


WILEY

Intl. Trans. in Op. Res. 31 (2024) 3443–3458  
DOI: 10.1111/itor.13260INTERNATIONAL  
TRANSACTIONS  
IN OPERATIONAL  
RESEARCH

# Estimating optimal objective values for the TSP, VRP, and other combinatorial problems using randomization

Shuhan Kou<sup>a,\*</sup> , Bruce Golden<sup>b</sup>  and Stefan Poikonen<sup>c</sup><sup>a</sup>*Department of Mathematics, University of Maryland, College Park, MD 20742, USA*<sup>b</sup>*Robert H. Smith School of Business, University of Maryland, College Park, MD 20742, USA*<sup>c</sup>*Business School, University of Colorado, Denver, CO 80204, USA**E-mail: shkou@terpmail.umd.edu [Kou]; bgolden@umd.edu [Golden]; stefan.poikonen@ucdenver.edu [Poikonen]*

Received 9 July 2022; received in revised form 25 November 2022; accepted 28 December 2022

## Abstract

Approximation of the optimal tour length in a Euclidean traveling salesman problem (TSP) has been studied by many researchers. In a previous study, we used the standard deviation in random tour lengths to approximate the optimal tour length in both Euclidean and non-Euclidean TSPs and we obtained good estimates. In this paper, we show that the strong power-law relationship between the standard deviation in random feasible solution values and the optimal solution value also holds for other Euclidean and near-Euclidean combinatorial optimization problems like the minimum spanning tree (MST) and maximum weight matching (MWM) problems. We then enhance the estimation ability of the model by considering a second predictor: the mean in random feasible solution values. Experimental results show that by using the mean, standard deviation, and randomization, we can accurately predict the optimal solution values for the TSP, MST, MWM, and the capacitated vehicle routing problem (VRP).

**Keywords:** combinatorial optimization; traveling salesman problem; vehicle routing problem; regression

## 1. Introduction

Given a graph  $G(V, E)$  with  $n$  vertices belonging to the set  $V = \{1, \dots, N\}$ , the solution to the traveling salesman problem (TSP) involves finding a tour where all vertices in  $V$  are visited exactly once. Such a tour with the minimum total distance is the optimal tour, and the distance traveled in this tour is the optimal tour length. As the TSP is NP-hard, finding the optimal TSP tour can require exponential computing time. However, with limited computing power, we can use an optimal tour length estimation model to assess the optimal tour length without finding the optimal tour.

\*Corresponding author.

© 2023 The Authors.

International Transactions in Operational Research © 2023 International Federation of Operational Research Societies.

Published by John Wiley & Sons Ltd, 9600 Garsington Road, Oxford OX4 2DQ, UK and 350 Main St, Malden, MA02148, USA.

There are various optimal tour length estimation models in the literature, and most of them concentrate on Euclidean TSPs. The model Beardwood et al. (1959) formulated for identically, independently, and uniformly distributed vertices on a Lebesgue set with a strictly positive measure in  $\mathbb{R}^2$  is given by

$$\lim_{N \rightarrow \infty} \frac{L}{\sqrt{N}} = \beta \sqrt{A_0},$$

where  $L$  denotes the optimal tour length,  $N$  represents the number of vertices,  $\beta$  is a constant that is approximately 0.75, and  $A_0$  is the measure of the Lebesgue set, which corresponds to the area in two-dimensional Euclidean space. For vertices distributed in rectangles, Daganzo (1984) found that this model can be applied to both Euclidean and Manhattan distance metrics. He derived upper bounds of  $\beta$  as 1.15 for the Manhattan distance metric and 0.90 for the Euclidean distance metric. Considering small instances with size  $5 \leq N \leq 30$  and the Euclidean distance metric, Chien (1992) found the best fit  $\beta$  to be 0.82.

Kwon et al. (1995) presented neural network models that make accurate estimations of  $L$  for instances with 10–80 uniformly independently and identically distributed (i.i.d.) vertices, using features including  $N$ ,  $A_1$ , the area of the smallest rectangle that covers all vertices,  $\sqrt{N}A_1$ , and the ratio of the length to the height of the service region.

Models including additional structural properties of TSP instances appear in the more recent literature. For example, Cavdar and Sokol (2015) proposed the following model:

$$L = 2.791 \sqrt{N(cstdev_x * cstdev_y)} + 0.2669 \sqrt{N(stdev_x * stdev_y) \frac{A_1}{\bar{c}_x \bar{c}_y}}, \quad (1)$$

where  $cstdev_x$  and  $cstdev_y$  measure the standard deviations of the distances to the vertices from the horizontal and vertical midpoint lines of the rectangular space containing all vertices,  $stdev_x$  and  $stdev_y$  are the standard deviations in  $x$  and  $y$  coordinates, and  $\bar{c}_x$  and  $\bar{c}_y$  are the average distances of vertices to the horizontal and vertical midpoint lines. Nicola et al. (2019) used stepwise regression and built a model including instance size  $N$ , maximum distance across all pairs of vertices, and the sum of distances to the nearest neighbor for each vertex. Akkerman et al. (2020) considered 21 different structural properties from the literature and 11 newly proposed structural properties before the feature selection step. Then, they used linear regression, random forest regression, and neural networks to construct multiple models. The best models developed by Cavdar and Sokol (2015), Nicola et al. (2019), and Akkerman et al. (2020) obtained promising results, with  $r$ -squared values higher than 0.95 on their test instances. These more recent works considered instances where vertices have known  $x$ – $y$  coordinates, and the distance between vertices is Euclidean.

