Jurnal Pendidikan Matematika
https://jurnal.uin-antasari.ac.id/index.php/jpm/index
Vol. 10 No.2 Tahun 2023, h. 59-68
DOI: 10.18592/jpm.v10i2.9756

PENERAPAN ALGORITMA NEAREST NEIGHBOR DALAM PERMASALAHAN TSP UNTUK MENENTUKAN RUTE TERPENDEK PENDISTRIBUSIAN KRUPUK RENGGINANG

Achmad Gilang Pamungkas Hani Putra, Eva Amilatus Solikah,
Siti Ainun Nisak
Universitas Nahdlatul Ulama Blitar
E-mail Korespondensi: pamungkas.putra79@gmail.com

Abstrak

Travelling Salesman Problem (TSP) adalah masalah optimasi yang digunakan untuk menentukan rute terpendek yang dimulai dan diakhiri pada titik yang sama, dengan mengunjungi semua titik yang diberikan tepat satu kali. Pada studi ini, permasalahan TSP diterapkan pada distribusi krupuk rengginang, di mana seorang pedagang harus merencanakan rute yang efisien dari tempat produksi menuju toko-toko untuk mengurangi biaya dan waktu perjalanan. Penelitian ini mengusulkan solusi optimasi menggunakan algoritma Nearest Neighbour, yang menghasilkan rute optimal sebagai berikut: A → B → H → I → F → G → C → D → E → J → A, dengan total jarak yang ditempuh sebesar 23,95 unit. Hasil ini menunjukkan bahwa algoritma *Nearest Neighbour* dapat memberikan solusi yang memadai dalam meminimalkan jarak tempuh pada kasus TSP ini.

Kata Kunci : *Algoritma Nearest Neighbour; Heuristik; Optimasi; Travelling Salesman Problem*

Pendahuluan

Rengginang, makanan sejenis kerupuk yang terbuat dari beras ketan, semakin diminati di pasar. Namun, seiring dengan meningkatnya permintaan, tantangan utama yang muncul adalah distribusi yang efisien, khususnya dalam mengoptimalkan rute pengiriman dari rumah produksi ke berbagai toko. Keterlambatan dalam pengiriman produk dapat mengurangi kepuasan konsumen, yang akhirnya memilih produk lain yang sudah tersedia. Hal ini menuntut solusi dalam menentukan rute distribusi yang optimal agar produk tetap tersedia tepat waktu (Afrizal & Sunaryono, 2016; Febrian, 2019). Dalam konteks ini, Travelling Salesman Problem (TSP), yang bertujuan untuk menemukan rute terpendek dengan mengunjungi setiap titik tepat satu kali dan kembali ke titik asal, menjadi model yang sangat relevan. TSP sering digunakan dalam berbagai aplikasi logistik dan distribusi barang, dan

merupakan salah satu masalah klasik dalam teori graf (Febrian, 2019). Namun, penerapannya pada distribusi makanan khas seperti rengginang belum banyak dibahas dalam literatur yang ada, sehingga penelitian ini mencoba mengisi kekosongan tersebut.

Penelitian sebelumnya telah banyak membahas penggunaan berbagai algoritma untuk menyelesaikan masalah TSP, seperti algoritma Genetic, Simulated Annealing, dan Ant Colony Optimization (Hernández, Orozco, & Sarmiento, 2018; Zhang, Zhang, & Wei, 2019). Setiap algoritma memiliki kelebihan dan kekurangan dalam hal kecepatan konvergensi dan kualitas solusi. Genetic Algorithm dan Simulated Annealing, misalnya, sering kali menghasilkan solusi yang lebih optimal, tetapi membutuhkan waktu komputasi yang lebih lama (Georgiadis, Bie, & Lee, 2020). Sebaliknya, algoritma Nearest Neighbor (NN) menawarkan pendekatan yang lebih sederhana dan lebih cepat, meskipun solusi yang dihasilkan cenderung suboptimal. Namun, NN terbukti efektif untuk masalah TSP dengan jumlah titik yang terbatas, seperti yang ditemukan dalam penelitian oleh Kotsialos, Gendreau, dan Mladenović (2017). Oleh karena itu, penerapan algoritma Nearest Neighbor dalam konteks distribusi rengginang menjadi pendekatan yang menarik, mengingat efisiensi komputasi yang lebih rendah dan kemampuan algoritma ini memberikan solusi yang cukup baik untuk masalah dengan skala kecil hingga menengah.

