



**Universidad
Andrés Bello®**
FACULTAD DE INGENIERÍA
INGENIERIA CIVIL INFORMÁTICA

ANÁLISIS, OPTIMIZACIÓN Y DOCUMENTACIÓN DE VRP

Tesis de pregrado para optar al título de Ingeniero Civil Informático

Bastian Ignacio Moya Carrasco

Felipe Ismael Moya Carrasco

Profesor tutor:

Dr. Gustavo Gatica

Santiago, Chile

2024

Resumen

Este trabajo se centra en replicar y evaluar el modelo presentado en el paper "*The finished vehicle routing problem with a heterogeneous transport fleet*" (HVRP–FVL), que aborda una genealogía del problema de ruteo de vehículos. Este modelo considera una flota heterogénea en logística de vehículos terminados e incorpora diversos costos de transporte, como peajes, costos laborales, consumo energético, penalizaciones por tiempo y emisiones de carbono. El objetivo del HVRP–FVL es minimizar el costo total de la flota mediante una metaheurística basada en algoritmos genéticos (GA) con codificación dual, cruzamiento adaptativo, mutación y operadores de escalada.

En el desarrollo de este trabajo, se revisaron tres versiones de este estudio, junto con fuentes adicionales, para esclarecer variables no declaradas o mal dimensionadas. A pesar de las inconsistencias encontradas en las fórmulas y valores reportados, se trabajó en replicar mecánicamente los casos de estudio provistos en el paper. Esto permitió evaluar la funcionalidad del modelo a nivel práctico, ya que por la falta de información no se puede validar el modelo completo.

El objetivo principal fue diseñar y replicar, mediante el uso de Python, la funcionalidad propuesta, que se utilizó para buscar rutas óptimas que minimizaran el uso de camiones, garantizaran la puntualidad dentro de las ventanas de tiempo y maximice la capacidad de carga de cada vehículo. Si bien no se logró corroborar la veracidad total del modelo teórico, se obtuvieron rutas optimizadas que cumplen con los criterios logísticos establecidos, demostrando la aplicabilidad de este enfoque en escenarios reales de transporte.

Tabla de contenidos

Resumen	ii
Tabla de contenidos	iii
Tabla de contenidos	iv
Ilustraciones	iv
Tablas.....	iv
Conceptos Clave	v
Introducción	1
Marco teórico.....	2
Evolución Histórica de la formulación de VRP	2
Metodología.....	3
Funcionalidades del Software	5
Variables especiales:.....	5
Proceso de asignación:	5
Salida de datos:	5
Implementación Técnica	6
Modelo Matemático	7
Descripción general:.....	7
Conjuntos y parámetros:	7
Variable de decisión:	7
Función Objetivo:	7
Restricciones:.....	7
Código Fuente	9
Resultados.....	10
Conclusiones	13
Referencias	14

Tabla de contenidos

Ilustraciones

Ilustración 1 Modelos originarios del problema VRP	2
Ilustración 2 Diagrama de componentes	6
Ilustración 3 Solución optimizada de acuerdo con el paper.....	10
Ilustración 4 Solución optimizada de 100 ordenes	11

Tablas

Tabla 1 Resultados uso_casos.ipynb caso 21 clientes.....	10
Tabla 2 Comparativa Resultados Fitness Caso 1	11
Tabla 3 Resultados uso_casos.ipynb caso 100 clientes.....	12

Conceptos Clave

*VRP: Vehicle routing problem (Problema de Ruteo de Vehículos).

*Heurísticas: Atajos que otorgan decisiones más rápidas que métodos tradicionales de computación.

*Pseudocódigo: Descripción de un algoritmo o programa escrito en lenguaje natural estructurado.

*Fitness: Características que miden la calidad del código.

Introducción

Este informe presenta el análisis, optimización y documentación del código implementado para resolver el **Vehicle Routing Problem (VRP)**, tomando como base el artículo [*"The Finished Vehicle Routing Problem with a Heterogeneous Transport Fleet"*](#). Este artículo sirvió como referencia principal para la realización de este trabajo y la presente tesis.

El artículo original aborda una variante del VRP enfocada en la planificación de rutas óptimas para la distribución de productos mediante una flota heterogénea de vehículos. Cada vehículo parte desde un único punto, o centro de distribución, hacia múltiples clientes, tomando en consideración diversas restricciones relacionadas con los costos, las características de los vehículos a llevar, los propios camiones y las ventanas de tiempo. Aunque el artículo proporciona un esquema en pseudocódigo, este no detalla completamente la implementación de ciertos aspectos del modelo ni especifica cómo se valida el fitness esperado.