Basel and Willemain (2001) ran a linear regression on 17 Euclidean instances from the TSPLIB repository (Reinelt, 1991) and introduced the following optimal tour length estimation model:

$$\ln(L) = 1.798 + 0.927 \ln(std_{RT}), \quad (2)$$

where  $std_{RT}$  is the standard deviation of tour lengths for 20,000 random tours. If we exponentiate both sides of Equation (2), we obtain  $L = e^{1.798} std_{RT}^{0.927}$ . Thus, this model describes the power-law relationship between  $std_{RT}$  and  $L$ . In addition, noting that the computation of  $std_{RT}$  does not

require the problem to be formulated in Euclidean space or have any known metric, we previously extended this estimation model to a wider range of TSP instances in Kou et al. (2022). This paper demonstrated an asymptotic linear relationship that exists between the  $\sqrt{NA}$  estimator and the  $std_{RT}$  estimator when vertices are uniformly i.i.d. in a unit square or a unit disk. We also used linear regression to study the predictive ability of the  $std_{RT}$  estimator. Instead of generating 20,000 random tours, we generated 1000 random tours to compute  $std_{RT}$  for a faster outcome and relatively good model validity. On all Euclidean test instances, the model from Equation (1) with different fitted constants presents  $r$ -squared values greater than 0.94. Notably, for  $T_{n,m}$  instances introduced by Hougardy and Zhong (2020), this model shows an  $r$ -squared value of 0.99 and a mean absolute percentage error (MAPE) of 2.8%. On most non-Euclidean test instances, including randomly generated instances and road maps in São Paulo, the results provided by the model from Equation (2) show  $r$ -squared values of 0.76–0.95 and MAPE ranging from 5.71% to 13.71%. The estimates provided by the  $std_{RT}$  predictor on  $L$  make us wonder whether the power-law relationship described in Equation (2) also holds for other combinatorial optimization problems and how the predictive results can be improved. In the Appendix, we, therefore, show that for two combinatorial optimization problems, which are solvable in polynomial time, the minimum spanning tree (MST) and maximum weight matching (MWM) problems, there is a strong power-law relationship between the standard deviation in feasible solutions and the optimal solution if the instances are Euclidean or near-Euclidean. These experimental results motivate us to generalize the idea of using random feasible solution values to estimate the optimal objective values to other NP-hard problems that lack known polynomial solvers.

In our model presented in this paper, we draw on Sutcliffe et al. (2012) who ran experiments showing that for the TSP, when the distance matrix satisfies the triangle inequality, there exists a strong correlation between  $L$  and  $mean - \sqrt{N}std$ , where  $std$  and  $mean$  are the standard deviation and mean for all tour lengths in the TSP. Thus, we also consider including the predictor  $mean_{RT}$ , which is the average length of random tours, to potentially improve our estimation model.

We structure the rest of this paper as follows. In Section 2, we enhance the estimation model by including  $mean_{RT}$  as another predictor and present experimental results for the TSP. In Section 3, we discuss the results of the original model using only the standard deviation predictor and this enhanced model to predict optimal capacitated vehicle routing problem (VRP) costs by considering random heuristic solutions. Conclusions are then presented in Section 4. In the Appendix, as mentioned earlier, we provide the experimental results for the MST and MWM.

## 2. Using $mean_{RT}$ and $std_{RT}$ to predict TSP tour length

In our previous work, we showed that using the  $std_{RT}$  predictor, one can obtain similar estimation results to those obtained using the traditional  $\sqrt{NA}$  predictor on Euclidean instances. For two out of three sets of Euclidean instances that we tested, the model using only the  $std_{RT}$  predictor even outperforms the model proposed by Cavdar and Sokol in Equation (1), which considers eight different instance features (Kou et al., 2022). We also tested the performance of the  $std_{RT}$  predictor on non-Euclidean instances and obtained reasonable estimates. A natural next step is to test whether adding the mean of random tour lengths as another predictor can enhance the predictive ability of the model. In this section, we use  $std_{RT}$  and  $mean_{RT}$  collected from 1000 random tours and  $L$

Table 1

The  $mean_{RT}$  and  $std_{RT}$  model on TSP instances

Instance set	Variable coefficients		Constant	Adj. $R^2$	MAPE
	$mean_{RT}$	$std_{RT}$			
Uniform	0.718**	−4.529***	2.969***	0.985	5.924%
Arcsine	0.629***	−5.898***	5.383***	0.977	6.612%
Normal	0.995***	−0.279***	0.173	0.989	4.298%
Multiplication	0.639***	−3.019***	3.570***	0.962	5.383%
Substitution	0.499***	2.533	−1.561	0.967	5.227%
Truncated multiplication	0.733***	−4.065***	2.738***	0.969	5.010%
Multiplication and substitution	0.658***	−3.193***	3.480***	0.962	5.636%
Normally distributed distance	0.954**	−13.704***	193.875	0.992	5.656%

\*\*\*  $p < 0.01$ , \*\*  $p < 0.05$ , \*  $p < 0.1$ .

solved using the Lin–Kernighan–Helsgaun (LKH) (Helsgaun, 2000) heuristic to study the results of fitting Equation (3) to different TSP instances:

$$L = a_1 \times mean_{RT} + b_1 \times std_{RT} + c_1. \quad (3)$$

It is worth noting that we do not apply log transformations in this model. In fact, we did an experiment with a variant of Equation (3) with a log transformation applied to each of the three variables, and we found that the transformation improved the predictive results when the dataset had a wider range in  $N$ . However, such a log transformation is not necessary for the experiments included in this paper. Table 1 provides the fitted coefficients and the predictive results of Equation (3) on some test instance sets. Uniform, Arcsine, and Normal are three sets of Euclidean TSP instances with sizes  $N = 10, 11, \dots, 99$ , and the name of each describes its distribution of vertices on the unit square. The multiplication set includes uniformly distributed instances with size  $N = 30, 40, \dots, 100$  and distance defined by random multiplication for  $\epsilon = 0.1, 0.2, \dots, 0.9$  as follows. For each pair of distinct vertices  $v_i$  and  $v_j \in V$ :

$$d_{rm}(v_i, v_j) = d_{EUC}(v_i, v_j) * a_{i,j},$$

where  $d_{EUC}(v_i, v_j)$  is the Euclidean distance between two vertices  $v_i$  and  $v_j$  and where  $a_{i,j}$  are i.i.d. random variables from a uniform distribution  $U(1 - \epsilon, 1 + \epsilon)$  for fixed  $\epsilon$ . Similarly, we define the non-Euclidean distance by random substitution between two vertices  $v_i$  and  $v_j$ :

$$d_{rs}(v_i, v_j) = \begin{cases} d_{EUC}(v_i, v_j) & \text{with probability } 1 - \theta \\ d_{EUC}(v_m, v_n) & \text{with probability } \theta. \end{cases}$$

