

Research Article

Advancing Transportation Routing Decisions Using Riemannian Manifold Surfaces

Emre Tokgöz ¹, **Iddrisu Awudu**,² and **Saravanan Kuppusamy** ³

¹*School of Engineering, Quinnipiac University, Hamden, CT 06518, USA*

²*School of Business, Quinnipiac University, Hamden, CT 06518, USA*

³*Rohrer College of Business, Rowan University, Glassboro, NJ 08028, USA*

Correspondence should be addressed to Emre Tokgöz; emre.tokgoz@qu.edu

Received 2 September 2019; Revised 30 May 2020; Accepted 23 July 2020; Published 29 September 2020

Academic Editor: Yuchuan Du

Copyright © 2020 Emre Tokgöz et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

We consider several real-world driving factors such as the time spent at traffic signs (e.g., yield signs and stop signs), speed limits, and the topology of the surface to develop realistic and accurate routing solutions. Though these factors increase the complexity of modeling, they provide the flexibility to evaluate the routing solutions from different perspectives: cost, distance, and time, to name a few. First, we develop a set of algorithms based on the Riemannian manifold surface (RMS) to factor in the Earth's curvature to calculate distances. Second, we present a multiobjective, nonlinear, mixed-integer model (MINLP) that minimizes the distance traveled, time traveled, traveling costs, and time spent on traffic signs to design and evaluate the routes where the waiting times associated with traffic lights, stop signs, and yield signs are stochastic. Finally, we apply MINLP and RMS-based algorithms to a set of real-life and short- and long-distance transportation problems and analyze the results from computational experiments and discrete event simulations. We show that our approaches are on par with the state-of-the-art application, Google Maps, and yield realistic routing solutions that generate significant cost savings.

1. Introduction

Transportation is one of the essential activities in a supply chain. For example, goods manufactured at plant locations must be distributed to customer locations through warehouses, distribution centers, or cross-docking facilities. Typically, a fleet of trucks (owned or outsourced) performs the task of distributing goods. In the view of fuel prices, the complexity of the supply chain, and the uncertainties in the market place, the trucking operations must be efficient in order to keep costs under control. For instance, in 2019, US companies spent 668 billion dollars on trucking costs alone [1]. As a result, businesses are continually looking for a new approach or technique to reduce transportation costs. In product distribution, the sequence (or route) in which goods are delivered by a truck determines the distance traveled by the truck, which, in turn, determines the cost of transportation. The distance between locations is often used as a surrogate

measure for the transportation cost. It is therefore vital to understand how distances are calculated.

The assumptions related to distance calculations play a key role in determining the accuracy of the measurement. One of the assumptions is the surface on which the distance is measured. Often, the Euclidean surface is used because it is less complex to calculate distances [2]. However, the Euclidean surface generates inaccuracies in distance calculations because it does not take into account the curvature of the Earth. As a result, the use of the Euclidean surface has implications such as suboptimal routing decisions, resulting in increased transportation costs. Motivated by the above, we develop an approach based on the Riemannian manifold surface (RMS) that factors in the curvature of the Earth when calculating distances. For instance, the curvature of a Riemannian manifold surface indicates how much Earth's surface is curved in a local neighborhood. As the surface changes, the curvature changes. We use this curvature change to calculate the shortest path distances.

Subsequently, we demonstrate that the RMS-based approach improves the distance calculation, yields realistic distances, and generates significant transportation cost savings.

In this work, to factor in the curvature of the Earth, we use the generalized notion of a “straight line” to “curved spaces,” i.e., a *geodesic*. The calculation of geodesic distances involves the use of local topological regions on Earth’s surface. For example, while sprinting on multilane running tracks, a runner on the inner lane starts at the back, while a runner on the outer lane starts at the front. The starting points of the runners are determined by factoring in the geodesic distances, which are two dimensional as the running tracks are flat. However, driving a vehicle involves three-dimensional geodesic calculations that we use in this research.

Although companies use distance as a critical measure to evaluate the routing decisions, end consumers use various indicators (criteria) to determine routes for work and leisure travel. For instance, travel time, travel costs, travel distance, and waiting time spent at traffic signs (e.g., stop signs, yield signs, and traffic lights) are often used. Nevertheless, existing global positioning and global information systems do not provide, for example, the impact of traffic signals on travel cost and time and the flexibility to evaluate routing decisions using various criteria. Moreover, these systems do not consider local shortest geodesic distances. Inspired by the above, we develop a multiobjective, nonlinear, mixed-integer model (MINLP) where the end users can evaluate the routes with multiple criteria based on travel distance, cost, time, and traffic controls such as stop signs, traffic lights, and yield signs. Next, we illustrate the RMS-based approach and the MINLP model, and later, we perform simulation experiments using real-life instances.

1.1. An Illustration of Riemannian Manifold Surfaces and Multiple Criteria Route Evaluation. In this section, we illustrate two key concepts that are used in our methodology: (i) Riemannian manifold surface- (RMS-) based approach to calculate the distance; (ii) multiple criteria to evaluate the routes. For illustration purposes, we choose two points, Origin, O , and Destination, D . Let O and D be Hamden and New Haven, two cities in the state of Connecticut, for the ease of illustration as the authors are familiar with the geography. There are three routes between Hamden and New Haven: (A) via I-91 South—using highway; (B) via Whitney Avenue—using local roads; (C) via Ridge Road—using local roads.

To illustrate the concept of RMS, we focus on Route A. Figure 1 is a schematic representation of Route A. In this figure, the solid lines represent the boundaries of the road, and the road is divided into two lanes: Lane 1 and Lane 2. For instance, a vehicle traveling from the origin to the destination can either start from Lane 1 or 2. If the vehicle starts from Lane 1, switches to Lane 2 when the road curves, and stays in Lane 2 until it reaches the destination, the path traced by the vehicle is referred as the *innermost lane*. Conversely, if the vehicle starts at Lane 2, switches to Lane 1 when the road curves, and stays in Lane 1 until it reaches the

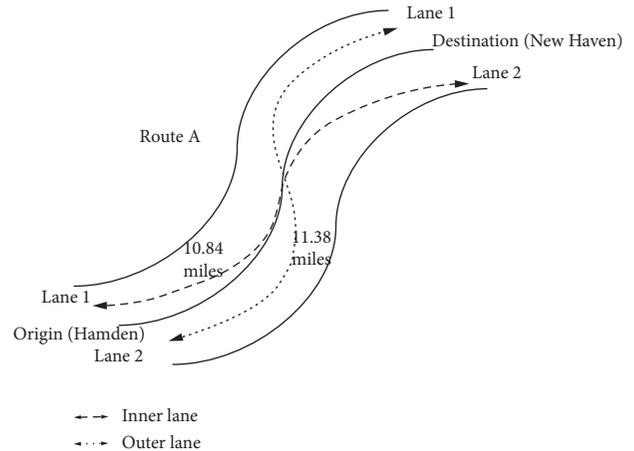


FIGURE 1: An illustration of RMS.

destination, the path traced by the vehicle is referred as the *outermost lane*. The dashed lines in Figure 1 represent the inner- and outermost lanes of the road. The distance traveled by the vehicle using inner- and outermost lanes is 10.84 miles and 11.38 miles, respectively. Note that both the inner and outer lanes are created based on the shortest and the longest geodesic paths, respectively, that takes the Earth’s curvature into account which is a central theme of the RMS-based approach (the details are provided later in the paper). The approach based on RMS yields savings of $11.38 - 10.84 = 0.53$ miles for Route A. Similarly, savings on Routes B and C are identified using RMS, and Table 1 summarizes the results. Note that when two locations are only a few miles apart, the impact of the Earth’s curvature on the distance calculated is very minimal. This can be seen in 1, as the difference between the outermost and innermost lanes is less than a mile for all routes. Later, in the numerical section, we show that the difference between the outermost and innermost lanes (i.e., distance savings) can be significant for long-distance routes. In such cases, it is feasible for a traveler to devise a driving plan to realize the savings in the distance.

Next, using the above real-world instance, we illustrate how a decision maker can use multiple criteria to evaluate routes A, B, and C. We consider the following criteria: distance traveled, waiting time at traffic signs, travel cost, the number of traffic signs, and travel time. Table 2 provides a summary of the evaluation of the routes, and the associated details are in Section 6. Note that for each criterion, the best values are marked in bold.

For instance, if the distance traveled or the traveling cost is used as a sole criterion, then Route B is the best. However, Route B is the worst when the time traveled is considered. In this case, both A and C are better than B. When the waiting time at traffic signs is considered, Route C is the best. For instance, Route A has 1 stop sign, 1 yield sign, and 16 traffic lights which totals to 18 traffic signs. Route B and C has a total of 29 and 21 traffic signs, respectively. If the number of traffic signs is used as a criterion, Route A is the best. The above illustration highlights the importance of using multiple criteria in route evaluation as the route selection varies with the criteria considerably.

TABLE 1: Summary for routes A, B, and C.

Routes	Distance-based outermost lanes (miles)	Distance-based innermost lanes (miles)	Savings (miles)
A	11.38	10.84	0.53
B	7.22	7.21	0.01
C	7.366	7.367	0.001

TABLE 2: Evaluation summary for routes A, B, and C.

Routes	Distance (RMS) (miles)	Waiting time at traffic signs (mins)	Travel cost	Total number of traffic signs	Traveling time (mins)
A	10.84	4.1	\$0.868	18	15.93
B	7.21	5.65	\$0.715	29	18.01
C	7.37	3.3	\$0.729	21	15.93

In summary, we develop a set of algorithms that are based on Riemannian manifold surfaces to address the need to calculate the distance accurately. In addition, we formulate a mathematical programming model that considers the real-world driving factors such as the time spent at traffic signs (e.g., yield signs and stop signs) and speed limits to evaluate routing decisions from multiple perspectives: cost, distance, and time, to name a few. Furthermore, we apply our approaches to real-life transportation problems and analyze the results from computational experiments.

The rest of this paper is organized as follows: Section 2 summarizes the relevant literature and the research contributions of this paper. Sections 3 and 4 offer details on the RMS-based approach to calculate the distance. Section 5 provides details on mathematical programming formulations that use multiple criteria to evaluate the routes. Section 6 summarizes computational experiments and simulation results. Section 7 concludes with a general discussion of managerial insights and avenues for future research.

2. Literature Review

In this section, we first present the literature related to the distance calculation since one of the main contributions of our research is in developing a new methodology to calculate distances. Next, we look into relevant literature where multiple criteria are used to evaluate routing solutions. Finally, we present our contributions in relation to the past work.