Penelitian ini bertujuan untuk menerapkan algoritma Nearest Neighbor dalam penyelesaian TSP untuk distribusi rengginang, yang diharapkan dapat memberikan solusi yang efisien dalam hal jarak dan waktu tempuh. Selain itu, penelitian ini juga membandingkan hasil yang diperoleh dengan solusi distribusi lainnya yang lebih kompleks, untuk mengevaluasi keunggulan dan keterbatasan dari pendekatan yang lebih sederhana ini dalam konteks logistik lokal.

Dengan demikian, tujuan utama penelitian ini adalah untuk mengkaji penerapan graf dan algoritma Nearest Neighbor dalam menentukan jalur distribusi krupuk rengginang secara optimal, serta memberikan kontribusi dalam pengembangan aplikasi logistik untuk produk tradisional yang membutuhkan efisiensi distribusi.

Metode Penelitian

Penelitian ini menggunakan pendekatan kuantitatif, yang berfokus pada penggunaan data numerik dalam pengumpulan, pengolahan, dan analisis data (Zuhairi dalam Suci, 2020). Tujuan dari penelitian ini adalah untuk menentukan jalur atau sirkuit terpendek yang harus dilalui, dari titik awal hingga titik akhir, menggunakan algoritma Nearest Neighbor.

Data yang digunakan dalam penelitian ini berupa jarak antar titik pemberhentian yang akan dilalui selama perjalanan. Pengumpulan data dilakukan melalui wawancara dengan pihak yang terkait, yang menyediakan informasi mengenai lokasi-lokasi yang harus dikunjungi dan jarak antar lokasi tersebut. Setelah data terkumpul, data jarak antar titik diproses dan dihitung

menggunakan algoritma Nearest Neighbor untuk memperoleh rute terpendek yang menghubungkan seluruh titik secara efisien.

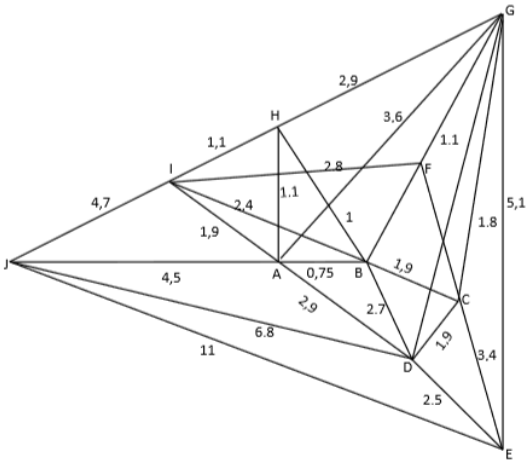
Hasil dan Pembahasan

Langkah pertama adalah pengumpulan data dengan cara menghitung seluruh jarak antar toko. Data tersebut dapat ditampilkan da;am bentuk tabel berikut ini:

Tabel 4.1 Data jarak antar toko

	A	B	C	D	E	F	G	H	I	J
A	0	0,75	2,1	2,9	5,3	2,4	3,6	1,1	1,9	4,5
B	0,75	0	1,9	2,7	5,1	2,2	3,4	1	2,4	5
C	2,1	1,9	0	1,9	3,4	1,2	1,8	2,4	3,8	6,5
D	2,9	2,7	1,9	0	2,5	2,2	3,3	3,3	4,1	6,8
E	5,3	5,1	3,4	2,5	0	4,5	5,1	6,1	6,6	11
F	2,4	2,2	1,2	2,2	4,5	0	1,1	1,8	2,8	6,8
G	3,6	3,4	1,8	3,3	5,1	1,1	0	2,9	4	7,9
H	1,1	1	2,4	3,3	6,1	1,8	2,9	0	1,1	5,6
I	1,9	2,4	3,8	4,1	6,6	2,8	4	1,1	0	4,7
J	4,5	5	6,5	6,8	11	6,8	7,9	5,6	4,7	0

Langkah kedua adalah permodelan permasalahan menggunakan teori graf. Permasalahan ini dapat ditampilkan dalam bentuk graf sebagai berikut :



Gambar 4.1 Model permasalahan

Gambar tersebut menunjukkan jaringan komunikasi yang kompleks dengan titik sentral yaitu 'G'. 'G' terhubung dengan semua titik dan memiliki jalur komunikasi dengan intensitas tinggi, menunjukkan peran penting dalam organisasi, mungkin sebagai pemimpin atau koordinator. Terdapat kluster dengan komunikasi intensif seperti G-F-H dan G-C-E, mengindikasikan keterlibatan dalam proyek bersama atau fungsi terkait. Di sisi lain, titik 'J' dan 'I' relatif terisolasi dengan koneksi terbatas, menunjukkan peran independen atau terpinggirkan. Jalur komunikasi yang panjang dan tidak langsung, seperti antara 'J' dan 'B', dapat menjadi hambatan komunikasi. Intensitas komunikasi yang tidak seimbang, dengan 'G' sebagai pusat, menunjukkan kemungkinan distribusi informasi atau beban kerja yang tidak merata.