That is to say, we randomly replace the edge length connecting two vertices with a random (other) edge length from the graph with probability  $\theta$ . The substitution set includes uniformly distributed instances of the same size,  $N = 30, 40, \dots, 100$ , and with distance defined by random substitution for  $\theta = 0.1, 0.2, \dots, 0.9$ . In the same way, we generate random multiplication instances, we replace all edge lengths that are above 0.8 with 0.8, and we call these instances the truncated multiplication

Table 2  
The  $std_{RT}$  model on TSP instances

Instance set	Variable coefficients			
	$\ln(std_{RT})$	Constant	Adj. $R^2$	MAPE
Uniform	1.552***	2.091***	0.881	14.265%
Arcsine	1.642***	1.584***	0.921	12.079%
Normal	1.717***	3.503***	0.950	10.840%
Multiplication	0.873***	2.198***	0.380	25.827%
Substitution	1.777***	1.870***	0.932	7.922%
Truncated multiplication	1.458***	2.092***	0.633	22.222%
Multiplication and substitution	1.052***	2.182***	0.483	23.833%
Normally distributed distance	−0.526***	10.295***	0.667	30.008%

\*\*\*  $p < 0.01$ , \*\*  $p < 0.05$ , \*  $p < 0.1$ .

set. The second to the last row of Table 1 contains all instances included in the random multiplication and random substitution sets. In the last row, instances with normally distributed distances are instances that do not have vertices located in an  $x$ – $y$  coordinate system; rather, we generate non-diagonal entries in the distance matrix according to a normal distribution  $\mathcal{N}(100, \alpha)$ . We generate 50 instances in this set with size  $N = 50$  and  $\alpha$  ranging from 0, 1, ..., 49. Thus, the instances with normally distributed distances do not have any metric structure.

Using these normally distributed distance matrices, the objective is to find the optimal TSP tour length. In fact, in addition to  $std_{RT}$  and  $mean_{RT}$  predictors, we also tried including the minimum random tour length as another predictor. However, we found that the minimum random tour length predictor is almost always not significant and is correlated with a linear combination of  $std_{RT}$  and  $mean_{RT}$ , so we excluded this predictor from further consideration. For example, the multiplication set in Table 1 has the lowest adjusted  $r$ -squared value. When we add the minimum of 1000 random tour lengths as another predictor, the adjusted  $r$ -squared value only increases by 0.001 to 0.963 and the MAPE decreases to 5.194%. We also note that the  $p$ -value for the minimum random tour length predictor is 0.209. Hence, we do not include it as a predictor in our model.

For comparison, we also include the results of model (4), which comes from our previous paper (Kou et al., 2022). In Table 2, we present the fitted coefficients for  $\ln(std_{RT})$  and the constants as well as the predictive results of model (4) on the test instance sets from Table 1:

$$\ln(L) = b_2 \ln(std_{RT}) + c_2. \quad (4)$$

Combining the results from Tables 1 and 2, we observe that using model (4), we can predict the value of  $L$  reasonably well when TSP instances are Euclidean. On the other hand, for non-Euclidean instances, using only  $std_{RT}$  as a predictor seems to be insufficient, and adding the  $mean_{RT}$  predictor can always increase the  $r$ -squared value and decrease the MAPE. For instance sets that use  $std_{RT}$  alone and yield relatively poor results, like the multiplication set, the addition of the  $mean_{RT}$  predictor boosts the predictive results.

We also note that  $L$  is positively correlated with  $mean_{RT}$  and negatively correlated with  $std_{RT}$ , which is intuitive since a higher standard deviation in random tours means that there is a higher probability of finding shorter tours. In particular, we note that in the last row of Table 2, the

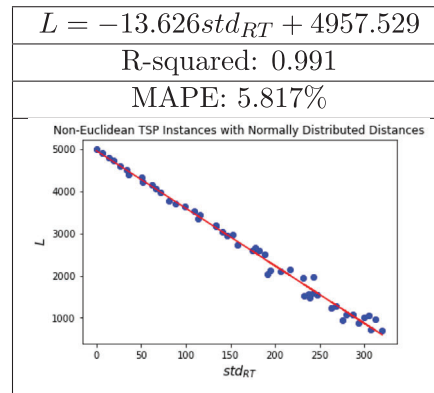


Fig. 1. Non-Euclidean TSP instances with normally distributed distances.

coefficient for the  $\ln(std_{RT})$  predictor is negative. In fact, if we do not take a log–log transformation in this case, the obtained regression model has an adjusted  $r$ -squared value of 0.991 and an MAPE of 5.817% as shown in Fig. 1. Recall that, for instances in this set, the nondiagonal entries of the distance matrices are generated using normal distributions  $\mathcal{N}(100, \alpha)$  for different values of  $\alpha$ . With the expected random tour length being the same for all instances, larger  $\alpha$  leads to a higher probability of finding shorter edges to complete the optimal tour. Thus, we can observe that  $L$  is negatively correlated with  $std_{RT}$ . In this case, adding the  $mean_{RT}$  predictor (see the last row of Table 1) improves the predictive result only slightly. On the other hand, for the remaining instance sets, using only the  $mean_{RT}$  instead of the  $std_{RT}$  predictor yields better predictions. Thus, to form a model that can generalize to different types of instances, it is necessary to include both predictors. For the substitution set, the only case in which the coefficient for  $std_{RT}$  is positive in Table 1, the  $std_{RT}$  predictor in the model is not significant.

In the Appendix, we demonstrate that random feasible solutions to the MST and MWM problems can, similarly, be used to estimate the optimal solution values for these easier combinatorial optimization problems.

