Optimal Deployment of Movable Sensors for Air Quality Monitoring

Saurabh Kumar Maurya[1]; Hemant Gehlot[2]; and Sachchida Nand Tripathi[3]

Abstract: Monitoring air quality is crucial to obtaining health-related information but is challenging due to spatially and temporally varying pollution levels. This paper addresses the problem of optimally relocating air quality sensors over time to effectively capture these variations, prioritizing areas with higher pollution concentrations and adhering to constraints on sensor deployment and relocation. Unlike traditional static sensor deployment methods, our approach dynamically relocates sensors to adapt to changing pollution patterns, providing a more accurate and efficient monitoring solution. To solve the proposed nonlinear integer programming problem, we developed a genetic algorithm that efficiently explores the solution space and provides near-optimal solutions within reasonable computation time. We demonstrate that the proposed method effectively adapts to dynamic pollution patterns through two case studies—a simulated road network and a real-world urban scenario. Numerical tests show that the proposed algorithm achieves solutions within 0.01% of the optimal value and reduces computation time by two orders of magnitude compared to enumeration even when only three sensors are deployed. This research provides a framework for urban planners and environmental agencies to deploy air quality sensors more effectively, enabling better decision making in pollution monitoring and mitigation. **DOI: 10.1061/JITSE4.ISENG-2669.**

Introduction

Air pollution is one of the most challenging health threats of our time. Approximately four million people are estimated to have died in 2019 from problems associated with ambient air pollution (WHO 2022). Air quality monitoring helps identify areas with poor air quality and the responsible pollutants. This information is used to implement air pollution control measures to improve air quality and, thus, leads to better health outcomes. Various low-cost sensors have recently gained popularity in air quality monitoring due to their compact design, low weight, low power and less maintenance requirements than reference monitors (Gao et al. 2015; Castell et al. 2017; Spinelle et al. 2017). Although low-cost sensors exhibit high variability in measurements due to environmental variations, require frequent calibration, and lack the precision of more expensive reference monitors (Nalakurthi et al. 2024), they work well in practice. They are becoming very popular due to their low cost compared to the cost of reference monitors (because it is possible to monitor more locations with such sensors). For an idea of the cost comparison between sensors and monitors, the cost of an air quality sensor can be USD 3,000, whereas the cost of a reference monitor can be as high as USD 122,000 (Times of India 2023; CleanAir 2023). Thus, deploying sensors at various locations is much more economical. Alternative methods exist for monitoring ambient pollution, such as satellite imagery and dispersion modeling, apart from monitoring through sensors. Satellite imagery and dispersion models are applicable for coarse resolutions ($>1-10$ km) and thus cannot provide accurate concentration values for fine scales (10–500 m) that are useful in identifying pollution hotspots in urban areas influenced by localized sources such as traffic (Apte et al. 2017). Predicting concentrations of air pollutants in urban settings becomes more complex due to structures such as buildings that can generate different wind environments and cause changes in the dispersion and flow of pollutants (Yang et al. 2020). Furthermore, satellite imagery measures aerosol optical depth to obtain pollutants' concentrations and not the exact values at the ground level (Li et al. 2011). Thus, satellite imagery may not be able to accurately capture important pollutants such as black carbon, ultrafine particles, and others (Apte et al. 2017). Moreover, dispersion models are only as reliable as their underlying emission inventories. Therefore, such models may be unable to detect emissions due to unexpected sources and thus are not as accurate as is obtaining concentration values through sensors. In addition, dispersion modeling is usually costly, makes unrealistic assumptions about dispersion patterns (such as Gaussian dispersion), and requires extensive cross-validation using monitoring data (Jerrett et al. 2005). Therefore, this paper focuses on monitoring ambient pollution through sensors.

Most existing studies that focused on deploying low-cost sensors deployed sensors either arbitrarily or use rules of thumb. However, budget availability can restrict the number of sensors that can be deployed (Sun et al. 2019; Ajnoti et al. 2024). Therefore, optimally deploying sensors under such constraints becomes important. Some studies focused on optimally deploying sensors for air quality monitoring. For example, Sun et al. (2019) proposed an optimization formulation for sensor deployment to maximize the total satisfaction across different portions in a region, where the satisfaction of a portion is a decreasing function of the distance of the portion to the closest deployed sensor. Thus, the parameter that governs the satisfaction of a portion is the distance of the portion to the closest deployed sensor. This objective function ensures that all portions of a region have sensors as close as possible while giving equal importance to different portions. The definition of the satisfaction function comes from the intuition that humans rely more on air pollution information from the nearest deployed instrument when multiple monitoring instruments are available in an

[1]Dept. of Civil Engineering, IIT Kanpur, Kanpur, Uttar Pradesh 208016, India.

[2]Professor, Dept. of Civil Engineering, IIT Kanpur, Kanpur, Uttar Pradesh 208016, India (corresponding author). ORCID: https://orcid.org/0000-0002-1356-2390. Email: hemantg@iitk.ac.in

[3]Professor, Dept. of Civil Engineering and Dept. of Sustainable Energy Engineering, IIT Kanpur, Kanpur, Uttar Pradesh 208016, India.

Note. This manuscript was submitted on October 14, 2024; approved on June 23, 2025; published online on August 25, 2025. Discussion period open until January 25, 2026; separate discussions must be submitted for individual papers. This paper is part of the ***Journal of Infrastructure Systems***, © ASCE, ISSN 1076-0342.

area (Sun et al. 2018). Note that Sun et al. (2019) considered the constraint on the number of sensors that can be deployed but did not consider placement-related constraints such as the impracticability of deploying sensors at some locations (such as open areas, areas near water bodies, and others) and the requirement of deploying sensors at important locations (such as industrial areas, congested roads, and others). Moreover, Sun et al. (2019) considered deploying only one type of air quality monitoring instrument, although different types of instruments with varying cost, accuracy, and other features can exist. Ajnoti et al. (2024) addressed these gaps by proposing an optimization formulation that focused on deploying two different types of air quality monitoring instruments (sensors and reference monitors) to maximize satisfaction under constraints such as limited budget availability for deploying monitoring instruments, the requirement of placing at least one sensor in a given set of portions in a region, the requirement of not placing any reference monitor in a given set of portions in a region, and others. Unlike Sun et al. (2019), which only considered population data, Ajnoti et al. (2024) considered both ambient pollution levels and population data while deciding the importance of different portions of the region in the optimization formulation. Studies such as Sun et al. (2019) and Ajnoti et al. (2024) that focused on the optimal deployment of sensors do not consider the change in ambient pollution over time. However, there may be changes in ambient pollution levels within a day, across days, and other periods, driven by factors such as traffic patterns, industrial activities, and meteorological conditions (Apte et al. 2017).

Recently, some studies considered deploying vehicles fitted with sensors to capture spatio-temporal variations in pollution (Hagemann et al. 2014; Apte et al. 2017). Movable sensors can capture spatial and temporal variability in pollutant concentrations due to localized sources such as traffic in urban environments, which is difficult to achieve with stationary sensors (Van Poppel et al. 2013; Lim et al. 2019; Cummings et al. 2021; Whitehill et al. 2024). For example, Apte et al. (2017) used vehicles mounted with sensors to develop a large urban air quality data for an area in Oakland, CA, with greater spatial precision than possible using fixed monitoring instruments. Recently, various studies also found that the data obtained from movable sensors can improve the granularity of land-use regression models used for predicting spatiotemporal variability of $PM_{2.5}$ concentrations (Lim et al. 2019; Shi et al. 2016; Deville Cavellin et al. 2016). In summary, several studies pointed to the growing importance of using movable sensors compared to stationary sensors in urban settings where pollutants' concentrations can vary significantly over small distances and short periods. We also provide the following example in favor of movable sensors. Suppose there are N locations. In the first case, we decide to deploy stationary sensors, allocate the available budget to buying k sensors, and deploy them at all times at a fixed set of k locations. In the other case, we deploy movable sensors, thus dividing the available budget into buying $k'(< k)$ sensors and keeping the remaining budget for the relocation of sensors. Suppose there is a location x where we do not deploy any of the k sensors in the former case (i.e., the stationary sensors case). Then, directly measuring the concentration value at location x in this case is not possible, even when the concentration value becomes very high for some period at this location. In that case, only methods such as dispersion modeling, interpolation, satellite imagery, and others could be used to estimate the concentration values at location x. Still, those may not provide accurate results if the concentration value at location x is significantly different from that at nearby locations and is affected by localized sources. However, if we deploy movable sensors, the proposed optimization formulation (to be presented in the next section) will prioritize deploying sensors at locations such as x during periods when the concentration value at x is high.

Since covering all the target locations to capture ambient pollution levels at a given time might not be possible, and costs might be attached to the movement or shifting of sensors, optimally deciding the movement of sensors over time becomes important. However, the studies previously mentioned that focused on dynamically moving sensors did not optimize sensor movement.

In summary, most studies that deployed low-cost sensors did so either arbitrarily or used rules of thumb. Moreover, many studies that considered the deployment of low-cost sensors focused on the static deployment of sensors. Some studies considered dynamically deploying sensors to account for changes in ambient pollution over time but they did not optimally deploy sensors. Thus, no studies considered the optimal deployment of movable sensors to consider the change in ambient pollution over time.

Therefore, the contributions of this paper are as follows. We first develop an optimization formulation for the deployment of movable sensors to consider temporal variations in ambient pollution to maximize satisfaction across all portions of a region during a given time duration (sensors that can be relocated from one location to another location over time are referred to as movable sensors in this paper; note that the mode used to move a sensor can be any vehicle, which is not the focus of this paper). The proposed formulation considers constraints on the number of sensors present at any time for monitoring air pollution and the maximum number of relocations that can occur in a given duration. To the best of the authors' knowledge, this study is the first to focus on optimal relocations of air quality monitoring sensors over time while considering the aforementioned constraints. Note that the constraint on the maximum number of relocations introduces several other constraints for defining sensor relocations in the proposed optimization formulation. Therefore, the proposed formulation for the dynamic deployment of sensors in this paper is more challenging than the corresponding optimization formulation that focuses on the static deployment of sensors. Second, the proposed optimization problem is a nonlinear integer program; therefore, we develop a genetic algorithm (GA) to solve it efficiently. In the proposed GA, we design crossover and mutation operators to ensure that all the solutions generated from the GA satisfy at least some constraints of the proposed optimization problem. Thus, the proposed GA searches for optimal solutions in a reduced space, contributing to its computational efficiency. Finally, we test the proposed GA to determine sensor deployment locations over time for two case studies: (i) sensors are deployed along a road network to focus on vehicular pollution, and (ii) sensors are deployed for citywide air pollution monitoring.

The remaining sections of this paper are structured as follows. The next section presents the developed optimization formulation. The section after that explains the methodology used to solve the proposed optimization problem. Next, the results obtained by applying the developed GA to the case studies previously mentioned are presented. Finally, practical data collection methods, managerial insights, limitations, and conclusions are discussed.

Problem Formulation

In this section, we describe the formulation for the optimal deployment of air quality monitoring sensors. First, the notation used in this paper is discussed as follows.