2.1. Distance Calculation Methodologies. In practice, there have been several software applications developed to calculate distances using travel information such as oncoming traffic situation, resting areas, and time to destination. However, the visibility into mathematical methodologies that are used in such applications is very minimal. Therefore, it is a challenging task to subject the methodologies used by software to rigorous academic discussion. On the contrary, in the academia, research related to distance calculation methodologies is limited. In the papers where the distance calculation is addressed, typically, rectangular or Euclidean distance functions are used. In their seminal work, Love and Morris [3] provide mathematical proofs for calculating transportation distances using convex functions that extend

beyond the rectangular or Euclidean distance calculations. Brimberg and Love [4] consider a new distance measure, the weighted one-two norm, which is a positive linear combination of the preceding norms as compared to the Euclidean or rectangular distance calculations. Fernández et al. [5] compare the weighted l_p -norm, which has been proven to be an accurate distance calculation function, to the l_{2b} -norm which uses different parameter estimates for the distance prediction. This methodology is extended by Brimberg and Love [4] and Üster and Love [6]. Other papers such as Von Hohenbalken and West [7]; Miyagawa [8]; Alpaydin et al. [9]; Griffith et al. [10]; Herica and Servio [11]; Havelock et al. [12] have all used different methodologies to calculate distances, from the Manhattan and Euclidean distances to Lobachevskian and p -norm methodologies.

2.2. Multiple Objective Routing Solutions. An extensive amount of research has been done in formulating methodologies to aid the decision maker with optimal routing decisions; see, Toth and Vigo [13] for a review on vehicle routing problems (VRP). We review only the research that considers multiple criteria/objectives to evaluate the routing solutions. For instance, Bowerman et al. [14] develop an approach to model the urban school bus routing problem that considers multiple criteria such as route length, number of routes, load balancing, length balancing, and student walking distance. Zografos and Androustopoulos [15] identify routing solutions that minimize both transportation time and risk as a part of a decision support system to transport hazardous materials and address emergency situations. Demir et al. [16] consider a variant of the pollution routing problem that minimizes both the fuel consumption and driving time. Kovacs et al. [17] extend the VRP from a service perspective by considering several objectives related to improving driver consistency, arrival time consistency, and cost minimization. Kumar et al. [18] consider a production and pollution routing problem with time windows that minimizes the total operational cost and emissions. See Jozefowicz et al. [19] for an extensive review on research related to multiobjective optimization in routing problems. Though our research uses criteria that are related to the total cost and the distance, it also considers the number of traffic signs and the time spent on these signs as criteria that have not been considered in the past research.

As mentioned above, one of our primary goals is to highlight the lack of research in using criteria related to traffic signs and to stress the significance of using multiple criteria when making routing decisions. Therefore, we restrict our attention to modeling multiple criteria and demonstrating the model's usefulness by applying our techniques to a few real-life instances. We note that our research does not involve designing heuristics for the routing problem. As future research, our work can be extended by employing approaches such as genetic algorithm-based techniques (e.g., [20, 21]), greedy heuristics with variable neighborhood search (e.g., [22]), learning-based heuristics (e.g., [23]), and fuzzy programming (e.g., [24]), to name a few, to solve large-scale, multiobjective, routing problems.

2.3. Research Contributions of This Paper. Our work is closely related to the study by Alpaydin et al. [9], Griffith et al. [10], and Havelock et al. [12]. Alpaydin et al. [9] propose a nonparametric approach using neural networks for estimating actual distances compared to the known distance estimators which are parameterized functions of the coordinates of the points. Griffith et al. [10] explore the non-Euclidean nature of network spaces, including Lobachevskian space (via the Poincaré disc), and compare the Euclidean, Manhattan, Minkowskian, and Lobachevskian distance calculation predicating parameters. The authors conclude that the skewed transportation networks in large-scale regional landscapes are characterized by a Lobachevskian geometry, whereas most smaller scale landscapes appear to be best characterized by a Minkowskian space whose parameters are between those of a Manhattan and Euclidean space. Havelock et al. [12] propose the use of learning-based strategies to solve the problem of distance calculation by using a stochastic method called the adaptive tertiary search (ATS) strategy. This methodology works by utilizing the information provided in the coordinates of the nodes to calculate the actual distances.

While we build on the previous work, our research differs in the following ways:

- (1) We develop a novel methodology based on the Riemannian manifold surface (RMS), a concept that uses the road network topology and factors Earth's curvature into account. As part of our methodology, we propose a set of algorithms where we adapt and modify the linked chain methodology (LCM) developed by Tokgöz et al. [25] to calculate distances.
- (2) We formulate a multiobjective, nonlinear, mixed-integer model (MINLP) to design and evaluate the routes where distance traveled, time traveled, traveling costs, and time spent on traffic signs are minimized. In this model, the waiting times associated with traffic lights, stop signs, and yield signs are stochastic.
- (3) We apply our approaches to a set of real-life and short- and long-distance transportation problems and analyze the results from computational

experiments and discrete event simulations. We show that our approach yields realistic solutions and generates significant cost savings compared to using the state-of-the-art applications (e.g., Google Maps).

3. Riemannian Manifold Surfaces and Linked Chain Method

Carl Friedrich Gauss proved *Theorema Egregium* (meaning remarkable theorem in Latin) in 1828 by establishing an important property of surfaces. The theorem states that the Gaussian curvature of a surface can be determined entirely by measuring distances along the paths on the surface, and the curvature does not depend on how the surface is embedded in a three-dimensional space. Later, Bernhard Riemann extended Gaussian theory to higher-dimensional spaces that allows distances and angles to be measured and the notion of curvature to be defined. The Riemannian approach is an intrinsic way to the manifold (a topological space that is locally Euclidean) and does not depend upon how the manifold is embedded in higher-dimensional spaces. A Riemann manifold M is a metric space on a smooth manifold with a smooth section of the positive-definite quadratic forms on the tangent bundle when the length of a continuously differentiable curve $\gamma: [a, b] \rightarrow M$ is defined by the following equation:

$$l(\gamma) = \int_{[a,b]} \left\| \frac{d\gamma(t)}{dt} \right\| dt. \quad (1)$$

The definition of length in (1) allows every connected Riemann manifold M to become a metric space on which the distance between a and b on M is defined as the infimum of $l(\gamma)$ when γ is a continuously differentiable curve joining a and b . Later, we use (1) to calculate the geodesic length which is instrumental in calculating route distances. One of the most popular use of the Riemannian manifold is in the development of general theory of relativity by Albert Einstein; in particular, his equations for gravitation are constraints on the curvature of space-time.

The curvature of a Riemannian manifold surface (RMS) indicates how much the surface is curved in a local neighborhood. For instance, the Riemann manifold is a plane in the Euclidean space when the curvature is zero; however, when the curvature changes from zero to one, the surface changes from a plane to a 3-dimensional sphere. This change in the curvature is important when calculating shortest path distances on Earth's surface. This surface is a natural manifold, and the pathways on this manifold are natural geodesics.

The research on RMS-based approaches in transportation is almost nonexistent except by Tokgöz et al. [25]. For instance, Tokgöz et al. [25] solved a location routing problem (LRP) on RMS and showed that their solution methodology generated results that are more realistic than using Euclidean surfaces since the Earth's surface has changing local RMS curvatures. The authors developed an approach called the linked chain method (LCM) to calculate geodesic distances. We adapt and modify the LCM approach

TABLE 3: Notations.

M	Manifold surface
O	Origin on M
D	Destination on M
M_0	Predetermined neighborhood of O - D that contains both O and D
$\max M_0 $	Maximum distance from the origin, O , to the boundary of M_0
r	Radius of LCM circles
r_0	An initially chosen constant radius determined by the decision maker (s)
γ_{ir}	The i^{th} geodesic identified with radius r
$C_{\gamma_{ir}}$	The LCM circle that contains the geodesic γ_{ir}
n_r	Number of LCM circles with radius r that covers M_0
ϵ	A constant that is used to increment the radius of a LCM circle
\mathcal{G}	Set of geodesics
\mathcal{P}	Set of all vectors with each vector containing a geodesic, length of the geodesic, and the LCM circle that contains the geodesic
\mathcal{N}	Set of linked list vectors formed by the geodesics

to our problem. Typically, in the LCM approach, given an origin (O) and destination (D), circles are generated to cover the focus area that contains the O - D pair. Unlike Tokgöz et al. [25], these circles are not randomly generated, and in addition, the circles represent the underlying road structure in our case. For instance, using the LCM, we generate inner- and outermost lanes for highways that provide a range of traveling distance which has not been addressed in the past research. Subsequently, the circles that cover geodesics are identified, and the length of the geodesics is calculated. The overlapping circles that contain geodesics are identified, and they are linked (hence, this approach is called the “linked chain”) to create a path that connects O and D . Next, we provide details on the linked chain method.

4. Algorithmic Solution for Shortest Path Geodesic Distance Calculation

In this section, we introduce four algorithms to determine the shortest path between two points: origin, O , and destination, D , that can be used for routing vehicles. In addition, there are two theorems and a corollary stated and proven by using isomorphism and LCM-related routing results. The set of notations used in the development of algorithms is given in Table 3.

We assume geodesics are determined based on the pathway intersections and illustrate this assumption by using a T-intersection. For instance, a T-intersection, see Figure 2, has three arms, and each arm could be a geodesic. Note that geodesics can be curved lines, but for illustration purposes, geodesics are shown as straight lines, see Figure 2. We assume that there exist nonoverlapping circles containing all the geodesics; these circles form the topology on the manifold. As a result, each circle can have only one geodesic. For instance, in Figure 2, three geodesics (three arms) are covered by three different circles. The radius of these circles is different due to varying lengths of the roads that form the T-intersection. Moreover, if a path uses the T-intersection, two circles of varying radii that cover two geodesics will be a part of this path. For instance, geodesics 1 and 2 could be one path and geodesics 1 and 3 can be another. Our first objective is to minimize the total distance

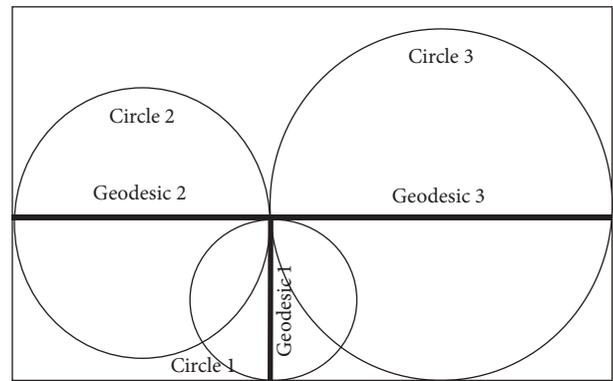


FIGURE 2: An illustration of geodesics using the T-intersection.

traveled which is calculated using (1) by replacing a and b with O and D ; i.e.,

$$\min z_1 = \min (l(\gamma)) = \int_{[O,D]} \left\| \frac{d\gamma(t)}{dt} \right\| dt. \quad (2)$$

4.1. Subroutine 1. Algorithm 1 shows the pseudocode for the first subroutine. First, a surrounding area M_0 within M of O and D is selected. Next, r is set to 0.1, and circles with radius r are drawn so that the entire surrounding area is covered with circles. Some of these circles may have geodesics in them and some may not. Once geodesics are identified for a radius r , i.e., γ_{ir} 's, they are added to a set \mathcal{G} . This process of identifying geodesics is repeated by incrementing r , i.e., $r = r + \epsilon$ where ϵ is set to 0.001. Note that a geodesic that is identified by multiple radii will be added only once to \mathcal{G} . As a result, geodesics in \mathcal{G} are not duplicated. Each one of these geodesics could have a minimum and a maximum length based on its curvature, and these lengths are calculated using Algorithm 4.