### 3. Mean and standard deviation for the VRP

The VRP is another extensively studied combinatorial optimization problem. Unlike in the TSP, multiple vehicles are allowed to visit vertices, but each vertex must be visited exactly once. In the VRP, vehicles have a fixed capacity and the objective is to find the minimum total distance required to visit all vertices subject to the capacity constraint. Using the modified Clarke and Wright algorithm from Sinha Roy et al. (2022), we can easily generate numerous feasible solutions to VRP instances. In contrast, to our other experiments, for which we find random feasible solutions, these feasible solutions are random heuristic solutions. Instead of strictly considering the three best savings probabilistically as in Sinha Roy et al. (2022), we consider either the three, four, or five best savings with equal probability when generating feasible VRP solutions using the modified Clarke and Wright algorithm. In addition and as in Sinha Roy et al. (2022), we apply Yellow's parameter



Table 3  
Different models on VRP instances

Fitted model	Adj. $R^2$	MAPE
$D = 1.013mean_{CW3} - 0.742std_{CW3} - 1107.384$	1.000	1.100%
$D = 1.013mean_{CW4} - 0.854std_{CW4} - 1094.554$	1.000	1.076%
$D = 1.012mean_{CW5} - 0.972std_{CW5} - 1076.734$	1.000	1.054%
$\ln(D) = -0.679 \ln(std_{CW3}) + 13.548$	0.167	36.283%
$\ln(D) = -0.813 \ln(std_{CW4}) + 14.342$	0.207	35.749%
$\ln(D) = -0.902 \ln(std_{CW5}) + 14.872$	0.237	34.963%
$D = [0.9 + \frac{0.364N}{C^2}] \sqrt{NA}$	N/A	15.687%

Table 4  
Summary statistics of  $mean_{CW_i}$  and  $std_{CW_i}$

Statistics	$N$	Mean	St. Dev	Min	Max
$mean_{CW3}$	162	18,865.493	9055.909	9022.304	56,631.051
$mean_{CW4}$	162	18,896.302	9051.016	9070.492	56,611.552
$mean_{CW5}$	162	18,927.513	9047.765	9155.715	56,641.617
$std_{CW3}$	162	314.247	86.008	117.016	551.531
$std_{CW4}$	162	322.537	82.049	126.022	520.775
$std_{CW5}$	162	328.957	80.477	129.914	518.488

in the calculation of savings. Let  $mean_{CW_i}$  and  $std_{CW_i}$  represent the mean and standard deviation in 1000 distances output by the modified Clarke and Wright algorithm when considering the best  $i$  savings, and then we can see how the mean and standard deviation work in predicting the optimal distance for the VRP, which is denoted by  $D$ .

For the VRP, we consider models such as (5) and (6) for  $i = 3, 4$ , and 5 and use obtained data to fit constant coefficients  $a_3, b_3, b_4, c_3$ , and  $c_4$ .

$$D = a_3 \times mean_{CW_i} + b_3 \times std_{CW_i} + c_3, \quad (5)$$

$$\ln(D) = b_4 \times \ln(std_{CW_i}) + c_4. \quad (6)$$

To understand the relationship between the mean and standard deviation of feasible heuristic solution values and  $D$ , we consider instances from the VRP study by Queiroga et al. (2021). We use 162 instances from the set with names starting with XML\_100\_111. These instances have a random uniform depot and customer positioning, unitary demands, and average route sizes ranging from  $U[3, 5]$  to  $U[25, 50]$ , and the optimal solutions to these instances are given. For each instance, 1000 solutions are obtained using the modified Clarke and Wright heuristic with three best savings, another 1000 solutions with four best savings, and another 1000 with five best savings. To be specific, in the case of considering the best  $i$  savings, routes are merged based on considering the largest  $i$  savings with equal probability  $\frac{1}{i}$ . Let  $C$  denote the capacity of the vehicle,  $A$  denote the convex hull area, and  $N$  denote the number of vertices. We compare the performance of different models in Table 3. In addition, we include some summary statistics of the  $mean_{CW_i}$  and  $std_{CW_i}$  predictors in Table 4 to help understand how the number of savings under consideration can influence the predictive model.

As we can observe from Table 3, models using both the  $mean_{CW_i}$  and  $std_{CW_i}$  predictors can predict the value of  $D$  reasonably well, while models using only the  $std_{CW_i}$  predictor cannot predict  $D$  well. Note that in Table 3, all predictors and constants except for the bold one are significant at  $p = 0.01$ . In addition, the predictive results improve slightly as we increase the number of best savings being considered. For comparison, the last row of Table 3 provides the result of a more traditional model on predicting the optimal VRP distance (Akkerman et al., 2020). The constant 0.364 in the formula is the fitted area shape constant, and since there is not a constant term here, we do not have an adjusted  $r$ -squared value for the last model for comparison. In addition, for this nonlinear model, we do not include the significance level for independent predictors. However, by comparing the MAPEs, we can observe that the  $mean_{CW_i}$  and  $std_{CW_i}$  model clearly outperforms the traditional model. We also note that generating 1000 feasible solutions for a VRP is not very time consuming. For reference, we code the modified Clarke and Wright algorithm in Python, and for instances with size  $N = 100$ , generating 1000 solutions takes an average of 13.67 seconds on an M1 chip with 8 GB RAM.

Since we observe good performance from our model, we extend this experiment to all 10,000 instances with known optimal solutions introduced by Queiroga et al. (2021). Additionally, we note from Table 4 that considering more savings can, in general, result in worse VRP heuristic solutions on average. The purpose of our experiment is not to solve the VRP instances optimally or near-optimally. In fact, we identify that the summary statistics of  $mean_{CW_i}$  and  $std_{CW_i}$  do not vary greatly for  $i = 3, 4$ , and 5. Thus, for simplicity, in all experiments below we consider the modified Clarke and Wright algorithm with the top three savings.

The 10,000 instances introduced by Queiroga et al. have a two-dimensional Euclidean distance metric and the same number of customers,  $N = 100$ , and the depot and customers have integer coordinates corresponding to points in a  $[0, 1000] \times [0, 1000]$  grid. Note that the 162 instances from the previous experiment constitute a subset of these instances. These 10,000 instances have either a random, centered, or cornered depot. The customer positioning is either random, clustered, or half random and half clustered. There are also seven different types of demands: unitary, small values with large variance, small values with small variance, large values with large variance, large values with small variance, variance depending on the quadrant, or demand composed of many small values and a few large values. There are also six different average route sizes ranging from very short to extremely long. Details of instance formulation can be found in Queiroga et al. (2021). The key point here is that although instances in this dataset share some similarities, they differ greatly in depot and customer distribution, demand, and average route size.