We consider a set $V = \{1, \ldots, N\}$ consisting of all potential locations for deployment of sensors in a given area. The locations in set V should be defined such that they represent the important locations from the perspective of monitoring air pollution, e.g., major road intersections and important public places such as

schools, railway stations, and others. Let T be the maximum number of time-steps for which we focus on deploying sensors in the given area. Consider that time $t \in \{1, \ldots, T\}$ progresses in discrete time-steps, representing the resolution at which sensors can be moved (or relocated) from one location to another. The length of a time-step depends on how frequently it is required to relocate a sensor. For example, a time-step can be defined as an hour, a day, a week, and others depending on the desired frequency of sensor relocations. Moreover, the duration of a time-step should be chosen such that it is sufficiently large as the average time required to move a sensor between any two locations in set V (e.g., choosing a time-step to be equal to a minute may not be a good decision because that may not be enough time to relocate sensors). In our problem, we give importance to different locations in set V in terms of the respective ambient pollution concentrations generated at these locations such that the locations that generate higher concentration values are prioritized when deploying sensors. Estimates of the concentration values at locations in set V can be obtained in multiple ways, e.g., if set V has locations along a road network, we can obtain concentration values around these locations using traffic simulation models. We may also obtain the concentration values around the locations in set V through other methods such as satellite imagery (since we require ambient pollution concentration values at potential locations as inputs for the optimal deployment of sensors, we suggest methods such as traffic simulation, satellite imagery, and others to estimate such values before the actual sensor deployment). For all $i \in V$ and $t \in \{1, \ldots, T\}$, let e_t^i be the concentration value obtained through the aforementioned methods at location i for time-step t (these concentration values are assumed to be known beforehand rather than them being estimated in real-time). Moreover, for all $i \in V$ and $t \in \{1, \ldots, T\}$, let z_t^i be a binary variable equal to 1 if a sensor is deployed at location i in time-step t; otherwise, it equals 0. Suppose we have the budget to deploy $k(\leq N)$ sensors at any given time-step (this budget corresponds to the restriction on the total cost available to buy the sensors). At most, one sensor can be deployed at any location belonging to the set V at a given time-step. For all $i \in V$ and $t \in \{2, \ldots, T\}$, let p_t^i be equal to 0 if $z_{t-1}^i = z_t^i$ (i.e., either a sensor is deployed at location i at both the time-steps $t-1$ and t or no sensor is deployed at location i at both the time-steps $t-1$ and t); otherwise, it is equal to 1.

Suppose a sensor is deployed at location j in time-step $t-1$ but not in time-step t, and a sensor is deployed at location q in time-step t but not in time-step $t-1$ (suppose the deployment of sensors at all other locations stays the same at that in time-steps $t-1$ and t). In that case, a sensor may be considered to have relocated from location j in time-step $t-1$ to location q in time-step t. Note that $\sum_{i \in V} p_t^i = 2$ in this case because $p_t^j = p_t^q = 1$ and $p_t^i = 0$, $\forall\ i \in V \setminus \{j, q\}$. Therefore, we define the total number of relocations that take place from time-step $t-1$ to time-step t to equal $\sum_{i \in V} p_t^i/2$. Note that a cost can be associated with the relocation of a sensor; thus, there may be a constraint on the total number of relocations that can take place for a given budget. We assume that the cost associated with each relocation is the same; thus, we denote the total number of relocations that can take place across T time-steps as R (i.e., R is determined by the budget available for relocating sensors). We only focus on the number of relocations in consecutive time-steps and do not focus on the routes chosen while relocating sensors. Moreover, we assume that there is a mechanism for relocating sensors over time, although the mode of transport is not relevant. For example, connected vehicles equipped with sensors can serve this purpose because they can ensure that relocations happen in a coordinated way.

Under this setting, we now define the objective function of our problem. For all $i \in V$ and $t \in \{1, \ldots, T\}$, let d_t^i be the distance between location i and the closest deployed sensor from i in time-step t (thus, d_t^i is not given and is determined after solving the proposed optimization problem). We define the satisfaction function $g(d_t^i)$ as a decreasing function of d_t^i. Previous studies such as Sun et al. (2019) and Ajnoti et al. (2024) used the idea of a satisfaction function but for static deployment of air quality monitoring instruments. In this paper, the objective function is defined as a metric that is a weighted function of the satisfaction functions summed across different locations and time-steps. This objective function ensures that all the potential locations at various time-steps have sensors as close as possible while giving importance to different locations in proportion to the respective concentration values. In this paper, we choose the weight corresponding to location i in time-step t in the objective function to be equal to the corresponding concentration value of a pollutant (denoted by e_t^i) because we want to prioritize sensor deployment in locations where the ambient concentration values are high. That is, we define the objective function as the sum of $g(d_t^i)e_t^i$ over all the locations $i \in V$ and time-steps $t \in \{1, \ldots, T\}$. We assume that the satisfaction function $g(d)$ satisfies the following criteria: (i) it is a strictly decreasing function with d, i.e., for any $d1 < d2$, we have $g(d1) > g(d2)$ (such that the satisfaction at a location increases as the distance to the closest deployed sensor from that location decreases), (ii) for any $d > 0$, we have $g(d) > 0$ (this condition ensures that there is positive satisfaction at a location regardless of the distance between the given location and the closest deployed sensor), and (iii) for $d = 0$, we have $g(d) = 1$ (this is a boundary condition for the satisfaction function). We now present the optimization formulation for the dynamic deployment of movable sensors.

$$\text{Max} \sum_{t=1}^{T} \sum_{i \in V} e_t^i g(d_t^i) \tag{1}$$

$$\text{s.t.} \sum_{i \in V} z_t^i = k, \quad \forall\ t \in \{1, 2, \ldots, T\} \tag{2}$$

$$\sum_{t=2}^{T} \sum_{i \in V} p_t^i \leq 2R \tag{3}$$

$$p_t^i \geq z_t^i - z_{t-1}^i, \quad \forall\ i \in V, \quad t \in \{2, 3, \ldots, T\} \tag{4}$$

$$p_t^i \geq z_{t-1}^i - z_t^i, \quad \forall\ i \in V, \quad t \in \{2, 3, \ldots, T\} \tag{5}$$

$$p_t^i \leq z_t^i + z_{t-1}^i, \quad \forall\ i \in V, \quad t \in \{2, 3, \ldots, T\} \tag{6}$$

$$p_t^i \leq 2 - (z_t^i + z_{t-1}^i), \quad \forall\ i \in V, \quad t \in \{2, 3, \ldots, T\} \tag{7}$$

$$d_t^i \leq d_{ij} z_t^j + (1 - z_t^j)M, \quad \forall\ i \in V, \quad j \in V, \quad t \in \{1, \ldots, T\} \tag{8}$$

$$y_t^{ij} \leq z_t^j, \quad \forall\ i \in V, \quad j \in V, \quad t \in \{1, \ldots, T\} \tag{9}$$

$$\sum_{j \in V} y_t^{ij} = 1, \quad \forall\ i \in V, \quad t \in \{1, \ldots, T\} \tag{10}$$

$$d_t^i = \sum_{j \in V} d_{ij} y_t^{ij}, \quad \forall\ i \in V, \quad t \in \{1, \ldots, T\} \tag{11}$$

$$y_t^{ij} \in \{0,1\}, \quad \forall\ i \in V, \quad j \in V, \quad t \in \{1, 2, \ldots, T\} \tag{12}$$

$$p_t^i \in \{0,1\}, \quad \forall\ i \in V, \quad t \in \{2, 3, \ldots, T\} \tag{13}$$

$$z_t^i \in \{0,1\}, \quad \forall\ i \in V, \quad t \in \{1, 2, \ldots, T\} \tag{14}$$

The expression in Eq. (1) depicts the objective function that aims to maximize overall satisfaction across different locations and time-steps. The constraint in Eq. (2) ensures that at any time-step t, k sensors are deployed. The constraint in Eq. (3) ensures that the total number of relocations that take place is less than or equal to the maximum number of allowed relocations R. The constraints in Eqs. (4), (5), (6), and (7) ensure that for all $i \in V$ and $t \in \{2, \ldots, T\}$, p_t^i is equal to 0 if $z_{t-1}^i = z_t^i$; otherwise, it is equal to 1. For all $i, j \in V$, let d_{ij} be the distance between locations i and j. Moreover, let M be a very large positive number. Then, the constraint in Eq. (8) ensures that for all $i \in V$ and $t \in \{1, \ldots, T\}$, d_t^i is less than or equal to the distance d_{ij} between location i and every location j that has a sensor deployed in time-step t. For all i, $j \in V$ and $t \in \{1, \ldots, T\}$, let y_t^{ij} be equal to 1 if the closest location from i that has a sensor deployed in time-step t is location j. The constraint in Eq. (9) ensures that for all $i, j \in V$ and $t \in \{1, \ldots, T\}$, y_t^{ij} is equal to 0 if a sensor is not deployed at location j in time-step t. Eq. (10) ensures that for any location $i \in V$ and time-step $t \in \{1, \ldots, T\}$, there exists only one location j for which y_t^{ij} is equal to 1. Eq. (11) ensures that for any location $i \in V$ in time-step $t \in \{1, \ldots, T\}$, d_t^i is equal to the distance from location i to location j that has the closest deployed sensor from location i (because of Eq. (10) and from the fact that the variables y_t^{ij} are binary). Eqs. (12), (13), and (14) ensure that $\{y_t^{ij}\}$, $\{p_t^i\}$, and $\{z_t^i\}$ are binary variables.

Several functions can satisfy the criteria for function $g(d)$ previously mentioned. Recall that $g(d)$ should be a strictly decreasing function and should take positive values for all values of d and the value of 1 when $d = 0$. Thus, only nonlinear functions can be potential candidates for $g(d)$. For example, $\exp(-d)$ is one of the functions that satisfies the aforementioned criteria for the $g(d)$ function. Similarly, the function $1/(d+1)$ can also be used because it satisfies the required criteria. In this paper, we assume $g(d) = \exp(-d)$, as in previous studies such as Sun et al. (2019) and Ajnoti et al. (2024). Thus, the objective function $\sum_{t=1}^{T} \sum_{i \in V} e_t^i g(d_t^i)$ is a nonlinear function; therefore, the proposed optimization formulation is a nonlinear integer program because of the presence of integer variables, such as p_t^i and z_t^i. We now present the methodology for solving the proposed optimization formulation.

Solution Methodology

In this section, we present the GA developed to solve the problem described in the previous section. Since the proposed optimization formulation is a nonlinear integer program, classical methods of optimization may not be the best to apply because they can become stuck in local optima (Deb 2001). Moreover, exact algorithms for solving integer programs such as Branch and Bound can be computationally burdensome for large problem instances because they may end up exploring all possible solutions in the worst case (Morrison et al. 2016). Therefore, traditional solvers that use variants of such exact algorithms also become computationally burdensome when they solve large instances of integer programs (Morrison et al. 2016). That is because integer programming, in general, is NP-hard and, thus, the existing solvers do not scale well as the size of the problem increases (Karp 2010). However, meta-heuristics such as GA are better suited for solving such problems (Deb 2001).

GA is an optimization algorithm that draws its inspiration from evolutionary biology and mimics the process of how genes are transferred across generations. GA involves evolutionary processes such as selection, crossover, and mutation (we subsequently describe in detail these operators). With the help of these operators, the algorithm is unlikely to become stuck at local optima. However, GAs do not have proof of convergence to the optimal solution. Thus, the performance of GAs can only be analyzed by numerically testing with other benchmarked approaches. Therefore, in the next section, we compare the performance of the developed GA through numerical tests (since the objective of this paper is to demonstrate the feasibility of optimally deploying air quality sensors over time rather than developing the most efficient approach for solving this problem, we only compare the performance of the developed GA with one approach; thus, testing GAs performance with other commonly available tools remains a future direction). For more details regarding GA, refer to Deb (2001).