Figure 3 illustrates the steps involved in Algorithm 1. For illustration purposes, a focus area M_0 that is similar to a busy neighborhood in Manhattan, New York, is selected. In this illustration, the focus area is represented by a rectangle that has 8 intersections: four on the top row and four on the bottom row, see Figure 3(a). First, the entire M_0 is covered with the circles of initial radius r_0 . For the purposes of

```

Input: (a) coordinates of  $O$ ,  $D$ , and  $M_0$ ; (b) GIS map of the roads within  $M_0$  on  $M$ .
Run: (a)  $\mathcal{G} = \emptyset$ ; (b)  $r = r_0$ , where  $r_0 =$  initial radius; (c)  $\epsilon = 0.001$ .
(1) for  $r = r_0$  to  $\max|M_0|$  do
(2)   Cover  $M_0$  with LCM-linked circles with radius  $r$  by tracking all possible roads between  $O$  and  $D$ 
(3)   for  $i = 1$  to  $n_r$  do
(4)     Assign  $\gamma_{ir}$  to be the  $i^{\text{th}}$  geodesic covered for radius  $r$ 
(5)     Assign  $C_{\gamma_{ir}}$  to be the circle that covers  $\gamma_{ir}$ 
(6)     if  $\gamma_{ir} \notin \mathcal{G}$  then
(7)        $\mathcal{G} \leftarrow \mathcal{G} \cup \{\gamma_{ir}\}$ 
(8)     else
(9)        $r = r + \epsilon$ 
(10)    end if
(11)  end for
(12) end for

```

ALGORITHM 1: Geodesic search pseudocode.

clarity, the fully covered M_0 in Figure 3(a) is not shown; therefore, some of the areas in the figure are not covered. Next, the circles with geodesics are identified. In Figure 3(a), four geodesics are identified (on the top row) by four circles that are indicated by the dashed lines. We assign circles $C_{\gamma_{1r_0}}$, $C_{\gamma_{2r_0}}$, $C_{\gamma_{3r_0}}$, and $C_{\gamma_{4r_0}}$ to geodesics γ_{1r_0} , γ_{2r_0} , γ_{3r_0} , and γ_{4r_0} , respectively. Finally, the above geodesics are added to set \mathcal{G} . The above steps are repeated with a new radius, $r_0 = r_0 + \epsilon$. If geodesics are identified with the circles of new radii, they will be added to \mathcal{G} , otherwise, not.

Figure 3(b) illustrates the geodesics that are identified when the radius is equal to r_1 . In this figure, the circles that identify geodesics are represented by the dashed lines. We assign circles $C_{\gamma_{1r_1}}$, $C_{\gamma_{2r_1}}$, $C_{\gamma_{3r_1}}$, and $C_{\gamma_{4r_1}}$ to geodesics γ_{1r_1} , γ_{2r_1} , γ_{3r_1} , and γ_{4r_1} , respectively. We add the above geodesics to set \mathcal{G} . Figure 3(c) is an illustration of the focus area when all geodesics are identified. In this figure, 22 geodesics are identified, which are numbered from 1 to 22 by 22 different circles which are in \mathcal{G} .

4.2. Subroutine 2. Algorithm 2 shows the pseudocode for the second subroutine. We use this subroutine to calculate the length of geodesics, γ , that are in \mathcal{G} . Initially, two empty sets are created, \mathcal{G}^{new} and P . First, in set \mathcal{G} , we select a radius r and examine the geodesics, γ_{ir} , identified with that radius. Second, the length, $l(\gamma_{ir})$, of every geodesic is calculated using the following function:

$$l(\gamma_{ir}) = \int_{C_{\gamma_{ir}}} \left\| \frac{d\gamma_{ir}(t)}{dt} \right\| dt. \quad (3)$$

Third, we define $\hat{p}_{\gamma_{ir}}$ to be a vector with three elements: the geodesic γ_{ir} , the length of the geodesic $l(\gamma_{ir})$, and the circle $C_{\gamma_{ir}}$ that covers the geodesic. Subsequently, we add the geodesic γ_{ir} to set \mathcal{G}^{new} and delete the same geodesic from set \mathcal{G} . Finally, $\hat{p}_{\gamma_{ir}}$ is added to set P . The steps explained above are repeated until the distances for all the geodesics are calculated, i.e., until \mathcal{G} is empty. As an output from this

subroutine, we identify a set P that consists of vectors that contain the geodesics, length of geodesics, and unique circles that contain the geodesics.

4.3. Subroutine 3. In topology, geodesics are connected and form a path if circles that contain them are overlapping. This subroutine searches for overlapping circles to determine the paths within the focus area, M_0 . Algorithm 3 shows the pseudocode for the subroutine. Initially, an empty set, N , is created to store all possible paths, i.e., linked list vectors of geodesics. Then, two elements $\hat{p}_{\gamma_{ir}}$ and $\hat{p}_{\gamma_{jr}}$ of set P are chosen such that $i \neq j$. If circles C_{ir} and C_{jr} that contain γ_{ir} and γ_{jr} are overlapping, the link is added to set N ; otherwise, a different pair of \hat{p} is chosen. This process is repeated until all pairwise comparisons in P are exhausted. As a final output, a set N that consists of all possible paths is generated.

4.4. Subroutine 4. Algorithm 4 shows the pseudocode for the fourth subroutine. First, P is examined to identify the circles that contain origin, O , and destination, D . Once the circles are identified, they are split into nonoverlapping circles and geodesics. This is done to improve the accuracy of calculation of the distance between O and D . For instance, if C_o is the circle that contains O and γ_o is the geodesic in C_o , then C_o is split into $C_{o^{\text{right}}}$ and $C_{o^{\text{left}}}$ and γ_o is split into γ_o^{right} and γ_o^{left} . Subsequently, the corresponding vectors, $\hat{p}_{\gamma_o^{\text{right}}}$ and $\hat{p}_{\gamma_o^{\text{left}}}$, are added to P . Similarly, C_d , the circle that contains D , is split into $C_{d^{\text{right}}}$ and $C_{d^{\text{left}}}$ and γ_d is split into γ_d^{right} and γ_d^{left} and the corresponding vectors, $\hat{p}_{\gamma_d^{\text{right}}}$ and $\hat{p}_{\gamma_d^{\text{left}}}$, are added to P . We assume h number of pathways exist between O and D , and the set N_h , a subset of N , contains all those pathways. The length, $l(\gamma_h)$, of all pathways is calculated to determine all possible distances from O to D . Finally, the path with the least distance, l_{\min} , is selected to be the shortest path between O and D .

Note that steps 14–16 in Algorithm 4 are used to calculate minimum and maximum geodesic distances $l(\gamma_{ir})$ for

