A probabilistic optimization model for allocating freeway sensors

Adam Danczyk a, Xuan Di b, Henry X. Liu b,c,*

a Department of Civil, Environmental, and Geo-Engineering, University of Minnesota, Twin Cities, United States
b Department of Civil and Environmental Engineering, University of Michigan, Ann Arbor, Michigan, United States
C University of Michigan Transportation Research Institute, Ann Arbor, Michigan, United States

Abstract

This paper proposes a sensor location model to identify a sensor configuration that minimizes overall freeway performance monitoring errors while considering the consequences of probabilistic sensor failures. To date, existing sensor location models for freeway monitoring inherently assume that either deployed sensors never fail or the consequences of sensor failure are trivial matters. However, history has revealed that neither assumption is realistic, suggesting that ignoring failures in sensor allocation models may actually produce a significantly suboptimal configuration in the real world. Our work addresses this dilemma by developing a probabilistic optimization model that will minimize the error expectation by examining all possible failure scenarios, each with an occurrence probability. To ensure the scenario completeness and uniqueness, a sensor failure scenario is represented by using a binary string with 1 indicating an operational sensor at a given site and 0 for sensor failure or no sensor deployed. When applied to a case study network, it is shown that an optimal configuration that considers sensor failure is significantly different from an optimal configuration that ignores sensor failure, revealing that sensor failures pose non-trivial consequences on performance monitoring accuracy.

1. Introduction

In recent years, the need for highly accurate measurements of system performance has promoted the development of sensor location models to optimize the allocation of limited sensor resources to minimize measurement errors. However, the sensor location models to date inherently assume that either deployed sensors never fail or that the consequences of sensor failure are trivial matters. Neither assumption is true, as history reveals real cases of failure rates that exceed 25% of all sensor deployments (Federal Highway Administration, 2006). Moreover, the necessary repairs for these failures may be too costly for transportation agencies on a limited budget, therefore prolonging the state of failure and magnifying the consequences of inaccurate performance monitoring. Given the increase of traffic management technologies (e.g. ramp metering, travel time reporting, etc.) that rely on accurate information, the need for a sensor configuration that is robust to sensor failure is ever-so-critical, but existing sensor location models designed for idealized no-failure environments may not produce the “best” configuration. The question becomes how such an optimal configuration could be found within the complexity of an imperfect world.

* Corresponding author at: Department of Civil and Environmental Engineering, University of Michigan, Ann Arbor, Michigan, United States.
E-mail address: henryliu@umich.edu (H.X. Liu).
The sensor location problem for no-failure sensors has been extensively studied in the past. A large body of literature has contributed to the exploration of the relationship between performance measure accuracies and density of roadway sensor deployment (Ozbay et al., 2004; Kwon et al., 2006; Ban et al., 2007; Barton et al., 2007). Fujito et al. (2006), however, revealed that there does not exist a systematic trend regarding whether travel time is over- or underestimated as sensor spacing increases. The resultant performance highly depends on the positions of the removed sensors. Accordingly, a strategic deployment was proposed that sensors should be placed at bottlenecks. Such a strategic deployment concept has also been used in numerous network-based sensor location problems, such as for inferring link flow (Hu et al., 2009; Ng, 2012; He, 2013), identifying origin–destination paths (Yang and Zhou, 1998; Bianco et al., 2001; Yang et al., 2006; Fei and Mahmassani, 2007, 2011; Fei et al., 2007, 2013; Simonelli et al., 2012; Hu and Liou, 2014), estimating origin–destination demands (Zhou and List, 2010), toll collection purposes (Zhang and Yang, 2004; Zangui et al., 2015), predicting travel time (Park and Haghani, 2015), monitoring traffic to reduce network risks (Gendreau et al., 2000) (e.g., policing for drunk drivers), and improving emergency response and crash characterization (Geetla et al., 2014). Interested readers can refer to Gentili (Park and Haghani, 2015), monitoring traffic to reduce network risks (Gendreau et al., 2000) (e.g., policing for drunk drivers), and improving emergency response and crash characterization (Geetla et al., 2014). Interested readers can refer to Gentili (Park and Haghani, 2015), monitoring traffic to reduce network risks (Gendreau et al., 2000) (e.g., policing for drunk drivers), and improving emergency response and crash characterization (Geetla et al., 2014).

Other sensor location models have been designed for point sensors along one-directional freeway corridors to minimize performance monitoring errors and allow for additional budgetary restrictions. Sherali et al. (2006) proposed a model for optimally allocating automatic vehicle identification (AVI) tag readers along a transportation corridor to optimize performance measure accuracies. Ban et al. (2009) produced a dynamic programming model to determine optimal locations of movable point sensors to compute freeway travel times. While easily solvable through a polynomial solution algorithm, the model in Ban et al. (2009) can only solve for a fixed number of sensors and does not account for sensor failures.

Liu and Danczyk (2009) and Danczyk and Liu (2010) proposed a non-linear model to optimally allocate point sensors, particularly loop detectors. This model only requires some basic knowledge of traffic conditions on the freeway corridor in question and is subject to monetary and resource budgetary constraints. It is reformulated into a mixed-integer linear model that answers the same question, where a ‘Neighboring Sensor’ assumption is employed to ensure performance could only be assessed between neighboring sensors, as generally done in the practice. While this mixed-integer linear model is easier to solve than its non-linear predecessor, it assumes that sensors do not fail, thus producing a configuration that may not be the “best” when failures occur.

This paper aims to propose a probabilistic sensor location model for point sensors along freeway corridors that considers the probability of sensor failures and seeks an optimal configuration that minimizes the expectation of performance monitoring errors. To date, sensor failure remains largely unexplored due to the complexity involved with such failures, with a noticeable exception of Li and Ouyang (2011) focusing on the vehicle ID inspection sensors. This work bridges a void in the knowledge regarding sensor location problem and sensor failures by addressing both elements. The proposed model can be solved using a heuristic search procedure to reach a viable solution.

The remainder of the paper is organized as follows. Section 2 begins by discussing briefly the mixed-integer linear sensor location model proposed by Danczyk and Liu (2010), where the fundamental ideas of the no-failure model are reestablished for the case of sensor failure. In Section 3, constraints are formulated to guarantee that all failure scenarios are uniquely identified and a probabilistic optimization model is proposed. Given the complexity of the proposed model, in Section 4, a heuristic search procedure is proposed to identify a configuration that is optimal. Lastly, a case study network with predetermined failure probabilities is used in Section 5 to demonstrate that accounting for sensor failures produces a different configuration than that without considering sensor failures. Conclusions and possible application of the proposed model are discussed in Section 6.

2. Formulation of the failure-based sensor location problem

Since the failure-based sensor location problem is closely related to our previous work on the no-failure model, for the completeness of this paper, we start our discussion by a brief overview of the mixed-integer formulation for no-failure configurations. More details on the no-failure model can be found in Danczyk and Liu (2010).

Let us consider a typical freeway corridor, such as the one shown in Fig. 1. The freeway receives an entry flow rate at its upstream point \( q_0 \) during the study period, \( T \). Along the route, entrance ramps add additional flow (e.g. \( q_1 \), \( q_3 \), \( q_5 \)) while exit ramps reduce flow (e.g. \( q_2 \), \( q_4 \), \( q_6 \)). The freeway is divided into \( N \) cells, each cell being designated as a potential site to place a sensor. Without loss of generality, we assume that this sensor would be located in the middle of the cell. These cells do not need to be of equal length, but for purposes of this work they will be uniform.

![Fig. 1. Sketch for an example freeway, divided into cells.](image-url)
The problem now becomes discretized, where the binary variable \( y_i \) represents sensor site coverage at site \( i \) (a value of 1 if a sensor is present, or a value of 0 if not). Similarly, another binary variable \( x_{ij} \) represents link coverage between site \( i \) and site \( j \) (a value of 1 if a measurement can be produced between site \( i \) and site \( j \)). If sensor coverage is provided on link \((i, j)\), then sensors must be located at site \( i \) and site \( j \). Likewise, if either cell lacks sensor coverage, then the link cannot be covered. However, the presence of sensors at site \( i \) and site \( j \) do not guarantee that link \((i, j)\) is covered due to the neighboring sensor assumption, which will be discussed next.

