

Article

# An Information Gradient Approach to Optimizing Traffic Sensor Placement in Statewide Networks

Yunxiang Yang , Hao Zhen and Jidong J. Yang \* 

Smart Mobility and Infrastructure Laboratory, College of Engineering, University of Georgia, Athens, GA 30605, USA; yyang117@uga.edu (Y.Y.); hao.zhen@uga.edu (H.Z.)

\* Correspondence: jidong.yang@uga.edu

**Abstract:** Traffic sensors are vital to the development and operation of Intelligent Transportation Systems, providing essential data for traffic monitoring, management, and transportation infrastructure planning. However, optimizing the placement of these sensors, particularly across large and complex statewide highway networks, remains a challenging task. In this research, we presented a novel search algorithm designed to address this challenge by leveraging information gradients from K-nearest neighbors within an embedding space. Our method enabled more informed and strategic sensor placement under budget and resource constraints, enhancing overall network coverage and data quality. Additionally, we incorporated spatial kriging analysis, harnessing spatial correlations of existing sensors to refine and reduce the search space. Our proposed approach was tested against the widely used Genetic Algorithm, demonstrating superior efficiency in terms of convergence time and producing more effective solutions with reduced information loss.

**Keywords:** traffic sensor location optimization; information gradient; spatial kriging; Node2Vec; embedding; K-nearest neighbors; genetic algorithm



**Citation:** Yang, Y.; Zhen, H.; Yang, J.J. An Information Gradient Approach to Optimizing Traffic Sensor Placement in Statewide Networks. *Information* **2024**, *15*, 654. <https://doi.org/10.3390/info15100654>

Academic Editors: Antonio Comi and Francis Cirianni

Received: 22 September 2024

Revised: 12 October 2024

Accepted: 14 October 2024

Published: 18 October 2024



**Copyright:** © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

Traffic sensors are vital components of Intelligent Transportation Systems, forming the backbone of modern traffic management, particularly in the era of emerging artificial intelligence (AI). These sensors enable real-time insights into traffic dynamics, optimizing traffic flow and significantly improving road safety. Moreover, the data they gather are essential for AI-driven tools, which harness predictive analytics and real-time decision-making to transform traffic management practices. This synergy between traffic sensors and AI technologies results in more efficient and responsive transportation systems, underlining the critical role that traffic sensors play in contemporary traffic management solutions.

While traffic sensors are essential, it is impractical to equip every road segment with permanent sensors due to resources and budget constraints. As road networks continue to expand, determining the optimal locations for sensor installation to maximize public benefits has become increasingly important. This challenge falls within the scope of the traffic sensor location problem (TSLP), which focuses on strategically selecting specific network links or nodes for sensor installation to effectively capture network-wide traffic flows. TSLP also plays a crucial role within the broader framework of Transport System Modeling (TSM) [1], which are essential for understanding mobility patterns and supporting the simulation, design, planning, and control of transport systems. Beyond traditional data, leveraging diverse data sources can significantly enhance TSM. For instance, Alonso et al. [2] combined floating car data (FCD) with traffic information from traditional loop detectors to develop fundamental diagrams for analyzing traffic conditions on urban road networks. In addition to FCD, other forms of probe data can also be incorporated. Combining insights from fixed-location sensors with mobile or probe sensor data can further enhance transport system modeling, enabling more informed decisions on network

operations, infrastructure investments, and congestion management. Furthermore, it is also important for TSLP to account for potential discontinuities induced by certain barriers [3].

It is worth noting that while TSLP could be formulated for different objectives, such as travel times estimation [4,5] or speed monitoring [6], the most common and important goal of the TSLP is to enhance the observability of traffic flows at the network level [7], which can be categorized in different ways, such as the flow information on specific link/path segments [8] or the O/D trips [9]. In this paper, we focused on the segment-level flow estimation across the entire state of Georgia, which belongs to the flow observability defined in [7].

Addressing the TSLP for large-scale networks presents a multifaceted challenge. Traditional approaches to addressing the TSLP aim to enhance flow observability by utilizing linear or nonlinear programming and network theory to formulate and solve the problem [7]. These methods involve setting up systems of equations based on the sensor data, with the observability determined by the rank of the matrix formed by these equations. However, applying this framework to sensor planning on real-world large networks presents significant challenges [10]. First, while TSLP is a multifaceted challenge complicated by factors such as sensor types, the likelihood of sensor failures, budgetary limitations, etc., the fundamental challenge for TSLP arises from the increasing size of road networks, resulting in escalated computational complexity. Formulating the problem within algebraic and network theory domains becomes exceedingly complex for large transportation networks. Second, solving the resulting large systems of equations is often computationally intractable, particularly when dealing with large-scale networks like those in metropolitan areas [10]. Consider a road network containing  $m$  road segments (or links); the search space is  $2^m$  [2]. If  $n$  sensors are to be deployed in this road network, the search domain is  $\frac{m!}{n!(m-n)!}$ , which would still be formidable for metropolitan-wise networks or even city-wide networks. To address these challenges, leveraging consolidated network analysis approaches with AI in the transport sector can provide valuable insights into solving TSLP more efficiently.

In practice, sensor location planning often lacks systematic methodology and involves some degree of subjectivity. For instance, in Georgia, the process for planning continuous counting stations (CCSs) follows three primary stages: criteria-based candidate selection, field evaluation, and installation of new CCS sites. The process considers factors such as traffic patterns, critical nodes on high-volume roads, and areas of interest to the Georgia Department of Transportation (GDOT) to ensure effective planning and federal compliance [11]. Normally, the budget is fixed so that the number of sensors that can be installed is predetermined. The challenge then becomes identifying the optimal locations for these sensors across a vast statewide network. This paper focuses on optimizing sensor locations given the number of sensors and proposes a novel framework along with an efficient algorithm to address the statewide TSLP.