Sabiendo eso, se realizó un análisis exhaustivo del artículo y se diseñó un código funcional desde cero, incorporando mejoras y ajustes necesarios para adaptar el modelo a un escenario práctico. Durante este proceso, se logró replicar las rutas optimizadas siguiendo el esquema presentado en el artículo, aunque con algunas diferencias en los valores de Fitness obtenidos.

El objetivo principal es desarrollar un algoritmo para el VRP que optimice las rutas de una flota heterogénea de vehículos, tomando en consideración diversas restricciones operativas y logísticas:

- **Capacidad máxima de carga:** Cada vehículo tiene un límite de carga, tanto en términos de peso como de dimensiones, que no debe ser excedido.
- **Ventanas de tiempo:** Cada cliente debe ser atendido dentro de un intervalo de tiempo específico.
- **Cobertura de clientes:** Cada cliente debe ser visitado exactamente una vez con su entrega completa, y cada vehículo debe regresar al centro de distribución después de completar sus entregas.

A través de este trabajo, se busca comprobar la eficacia del modelo propuesto en el artículo y su aplicabilidad en contextos logísticos reales.

Marco teórico

Evolución Histórica de la formulación de VRP

Los problemas tipo VRP ha recibido varias variaciones desde la primera vez que fue planteado en 1956, pero siempre se sigue con la base de salir de un punto hacia los distintos destinos, lo cuales solo se visita una vez, para luego volver al inicio.

Desde el primer VRP planteado por Flood nacieron variaciones, como se muestra en la siguiente figura:

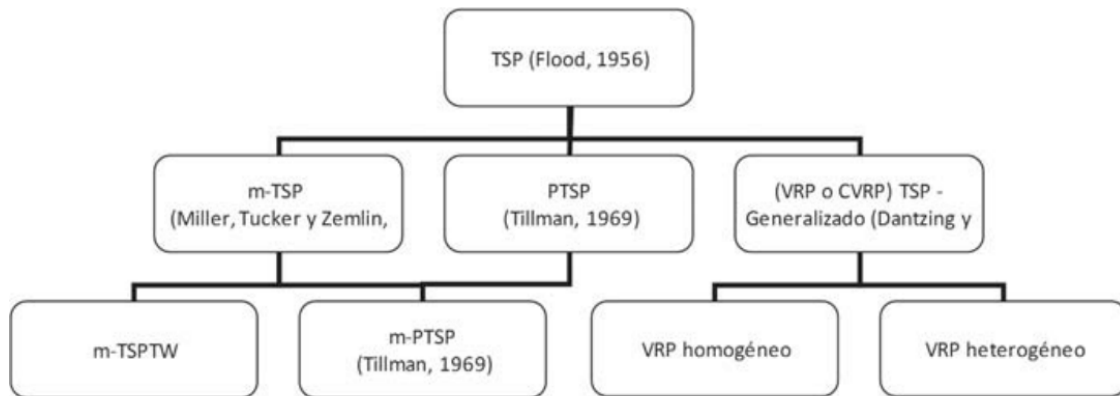


Ilustración 1 Modelos originarios del problema VRP

Cada una de las variaciones presenta modificaciones específicas, como la evaluación de un número determinado de rutas, el análisis del costo mínimo requerido o la incorporación de ligeros ajustes entre ellas (Rocha Medina, L. B., González La Rota, E. C., & Orjuela Castro, J. A., 2011).

[1] Rocha Medina, L. B., González La Rota, E. C., & Orjuela Castro, J. A. (2011). Una revisión al estado del arte del problema de ruteo de vehículos: Evolución histórica y métodos de solución. Ingeniería, 16(2), 35-55.

Metodología

El desarrollo del presente trabajo se realizó utilizando el lenguaje de programación **Python** en el entorno de desarrollo **Visual Studio Code**.

Herramientas y entorno de desarrollo

- **Lenguaje de programación:** Python fue elegido por su flexibilidad y su amplia disponibilidad de bibliotecas para procesamiento de datos, optimización y simulación.
- **Entorno de desarrollo:** Visual Studio Code, una herramienta versátil y ampliamente utilizada en el desarrollo de proyectos de software.
- **Gestión de datos:** Los datos necesarios para el problema provistos en el paper, como las características de los vehículos, los pedidos y las distancias entre nodos, se almacenaron en archivos de texto, permitiendo una fácil modificación y análisis.