Langkah ketiga adalah mencari solusi dari permasalahan. Solusi dari permasalhan TSP yang telah dimodelkan dalam bentuk graf adalah sebuah Sirkuit Hamilton Terpendek (SHT). Untuk mendapatkan SHT tersebut kita gunakan algoritma Nearest Neighbor (NN).

Berikut langkah-langkah untuk mendapatakan SHT menggunakan algoritma NN:

1. Tentukan titik awal sirkuit yang sekaligus menjadi tiitk akhir dari sirkuit.
2. Dari titik awal tersebut tentukan tentukan titik-titik lain ynag terhubung denganna dan belum dikunjungi.
3. Pilih bobot terkecil dari garis yang menghubungkan titik awal dengan semua titik yang belum dikunjungi tersebut.
4. Titik dari garis yang terpilih menjadi titik awal pencarian berikutnya kemudian titik tersebut ditandai telah dikunjungi.
5. Ulangi langkah 2 sampai dengan 4 hingga tidak diketemukan lagi titik yang belum bertanda sudah dikunjungi.

Perjalanan mulai dari titik A. Titik A ini juga nanti akan menjadi titik akhir perjalanan, yakni perjalanan akan kembali ke titik awal perjalanan membentuk sebuah sirkuit.

Dari titik A ini maka ada 9 titik lain yang terhubung lansung, yaitu B, C, D, E, F, G, H, I dan J. Kesembilan titik tersebut belum satupun pernah dikunjungi. Buat daftar jarak dari titik A ke keempat titik tersebut dengan merujuk ke Tabel 1. Berdasarkan daftar jarak dapat ditentukan jarak terkecil, yaitu dari titik A ke titik B sebesar 0,75. Dengan demikian maka jalur pertama yang dilalui adalah A → B. Selanjutnya B akan menjadi titik awal penentuan jalur berikutnya dengan mengikuti algoritma penentuan jalur yang sama.

Dari titik B ada 9 titik lain yang terhubung lansung, yaitu , A, C, D, E, F, G, H, I dan J. Dari kesembilan titik tersebut maka titik C, D, E, F, G, H, I dan J tersebut belum pernah dikunjungi. Buat daftar jarak dari titik B ke kedelapan titik tersebut dengan merujuk ke Tabel 1. Berdasarkan daftar jarak dapat ditentukan jarak terkecil, yaitu dari titik B ke titik H sebesar 1. Dengan demikian maka jalur kedua yang dilalui adalah B → H. Selanjutnya, H akan menjadi titik awal penentuan jalur berikutnya dengan mengikuti algoritma penentuan jalur yang sama.

Dari titik H ada 9 titik lain yang terhubung lansung, yaitu A, B, C, D, E, F, G, I dan J. Dari kesembilan titik tersebut maka titik C, D, E, F, G, I dan J belum pernah dikunjungi. Buat daftar jarak dari titik H ke ketujuh titik tersebut dengan merujuk ke Tabel 1. Berdasarkan daftar jarak dapat ditentukan jarak terkecil, yaitu dari titik H ke titik I sebesar 1,1. Dengan demikian maka jalur ketiga yang dilalui adalah H → I. Selanjutnya , I akan menjadi titik awal penentuan jalur berikutnya dengan algoritma penentuan jalur yang sama.

Dari titik I ada 9 titik lain yang terhubung lansung, yaitu A, B, C, D, E, F, G, H dan J. Dari kesembilan titik tersebut maka titik C, D, E, F, G, dan J belum pernah dikunjungi. Buat daftar jarak dari titik I ke keenam titik tersebut dengan merujuk ke Tabel 1. Berdasarkan daftar jarak dapat ditentukan jarak terkecil, yaitu dari titik I ke titik F sebesar 2,8. Dengan demikian maka jalur keempat yang dilalui adalah I → F. Selanjutnya , F akan menjadi titik awal penentuan jalur berikutnya dengan algoritma penentuan jalur yang sama.