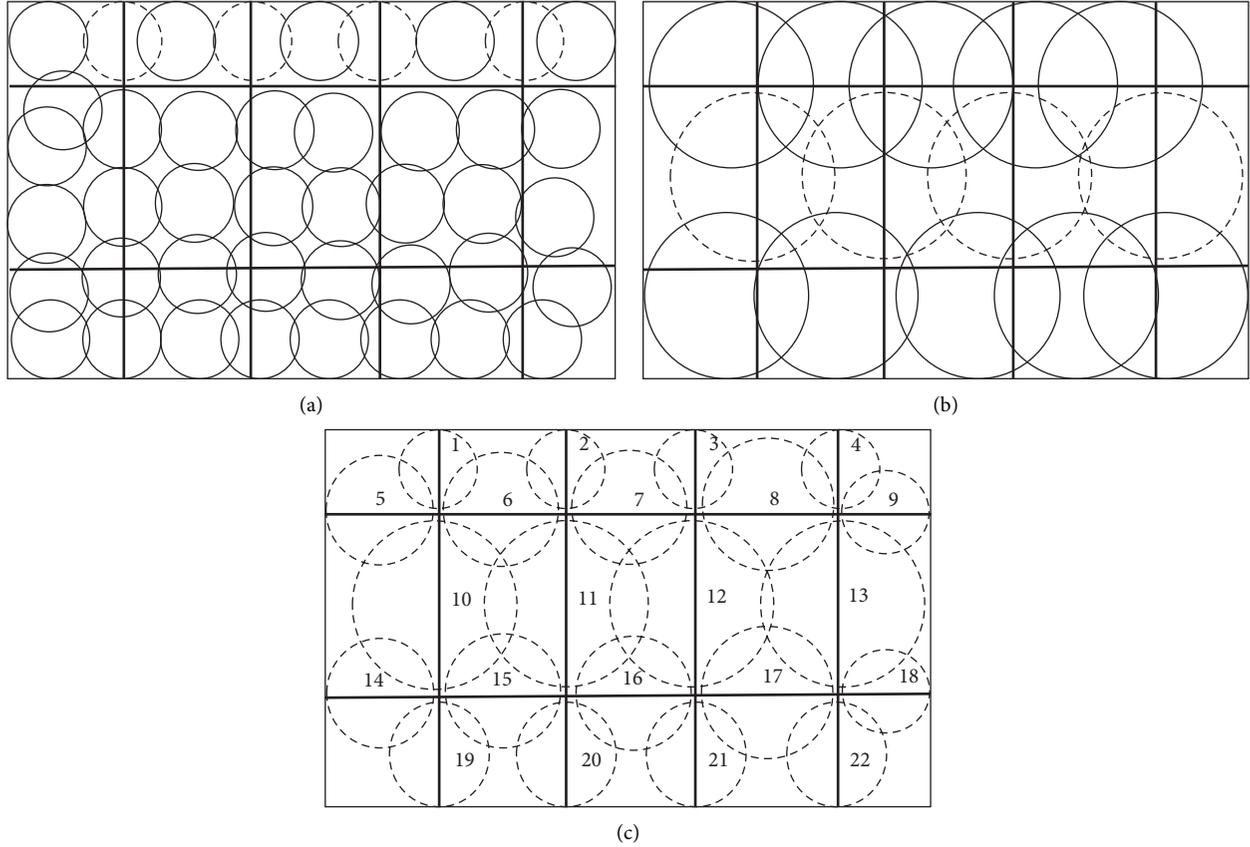


FIGURE 3: An illustration of Algorithm 1. (a) Focus area with $r=r_0$. (b) Focus area with $r=r_1$. (c) Focus area with multiple radii.

```

Run: Algorithm 1
Input: (a) set  $\mathcal{S}$  from Algorithm 1; (b)  $\mathcal{S}^{\text{new}} = \emptyset = P$ .
(1) for  $r = r_0$  to  $\max|M_0|$  do
(2)   for  $i = 1$  to  $n_r$  do
(3)     for  $\gamma_{ir}$  in  $\mathcal{S}$  do
(4)       Calculate  $l(\gamma_{ir})$ 
(5)       Define  $\hat{p}_{\gamma_{ir}} = (\gamma_{ir}, l(\gamma_{ir}), C_{\gamma_{ir}})$ 
(6)        $\mathcal{S}^{\text{new}} \leftarrow \mathcal{S}^{\text{new}} \cup \{\gamma_{ir}\}$ 
(7)        $\mathcal{S} \leftarrow \mathcal{S} - \{\gamma_{ir}\}$ 
(8)        $P \leftarrow P \cup \{\hat{p}_{\gamma_{ir}}\}$ 
(9)     end for
(10)  end for
(11) end for
(12) Assign  $\mathcal{S} = \mathcal{S}^{\text{new}}$ 
    
```

ALGORITHM 2: Geodesic identifier and distance calculation pseudocode.

all γ_{ir} from intersection to intersection which are then used to calculate the minimum and maximum length route between O and D . Algorithms 1–4 yield the following set of theorems.

Theorem 1. C and \mathcal{S} are isomorphic.

Proof. There exists a set of unique circles, C , that covers the set of geodesics within the given bounded region. One circle

can cover only one geodesic, and one geodesic can be covered by only one circle. It is evident from Algorithm 2 that there exists a unique circle per geodesic; therefore, there exists a map $f : C \rightarrow \mathcal{S}$ that is one-to-one. To show the converse, assume there are two circles in \mathcal{S} that correspond to the same geodesic. In this case, two circles must have the same overlapping intersections which would conflict with Algorithm 1 implementation. Therefore, f is an onto map; there exists a 1-1 and onto map that projects C to \mathcal{S} . The

```

Run: Algorithm 2
Input  $N = \emptyset$ 
(1) for  $i = 1$  to  $n_r$  do
(2)   while  $\widehat{P}_{\gamma_{ir}} \in P$  do
(3)     for  $j = 1$  to  $n_r$  do
(4)       if  $i \neq j$  and  $C_{\gamma_{ir}} \cap C_{\gamma_{jr}} \neq \emptyset$  then
(5)         Link  $\gamma_{ir}$  and  $\gamma_{jr}$  and add the link to  $N$ 
(6)         Continue updating  $N$  with links for  $\widehat{P}_{\gamma_{ir}} \in P$ 
(7)       end if
(8)     end for
(9)   end while
(10) end for

```

ALGORITHM 3: Geodesic path construction pseudocode.

```

Run Algorithm 3
(1) for  $\widehat{P}_{\gamma_{ir}} \in P$  do
(2)   if  $O \in C_{\gamma_{ir}}$  then
(3)     Assign  $\gamma_o := \gamma_{ir}$  and  $C_o := C_{\gamma_{ir}}$ 
(4)     Split  $C_o$  into  $C_o^{\text{right}}$  and  $C_o^{\text{left}}$  and  $\gamma_o$  into  $\gamma_o^{\text{right}}$  and  $\gamma_o^{\text{left}}$ 
(5)     Calculate  $l(\gamma_o^{\text{left}})$  and  $l(\gamma_o^{\text{right}})$ 
(6)     Define  $\widehat{P}_{\gamma_o^{\text{right}}} = (\gamma_o^{\text{right}}, l(\gamma_o^{\text{right}}), C_{\gamma_o^{\text{right}}})$  and  $\widehat{P}_{\gamma_o^{\text{left}}} = (\gamma_o^{\text{left}}, l(\gamma_o^{\text{left}}), C_{\gamma_o^{\text{left}}})$ 
(7)      $P \leftarrow P \cup \{\widehat{P}_{\gamma_o^{\text{right}}}, \widehat{P}_{\gamma_o^{\text{left}}}\}$ 
(8)   end if
(9)   if  $D \in C_{\gamma_{ir}}$  then
(10)    Repeat steps 3–6 by replacing  $O$  with  $D$ 
(11)     $P \leftarrow P \cup \{\widehat{P}_{\gamma_d^{\text{right}}}, \widehat{P}_{\gamma_d^{\text{left}}}\}$ 
(12)   end if
(13)   if  $\{\widehat{P}_{\gamma_o^{\text{right}}}, \widehat{P}_{\gamma_o^{\text{left}}}\} \cup \{\widehat{P}_{\gamma_d^{\text{right}}}, \widehat{P}_{\gamma_d^{\text{left}}}\} \in N_h \subset N$  then
(14)     Find the minimum and maximum of  $l(\gamma_{ir})$  calculated in Algorithm 2
(15)     Calculate the total length of the geodesics for all  $N_h$  using step 14
(16)      $l_{\min} = \min\{N_h\}$  and  $l_{\max} = \max\{N_h\}$  are the minimum and maximum length geodesics for all  $N_h$  that connect  $O$  and  $D$ 
(17)   end if
(18) end for

```

ALGORITHM 4: Path finder and distance minimizer pseudocode.

inverse map $f^{-1} : G \rightarrow C$ maps each γ_{ir} in G to $C_{\gamma_{ir}}$ in C . \square

Theorem 2. LCM determines the distance minimizing shortest path geodesic between two points chosen on a compact path-connected manifold region on Earth's surface.

Proof. A compact path-connected manifold region on Earth's surface indicates the existence of a path that allows the existence of geodesics between the two points chosen in a compact region. Algorithm 2 is used to determine all possible pathways between the two points, while Algorithm 4 is used to determine the distance minimizing shortest path on the compact path-connected predetermined region. The set N_h in Algorithm 4 is the set of all pathways on the manifold's surface. \square

Corollary 1. LCM applied on Earth's surface does not always yield the shortest path geodesics between the two points.

Proof. Assume there exists a geodesic that yields the shortest path between the two chosen points on Earth's surface. Such a geodesic would require to identify predetermined pathways with finite lengths and nonoverlapping loops; however, the looping nature of the manifold region could result in infinite distances. For instance, consider a region that only contains ocean. In this case, the chosen manifold's curvature is always 1; therefore, there exist infinitely many geodesics on this surface due to the looping nature on a spherical surface. Such a compact region of Earth's surface can generate infinite geodesics and infinite pathways. In this case, Algorithms 1–4 are not applicable. \square

5. A Multiobjective, Mixed-Integer, Nonlinear Mathematical Model

In this section, in addition to (2), we introduce four objective functions that minimizes (i) total time spent on traffic signs, (ii) total travel cost, (iii) total number of traffic signs, and (iv) total travel time. As the additional objective functions

involve traffic signs such as stop signs, traffic lights, yield signs, and the speed limits, we introduce the assumptions and parameters that are associated with them. In some cases, objective functions such as travel distance and time might be correlated, but having them as individual functions provide the flexibility for the decision maker. For instance, in the densely populated areas (e.g., northeastern parts of US), mostly travel time and distance are not correlated. Note that we do not model road congestion in our work which could be an avenue for future research; however, the impact of the network congestion is captured by the objective functions such as travel time and cost. The set of notations used in the mathematical model is given in Table 4.

5.1. Assumptions and Parameters

- (i) *Stop Signs*. There are four types of stop signs used for mathematical formulation: 1-way, 2-way, 3-way, and 4-way. Let η_j^u be the number of stop signs for the j -way stop on route u for all $1 \leq u \leq b$, where b is the total number of routes that connect O to D . The expected duration of a stop sign cannot be zero for lawful reasons; therefore, we assume that the wait time of a vehicle on the j -way stop sign, w_j , a stochastic parameter, varies between waiting times $\min\{w_j\}$ and $\max\{w_j\}$. The set of all stop signs is

$$K^u = \{K_1^u, K_2^u, K_3^u, K_4^u\}, \quad (4)$$

where

$$K_j^u = \{k_{j-1}^u + 1, k_{j-1}^u + 2, \dots, k_j^u\}, \quad (5)$$

represents the set of j -way stop signs k_{j-1}^u on route u for all $1 \leq u \leq b$ and $1 \leq j \leq 4$. For instance, a specific choice of a vehicle's wait time on a j -way stop sign can vary between waiting times $\min\{w_j\} = 3$ and $\max\{w_j\} = 3j + 3$ seconds.

- (ii) *Traffic Lights*. As one may have to wait at a traffic light longer than another (e.g., busy vs. nonbusy intersections), the maximum time spent can vary among traffic lights. Hence, it is reasonable to use the maximum time spent at traffic lights to categorize them. We categorize the traffic lights based on their maximum durations. Let the set

$$\Delta_i^u = \{\delta_{i-1}^u + 1, \delta_{i-1}^u + 2, \dots, \delta_i^u\}, \quad (6)$$

represent the category- i traffic lights based on β_i^u and the maximum amount of time that a vehicle can spend waiting on red-light duration on route u such that $1 \leq u \leq b$. For instance, Δ_1^{100} represents the 1st category of traffic lights that has a maximum duration of 25 seconds on the 100th route. The set of traffic lights for b different paths from O to D takes the form

$$\Delta^u = \{\Delta_1^u, \Delta_2^u, \dots, \Delta_e^u\}, \quad (7)$$

for all $1 \leq u \leq b$ and $e \in I_0$ where I_0 is the number of traffic light categories. The number of traffic lights on the u^{th} pathway in category i is represented by $|\Delta_i^u|$, and the total number of traffic lights on each path u from O to D is

$$\delta^u = \sum_{i=1}^e |\Delta_i^u|, \quad (8)$$

for all $1 \leq u \leq b$. For the example given above, if the set Δ_1^{100} has 5 elements, then $|\Delta_1^u| = 5$ indicates that there are 5 traffic lights with 25 seconds of maximum wait time on route 100. Let t_i^u represent the stochastic parameter that varies between 0 and β_i^u for waiting at the category- i traffic light on route u .