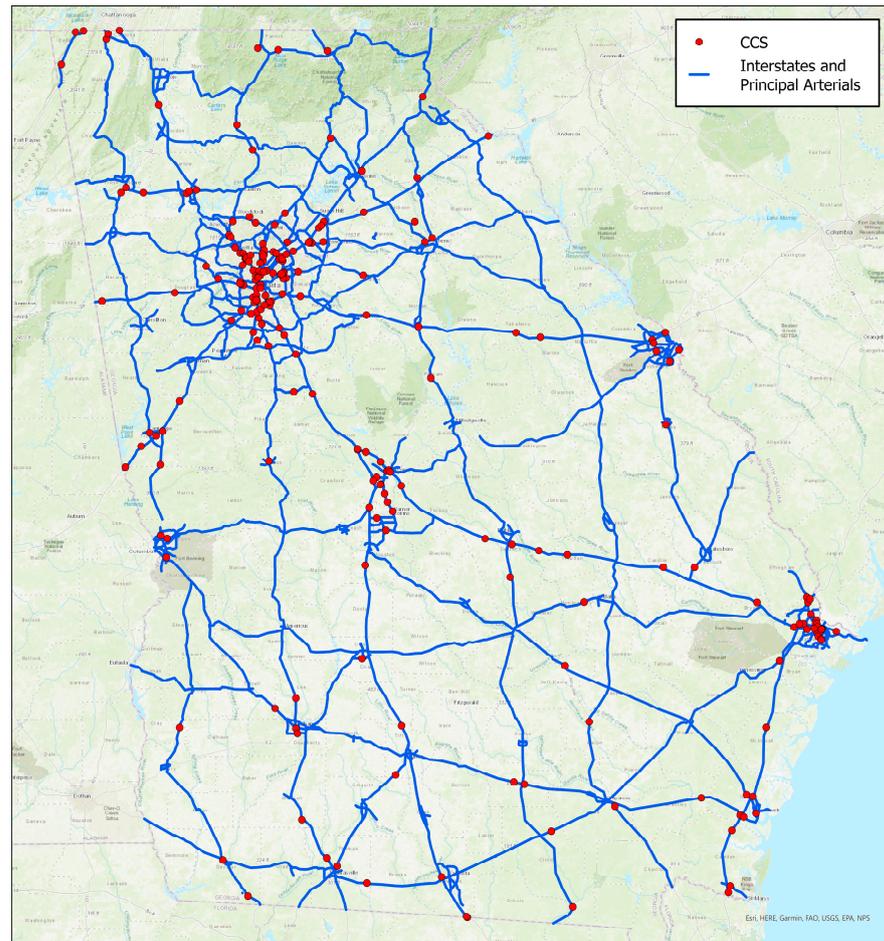
## 2. Problem Formulation and the Proposed Framework

### 2.1. Problem Formulation

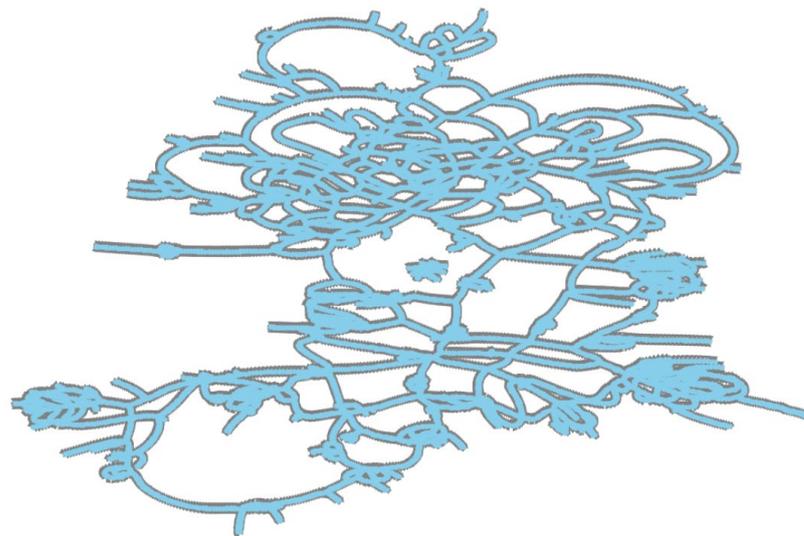
#### 2.1.1. Data Input

In this paper, we used 2015 Georgia's statewide highway network, which consists of 43,370 segments with 326 Continuous Count Stations (CCSs). For the purpose of this study, traffic sensors were referred to as CCS. To demonstrate our approach, we focused on the major highway network (functional class 1, 2 and 3) which includes interstates, freeways and expressways and principal arterials, resulting in 15,318 road segments with 246 CCS, depicted as red dots in Figure 1.

Then, we constructed a directed graph  $G = (V, L)$  to capture the directional traffic flows and network topology [12], where  $V$  is the set of nodes, with each node representing a road segment in the network, and the  $L$  is set of edges, indicating direct connection between road segments. The graph representation of Georgia's statewide highway network is depicted in Figure 2.

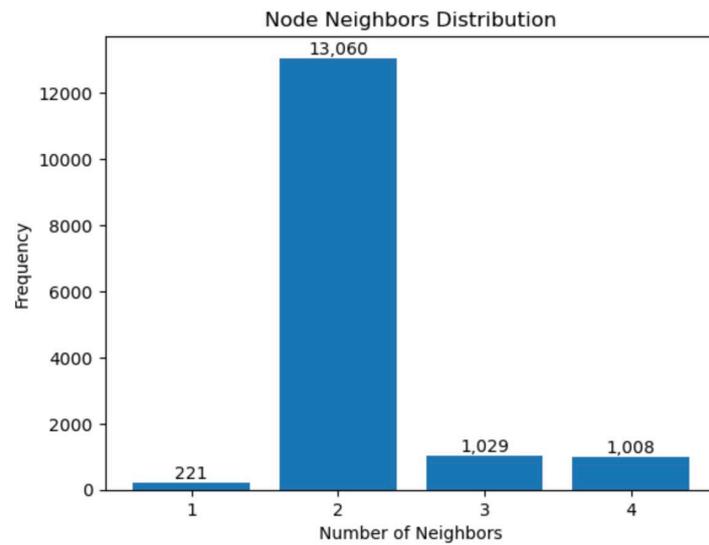


**Figure 1.** 2015 Georgia’s major road network with CCS.



**Figure 2.** Directed graph representation for the Georgia statewide highway network.

The distribution of direct neighbors is shown in Figure 3, revealing that the majority of the nodes have two neighbors, followed by the segments with three and four neighbors regarded as intersections, while only a few segments with one neighbor indicated the ends of road segments.