Dari titik F ada 9 titik lain yang terhubung lansung, yaitu A, B, C, D, E, G, H, I dan J Dari kesembilan titik tersebut maka titik C, D, E, G, dan J belum pernah dikunjungi. Buat daftar jarak dari titik F ke kelima titik tersebut dengan merujuk ke Tabel 1. Berdasarkan daftar jarak dapat ditentukan jarak terkecil, yaitu dari titik F ke titik G sebesar 1,1. Dengan demikian maka jalur kelima yang dilalui adalah F → G. Selanjutnya , G akan menjadi titik awal penentuan jalur berikutnya dengan algoritma penentuan jalur yang sama.

Dari titik G ada 9 titik lain yang terhubung lansung, yaitu A, B, C, D, E, F, H, I dan J. Dari Sembilan titik tersebut maka titik C, D, E dan J belum pernah dikunjungi. Buat daftar jarak dari titik G ke kelima titik tersebut dengan merujuk ke Tabel 1. Berdasarkan daftar jarak dapat ditentukan jarak terkecil, yaitu dari titik G ke titik C sebesar 1,8. Dengan demikian maka jalur

keenam yang dilalui adalah G → C. Selanjutnya , C akan menjadi titik awal penentuan jalur berikutnya dengan algoritma penentuan jalur yang sama.

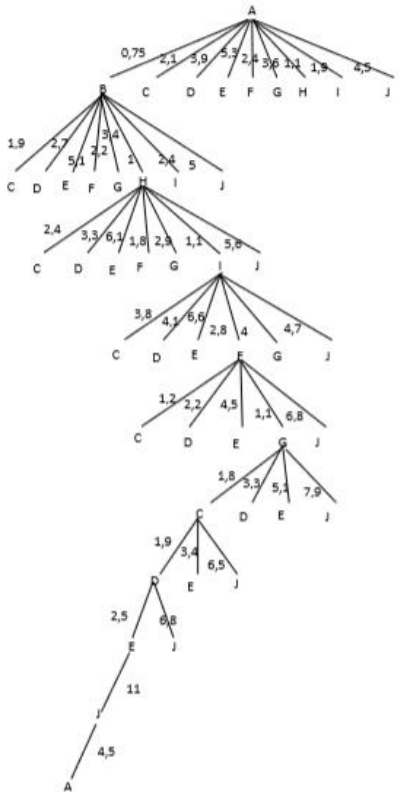
Dari titik C ada 9 titik lain yang terhubung lansung, yaitu A, B, D, E, F, G, H, I dan J. Dari kesembilan titik tersebut maka titik D, E dan J belum pernah dikunjungi. Buat daftar jarak dari titik C ke keempat titik tersebut dengan merujuk ke Tabel 1. Berdasarkan daftar jarak dapat ditentukan jarak terkecil, yaitu dari titik C ke titik D sebesar 1,9. Dengan demikian maka jalur ketujuh yang dilalui adalah C→ D. Selanjutnya , D akan menjadi titik awal penentuan jalur berikutnya dengan algoritma penentuan jalur yang sama.

Dari titik D ada 9 titik lain yang terhubung lansung, yaitu A, B, C, E, F, G, H, I dan J. Dari kesembilan titik tersebut maka hanya titik E dan J belum pernah dikunjungi. Buat daftar jarak dari titik D ke keempat titik tersebut dengan merujuk ke Tabel 1. Berdasarkan daftar jarak dapat ditentukan jarak terkecil, yaitu dari titik D ke titik E sebesar 2,5. Dengan demikian maka jalur kedelapan yang dilalui adalah D→ E. Selanjutnya , E akan menjadi titik awal penentuan jalur berikutnya dengan algoritma penentuan jalur yang sama.

Dari titik E ada 9 titik lain yang terhubung lansung, yaitu A, B, C, D, F, G, H, I dan J. Dari kesembilan titik tersebut maka hanya tinggal titik J yang belum pernah dikunjungi sekalipun. Dengan demikian tidak ada pilihan jalur lain kecuali E→ J, jarak E → J sebesar 11. Dengan demikian maka jalur kesembilan yang dilalui adalah E→ J. Selanjutnya, J akan menjadi titik awal penentuan jalur berikutnya dengan mengikuti algoritma penentuan jalur yang sama.

Karena semua titik pada graf sekarang telah dikunjungi semua maka tidak tersisa satu jalurpun kecuali jalur J → A dengan jarak sebesar 4,5. Atau dengan kata lain, jalur perjalanan harus ditutup dengan kembali ke titik awalnya atau titik A untuk membentuk Sirkuit Hamilton.