- (iii) *Yield Signs*. There are two types of traffic yields considered for a vehicle to join traffic: the yield signs to enter the highway traffic and the city traffic. Let y_1^u represent the number of yield signs to enter the highway and y_2^u represent the number of yield signs to enter the city traffic on route u for all $1 \leq u \leq b$. Let $y^u = y_1^u + y_2^u$ for each u , and let $\mu_{y_k}^u$ be the stochastic parameter representing the wait time at y_k^u for $k = 1, 2$. For instance, the average time it takes for a vehicle to enter a highway when there is a yield sign can be assumed to vary between 4 and 8 seconds, and the time it takes to enter the city traffic when there is a yield can be assumed to vary between 5 and 10 seconds on average.
- (iv) *Speed Limit*. The distance calculations depend on speed limit, geodesic distance, and traffic sign durations. The LCM (linked chain method) circles are used to determine the geodesic distances from one intersection to the other on each route. Each LCM circle contains the corresponding road's speed limit. We assume the set of speed limits to be

$$V = \{v_1, v_2, \dots, v_q\}, \quad (9)$$

where there are q different number of speed limits that could occur from O to D . If there are more than one in a LCM circle, then we consider an average of speed limits for time-related calculations. For instance, using geodesics γ_s from Algorithm 2 (subscript ir is modified as s) and the corresponding length $l(\gamma_s)$, the distance traveled on the LCM circle C_{γ_s} can be calculated as follows: $(l(\gamma_s))/v_l$, where $v_l \in V$ is the speed limit value on the road covered by C_{γ_s} . Later, we use this formula in the MIP formulations to determine the travel time from O to D .

5.2. Objective Functions and Constraints. In this section, we introduce five objective functions and develop three different formulations. Let γ_s^u represent the geodesic s on route u that

TABLE 4: A summary of model notations.

b	Number of routes that connects the origin (O) and the destination (D)
K^u	Set of 1-way, 2-way, 3-way, and 4-way stop signs on route u
K_j^u	Set of j -way stop signs on route u
η_j^u	Number of j -way stop signs on route u
w_j	Wait time at a j -way stop sign
I_0	Number of traffic light categories
Δ^u	Set of traffic lights on route u
Δ_i^u	Set of the i^{th} category traffic lights on route u
δ_i^u	Number of the i^{th} category of traffic lights on route u
δ^u	Total number of traffic lights of all categories on route u
t_i^u	Wait time at an i^{th} category traffic light on route u
y_1^u	Number of yield signs to enter highway on route u
y_2^u	Number of yield signs to enter city traffic on route u
$\mu_{y_k}^u$	Wait time at the y_k^{th} sign on route u
V	Set of q -types of speed limits from O to D
γ_s^u	Geodesic s on route u that connects O and D
Ψ^u	Set of all possible geodesics on route u
c_1^u	Fixed per-mile fuel cost on route u
c_2^u	Variable per-mile operational cost on route u
z_h	Objective function with criteria h , where $h=1$ (distance), $=2$ (wait time), $=3$ (cost), $=4$ (number of traffic signs), $=5$ (travel time), and $=6$ (weighted sum)
$\sigma_h^{(1)}$	Assigned weight for criteria h

connects O and D and $\Psi^u = \{\gamma_s^u\}_{s \in I_1}$ is a set of all possible geodesics between O and D with the index set I_1 . Let c_1^u and c_2^u be the fixed per-mile fuel cost and per-mile operational cost, respectively. The constrained, mixed-integer, nonlinear program formulation for the b -path problem to travel from O to D that includes z_1 can be formulated as follows:

$$\text{(total distance)} \quad \min z_1 = \int_{[O,D]} \left\| \frac{d\gamma(s)}{ds} \right\| ds, \quad (10)$$

$$\text{(total waiting time)} \quad \min z_2 = \sum_{i=1}^e t_i^u |\Delta_i^u| + \sum_{j=1}^4 w_j \eta_j^u + \sum_{k=1}^2 y_k^u \mu_{y_k}^u, \quad (11)$$

$$\text{(total cost)} \quad \min z_3 = c_1^u \int_{[O,D]} \|d\gamma_s^u\| + c_2^u z_2^u, \quad (12)$$

$$\text{(total number of traffic signs)} \quad \min z_4 = \sum_{i=1}^e |\Delta_i^u| + \sum_{j=1}^4 \eta_j^u + \sum_{k=1}^2 y_k^u, \quad (13)$$

$$\text{(total travel time)} \quad \min z_5 = \sum_s \frac{l(\gamma_s^u)}{v_l^u}, \quad (14)$$

$$\text{subject to} \quad \min\{\mu_{y_1}^u\} \leq \mu_{y_1}^u \leq \max\{\mu_{y_1}^u\}, \quad \forall y_1 = 1, \dots, y_1^u \text{ and } 1 \leq u \leq b, \quad (15)$$

$$\min\{\mu_{y_2}^u\} \leq \mu_{y_2}^u \leq \max\{\mu_{y_2}^u\}, \quad \forall y_2 = 1, \dots, y_2^u \text{ and } 1 \leq u \leq b, \quad (16)$$

$$\min\{w_j^u\} \leq w_j^u \leq \max\{w_j^u\}, \quad \forall j = 1, 2, 3, 4; k_{j-1}^u \in K^u \text{ and } 1 \leq u \leq b, \quad (17)$$

$$0 \leq t_i^u \leq \max\{\beta_i^u\}, \quad \forall i \in I_0 = \{1, 2, \dots, e\} \text{ and } 1 \leq u \leq b, \quad (18)$$

$$\gamma_s^u \in \Psi^u, \quad \forall s \in I_1 \text{ and } 1 \leq u \leq b, \quad (19)$$

$$Z_h = \{z_h^u\}_{u=1}^b, \quad \forall h = 1, 2, \dots, 5, \quad (20)$$

$$v_l^u \in V, \quad \forall l \in I_2 \text{ and } 1 \leq u \leq b. \quad (21)$$

Recall that algorithms from Section 4 are used to determine the shortest geodesic path, and the corresponding objective function z_1 minimizes the travel distance. We reintroduce z_1 in (10) for clarity. In this formulation, u is an integer variable that denotes the route. In addition, $\gamma(t)$ in (10) is a nonlinear, continuous variable that is used to calculate the geodesic distance on nonlinear curves. The waiting times associated with traffic lights, stop signs, and yield signs are stochastic parameters. Since the number of traffic signs on each route is well-known and do not vary, they are constant values. Equation (11) minimizes the total waiting time on traffic signals while traveling from O to D . Equation (12) minimizes the cost incurred (in dollars) to travel from O to D . Equation (13) minimizes the total number of traffic signs, and equation (14) minimizes the

total travel time. Equations (15) and (16) determine the waiting times at the city and highway yield signs, respectively. Equation (17) determines the waiting time at stop signs, and equation (18) provides the waiting times at the traffic lights. The set element specifications are given in (19)–(21). Equations (10)–(14) determine the minimum values of z_h^u for all h and u .

5.2.1. Weighted Sum Formulation (WSF). The weighted sum formulation (WSF) allows the decision maker to choose a route based on the weights assigned to the objective functions z_1 through z_5 . Function $f(\cdot)$ introduced below normalizes the objective function values z_1 – z_5 for each route, and it is used as a part of minimizing the weighted sum. This option of formulation requires weighted sum formulation of equations (15)–(19) by introducing the weight function $\sigma_h : A_1 \rightarrow \mathbb{R}^+$ with the corresponding objective function:

$$\min z_6 = \sum_{h=1}^5 \sigma_h^{(1)} f(z_h^u), \quad (22)$$

where A_1 is the finite index set and $f(z_h^u)$ are the normalized values of the objective functions z_h^u calculated for each route u by using the formula:

$$f(z_h^u) = \frac{z_h^u}{\sum_{j=1}^b z_h^j}, \quad (23)$$

for each $h = 1, \dots, 5$ and each u such that $1 \leq u \leq b$. The values of $f(z_h^u)$ are determined for all h and u that forms a set; this set of values is used for determining the minimum value of z_6 . The weights $\sigma_h^{(1)}$ are known constants that are assigned by the decision maker, and the sum of the weights must be equal to 100%.

5.2.2. Best Vector Set Solution (BV). In this section, we propose a best vector set solution approach where the vector set consists of the best solutions determined by minimizing each z_i , $\forall i = 1, \dots, 5$. Since the optimal route can vary with z_i , the corresponding vector set solution can be different. In this case,

$$\min(z_7^u) = (z_1^u, \dots, \min(z_i^u), \min(z_j^u), \dots, z_5^u), \quad (24)$$

where $i, j = 1, \dots, 5$, $u = 1, \dots, b$, b is the total number of the routes, and the best vector set $BV = \{\min(z_i^u)\}$ which is a subset of all solutions. For instance, assume $b = 4$. Let route 1 (i.e., $u = 1$) be optimal for z_1 and z_2 , route 2 ($u = 2$) be optimal for z_3 and z_5 , route 3 ($u = 3$) be optimal for z_3 , z_4 , and z_5 , and route 4 ($u = 4$) be not optimal for any z_i . The best vector (BV) solution is given as follows:

$$BV = \left\{ \begin{array}{l} (\min(z_1^1), \min(z_2^1), z_3^1, z_4^1, z_5^1), (z_1^2, z_2^2, \min(z_3^2), z_4^2, \min(z_5^2)), \\ (z_1^3, z_2^3, \min(z_3^3), \min(z_4^3), \min(z_5^3)) \end{array} \right\}. \quad (25)$$

Note in BV that there is no minimum objective function value for route 4 as it is not optimal for any z_i . The routes $u = 1, 2, 3$ in the BV solution may or may not be displayed in the corresponding vectors listed in the set. Based on the end user's preference, a specific route could be picked from the vector set. For instance, if the user is interested in saving time and money, a route (s) that minimizes both travel time and travel cost can be chosen from the vector set. Later, we illustrate the BV solution for two real-life routing problems in Section 6.