**Figure 3.** Distribution of the number of neighbors for the Georgia state highway network.

### 2.1.2. TSLP Objective Function

The TSLP in this study can be stated as conditional upon the existing sensor locations, choosing the locations of new sensors to optimize the network coverage by maximizing network-level information gain. Optimality is sought in an embedding space that considers network topology [13]. Maximizing information gain is equivalent to minimizing the Kullback–Leibler (KL) divergence between the data distribution,  $P(x)$ , and the model distribution,  $Q(x)$ .  $P(x)$  captures how sensors are distributed, while  $Q(x)$  is approximated by kernel density estimation (KDE) of segments for the entire network [14]. The KL divergence also referred to as information divergence or relative entropy serves as a metric for quantifying the discrepancy between the two distributions. The general form of KL is given by Equation (1) below:

$$D_{KL}(P \parallel Q) = \sum_{x \in \chi} P(x) \log \left( \frac{P(x)}{Q(x)} \right) \tag{1}$$

The model distribution  $Q(x)$  is derived from the outputs, i.e., from a travel demand model. Meanwhile, the data distribution  $P(x)$  corresponds to the choice set of locations, encompassing both the existing sensor sites ( $x_e$ ) and the planned sensor sites ( $x_p$ ). The goal is to select the sites of planned sensors ( $x_p$ ) to minimize the KL divergence between  $P(x)$  and  $Q(x)$  within the embedding space, as previously discussed. This objective function is formally expressed in Equation (2).

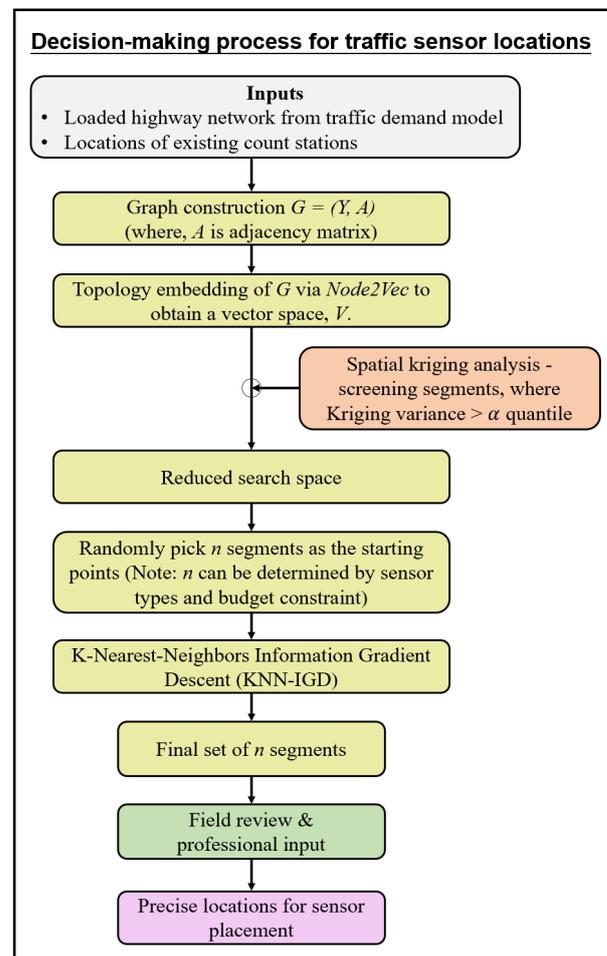
$$\min_{x_p} \sum_{\{x_e, x_p\}} P(x) \log \left( \frac{P(x)}{Q(x)} \right), \quad x_p \in \chi_{\text{kriging}} \tag{2}$$

where  $\chi_{\text{kriging}}$  denotes the reduced search space by Kriging variance, which will be discussed in Section 3.2.

### 2.2. Proposed Solution Framework

Our framework to handle the TSLP in a statewide network is illustrated in Figure 4. To accelerate the search process, we used kriging analysis to reduce the search space by identifying areas with high kriging variance as target zones. The TSLP then solved the same optimization problem within this reduced target network (Equation (2)). Our search method, referred to as K-Nearest-Neighbor Information Gradient Descent (KNN-IGD), was inspired by the stochastic gradient descent method used in machine learning, where gradients are computed with respect to model parameters. In our case, the gradient was computed with respect to spatial locations in the embedding space of the network topology.

Given the large number of factors influencing sensor placement, our algorithm generated a solution set of candidate segments, which can then be further reviewed by domain experts to determine the precise sensor locations.



**Figure 4.** Overview of the proposed framework.

### 3. Methods

In this section, we will discuss several key techniques employed, including topology embedding, spatial kriging analysis, and our KNN-IGD algorithm.