2.1. Neighboring sensor assumption

When using point sensors to measure freeway performance, it is not uncommon that neighboring sensors are paired together to evaluate the performance measure. For example, along a given segment, performance can be assessed by averaging the conditions measured by the sensors at either ends and assuming this averaged condition can be uniformly spread over the segment. The performances measured in each segment can then be summed for all segments along the corridor, producing an overall corridor-wide measure of performance.

Given this performance measurement method, a Neighboring Sensor assumption is applied to the probabilistic sensor location model discussed in this paper. This assumption states that performance may only be measured between neighboring sensors (no other sensors in between), in contrast to measuring performance between any two sensors. Without this assumption, the model may allocate sensor resources in a manner to measure performance between any two sensors, which would likely be worse given the manner in which performance is traditionally assessed in practice.

2.2. Measurement error

The goal of the sensor location model is to minimize performance measurement errors. For any two sites, there exists a measurement error, \( E_{ij} \), between the ground-truth measurement of a certain performance measure and the observed measurement of that performance measure from the point sensors for a certain analysis period. In this paper, we choose the analysis time frame as the peak period in a typical weekday with prevailing traffic conditions. This error occurs because point sensors, which capture spot conditions at their deployment site, cannot fully capture the real traffic conditions between two points, which may vary. For this study, travel time is used as the performance measure. The error factor is the Mean Square Error (MSE) between the averaged ground truth travel time \((TT^{G}_{ij})\) and the averaged measured travel time \((TT^{M}_{ij})\) along a segment linking any site \( i \) and site \( j \) during time interval \( t \). Note that the MSE is just one alternative for estimation error evaluation. In practice, a planning group could select a different form of error to determine variability at a particular site. For this paper, we will use MSE of travel time.

Ground truth travel time, defined as the distance between two sites divided by the harmonic average of all passing vehicles’ travel time, requires knowing either all vehicles’ velocity profiles or their actual travel times between these sites. However, it is usually not likely to be observed or measured from field data. One alternative is to use microsimulation. As part of a planning process, a corridor would be modeled in simulation using traffic data collected as part of a study. The calibrated model would provide ground-truth measurement of the performance measure in question between any two sites identified in the simulation. In this paper, we assume that ground truth travel times are known for all the sites along this corridor as a prior. It is also assumed that average time-mean speeds are known for all the sites along this corridor. Measured travel time between site \( i \) and site \( j \) is found by distributing the average of the time-mean speed at site \( i \) and site \( j \) over the segment. In practice, the segment between two adjacent point sensors is estimated to have an average condition reported by both sensors (i.e., when one sensor reports 45 mph and the other reports 55 mph, then average speed is 50 mph, which is the arithmetic average of 45 and 55 mph). Although this is a bit rudimentary, it is widely seen by practicing agencies as providing a “good enough” estimate of conditions, although the accuracy may become less reliable as the spacing of sensors increases. Therefore, it is best to deploy sensors in a manner that most comprehensively measures corridor conditions accurately.

Since the corridor is one-directional, errors can only be assessed in the traveling direction, where the stationing location of site \( i \) \((S_j)\) precedes the stationing location of site \( j \) \((S_j)\). So, for any \( i < j \), \( S_i < S_j \). These differences in travel time are found for all \( t \) time intervals and then summed over the entire analysis period, \( T \):

\[
E_{ij} = \left\{ \begin{array}{ll}
0 & \text{if } S_j \leq S_i \\
\sum_{t=1}^{T} \left( TT^{C}_{ij} - TT^{M}_{ij} \right)^2 & \text{otherwise}
\end{array} \right.
\]

Note. This error factor must also be computed for freeway sections that exist beyond the allocated sensor infrastructure. For example, a section of freeway exists between the corridor’s starting point (denoted as site \( i' \)) and the first allocated sensor, just as a section of freeway may exist between the last allocated sensor and the corridor’s ending point (denoted as site \( j' \)). For these sections, the velocity can be measured at the sensor and distributed over the link or weighted between the measured conditions at the sensor and a free-flow condition, depending on user preference. For a section that runs exclusively from site \( i' \) to site \( j' \) (meaning no sensors are deployed on the corridor), the measured travel time is inferred as a user-defined intuitive guess, such as assuming a steady free-flow traffic condition, given that no data can be collected to prove otherwise. In this paper, the corridor’s starting and ending points will assume free-flow travel speeds.
2.3. Mixed-integer linear model for no-failure configurations

The mixed-integer linear model formulation developed by Danczyk and Liu (2010) serves as a mathematical model that optimizes a proposed sensor deployment within a certain budget by seeking out a configuration with lowest error (or best accuracy) of a measured performance metric, it is shown below in Formulation (1):

Minimize: \[ \sum_{i=0}^{f} \sum_{j=i+1}^{f} E_{ij} x_{ij} \]  
Subject To: \[ \sum_{j=i+1}^{f} x_{ij} - y_i = 0 \quad i \in 1, \ldots, N \]  
\[ \sum_{i=0}^{f} x_{ij} = \sum_{k=j}^{f} x_{jk} \quad \forall j \in 1, \ldots, N \]  
\[ \sum_{j=i}^{f} x_{ij} = 1 \]  
\[ \sum_{i=0}^{f} y_i = 1 \]  
\[ \sum_{j=f}^{N} y_j \leq R \]  
\[ \sum_{j=i}^{N} C_{ij} y_j \leq B \]  
\[ x_{ij} = 0 \quad \forall i \geq j \]  
\[ x \text{ binary} \]  

The objective function (1a) seeks to minimize the overall error that is present with a given configuration. The resulting link-based sensor coverage for this configuration can be best described as a chain of allocated sensors, starting at the corridor’s start point (site \( i_0 \)) and ending the corridor’s end point (site \( j_0 \)) with each allocated sensor serving as a connection between two adjacent links. Eq. (1b) ensures that link coverage is only present between sites where sensor site coverage is also present. Eq. (1c) ensures that link coverage is equally available upstream or downstream of any given site \( j \). Eq. (1d) guarantees that one link starts at the corridor’s start point (site \( i_0 \)) while Eq. (1e) guarantees that one link terminates at the corridor’s end point (site \( j_0 \)). Eqs. (1f) and (1g) serve as budgetary restrictions, where Eq. (1f) limits the number of deployed sensors to a maximum number of allotted sensors, \( R \), and Eq. (1g) limits the total deployments of sensors at any site \( j \) with a predetermined cost \( C_j \) from exceeding a certain total budget, \( B \). Eq. (1h) allows for only links traveling in a downstream direction to receive coverage by setting all other possible links to a value of zero. Lastly, Eq. (1i) upholds binary variable restrictions. This objective function and these constraints are explained in more detail in Danczyk and Liu (2010).

2.4. Graph representation of the mixed-integer linear program

Danczyk and Liu (2010) demonstrate that the mixed integer linear program (MILP) can be represented graphically. In Fig. 2, a typical freeway corridor is illustrated and a corresponding graph \( G(N, A) \) is shown, where \( N \) is the number of possible sensor sites plus one phantom starting site and one phantom ending site, and \( A \) represents the eligible arcs, or links. Here all possible sensor sites are nodes, and the graph is fully-connected with directional links (along with the traffic direction). Arc costs in the graph represent performance measurement errors between two sensors. All enumerable configurations, or paths in the graphical representation, have the same start point at site \( i_0 \) and the same end point at site \( j_0 \). Since this problem is on a one-directional corridor, cycles cannot occur. The mixed-integer linear problem can then be solved by searching the shortest path between the phantom origin site and the phantom destination site. However, because constraints (1f) and (1g), this problem is actually a resource constrained shortest path problem (RCSP).

The MILP formulation along with its solution algorithm produces an answer to the sensor location problem that attempts to minimize the measurement errors. However, it assumes that the deployed sensors are fully operational and will never fail. This assumption needs to be relaxed for real-world applications, as sensor failures are not uncommon and pose dire consequences to the “optimality” of the proposed configuration. To accommodate this deficiency, the model is revised in the next section to account for the failing sensor dilemma.
3. Failure-based sensor location problem

When imagining sensor failures for a given set of deployed sensors, a number of possible combinations exist including both failure and non-failure sensors. Each combination of failures and non-failures for a given sensor configuration is defined as a failure scenario. The total number of scenarios is dependent on the number of sensors deployed. For example, if $M$ sensors are deployed along a freeway corridor, then a total of $2^M$ failure scenarios would exist. For a given failure scenario, the unique combination of operational (non-failure) sensors would produce a corridorwide measurement error unique to that scenario, most likely an error that is different from what could be captured if all sensors that were part of that overall deployment were fully functional. Additionally, each failure scenario would have a certain likelihood of occurrence. Given these, the product of the likelihood of occurrence and the incurred error for a given scenario would be summed together among all possible combinations of operational and non-operational sensor states to produce an error expectation that is to be minimized.