Proceso de desarrollo

1. **Análisis preliminar:**
 - Se revisaron las inconsistencias y elementos faltantes en el artículo original. Este análisis reveló vacíos conceptuales y valores sin declarar, lo que requirió realizar múltiples ajustes y reelaboraciones durante el desarrollo del código.
2. **Diseño del modelo y lectura de datos:**
 - Se implementó un sistema para la lectura de datos desde archivos de texto, permitiendo así estructurar y validar la información básica del problema (capacidad de los vehículos, restricciones de carga, ventanas de tiempo, entre otros).
 - Se adaptaron las características del modelo al contexto del caso de estudio.
3. **Implementación de las funciones de fitness:**
 - Las funciones de fitness fueron diseñadas siguiendo los criterios planteados en el artículo de referencia. Estas funciones permiten evaluar la calidad de las soluciones generadas, considerando factores como:
 - Minimización del número de camiones.
 - Cumplimiento de ventanas de tiempo.
 - Maximización de la carga transportada por cada vehículo.
 - Las funciones fueron validadas a través de pruebas con datos reales.
4. **Optimización mediante algoritmos genéticos:**
 - El algoritmo genético implementado incluyó las siguientes etapas:

- **Cálculo del fitness inicial:** Se evaluó la población inicial generada.
 - **Selección:** Se seleccionaron las soluciones con mayor probabilidad de mejorar el resultado.
 - **Cruzamiento y mutación:** Se aplicaron operadores adaptativos para explorar nuevas configuraciones de rutas.
 - **Actualización de la población:** Se integraron las nuevas soluciones generadas, repitiendo el ciclo hasta converger en una solución óptima.
- Las probabilidades de cruzamiento y mutación se calcularon dinámicamente en cada iteración para garantizar un equilibrio entre exploración y explotación.

5. Validación y ajuste del modelo:

- Se evaluaron las soluciones obtenidas con base en los resultados esperados y se ajustaron los parámetros del modelo según las observaciones.

Resultados esperados

El proceso descrito permitió comprobar 2 cosas:

- El funcionamiento del código en su función principal (cálculo de fitness, generación de poblaciones, etc.)
- El funcionamiento del código en los casos de estudios provistos, los cuales sirvieron como base para probar el mismo.

Funcionalidades del Software

Variables especiales:

- N1
- N2

Proceso de asignación:

- Input de data: El software recibe la data a utilizar en los algoritmos desde varios archivos de texto (.txt).
- Evaluación y análisis: La evaluación de cada variable de vehículo o cliente se evaluará mediante un algoritmo desarrollado en Python para determinar la manera óptima de viaje y entrega de vehículos.
- Criterios de asignación: Los criterios de entrega serán dependientes de la carga máxima de los tipos de vehículos, considerando también las dimensiones de estos.

Salida de datos:

- El programa retorna el trayecto óptimo y junto con el valor de Fitness de cada parte (f_1, f_2 y f_3)

Implementación Técnica

Usando librerías de ampliación de funciones matemáticas, tratamiento de texto y copia de datos. Estas serían math, re y copy respectivamente, además de otras para funciones específicas.

- Entrada de datos: Los datos de los clientes, los camiones, autos y parámetros utilizados se extrajeron del paper y fuentes afines, y se introducen en el programa a través de archivos txt estructurados
- Creación variables N1 y N2: Utilizando los datos extraídos, el programa sigue las instrucciones del paper, creando N1 y N2 los cuáles serán las bases para el GA
- Generación rutas optimas: Habiendo hecho las n° iteraciones el programa le entrega al usuario la solución encontrada, el cual consiste en la ruta y el fitness.