We use the term string in the proposed GA to represent a solution to our problem. A string is represented as a TN-sized array consisting of T segments corresponding to the T time-steps. Each segment of a string has N boxes corresponding to N potential locations for sensor deployment. The i^{th} box from left in the t^{th} segment of a string takes a numerical value if a sensor is deployed at location i in time-step t in the corresponding solution; otherwise, it takes a null value. The convention for the boxes that contain numerical values is to put 1 in the first such box from the left in a segment, then put the number 2 in the second such box from the left in that segment, and so on. The boxes that have sensors deployed (i.e., the boxes containing numerical values) are given different numbers in a segment to explain different operators in GA; however, all the sensors are assumed to be identical in every aspect. Similarly, the first null value in any segment is represented as N_1, the second null value in any segment is represented as N_2, and so on. The developed GA ensures that each segment of a string has exactly k boxes that have numerical values (thus, k sensors are deployed at each time-step in each GA solution). An example of a string for two time-steps (i.e., $T = 2$) and four locations (i.e., $|V| = N = 4$) with $k = 2$ is shown in Fig. 1. The bold vertical line is used to separate two consecutive time-steps. The first four boxes from the left correspond to the deployment of sensors at different locations for the first time-step, and the next four boxes correspond to the deployment of sensors at different locations for the second time-step. Note that exactly two boxes in both segments have numerical values, representing the locations where sensors are present at these time-steps.

The fitness function of a given string is a metric representing the goodness of the corresponding solution in the context of the underlying objective and constraint functions. For a string s, the fitness function is defined as follows:

$$f(s) = \begin{cases} \sum_{t=1}^{T} \sum_{i \in V} e_t^i g(d_t^i) & \text{if } s \text{ is a feasible string} \\ S_{\min} - P & \text{otherwise} \end{cases} \tag{15}$$

Here, $S_{\min}$ = minimum value of fitness among all feasible strings within a population (a string is said to be feasible if the solution corresponding to that string satisfies all the constraints of the proposed optimization problem). Moreover, $P = (\sum_{t=2}^{T} \sum_{i \in V} p_{i,t}) - 2R$. After performing crossover and mutation on strings, the constraint on the total number of relocations, i.e., the constraint in Eq. (3), may be violated; thus, the penalty

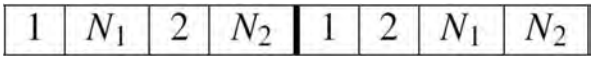

Fig. 1. Illustration of a string in GA.

P has been added to the fitness function determination. Thus, the fitness value of a feasible string is equal to the objective function value; however, the fitness value of an infeasible string is equal to the minimum fitness value of all feasible strings within the population minus the penalty for violating the constraint of Eq. (3). Thus, the fitness value of any infeasible string in a population never exceeds the fitness value of a feasible string in the same population. Note that it is ensured that all the strings in the initial population satisfy all the constraints of the proposed optimization formulation. However, GA uses operators such as crossover and mutation, and infeasible strings might be generated after using these operators. All the constraints, except those given in Eq. (3), are satisfied after applying the operators of GA (this is subsequently explained in detail).

We now describe the different operators of the proposed GA. The first operator used in GA is the selection operator that creates a new population of strings from a given population based on the fitness values of the strings in the given population. In this algorithm, we use k-tournament selection (Deb 2001), a method used to select strings to create the population for the next generation. This selection process is inspired by the concept of a competitive tournament, where k strings are randomly chosen, and the string with the highest fitness value is selected. This process is repeated until we obtain the required number of strings in the new population. We now discuss the crossover operator that we developed in this algorithm. We define the crossover probability P_c depending on which it is decided whether a pair of strings will undergo crossover or not. We refer to the pair of strings that undergo crossover as parent strings. Let $P1$ and $P2$ be a set of parent strings. An example of parent strings with two time-steps, four potential locations and $k = 2$ is shown in Fig. 2(a). For every pair of parent strings, one temporary string $T1$ is generated, which is of the same size as the parent strings. In a temporary string, each box is assigned the value of 0 or 1 with equal probability. An example of a temporary string $T1$ for two time-steps, four locations, and $k = 2$ is shown in Fig. 2(b). The offspring strings $O1$ and $O2$ obtained after the crossover are shown in Fig. 2(c). For creating the first offspring string $O1$, the values of the boxes in $P1$ that correspond to the boxes having 1 in the temporary string $T1$ are directly copied in $O1$. In the shown example, the temporary string has 1 in the third and fourth boxes in the first segment and 1 in the second and third boxes in the second segment. Thus, the values of $P1$ from all these boxes are copied in the same boxes in $O1$. For example, the values 2 and N_2 present in the third and fourth boxes in the first segment in $P1$ are copied in the third and fourth boxes of the first segment in $O1$. The values N_1 and 2 that are present in the second and third boxes in the second segment in $P1$ are copied in the second and third boxes of the second segment in $O1$. To fill the remaining boxes in each segment of $O1$, we select those values in $P1$ that have not yet been copied in $O1$ and put them in the same order as they appear in $P2$. For example, the values 1 and N_1 are copied in the second and first boxes of the first segment in $O1$ because N_1 is to the left of 1 in the first segment of $P2$. To create a second offspring $O2$, the values of the boxes in $P2$ that correspond to the boxes that have 1 in the temporary string $T1$ are directly copied in $O2$. In this example, the temporary string has 1 in the third and fourth boxes in the first segment and 1 in the second and third boxes in the second segment. Thus, the values of $P2$ from all these boxes are copied in the same boxes of $O2$. To fill the remaining boxes in each segment of $O2$, we select those values in $P2$ that have not yet been copied in $O2$ and put them in the same order as they appear in $P1$. Thus, in this way, we obtain both the offsprings $O1$ and $O2$.

P1	1	N_1	2	N_2	1	N_1	2	N_2
P2	N_1	1	2	N_2	N_1	N_2	1	2

(a)

T1	0	0	1	1	0	1	1	0

(b)

O1	N_1	1	2	N_2	N_2	N_1	2	1
O2	1	N_1	2	N_2	N_1	N_2	1	2

(c)

Fig. 2. Example of crossover operator: (a) parent strings; (b) temporary string; and (c) offspring strings.

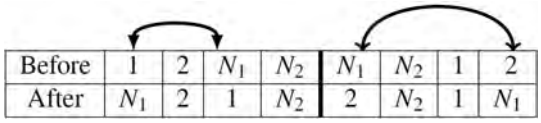

Fig. 3. Example of mutation operator.

We now discuss the mutation operator proposed in this algorithm. We define mutation probability P_m based on which it is decided whether a string undergoes mutation or not. If a string undergoes mutation, we swap values of two randomly selected boxes in each segment. An example of mutation is shown in Fig. 3 (the first row shows a string before mutation, and the second row shows that string after mutation). In this example, the values of the first and third boxes from the first segment are swapped. In the second segment, the values of the first and the fourth boxes are swapped.

We perform elitism in each generation of the proposed GA to retain a few of the best strings. In this process, we directly transfer some of the best strings from the current population to the new population without subjecting them to various operators used in GA. Let β_i be the mean fitness value of the strings in the population obtained in the i^{th} generation of GA. Let G be the maximum number of generations allowed in GA. In each generation, we undergo the process of selection, crossover, and mutation to obtain a new population until one of the following stopping criteria is first met: the relative change in the mean fitness value across consecutive generations, i.e., $|(\beta_i - \beta_{i-1})/\beta_{i-1}|$, falls below a predetermined threshold α or when generation number i becomes equal to G.

Results

In this section, we discuss the results obtained from solving the proposed optimization problem. This section is divided into two subsections. In the first subsection, we present the results of the deployment of movable sensors along a road network to focus on vehicular pollution. The ambient pollution levels along the road network (i.e., the values e_t^i in the objective function) in this case are obtained using a traffic simulation model. In the second subsection, we present the results when movable sensors are deployed for city-wide monitoring. In that case, the e_t^i values used in the optimal deployment of sensors are obtained from real-world data for a city in India.

In the results that we present, we only use fine particulate matter ($PM_{2.5}$) to denote concentration values because $PM_{2.5}$ concentrations strongly impact individuals' health (Lin et al. 2017).

However, the optimization problem that we propose is not restricted to only taking $PM_{2.5}$ concentrations as the inputs; rather, other indicators of pollution such as CO, NO_x, O_3, NH_3, and others can also be used. We execute all the results on a computer with an Intel 12th Gen i7 processor, 32GB RAM, and a 500-GB hard disk. Moreover, all the codes are executed in MATLAB version R2022a (The MathWorks Inc. 2022).

Deployment of Movable Sensors for Vehicular Pollution Monitoring

In this subsection, we present the results of the deployment of sensors alongside a road network. This subsection is divided into three parts. In the first part, the results of the analysis of sensor relocations as obtained by GA are provided. The purpose of the second part is to check the performance of the proposed GA in comparison to enumeration. In the last part, sensitivity analysis with the maximum number of relocations (i.e., R) is presented.

In this section, we assume that vehicular pollution is the sole reason for the pollution generated along the potential locations considered for the deployment of sensors on the road network. However, the optimization formulation does not restrict the potential locations to lie along the road network; thus, the ambient pollution generated from nonvehicular sources can also be considered (we will be subsequently presenting this in a case study). The standard network we consider for traffic simulation is used in transportation network analysis called the Sioux Falls network (Transportation Networks for Research Core Team 2024). Fig. 4 shows the Sioux Falls network.

To generate time-varying ambient pollution levels along the road network, we use the Cell Transmission Model (CTM), a widely accepted traffic simulation model (Daganzo 1994). CTM is a numerical approximation method to determine traffic flows using hydrodynamic traffic flow theory (Daganzo 1994; Ukkusuri et al. 2012). In CTM, the road network is divided into various parts called cells, and the flow pattern is observed for each cell at different time-steps. Since CTM divides the entire network into various cells, it is a mesoscopic traffic simulator and is more computationally efficient than microscopic traffic simulators. Fig. 4 shows 56 cells and 67 links in the Sioux Falls network. The centers of all the cells in the considered network represent the set V of the potential locations for sensor deployment. CTM takes inputs as the time-varying traffic demand and the road network on which the traffic needs to be simulated. In this road network, we consider a single origin-destination (O-D) pair given by cells 1 (origin) and 56 (destination), as shown in Fig. 4 (we consider a single O-D pair for ease of visualization and analysis, although CTM can consider multiple O-D pairs). We assume that the traffic demand between the given O-D pair is similar to the demand pattern of real-life traffic in which peaks occur in the morning and evening during a day (Nasir et al. 2014). There can be deviations in the demand pattern on some days due to unexpected events, but there is usually a morning peak and an evening peak across each day (Crawford et al. 2017). Fig. 5 shows the variation in traffic demand over time for the given O-D pair (here, the traffic demand is shown for every minute and for a total of 1,440 minutes, which is equal to the duration of a day).

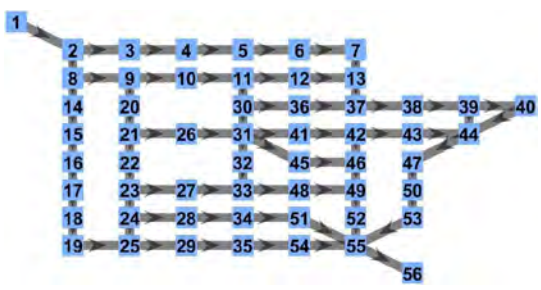

Fig. 4. Sioux Falls network.

