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

Bachelor Thesis

Koen Stevens, S5302137

2025-04-18

Abstract

1 Introduction

The Traveling Salesman Problem (TSP) 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 from 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

In this section we review the existing literature on the Beardwood formula and some applications, and on the Lin-Kernighan heuristic and its implementations.

2.1 Applications of the Beardwood formula

We are interested in the performance of formula 1 for reasonable amounts of locations a delivery person can visit in a workday, say $10 \leq n \leq 90$. Table 1 lists values for β for some values of n, where the points were generated uniformly and the L_2 distance metric was used.

Table 1: Empirical estimates of β as a function of n, $20 \leq n \leq 90$ (Lei et al. (2015))

n	$\beta(n)$
20	0.8584265
30	0.8269698
40	0.8129900
50	0.7994125
60	0.7908632
70	0.7817751
80	0.7775367
90	0.7773827

Figliozzi (2008) is the first research to apply approximation formulas to real-world instances of TSPs (and VRPs (Vehicle Routing Problems)). An extension of formula 1 that works for VRPs is assessed in a real-world setting. It is found that this model has an R^2 of 0.99 and MAPE (Mean Absolute Prediction Error) of 4.2%. This prediction error is slightly higher than when it is applied to a setting where euclidean distances are considered (3.0%), but the formula still performs well.

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:

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

By construction, c is greater or equal to 1, a value closer to 1 indicates a more efficient network. Then, β_c is estimated by $\beta_c = c\beta$. This value c, is estimated for three different areas in São Paulo, for which the results are listed in table 2. These values indicate real travel distances are on average 2.76 times longer in area 1 compared to the L_2 metric. These values were obtained by uniformly generating n locations (for n ranging from 3 to 250), computing near-optimal tour lengths under the Euclidean metric, and solving for β, then scaling by the empirical circuity factor.

Table 2: Estimates of the circuity factor c and its corresponding β_c (Merchán and Winkenbach (2019))

	Area 1	Area 2	Area 3
c	2.76	2.34	1.82
β_c	2.48	2.10	1.64

It is important to note, however, that the assumptions in this study may limit the generality of the findings. In particular, the use of uniformly distributed locations does not accurately reflect the spatial distribution of delivery points in real urban environments, where locations tend to cluster in residential, commercial, or industrial zones. Additionally, within small urban areas, high-rise buildings and single-family homes may coexist in the same neighborhoods, further challenging the assumption of uniformly distributed delivery points. Furthermore, the circuity factor c can vary significantly within a single city, depending on local street patterns, infrastructure, and topography. These variations suggest that a fixed circuity factor may oversimplify the complexity of real-world delivery contexts, especially when applied to smaller subregions or neighborhoods.

2.2 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.

Figliozzi, Miguel Andres. 2008. "Planning Approximations to the Average Length of Vehicle Routing Problems with Varying Customer Demands and Routing Constraints." *Transportation Research Record* 2089 (1): 1–8.

Lei, Hongtao, Gilbert Laporte, Yajie Liu, and Tao Zhang. 2015. "Dynamic Design of Sales Territories." *Computers & Operations Research* 56: 84–92.

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	3.6832
groningen	Binnenstad	4.1028
groningen	Oosterpoort	3.2324
groningen	Rivierenbuurt	4.2625
groningen	De Wijert	3.3988
groningen	Oosterparkwijk	3.5939
groningen	De Hoogte	3.7108
groningen	Korrewegwijk	3.5945
groningen	Schildersbuurt	3.8140
groningen	Paddepoel	3.3608
groningen	Oranjewijk	3.2170
groningen	Tuinwijk	4.6051
groningen	Selwerd	3.1126
groningen	Vinkhuizen	2.6504
groningen	Hoogkerk-zuid	2.8405
groningen	Gravenburg	2.2658
groningen	De Held	3.9560
groningen	Reitdiep	3.0784
groningen	Hoornse Meer	3.0772
groningen	Corpus den Hoorn	3.0551
groningen	Eemspoort	3.5762
groningen	Euvelgunne	3.5178
groningen	Driebond	3.0449
groningen	Winschoterdiep	3.3515
groningen	Eemskanaal	2.9778