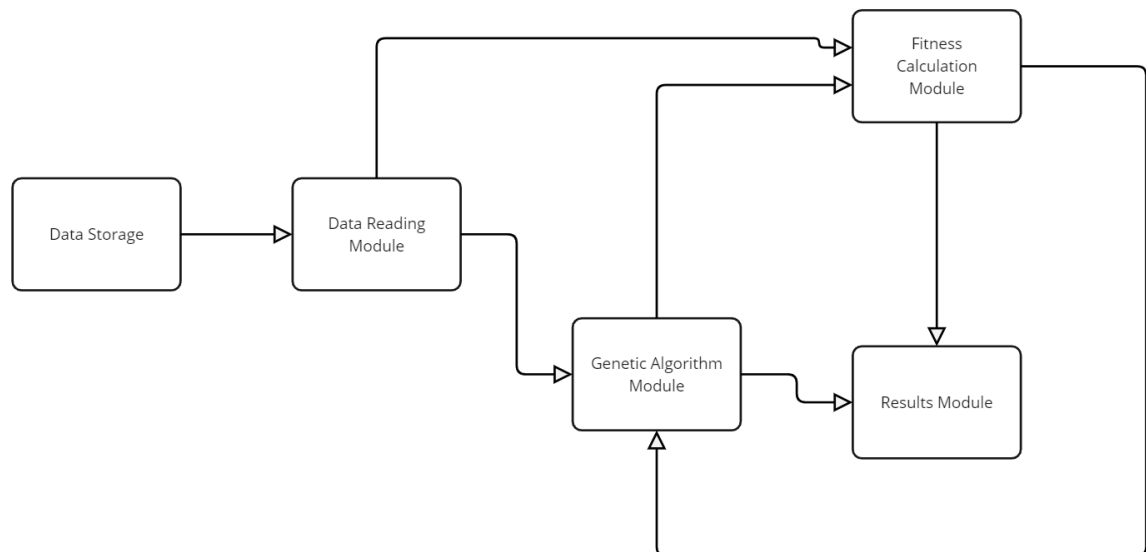


Ilustración 2 Diagrama de componentes

- Data Storage: Este módulo actúa como el repositorio de archivos utilizados por el sistema.
- Data Reading Module: Lee y transforma los archivos almacenados en el módulo de almacenamiento en estructuras de datos listas para su uso.
- Genetic Algorithm Module: Implementa el algoritmo genético para la optimización de rutas.
- Fitness Calculation Module: Evalúa la calidad de cada ruta generada por el algoritmo genético
- Results Module: Maneja el almacenamiento y la presentación de los resultados generados por el sistema.

Modelo Matemático

Se utiliza el modelo matemático provisto en el paper “*The finished vehicle routing problem with a heterogeneous transport fleet*”:

Descripción general:

El problema considera un conjunto de nodos N que deben ser visitados por una flota heterogénea de vehículos K con distintas dimensiones y cantidad U_k . El objetivo es minimizar el costo total de transporte F

Conjuntos y parámetros:

- V : Conjunto de vehículos.
- N : Conjunto de nodos (clientes), donde 0 es el depósito.
- Q_v : Capacidad máxima del vehículo v .
- w_p : Demanda del cliente i (en toneladas).
- e_i, l_i : Ventana de tiempo para iniciar el servicio en el cliente i (inicio e_i , fin l_i).
- t_{ij} : Tiempo de viaje entre el nodo i y el nodo j .
- c_{ij} : Costo de viaje entre el nodo i y el nodo j .
- d_{ij} : Distancia entre el nodo i y el nodo j .
- D_{ip} : Demanda del cliente i de vehículos tipo p .
- G_k, L_k, W_k, H_k : Limitaciones de carga, largo, ancho y alto del vehículo de tipo k .
- β : Constante específica por vehículo.

Variable de decisión:

- x_{iju}^k : Variable binaria que toma valor 1 si el vehículo u del tipo k se encuentra en el arco (i, j) , y 0 en caso contrario
- y_{iu}^k : Variable binaria que toma valor 1 si el vehículo u del tipo k visitará al cliente i , 0 en caso contrario

Función Objetivo:

La función objetivo es minimizar el coste (rmc) de toda la operación

$$F = f_1 + f_2 + f_3$$

Donde f_1 son los costos de transporte, f_2 las penalizaciones de tiempo y f_3 los costes por emisiones de carbono.

Restricciones:

- Asignación única: Los clientes deben ser visitados por un único vehículo, sin separar los envíos

$$\sum_{u=1}^{U_k} \sum_{k=1}^K y_{iu}^k = 1, \forall i \in N$$

- Conservación del movimiento: para cada cliente i , el número de vehículos que entran debe ser igual al salir

$$\sum_{i=0}^N x_{iju}^k = \sum_{i=0}^N x_{jiu}^k, \forall j \in N, \forall k \in K; \forall u \in U_k$$

- Restricción de capacidad: El peso total al unir clientes a una ruta no debe superar el peso permitido por el vehículo

$$\sum_{p=1}^P \sum_{i=1}^N D_{ip} w_p y_{iu}^k \leq G_k, \forall k \in K, \forall u \in U_k$$

- Restricción de espacio: El largo total al unir clientes a una ruta no debe superar el peso permitido por el vehículo

$$\sum_{p=1}^P \sum_{i=1}^N D_{ip} l_p y_{iu}^k \leq L_k, \forall k \in K, \forall u \in U_k$$

Código Fuente

El código desarrollado para este trabajo está disponible en el siguiente repositorio de GitHub:

https://github.com/moyaxon/tesis_entrega

En este repositorio, se pueden encontrar:

- La implementación completa del algoritmo genético.
- Las funciones de fitness utilizadas para evaluar las rutas.
- Los datos de prueba y los resultados obtenidos en los casos de estudio.