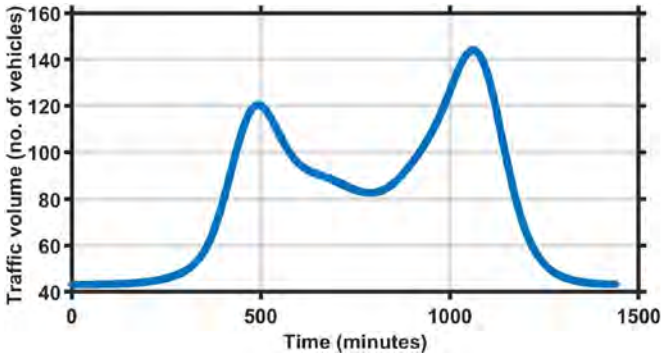

Fig. 5. Variation in traffic demand over time.

The total demand between the given O-D pair is equally distributed among all the paths that connect this O-D pair. This demand distribution assumption may not be the most realistic but is computationally efficient relative to distributing demand based on dynamic traffic assignment methods that often become computationally burdensome (Peeta and Ziliaskopoulos 2001). We simulate the traffic approximately every minute for the 24-h duration using CTM. To simulate the CTM, we use a traffic fundamental diagram of trapezoidal shape for each cell with outflow capacity equal to 1,800 vehicles h^{-1}, jam density equal to 200 vehicles km^{-1}, free-flow speed equal to 50 $km\,h^{-1}$ and the ratio of the shockwave speed to the free-flow speed as 0.4 (Zhu et al. 2013). The equations corresponding to CTM are not presented here for brevity, and interested readers may refer to Ukkusuri et al. (2012). The output of CTM is the number of vehicles present in each cell at different time-steps (i.e., the traffic occupancy of different cells at various time-steps). We use the traffic occupancy values obtained from CTM to obtain the average speed of vehicles for different cells at different time-steps using the method mentioned in Zhu et al. (2013). After that, we use the average speeds to obtain ambient $PM_{2.5}$ emission values (in terms of mg) as inputs for the optimal deployment of sensors through known emission factors (Huang et al. 2022; Zhu et al. 2013). We use ambient emission values in terms of mass (mg) to determine the values of e_t^i in this case study rather than ambient concentration values (in terms of mass/volume) because we assume that the emissions generated from each cell are dispersed uniformly across the cell and, thus, the corresponding concentration values are constant factor times the corresponding emission values (because all the cells in CTM are assumed to have identical sizes). Thus, the optimal deployment of sensors obtained using emissions is the same as that obtained using the corresponding concentration values because the optimal solution of an optimization problem does not change if the objective function is multiplied by a positive constant. We set the duration of a time-step in the optimal deployment of sensors to be equal to 4 h (i.e., the time gap between successive sensor relocations cannot be less than 4 h); thus, we average the emission values obtained per minute from the aforementioned process over the duration of 4 h to obtain a single emission value for each cell. Since we simulate CTM for 24 h, and the duration of a time-step in the optimization problem is 4 h, we have T equal to 6.

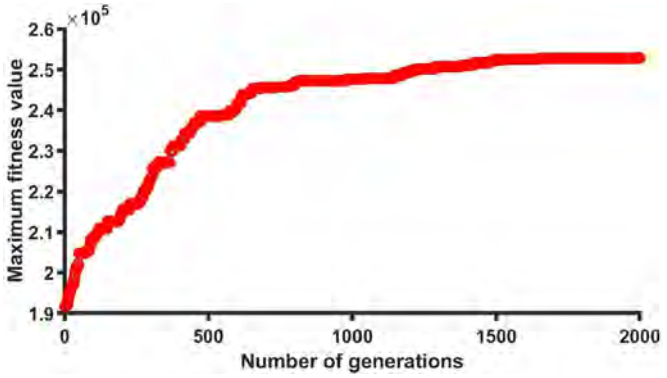

Fig. 6. Illustration of variation in maximum fitness value across generations in GA.

Analysis of Sensor Relocations over Time

In this section, we focus on the solutions obtained for the optimal deployment of sensors in the Sioux Falls network through GA. We set $k = 10$ (i.e., 10 sensors are deployed at each time-step) and $R = 24$ (i.e., the maximum number of relocations is equal to 24) and keep the other parameters as before. In this paper, we choose the values of k and R synthetically and subsequently provide the sensitivity analysis with these parameters. We set GA parameters such as crossover probability $P_c = 0.65$, mutation probability $P_m = 0.1$, maximum number of generations $G = 2000$, and predetermined threshold $\alpha = 0.01$. We set these parameters after doing several tests to ascertain that GA provides reasonable performance with these parameters. Fig. 6 illustrates the variation in the maximum fitness value across generations as obtained through GA. As the number of generations increases, the maximum fitness value obtained by GA increases because we perform elitism in GA. Moreover, the maximum fitness value converges when the number of generations reaches approximately 1,600, indicating that the parameters have been satisfactorily chosen for GA to reach convergence.

Fig. 7 shows the deployment of sensors for time-steps ranging from $t = 1$ to $t = 4$ as obtained by GA (note that $T = 6$ and, thus, the deployment obtained from GA is available for six time-steps but is only shown for the first four time-steps for brevity). At any given time-step, all the cells in the network are divided into two categories as follows: dark colored cells represent those cells where sensors are deployed at the given time-step, and light colored cells are those where sensors are not deployed at the given time-step. The $PM_{2.5}$ values (as obtained by the aforementioned traffic simulation process by CTM) across the region surrounding this road network are also presented to better analyze the solution obtained using GA. High $PM_{2.5}$ values are characterized by dark color and low $PM_{2.5}$ values are characterized by light color. Moreover, the aforementioned process of determining $PM_{2.5}$ values involving CTM calculates these values only for the cells across the road network and thus the $PM_{2.5}$ values for the remaining places in the shown region are interpolated. In Fig. 7, a sensor is deployed at cells such as 2, 8, 9, and 55 at all time-steps. That is because these cells are situated close to the origin and destination of the traffic demand and, thus, the $PM_{2.5}$ values are always high near these cells as is visible by the dark contours in the

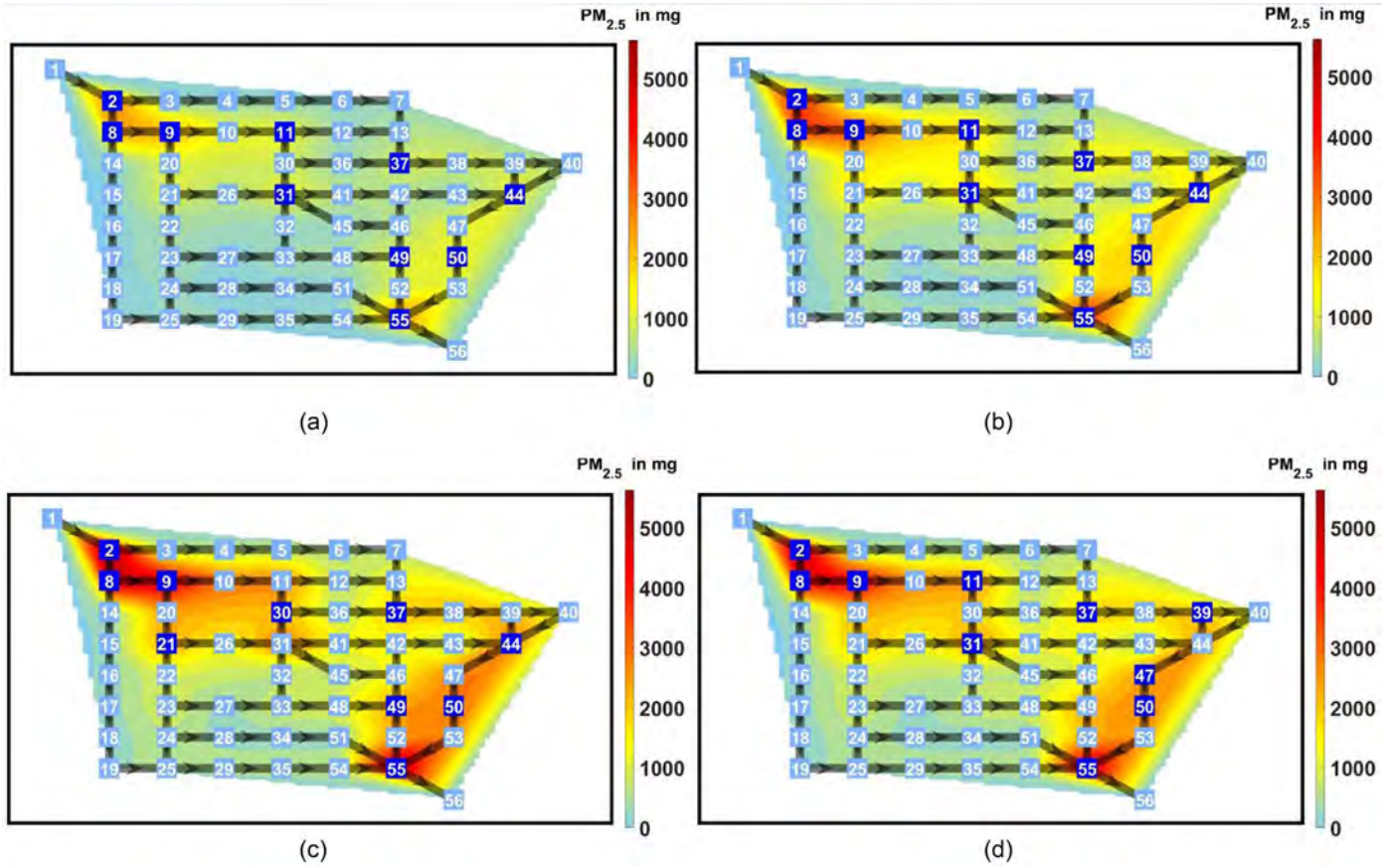

Fig. 7. Illustration of sensor deployment in Sioux Falls network as obtained by GA for time-steps $t = 1$ to $t = 4$: (a) $t = 1$; (b) $t = 2$; (c) $t = 3$; (d) $t = 4$.