#### 3.1. Topology Embedding

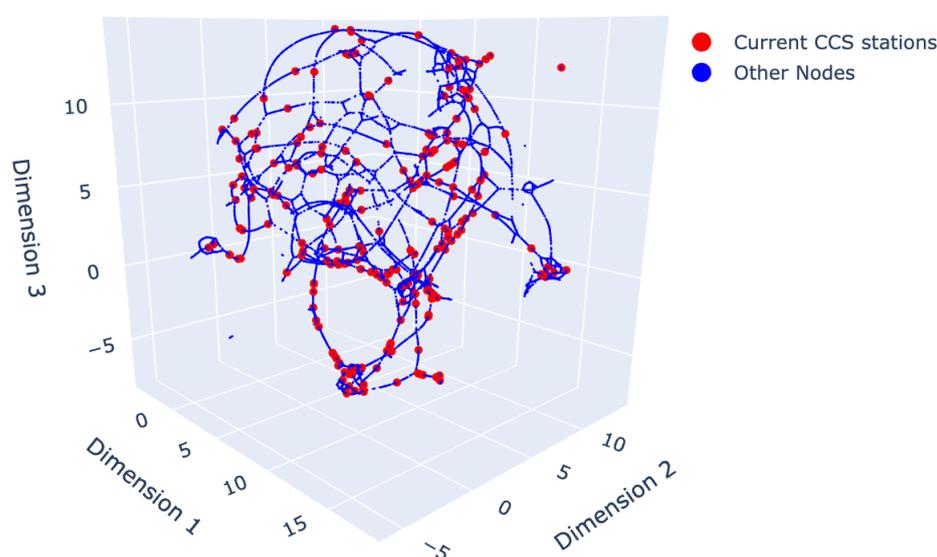
To accurately capture meaningful representations of the directed graph while preserving its topological structure, we employed the Node2Vec method introduced by Grover and Leskovec [15]. Node2Vec is a scalable, learning-based approach designed to encode network topology into a vector embedding space. It effectively learns vector representations that capture the intricate relationships between nodes, making these embeddings highly useful for various graph analysis tasks. The specific parameters for Node2Vec in our implementation are detailed in Table 1.

Specifically, we set  $p = q = 1$  as our goal to encode the network topology while preserving its structural integrity without introducing bias toward either the local or global structure of the network. To visualize the embeddings generated by Node2Vec, we used Uniform Manifold Approximation and Projection (UMAP). UMAP has proven highly effective in providing a competitive low-dimensional manifold representation while preserving more of the global structure in the embeddings [16]. As shown in Figure 5, UMAP strikes a balance between local and global connectivity. The current CCS locations

are represented by red dots, reflecting a representative sampling of network segments within the embedding space.

**Table 1.** Parameters of Node2Vec algorithm.

Parameter	Value	Description
$l_{wl}$	30	Walk length, i.e., the number of nodes in each walk
$N_{nw}$	200	Number of walks per node
$p$	1	The likelihood of backtracking the walk and immediately revisiting a node in the random walk.
$q$	1	The In-Out parameter $q$ allows the traversal calculation to differentiate between inward and outward nodes.
$d_{dim}$	8	The output node2vec embeddings dimension



**Figure 5.** Visualization of Georgia state highway network topology embeddings.

### 3.2. Spatial Kriging Analysis

Spatial kriging is a geostatistical technique used to interpolate the value of a random field (such as elevation, rainfall, or chemical concentrations) at specific locations based on observations from nearby points [17]. This method accounts for the statistical relationship between measured points and provides the Best Linear Unbiased Estimator (BLUE) [18]. Given the scale of the network, we first narrowed the search space. In this study, we applied ordinary kriging [19], using latitude and longitude as coordinates for existing sensors (i.e., CCSs) across the network, with Annual Average Daily Traffic (AADT) as the measurement variable. The kriging variance of AADT is shown in Figure 6. The results indicate lower variance in the Atlanta metropolitan area, attributed to the higher density of existing sensors, whereas rural regions of Georgia display higher variance due to sparser sensor coverage.

In this study, the target area was determined by applying a percentile threshold ( $\alpha$ ) to the kriging variance, focusing on regions with higher variance [20]. It is important to select an appropriate threshold value to balance the confidence of finding optimal solutions with the computational load. For demonstration purposes, we set  $\alpha = 50\%$ , i.e., reducing the search space by half. The resulting target area, where the kriging variance exceeds the 50th percentile, is visually depicted in Figure 7. Figure 8 further illustrates the corresponding highway network within this high-variance region.

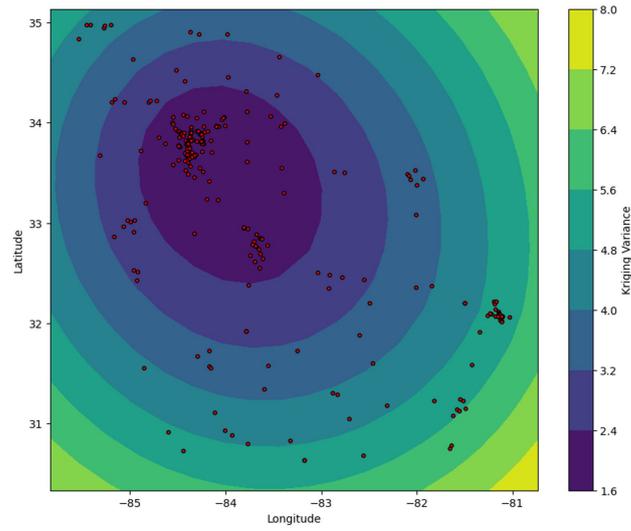


Figure 6. 2-D visualization of kriging variance.

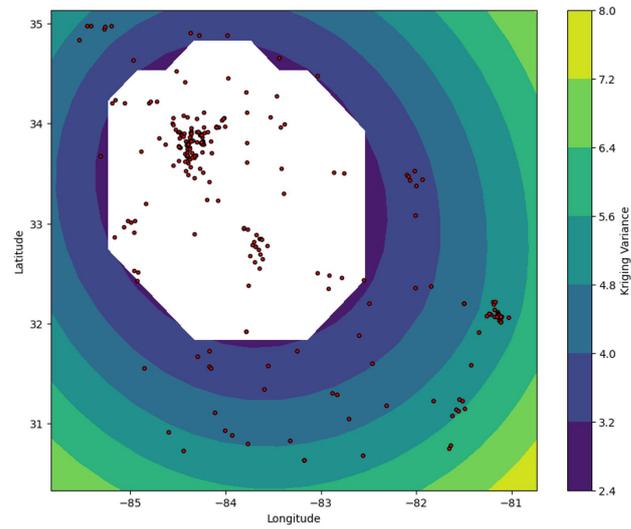


Figure 7. Target areas with kriging variance > 50%.