5.3. An Algorithmic Solution. In this section, we develop an algorithmic solution for the mathematical models presented in Section 5.2. Recall that equations (10), (12), and (14) are nonlinear since the distance calculations of geodesics depend on nonlinear curves. The numbers of traffic controls such as stop signs, yield signs, and traffic lights are known constants for each route. Therefore, the objective function in equation (13) is a constant for a route. Algorithm 5 uses the outputs of Algorithms 1–4 from Section 4.

In this algorithm, constraints (15)–(18) introduced in Section 5.2 are randomized by using the minimum and maximum values (that are constant values and assumed

to be known) for traffic signs. These constraint values are then used for calculating z_2 – z_5 . Values of σ_h are the inputs to the algorithm for calculating the weighted sum value z_6 . The values attained from z_1 – z_5 are then used for calculating the best vector set solution z_7 ; the decision maker can choose a route from a variety of routes by using z_7 that depends on the minimum values attained for z_1 – z_5 . A real-life application of Algorithm 5 with specific entries for minimum and maximum values is given in Section 6.

For instance, the set P which is an output of Algorithm 2 is a critical input to Algorithm 5. The set P consists of \hat{p}_{γ_s} vectors which has the following information: (i) the geodesic γ_s , (ii) the length of the geodesic $l(\gamma_s)$, and (iii) the LCM circle C_{γ_s} that has γ_s . The stochastic parameters are calculated by using a uniform distribution.

Note that running Algorithm 5 for sufficient number of times and averaging the outcomes will yield the desired objective function values. The highest and lowest values of the parameters presented in constraints (15)–(18) determine the minimum and maximum values for the objective function z_2 . The locations and numbers of traffic signs along with the speed limits and maximum durations of traffic lights are assumed to be known.

```

for  $u = 1$  to  $b$  do
  input the sets  $K^u$ ,  $\Delta^u$ ,  $\delta^u$ ,  $V$ ,  $y_1^u$ , and  $y_2^u$  and the output from Algorithm 4 (this input determines the sets  $\Psi^u$  and  $P$ )
  Input the costs  $c_1(u)$  and  $c_2(u)$ 
  Input the weights  $\sigma_h^{(1)}$  and  $\sigma_h^{(2)}$  for all  $h$ 
  for  $y_1^u = 1$  to  $p_1^u$  do (to calculate yield sign duration for a route)
    for  $y_2^u = 1$  to  $p_2^u$  do
       $\mu_{y_1}^u == \text{Rand}(\min\{\mu_{y_1}^u\}, \max\{\mu_{y_1}^u\})$ 
       $\mu_{y_2}^u == \text{Rand}(\min\{\mu_{y_2}^u\}, \max\{\mu_{y_2}^u\})$ 
    end for
  end for
  for  $j = 1$  to 4 do (to calculate stop sign duration for a route)
     $w_j^u == \text{Rand}(\min\{w_j^u\}, \max\{w_j^u\})$ 
  end for
  for  $i = 1$  to  $e$  do (to calculate traffic light duration for a route)
     $t_i^u == \text{Rand}(0, \max\{\beta(i, u)\})$ 
  end for
  for  $h = 1$  to 5 do (to calculate  $z_6$  and  $z_7$  after receiving user preferences)
    input  $\sigma_h^{(1)}$  and  $\sigma_h^{(2)}$ 
    calculate  $z_h$ 
    Sort  $z_6$  and  $z_7$  from min to max by using Quick sort algorithm
    display min  $z_1 - z_7$  values with the corresponding specific route  $u$  choices
  end for
end for

```

ALGORITHM 5: Optimization of objective functions $z_1 - z_7$.

6. Numerical Analysis and Simulation

In this section, we apply the methodologies from Sections 4 and 5 to two different real-life routing cases, short and long distances, and analyze the results from computational experiments and discrete event simulations. The first case includes three different short-distance routes that connect a location in Hamden (O) and another location in New Haven (D), two small cities located in the state of Connecticut. One of the routes is via a highway, while the other two routes are via local roads. For this O-D pair, we assume a travel pattern of two trips per day for all year. The second case includes three different long-distance routes that connect an origin and a destination in the states of Connecticut (CT) and California (CA), respectively. Note that the set of long-distance routes is chosen to mimic the actual truckload routes of a multinational stainless steel and specialty metal manufacturer located in the state of Connecticut. For this O-D pair, we assume a travel pattern of one-round trip per two weeks for all year. In both short- and long-distance routing cases, though there are many routes existing between O-D pairs, we restrict our attention to only three routes to demonstrate the theoretical results and the benchmark with the state-of-the-art application, Google Maps™. A summary of parameter values that are used in the computational experiments (unless otherwise stated, the parameter values are identical for both short- and long-distance routing cases) is given in Table 5.

Several types of traffic signals are present in each route: traffic lights (TL), yield signs (YS), and stop signs (SS). Furthermore, under the SS category, four different types of stop signs are considered. The time spent in waiting at the

traffic signals varies with the type. We assume minimum and maximum waiting times for each type of the traffic signals: traffic lights (15 seconds, 75 seconds), yield signs (3 seconds, 9 seconds), and j -way stop signs ($3j$ seconds, $3j + 3$ seconds) where j is a type of a stop sign for all $j = 1, \dots, 4$ [26]. The average speed limits are considered for every route. All three highways that connect CT and CA are assumed to have 70 miles per hour speed limit, while the city roads are assumed to be 35 miles per hour, and the highway that connects Hamden and New Haven is assumed to be 55 miles per hour.

In addition, fuel costs (c_1^u) and operational costs (c_2^u) that include maintenance, repair, and depreciation are used to calculate the traveling costs for each route. From the American Automobile Association (AAA), we assume the fuel price to be \$2.376 per gallon (AAA, 2019) and fuel consumption of 29 city/41 highway miles per gallon. We set the fuel consumption parameter, c_1^u , equal to \$0.0813 per mile (i.e., \$2.376/29). If a vehicle travels mostly on highways, then c_1^u is \$0.0579 per mile (i.e., \$2.376/41). To calculate the maintenance cost per mile, c_2^u , we use two data points: (i) the national, maintenance cost average for a new Sedan which is \$242 per year (AAA, 2019); (ii) annual miles traveled on an average which is 13,476 [27]. Using (i) and (ii), we set c_2^u equal to \$0.0179 (i.e., 242/13,476).

There are three scenarios considered for each evaluation criteria of the objective functions z_1 , z_2 , z_3 , and z_5 : best, worst, and average. The best and worst scenarios correspond to the objective function values when the minimum and maximum duration are used for waiting times at the traffic signals, respectively. The average scenario corresponds to the mean of the objective function values when the waiting times are allowed to vary randomly between the minimum and maximum. For instance, for each objective function, we

TABLE 5: A summary of parameter settings.

$b = 3$
$\eta_1^1 = 1, \eta_2^1 = \eta_3^1 = \eta_4^1 = 0$ (short-distance routing)
$\eta_1^2 = \eta_2^2 = \eta_3^2 = 0, \eta_4^2 = 3$ (short-distance routing)
$\eta_1^3 = 3, \eta_2^3 = 3, \eta_3^3 = 2, \eta_4^3 = 1$ (short-distance routing)
$\eta_1^1 = 3, \eta_2^1 = \eta_3^1 = 0, \eta_4^1 = 1$ (long-distance routing)
$\eta_1^2 = 2, \eta_2^2 = \eta_3^2 = 0, \eta_4^2 = 1$ (long-distance routing)
$\eta_1^3 = 1, \eta_2^3 = \eta_3^3 = 0, \eta_4^3 = 1$ (long-distance routing)
$w_j = U \sim [3j, 3j + 3], \forall j, j = 1, 2, 3, 4$
$I_0 = 1$
$\delta^1 = 16, \delta^2 = 21, \delta^3 = 11$ (short-distance routing)
$\delta^1 = 13, \delta^2 = 15, \delta^3 = 12$ (long-distance routing)
$t_i^u = U \sim [15, 75], \forall u, u = 1, \dots, 3$
$k = 1, 2$
$\mu_{y_k}^u = U \sim [3, 9], \forall u$ and k
$V = \{35, 55, 75\}$
$c_1^u = \$0.0579$ per mile
$c_2^u = \$0.0179$ per mile
$\sigma_h^{(1)} = 0.2, \forall h, h = 1, \dots, 5$

perform 1000 replications of simulation of the waiting times and collect the average statistics.

The best, worst, and average scenarios are different for z_1 ; Algorithms 1–4 are used to calculate the total distance values for z_1 . This calculation has two phases: for instance, in the first phase, for every geodesic and linked chain circle combination, the minimum length of the geodesic (i.e., the minimum distance to travel from intersection to intersection) is determined; in the second phase, using the minimum length geodesics, the shortest path from O to D is identified. The best case value of z_1 is calculated using the shortest path geodesic (i.e., inner lane) distance, while the worst-case value is calculated using the longest geodesic distance (i.e., outer lane) on each route.

Since z_3 includes the cost incurred due to waiting at traffic signals, the best- and worst-case values of z_3 correspond to the shortest and longest duration of waiting times at the traffic signals. Similarly, the waiting times at the traffic signals are included in the total travel time; as a result, best- and worst-case values of z_5 correspond to the shortest and longest duration of waiting times. Note that there are no best- and worst-case values for z_4 as the number of traffic signals does not vary on a given route. Next, we analyze the computational results from the short-distance case.

6.1. Short-Distance Routing: A Case Study. For demonstration, we choose two locations, a coffee shop in Hamden and the train station in New Haven, where Hamden and New Haven are two cities in the state of Connecticut. There are three routes between the two places: (A) via I-91 South—using highway; (B) via Whitney Avenue—using local roads; (C) via Ridge Road—using local roads. On each route, there is a certain number of traffic lights, yield signs, and stop signs.