J. Infrastruct. Syst., 2025, 31(4): 04025030

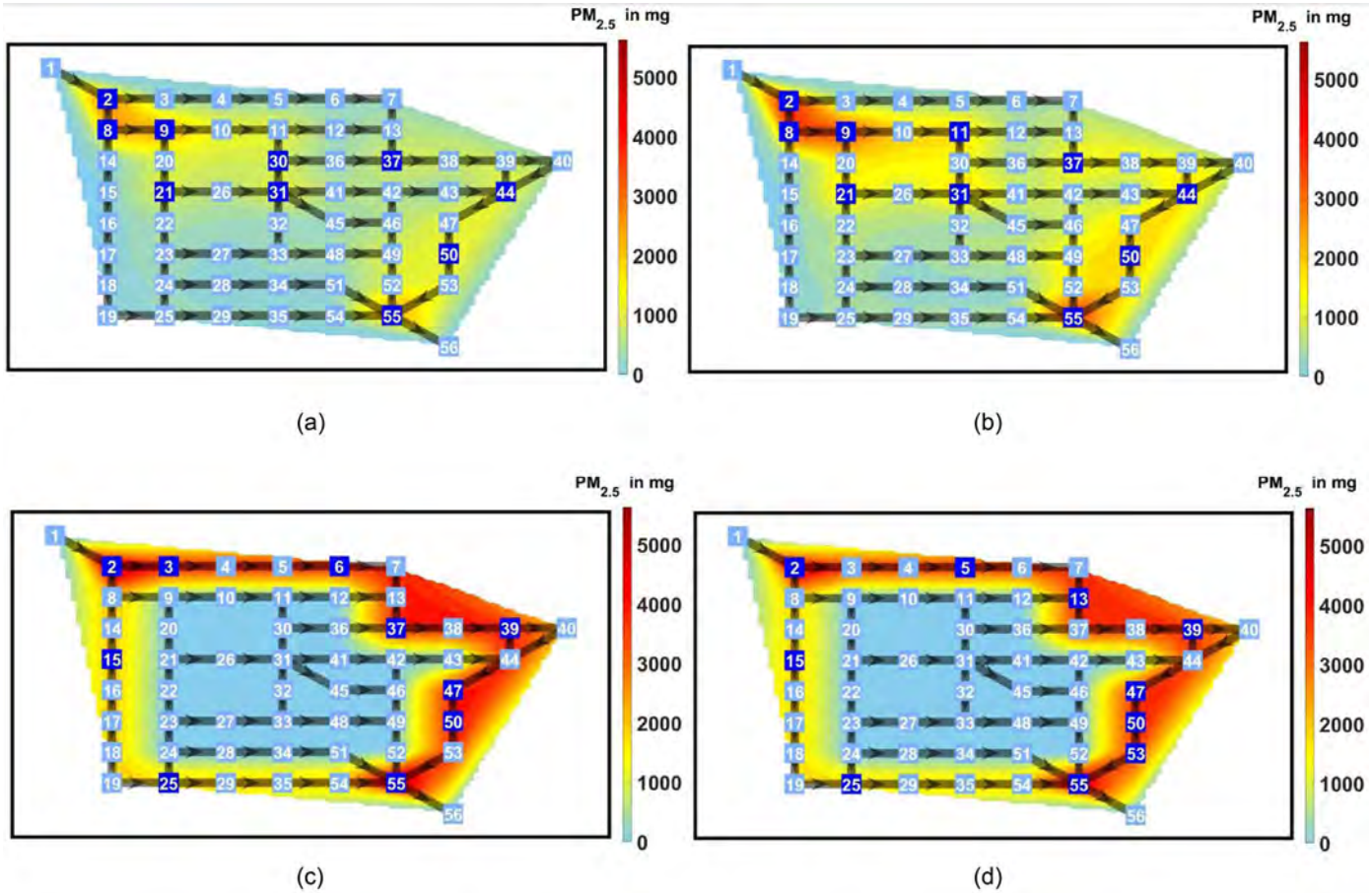

Fig. 8. Illustration of sensor deployment in Sioux Falls network as obtained by GA for time-steps $t = 1$ to $t = 4$ when link 8–9 is blocked for time-steps $t = 3$ and $t = 4$: (a) $t = 1$; (b) $t = 2$; (c) $t = 3$; (d) $t = 4$.

figure. It can also be seen that there are sensors at cells 11 and 31 in time-step 2; however, those sensors are relocated to cells 21 and 30 in time-step 3, signifying the importance of allowing sensor relocations within a day. Moreover, no sensor is deployed at cell 53 in time-step 4; however, a sensor is deployed at cell 31 in this time-step, even though the $PM_{2.5}$ value is higher at cell 53 than at cell 31 for this time-step. That is because there are already sensors deployed at cells adjacent to cell 53, i.e., cells 50 and 55; therefore, deploying sensors at cells other than cell 53 might maximize the overall satisfaction value (i.e., the objective function value). This shows that the potential deployment is determined by the following two factors: one of the factors is to prioritize those cells for deploying sensors that have high $PM_{2.5}$ values (i.e., e_t^i), and the other factor is to prioritize those cells on whose deployment the distance to the closest deployed sensor from each cell (i.e., d_t^i) is low (in line with the objective function given by $\sum_{t=1}^{T} \sum_{i \in V} e_t^i g(d_t^i)$).

A road may be blocked due to accidents, repair work, and others, which may change the $PM_{2.5}$ values around that road for the affected duration. To simulate such scenarios, we now block link 8–9 in the Sioux Falls network for time-steps $t = 3$ and $t = 4$. That is, for these two time-steps, no flow takes place through link 8–9. The total demand between the O-D pair $(1,56)$ and other parameters such as N, k, R, T, and others are kept the same as in the case when no link is blocked. Fig. 8 presents the sensor deployment obtained for different time-steps by GA for this case (as before, we only show the deployment for the first four time-steps for brevity, although $T = 6$).

Most paths connecting origin cell 1 and destination cell 56 pass through link 8–9; thus, when link 8–9 is blocked, traffic is diverted to the periphery of the network. Therefore, most $PM_{2.5}$ values are concentrated along the periphery of the network in time-steps 3 and 4 (this can be seen by the dark contour along the periphery of the network). Thus, when this link is blocked for time-steps 3 and 4, sensor deployment is observed only for the paths situated on the periphery of the network. For example, sensors are deployed at cells such as cells 8 and 9 in time-steps 1 and 2 but are not deployed in time-steps 3 and 4. Moreover, a sensor is deployed at cells 15 and 25 in time-step 3, but no sensor is deployed at cells 4, 40, and 44 at this time-step, even though these cells have higher $PM_{2.5}$ values than those at cells 15 and 25. That is because the cells adjacent to 4, 40 and 44 have sensors deployed in time-step 3. Thus, the optimal solution takes into account two factors previously mentioned: prioritizing cells with high $PM_{2.5}$ values and making sure that the distance to the closest deployed sensor from each cell is low.

Performance Comparison of GA with Enumeration

In this section, we evaluate the quality of the solution obtained by GA in comparison to the optimal solution and analyze the computational efficiency of GA. For that, we compare the results obtained by GA with those obtained by enumeration. Note that enumeration is a method to find the optimal solution by generating all the feasible solutions and comparing the feasible solutions to find the optimal solution. We consider the values of the parameters R and T to be the same as previously described (we discuss the details of other parameters such as N and k in subsequent discussions).

We compare the performance of GA with enumeration as the size of the problem increases. The proposed optimization problem falls in the category of combinatorial optimization because there are a finite number of feasible solutions (Papadimitriou and Steiglitz 1998). Thus, the total number of feasible solutions might increase rapidly with the increase in the size of the problem. One factor that determines the size of the problem is k (i.e., the number of sensors deployed at any time-step); therefore, we plan to check the performance of GA with enumeration as k increases. Since $k \leq N$, we also increase N as k increases. We fix $N/k = 2$ since the total number of combinations of k sensors out of N potential locations/cells (i.e., ${}^{N}C_{k}$) is maximum for a given N when N/k is approximately equal to 2 (Sleziak 2021). Since we fixed N/k to 2, we need to increase N (i.e., the number of cells for the potential deployment of sensors); thus, the set V of potential locations/cells for deployment increases as follows: if $k = 1$, then $N = 2$ and cells 1 and 2 in the Sioux Falls network are the potential locations of deployment; if $k = 2$, then $N = 4$ and cells 1, 2, 3, and 4 in Sioux Falls network are potential cells for deployment, and so on. Although the set V consisting of potential locations for sensors varies as k increases, $PM_{2.5}$ values taken as inputs for the optimal deployment of sensors are kept the same regardless of the k value and are obtained from the simulation of CTM over the entire Sioux Falls network without blocking any link at any time-step (with the parameters in the simulation being the same as that described in the previous section).

Fig. 9 provides the computational time (in log-scale) for GA and enumeration with varying numbers of sensors (i.e., k). The computational time for both GA and enumeration increases with an increase in k. The increase in the computational time for GA is observed because the number of boxes in a string of GA is equal to NT; thus, as k increases, the string size increases (since N increases as N/k is set to 2). For enumeration, the computational time increases with k as the total number of feasible solutions increases with the increase in k because of the following argument. Suppose R is sufficiently large such that it does not impose any constraint in the optimization of sensor deployment. Then, the total number of feasible solutions of the optimization problem is equal to $({}^{N}C_{k})^{T}$. For example, when we have $N/k = 2$ and $T = 6$, then the total number of feasible solutions when k is equal to 1, 2, and, 3 is equal to $({}^{2}C_{1})^{6}$ (i.e., 64), $({}^{4}C_{2})^{6}$ (i.e., 46656), and $({}^{6}C_{3})^{6}$ (i.e., 64,000,000), respectively. Thus, the total number of feasible solutions increases very rapidly with an increase in k (we say very rapidly because there is an exponential increase in the number of feasible solutions with an increase in k), as does the computational time. The aforementioned argument is provided for the case in which R is sufficiently large so that R does not impose any constraint in optimization; however, we expect a similar argument to hold when R imposes a constraint in optimization. We only plotted the computational time for enumeration until $k = 3$ in Fig. 9 because the computational time increases rapidly with k.

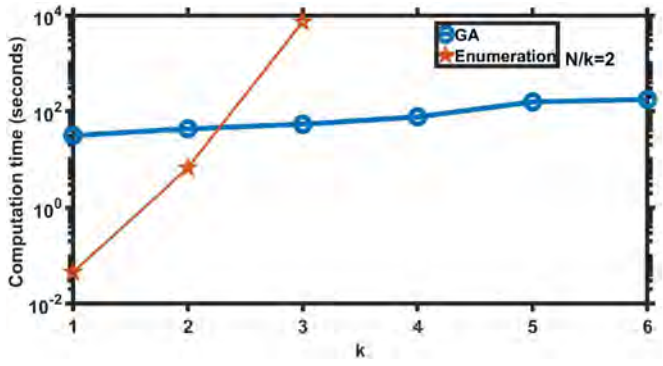

Fig. 9. Variation in computation time with k for different approaches.

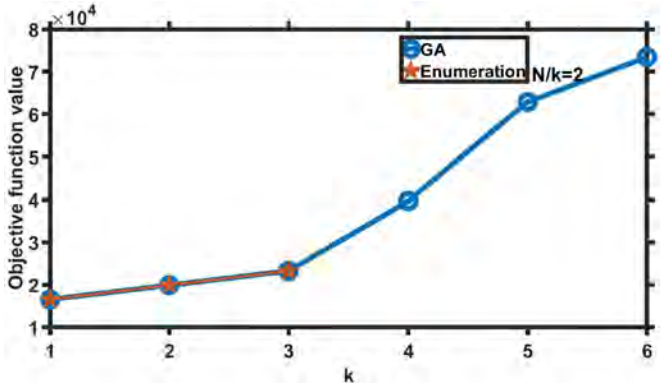

Fig. 10. Variation in objective function value with k for different approaches.

Fig. 10 presents the objective function value obtained by GA and enumeration with varying k. We plot the values for enumeration only until $k = 3$ since the computation time for enumeration increases very rapidly with k, as previously mentioned. Note that the objective function values obtained by GA and enumeration are almost the same (the relative difference in these values is less than 0.01%); thus, the line corresponding to GA is hidden behind that of enumeration when $k \leq 3$ in Fig. 10. The objective function value increases with an increase in k because the number of potential locations N (i.e., size of set V) for the deployment of sensors also increases and, thus, the objective function value (determined by $\sum_{t=1}^{T} \sum_{i \in V} e_t^i g(d_t^i)$) increases. From now on, we will only present results obtained by GA because the objective function value obtained by GA is very close (the relative difference is less than 0.01% for the tested cases) to that obtained by enumeration, and GA is computationally very efficient (the computational time taken by GA is approximately two orders of magnitude less than the computational time taken by enumeration when $k = 3$) as compared to enumeration.