Berikut ilustrasi dari langkah - langkah di atas untuk mendapatkan Sirkuit Hamilton menggunakan algoritma NN:



Gambar 4.2 Ilustrasi kerja algoritma NN

Berdasarkan ilustrasi pada gambar maka rute yang terpilih adalah ABHIFGCDEJA. Total jarak rute tersebut adalah $0{,}75 + 1 + 1{,}1 + 2{,}8 + 1{,}1 + 1{,}8 + 1{,}9 + 2{,}5 + 11 = 23{,}95$. Rute ABHIFGCDEJA dengan total jarak sebesar 23,95 ini adalah jarak yang harus ditempuh untuk mengunjungi seluruh titik pada graf dan kembali ke titik awal yang didapatkan menggunakan algoritma NN.

Meskipun algoritma Nearest Neighbor (NN) yang diterapkan dalam penelitian ini memberikan solusi yang cepat dan sederhana untuk masalah Travelling Salesman Problem (TSP) dalam konteks distribusi krupuk rengginang, hasil yang diperoleh belum tentu optimal. Seperti yang telah dijelaskan dalam penelitian terdahulu oleh Albuquerque et al. (2017), algoritma NN adalah metode greedy yang memilih langkah terbaik saat itu tanpa mempertimbangkan dampaknya pada solusi keseluruhan. Hal ini berpotensi menghasilkan solusi suboptimal, terutama ketika jumlah titik yang harus dikunjungi semakin banyak. Dalam penelitian ini, meskipun hasil yang diperoleh dengan algoritma NN adalah jarak total 23,95, yang relatif memadai untuk jumlah titik terbatas, solusi ini bisa saja tidak optimal jika dibandingkan dengan metode lain yang lebih kompleks.

Metode lain yang sering dibandingkan dengan NN dalam menyelesaikan masalah TSP adalah Algoritma Genetika (GA) dan Simulated Annealing (SA). Algoritma Genetika, yang mengadopsi prinsip seleksi alam untuk menghasilkan solusi terbaik, memiliki kemampuan untuk

mengeksplorasi berbagai kemungkinan jalur secara lebih luas dan sering kali menghasilkan solusi yang lebih optimal. Simulated Annealing, menurut penelitian Yücesan et al. (2019), meskipun lebih lambat dibandingkan NN, mampu menghindari terjebak pada solusi lokal dan menghasilkan solusi global yang lebih baik, dengan menggunakan teknik pemanasan bertahap yang mengadaptasi kemungkinan solusi terbaik seiring waktu. Namun, baik GA maupun SA memerlukan lebih banyak waktu komputasi dan cenderung lebih rumit untuk diterapkan, terutama pada masalah dengan jumlah titik yang kecil hingga menengah.

Di sisi lain, metode yang memberikan solusi optimal secara pasti namun dengan kompleksitas yang sangat tinggi adalah Algoritma Dynamic Programming (DP). Meskipun DP menawarkan solusi optimal, seperti yang dijelaskan oleh Leong et al. (2018), algoritma ini sangat tidak efisien ketika jumlah titik yang harus dikunjungi meningkat, karena kompleksitas waktunya yang mencapai O (n^2 * 2^n). Ini membuatnya kurang praktis untuk aplikasi dunia nyata dengan banyak titik. Dalam konteks distribusi krupuk rengginang yang hanya melibatkan sejumlah kecil toko, penggunaan NN sudah cukup efisien dan memberikan hasil yang memadai, namun untuk masalah TSP dengan lebih banyak titik atau kebutuhan solusi yang sangat optimal, algoritma lain seperti GA atau SA bisa dipertimbangkan meskipun dengan biaya komputasi yang lebih tinggi.

Berdasarkan hasil penelitian ini, dengan total jarak 23,95, dapat disimpulkan bahwa meskipun algoritma Nearest Neighbor memiliki kelemahan dalam menghasilkan solusi yang optimal, algoritma ini tetap menjadi pilihan yang sangat efisien dan praktis untuk masalah TSP dengan jumlah titik yang terbatas. Oleh karena itu, untuk aplikasi yang melibatkan lebih banyak titik, disarankan untuk menggunakan algoritma lain yang lebih kompleks, seperti Simulated Annealing atau Algoritma Genetika, meskipun harus siap dengan waktu komputasi yang lebih lama dan implementasi yang lebih rumit.