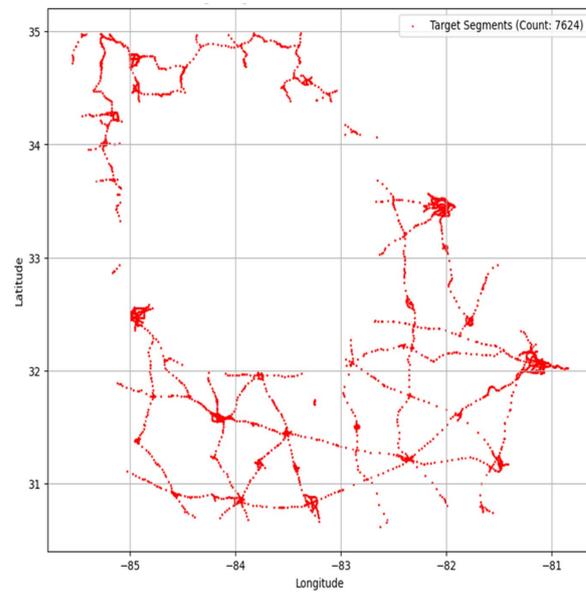


Figure 8. Reduced search space.

### 3.3. K-Nearest-Neighbor Information Gradient Descent (KNN-IGD)

Depending on the screening threshold selected, the target area often remains large, making traditional optimization methods impractical. To address this, we proposed a practical, scalable approach that leverages graph representation and information theory. Starting with random candidate locations, our method navigates the embedding space by following information gradients relative to  $K$ 's nearest neighbors while occasionally exploring beyond the neighbor set when there is no information gained from the neighbors. The implementation of KNN-IGD is detailed in Algorithm 1, with key aspects discussed subsequently.

---

#### Algorithm 1. KNN-IGD

---

Input:

Number of sensors:  $n$ ; the network screening threshold:  $\alpha$ ; the number of nearest neighbors:  $k$ ;  
 exploration rate:  $\theta$ ; search space:  $S$ .

1. Initialization:

Generate a set of randomly sampled  $n$  nodes ( $S_n$ ) in the reduced search space per  $\alpha$ .

$S_{near} = \emptyset$  //  $S_{near}$  tracks explored and current neighbor locations. //

$S_{far} = S$  //  $S_{far}$  tracks unexplored potential locations. //

$S_n^* = S_n$  //  $S_n^*$  retains the currently selected locations of  $n$  sensors. //

2. While ( $S_{far} \neq \emptyset$ ):

3. For each node  $i$  in  $S_n$ :

4.  $IG_{ji} = (KL_{j \in S_{knn}} - KL_i) \times AADT_j$  //  $S_{knn}$  is the neighbor set of node  $i$ . //

5.  $j^* = \underset{j \in S_{knn}}{\operatorname{argmin}}(IG_{ji})$

6.  $S_{near} \cup S_{knn}$

7. Update  $M_{knn}$  by masking visited nodes

8. If  $IG_{j^*i} < 0$ :

9.  $S_n^*[i] = j^*$  // update  $S_n^*$  //

10. Else: // explore the locations in the far set. //

11.  $S_{far} = S \setminus S_{near}$  //  $S_{far}$  is the complementary set of  $S_{near}$ . //

12.  $\epsilon \sim \text{Uniform}(0, 1)$ :

13. If  $\epsilon < \theta$ : randomly sample a node  $i'$  from  $S_{far}$

14.  $IG_{j'i'} = (KL_{j \in S_{knn}} - KL_{i'}) \times AADT_j$

15.  $j^* = \underset{j \in S_{knn}}{\operatorname{argmin}}(IG_{j'i'})$

16.  $S_{near} \cup S_{knn}$

17. Update  $M_{knn}$  by masking visited nodes

18. If  $IG_{j^*i'} < 0$ :

19.  $S_n^*[i] = j^*$  // update  $S_n^*$  //

20. If  $S_n^* \neq S_n$ : // check if there is any update over the current iteration. //

21. Return  $S_n^*$

22. Else:  $S_n = S_n^*$  // update  $S_n$  for next iteration. //

---

#### 3.3.1. KNN Matrix

The K-Nearest Neighbors (KNNs) [21] is a widely used non-parametric algorithm known for its simplicity and effectiveness in various applications. At its core, KNN operates on the principle of similarity within a feature space, assuming that similar instances are likely to be found near one another. For a given data point, the algorithm identifies its  $k$  nearest neighbors based on a chosen distance metric, typically Euclidean distance, to make predictions or classifications.

Let  $S = \{x_1, x_2, \dots, x_N\}$  be the set of target segments in the embedding space, where each embedding  $x_i \in \mathbb{R}^d$  encodes the topological structure of a segment within the highway network into a  $d$ -dimensional vector space. For each segment (node),  $K$  is the number of

nearest neighbors. For this study, the embedding dimension was  $d = 8$ . The Euclidean distance between two embedding vectors  $x_i, x_j$  is computed using Equation (3).

$$d(x_i, x_j) = \sqrt{\sum_{m=1}^d (x_{im} - x_{jm})^2}, \quad (3)$$

where  $m$  indexes embedding dimension. For each vector  $x_i$  in the set  $S$ , the distances  $d(x_i, x_j)$  are computed for all  $x_{j \neq i} \in X$ . The  $K$  nearest neighbors are then found by selecting the  $K$  shortest distances. Consequentially, the KNN matrix  $M^{N \times K}$  is constructed, where  $N$  is the number of segments in the search set, and  $K$  is the number of nearest neighbors. In this study,  $N = 7624$  and  $K = 10$ . The parameter  $K$  controls the local search area of each node within the embedding space, with larger  $K$  values encompassing more neighbors to be evaluated during the searching process.

### 3.3.2. Information Gradient

Gradients play an important role in many optimization techniques, particularly in machine learning and other computational fields [22]. Gradient effectiveness refers to how well gradients guide an algorithm toward an optimal solution within a given problem space. By providing critical information about the direction of improvement, gradients enable efficient convergence toward a local or global minimum or maximum. In our case, the goal was to minimize information loss, as defined by KL in Equation (1), which we refer to as the “information gradient” in this paper. Starting from a given node, we retrieved its  $K$  nearest neighbors using the KNN matrix described previously and computed the volume-weighted information gradient for each neighbor node using Equation (4).

$$IG_{ji} = (KL_{j \in S_{kni}} - KL_i) \times AADT_j, \quad (4)$$

where  $IG_{ji}$  is the volume-weighted information gradient when moving from node  $i$  to its neighbor node  $j$ .  $S_{kni}$  is the neighbor set of node  $i$ .  $AADT_j$  is the traffic volume for neighbor  $j$ . Weighting by AADT aims to maximize the observation of traffic flow.