For this work, it is assumed that a sensor failure is something that can be identified and no data is produced from this sensor during failure. This is in contrast to sensor failures where data is still produced, but that data is incorrect or inaccurate. Because of the neighboring sensor assumption, when a sensor fails, the two neighboring operational sensors around the failed sensor will be paired together to produce measurements. Additionally, this research assumes that sensor failure probabilities can be estimated prior to deployment, which is not unreasonable given that past experiences can reveal the functionality of a given sensor type or deployment location. Furthermore, a sensor failure is deemed to occur at all installments at a given site. For example, if a site has three lanes and an individual sensor is installed in each lane, it is assumed that sensor failure occurs on all three lanes and no data can be produced.

The goal of the proposed sensor location model would be to determine a sensor configuration by minimizing the error expectation given budgetary constraints and predetermined sensor failure probabilities. We want to highlight that sensor failure is stochastic in nature. In stochastic programming, we can either choose to minimize one point estimate (such as average estimation errors assuming the risk-neutral attitude or estimation error under the worst/best scenario with the risk-averse/risk-prone attitude) or to consider representative scenarios and characterize parameters as random variables. Accordingly, stochastic optimization is solved either with “expected-value” or scenario-based deterministic equivalent (Zhou and List, 2010). In this paper, because the proposed model aims to serve as a decision-support tool for long-term planning situations, we will formulate the probabilistic sensor location problem as its deterministic equivalent by minimizing expected estimation errors and recommending the truly optimal one after considering all potential alternative configurations and sensor failures. This objective is similar to what was proposed in Li and Ouyang (2011) where a reliable sensor deployment problem for network traffic surveillance maximizes the expected flow and path coverage benefits across all failure scenarios. However, readers should always keep in mind that the optimal sensor location solution obtained from the proposal model may not be optimal for a particular sensor failure scenario.

To obtain the error expectation, all failure scenarios have to be explored and the occurrence likelihood and measurement error of each failure scenario have to be evaluated. The key here is that all possible failure scenarios have to be uniquely identified and be included in the problem formulation once and only once. Construction of constraints to ensure the scenario uniqueness and exhaustiveness, it turns out, is not a trivial task. The difficulty of formulating such constraints comes from the following two aspects. First, for a given sensor configuration, it must be guaranteed that each failure scenario contains a set of operational and non-operational sensors that is uniquely different from all other failure scenarios. The absence of this
uniqueness criterion would potentially result in over-counting certain low-error scenario configurations and ignoring certain high-error scenario configurations. Second, given the budgetary constraints on the total number of deployed sensors, it must be guaranteed that the correct number of failure scenarios is explored so that the objective function accounts for all possible sensor failures. This is also non-trivial because the optimal number of deployed sensors (which may be different from the available budget) and the associated number of failure scenarios is unknown beforehand. The absence of this guarantee would allow the model to freely focus on only a fraction of applicable failure scenario configurations and produce a suboptimal solution.

In the following, we will start the model formulation by describing the constraints, followed by the description of the objective function. As it will become clear in the next section, three types of constraints will be discussed, including definitional constraints, budgetary constraints, and scenario completeness and uniqueness constraints.

### 3.1. Constraints formulation

In the original mixed-integer linear model described in Section 2, two primary variables, \( x_{ij} \) and \( y_i \), were defined to reflect the sensor configuration with fully functional sensors. For the sensor location model with probabilistic failure proposed in this paper, two additional variables will be defined to represent the resultant sensor configuration once sensor failures are included in a given failure scenario. The first variable, \( w_s \), represents the presence of an operational sensor at any site \( i \) for any scenario \( s \). This variable bears a close resemblance to the original \( y_i \) and has a relationship defined in Eq. (2a). The relationship states that if a sensor is deployed at site \( i \) (\( y_i = 1 \)), then that sensor has the possibility of failing (\( w_s = 1 \)) or not failing (\( w_s = 0 \)), depending on the scenario, \( s \). However, if no sensor is deployed at site \( i \) (\( y_i = 0 \)), then that site cannot be operational (\( w_s = 0 \)) for any failure scenario. This is true for all possible failure scenarios, given that the maximum number of scenarios is \( 2^R \) given the upper budgetary restriction of \( R \) sensors.

\[
w_s \leq y_i \quad \forall i \in \{1, \ldots, N\}, \quad s \in \{1, \ldots, 2^R\} \tag{2a}
\]

Therefore, when a sensor is deployed to a site (\( y_i = 1 \)) and failures are considered, only two states exist: an operational state (\( w_s = 1 \)) and a non-operational state (\( w_s = 0 \)). In other words, any deployed sensor fails in half of the scenarios because all possible combinations of sensor failures need to be considered. The exception, of course, is at the site where no sensor was deployed, thus resulting in a total of zero operational states. This is described in Eq. (2b).

\[
\sum_{s=1}^{2^R} w_s = 2 \sum_{i=1}^{N} y_i - 1 \times y_i \quad \forall i \in \{1, \ldots, N\} \tag{2b}
\]

Eq. (2b) guarantees that sensor failures are assessed in a manner that prevents too many or too few operational states from being assigned to any deployed sensor. For example, if 3 sensors are deployed and site \( i \) contains a deployed sensor, then the total number of operational states at that site must equal a value of 4. Similarly, if 3 sensors are deployed and site \( i \) does not contain a deployed sensor, then the total number of operational states at that site must be equal to zero. Without this constraint, the model would be allowed to freely assign as many operational states to a given sensor as desired and consequently produce failure scenarios that do not actually include failed sensors.

The second variable, \( V_{ij} \), represents link coverage between site \( i \) and site \( j \) for any scenario \( s \), where sensors are deployed at site \( i \) and site \( j \) to provide the coverage. This variable is closely related to the original \( x_{ij} \). The relationship between link coverage and site coverage is shown in Eq. (2c).

\[
w_s V_{ij} \geq V_{ij} \quad \forall i \in \{1, \ldots, N\}, \quad \forall j \in \{1, \ldots, N\}, \quad \forall s \in \{1, \ldots, 2^R\} \tag{2c}
\]

Effectively, these variables and their relationships are forming a structure that is identical to that of the mixed-integer linear model from Formulation (1). Using the link-based concepts employed in Eqs. (1c), (1d), and (1e), similar constraints can be formulated for a given failure scenario. These constraints are shown in Eqs. (2d), (2e), and (2f) respectively to provide link-based coverage across the entire corridor and uphold the Neighboring Sensor Assumption. In other words, these equations constrain link-based sensor coverage so that a chain of sensors forms from the corridor’s start point to the corridor’s end point, with each allocated sensor as a connection between two adjacent links. The only difference between these equations and the one in Formulation (1) is that a sensor chain is designed for each scenario, which includes both operational and non-operational sensors, instead of solely for the no-failure configuration.

\[
\sum_{i=1}^{j-1} V_{ij} = \sum_{k=j+1}^{j} V_{ik} \quad \forall j \in \{1, \ldots, N\}, \quad \forall s \in \{1, \ldots, 2^R\} \tag{2d}
\]

\[
\sum_{j=1}^{f} V_{ij} = 1 \quad \forall s \in \{1, \ldots, 2^R\} \tag{2e}
\]

\[
\sum_{i=1}^{N} V_{is} = 1 \quad \forall s \in \{1, \ldots, 2^R\} \tag{2f}
\]
Similarly, the link-to-site coverage relationship defined in Eq. (1b) and the one-directional corridor restriction defined in Eq. (1h) can be restructured using the new variables. These relationships are reflected in Eqs. (2g) and (2h) for a given scenario.

\[
\sum_{j=1}^{i} V_{ij} = w_i = 0 \quad \forall i \in \{1, \ldots, N\}, \quad s \in \{1, \ldots, 2^M\} \tag{2g}
\]

\[
V_{ij} = 0 \quad \forall \{i \geq j\}, \quad \forall s \in \{1, \ldots, 2^M\} \tag{2h}
\]

Lastly, any configuration with operational or non-operational sensors must maintain the necessary budgetary restrictions defined in the original problem. Since any sensor deployment, whether operational or non-operational, incurs a cost, the budgetary restrictions would establish a limit on all deployments, much like in the problem defined in Formulation (1). Consequently, the budgetary restrictions from the mixed-integer linear model are applied here and defined in Eqs. (2i) and (2j).

\[
\sum_{i=1}^{N} y_i \leq R \tag{2i}
\]

\[
\sum_{i=1}^{N} C_i y_i \leq B \tag{2j}
\]

### 3.1.1. Constraints for scenario completeness and uniqueness

Now we will discuss the constraints to ensure that all possible scenarios are explored and each scenario contains a different set of operational and non-operational sensors so that it is uniquely identified. To describe this mathematically, the sensor deployment along a corridor with \( N \) possible sites is imagined as an \( N \)-bit binary string, where 1 and 0 corresponds with an operational state and non-operational (or no deployed sensor) state, respectively. Clearly, if the scenarios contain different combinations of operational and non-operational sensors, the binary numeric value would be different.

