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

Bachelor Thesis

Koen Stevens, S5302137

2025-04-19

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 considering real road networks. Using OpenStreetMap data, TSP instances are simulated in a wide variety of different urban areas in the Netherlands, then solve these for the actual shortest paths using the Lin–Kernighan (Lin and Kernighan 1973) heuristic. Then the β from equation 1 is estimated and the performance of this formula is analyzed. Additionally, the results for β and the performance across the selected areas is compared.

In section 2 a deep dive in the context and previous research in this field is provided. In section 3 the experimental design is documented.

2 Literature Review

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

2.1 Applications of the Beardwood formula

This research is concerns 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 (Figliozzi (2008)).

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

To be able to efficiently solve many TSPs, to find a good estimate for β, a fast and reliable solution algorithm is needed. The Lin-Kernighan (Lin and Kernighan 1973) heuristic provides outcome, it is generally considered to be one of the most effective methods of generating (near) optimal solutions for the TSP. In this research a modified implementation of the heuristic is used (Helsgaun 2000). The run times of both heuristics increase by approximately $n^{2.2}$, but the modified heuristic is much more effective. It is able to find optimal solutions to large instances in reasonable times (Helsgaun 2000).

PARAGRAPH ABOUT HOW THE HEURISTIC WORKS

3 Experimental design

3.1 Data

3.2 Generation and solving of TSPs

4 Results

5 Discussion

6 Conclusion

7 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.

Helsgaun, Keld. 2000. "An Effective Implementation of the Lin–Kernighan Traveling Salesman Heuristic." *European Journal of Operational Research* 126 (1): 106–30.

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

Lin, Shen, and Brian W Kernighan. 1973. "An Effective Heuristic Algorithm for the Traveling-Salesman Problem." *Operations Research* 21 (2): 498–516.

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/.

8 Appendix

Province	Neighborhood	Beta
groningen	Hortusbuurt	2.2197
groningen	Binnenstad	2.0428
groningen	Oosterpoort	2.0359
groningen	Rivierenbuurt	1.7779
groningen	De Wijert	1.7561
groningen	Oosterparkwijk	1.7698
groningen	De Hoogte	1.7028
groningen	Korrewegwijk	2.0592
groningen	Schildersbuurt	2.2240
groningen	Paddepoel	1.6704
groningen	Oranjewijk	1.9470
groningen	Tuinwijk	2.8923
groningen	Selwerd	1.5238
groningen	Vinkhuizen	1.4784
groningen	Hoogkerk-zuid	1.5556
groningen	Gravenburg	1.3061
groningen	De Held	1.9358
groningen	Reitdiep	1.6557
groningen	Hoornse Meer	1.5883
groningen	Corpus den Hoorn	1.6074
groningen	Eemspoort	1.7349
groningen	Euvelgunne	1.9058
groningen	Driebond	1.8945
groningen	Winschoterdiep	1.9701
groningen	Eemskanaal	1.7716
groningen	Helpman	2.0734
groningen	Lewenborg	1.9133
groningen	Beijum	1.8017
groningen	Maarsveld	1.6757
$\text{noord}_{\text{holland}}$	Schrijverswijk	1.7461
$\text{noord}_{\text{holland}}$	Stad van de Zon	1.4595
$\text{noord}_{\text{holland}}$	Stadshart	1.4584
$\text{noord}_{\text{holland}}$	Jordaan	1.8419
$\text{noord}_{\text{holland}}$	Slotervaart	1.7458
$\text{noord}_{\text{holland}}$	IJburg	1.3445
$\text{noord}_{\text{holland}}$	Oostelijke Eilanden	1.7011
$\text{noord}_{\text{holland}}$	Oostelijk Havengebied	1.7282
$\text{noord}_{\text{holland}}$	Frederik Hendrikbuurt	2.2905
$\text{noord}_{\text{holland}}$	Van Lennepbuurt	1.8079
$\text{noord}_{\text{holland}}$	Da Costabuurt	2.5017
$\text{noord}_{\text{holland}}$	Kinkerbuurt	1.9756
$\text{noord}_{\text{holland}}$	Kersenboogerd	1.6555

Province	Neighborhood	Beta
$\text{noord}_{\text{holland}}$	Pax	2.1976
$\text{noord}_{\text{holland}}$	Graan voor Visch	2.2232
$\text{noord}_{\text{holland}}$	Vrijschot-Noord	2.4097
$\text{noord}_{\text{holland}}$	Toolenburg	1.2980
$\text{noord}_{\text{holland}}$	Floriande	1.9087
$\text{noord}_{\text{holland}}$	Overbos	1.7870
$\text{noord}_{\text{holland}}$	Bornholm	1.8147
$\text{noord}_{\text{holland}}$	Beukenhorst-Oost	1.6793
$\text{noord}_{\text{holland}}$	De Hoek	2.5402
$\text{noord}_{\text{holland}}$	West	2.0184
$\text{noord}_{\text{holland}}$	Zuid	1.6708
$\text{noord}_{\text{holland}}$	Oost	1.8782
$\text{noord}_{\text{holland}}$	Noord	1.6478
$\text{noord}_{\text{holland}}$	De President	1.4994
$\text{noord}_{\text{holland}}$	Graan voor Visch-Zuid	1.7247
$\text{noord}_{\text{holland}}$	Zuidwijk	1.3947
$\text{noord}_{\text{holland}}$	Buitenveldert-West	1.1970
$\text{noord}_{\text{holland}}$	Buitenveldert	1.1434
$\text{noord}_{\text{holland}}$	Apollobuurt	1.7353
$\text{noord}_{\text{holland}}$	Stadionbuurt	1.4852
$\text{noord}_{\text{holland}}$	Prinses Irenebuurt e.o.	1.8953
$\text{noord}_{\text{holland}}$	Hoofddorppleinbuurt	1.6765
$\text{noord}_{\text{holland}}$	Willemspark	1.9443
$\text{noord}_{\text{holland}}$	Schinkelbuurt	1.6466
$\text{noord}_{\text{holland}}$	Vondelparkbuurt	1.2894
$\text{noord}_{\text{holland}}$	Helmersbuurt	1.8133
$\text{noord}_{\text{holland}}$	Overtoomse Sluis	1.9465
$\text{noord}_{\text{holland}}$	Museumkwartier	1.7108
$\text{noord}_{\text{holland}}$	Rivierenbuurt	1.6913
$\text{noord}_{\text{holland}}$	IJselbuurt	1.5742
$\text{noord}_{\text{holland}}$	Scheldebuurt	1.3664
$\text{noord}_{\text{holland}}$	Rijnbuurt	1.6533
$\text{noord}_{\text{holland}}$	De Baarsjes	1.8024
$\text{noord}_{\text{holland}}$	Landlust	1.7613
$\text{noord}_{\text{holland}}$	Staatsliedenbuurt	1.8155
$\text{noord}_{\text{holland}}$	Spaarndammerbuurt	2.2497
$\text{noord}_{\text{holland}}$	De Pijp	2.2851
$\text{noord}_{\text{holland}}$	Grachtengordel	1.7890
$\text{noord}_{\text{holland}}$	Oud-Zuid	1.4385