Sensitivity Analysis with Maximum Number of Relocations

We now discuss how the objective function value varies with an increase in the maximum number of relocations R. While varying R, we set $N = 56$, $k = 10$, and $T = 6$. Fig. 11 shows the variation in the objective function value with R as obtained by GA.

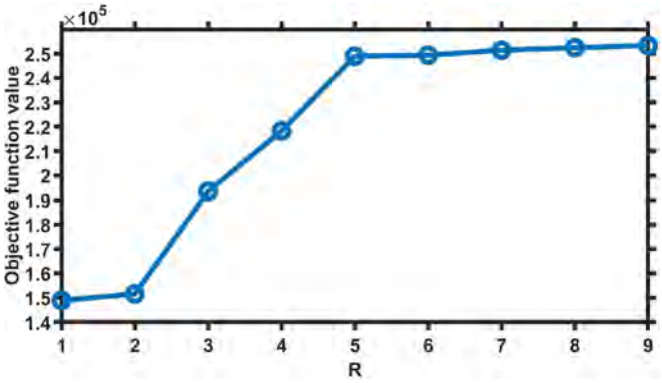

Fig. 11. Variation in objective function value with maximum number of relocations R.

The objective function value increases when R increases because as R increases, the constraint on the maximum number of relocations [given in Eq. (3)] becomes more relaxed and, thus, the total number of feasible solutions increases. Therefore, the objective function value will either remain the same or increase as R increases. The objective function value increases at a faster rate when R increases from 2 to 5 as compared to when R increases from 5 to 9. That is because at $R = 5$, the objective function value has reached close to the optimal value for the case in which there is no constraint on the maximum number of relocations (R), and further increasing R will not substantially increase the objective function value [we confirmed the optimal value when there is no constraint on R through extensive randomized sampling of feasible solutions (because running enumeration is not computationally practical in this case) and that value is equal to 2.54×10^5].

Deployment of Movable Sensors for Citywide Pollution Monitoring

In this section, we consider the deployment of movable sensors for citywide monitoring rather than deploying sensors only along the road networks, as shown in the last sections. The $PM_{2.5}$ values used as inputs in the optimal sensor deployment are obtained from a real-world source, specifically, from Lucknow—the capital of the state of Uttar Pradesh in India. Fig. 12 displays a map of Lucknow city. The circular points show the 36 potential locations for sensor deployment. The $PM_{2.5}$ data at these 36 locations are available through the air monitoring sensors deployed at these places. We use these $PM_{2.5}$ values as inputs for our sensor deployment problem considering a hypothetical case study in which it may not be possible to deploy sensors at all these 36 locations (i.e., $k < 36$), but it may be possible to relocate sensors across these locations to better capture time-varying pollution. The considered $PM_{2.5}$ data are for the month of November 2022 on an hourly basis. We use $PM_{2.5}$ values recorded by the sensors for the month of November because the variation in the $PM_{2.5}$ values is much higher in November than in most other months for the studied region. We set the duration of a time-step for sensor deployment (i.e., the smallest time gap between relocations of sensors) to be equal to one day and averaged the hourly $PM_{2.5}$ values across a day to obtain $PM_{2.5}$ values for different days. There were some missing data points (less than 3% of the entire dataset) for certain hours in our dataset; thus, those points were subsequently interpolated using the moving average method (Hansun 2013). We now consider the problem of optimal sensor deployment with $N = 36$ (because there are 36 potential locations in Lucknow), $k = 10$, and $R = 24$ (we discuss the value of T selected in the subsequent subsections). Note that the parameters used in GA are the same as previously used.

This section is divided into two subsections. In the first subsection, we present the results of the analysis of the solution obtained for sensor deployment across Lucknow. The purpose of the subsection after that is to perform sensitivity analysis with the maximum number of time-steps (T).

Analysis of Solution Obtained for Sensor Deployment

In this section, we focus on a typical solution provided by GA for the deployment of sensors for the considered region. We set $T = 6$ (i.e., we focus on sensor deployment for a total duration of six days during which sensor relocations can happen daily), and the remaining parameters are set as previously described. Fig. 13 provides the deployment of the sensors obtained by GA, with circular points representing the locations of sensors for time-steps $t = 1$ to $t = 4$ (as before, we only show the deployment for the first four time-steps for brevity, although $T = 6$). The contour plots at any time-step in Fig. 13 depict $PM_{2.5}$ values generated through interpolation based on the $PM_{2.5}$ data available for the 36 locations in Fig. 12. $PM_{2.5}$ values for all the time-steps in Fig. 13 are typically the highest in the central region of the city because this area is densely populated. $PM_{2.5}$ values at locations 14 and 25 at all time-steps are very low because these locations are situated far from the city area; thus, no sensor is deployed at these locations at any time-step. Moreover, sensors are deployed at locations 23 and 28 in time-step $t = 1$ because of the high $PM_{2.5}$ values that are observed at these locations even though these locations are far from the city center. The high $PM_{2.5}$ value at $t = 1$ at location 23 might be because of its proximity to National Highway 27, Outer Ring road and several educational institutes. Similarly, high $PM_{2.5}$ values at location 28 might be attributed to its close proximity to National Highway 30. Also, location 10 has higher $PM_{2.5}$ than location 19 in time-step 1 but a sensor is deployed at location 19 rather than at location 10 at this time-step, possibly because location 19 serves as the nearest location with a sensor to locations 22 and 24 (and thus the overall satisfaction is high with this deployment). The relocation of sensors can also be observed in several cases, e.g., there is no sensor at cell 28 in time-step $t = 4$ but it is present in the first three time-steps.

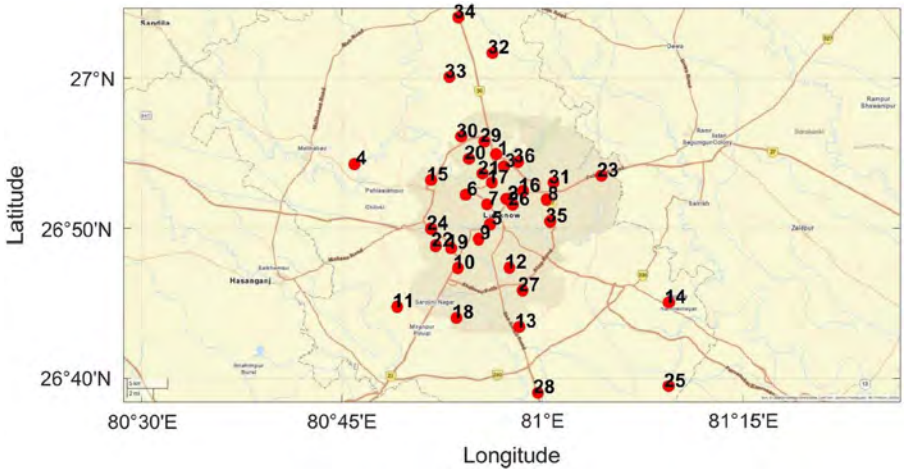

Fig. 12. Map of Lucknow. (Base map by Esri © OpenStreetMap contributors, TomTom, Garmin, Foursquare, METI/NASA, USGS.)

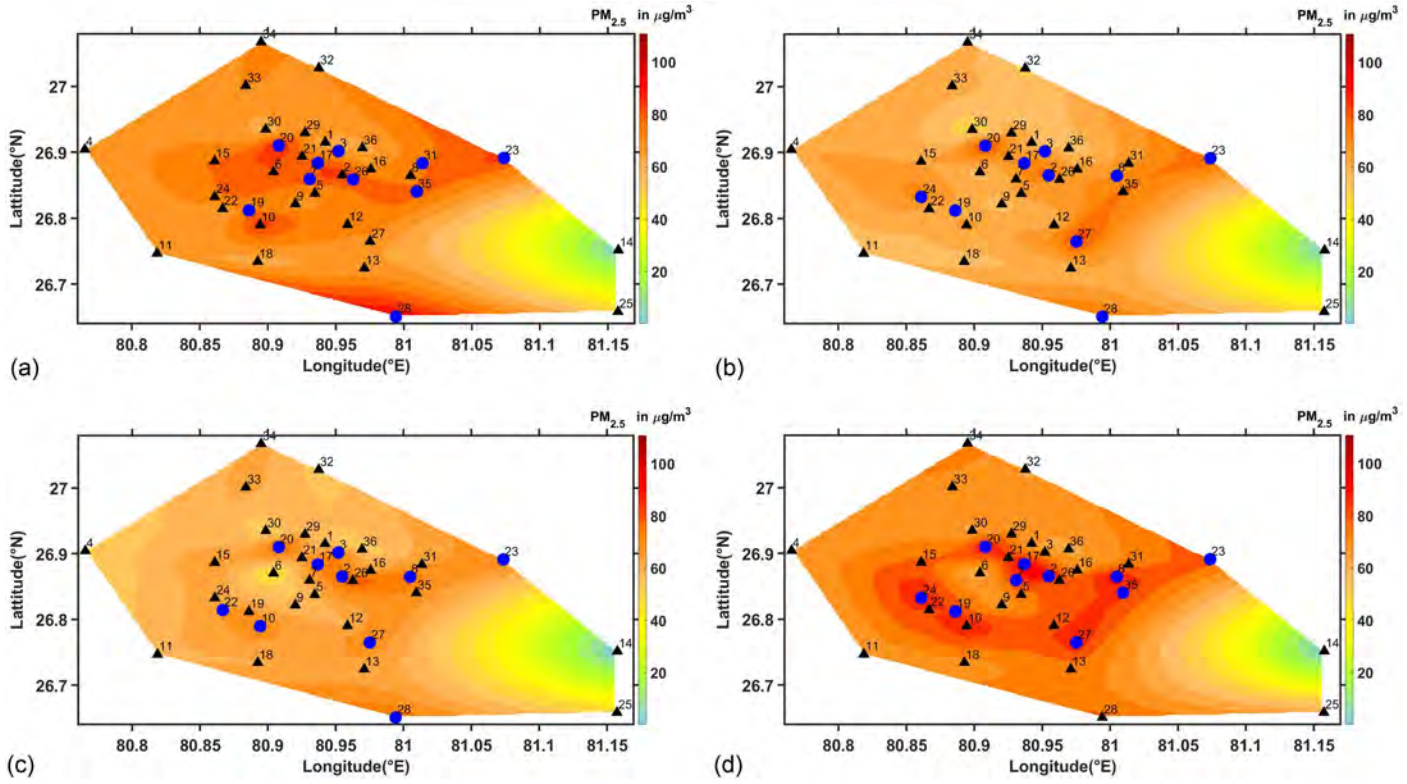

Fig. 13. Illustration of sensor deployment in Lucknow as obtained by GA for timesteps $t = 1$ to $t = 4$: (a) $t = 1$; (b) $t = 2$; (c) $t = 3$; (d) $t = 4$.