The difficulty here is that the optimal number of sensors is unknown beforehand. We know that the maximum number of failure scenarios equals \( 2^M \), which is a constant value. However, if the optimal number of sensors is actually a value of \( M \), where \( M < R \), then only \( 2^M \) scenarios need to be examined. Unfortunately, \( M \) is a variable and cannot be used to estimate the true number of scenarios, as the number of explored failure scenarios needs to be a constant in the objective function.

To overcome this dilemma and still guarantee a constraint to ensure the failure scenarios in question are unique, we propose the following conceptual procedure from which to build a constraint set:

1. Sequence the failure scenarios from the greatest binary numeric value to the least binary numeric value.
2. Determine uniqueness for the first \( 2^M \) scenarios (where \( M \) is the deployed number of sensors) by identifying numerical differences between the binary numeric values of these scenarios. These scenarios account for all combinations of failure and are defined here as applicable scenarios.
3. Ignore all scenarios that are beyond the first \( 2^M \) scenarios. These scenarios are redundant (they are only created to satisfy the mathematical expressions given an unknown value for \( M \)) and are defined here as non-applicable scenarios.

Satisfying the first point is the easiest, as a constraint can be formulated to arrange the scenarios from greatest to least by simply establishing that the \( n \)-th scenario’s binary numeric value is less than the \( (n - 1) \)-th binary numeric value. However, the second and third points make this procedure difficult, as formulating constraints to ignore some scenarios and not others is not easy. To overcome this, a new binary variable, \( g_s \), is proposed. This variable, which in this work can be called the ‘gap variable’ and is assigned to each scenario, is intended to help establish which scenarios to compare and contrast, given that the scenarios have been ordered sequentially. For example, if 4 sensors were deployed \( (M = 4) \) and all scenarios were organized from greatest binary numeric value to least binary numeric value, this gap variable would identify a gap in binary numeric value between the 1st and 2nd scenario, the 2nd and 3rd scenario, and all the way until the \( (2^4 - 1) \)-th and \( 2^4 \)-th scenario. For any scenarios that may exist beyond this, no gap would be made available, as there is no desire to seek differences between non-applicable scenarios. Additionally, this gap variable would aid in identifying applicable scenarios and non-applicable scenarios.

To make this gap variable serve effectively, the first step is to ensure that the desired number of gaps occurs. This desired number would depend on the number of \( M \) sensors deployed to the network. In the example from the previous paragraph, the first gap would appear at the second scenario \( (g_2 = 1) \), as this gap would compare the binary numeric value of the second scenario against the first scenario. Similarly, the next gap would appear at the third scenario \( (g_3 = 1) \), as this gap would compare the binary numeric value of the third scenario against the second scenario. These gaps would continue in all scenarios up to the \( 2^M \)-th scenario, where, in the previous example, the last gap would appear in the \( 2^4 \)-th scenario (16th scenario, where \( g_{16} = 1 \)). For any scenarios that may exist beyond that, the gap would be nonexistent (e.g. \( g_{17} = 0, g_{18} = 0 \), etc.), as to avoid contrasting non-applicable scenarios. Although the first scenario \( (s = 1) \) technically does not have a 0th scenario
with which to contrast binary numeric values, it will receive a gap as well ($g_s = 1$). Therefore each applicable scenario will be assigned a value of 1 for the gap variable, all non-applicable scenarios will have a gap value of 0. From this, it is clear that the number of gaps for $M$ sensors would be $2^M$ ($2^M - 1$ actual gaps for which to contrast applicable scenarios and then 1 additional gap for the first scenario). Thus, the total number of gaps for all scenarios can be formulated as shown in Eq. (2k). The scenarios that are investigated range from the 1st to the $2^R$-th.

$$
\sum_{s=1}^{2^R} g_s = 2^{i-1}
$$

(2k)

We also need to ensure that gaps have been assigned to only the applicable scenarios. As is, Eq. (2k) is free to assign gaps to scenarios that are conceptually non-applicable, such as the $2^R$-th scenario when $M < R$. To avoid this dilemma, Eq. (2l) is formulated to ensure that gaps are only given between applicable scenarios.

$$
g_s \times s \leq 2^{i-1} \quad \forall s
$$

(2l)

For example, if the 3rd scenario receives a gap ($g_3 = 1$), then the left side of the equation produces a value of 3, since ($s = 3$). If four sensors are deployed ($M = 4$), then the right side of the equation becomes a value of 16 and the inequality holds. However, if the 20th scenario receives a gap ($g_{20} = 1$, $s = 20$), then the left side of the equation becomes a value of 20 and the inequality is false. This prohibits gaps from being applied to scenarios beyond the $2^M$-th scenario. And, when coupled with Eq. (2k), it is guaranteed that gaps will be applied to the right scenarios.

Now, a constraint can be formulated to guarantee uniqueness to scenarios. Eq. (2m) establishes that all applicable failure scenarios are ordered sequentially from greatest to least binary numeric value. This sequential ordering is verified by the gap variable to ensure that a difference of at least a value of one (1) exists between the ordered scenarios. When non-applicable scenarios are found, the gap variable is set to a value of zero so that these scenarios are not contrasted with the applicable scenarios. Since the first scenario technically does not provide a gap, only the 2nd through the $2^R$-th scenario are explored.

$$
\sum_{i=1}^{N} \left[ w_{i} \times 2^{N-i} \right] - \sum_{i=1}^{N} \left[ w_{gs(i-1)} \times 2^{N-i} \right] \leq -g_s \quad \forall s \in \{2, \ldots, 2^R\}
$$

(2m)

Given the intricacies of the constraints in these sections, an example is shown to reveal the end result that meets the requirements dictated thus far, shown in Table 1. In this example, three sites are available for sensor deployment ($N = 3$), three sensors are available in the budget ($R = 3$), and two deployed sensors results in the best configuration ($M = 2$). Although 8 failure scenarios are produced, only 4 failure scenarios are applicable to this example. Thus, the scenarios are ordered sequentially based on the binary numeric value and gap values are assigned to applicable scenarios.

Constraints (2k), (2l) and (2m) are essential to this model, as they guarantee the uniqueness of the operational and non-operational sensor configuration for a given failure scenario when compared with all other failure scenarios. Without Eqs. (2k) and (2l), the model would have no means to determine whether scenarios are different, nor would it be able to determine which scenarios should be different (e.g. there is no need to contrast Failure Scenario 5–8 in Table 1, thus the gap variable is a value of 0). Without Eq. (2m), the model would not be able to inspect the differences in the resulting failure scenario configurations through the binary numeric values, potentially allowing it to erroneously assign operational sensors within failure scenarios in a mathematically favorable, but unrealistic, manner.