For this set of VRP instances, we test the performance of the models included in Table 3 as well as models including only  $mean_{CW_3}$  or  $std_{CW_3}$  to study the influence of both predictors. In addition to comparing these models with the one in the bottom row of Table 3, we add another model considering the following list of features: number of customers  $N$ , capacity of the vehicle  $C$ , perimeter of the convex hull  $P$ , area of the convex hull  $A$ , height of the enclosing rectangle  $H$ , width of the enclosing rectangle  $W$ , average distance between customers  $d_c$ , means of variances in customer latitude and longitude  $mean_{Var(x,y)}$ , average distance between the depot and all customers  $d_d$ , average demand  $\overline{demand}$ , and variance in demand  $V(demand)$ . This list of features comes from Akkerman et al. (2020). They built linear regression, random forest, and neural network models



Table 5  
Different models on 10,000 VRP instances

Fitted model	Adj. $R^2$	MAPE
$D = 0.975\text{mean}_{CW3} - 1.780\text{std}_{CW3} - 349.382$	0.999	1.808%
$\ln(D) = 0.337 \ln(\text{std}_{CW3}) + 7.934$	0.136	41.778%
$D = 41.312\text{std}_{CW3} + \mathbf{1.033 * 10^4}$	0.167	49.147%
$D = 0.968\text{mean}_{CW3} - \mathbf{505.370}$	0.998	1.942%
$D = [0.9 + \frac{0.406N}{C^2}] \sqrt{NA}$	N/A	38.083%
$D = -10.186C + 0.007A - 0.462P + 0.070W + 0.637H$ $-9.084d_c + 0.016\text{mean}_{\text{Var}(x,y)} + 21.679d_d + 147.258\text{demand}$ $+2.979V(\text{demand}) + \mathbf{3809.593}$	0.395	40.525%

using 38 features. After feature selection, 27 of the 38 were included in their linear model. On their VRP instance set, the linear model attains an adjusted  $r$ -squared value of 0.836 and a relative mean absolute error of 4.8%.

We compare six models on the 10,000 VRP instances in Table 5. The last model is based on the 11 features that are most important and easy to compute from Akkerman et al. (2020), mentioned earlier. We obtain this model by stepwise regression using backward elimination. Note that  $N = 100$  for these 10,000 instances, so the feature  $N$  is not included in the model after stepwise regression. From this table, we see that our  $\text{mean}_{CW3}$  and  $\text{std}_{CW3}$  model clearly outperforms the model in row 6 which is based on Akkerman et al. (2020).

From Table 5, we clearly observe that the model containing both the  $\text{mean}_{CW3}$  and the  $\text{std}_{CW3}$  predictors outperforms the rest. From rows 1 to 4, we note the improvement in prediction is mostly due to the additional  $\text{mean}_{CW3}$  predictor, not the model structure. Also, the model in row 1 not only outperforms the model using only the  $\text{std}_{CW3}$  predictor but also the classic model in row 5 and the model containing 10 instance features in the last row. Based on the models in rows 1 and 4, one may question whether  $\text{std}_{CW3}$  is really helpful in predicting  $D$ . As in Table 3, we indicate nonsignificant predictors (at  $p = 0.01$ ) in bold. In fact, for this instance set, although the improvement is not obvious, the  $\text{std}_{CW3}$  predictor is significant, with a  $p$ -value below 0.0001. In addition, in the next set of experiments, we observe that adding the  $\text{std}_{CW3}$  predictor can indeed improve the predictive result.

For 131 instances with known optimal solutions and size  $N > 50$  from the Capacitated Vehicle Routing Problem Library (CVRPLIB (Uchoa et al., 2017)), we perform similar experiments. These instances come from 10 different papers and have sizes ranging from  $N = 50$  to 469; thus, the generation of instances was far from homogeneous, and vertex locations come from different distributions. We compare the performance of the same models in Table 5 on instances in this set to verify that the  $\text{mean}_{CW3}$  and  $\text{std}_{CW3}$  model is still valid.

From Table 6, we observe that the model containing both the  $\text{mean}_{CW3}$  and the  $\text{std}_{CW3}$  predictors outperforms the others, with an adjusted  $r$ -squared value close to 1 and an MAPE around 2.5%. Again, based on the results in the first four rows, we see that both predictors are essential when estimating  $D$ . In contrast to the results in Table 5, we note that including the  $\text{std}_{CW3}$  predictor can greatly improve the performance of the model. To be specific, including the  $\text{std}_{CW3}$  predictor decreases the MAPE by around 7% as compared to the model with only  $\text{mean}_{CW3}$ . Lastly, for this set of experiments, the model in row 1 outperforms the classic model and the model with multiple instance features. As in Table 5, we indicate nonsignificant predictors (at  $p = 0.01$ ) in bold. To

Table 6  
Different models on 131 CVRPLIB instances

Fitted model	Adj. $R^2$	MAPE
$D = 0.971mean_{CW3} - 4.628std_{CW3} + \mathbf{14.498}$	1.000	2.541%
$\ln(D) = 1.084 \ln(std_{CW3}) + 4.645$	0.839	57.457%
$D = 173.574std_{CW3} + \mathbf{2534.214}$	0.468	237.903%
$D = 0.959mean_{CW3} - \mathbf{148.072}$	0.999	9.237%
$D = [0.9 + \frac{0.313N}{C^2}] \sqrt{NA}$	N/A	42.166%
$D = 126.300N - 44.977C - 0.078A + 28.250P - 65.581W$ $- 89.895H + 87.761d_c + 65.514d_d + 438.801\overline{demand}$ $+ 0.072V(\overline{demand}) - 13140$	0.658	375.240%

Table 7  
Different models on all VRP instances

Fitted model	Adj. $R^2$	MAPE
$D = 0.974mean_{CW3} - 1.809std_{CW3} - 318.584$	0.999	2.277%
$\ln(D) = 0.484 \ln(std_{CW3}) + 7.198$	0.258	46.339%
$D = 44.042std_{CW3} + 9890.980$	0.174	63.344%
$D = 0.967mean_{CW3} - 478.002$	0.998	2.632%
$D = [0.9 + \frac{0.390N}{C^2}] \sqrt{NA}$	N/A	38.191%
$D = 151.133N - 9.295C + 0.005A$ $22.331d_d + 125.492\overline{demand}$ $+ 0.008V(\overline{demand}) - 13860$	0.387	55.749%

conclude our experiment on VRP instances, although there is a strong linear relationship between  $D$  and  $mean_{CW3}$ , both predictors are essential in order to build a model with a high adjusted  $r$ -squared value and a low MAPE. This precise approximation is important for the VRP, for which fast and powerful heuristics such as LKH do not exist and for which it is time-consuming to compute the optimal solution.