Resultados

Dentro del paper utilizado para este trabajo se encuentran 2 casos de prueba con sus rutas obtenidas, y además en el primer caso se entrega el fitness resultante. Es por esta razón que se analizan los resultados de Fitness y las rutas obtenidas del caso 1, y para el caso 2 solo se analizan las rutas resultantes:

Table 5 The optimized solutions

Route No	Transport vehicle type	Route details	Length loading rate(%)	Weight loading rate(%)
1	1	0→D20→D21→0	89.41	72.42
2	3	0→D19→D18→D15→0	96.86	80.60
3	2	0→D14→D17→0	85.76	79.11
4	3	0→D13→D12→D11→0	97.96	86.45
5	2	0→D8→D10→0	83.21	76.75
6	2	0→D16→D1→0	99.17	80.76
7	1	0→D4→D2→0	90.97	71.33
8	3	0→D3→D6→D9→0	97.34	86.54
9	2	0→D7→D5→0	94.66	80.26

Ilustración 3 Solución optimizada de acuerdo con el paper

Ruta n°	Tipo Vehículo	Ruta
1	1	[20, 21]
2	3	[19, 18, 15]
3	2	[14, 17]
4	3	[13, 12, 11]
5	2	[8, 10]
6	2	[16, 1]
7	1	[4, 2]
8	3	[3, 6, 9]
9	2	[7, 5]

Tabla 1 Resultados uso_casos.ipynb caso 21 clientes

	f_1	f_2	f_3	F
<i>Paper</i>	35078	95	32391	67564
<i>Replica</i>				
<i>desarrollada</i>	43460	374	590913	634749

Tabla 2 Comparativa Resultados Fitness Caso 1

Table 2 The optimized solution with 100 customer orders

Route No	Transport vehicle type	Route Details
1	3	D0→D46→D22→D93→D0
2	2	D0→D97→D95→D0
3	3	D0→D1→D42→D63→D40→D0
4	3	D0→D67→D26→D34→D0
5	2	D0→D98→D5→D0
6	2	D0→D32→D9→D0
7	3	D0→D92→D7→D54→D0
8	3	D0→D83→D70→D86→D0
9	2	D0→D35→D81→D0
10	2	D0→D37→D2→D0
11	3	D0→D4→D23→D57→D0
12	3	D0→D27→D52→D21→D0
13	2	D0→D80→D24→D0
14	3	D0→D100→D60→D19→D0
15	3	D0→D50→D94→D18→D0
16	3	D0→D85→D43→D53→D0
17	3	D0→D87→D49→D20→D0
18	3	D0→D55→D76→D71→D0
19	1	D0→D96→D90→D0
20	3	D0→D10→D89→D8→D0
21	3	D0→D99→D72→D0
22	3	D0→D65→D11→D13→D0
23	3	D0→D75→D29→D12→D44→D0
24	3	D0→D82→D73→D91→D0
25	2	D0→D78→D48→D0
26	2	D0→D69→D15→D0
27	2	D0→D17→D84→D88→D0
28	2	D0→D61→D45→D25→D0
29	2	D0→D47→D39→D0
30	2	D0→D64→D58→D0
31	2	D0→D31→D6→D0
32	2	D0→D16→D38→D0
33	2	D0→D62→D36→D0
34	1	D0→D14→D0
35	2	D0→D51→D56→D0
36	3	D0→D28→D33→D68→D0
37	2	D0→D41→D79→D0
38	2	D0→D59→D30→D0
39	2	D0→D66→D3→D0
40	2	D0→D74→D77→D0