Kesimpulan

Permasalahan Travelling Salesman Problem (TSP) dalam konteks distribusi krupuk rengginang telah diselesaikan dengan menggunakan algoritma Nearest Neighbor (NN), yang terbukti memberikan solusi yang cepat dan efisien. Berdasarkan hasil pembahasan, rute yang terpilih adalah A → B → H → I → F → G → C → D → E → J → A, dengan total jarak yang ditempuh sebesar 23,95. Hasil ini menunjukkan jarak terpendek yang harus ditempuh oleh pedagang dalam mengunjungi seluruh titik pada graf dan kembali ke titik awal, yang didapatkan melalui langkah-langkah algoritma NN.

Walaupun algoritma NN memberikan solusi yang cepat, hasil yang diperoleh tidak selalu optimal, terutama pada kasus dengan jumlah titik yang lebih banyak. Meskipun demikian, untuk jumlah titik yang relatif terbatas, seperti pada permasalahan ini, NN tetap menjadi pilihan yang efisien dan

memadai dalam hal waktu komputasi dan implementasi. Penelitian lebih lanjut dengan menggunakan algoritma lain yang lebih kompleks seperti Simulated Annealing atau Algoritma Genetika dapat dipertimbangkan jika diperlukan solusi yang lebih optimal dengan pengorbanan waktu komputasi yang lebih tinggi.

Demikian, penelitian ini memberikan kontribusi dalam penerapan algoritma NN untuk menyelesaikan masalah TSP dalam konteks distribusi barang, dengan hasil yang berguna untuk perencanaan logistik dan penghematan biaya perjalanan.

Daftar Pustaka

Afrizal Herdyanto Sunaryono, Y. P. (2016). Pemilihan Rute Perjalanan Terpendek Menggunakan Algoritma. prosiding matematika.

Albuquerque, G., Ferreira, L., & Silva, E. (2017). A comparison of heuristic methods for the traveling salesman problem. *European Journal of Operational Research, 258*(3), 951-961. https://doi.org/10.1016/j.ejor.2016.11.045

Andik L. 2022. Penerapan K-NN dalam Permasalahan TSP untuk Menentukan Rite Terpendek Pengambilan Ulat Hongkong. Skripsi. Tidak Diterbitkan. Fakultas Ilmu Eksakta. Universitas Nahdlatul Ulama: Blitar.

Febrian, D. (2019). Aplikasi Metode Tetangga (Nearest Neighbour Algorithm) Terdekat untuk Mencari Rute Terpendek Perjalanan Wisata Museum dan Wisata Religi di Kota Medan. 5(1), 1–13.

Leong, C. K., Tan, W. K., & Lee, W. B. (2018). Solving the traveling salesman problem using dynamic programming approach. *International Journal of Mathematical, Engineering, and Management Sciences*, *3*(3), 258-271. https://doi.org/10.22161/ijmems.3.3.8

Suci,W. (2020). Perbandingan Media Pembelajaran terhadap Hasil Belajar Al-Islam di SMA Muhammadiyah 1 gisting Kabupaten Tanggamus Tahun Pelajaran 2019/2020. Skripsi. Lampung: Institut Agama Islam Negeri (IAIN) Metro.

Yücesan, E., Ergen, B., & Yavuz, N. (2019). Solving the traveling salesman problem using simulated annealing and genetic algorithm techniques: A comparison. *Computers and Industrial Engineering, 135*, 301-308. https://doi.org/10.1016/j.cie.2019.06.015

Georgiadis, P., Bie, Z., & Lee, Y. (2020). Optimizing logistics in urban areas: The role of the travelling salesman problem. *International Journal of Production Economics, 221*, 107473. https://doi.org/10.1016/j.ijpe.2019.107473

Hernández, S., Orozco, J., & Sarmiento, J. (2018). A comparison of metaheuristics for solving the Traveling Salesman Problem. *Computers & Operations Research, 91*, 88-97. https://doi.org/10.1016/j.cor.2017.08.004

Kotsialos, A., Gendreau, M., & Mladenović, N. (2017). Hybrid algorithms for the Travelling Salesman Problem. *Computers & Operations Research*, *86*, 133-141. https://doi.org/10.1016/j.cor.2017.06.003

Zhang, X., Zhang, Y., & Wei, X. (2019). A Hybrid Algorithm for Solving the Traveling Salesman Problem with Time Windows. *Soft Computing*, *23*(12), 4185-4198. https://doi.org/10.1007/s00542-019-04602-z