To summarize our VRP experiments, we compare the six models in Tables 5 and 6 on the joint set of 10,000 VRP instances and 131 CVRPLIB instances. Note that this set includes all VRP instances we experimented with, and we observe from Table 7 that the  $mean_{CW3}$  and  $std_{CW3}$  model still outperforms the remaining models on this diverse set, with a significantly lower MAPE of 2.277%. Again, the model in the last row comes from backward stepwise regression on all VRP instances by using the adjusted  $r$ -squared value as the criteria. Thus, the predictors present differ slightly from the last row in Tables 5 and 6. In addition, all predictors in Table 7 are significant at  $p = 0.01$ .

#### 4. Conclusion

The TSP is expensive to solve optimally, so many researchers have studied how to approximate the optimal tour length. In a previous paper, by running regressions on the different sets of instances, we found that  $std_{RT}$  could be a reasonable predictor when instances are Euclidean or close to Euclidean. In the Appendix of this paper, we demonstrate that the strong power-law relationship

between the standard deviation in feasible solution values and the optimal solution value also holds for the Euclidean MST and MWM, but not for the non-Euclidean instances.

To increase the estimation accuracy, we add another predictor, which is the mean of random feasible solution values. The  $mean_{RT}$  predictor is used to predict optimal TSP tour length, and the new model including both  $std_{RT}$  and  $mean_{RT}$  yields good predictions for all test instances. The improvement in predictive results is quite obvious for some instances that cannot be reasonably predicted using only the  $std_{RT}$  predictor. We also show that there is a strong linear relationship between the mean and standard deviation in feasible solution values and the optimal solution value for the MST and the MWM, no matter whether the instance is Euclidean or not. (For non-Euclidean instances, we need both mean and standard deviation.) In addition, we find that the mean and standard deviation predictors can also be successfully applied to predict the optimal total distance in the VRP. This is noteworthy since VRPs are more difficult to solve than TSPs and there are numerous practical reasons for wanting to estimate optimal VRP distances. In future work, we might explore applications to other hard-to-solve problems.

## References

- Akkerman, F., Mes, M., Heijnen, W., 2020. Distance approximation for dynamic waste collection planning. In Lalla-Ruiz, E., Mes, M., Voß, S. (eds) *Computational Logistics*. International Conference on Computational Logistics 2020. Lecture Notes in Computer Science, vol 12433. Springer, Cham, pp. 356–370.
- Basel, J., Willemain, T.R., 2001. Random tours in the traveling salesman problem: analysis and application. *Computational Optimization and Applications* 20, 2, 211–217.
- Beardwood, J., Halton, J.H., Hammersley, J.M., 1959. The shortest path through many points. In *Mathematical Proceedings of the Cambridge Philosophical Society*, Vol. 55. Cambridge University Press, Cambridge, UK, pp. 299–327.
- Cavdar, B., Sokol, J., 2015. A distribution-free TSP tour length estimation model for random graphs. *European Journal of Operational Research* 243, 2, 588–598.
- Chazelle, B., 2000. A minimum spanning tree algorithm with inverse-Ackermann type complexity. *Journal of the ACM* 47, 6, 1028–1047.
- Chien, T.W., 1992. Operational estimators for the length of a traveling salesman tour. *Computers & Operations Research* 19, 6, 469–478.
- Daganzo, C.F., 1984. The length of tours in zones of different shapes. *Transportation Research, Part B: Methodological* 18, 2, 135–145.
- Duan, R., Pettie, S., 2014. Linear-time approximation for maximum weight matching. *Journal of the ACM* 61, 1, 1–23.
- Hagberg, A., Swart, P., Schult, D., 2008. Exploring network structure, dynamics, and function using NetworkX. Technical report, Los Alamos National Lab (LANL), Los Alamos, NM.
- Helsgaun, K., 2000. An effective implementation of the Lin–Kernighan traveling salesman heuristic. *European Journal of Operational Research* 126, 1, 106–130.
- Hougardy, S., Zhong, X., 2020. Hard to solve instances of the Euclidean Traveling Salesman Problem. *Mathematical Programming Computation* 13, 1, 51–74.
- Kou, S., Golden, B., Poikonen, S., 2022. Optimal TSP tour length estimation using standard deviation as a predictor. *Computers & Operations Research* 148, 105993.
- Kwon, O., Golden, B., Wasil, E., 1995. Estimating the length of the optimal TSP tour: an empirical study using regression and neural networks. *Computers & Operations Research* 22, 10, 1039–1046.
- Merchán, D., Winkenbach, M., 2019. An empirical validation and data-driven extension of continuum approximation approaches for urban route distances. *Networks* 73, 4, 418–433.

- Nicola, D., Vetschera, R., Dragomir, A., 2019. Total distance approximations for routing solutions. *Computers & Operations Research* 102, 67–74.
- Pettie, S., Ramachandran, V., 2002. An optimal minimum spanning tree algorithm. *Journal of the ACM* 49, 1, 16–34.
- Queiroga, E., Sadykov, R., Uchoa, E., Vidal, T., 2021. 10,000 Optimal CVRP solutions for testing machine learning based heuristics. AAAI-22 Workshop on Machine Learning for Operations Research (ML4OR).
- Reinelt, G., 1991. TSPLIB—a traveling salesman problem library. *ORSA Journal on Computing* 3, 4, 376–384.
- Sinha Roy, D., Golden, B., Masone, A., Wasil, E., 2022. Using regression models to understand the impact of route-length variability in practical vehicle routing. *Optimization Letters* 17, 1–13.
- Sutcliffe, P.J., Solomon, A., Edwards, J., 2012. Computing the variance of tour costs over the solution space of the TSP in polynomial time. *Computational Optimization and Applications* 53, 3, 711–728.
- Uchoa, E., Pecin, D., Pessoa, A., Poggi, M., Vidal, T., Subramanian, A., 2017. New benchmark instances for the capacitated vehicle routing problem. *European Journal of Operational Research* 257, 3, 845–858.
- Virtanen, P., Gommers, R., Oliphant, T.E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J., van der Walt, S.J., Brett, M., Wilson, J., Millman, K.J., Mayorov, N., Nelson, A.R.J., Jones, E., Kern, R., Larson, E., Carey, C.J., Polat, I., Feng, Y., Moore, E.W., VanderPlas, J., Laxalde, D., Perktold, J., Cimrman, R., Henriksen, I., Quintero, E.A., Harris, C.R., Archibald, A.M., Ribeiro, A.H., Pedregosa, F., van Mulbregt, P., 2020. SciPy 1.0: fundamental algorithms for scientific computing in Python. *Nature Methods* 17, 3, 261–272.