Sensitivity Analysis with Maximum Number of Time-Steps T

In this section, we present the sensitivity analysis with the maximum number of time-steps T. The parameters N, k, and R are kept the same as those in the previous section. Figs. 14 and 15 show the objective function value and computational time, respectively, when T varies from 6 to 14. It is possible to conduct sensitivity analysis with T in this case study by increasing from $T = 6$ to $T = 14$ while keeping the duration of a time-step in optimization equal to a day since we have the $PM_{2.5}$ data for the entire month of November (however, it is not possible to increase T in the previous case study while ensuring that the traffic demand is for an entire day, i.e., 24 h, and the duration of a time-step in the optimization does not become smaller than 4 h). Fig. 14 shows that the objective function value increases with T because the objective function [Eq. (1)] has a summation involving T terms. Additionally, the computational time in Fig. 15 increases as T increases because the size of the string is equal to NT and, thus, the computational time associated with GA increases with longer strings.

Fig. 15 shows that the computational time it takes for GA to obtain the solution when $T = 14$ is nearly 20 min. Thus, the proposed algorithm provides a practical way to solve this problem as the movable sensor deployment can be planned in advance even for a duration such as two weeks.

Discussion of Practical Data Collection Methods

In this section, we discuss practical methods for deriving $PM_{2.5}$ data. The most common way of obtaining $PM_{2.5}$ data is through air quality monitors and sensors. For example, we used the $PM_{2.5}$ data obtained through air quality sensors to develop the case study

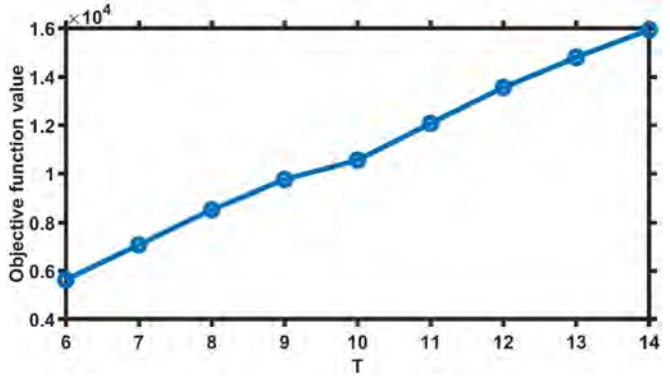

Fig. 14. Variation in objective function value with maximum number of time-steps T.

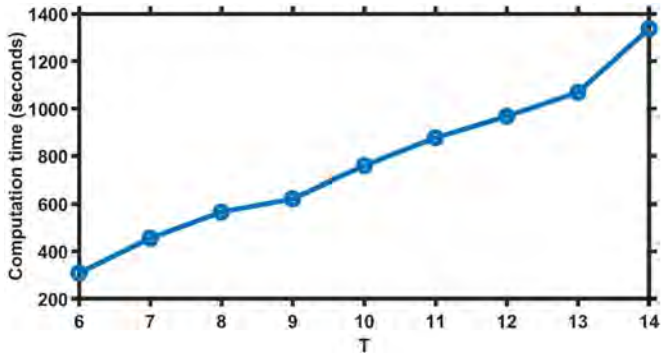

Fig. 15. Variation in computation time of GA value with maximum number of time-steps T.

for the dynamic deployment of sensors in Lucknow. Public agencies such as the Central Pollution Control Board (CPCB) and several private agencies maintain air quality reference monitors and sensors in India to track $PM_{2.5}$ data (there are similar agencies elsewhere across the world) (Sharma and Mauzerall 2022). In places in which air quality sensors or monitors may not be deployed, $PM_{2.5}$ values can also be estimated using satellite data from sources such as the Moderate Resolution Imaging Spectroradiometer (MODIS) aboard NASA's Terra and Aqua satellites (Li et al. 2011). These sources use Aerosol Optical Depth measurements to derive $PM_{2.5}$ levels through statistical models. Moreover, spatial interpolation techniques such as Kriging and inverse distance weighting (IDW) are used to obtain $PM_{2.5}$ values for locations at which sensors or monitors are not available to interpolate the required values from locations having monitors and sensors (Shukla et al. 2020).

Important Managerial Insights

We now present important managerial insights from the results previously presented. Note that not all locations with high $PM_{2.5}$ values require sensors if there are already sensors placed nearby. Moreover, the dynamic deployment of sensors is useful when there is a reasonable change in $PM_{2.5}$ values over time. Thus, when there is no significant variation in $PM_{2.5}$ values over time, it is suggested that static optimization formulation be used for deployment because doing so will be computationally less burdensome. The decision on whether the time-step for relocation should be a few hours or a day depends on the objective of the air quality monitoring agencies. For example, if the target is to monitor changes in ambient air pollution due to changes in travel patterns within a day, then performing relocations within a day makes sense. In contrast, if the objective is to monitor changes in air pollution due to industrial activities or construction operations, it might be better to relocate sensors within a week because there might be different levels of these activities across a week. For example, there would be differences in such activities on weekdays when compared to on weekends. Furthermore, the time-step for relocation is also governed by other considerations, e.g., setting a small time-step for relocations amounts to frequent relocations and, thus, adds to the costs associated with relocations. Moreover, the time-step for relocating sensors should be sufficiently large for shifting sensors across various potential locations.

Limitations

We now mention the limitations of the proposed approach for the deployment of air quality sensors (in the discussion of future directions in the next section, we provide suggestions to address these limitations). We assumed that the $PM_{2.5}$ values used as inputs in the optimization formulation are deterministic; thus, we do not allow any provision for inaccuracies or uncertainty in $PM_{2.5}$ values. Moreover, we considered only a single type of sensor but there can be various types of sensors based on their cost, accuracy, and other factors—and considering that will also be important. There can be several other real-world constraints that have not been considered in this paper; for example, deploying a minimum number of sensors at all times among a set of important locations (such as industrial areas, congested roads, and others) might be required. When performing traffic simulation with CTM to obtain $PM_{2.5}$ values, we assumed a single O-D pair, a particular demand distribution across a day, and an equal distribution of traffic across all paths of the O-D pair. However, these assumptions may not always hold in the real world. Moreover, mesoscopic traffic simulators such as CTM are computationally more efficient than microscopic traffic simulators; however, the former are less realistic than the latter and, thus, testing with microscopic simulators may provide more realistic $PM_{2.5}$ values. These assumptions related to traffic properties and traffic simulation impact the obtained $PM_{2.5}$ values; however, the proposed optimization formulation can also handle $PM_{2.5}$ values obtained by other assumptions.

Conclusions

In this paper, we introduced a formulation for achieving the optimal deployment of sensors that can be relocated over time for air pollution monitoring to maximize the overall satisfaction function. In this optimization problem, we consider constraints on the number of sensors deployed at any time-step and the maximum number of relocations that can occur. We formulated this problem as a nonlinear integer program. To efficiently solve the optimization problem, we developed a genetic algorithm. We evaluated the performance of this algorithm by comparing it with enumeration (where all the feasible solutions are listed and the optimal solution is determined). It is found that the objective function values obtained by GA are very close to those obtained by enumeration, and GA is computationally very efficient compared to enumeration for the tested instances. We tested the developed algorithm for two case studies. We first considered the case in which sensors are deployed along a road network to focus on vehicular pollution, and the inputs on $PM_{2.5}$ values for optimal deployment of sensors were obtained through traffic simulation. In this case study, we considered two scenarios: first, when no link in the network is blocked and, second, when one of the links is blocked. The blocked link scenario simulates conditions such as road blockage due to major accidents, road maintenance activities, and others. We observed that when a link is blocked, the distribution of $PM_{2.5}$ values changed drastically across the road network, leading to sensor relocations to the periphery of the network compared to the scenario in which the link is not blocked. This shows the need to optimally relocate sensors to take into account the time-varying dynamics of pollution. In the second case study, we considered deploying sensors across different locations in a city using real-world $PM_{2.5}$ values as inputs for optimal deployment. The $PM_{2.5}$ values changed significantly in the city during the duration of deployment; thus, there were several locations in which sensors were relocated based on the solution provided by the GA. Thus, this case study also highlights the need to optimally deploy the sensors over time.

There can be several interesting future extensions of this work. In this paper, we considered the ambient concentration values as weights for different locations in the focused region; however, considering other weights such as mobility patterns, population, and others while deploying sensors might be an interesting extension. Considering factors such as wind direction and wind speed while focusing on optimal deployment of sensors would also be an interesting future extension. We suggested using sources such as traffic simulation and satellite imaging to obtain estimates of concentration values. However, sources such as satellite-based imaging and traffic simulation may not be that accurate in obtaining $PM_{2.5}$ values compared to sensors; thus, one may update the estimates of $PM_{2.5}$ values across the network through the data obtained after deploying sensors. Thus, one may extend this work to perform real-time deployment of sensors by using algorithms that can predict $PM_{2.5}$ values in the near future. We did not consider any provision for having inaccuracies or uncertainty in the input concentration values and considering that will also be an important step. A possible approach is to consider the concentration values

as stochastic to account for uncertainty in the concentration values. Considering various types of sensors based on their cost, accuracy, and other factors will also be important. The difference in sensor costs can be incorporated by ensuring separate costs in the budget constraint. There can also be placement-related constraints such as the impracticability of deploying sensors at some locations (such as open areas, areas near water bodies, and others) and the requirement of deploying sensors at important locations (such as industrial areas, congested roads, and others). These constraints can also be extended for time-varying conditions [from the static setting of Ajnoti et al. (2024)]. When testing using multiple O-D pairs, other demand distributions and traffic assignment procedures during traffic simulation will also be an important step. Exploring microscopic traffic simulators such as Simulation of Urban Mobility (SUMO), which provide more realistic traffic simulations, will also be important. Testing the proposed formulation on a larger study region with real-world data will provide richer visualization and analysis of sensor relocations and, thus, will be an important future study. Exploring recently developed machine learning-based optimization approaches for the dynamic deployment of sensors is also an interesting direction. Finally, testing the performance of the developed GA with other approaches such as commercial solvers is also an important future direction.

Data Availability Statement

Some or all data, models, or codes that support the findings of this study are available from the corresponding author upon reasonable request.

Acknowledgments

This research work was supported by Duke University, Office of Research Support under the Subaward No. 349-0685. We also acknowledge sources that provided indirect support (other than direct funding) for this work. We first extend our heartfelt gratitude to Michael Howard Bergin and Sandeep Madhwal for Lucknow's sensor data. Sachchida Nand Tripathi also acknowledges the support received under Centre of Excellence (CoE) Advanced Technologies for Monitoring Air-quality iNdicators (ATMAN) approved by the PSA office, Government of India, and supported by a group of philanthropic funders, including the Bloomberg Philanthropies, Open Philanthropy, and Clean Air Fund.

Author Contributions

Saurabh Kumar Maurya: Data curation; Formal analysis; Investigation; Software; Visualization; Writing – original draft; Writing – review and editing. Hemant Gehlot: Conceptualization; Formal analysis; Investigation; Methodology; Supervision; Writing – review and editing. Sachchida Nand Tripathi: Funding acquisition; Resources; Supervision.

