medidas de tendencia central y medidas de dispersión para cincuenta experimentos realizados para cada tamaño del problema. Se incluyen los resultados de la exploración exhaustiva y del modelo de optimización presentado para la vecindad de reinserción de subrutas. Su desempeño se compara en términos de tiempo de ejecución de cada algoritmo medido en segundos.

Clientes	Algoritmo	Media	Desviación Estándar	Mediana
35	Modelo	0.06s	0.01s	0.05s
	Exploración exhaustiva	0.05s	0.01s	0.05s
62	Modelo	0.28s	0.01s	0.27s
	Exploración exhaustiva	0.35s	0.03s	0.34s
150	Modelo	2.16s	0.06s	2.14s
	Exploración exhaustiva	3.67s	0.22s	3.62s

Tabla 1.1: Comparación de los algoritmos en cuanto a tiempo de ejecución para la vecindad de reinserción de subruta para 35, 62 y 150 clientes.

Los resultados presentados en la tabla 1.1 permiten realizar una comparación entre la exploración exhaustiva y los modelos de optimización. A continuación se hace un análisis de la media, mediana y desviación estándar presentadas en la tabla 1.1:

- **Media:** Para 35 clientes, el modelo de optimización tuvo un tiempo promedio de 0.06 segundos, mientras que la exploración exhaustiva requirió 0.05 segundos. Esto sugiere que, aunque ambos métodos tienen un desempeño similar en problemas pequeños, el modelo de optimización es ligeramente más lento para la resolución de problemas con 35 clientes. Sin embargo, para problemas más grandes, como el de 150 clientes, el modelo (2.16 segundos) es significativamente más rápido que la exploración (3.67 segundos).
- **Mediana:** En todos los casos, la mediana del modelo y la exploración exhaustiva son muy similares a la media correspondiente, lo que indica que la mayoría de los tiempos se distribuyen alrededor de la media.
- **Desviación estándar:** Para 150 clientes, el modelo presenta una desviación estándar de 0.06, mientras que la exploración exhaustiva tiene una desviación estándar de 0.22. Esto significa que los tiempos del modelo son menos dispersos alrededor de la media.

Además del análisis de las medidas de tendencia central y dispersión, se realizó una prueba de normalidad para verificar si los tiempos de ejecución se distribuyen de manera normal. La prueba de normalidad utilizada fue la prueba de Shapiro-Wilk [3], la cual es particularmente adecuada para muestras pequeñas, como las que se presentan en este estudio. La hipótesis nula (H_0) indica que la muestra sigue una distribución normal, mientras que la hipótesis alternativa (H_a) plantea que si distribuyen de manera normal. El resultado de esta prueba de hipótesis fue que los tiempos de ejecución de ambos algoritmos no siguen una distribución normal, en la tabla 1.2 se muestran los resultados de la prueba de normalidad.

Clientes	Algoritmo	Estadístico	p-valor
35	Modelo	0.542	0.000
	Exploración exhaustiva	0.770	0.000
62	Modelo	0.895	0.006
	Exploración exhaustiva	0.918	0.024
150	Modelo	0.925	0.035
	Exploración exhaustiva	0.917	0.023

Tabla 1.2: Resultados de la prueba de normalidad Shapiro-Wilk para la reinserción de subrutas.

Debido a que no se puede rechazar la hipótesis nula, se asume que las distribuciones de los tiempos de cada algoritmo son normales y se realiza una prueba de

hipótesis utilizando el test de Wilcoxon para datos emparejados [3]. La hipótesis nula (H_0) establece que no hay diferencia en las medianas de los tiempos de ejecución de los algoritmos, mientras que la hipótesis alternativa (H_a) plantea que existe una diferencia en las medianas. La tabla 1.3 muestra los valores para esta prueba.

Clientes	Estadístico	p-valor
35	175.000	0.245
62	0.000	0.000
150	0.000	0.000

Tabla 1.3: Resultados de la prueba de Wilcoxon para la reinserción de subrutas.

Como se puede observar en la tabla 1.3, para instancias del problema con 62 y 150 clientes, se puede rechazar la hipótesis nula (H_0), lo cual indica que el modelo de optimización para problemas con una gran cantidad de clientes, es mas rápido que una exploración exhaustiva.

En el siguiente epígrafe se muestran los resultados para la vecindad de intercambio de subrutas.

1.1.2. Resultados para el intercambio de subrutas

La tabla 1.4 muestra las medidas de tendencia central y dispersión para los resultados obtenidos en la vecindad de intercambio de subrutas. Su desempeño se compara en términos de tiempo de ejecución medido en segundos.