## Appendix: Mean and standard deviation for the MST and MWM

For a given undirected graph on  $N$  vertices, the MST is a set of  $N - 1$  edges that connects all vertices in the graph such that the sum of the edge lengths is minimized. The left side of Fig. A1 shows an example of the MST in two-dimensional Euclidean space, where the red solid edges are in the MST and the black dashed edges are not. In our experiments, we focus on complete graphs. According to Chazelle (2000), classic algorithms, like Prim's algorithm or Kruskal's algorithm, run in  $O(m \log N)$  time, which is  $O(N^2 \log N)$  for complete graphs. In addition, Chazelle (2000) and Pettie and Ramachandran (2002) both provided a best known upper bound of  $O(m\alpha(m, N))$  for their algorithms, where  $\alpha$  is the functional inverse of Ackermann's function. Applying either classic or faster algorithms, we can solve MST problems in polynomial time.

A matching in a graph is a set of pairwise nonadjacent edges, and the MWM considers a matching in which the sum of edge lengths is maximized. The right side of Fig. A1 shows an example of the MWM in two-dimensional Euclidean space, where the red solid edges are selected in the MWM.

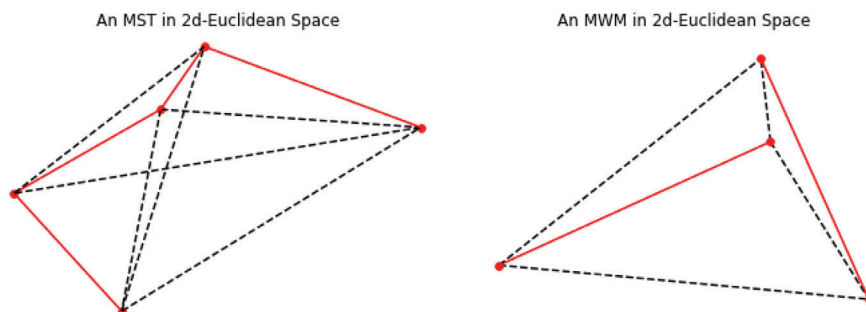


Fig. A1. Example of MST and MWM

Table A1  
The  $std_{ST}$  model on MST instances

Instance set	Variable coefficients			
	$std_{ST}$	Constant	Adj. $R^2$	MAPE
Uniform	1.055***	1.003***	0.922	8.130%
Euclidean	0.848***	1.062***	0.952	9.348%
Multiplication	−0.065	1.443***	0.002	30.348%
Substitution	0.949***	0.911***	0.802	7.214%
Joint Set	0.162	1.344***	0.018	24.314%
Normally Distributed Distance	−1.710***	16.512***	0.887	21.733%

\*\*\*  $p < 0.01$ , \*\*  $p < 0.05$ , \*  $p < 0.1$ .

According to Duan and Pettie (2014), the best known time complexity for the MWM problem is  $O(N^3)$  when a complete graph with noninteger weights is given.

Our experiment is designed as follows. The weight of the MST,  $T$ , is computed using the MST function from the Scipy package (Virtanen et al., 2020). For each instance, we generate 1000 random spanning trees as feasible solutions to the MST and then compute the mean and the standard deviation in their weights as  $mean_{ST}$  and  $std_{ST}$ . For MWM problems, we use the NetworkX package (Hagberg et al., 2008) to compute the maximum weight  $M$ , and 1000 random perfect matchings are considered as feasible solutions. We use  $mean_M$  and  $std_M$  to denote the mean and the standard deviation in the weights of perfect matchings. We use linear regression to fit models (A1) to (A4) below, and we find the numerical values for the constant coefficients  $a_1, a_3, b_1, b_3$ , and  $c_3$  for each MST instance set and constants  $a_2, a_4, b_2, b_4$ , and  $c_4$  for each MWM instance set:

$$\ln(T) = a_1 \times \ln(std_{ST}) + b_1, \quad (A1)$$

$$\ln(M) = a_2 \times \ln(std_M) + b_2, \quad (A2)$$

$$T = a_3 \times mean_{ST} + b_3 \times std_{ST} + c_3, \quad (A3)$$

$$M = a_4 \times mean_M + b_4 \times std_M + c_4. \quad (A4)$$

For the MST and MWM, we can solve for their optimal solutions easily with limited computing power. In addition, their instance structures are similar to those of the TSP and VRP, where we have either an  $x$ – $y$  coordinate system for the vertices or a given distance matrix. Hence, if we can show that the power-law relationship between the standard deviation of random solution values and the optimal solution value holds for these two problems, we will be more confident in answering yes to the following question: Does the standard deviation predictor work for optimization problems other than the TSP?

### Results for the MST

To compare the experimental results of models in the form of Equations (A1) and (A3), we summarize the experimental results and the values of fitted coefficients in Tables A1 and A2.

Table A2

The  $mean_{ST}$  and  $std_{ST}$  models on MST instances

Instance set	Variable coefficients			Adj. $R^2$	MAPE
	$mean_{ST}$	$std_{ST}$	Constant		
Uniform	0.054***	1.327***	1.057***	0.948	7.264%
Euclidean	0.030***	1.657***	0.816***	0.946	9.945%
Multiplication	0.161***	−2.666***	5.173***	0.931	7.842%
Substitution	0.085***	−0.508	2.927***	0.831	5.819%
Joint Set	0.138***	−2.143***	4.543***	0.861	8.529%
Normally Distributed Distance	1.508	−17.905***	−1198.605***	0.969	15.464%

\*\*\*  $p < 0.01$ , \*\*  $p < 0.05$ , \*  $p < 0.1$ .

Our uniform instances are 90 uniformly distributed instances on the unit square with size  $N = 11, 12, \dots, 100$ . For Euclidean instances in the second row of both tables, consider a set of Euclidean instances with size  $N = 11, 12, \dots, 100$  and  $x$ – $y$  coordinates of vertices that have either uniform, normal, or arcsine distributions. In this instance set containing 270 instances, the structural properties differ among instances. We can still observe relatively high  $r$ -squared values and low MAPE values when using  $std_{ST}$  to predict  $T$ .

