Assessment of approximation method for TSP path length on road networks: a simulation study

Bachelor Thesis

Koen Stevens, S5302137

2025-04-17

Abstract

1 Introduction

The Traveling Salesman Problem is an important problem in operations research. It is particularly relevant for last-mile carriers and other logistics companies where efficient routing directly impacts cost, time and service quality. Since the number of parcels worldwide has increased between 2013 and 2022 and is expected to keep increasing [Statista (2025)], the need for fast, scalable route planning methods becomes ever more pressing.

The TSP is an NP-hard problem, it is computationally intensive to find the exact solution for large instances. In many real-world scenarios, the exact optimal routes may not be needed, but instead a rough, reliable estimate of the optimal route length. For instance, consider a postal delivery company. This firm may need to assign a certain amount of deliveries or a certain area to each postman. Reliable estimates for the route length can provide valuable information for making such decisions.

Efficient approximation methods provide a solution for such practical applications where exact solutions are too computationally intensive to conduct or not feasible due to insufficient data. These methods aim at approximating the expected optimal total travel time or distance, while using minimal data and computational effort.

There is extensive research on such approximation methods and how they perform in the euclidean plane. Consider n uniformly drawn locations inside some area in $\mathbb{R}^2$ with area A. Beardwood, Halton, and Hammersley (1959) prove the relation:

$$L \to \beta\sqrt{nA}, \quad \text{as } n \to \infty \tag{1}$$

as an estimation for the length of the shortest TSP path measured by Euclidean distance through these random locations, where β is some proportionality constant. This formula is a very elegant result, and it requires very little data. However, its assumptions, uniform random locations and euclidean space differ from real-world applications, which are defined by complex geographic features, such as road networks.

This research investigates how well this approximation method performs when we consider real road networks. Using OpenStreetMap data, we simulate TSP instances in a wide variety of different urban areas in the Netherlands, then solve these for the actual shortest paths using the Lin–Kernighan heuristic. Then we estimate the β from equation 1 and analyze the performance of this formula. Additionally, we compare the results for β and the performance across the selected areas.

In section 2 we dive deeper in the context and previous research in this field. Then, in section 3 we show the methodology …

2 Literature Review

2.1 Applications of the Beardwood formula

2.2 TSPs on road networks

Merchán and Winkenbach (2019) use circuity factors to measure the relative detour incurred for traveling in a road network, compared to the euclidean distance. This circuity factor is defined as, where p and q are locations:

$$c = \frac{d_c(p,q)}{d_{L_2}(p,q)} \tag{2}$$

c is greater or equal to 1, a value closer to 1 indicates a more efficient network. This circuity factor has been estimated many times for different road networks, with results ranging between 1.12 and 5.60 [Merchán and Winkenbach (2019)]. This is an efficient method of implementing the impact of traveling over a road network into the models for solving the TSPs, since the euclidean distance can simply be multiplied by a constant. However, this might be an oversimplification of reality: this circuity factor might differ strongly from area to area or differ by route length. Additionally, the locations used were generated from uniform distribution across the area. This might also be unrealistic since population density differs strongly from area to area, especially in a city where there is highrise, then the distribution of the delivery locations is clearly not uniform.

2.3 Lin-Kernighan Heuristic

3 Methodology

4 References

Beardwood, Jillian, John H Halton, and John Michael Hammersley. 1959. "The Shortest Path Through Many Points." In *Mathematical Proceedings of the Cambridge Philosophical Society*, 55:299–327. 4. Cambridge University Press.

Merchán, Daniel, and Matthias Winkenbach. 2019. "An Empirical Validation and Data-Driven Extension of Continuum Approximation Approaches for Urban Route Distances." *Networks* 73 (4): 418–33.

Statista. 2025. “Global Parcel Shipping Volume Between 2013 and 2027 (in Billion Parcels)*.” https://www.statista.com/statistics/1139910/parcel-shipping-volume-worldwide/.