### 3.1.2. Computation of scenario likelihood

The occurrence likelihood of any given failure scenario, $s$, depends on the sensor failure probabilities. To find this likelihood, the operational probabilities of the sites along the corridor need to be determined. It is first assumed that a given

<table>
<thead>
<tr>
<th>Sensor allocation</th>
<th>Site 1</th>
<th>Site 2</th>
<th>Site 3</th>
<th>Binary numeric value</th>
<th>Numeric value difference</th>
<th>Gap variable g(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Failure Scenario 1</td>
<td>$y(i)$</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>NA</td>
<td>1</td>
</tr>
<tr>
<td>Failure Scenario 2</td>
<td>$w(i,1)$</td>
<td>1</td>
<td>0</td>
<td>5</td>
<td>-1</td>
<td>1</td>
</tr>
<tr>
<td>Failure Scenario 3</td>
<td>$w(i,2)$</td>
<td>1</td>
<td>0</td>
<td>4</td>
<td>-1</td>
<td>1</td>
</tr>
<tr>
<td>Failure Scenario 4</td>
<td>$w(i,3)$</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>-3</td>
<td>1</td>
</tr>
<tr>
<td>Failure Scenario 5</td>
<td>$w(i,4)$</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-1</td>
<td>1</td>
</tr>
<tr>
<td>Failure Scenario 6</td>
<td>$w(i,5)$</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Failure Scenario 7</td>
<td>$w(i,6)$</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Failure Scenario 8</td>
<td>$w(i,7)$</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Failure Scenario 9</td>
<td>$w(i,8)$</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Sum for failure scenarios</td>
<td>2</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>4</td>
</tr>
</tbody>
</table>

Bold values indicates no-failure scenario.
sensor type, $d$, has a certain operational probability, $P_d$, that is known. This probability would be likely attributed to the design and durability of the sensor type in question. Secondly, it is assumed that any given site $i$ has a certain probability of successful data collection, $P_{site(i)}$, which is also known. This probability would be likely attributed to elements of the site, such as environmental harshness (e.g. rain, snow, etc.), vulnerability of the data relay device (e.g. length of transmission wires, interference for wireless transmission, signal loss, etc.), and stability of existing infrastructure (e.g. crumbling road infrastructure where loop detectors are imbedded, shaky pole for camera, etc.). With this knowledge, the probability of acquiring successful data from a given site $i$ with a given sensor type, $Pr_{ij}$, can be computed, as shown in Eq. (2o). Since these probabilities are assumed to be constant, the probability of acquiring successful data at a given site can be considered constant.

$$Pr_{ij} = P_d + P_{site(i)}$$  \hspace{1cm} (2n)

With the probabilities of successful operation at a given site $i$, $Pr_{ij}$, and the presence of operational sensors at that given site $i$ during scenario $s$, $w_i$, both known, the likelihood of a given scenario occurring, $Ps$, can be computed. This likelihood is found by taking the product of the successful operational probabilities, $Pr_i$, for all sensors that remain operational in a given scenario and the failure probabilities, $(1 - Pr_i)$, for all sensors that have failed in the given scenario. Any site $i$ in which no sensors have been deployed ($y_i = 0, w_i = 0$) automatically defaults to a failure probability of 1. Eq. (2o) illustrates the calculation of this likelihood.

$$Ps = \prod_{i=1}^{N} [(w_i \times Pr_i) + ((1 - w_i) \times (1 - (Pr_i \times y_i)))] \hspace{1cm} \forall s$$  \hspace{1cm} (2o)

To better illustrate the calculation of scenario likelihood, an example is shown in Fig. 3. In this example, two sensors have been deployed on a freeway corridor, one at Site 1 and the other at Site 3. Each site has a respective operational probability, determined based on known uptime requirements for these sensors at each particular site. As a result, four failure scenarios exist with their own configuration and likelihood of occurrence. The values for $V_{ij}, w_i$, and $Pr$ (shown as $V(i,j,s), w(i,s), Pr (i)$, and $Ps$ in the figure) are illustrated for their respective scenarios to better clarify the purpose they serve.

As shown in Fig. 3, the proposed sensor configuration deploys 2 sensors. This creates 4 scenarios in which the sensors may operate when sensor failure is considered. Scenario 1 is an instance where both sensors are still fully operational; this scenario is expected to occur 35% of the time, given the operational probability of 0.7 for Site 1 and 0.5 for Site 3. Scenario 2 is an instance where Site 1 remains operational, but Site 3 has failed; this scenario is expected to occur 35% of the time as well, given the operational probability of 0.7 for Site 1 and the non-operational probability of $(1 - 0.5 = 0.5)$ for Site 3. Scenario 3 is an instance where Site 1 has failed, but Site 3 remains operational; this scenario is expected to occur 15% of the time, given the non-operational probability of $(1 - 0.7 = 0.3)$ for Site 1 and the operational probability of 0.5 for Site 3. Lastly, Scenario 4 is an instance where both Site 1 and Site 3 have failed; this scenario is expected to occur 15% of the time as well, given the non-operational probability of $(1 - 0.7 = 0.3)$ for Site 1 and $(1 - 0.5 = 0.5)$ for Site 3, respectively. These four scenarios—which sum to 100%—represent all possible combinations of operational and non-operational sensors for this proposed sensor configuration.

If looking at the combinations of operational and non-operational sensors in each scenario, we can see that each scenario has its own configuration, which is a subset of the deployed sensors in the overall no-failure configuration. Each subset configuration would have its own unique corridorwide error. This will become important later, as the expected increases in error of these subset configurations may alter the expected performance of a certain sensor configuration. For example, the proposed sensor configuration in Fig. 3 may perform extremely well (very little error between ground-truth and measured conditions) when no sensor failures are considered, but examining the failure scenarios (i.e. Scenario 2) may reveal that those subset configurations yield extremely poor performance (very high error between ground-truth and measured conditions). When factoring in these poor-performance scenarios, relative to their expected rate of occurrence, it may be found that the proposed sensor configuration may not truly be the most optimal one when failures are considered.

3.2. Objective function formulation

The goal of the sensor location problem is to minimize the average error incurred given a variety of sensor failure scenarios. The proposed constraints so far have allowed for the establishment of these failure scenarios. Individual error per scenario can easily be found by utilizing the scenario link-coverage variable, $V_{ij}$, and multiplying by the error factor, $E_{ij}$. This is similar to the expression found in Eq. (1a), except using the scenario-based link-coverage variable, $V_{ij}$, instead of the static no-failure link-coverage variable, $x_{ij}$. The impact of the error from this scenario is dependent on the likelihood of occurrence, $Ps$. The product of likelihood of occurrence, $Ps$, and the incurred error for that scenario, $s$, results in the impact of the error for that scenario. Summing the impacts of the error for all applicable failure scenarios produces the average incurred error that is sought in this problem.

This conceptual formulation immediately produces a dilemma, as the number of scenarios is not known at the beginning of the problem. The only guarantee is that, at most, $2^M$ scenarios exist. However, if in fact there are only $2^M$ applicable failure scenarios due to $M$ deployed sensors, where $M < R$, then the objective function will explore the $(2^R - 2^M)$ non-applicable sce-
narios as well, where these non-applicable scenarios mathematically represent a full sensor failure with a resulting error. Exploring these non-applicable scenarios would repetitively count the full failure scenario and consequently bias the overall average error.

To limit the objective function to focus solely on applicable scenarios, the gap variable, \( g_s \), is used. Recall that this binary variable only achieves a non-zero value (i.e. \( g_s = 1 \)) for any applicable failure scenario. All other non-applicable scenarios do not receive a non-zero value (i.e. \( g_s = 0 \)) for this gap variable, as they do not require to be contrasted with one another. Multiplying \( g_s \) into the objective function for a given scenario allows for error to only be assessed from \( 2^M \) applicable failure scenarios (with \( M \) optimally deployed sensors). All non-applicable scenarios are ignored, since \( g_s \) would have a value of zero for these scenarios. This effectively prevents the exploration of non-applicable scenarios.