### 3.3.3. Exploitation and Exploration

Similar to the exploitation and exploration concept in reinforcement learning [23], our implementation occasionally encounters situations where the gradient becomes minimal, particularly when  $K$  is small. In these cases, KNN-IGD struggles to make progress in gaining information. This occurs when the differences in KL divergence between a node and its neighbors are too small, causing the algorithm to become “stuck” in a local optimum, repeatedly selecting the same neighbor without exploring other regions of the search space. To address this issue, we introduced an exploration mechanism that allows the search to extend beyond the neighbor set. In this context, exploitation refers to searching within the neighbor set, while exploration involves sampling beyond it. As shown in Algorithm 1, we used parameter  $\epsilon$  to control the probability of exploration when there is no information gained from the neighbor set.

### 3.4. Genetic Algorithm

To demonstrate the efficacy of our KNN-IGD algorithm, we compared it with the Genetic Algorithm (GA), which has been extensively applied to complex real-world optimization problems. GA is inspired by the principle of biological evolution [24], iteratively exploring a set of candidate solutions and leveraging population characteristics to guide the search toward optimal solutions [25]. It mimics the process of natural evolution, where a population of candidate solutions, referred to as individuals, evolves through biological operators such as selection, crossover, and mutation [26]. The process begins with a randomly generated population. In each generation, the fitness of each individual is evaluated using a fitness function. Individuals with higher fitness scores are more likely to be selected for reproduction through crossover and mutation, creating a new population [27]. This

iterative process continues until either a predetermined fitness threshold is reached or a maximum number of generations is completed [28].

The effectiveness of Genetic Algorithms in identifying global optima is well-documented in the literature. Its ability to maintain a diverse set of solutions makes it particularly well-suited for exploring large, complex search spaces, helping to avoid premature convergence to local optima, a common issue in optimization algorithms. A previous work [13] has demonstrated strong performance of GA in optimizing sensor locations. In this study, we used GA as a baseline to compare it with our proposed KNN-IGD algorithm.

#### 4. Results

Given the inherent randomness in both GA and KNN-IGD, we evaluated their performance based on sampling distribution rather than single fixed-point outcomes. Specifically, we run each algorithm multiple times, benchmarking them against a random sample from the target search set [29]. This approach allows us to compute statistics for a more reliable and meaningful comparison [30].

##### 4.1. Implementation of GA as a Baseline

For a fair comparison, the GA was implemented with optimized parameters [31], as shown in Table 2.

**Table 2.** GA optimized parameters.

Parameter	Value
Number of generations	200
Number of parents mating	30
Population size	100
Number of genes	7624
Gene space	[0, 1]
Parent selection	roulette wheel
Crossover	single point
Mutation	random
Mutation percent for genes	10

##### 4.2. Comparison of KNN-IGD and GA

In our experiments, we set the number of new sensors to  $n = 5$ . We run both KNN-IGD and GA 10 times, resulting in 50 segments selected by each method. In terms of computation time, the average convergence time for GA was 46.85 s, whereas KNN-IGD took only 11.79 s. To visually compare the performance of KNN-IGD and GA against random sampling, we plotted the sample distributions in Figure 9. The results show that KNN-IGD samples fall into the lower  $KL$  region compared with GA and random sampling (RS), demonstrating superior performance.

To statistically validate the effectiveness of KNN-IGD, we conducted one-tailed  $t$ -Test [32], with the results summarized in Table 3. Particularly, three hypotheses were constructed and tested. The small  $p$ -values strongly support the rejection of the null hypotheses. The  $KL$  values for the solution set generated by KNN-IGD are significantly smaller than those produced by GA, and the  $KL$  values for GA are significantly smaller than those for RS. Note that smaller  $KL$  values indicate better solutions with lower information loss.

**Table 3.** One-tailed  $t$ -test results.

Null Hypothesis ( $H_0$ )	t-Statistic	$p$ -Value
$KL_{KNN-IGD} \geq KL_{RS}$	−26.91	<0.001
$KL_{GA} \geq KL_{RS}$	−20.24	<0.001
$KL_{KNN-IGD} \geq KL_{GA}$	−10.34	<0.001

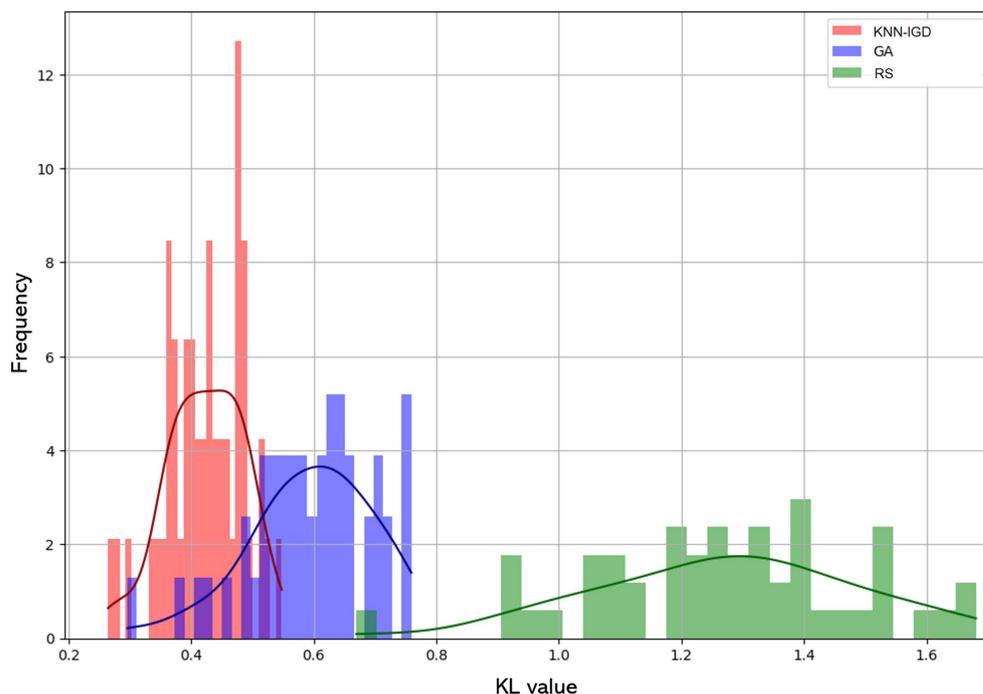


Figure 9. Histogram of KL values for KNN-IGD, GA, and RS.