5 Appendix

Province	Neighborhood	Beta
groningen	Hortusbuurt	2.2208
groningen	Binnenstad	2.0437
groningen	Oosterpoort	2.0470
groningen	Rivierenbuurt	1.7724
groningen	De Wijert	1.7573
groningen	Oosterparkwijk	1.7643
groningen	De Hoogte	1.6945
groningen	Korrewegwijk	2.0534
groningen	Schildersbuurt	2.2178
groningen	Paddepoel	1.6677
groningen	Oranjewijk	1.9397
groningen	Tuinwijk	2.8835
groningen	Selwerd	1.5161
groningen	Vinkhuizen	1.4757
groningen	Hoogkerk-zuid	1.5459
groningen	Gravenburg	1.2975
groningen	De Held	1.9346
groningen	Reitdiep	1.6487
groningen	Hoornse Meer	1.5939
groningen	Corpus den Hoorn	1.6125
groningen	Eemspoort	1.7375
groningen	Euvelgunne	1.9185
groningen	Driebond	1.9013
groningen	Winschoterdiep	1.9654
groningen	Eemskanaal	1.7624
groningen	Helpman	2.0727
groningen	Lewenborg	1.9173
groningen	Beijum	1.7968
groningen	Maarsveld	1.6806
noord$_{\text{holland}}$	Schrijverswijk	1.7545
noord$_{\text{holland}}$	Stad van de Zon	1.4652
noord$_{\text{holland}}$	Stadshart	1.4591
noord$_{\text{holland}}$	Jordaan	1.8435
noord$_{\text{holland}}$	Slotervaart	1.7482
noord$_{\text{holland}}$	IJburg	1.3494

Province	Neighborhood	Beta
$noord_{holland}$	Oostelijke Eilanden	1.7053
$noord_{holland}$	Oostelijk Havengebied	1.7309
$noord_{holland}$	Frederik Hendrikbuurt	2.2883
$noord_{holland}$	Van Lennepbuurt	1.7913
$noord_{holland}$	Da Costabuurt	2.4902
$noord_{holland}$	Kinkerbuurt	1.9689
$noord_{holland}$	Kersenboogerd	1.6491
$noord_{holland}$	Pax	2.2029
$noord_{holland}$	Graan voor Visch	2.2168
$noord_{holland}$	Vrijschot-Noord	2.4070
$noord_{holland}$	Toolenburg	1.2956
$noord_{holland}$	Floriande	1.9073
$noord_{holland}$	Overbos	1.7891
$noord_{holland}$	Bornholm	1.8117
$noord_{holland}$	Beukenhorst-Oost	1.6960
$noord_{holland}$	De Hoek	2.5427
$noord_{holland}$	West	2.0170
$noord_{holland}$	Zuid	1.6601
$noord_{holland}$	Oost	1.8783
$noord_{holland}$	Noord	1.6477
$noord_{holland}$	De President	1.4798
$noord_{holland}$	Graan voor Visch-Zuid	1.7271
$noord_{holland}$	Zuidwijk	1.3949
$noord_{holland}$	Buitenveldert-West	1.1917
$noord_{holland}$	Buitenveldert	1.1390
$noord_{holland}$	Apollobuurt	1.7345
$noord_{holland}$	Stadionbuurt	1.4951
$noord_{holland}$	Prinses Irenebuurt e.o.	1.8844
$noord_{holland}$	Hoofddorppleinbuurt	1.6764
$noord_{holland}$	Willemspark	1.9469
$noord_{holland}$	Schinkelbuurt	1.6475
$noord_{holland}$	Vondelparkbuurt	1.2907
$noord_{holland}$	Helmersbuurt	1.8120
$noord_{holland}$	Overtoomse Sluis	1.9484
$noord_{holland}$	Museumkwartier	1.7059
$noord_{holland}$	Rivierenbuurt	1.6828
$noord_{holland}$	IJselbuurt	1.5692
$noord_{holland}$	Scheldebuurt	1.3681
$noord_{holland}$	Rijnbuurt	1.6430
$noord_{holland}$	De Baarsjes	1.7969
$noord_{holland}$	Landlust	1.7649
$noord_{holland}$	Staatsliedenbuurt	1.8221

Province	Neighborhood	Beta
$\text{noord}_{\text{holland}}$	Spaarndammerbuurt	2.2238
$\text{noord}_{\text{holland}}$	De Pijp	2.2767
$\text{noord}_{\text{holland}}$	Grachtengordel	1.7950
$\text{noord}_{\text{holland}}$	Oud-Zuid	1.4400