Notation

The following symbols are used in this paper:

e_t^i = concentration value of focused pollutant at location i for time-step t;
d_{ij} = distance between locations i and j;
d_t^i = distance between location i and closest deployed sensor from i in time-step t;
G = maximum number of generations/iterations allowed in GA;
$g(d_t^i)$ = satisfaction function of location i in time-step t;
k = number of sensors deployed at any time-step t;
N = total number of potential locations for sensor deployment;
P_c = crossover probability in GA;
P_m = mutation probability in GA;
p_t^i = binary variable equal to 0 if $z_{t-1}^i = z_t^i$ and equal to 1 otherwise;
R = total number of relocations that can take place across T time-steps;
T = maximum number of time-steps for which we focus on deploying sensors in a given area;
V = set consisting of all potential locations for deployment of sensors in a given area;
y_t^{ij} = binary variable equal to 1 if the closest location from i that has a sensor deployed in time-step t is location j and equal to 0 otherwise;
z_t^i = binary variable equal to 1 if a sensor is deployed at location i in time-step t and equal to 0 otherwise; and
α = predetermined threshold for termination in GA.

References

Ajnoti, N., H. Gehlot, and S. N. Tripathi. 2024. "Hybrid instrument network optimization for air quality monitoring." *Atmos. Meas. Tech.* 17 (6): 1651–1664. https://doi.org/10.5194/amt-17-1651-2024.

Apte, J. S., K. P. Messier, S. Gani, M. Brauer, T. W. Kirchstetter, M. M. Lunden, J. D. Marshall, C. J. Portier, R. C. Vermeulen, and S. P. Hamburg. 2017. "High-resolution air pollution mapping with Google street view cars: Exploiting big data." *Environ. Sci. Technol.* 51 (12): 6999–7008. https://doi.org/10.1021/acs.est.7b00891.

Castell, N., F. R. Dauge, P. Schneider, M. Vogt, U. Lerner, B. Fishbain, D. Broday, and A. Bartonova. 2017. "Can commercial low-cost sensor platforms contribute to air quality monitoring and exposure estimates?" *Environ. Int.* 99 (Feb): 293–302. https://doi.org/10.1016/j.envint.2016.12.007.

CleanAir. 2023. "Aeroqual s500 starter kit." Accessed December 16, 2024. https://www.cleanair.com/product/aeroqual-s500-starter-kit/.

Crawford, F., D. Watling, and R. Connors. 2017. "A statistical method for estimating predictable differences between daily traffic flow profiles." *Transp. Res. Part B Methodol.* 95 (Jan): 196–213. https://doi.org/10.1016/j.trb.2016.11.004.

Cummings, L. E., J. D. Stewart, R. Reist, K. M. Shakya, and P. Kremer. 2021. "Mobile monitoring of air pollution reveals spatial and temporal variation in an urban landscape." *Front. Built Environ.* 7 (May): 648620. https://doi.org/10.3389/fbuil.2021.648620.

Daganzo, C. F. 1994. "The cell transmission model: A dynamic representation of highway traffic consistent with the hydrodynamic theory." *Transp. Res. Part B Methodol.* 28 (4): 269–287. https://doi.org/10.1016/0191-2615(94)90002-7.

Deb, K. 2001. Vol. 16 of *Multi-objective optimization using evolutionary algorithms*. Chichester, UK: Wiley.

Deville Cavellin, L., S. Weichenthal, R. Tack, M. S. Ragettli, A. Smargiassi, and M. Hatzopoulou. 2016. "Investigating the use of portable air pollution sensors to capture the spatial variability of traffic-related air pollution." *Environ. Sci. Technol.* 50 (1): 313–320. https://doi.org/10.1021/acs.est.5b04235.

Gao, M., J. Cao, and E. Seto. 2015. "A distributed network of low-cost continuous reading sensors to measure spatiotemporal variations of PM2.5 in Xi'an, China." *Environ. Pollut.* 199 (Apr): 56–65. https://doi.org/10.1016/j.envpol.2015.01.013.

Hagemann, R., U. Corsmeier, C. Kottmeier, R. Rinke, A. Wieser, and B. Vogel. 2014. "Spatial variability of particle number concentrations and NO_x in the Karlsruhe (Germany) area obtained with the mobile

laboratory 'AERO-TRAM'." *Atmos. Environ.* 94 (Sep): 341–352. https://doi.org/10.1016/j.atmosenv.2014.05.051.

Hansun, S. 2013. "A new approach of moving average method in time series analysis." In *Proc., 2013 Conf. on New Media Studies (CoNMedia)*, 1–4. New York: IEEE.

Huang, H., J. Zhang, H. Hu, S. Kong, S. Qi, and X. Liu. 2022. "On-road emissions of fine particles and associated chemical components from motor vehicles in Wuhan, China." *Environ. Res.* 210 (Jul): 112900. https://doi.org/10.1016/j.envres.2022.112900.

Jerrett, M., A. Arain, P. Kanaroglou, B. Beckerman, D. Potoglou, T. Sahsuvaroglu, J. Morrison, and C. Giovis. 2005. "A review and evaluation of intraurban air pollution exposure models." *J. Exposure Sci. Environ. Epidemiol.* 15 (2): 185–204. https://doi.org/10.1038/sj.jea.7500388.

Karp, R. M. 2010. *Reducibility among combinatorial problems*. New York: Springer.

Li, C., N. C. Hsu, and S.-C. Tsay. 2011. "A study on the potential applications of satellite data in air quality monitoring and forecasting." *Atmos. Environ.* 45 (22): 3663–3675. https://doi.org/10.1016/j.atmosenv.2011.04.032.

Lim, C. C., H. Kim, M. R. Vilcassim, G. D. Thurston, T. Gordon, L.-C. Chen, K. Lee, M. Heimbinder, and S.-Y. Kim. 2019. "Mapping urban air quality using mobile sampling with low-cost sensors and machine learning in Seoul, South Korea." *Environ. Int.* 131 (Oct): 105022. https://doi.org/10.1016/j.envint.2019.105022.

Lin, H., et al. 2017. "Hourly peak $PM_{2.5}$ concentration associated with increased cardiovascular mortality in Guangzhou, China." *J. Exposure Sci. Environ. Epidemiol.* 27 (3): 333–338. https://doi.org/10.1038/jes.2016.63.

Morrison, D. R., S. H. Jacobson, J. J. Sauppe, and E. C. Sewell. 2016. "Branch-and-bound algorithms: A survey of recent advances in searching, branching, and pruning." *Discrete Optim.* 19 (Feb): 79–102. https://doi.org/10.1016/j.disopt.2016.01.005.

Nalakurthi, N. V. S. R., I. Abimbola, T. Ahmed, I. Anton, K. Riaz, Q. Ibrahim, A. Banerjee, A. Tiwari, and S. Gharbia. 2024. "Challenges and opportunities in calibrating low-cost environmental sensors." *Sensors* 24 (11): 3650. https://doi.org/10.3390/s24113650.

Nasir, M. K., R. Md Noor, M. A. Kalam, and B. M. Masum. 2014. "Reduction of fuel consumption and exhaust pollutant using intelligent transport systems." *Sci. World J.* 2014 (1): 836375. https://doi.org/10.1155/2014/836375.

Papadimitriou, C. H., and K. Steiglitz. 1998. "Combinatorial optimization: Algorithms and complexity." Accessed October 13, 2024. https://books.google.co.in/books?id=cDY-joeCGoIC.

Peeta, S., and A. K. Ziliaskopoulos. 2001. "Foundations of dynamic traffic assignment: The past, the present and the future." *Networks Spatial Econ.* 1 (3): 233–265. https://doi.org/10.1023/A:1012827724856.

Sharma, D., and D. Mauzerall. 2022. "Analysis of air pollution data in India between 2015 and 2019." *Aerosol Air Qual. Res.* 22 (2): 210204. https://doi.org/10.4209/aaqr.210204.

Shi, Y., K. K.-L. Lau, and E. Ng. 2016. "Developing street-level $PM_{2.5}$ and PM_{10} land use regression models in high-density Hong Kong with urban morphological factors." *Environ. Sci. Technol.* 50 (15): 8178–8187. https://doi.org/10.1021/acs.est.6b01807.

Shukla, K., P. Kumar, G. S. Mann, and M. Khare. 2020. "Mapping spatial distribution of particulate matter using kriging and inverse distance weighting at supersites of megacity Delhi." *Sustainable Cities Soc.* 54 (Mar): 101997. https://doi.org/10.1016/j.scs.2019.101997.

Sleziak, M. 2021. "Combinatorial proof that central binomial coefficients are the largest ones." Accessed October 13, 2024. https://math.stackexchange.com/q/2496836.

Spinelle, L., M. Gerboles, M. G. Villani, M. Aleixandre, and F. Bonavitacola. 2017. "Field calibration of a cluster of low-cost commercially available sensors for air quality monitoring. Part B: No, CO and CO_2." *Sens. Actuators, B* 238 (Nov): 706–715. https://doi.org/10.1016/j.snb.2016.07.036.

Sun, C., V. O. Li, and J. C. Lam. 2018. "Optimal citizen-centric sensor placement for citywide environmental monitoring: A submodular approach." In *Proc., IEEE Int. Conf. on Sensing, Communication and Networking (SECON Workshops)*, 1–4. New York: IEEE.

Sun, C., V. O. K. Li, J. C. K. Lam, and I. Leslie. 2019. "Optimal citizen-centric sensor placement for air quality monitoring: A case study of city of Cambridge, the United Kingdom." *IEEE Access* 7 (Apr): 47390–47400. https://doi.org/10.1109/ACCESS.2019.2909111.

The MathWorks Inc. 2022. "Matlab version: 9.12.0.2327980 (r2022a) update 7." Accessed October 13, 2024. https://www.mathworks.com.

Times of India. 2023. "Centre asks states not to procure imported air quality monitors, indigenous systems to be deployed." Accessed December 16, 2024. https://timesofindia.indiatimes.com/india/centre-asks-states-not-to-procure-imported-air-quality-monitors-indigenous-systems-to-be-deployed/articleshow/95901936.cms.

Transportation Networks for Research Core Team. 2024. "Transportation networks for research." Accessed April 20, 2024. https://github.com/bstabler/TransportationNetworks.

Ukkusuri, S. V., L. Han, and K. Doan. 2012. "Dynamic user equilibrium with a path based cell transmission model for general traffic networks." *Transp. Res. Part B Methodol.* 46 (10): 1657–1684. https://doi.org/10.1016/j.trb.2012.07.010.

Van Poppel, M., J. Peters, and N. Bleux. 2013. "Methodology for setup and data processing of mobile air quality measurements to assess the spatial variability of concentrations in urban environments." *Environ. Pollut.* 183 (Dec): 224–233. https://doi.org/10.1016/j.envpol.2013.02.020.

Whitehill, A. R., M. Lunden, B. LaFranchi, S. Kaushik, and P. A. Solomon. 2024. "Mobile air quality monitoring and comparison to fixed monitoring sites for instrument performance assessment." *Atmos. Meas. Tech.* 17 (9): 2991–3009. https://doi.org/10.5194/amt-17-2991-2024.

WHO (World Health Organization). 2022. "Ambient (outdoor) air pollution." Accessed March 16, 2024. https://www.who.int/news-room/fact-sheets/detail/ambient-(outdoor)-air-quality-and-health.

Yang, J., B. Shi, Y. Shi, S. Marvin, Y. Zheng, and G. Xia. 2020. "Air pollution dispersal in high density urban areas: Research on the triadic relation of wind, air pollution, and urban form." *Sustainable Cities Soc.* 54 (Mar): 101941. https://doi.org/10.1016/j.scs.2019.101941.

Zhu, F., H. K. Lo, and H.-Z. Lin. 2013. "Delay and emissions modelling for signalised intersections." *Transportmetrica B: Transport Dyn.* 1 (2): 111–135. https://doi.org/10.1080/21680566.2013.821689.