We also experiment with non-Euclidean instances. First, we construct a set of instances with  $N = 30, 40, \dots, 100$  vertices uniformly distributed on a unit square and apply the non-Euclidean distance by random multiplication for  $\epsilon = 0.1, 0.2, \dots, 0.9$ . Next, we construct a set of instances with  $N = 30, 40, \dots, 100$  vertices uniformly distributed on a unit square and apply the non-Euclidean distance by random substitution for  $\theta = 0.1, 0.2, \dots, 0.9$ . Finally, we construct a joint set that contains instances from the two previous sets. In both Tables A1 and A2, these are denoted by multiplication, substitution, and a joint set, respectively. According to Table A1, for the random multiplication set, the performance of  $std_{ST}$  in predicting  $T$  is extremely poor. On the random substitution set, we observe a moderate to strong power-law relationship between  $std_{ST}$  and  $T$ . Not surprisingly, the result for the combined set is in between.

The previous experiments still consider instances with some metric structure. We also experiment on instances for which we do not locate vertices in an  $x$ – $y$  coordinate system; rather, we generate nondiagonal entries in the distance matrix according to a normal distribution  $\mathcal{N}(100, \alpha)$ . We generate 50 instances in this set with size  $N = 50$  and  $\alpha$  ranging from 0, 1,  $\dots$ , 49. The results for this set are provided in the last row of the two tables. For these instances without any geometric interpretation, the standard deviation predictor still provides reasonable predictive results. Combining the last two sets of experiments on non-Euclidean instances, we observe that the predictive result of  $std_{ST}$  is generally not as good as it is for Euclidean instances. In addition,  $std_{ST}$  can be a reasonable predictor for some sets, but not for all.

Based on Tables A1 and A2, we observe that adding the  $mean_{ST}$  predictor to the model can enhance predictive ability most of the time. The results are more obvious for the multiplication set and the joint set, for which the relationship between  $std_{ST}$  and  $T$  is barely observable. For the Euclidean instances in the first and second rows, using the model from Equation (A3) provides results either better than or comparable to those obtained using the model from Equation (A1). According to the results from Tables A1 and A2, for the MST,



Table A3  
The  $std_M$  model on MWM instances

Instance set	Variable coefficients			
	$std_M$	Constant	Adj. $R^2$	MAPE
Euclidean	1.084***	2.983***	0.901	18.867%
Multiplication	3.002***	1.007***	0.854	21.661%
Substitution	0.181***	2.112***	0.290	24.845%
Joint Set	0.870***	2.916***	0.294	25.099%

\*\*\*  $p < 0.01$ , \*\*  $p < 0.05$ , \*  $p < 0.1$ .

Table A4  
The  $mean_M$  and  $std_M$  models on MWM Instances

Instance set	Variable coefficients				MAPE
	$mean_M$	$std_M$	Constant	Adj. $R^2$	
Euclidean	1.450***	0.345**	−0.451***	1.000	1.417%
Multiplication	1.324***	2.446***	−1.245***	0.999	2.508%
Substitution	0.986***	1.314***	0.153	0.977	3.768%
Joint Set	0.993***	1.202***	0.153	0.979	3.622%

\*\*\*  $p < 0.01$ , \*\*  $p < 0.05$ , \*  $p < 0.1$ .

we can observe a strong correlation between the feasible solution values and optimal solution value.

### Results for the MWM

Similar to our experiments on MST instances, we test the predictive ability of  $std_M$  on  $M$  and the combination of  $mean_M$  and  $std_M$  on  $M$  in different sets of MWM instances. As mentioned earlier, we use linear regression to generate the constants in Equations (A2) and (A4) on different instance sets. The fitted coefficients and results of predicting  $M$  using only the  $std_M$  predictor are summarized in Table A3. The fitted coefficients and results of the model including both  $mean_M$  and  $std_M$  are shown in Table A4.

For each of the following instance sets, we generate each set of instances with size  $N = 20, 22, \dots, 100$  for the purpose of generating perfect matchings. We first consider Euclidean instances with vertices either uniformly distributed on a unit square or with a bivariate normal distribution  $\mathcal{N}\left(\begin{bmatrix} 0.5 \\ 0.5 \end{bmatrix}, \begin{bmatrix} 0.01 & 0 \\ 0 & 0.01 \end{bmatrix}\right)$ . These would be the instances in the Euclidean set in row 1 of Tables A3 and A4, and, as we observe, there is a strong power-law relationship between  $std_M$  and  $M$  in this set of Euclidean instances. However, the adjusted  $r$ -squared value is almost 1 when using both  $mean_M$  and  $std_M$  to estimate  $M$ .

Using the two definitions of non-Euclidean distances established with respect to the TSP, on unit squares, we construct a set of instances with non-Euclidean distances by random multiplication for  $\epsilon = 0.1, 0.3, \dots, 0.9$ , a set of instances with non-Euclidean distances by random substitution for

$\theta = 0.1, 0.3, \dots, 0.9$ , and a joint set containing instances from the two previous sets. In Tables A3 and A4, these sets are called multiplication, substitution, and joint set, respectively.

According to Table A3, for the random substitution set, the performance of  $std_M$  in predicting  $M$  is relatively poor. In the random multiplication set, we observe a moderate to strong power-law relationship between  $std_M$  and  $M$ . Also, while the result for the combined set is in between, we find that, in general,  $std_M$  can sometimes, but not always, do well as a predictor of  $M$ .

We notice that, in comparison to the results in Table A3, the model with the  $mean_M$  predictor added (see Table A4) has a higher adjusted  $r$ -squared value and lower MAPE on all sets; specifically, the adjusted  $r$ -squared value is always above 0.97 and MAPE is always below 4% when both predictors are included in the model. It is also interesting to note that the coefficients of the  $std_M$  predictor are always positive, which is reasonable given that the problem requires finding maximum weight instead of minimum weight values. We observe the relationship between the feasible solution values and optimal solution value for the MWM problem, and, in this case, based on the high adjusted  $r$ -squared values in Table A4, the linear relationship is strong.