Ilustración 4 Solución optimizada de 100 ordenes

Ruta n°	Tipo Vehículo	Ruta
1	3	[46, 22, 93]
2	2	[97, 95]
3	3	[1, 42, 63, 40]
4	3	[67, 26, 34]
5	2	[98, 5]
6	2	[32, 9]
7	3	[92, 7, 54]
8	3	[83, 70, 86]
9	2	[35, 81]
10	2	[37, 2]
11	3	[4, 23, 57]
12	3	[27, 52, 21]
13	2	[80, 24]
14	3	[100, 60, 19]
15	3	[50, 94, 18]
16	3	[85, 43, 53]
17	3	[87, 49, 20]
18	3	[55, 76, 71]
19	1	[96, 90]
20	3	[10, 89, 8]
21	3	[99, 72]
22	3	[65, 11, 13]
23	3	[75, 29, 12, 44]
24	3	[82, 73, 91]
25	2	[78, 48]
26	2	[69, 15]
27	2	[17, 84, 88]
28	2	[61, 45, 25]
29	2	[47, 39]
30	2	[64, 58]
31	2	[31, 6]
32	2	[16, 38]
33	2	[62, 36]
34	1	[14]
35	2	[51, 56]
36	3	[28, 33, 68]
37	2	[41, 79]
38	2	[59, 30]
39	2	[66, 3]
40	2	[74, 77]

Tabla 3 Resultados uso_casos.ipynb caso 100 clientes

Conclusiones

El análisis y desarrollo realizado en este trabajo permitieron implementar y optimizar un algoritmo basado en heurísticas para resolver el **Problema de Rutas de Vehículos (VRP)** con una flota heterogénea. Comparando los resultados obtenidos con los del artículo base, se evidencia que el algoritmo propuesto es capaz de replicar e incluso mejorar ciertas soluciones dadas por el paper original, respetando las restricciones establecidas, como la capacidad de los vehículos y las ventanas de tiempo.

1. Resultados alcanzados:

- Se logró implementar un algoritmo genético desde cero, respetando las funciones de fitness descritas en el paper base. Esto permitió optimizar las rutas considerando múltiples restricciones, como la capacidad de los vehículos y las ventanas de tiempo de los clientes.
- El algoritmo entregó soluciones comparables al artículo base en términos de las rutas a usar por cada camión, lo que nos muestra que se logró el resultado esperado.

2. Limitaciones:

- Aunque el algoritmo genético ofreció buenos resultados, el tiempo de ejecución podría ser mejorado mediante la implementación de optimizaciones adicionales, ya que los datos no clarificados por parte de la fuente pueden cambiar drásticamente el resultado de los ejemplos.
- No se exploraron otras metaheurísticas, como el recocido simulado o algoritmos basados en colonias de hormigas, que podrían ofrecer perspectivas diferentes al problema.

3. Propuestas futuras:

- Integrar herramientas como Gurobi para resolver el problema de manera exacta y comparar los resultados con los obtenidos mediante heurísticas.
- Explorar métodos híbridos que combinen metaheurísticas y enfoques exactos para balancear mejor la calidad de las soluciones y el tiempo de ejecución.

En resumen, este trabajo demostró la viabilidad de aplicar algoritmos genéticos para resolver problemas complejos como el VRP con flota heterogénea, y sienta las bases para futuras investigaciones en optimización logística utilizando métodos computacionales avanzados.

Referencias

- Rocha Medina, L. B., González La Rota, E. C., & Orjuela Castro, J. A. (2011). Una revisión al estado del arte del problema de ruteo de vehículos: Evolución histórica y métodos de solución. Ingeniería, 16(2), 35-55.
- Liu, Jiansheng Liu and Yuan, Bin and Hu, Yingcong and Smith, Alice E., The Finished Vehicle Routing Problem with a Heterogeneous Transport Fleet. Available at SSRN: <https://ssrn.com/abstract=4239703> or <http://dx.doi.org/10.2139/ssrn.4239703>
- Tolga Bektaş, Gilbert Laporte, The Pollution-Routing Problem, Transportation Research Part B: Methodological, Volume 45, Issue 8, 2011, Pages 1232-1250, ISSN 0191-2615, <https://doi.org/10.1016/j.trb.2011.02.004>. (<https://www.sciencedirect.com/science/article/pii/S019126151100018X>)
- Emrah Demir, Tolga Bektaş, Gilbert Laporte, A comparative analysis of several vehicle emission models for road freight transportation, Transportation Research Part D: Transport and Environment, Volume 16, Issue 5, 2011, Pages 347-357, ISSN 1361-9209, <https://doi.org/10.1016/j.trd.2011.01.011>. (<https://www.sciencedirect.com/science/article/pii/S136192091100023X>)