Province	Neighborhood	Beta
groningen	Helpman	3.8994
groningen	Lewenborg	3.8750
groningen	Beijum	3.4823
groningen	Maarsveld	2.7156
$noord_{holland}$	Schrijverswijk	3.0681
$noord_{holland}$	Stad van de Zon	3.0368
$noord_{holland}$	Stadshart	3.6922
$noord_{holland}$	Jordaan	4.7617
$noord_{holland}$	Slotervaart	3.4576
$noord_{holland}$	IJburg	3.4142
$noord_{holland}$	Oostelijke Eilanden	4.0579
$noord_{holland}$	Oostelijk Havengebied	5.7842
$noord_{holland}$	Frederik Hendrikbuurt	7.5317
$noord_{holland}$	Van Lennepbuurt	5.5274
$noord_{holland}$	Da Costabuurt	8.2504
$noord_{holland}$	Kinkerbuurt	5.0759
$noord_{holland}$	Kersenboogerd	2.9840
$noord_{holland}$	Pax	3.5472
$noord_{holland}$	Graan voor Visch	3.8677
$noord_{holland}$	Vrijschot-Noord	3.8515
$noord_{holland}$	Toolenburg	2.5297
$noord_{holland}$	Floriande	3.7561
$noord_{holland}$	Overbos	3.2689
$noord_{holland}$	Bornholm	3.3881
$noord_{holland}$	Beukenhorst-Oost	3.6461
$noord_{holland}$	De Hoek	4.5682
$noord_{holland}$	West	3.2603
$noord_{holland}$	Zuid	3.2844
$noord_{holland}$	Oost	3.2730
$noord_{holland}$	Noord	2.7834
$noord_{holland}$	De President	2.3944
$noord_{holland}$	Graan voor Visch-Zuid	3.2154
$noord_{holland}$	Zuidwijk	2.3899
$noord_{holland}$	Buitenveldert-West	2.6021
$noord_{holland}$	Buitenveldert	2.4989
$noord_{holland}$	Apollobuurt	3.2761
$noord_{holland}$	Stadionbuurt	3.5855
$noord_{holland}$	Prinses Irenebuurt e.o.	3.3820
$noord_{holland}$	Hoofddorppleinbuurt	4.5651
$noord_{holland}$	Willemspark	3.9025
$noord_{holland}$	Schinkelbuurt	3.7872
$noord_{holland}$	Vondelparkbuurt	2.8246

Province	Neighborhood	Beta
$\text{noord}_{\text{holland}}$	Helmersbuurt	5.6882
$\text{noord}_{\text{holland}}$	Overtoomse Sluis	5.6386
$\text{noord}_{\text{holland}}$	Museumkwartier	4.0253
$\text{noord}_{\text{holland}}$	Rivierenbuurt	4.4952
$\text{noord}_{\text{holland}}$	IJselbuurt	4.3521
$\text{noord}_{\text{holland}}$	Scheldebuurt	3.4606
$\text{noord}_{\text{holland}}$	Rijnbuurt	5.7265
$\text{noord}_{\text{holland}}$	De Baarsjes	6.1257
$\text{noord}_{\text{holland}}$	Landlust	6.2927
$\text{noord}_{\text{holland}}$	Staatsliedenbuurt	5.1110
$\text{noord}_{\text{holland}}$	Spaarndammerbuurt	5.1041
$\text{noord}_{\text{holland}}$	De Pijp	6.6650
$\text{noord}_{\text{holland}}$	Grachtengordel	4.7770
$\text{noord}_{\text{holland}}$	Oud-Zuid	3.5313