Table 6 summarizes the computational results for the routes A, B, and C concerning criteria z_1 to z_6 . The best values are given in bold. We compare the results with Google Maps when the relevant information is available. We make the following observations:

- (1) When z_1 is considered, Route B is a good choice in all scenarios: best, worst, and average. The difference between the best and the worst cases is the highest for Route A which is 0.54 miles (11.38–10.84), about 5% per trip. In other words, if Route A is selected for traveling, using the innermost lanes can generate savings on the distance of about 5.
- (2) When z_2 is considered, Route C is a good choice in all scenarios. Though the simulated average values for all routes are close to each other, Route C has the minimum. Moreover, Route C has the minimum value in the best- and worst-case scenarios. The waiting time information is not available from Google Maps. Note that the simulated average values obtained for z_2 has a significant impact on total travel time.
- (3) Route B is the most cost effective when z_3 is considered. The simulated average value for Route B is \$0.727. When compared to Routes A and C, Route B generates an average cost savings of 0.185 and 0.166, respectively. With respect to z_4 , Route A is the best as it has a minimum number of traffic signs.
- (4) When z_5 is considered, Route C is a good choice in all scenarios. If the innermost lanes (i.e., best-case scenario) are used to travel, both Routes A and C can save travel time when compared to Google Maps of the order of 1 minute (i.e., 6%). In the average scenario, Route C is a good choice; however, Google Maps performs better with a travel time of 17 minutes. The difference between our results and Google Maps is due to the variability in the waiting times that impact the total travel time. Further, we note that when the total distance is considered, Route A (highway) is the worst choice; however, it does well when travel time is considered due to the minimum number of traffic signals and increased speed limits.
- (5) Route A is a good choice with a corresponding z_6 value of 0.18 when equal weights are assumed for all the criteria (i.e., $w_i = 0.2$ for all z_i where $i = 1, \dots, 5$). The priority vector solution, z_7 , for Routes A, B, and C is the set $BV = (10.84, 4.1, 0.868, 18, 15.93), (7.21, 5.65, 0.715, 29, 18.01),$ and $(7.366, 3.3, 0.0729, 22, 15.93)$. This set displays all the solution vectors and assist the decision maker(s) to examine the best options for all routes.

Overall, the above computational experiments for short-distance routes highlight the complexity of decision-making in evaluating routes when multiple criteria are considered. With criteria z_1 to z_6 , a commuter can pick and choose from several travel options based on individual preference. The geodesic distances calculated based on the RMS approach are on par with the state-of-the-art application, Google Maps. By including traffic sign duration and geodesic distances, our approach enhances the Google Maps' results. Next, we analyze the computational results from long-distance problem instances.

TABLE 6: Multiple criteria evaluation of short-distance routes.

Evaluation criteria	Scenario	Route A (highway)	Route B (local)	Route C (local)
z_1 (miles) <i>total travel distance</i>	Best	10.84	7.21	7.366
	Worst	11.38	7.22	7.367
	Average (simulation)	11.11	7.21	7.37
	Google Maps	10.90	7.20	7.60
z_2 (minutes) <i>total waiting at traffic signals</i>	Best	4.1	5.65	3.3
	Worst	20.3	27.45	15.4
	Average (simulation)	12.06	16.96	9.23
	Google Maps	N/A	N/A	N/A
z_3 (\$) <i>total travel cost</i>	Best	0.868	0.715	0.729
	Worst	0.927	0.738	0.741
	Average (simulation)	0.901	0.727	0.735
	Google Maps	N/A	N/A	N/A
z_4 (#) <i>number of traffic signals</i>	Stop signs	1	3	9
	Yield signs	1	5	2
	Traffic lights	16	21	11
	Total	18	29	22
	Google Maps	N/A	N/A	N/A
z_5 (minutes) <i>total travel time</i>	Best	15.93	18.01	15.93
	Worst	31.71	39.82	28.03
	Average (simulation)	24.32	28.53	21.97
	Google Maps	17	18	17
z_6 (minutes) <i>weighted time</i>	Best	0.18	0.19	0.19
	Worst	0.25	0.25	0.23

6.2. *Long-Distance Routing: A Case Study.* We choose an origin (O) and destination (D) located in the states of Connecticut (CT) and California (CA), respectively, and restrict our attention to three different long-distance routes that connect O and D : (A) via I-80 West; (B) via I-40 West; (C) via I-70 West. Note that all the above pathways are highway routes as they connect the east and west coasts of the US.

Table 7 summarizes the computational results for the routes A, B, and C with respect to criteria z_1 to z_6 . The best values are given in bold. We compare the results with Google Maps when the relevant information is available. The following are the observations extracted from Table 7:

- (1) Route A is a good choice in the best-case scenario (using innermost lanes) when z_1 is considered. On the other hand, in the worst (using outermost lanes) and average (simulation) scenarios, Route B is a good choice. When compared to Google's best, Route A in the best-case scenario which generates savings of 24 miles (i.e., $3003 - 2978.88$) per trip. Though the savings per trip is minimal, when a large fleet of trucks and hundreds of deliveries are considered, the total savings can be a considerable amount. Further, the difference between the best- and worst-case scenarios is at the maximum with Route A ($3029.88 - 2978.88 = 51$ miles). This difference impacts the total travel time, cost of traveling an additional 51 miles, and all the costs associated with the transportation (e.g., driver, truck maintenance, and fuel consumption costs).
- (2) Route C is the best choice in all scenarios when the total waiting time at traffic signs, z_2 , is considered.

Google Maps does not provide this information; therefore, no comparisons are made. When z_3 is considered, the difference between the simulation average values for all routes is minimal. However, in the long run, a small difference can make a big impact. For instance, Route B generates a savings of \$0.44 per trip (i.e., $295.96 - 295.52$) when compared to Route A. For a fleet of 1000 trucks and 50 trips per year, the total savings is equal to $0.44 * 1000 * 50 = \$22,000$. A similar comparison of Route B and Route C generates a savings of \$208,500. With respect to z_4 , both Routes A and C are the good choices as they have the minimum number of traffic signals.

- (3) Route A is the good choice in the best-case scenario when the total travel time, z_5 , is considered. However, in both worst and average cases, Route B is a good choice. Furthermore, Google Maps suggest Route A or C, while simulation averages indicate Route B is the best when the total travel time is considered. Note that the results from Google Maps are higher than our time estimates due to the difference in the speed limits that is used to calculate the travel times.
- (4) When equal weights are assumed for all the criteria (i.e., $w_i = 0.2$ for all z_i where $i = 1, \dots, 5$), any route is a good choice since all of them have a corresponding z_6 value of 0.16. Following is the set of priority vector solutions, z_7 , for Routes A, B, and C: $BV = (2979.8, 3.45, 293.489, 17, 3254.23)$, $(2986.54, 3.95, 294.18, 19, 3261.99)$, and $(3027.41, 3.25, 298.17, 17, 3305.88)$. This set displays all

TABLE 7: Multiple criteria evaluation of long-distance routes.

Evaluation criteria	Scenario	Route A (I-80)	Route B (I-40)	Route C (I-70)
z_1 (miles) <i>total travel distance</i>	Best	2978.88	2986.54	3027.41
	Worst	3029.33	3013.48	3058.40
	Average (simulation)	3005.08	3000.03	3043.23
	Google Maps	3008.00	3003.00	3043.90
z_2 (minutes) <i>total waiting at it traffic signals</i>	Best	3.45	3.95	3.25
	Worst	16.85	19.35	15.75
	Average (simulation)	10.28	11.85	9.52
	Google Maps	N/A	N/A	N/A
z_3 (\$) <i>total travel cost</i>	Best	293.49	294.18	298.17
	Worst	298.38	296.82	301.24
	Average (simulation)	295.96	295.52	299.69
	Google Maps	N/A	N/A	N/A
z_4 (#) <i>number of traffic signals</i>	Stop signs	4	3	2
	Yield signs	3	3	2
	Traffic lights	13	15	12
	Total	17	19	17
	Google Maps	N/A	N/A	N/A
z_5 (minutes) <i>total travel time</i>	Best	2556.92	2562.97	2596.49
	Worst	2609.45	2598.03	2625.4
	Average (simulation)	2585.40	2582.84	2617.80
	Google Maps	2640	2700	2640
z_6 (minutes) <i>weighted time</i>	Best	0.16	0.16	0.16
	Worst	0.16	0.16	0.16

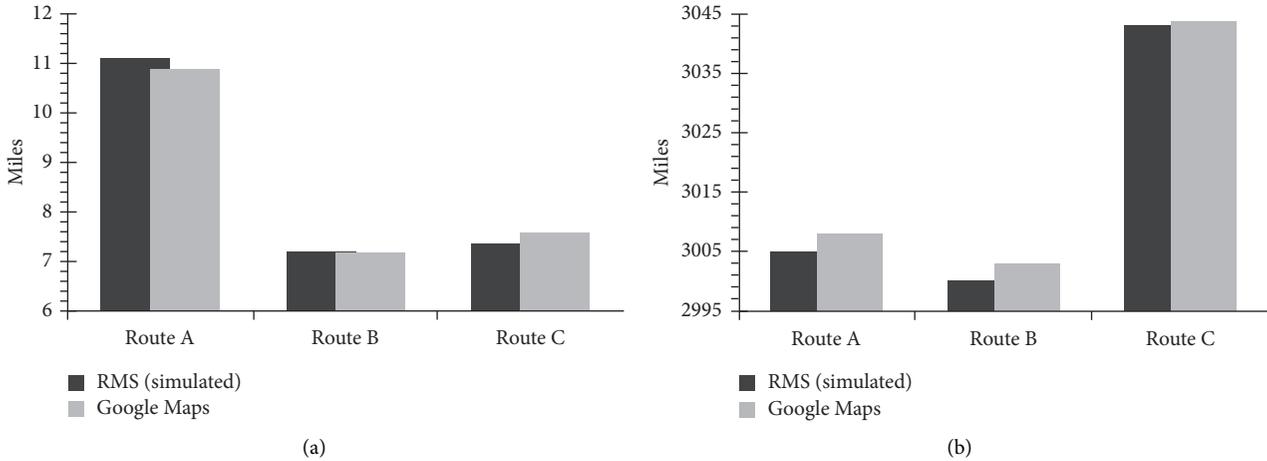


FIGURE 4: Distance comparison—simulation vs Google Maps. (a) Short-distance routing and (b) long-distance routing.

the solution vectors and assist the decision maker(s) to examine the best options for all routes.

In summary, the geodesic distances calculated based on the RMS approach with traffic signals considered for the routing solution outperform the state-of-the-art application, Google Maps.

Though our model does not consider the real-time traffic conditions and congestion effects, the results show that the geodesic distances calculated based on the RMS approach

can be helpful in enhancing the accuracy of the results from Google Maps, see Figure 4. For instance, using the innermost lanes (best case) on Route A, one can generate distance savings of 25 miles per trip when compared to the best result from Google Maps. Google Maps displays a single distance for traveling from the origin to destination, while the nature of the traffic roads suggests a distance to be traveled between two values: minimum and maximum geodesic distances. Though the savings may seem minimal, when the scope is increased from one trip to hundreds of trips per year and

from one truck to the entire fleet of trucks, the magnitude of the savings can be significant.