Eq. (2p) is the formulation of this objective function, which seeks to minimize the average incurred error from a given sensor configuration by analyzing all possible failure scenarios. It is influenced by the operational and non-operational states of sensors for a particular failure scenario, the weighted error incurred from that failure scenario, and the restriction that prevents examination of non-applicable failure scenarios.
3.3. Probabilistic sensor location model considering sensor failure

In summary, the complete formulation of the sensor location model with failing sensors is outlined in Formulation (3).

\[
\begin{align*}
\text{Min :} & \quad \sum_{i=1}^{R} \sum_{j=1}^{S} P_i \times E_{ij} \times V_{ij} \times g_s \\
\text{s.t.} & \quad w_{ni} \leq y_i \quad \forall i \in \{1, \ldots, N\}, \quad \forall s \in \{1, \ldots, 2^R\} \quad (3b) \\
& \quad \sum_{i=1}^{R} w_{ni} = 2 \sum_{i=1}^{N} y_i \times y_i \quad \forall i \in \{1, \ldots, N\} \quad (3c) \\
& \quad \sum_{i=1}^{R} V_{ij} = \sum_{k=1}^{S} V_{jks} \quad \forall j \in \{1, \ldots, N\}, \forall s \in \{1, \ldots, 2^R\} \quad (3d) \\
& \quad \sum_{j=1}^{S} V_{jks} = 1 \quad \forall s \in \{1, \ldots, 2^R\} \quad (3f) \\
& \quad \sum_{j=1}^{S} V_{jks} = 1 \quad \forall s \in \{1, \ldots, 2^R\} \quad (3g) \\
& \quad \sum_{i=1}^{R} y_i \leq R \quad (3i) \\
& \quad \sum_{i=1}^{R} C_j y_i \leq B \quad (3j) \\
& \quad \sum_{i=1}^{R} g_s = 2 \sum_{i=1}^{N} y_i \quad (3k) \\
& \quad g_s \times s \leq 2 \sum_{i=1}^{N} y_i \quad \forall s \in \{1, \ldots, 2^R\} \quad (3l) \\
& \quad \sum_{i=1}^{R} [w_{ni} \times 2^{N-i}] - \sum_{i=1}^{R} [w_{ni-1} \times 2^{N-i}] \leq -g_s \quad \forall s \in \{2, \ldots, 2^R\} \quad (3m) \\
& \quad P_S = \prod_{i=1}^{N} [(w_{ni} \times Pr_i) + ((1 - w_{ni}) \times (1 - (Pr_i \times y_i)))] \quad \forall s \in \{1, \ldots, 2^R\} \quad (3n) \\
& \quad y, w, V, g \text{ binary} \quad (3o)
\end{align*}
\]

Eq. (3a) minimizes the average error incurred by a sensor configuration considering all possible failure scenarios. Eq. (3b) establishes a relationship between no-failure sensor coverage and sensor coverage for a given failure scenario. Eq. (3c) restricts the number of scenarios where a given sensor is operational to exactly one-half of the total number of scenarios. Eqs. (3d)–(3h) establish the chain of sensor coverage over the entire corridor for a given failure scenario. Eqs. (3i) and (3j) serve as budgetary constraints for sensor deployment and are similar to the ones in the mixed-integer linear model discussed earlier. Eq. (3k) defines the total number of gaps, \(g_s\), that will be used for the deployed configuration and Eq. (3l) allocates these gaps to the applicable scenarios only. Eq. (3m) uses this gap variable to ascertain differences between the binary numeric values of failure scenarios to ensure scenario uniqueness. Eq. (3n) determines the likelihood of a given scenario occurring and is dependent on the deployed sensor configuration. Lastly, Eq. (3o) sets the binary restrictions for the variables used in this problem.
4. Solution algorithm

Formulation (3) is a mixed-integer nonlinear program (MINLP). Given the complexity of the program in Formulation (3), traditional solvers struggle to find the optimal solution. In this paper, we propose a heuristic solution algorithm to solve the probabilistic sensor location model. The heuristic solution algorithm is based on the graph representation of the MILP discussed in Section 2.4. Realizing that, without the consideration of sensor failure, the probabilistic model discussed in the last section (Formulation (3)) will collapse into the deterministic mixed-integer program (Formulation (1)), our heuristic algorithm evaluates the first k shortest paths (i.e. the first k optimal sensor configurations) using the graph representation of the no-failure optimization model and chooses the optimal no-failure one with the least error expectation. In this way, we avoid some of the complexity regarding scenario completeness and uniqueness because the failure scenarios and their associated occurrence likelihoods are well-defined given a predetermined sensor configuration. Therefore, before we discuss our heuristic solution algorithm, we will give a brief overview on the graph representation of the mixed-integer linear program discussed in Danczyk and Liu (2010).

4.1. Graph representation of the failure-based scenarios

For the probabilistic model proposed in this paper, the graphical representation introduced in Section 2.4 for non-failure MILP is also applicable. For any particular sensor deployment, a series of paths can be easily generated to represent the configurations for any failure scenario. These failure scenario paths would only use a subset of nodes that are included in the set of nodes used in the no-failure configuration. Using this concept, a heuristic solution algorithm introduced in Danczyk and Liu (2010) can be devised to explore a series of shortest no-failure paths and their respective no-failure and failure-based costs.

4.2. Heuristic solution algorithm using k-shortest path search

Each no-failure configuration, or path, has a corresponding cost, which can be considered as the incurred error from allocating sensor resources at respective sites. Thus, one could find the k-th most optimal no-failure path among all enumerable possibilities, using the graph representation discussed in the last section. A large body of literature discusses methods for finding the k-th shortest path for a given network (e.g., Pollack, 1961; Yen, 1971; Eppstein, 1999).

4.2.1. Calculation of error

For all possible configurations, the k-th optimal no-failure configuration has a given set of deployed sensors. The total corridorwide error for the k-th optimal no-failure configuration, denoted as $E(k)$, can be found by solving the mixed-integer linear program from Formulation (1). On the graphical network proposed in Fig. 2, this total error could be found by solving the resource constrained shortest path problem for the k-th optimal path. Given this k-th optimal no-failure configuration, the corresponding average corridorwide error resulting from failed sensors, denoted as $E(k)^*$, can be found by applying the set of deployed sensors to the probabilistic model proposed in Formulation (3) and solving it by exploring all possible failure scenario paths using the subset of nodes utilized by the k-th optimal no-failure path and weighting their impact by their likelihood of occurrence. The method used to find $E(k)$ and $E(k)^*$ are left to the preference of the interested party, but this work will assume that both values are easily available.

For this work, an assumption will be made that $E(k)^*$ produces a worse result than $E(k)$. This assumption may seem nonessential, as one would intuitively assume that sensor failures would naturally make a configuration worse, but opposite cases may exist. These cases are rare and generally occur when the 1st optimal configuration uses fewer sensors than the budgeted number, R. For example, 2 sensors may produce the 1st optimal no-failure result, despite an available budget of 3 sensors (R = 3). However, when examining the 2nd optimal no-failure configuration, 3 sensors may be necessary. Depending on where these three sensors are allocated, one of the failure scenarios for this 2nd optimal no-failure configuration may actually be the 1st optimal configuration. If the likelihood of occurrence of such failure scenario is sufficiently high, this could potentially produce an average error for a certain sensor configuration with probabilistic failure that is more accurate than its no-failure counterpart. It is important to note that this assumption applies only to how the error of a no-failure case versus a failure-based case is expected to differ and does not impact the result when actually calculating these errors.

The assumed relationship between the no-failure error and the failure average error for the k-th shortest path can be represented by Eq. (4). This is assumed true for all k-shortest paths.

$$E(k)^* \geq E(k) \quad \forall k$$

Eq. (4) may seem trivial, but is critical to help establish good stopping criteria. Its need often stems from the inherent limitations and assumptions made by practicing agencies when estimating travel times with point sensors. One such example is provided for reference. In the following table, a freeway corridor has a budget for up to 3 sensors. Four sites are eligible to receive sensors, including Site 1, Site 2, Site 3, and Site 4, shown in Table 2.

Following industry practice, travel times between any two points is estimated by averaging the historical speeds. For this example, adjacent sites very accurately estimate the ground-truth travel time, which results in a mean-square error of zero.
Table 2
Example freeway deployment.

<table>
<thead>
<tr>
<th>Site (roadway milepost)</th>
<th>Historical speed</th>
<th>Ground-truth travel time (from previous site to site)</th>
</tr>
</thead>
<tbody>
<tr>
<td>f (m.p. 0.0)</td>
<td>55 MPH (free-flow, assumed)</td>
<td>–</td>
</tr>
<tr>
<td>Site 1 (m.p. 1.0)</td>
<td>35 MPH</td>
<td>80 s</td>
</tr>
<tr>
<td>Site 2 (m.p. 2.0)</td>
<td>55 MPH</td>
<td>80 s</td>
</tr>
<tr>
<td>Site 3 (m.p. 3.0)</td>
<td>35 MPH</td>
<td>80 s</td>
</tr>
<tr>
<td>Site 4 (m.p. 4.0)</td>
<td>55 MPH</td>
<td>80 s</td>
</tr>
<tr>
<td>f’ (m.p. 5.0)</td>
<td>55 MPH (free-flow, assumed)</td>
<td>65.5 s</td>
</tr>
</tbody>
</table>

With the estimate of travel times from the sensors and information on actual ground-truth conditions, the mean-square error can be calculated for all other site pairs, shown in Table 3.