Clientes	Algoritmo	Media	Desviación Estándar	Mediana
35	Modelo	0.14s	0.01s	0.13s
	Exploración exhaustiva	0.18s	0.02s	0.18s
62	Modelo	1.49s	0.12s	1.53s
	Exploración exhaustiva	2.69s	0.35s	2.76s
150	Modelo	16.74s	1.43s	16.46s
	Exploración exhaustiva	37.58s	3.54s	37.30s

Tabla 1.4: Resultados de *swap* de subruta para 35, 62 y 150 clientes.

Los resultados presentados en la tabla 1.4 permiten realizar una comparación entre la exploración exhaustiva y el modelo de optimización. A continuación se hace un análisis de la media, mediana y desviación estándar presentadas en la tabla 1.4:

- **Media:** Para problemas pequeños, como los de 35 clientes, el modelo tiene un tiempo promedio de 0.14 segundos, mientras que la exploración exhaustiva requiere 0.18 segundos. En problemas grandes, como el de 150 clientes, el modelo es mucho más eficiente, con una media de 16.74 segundos frente a 37.58 segundos para la exploración exhaustiva.
- **Mediana:** En problemas de mayor tamaño, la mediana también refleja que los modelos tienen un desempeño más rápido. Por ejemplo, para 150 clientes, la mediana es 16.46 segundos para los modelos y 37.30 segundos para la exploración exhaustiva.
- **Desviación estándar:** En problemas pequeños, la variabilidad en los tiempos es baja para ambos métodos. Sin embargo, en problemas grandes, como los de 150 clientes, la desviación estándar del modelo (1.43 segundos) es mucho menor que la de la exploración exhaustiva (3.54 segundos), lo que indica una menor dispersión en los datos obtenidos del modelo.

Para los datos obtenidos en la vecindad de intercambio de subrutas, también se les realiza una prueba de normalidad Shapiro-Wilk, con el objetivo de analizar si distribuyen de manera normal. La hipótesis nula (H_0) indica que la muestra sigue una distribución normal, mientras que la hipótesis alternativa (H_a) plantea que si distribuyen de manera normal. Los resultados se muestran en la tabla 1.5.

Clientes	Algoritmo	Estadístico	p-valor
35	Modelo	0.967	0.464
	Exploración exhaustiva	0.975	0.681
62	Modelo	0.926	0.039
	Exploración exhaustiva	0.932	0.056
150	Modelo	0.938	0.085
	Exploración exhaustiva	0.976	0.717

Tabla 1.5: Resultados de la prueba de normalidad Shapiro-Wilk para la reinserción de subrutas.

Para problemas con 35 y 150 clientes, los datos distribuyen de manera normal, mientras que para instancias de 62 clientes, no se puede asumir la normalidad de los datos.

Debido a los resultados mostrados en la tabla 1.5, no se puede rechazar la hipótesis nula, por lo que se asume que los algoritmos para esos dos casos siguen

una distribución normal. Por lo tanto se realiza un prueba T-student para comparar las medias del modelo y de la exploración exhaustiva. La hipótesis nula (H_0) indica que no existen diferencias entre las medias de los algoritmos, y la hipótesis alternativa (H_a) indica que existe diferencias entre las medias de los grupos.

Clientes	Estadístico	p-valor
35	-8.966	0.000
150	-29.866	0.000

Tabla 1.6: Resultados de la prueba t-student para la reinserción de subrutas.

Los resultados de la tabla 1.6 indican la media del tiempo de ejecución de los modelos es realmente menor que la media del tiempo de ejecución de la exploración exhaustiva. Por lo tanto se puede afirmar que los modelos de optimización son mas rápidos que la exploración exhaustiva.

Los resultados obtenidos evidencian que el uso de modelos de optimización para explorar vecindades de una solución de un VRP puede mejorar el tiempo de búsqueda en las mismas. Sin embargo para usar estos modelos de optimización, es necesario que los datos que estos utilizan sean generados eficientemente, especialmente para estructuras de vecindad que posean un mayor número de vecinos.

En la siguiente sección, se presentan los resultados de la generación eficiente de los datos, evaluando cómo esta metodología contribuye a reducir la complejidad computacional y mejorar el rendimiento global en comparación con métodos tradicionales.

1.2. Resultados de la Generación Eficiente de Datos

En esta sección se presentan los tiempos asociados a la generación y utilización de los datos necesarios para la vecindad presentada en el capítulo ???. Los experimentos evalúan tres enfoques principales: la búsqueda exhaustiva, la resolución del modelo de optimización, y la obtención directa de la mejor solución de la vecindad a partir de los datos generados.