For quantitative comparison, Table 4 presents the KL mean, standard deviation, and entropy calculated from samples generated using RS, GA, and KNN-IGD.

Table 4. Comparison of sample statistics.

Methods	Mean	SD	Entropy
RS	1.27	0.21	6.62
GA	0.60	0.10	6.46
KNN-IGD	0.42	0.06	6.25

As shown in Table 4, the solution set generated by KNN-IGD has the lowest KL mean, followed by GA and RS, indicating the highest solution quality with minimal information loss among the three methods. Additionally, KNN-IGD exhibits the lowest standard deviation and entropy, demonstrating the highest confidence in its solution set compared with GA and RS.

### 5. Conclusions

In this paper, we present a flexible framework for addressing the traffic sensor location problem by leveraging graph representation and information theory. The framework involves representing a target highway network as a directed graph, followed by embedding the graph into a vector space. Our search method, KNN-IGD, navigates this embedding space by following information gradients relative to K nearest neighbors and occasionally exploring beyond the neighbor set when no further information gain is detected. To expedite the search process, we leverage spatial correlations among existing sensors and apply spatial kriging analysis to identify target zones with high kriging variance. While this study focuses on continuous count stations, typically equipped with inductive loops embedded in the pavement, some researchers have suggested the use of Automated Vehicle Monitoring (AVM) and FCD [33] for Origin-Destination flow estimation. Incorporating FCD and other probe sensor data alongside spatial kriging analysis of sensor data can improve traffic volume estimation. By leveraging both kriging and KNN-IGD methods, this integration has the potential to enhance the accuracy of traffic flow predictions, particularly in areas with limited sensor coverage. However,

due to the sampling nature of these data sources, they do not provide reliable traffic volumes directly. Nevertheless, such data could complement spatial kriging analysis by identifying areas with higher uncertainty in traffic flow. By integrating diverse data sources, fixed-location sensors can serve as ground truth for correlating with mobile sensor data, offering an alternative way to estimate traffic volumes in areas lacking count stations. This approach could significantly reduce the costs associated with operating and maintaining statewide fixed-location sensor networks.

For benchmarking, we compared KNN-IGD to the widely used GA and random sampling. KNN-IGD demonstrated significantly faster convergence than GA. Based on the one-tailed t Test, the solution sets of both KNN-IGD and GA showed substantial information gain compared with RS. Furthermore, KNN-IGD outperformed GA, providing significantly higher information gain. These results highlight the effectiveness and robustness of KNN-IGD in identifying optimal sensor locations within a large-scale, statewide network. While the framework is applied to the traffic sensor location problem in this study, its versatility allows it to be easily adapted for spatial optimization challenges in other domains.

Despite the promising results, we acknowledge certain limitations in this study and suggest future research directions. While GA was chosen as a competitive baseline, other optimization algorithms could also be evaluated as well. Moreover, the information gradient was computed using KL divergence based on a Node2Vec topological embedding. Exploring alternative embedding methods could provide further improvements. Another valuable direction is the integration of multisource data, such as probe data, to further enhance information gain, which warrants further investigation.

**Author Contributions:** Conceptualization, J.J.Y.; methodology, J.J.Y. and Y.Y.; software, Y.Y.; validation, Y.Y. and J.J.Y.; formal analysis, Y.Y.; investigation, Y.Y. and J.J.Y.; resources, J.J.Y.; data curation, Y.Y. and H.Z.; writing—original draft preparation, Y.Y. and H.Z.; writing—review and editing, J.J.Y.; visualization, Y.Y.; supervision, J.J.Y.; project administration, J.J.Y.; funding acquisition, J.J.Y. All authors have read and agreed to the published version of the manuscript.

**Funding:** This work presented in this paper is the continued effort of a research project (RP 20-28) funded by the Georgia Department of Transportation, United States.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** Some or all of the data that support the findings of this study are available from the corresponding author upon reasonable request, subject to the permission of the Georgia Department of Transportation in the United States.

**Acknowledgments:** The contents of this paper reflect the views of the authors, who are solely responsible for the facts and accuracy of the data, opinions, and conclusions presented herein. The contents may not reflect the views of the funding agency or other individuals.

**Conflicts of Interest:** The authors declare no conflicts of interest.

## References

1. Cirianni, F.M.M.; Comi, A.; Quattrone, A. Mobility Control Centre and Artificial Intelligence for Sustainable Urban Districts. *Information* **2023**, *14*, 581. [[CrossRef](#)]
2. Alonso, B.; Musolino, G.; Rindone, C.; Vitetta, A. Estimation of a Fundamental Diagram with Heterogeneous Data Sources: Experimentation in the City of Santander. *ISPRS Int. J. Geo-Inf.* **2023**, *12*, 418. [[CrossRef](#)]
3. Birgillito, G.; Rindone, C.; Vitetta, A. Passenger Mobility in a Discontinuous Space: Modelling Access/Egress to Maritime Barrier in a Case Study. *J. Adv. Transp.* **2018**, *2018*, 1–13. [[CrossRef](#)]
4. Owais, M. Traffic Sensor Location Problem: Three decades of research. *Expert Syst. Appl.* **2022**, *208*, 118134. [[CrossRef](#)]
5. Viti, F.; Verbeke, W.; Tampère, C.M.J. Sensor locations for reliable travel time prediction and dynamic management of traffic networks. *Transp. Res. Rec. J. Transp. Res. Board* **2008**, *2049*, 103–110. [[CrossRef](#)]
6. Mahmoud, O.; El Deeb, M.; Abbas, Y.A. Distributing portable excess speed detectors in AL Riyadh city. *Int. J. Civ. Eng.* **2020**, *18*, 1301–1314.
7. Gentili, M.; Mirchandani, P. Locating sensors on traffic networks: Models, challenges and research opportunities. *Transp. Res. Part C Emerg. Technol.* **2012**, *24*, 227–255. [[CrossRef](#)]