Using Formulation (1), the no-failure configuration is deemed optimal by deploying sensors to Site 1 and Site 4 only, which will be referred to in this example as “Configuration A.” This solution uses 2 sensors instead of the budgeted maximum of 3 sensors. It generates a total error of zero, or $E("Configuration A") = 0$. In review, no combination exists that uses 3 sensors that provides a more accurate estimate of travel time, which is why the two-sensor solution was deemed optimal. Please note that a second optimal solution deploys sensors at Site 3 and Site 4, but “Configuration A” is only used for purposes of demonstration.

When looking at worse solutions that utilize 3 sensors, particularly ones that may be used in the solution algorithm to test optimality, one particular solution is worth noting. This solution—referred to in this example as “Configuration B”—deployed sensors to Site 1, Site 3, and Site 4. As a result, this generates a no-failure error of 2089.79 using the table above, or $E("Configuration B") = 2089.79$. It is unusual to think that an additional sensor at Site 3 would make “Configuration B” worse than “Configuration A,” but this is again due to the point-based data reporting that is conducted with these types of sensors and a consequence of averaging travel conditions between these sensors, which is a common practice in the industry.

To assess the true error when sensor failures are considered, it will be assumed in this example that all sensor sites has a 99% uptime (i.e., a 1% failure rate), which is a common procurement requirement in practice. When analyzing the 8 possible failure scenarios for “Configuration B,” shown in Table 4, it can be quickly noted that the better “Configuration A” is, in fact, one of the scenarios. When factoring in these chances for failure into “Configuration B,” it is immediately noted that the new error is 2050.66, or $E("Configuration B") = 2050.66$. This immediately seems unusual, as a no-failure error of 2089.79 versus a failure-based error of 2050.66 basically states that “Configuration B” performs better when sensors are failing than when all working properly, which is extremely counterintuitive.

In summary, in the above example, the no-failure “Configuration A” is better than the no-failure “Configuration B” and the no-failure “Configuration B” is worse than failure corridor error. This becomes problematic in the solution algorithm, discussed below, because it risks terminating the process before an optimal failure-based solution is truly identified. Therefore, this assumption is necessary.

4.2.2. Stopping criteria

If it is shown that the incurred error captured with failing sensors for the $j$-th optimal no-failure configuration, $E(j)^*$, is less than the no-failure incurred error captured by the $k$-th optimal no-failure configuration $E(k)$ (mathematically described as $E(j)^* < E(k)$), where configuration $j$ is a better configuration for the no-failure case than configuration $k$ (mathematically described as $E(j) < E(k)$), then the $k$-th optimal no-failure configuration cannot be the optimal configuration when accounting for failing sensors. Knowing this reduces the number of configurations to investigate for optimality.

4.2.3. Heuristic algorithm

The heuristic algorithm we propose in this paper begins by finding the incurred corridorwide error captured with failing sensors for the 1st optimal no-failure configuration $[E(1)^*]$. Since this incurred error captured with failing sensors is the only known error at this time, it becomes the best known error and the 1st optimal no-failure configuration becomes the best known configuration. With this best known error, the no-failure error generated by the 2nd optimal no-failure configuration $E(2)$ is examined. If this no-failure error is worse than the best known error $E(1)^*$, then the best known configuration (1st optimal no-failure configuration) is deemed optimal. If that is not the case and a 3rd optimal no-failure configuration exists, no conclusion can be made and the search procedure must repeat itself. This time, the procedure finds the average incurred corridorwide error captured with failing sensors for the 2nd optimal no-failure configuration $[E(2)^*]$. If this incurred error $E(2)^*$ is better than the best known error $E(1)^*$, the incurred error and 2nd optimal no-failure configuration replaces the existing best known error and best known configuration, respectively (2nd optimal no-failure configuration is now the best known). Again, the best known error $E(2)^*$ is contrasted against the no-failure error produced by the 3rd optimal no-failure configuration $[E(3)]$. The procedure repeats until one of the following termination criteria are met:

1. All possible configurations have been examined.
2. A $k$-th optimal no-failure configuration yields a no-failure error $E(k)$ that is worse than the best known error found in previously explored configurations that consider sensor failures.
This search procedure is outlined in Fig. 4. While this procedure cannot guarantee that full enumeration does not occur, it does offer the opportunity to avoid such an extensive search.

Without the employed assumption in Eq. (4), the possibility of prematurely terminating the search procedure at a sub-optimal configuration does exist. This premature termination of the search procedure would only occur when one of the following two criteria is met:

1. A certain \( k \)-th optimal no-failure configuration yields higher errors when fully operational than when failures are considered (i.e. \( E(k) < E(k) \)).

2. A \( k \)-th optimal no-failure configuration that meets criteria 1 exists beyond the \( j \)-th optimal no-failure configuration (i.e. \( k > j \)), where the search procedure terminates at the \( j \)-th optimal no-failure configuration due to the requirements up to that point being met.

It is important to reiterate that such instances should only occur on a very rare basis and thus premature terminations of the search procedure will occur infrequently, if at all. Given this limited likelihood of occurrence, this search procedure and the assumptions it employs can be deemed as sufficiently acceptable for finding the truly optimal solution given sensor failures.

To better illustrate the search procedure, a simple example is provided in Table 5. The no-failure errors and failure-based errors for the \( k \)-shortest paths of a given network are provided. It is assumed that a total of \( K \) shortest paths exist, where \( K \) is sufficiently large so that a path longer than the shortest path almost always exists for exploration. As shown, the 1st optimal no-failure configuration (\( k = 1 \)) does not yield the optimal error once sensor failures are considered, although this would not be known initially. The search procedure examines the 1st optimal no-failure configuration and initially defines it as the best configuration, but quickly makes a correction when examining the 2nd optimal no-failure configuration and finding a better result. To guarantee this result as the best, the search procedure explores the 3rd, 4th, and 5th optimal no-failure configurations for better results (for both no-failure and failure-based errors) until it is shown that the error for the 6th optimal no-failure configuration is worse than the best known error. Thus, the search is terminated at the 6th optimal no-failure configuration.

4.2.4. Computational requirements

Depending on the corridor in question, finding an optimal solution may be time-consuming since, in the worst case, all paths have to be enumerated from the origin and the destination. This is not usually a problem because, unlike optimization problems catered to real-time problems (e.g. ramp metering, adaptive signal timing, etc.), allocating sensor resources is seldom done in the short-term. This problem is generally a one-time planning problem, leaving ample time to search for the best configuration. Still, the number of possible scenarios is not a trivial amount. Steps to reduce computation time could include:
Assume a maximum number of failed sensors of the proposed lot. An example of this would be to say "of my 16 proposed sensors, only 2 at most will fail." This would reduce the number of scenarios by discounting the intuitively rare instances of widespread failure, which is sensible because most transportation agencies with widespread sensor failures would simply cease public broadcast of the performance measures until the repair was made in order to avoid distribution of bad data that harms public trust. The drawback of assuming this is that the heuristic will not fully explore the number of possible scenarios and possibly select a sub-optimal answer.

**Table 5**

A simple example of the proposed search procedure at work.

<table>
<thead>
<tr>
<th>k</th>
<th>$E(k)$</th>
<th>$E(k)^*$</th>
<th>Previous best known error</th>
<th>New best known error</th>
<th>New best known solution</th>
<th>$E(k+1)$</th>
<th>Is $E(k+1)$ &gt; best known error?</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>5</td>
<td>15</td>
<td>Infinity</td>
<td>15</td>
<td>1</td>
<td>7</td>
<td>No, advance to $k = 2$</td>
</tr>
<tr>
<td>2</td>
<td>7</td>
<td>11</td>
<td>15</td>
<td>11</td>
<td>2</td>
<td>8</td>
<td>No, advance to $k = 3$</td>
</tr>
<tr>
<td>3</td>
<td>8</td>
<td>21</td>
<td>11</td>
<td>11</td>
<td>2</td>
<td>9</td>
<td>No, advance to $k = 4$</td>
</tr>
<tr>
<td>4</td>
<td>9</td>
<td>18</td>
<td>11</td>
<td>11</td>
<td>2</td>
<td>10</td>
<td>No, advance to $k = 5$</td>
</tr>
<tr>
<td>5</td>
<td>10</td>
<td>15</td>
<td>11</td>
<td>11</td>
<td>2</td>
<td>12</td>
<td>Yes, best configuration is 2nd optimal no-failure configuration; best error is 11. Terminate search</td>
</tr>
</tbody>
</table>

**Fig. 4.** Flow chart for finding the best configuration with failing sensors.
• Formulate the corridor into smaller subsections with reduced budgets. For example, rather than deploy 20 sensors over a 10-mile corridor, divide the corridor into two 5-mile segments and deploy 10 sensors. This would reduce the number of scenarios by simply reducing the budget. The drawback of pursuing this is that, at a corridor level, a sub-optimal solution may be selected due to the fact that subsections that benefit with more sensors may not get those additional sensors due to the evenly-divided budget.