La búsqueda exhaustiva fue implementada en *Common Lisp*, como se detalló en la sección anterior, y se utilizó como referencia para medir el desempeño de los otros enfoques. Los modelos de optimización, por otro lado, aprovecharon los datos generados eficientemente para resolver el problema con un tiempo considerablemente más bajo. Finalmente, se analizó la posibilidad de realizar una búsqueda

directa a través de los datos generados, lo que permitió evaluar su potencial como una alternativa práctica y rápida para encontrar soluciones de calidad.

En la Tabla 1.7, se detallan los tiempos de ejecución de cada algoritmo para una instancia de la vecindad de reinserción de dos clientes con 35 clientes.

Algoritmo	Media	Mediana	Desviación estándar
Exploración exhaustiva	25.62s	25.67s	1.04s
Modelo	9.59s	9.61s	0.07s
Búsqueda en los datos	0.94s	0.93s	0.03s

Tabla 1.7: Comparación de los resultados entre la exploración exhaustiva el modelo de optimización y la búsqueda en los datos

Como se evidencia en la tabla 1.7, la búsqueda en el mínimo de la vecindad en los datos generados en el capítulo ?? puede ser mas eficiente que realizar una búsqueda exhaustiva e incluso que resolver el modelo de optimización. Para validar la comparación resulta necesario verificar si los datos distribuyen de manera normal, para ello se realiza un test Shapiro-Wilk. Los resultados de la prueba se muestran en la tabla.

Algoritmo	Estadístico	p-valor
Exploración exhaustiva	0.966	0.429
Modelo	0.840	0.000
Búsqueda en los datos	0.966	0.432

Tabla 1.8: Resultados de la prueba de normalidad Shapiro-Wilk para la reinserción de subrutas.

Como se evidencia en la tabla 1.8 los resultados de la exploración exhaustiva y la búsqueda en los datos, presentan una distribución normal y con los resultados de los modelos no se puede asumir que distribuyan normal. Debido a que el modelo no distribuye de manera normal se realiza una prueba de Wilcoxon entre los resultados del modelo de optimización, los resultados de la búsqueda exhaustiva y los resultados de la búsqueda en los datos. En la tabla 1.9 se muestran los resultados de estas pruebas.

En la primera fila de la tabla 1.9 se muestra el estadístico y el p-valor de la prueba de Wilcoxon realizada entre los resultados del modelo de optimización y los resultados de la búsqueda exhaustiva. En la segunda fila los valores para la

Algoritmo	Estadístico	p-valor
Exploración exhaustiva	0.000	0.000
Búsqueda en los datos	0.000	0.000

Tabla 1.9: Resultados de la prueba de Wilcoxon entre los datos del modelo y los datos de la búsqueda exhaustiva y los datos de la búsqueda en los datos.

prueba realizada entre los resultados del modelo de optimización y los resultados de la búsqueda en los datos.

Los resultados mostrados en la tabla 1.9, indican que la diferencia entre los tiempos de ejecución del modelo de optimización con la exploración exhaustiva y la búsqueda en los datos es real. Por lo tanto se puede concluir que el tiempo de ejecución del modelo de optimización es mas rápido que el tiempo de ejecución de realizar una búsqueda exhaustiva, y a su vez mas lento que el tiempo de ejecución de realizar la búsqueda en los datos.

Los resultados obtenidos destacan dos conclusiones principales. En primer lugar, la generación eficiente de datos contribuye a reducir el tiempo necesario para resolver el modelo de optimización. En segundo lugar, la búsqueda directa a través de los datos generados demuestra ser mucho más rápida que resolver el modelo completo, lo que sugiere que una estrategia ideal podría ser enfocarse en la generación de datos y emplearlos directamente para la búsqueda de soluciones.

Bibliografía

- [1] Daniela González Beltrán. Generación automática de gramáticas para la obtención de infinitos criterios de vecindad en el problema de enrutamiento de vehículos. mayo 2019.
- [2] Gabriela Mijenes Carrera. Análisis de vecindades en problemas de enrutamiento de vehículos con múltiples depósitos y flotas heterogéneas. noviembre 2021.
- [3] Miguel Ángel Villasís-Keever Eric Flores Ruiz, María Guadalupe Miranda-Novales². El protocolo de investigación vi: cómo elegir la prueba estadística adecuada. estadística inferencial. 2017.
- [4] Camila Pérez Mosquera. Primeras aproximaciones a la búsqueda de vecindad infinitamente variable. 2017.
- [5] Héctor Felipe Massón Rosquete. Exploración de vecindades grandes en el problema de enrutamiento de vehículos usando técnicas estadísticas. mayo 2020.
- [6] José Jorge Rodríguez Salgado. Una propuesta para la evaluación automática de soluciones vecinas en un problema de enrutamiento de vehículos a partir del grafo de evaluación de una solución. junio 2020.
- [7] Eduardo Uchoa. New benchmark instances for the capacitated vehicle routing problem. 2016.