7. Conclusion and Future Research

Transportation is critical for both end consumers and the business world. One of the key components of the transportation is to identify the best route to travel that connects an origin and a destination. For instance, a route can be evaluated using multiple criteria such as travel cost and travel time. So, for some, the “best” route is cost effective, and for others, it is time sensitive. The approach to identify the routes, the assumptions, and the underlying calculations is important as they determine the accuracy of the routing solutions. In this research, we consider several real-world driving factors such as the time spent at traffic signs (e.g., yield signs and stop signs), speed limits, and the topology of the surface to develop realistic and accurate routing solutions. Though these factors increase the complexity of modeling, they provide the flexibility to evaluate the routing solutions from different perspectives: cost, distance, and time, to name a few.

In this research, we develop a nonlinear, multiobjective, mixed-integer mathematical model (MINLP) with five objectives that minimizes travel distance, travel time, travel cost, time spent on traffic signs, and the number of traffic signs to design and evaluate the routes between an origin and a destination where the waiting times associated with traffic lights, stop signs, and yield signs are stochastic. In addition, we present a weighted sum formulation (WSF) and a best vector set (BV) approach to assist in evaluating routing solutions. A set of algorithms are developed to solve the MINLP that are based on Riemannian manifold surfaces (RMS) to factor in the Earth’s curvature when calculating distances. A modified version of the linked chain method (LCM) is used to determine geodesic distances that are important in calculating the distances for possible routes between an origin and a destination. The geodesic distances determined in the first four algorithms based on RMS are used in the MINLP model to ensure that the routing solutions are realistic and accurate. For instance, using the LCM, we generate inner- and outermost lanes for highways that provide a range of traveling distances (instead of one distance) which has not been addressed in the past research.

We demonstrate the complexity of our algorithms and apply them in real-life instances. Two different real-life routing cases, short and long distances, for computational experiments and discrete event simulations are used for solving the MINLP and WSF and determining BV. The first case includes three different short-distance routes, and the second includes three long-distance routes. In both cases, we show that the geodesic distances calculated based on the RMS approach are on par with the state-of-the-art application, Google Maps. For instance, using the innermost lanes (best case) can generate savings on the distance traveled when compared to the best result from Google Maps. When a large fleet of trucks and hundreds of deliveries are considered, the total savings can be significant. In addition, we show that the MINLP, WSF, and BV assist decision-making in evaluating routes when multiple criteria

are considered. We believe that there are significant research opportunities for our work to be applied in the areas of traffic planning/management for routing decision strategies.

The results obtained in this work highlight the impact of geodesic distance calculations and the role of traffic signs in evaluating the routes which cannot be overlooked. We show that the waiting times at the traffic signs can be significant and play a critical role in evaluating the routes for short-distance transportation problems. For instance, Route B which uses the local roads to connect a coffee shop and a train station minimizes the total travel distance. But, the same route is a bad choice when the travel time is considered since the route has many traffic signs that create travel interruptions. On the other hand, for long-distance transportation problems, geodesic distance calculations play a major role. For instance, Route A, an actual truckload route of a multinational stainless steel and specialty metal manufacturer located in the state of Connecticut that follows the interstate highway I-80, has the minimum travel distance of 2,978.88 miles based on the shortest geodesic. When compared with the longest geodesic distance of 3029.33 miles, Route A generates a savings of 50 miles per trip per truck (1.7%) which can be significant when the total number of trips and the fleet size are considered.

Our research has the following practical implications: (i) it is difficult to obtain the realistic and accurate routing solutions from the available mapping software when they do not consider the nature of the Earth’s surface. Our approach that is based on the Riemannian manifold surface addresses this issue by considering the shortest path geodesics. (ii) Given the magnitude of the transportation spending, i.e., 0.91 trillion dollars in 2014, and the pervasiveness of the distance calculations, our approach can be applied in many different areas such as the supply chain (e.g., network design), transportation (e.g., vehicle routing), renewable energy (e.g., market distribution), and alternative technology (e.g., location-allocation of charging stations for electric vehicles), to name a few. (iii) Furthermore, our multicriteria mathematical model where we include the waiting times at the traffic signs to evaluate the routes based on distance, time, and cost addresses the lack of consideration of the traffic signs in the state-of-the-art applications (e.g., Google Maps).

Data Availability

A set of data is collected by using the distance calculator of Google Maps and locations readily available in Google Maps; therefore, the numerical results attained in this work as a part of the tables can be attained by other researchers by using Google Maps. The data set and the corresponding analysis can be shared by Dr. Emre Tokgoz via e-mail (Emre.Tokgoz@qu.edu) if requested.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

Professor Tokgoz was funded by the Quinnipiac School of Engineering for completing this research.

References

- [1] A. T. Kearney, *State of Logistics Report*, Penske Logistics, Washington, DC, USA, 2019.
- [2] J.-P. Rodrigue, *The Geography of Transport Systems*, Routledge, Taylor Francis, Abingdon, UK, 3rd edition, 2013.
- [3] R. F. Love and J. G. Morris, "Mathematical models of road travel distances," *Management Science*, vol. 25, no. 2, pp. 130–139, 1979.
- [4] J. Brimberg and R. F. Love, "A new distance function for modeling travel distances in a transportation network," *Transportation Science*, vol. 26, no. 2, pp. 129–137, 1992.
- [5] J. Fernández, P. Fernández, and B. Pelegrin, "Estimating actual distances by norm functions: a comparison between the $l_{k,p,\theta}$ -norm and the $l_{b_1,b_2,\theta}$ -norm and a study about the selection of the data set," *Computers & Operations Research*, vol. 29, no. 6, pp. 609–623, 2002.
- [6] H. Üster and R. F. Love, "Application of a weighted sum of order p to distance estimation," *IIE Transactions*, vol. 33, no. 8, pp. 675–684, 2001.
- [7] B. Von Hohenbalken and D. S. West, "Manhattan versus Euclid: market areas computed and compared," *Regional Science and Urban Economics*, vol. 14, no. 1, pp. 19–35, 1984.
- [8] M. Miyagawa, "First and second nearest distances in Archimedean tilings," *Forma*, vol. 29, pp. 5–11, 2014.
- [9] E. Alpaydin, İ. K. Altinel, and N. Aras, "Parametric distance functions vs. nonparametric neural networks for estimating road travel distances," *European Journal of Operational Research*, vol. 93, no. 2, pp. 230–243, 1996.
- [10] D. A. Griffith, I. Vojnovic, and J. Messina, "Distances in residential space: implications from estimated metric functions for minimum path distances," *GIScience & Remote Sensing*, vol. 49, no. 1, pp. 1–30, 2012.
- [11] S. L. Herica and G. B. Servio, "A method for modeling generalized distance functions," in *Proceedings of the IEEE International Conference on Service Operations and Logistics, and Informatics*, vol. 2, pp. 3075–3079, Beijing, China, October 2008.
- [12] J. Havelock, B. J. Oommen, and O.-C. Granmo, "Novel distance estimation methods using "stochastic learning on the line" strategies," *IEEE Access*, vol. 6, pp. 48438–48454, 2018.
- [13] P. Toth and D. Vigo, "An overview of vehicle routing problems," *The Vehicle Routing Problem, SIAM Monographs on Discrete Mathematics and Applications*, pp. 1–26, Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2002.
- [14] R. Bowerman, B. Hall, and P. Calamai, "A multi-objective optimization approach to urban school bus routing: formulation and solution method," *Transportation Research Part A: Policy and Practice*, vol. 29, no. 2, pp. 107–123, 1995.
- [15] K. G. Zografos and K. N. Androutsopoulos, "A decision support system for integrated hazardous materials routing and emergency response decisions," *Transportation Research Part C: Emerging Technologies*, vol. 16, no. 6, pp. 684–703, 2008.
- [16] E. Demir, T. Bektaş, and G. Laporte, "The bi-objective pollution-routing problem," *European Journal of Operational Research*, vol. 232, no. 3, pp. 464–478, 2014.
- [17] A. A. Kovacs, S. N. Parragh, and R. F. Hartl, "The multi-objective generalized consistent vehicle routing problem," *European Journal of Operational Research*, vol. 247, no. 2, pp. 441–458, 2015.
- [18] R. S. Kumar, K. Kondapaneni, V. Dixit, A. Goswami, L. S. Thakur, and M. K. Tiwari, "Multi-objective modeling of production and pollution routing problem with time window: a self-learning particle swarm optimization approach," *Computers & Industrial Engineering*, vol. 99, pp. 29–40, 2016.
- [19] N. Jozefowicz, F. Semet, and E.-G. Talbi, "Multi-objective vehicle routing problems," *European Journal of Operational Research*, vol. 189, no. 2, pp. 293–309, 2009.
- [20] T. George and T. Amudha, "Genetic algorithm based multi-objective optimization framework to solve traveling salesman problem," in *Advances in Computing and Intelligent Systems*, pp. 141–151, Springer, Singapore, 2020.
- [21] M. A. Mohammed, M. K. Abd Ghani, R. I. Hamed, S. A. Mostafa, M. S. Ahmad, and D. A. Ibrahim, "Solving vehicle routing problem by using improved genetic algorithm for optimal solution," *Journal of Computational Science*, vol. 21, pp. 255–262, 2017.
- [22] L. Delgado-Antequera, R. Caballero, J. Sánchez-Oro, J. M. Colmenar, and R. Martí, "Iterated greedy with variable neighborhood search for a multiobjective waste collection problem," *Expert Systems with Applications*, vol. 145, Article ID 113101, 2020.
- [23] A. Soeanu, S. Ray, J. Berger, A. Boukhtouta, and M. Debbabi, "Multi-depot vehicle routing problem with risk mitigation: model and solution algorithm," *Expert Systems with Applications*, vol. 145, Article ID 113099, 2020.
- [24] Y. Sun, "Green and reliable freight routing problem in the road-rail intermodal transportation network with uncertain parameters: a fuzzy goal programming approach," *Journal of Advanced Transportation*, vol. 2020, Article ID 7570686, 21 pages, 2020.
- [25] E. Tokgöz, S. Alwazzi, and T. B. Trafalis, "A heuristic algorithm to solve the single-facility location routing problem on Riemannian surfaces," *Computational Management Science Journal*, vol. 12, no. 3, pp. 297–415, 2015.
- [26] W. Wen, "A dynamic and automatic traffic light control expert system for solving the road congestion problem," *Expert Systems with Applications*, vol. 34, no. 4, pp. 2370–2381, 2008.
- [27] Bureau of Transportation Statistics, 2019, <https://www.bts.gov/content/us-vehicle-miles>.