8. Salari, M.; Kattan, L.; Lam, W.H.; Lo, H.; Esfeh, M.A. Optimization of traffic sensor location for complete link flow observability in traffic network considering sensor failure. *Transp. Res. Part B Methodol.* **2019**, *121*, 216–251. [[CrossRef](#)]
9. Owais, M.; Moussa, G.S.; Hussain, K.F. Sensor location model for O/D estimation: Multi-criteria meta-heuristics approach. *Oper. Res. Perspect.* **2019**, *6*, 100100. [[CrossRef](#)]
10. Li, R.; Mehr, N.; Horowitz, R. Submodularity of optimal sensor placement for Traffic Networks. *Transp. Res. Part B Methodol.* **2023**, *171*, 29–43. [[CrossRef](#)]
11. Georgia's Traffic Monitoring Program. Available online: [https://www.dot.ga.gov/DriveSmart/Data/Documents/Guides/2018\\_Georgia\\_Traffic\\_Monitoring\\_Program.pdf](https://www.dot.ga.gov/DriveSmart/Data/Documents/Guides/2018_Georgia_Traffic_Monitoring_Program.pdf) (accessed on 17 August 2024).
12. Gharaee, Z.; Kowshik, S.; Stromann, O.; Felsberg, M. Graph representation learning for road type classification. *Pattern Recognit.* **2021**, *120*, 108174. [[CrossRef](#)]
13. Yang, Y.; Yang, J.J. Strategic Sensor Placement in Expansive Highway Networks: A Novel Framework for Maximizing Information Gain. *Systems* **2023**, *11*, 577. [[CrossRef](#)]
14. van Erven, T.; Harremoës, P. Rényi Divergence and Kullback-Leibler Divergence. *IEEE Trans. Inf. Theory* **2014**, *60*, 3797–3820. [[CrossRef](#)]
15. Grover, A.; Leskovec, J. Node2vec: Scalable Feature Learning for Networks. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, 13–17 August 2016; pp. 855–864. [[CrossRef](#)]
16. McInnes, L.; Healy, J.; Melville, J. UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction. *arXiv* **2018**, arXiv:1802.03426.
17. Oliver, M.A.; Webster, R. Kriging: A method of interpolation for geographical information systems. *Int. J. Geogr. Inf. Sci.* **1990**, *4*, 313–332. [[CrossRef](#)]
18. Asa, E.; Saafi, M.; Membah, J.; Billa, A. Comparison of linear and nonlinear kriging methods for characterization and interpolation of soil data. *J. Comput. Civ. Eng.* **2012**, *26*, 11–18. [[CrossRef](#)]
19. Zhu, Q.; Lin, H. Comparing ordinary kriging and regression kriging for soil properties in contrasting landscapes. *Pedosphere* **2010**, *20*, 594–606. [[CrossRef](#)]
20. Yamamoto, J.K. An alternative measure of the reliability of ordinary kriging estimates. *J. Int. Assoc. Math. Geol.* **2000**, *32*, 489–509. [[CrossRef](#)]
21. Thomas, C.; Hart, P. Nearest neighbor pattern classification. *IEEE Trans. Inf. Theory* **1967**, *13*, 21–27.
22. Sebastian, R. An overview of gradient descent optimization algorithms. *arXiv* **2016**, arXiv:1609.04747.
23. Sutton, R.S.; Barto, A.G. *Reinforcement Learning: An Introduction*; A Bradford Book; MIT Press: Cambridge, MA, USA, 2018.
24. Talbi, E.G. *Metaheuristics: From Design to Implementation*; Wiley: Hoboken, NJ, USA, 2009.
25. Eiben, A.E.; Smith, J.E. *Introduction to Evolutionary Computing*; Springer: Berlin/Heidelberg, Germany, 2015.
26. Holland, J.H. *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence*; MIT Press: Cambridge, MA, USA, 1992.
27. Goldberg, D.E. *Genetic Algorithms in Search, Optimization, and Machine Learning*; Addison-Wesley: Reading, MA, USA, 1989.
28. Michalewicz, Z. *Genetic Algorithms + Data Structures = Evolution Programs*; Springer: Berlin/Heidelberg, Germany, 1996.
29. Nievergelt, J. Exhaustive search, combinatorial optimization and enumeration: Exploring the potential of raw computing power. In *International Conference on Current Trends in Theory and Practice of Computer Science*; Springer: Berlin/Heidelberg, Germany, 2000; pp. 18–35.
30. Maucher, M.; Schöning, U.; Kestler, H.A. Search heuristics and the influence of non-perfect randomness: Examining Genetic Algorithms and Simulated Annealing. *Comput. Stat.* **2011**, *26*, 303–319. [[CrossRef](#)]
31. Gad, A.F. PyGAD: An intuitive genetic algorithm Python library. *Multimed. Tools Appl.* **2024**, *83*, 58029–58042. [[CrossRef](#)]
32. Rouder, J.N.; Speckman, P.L.; Sun, D.; Morey, R.D.; Iverson, G. Bayesian t tests for accepting and rejecting the null hypothesis. *Psychon. Bull. Rev.* **2009**, *16*, 225–237. [[CrossRef](#)] [[PubMed](#)]
33. Comi, A.; Rossolov, A.; Polimeni, A.; Nuzzolo, A. Private car OD flow estimation based on automated vehicle monitoring data: Theoretical issues and empirical evidence. *Information* **2021**, *12*, 493. [[CrossRef](#)]

**Disclaimer/Publisher's Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.