• Explore only the most probable scenarios until a certain representative threshold is reached. An example of this would be to say “of a given configuration, estimate actual error using the most probable scenarios until 95% of the likelihood is reached.” This reduces the number of scenarios by discounting the rarest scenarios. Similar to other alternatives, the drawback of following this approach is that the heuristic will not fully explore the number of possible scenarios and possibly select a sub-optimal answer.

In reality, practitioners may not be interested in mathematically exploring such an extensive number of configurations. More likely, they would identify a limited number of configurations (e.g. 10 proposed configurations) with which to compare and contrast, given that one of the configurations is optimal without failure consideration and the others are alternatives. Even with a limited number of configurations, the proposed search procedure is still applicable. The k-shortest paths would be determined using only the identified configurations, ordered from most optimal to least optimal for the no-failure case. The search procedure would then run until all the configurations have been explored or it is shown that all remaining no-failure configurations are worse than an examined configuration with failures. From this, the best configuration given known failure rates can be identified among the limited set.

To demonstrate this heuristic in a real-world application, a case study network is proposed where new sensors are intended to be deployed, given known failure characteristics. The results of this case study are discussed in the next section.

5. Case study – Twin Cities I-94 corridor

5.1. Case study network

How does the model allocate sensors on a real network? To answer this question, a case study is conducted on the Interstate 94 (I-94) corridor in Minneapolis-St. Paul, Minnesota, as shown in Fig. 5. Using AIMSUN microscopic traffic simulation software (Transport Simulation Systems, 2009), a calibrated model of this 7-mile roadway was simulated. Focusing exclusively on the westbound direction, 87 eligible sites were identified as potential locations for new point sensor infrastructure, each site being approximately 300–500 feet apart. In the case study, the I-94 corridor was simulated for the PM peak period (4:00–7:00 PM) for Wednesday, July 26th, 2006, using the traffic data reported by the existing 16 loop detectors (shown in Fig. 5). At each of the 87 eligible sensor sites, traffic speeds in the simulation over this 3-h time interval were averaged for every 5 min and ground truth travel times between all sites were estimated. The mean-squared error for travel time were estimated between any proposed site $i$ and proposed site $j$. For configurations or scenarios where no sensors were deployed ($x_{ij} = 1$ or $V_{ij} = 1$, respectively) and no data was available, the mean-squared error was estimated to be the difference between actual corridorwide conditions and an assumed free-flow state.

5.2. Optimized no-failure configuration

In this case study, 16 loop detectors are budgeted for deployment ($R = 16$). It is assumed that the monetary budget, $B$, is very large and the costs at any site $j$, $C_j$, are very small, making the maximum number of budgeted sensors, $R$, be the influential budgetary restriction. Using the original mixed-integer linear program, sensors are deployed in such a manner along I-94 to minimize the error in travel time measurement, assuming that all sensors are fully functional and will not fail. The resulting configuration is shown in Fig. 6. The resultant mean-squared error of travel time is found to be 1.88, implying that this configuration produces highly accurate measures of corridor travel time.

Fig. 5. Loop detector configuration on I-94, as of July 2006. Loop detectors are represented by vertical segments. This figure is not to scale.
We want to point out that several detectors are clustered between the on-ramps and off-ramps at Riverside Ave./25th Ave. and Cretin Ave./Vandalia St. This seems counter-intuitive at the first sight, because there does not exist any flow sink or source in between, in other words, three detectors at one site should measure the exact same number. However, because high traffic volumes enter the freeway via on-ramps over the analysis period at both sites, bottlenecks are formed and the queue spills back which creates an environment where large sensor spacing generate high error rates. The model is deploying the three sensors consecutively to help capture that changing condition so that it can provide the most accurate travel time over the 3-h period. It is pretty common in the field to see turbulent flow dynamics even within areas where no new flows (demand) are introduced or removed, which may be a consequence of geometrics or some other environmental factor usually. The model employed in this paper believes three sensors within that segment (0.5 mile in length) will help generate the lowest corridor travel time.

Selection of a more appropriate “Error Factor” would help prevent situations like this. The one used here is simply mean-square error in travel time – this raw value by itself is prone to generate atypical deployment results, simply because it is so focused on one goal. As a practitioner, more robust measures for different traffic management tools can be developed, such as ramp metering, travel time, and incident management/detection. Introducing multiple objectives would likely end up with a more uniform deployment by doing that, which is why most agencies do that by rule-of-thumb.

5.3. Optimized failure-based configuration

If sensor failures were taken into account, would this deployment still be the best? For this case study, it is assumed that a data hub exists in the middle of the 7-mile corridor (approximately near the Huron Blvd exit), which is linked to a larger communications network. This assumed data hub receives all sensor data along I-94 and transmits it back to the managing agency for performance monitoring purposes. Each sensor that is deployed along this corridor ties into a single communications cable that parallels the corridor to the assumed data hub through a conduit beneath the ground. This communications cable is strictly linear and not part of a data ring, which means that, if severed due to a cable cut or a switch failure at a particular site, all sensor equipment beyond the sever point will lose communications capability with the assumed data hub and, thus, be considered non-operational. Therefore, as the distance between a proposed sensor and the assumed data hub increases, the probability for that sensor to “fail” increases as well. Intuitively, sensors closest to the hub are the least likely to have their connections severed (or fail) while sensors furthest from the hub are the most likely ones. For demonstration purposes in this example, we will assume the furthest eligible sensor site from the hub has a 3% chance of failure (97% of time being fully operational) and the eligible site closest to the hub has no chance of failure due to a severed cable (100% of time being fully operational). For any eligible site between the furthest site and the data hub, the occurrence of failure weighted linearly based on proximity. For example, a sensor site that is halfway between the data hub (with an adjacent site that incurs a 0% chance of failure) and the furthest eligible site (the site that incurs a 3% chance of failure) has an expected chance of failure of 1.5%; similarly, an eligible sensor site that is halfway between that site (with 1.5% chance of failure) and the data hub has an expected chance of failure of 0.75%, and so on. For this work, we will assume the sensor devices themselves will not fail, but that concerns of the communications cable being severed—due to a cut or a switch failure—is the element of interest for this design.

Considering the opportunities for sensors to fail when examining the optimal no-failure configuration discussed above, we find that the new mean-squared error for travel time has increased to be, on average, 300.72. This higher corridorwide error uses the exact same sensor configuration described in Section 5.2, but now considers all possible failure scenarios—which include all combinations of operational and non-operational sensors that make up the no-failure configuration. A mean-square error of 300.72 is a startling change compared to the original no-failure mean-squared error of 1.88, as it shows the sensitivity of the no-failure configuration once failure scenarios are examined. As discussed earlier in this paper, this is indicative that the configuration from Section 5.2 has certain failure scenarios that have a high likelihood of occurrence (e.g. such as by deploying many sensors away from the data hub) and/or contains a subset of operational sensors that are not aligned in an optimal manner (e.g. a key sensor or several key sensors in the original no-failure configuration is considered to have failed in this scenario).
The question now becomes whether another configuration exists that, while initially sub-optimal when sensors are assumed to not fail, is improved once failures are considered. To solve this, we utilize the heuristic algorithm identified in Section 4.2.3 to explore the 2nd-most optimal, the 3rd-most optimal, and so on until the stopping criteria are reached and a solution is available.

Fig. 7 illustrates the modified configuration proposed by the model that is deemed the most optimal given the sensor failure probabilities. Surprisingly, the solution algorithm found that the 346th most optimal no-failure configuration was actually the optimal configuration once sensor failures were considered. Without sensor failures, this configuration yields a mean-squared error for travel time of 10.84, a decline in accuracy from the optimal value of 1.88. However, given the clustering approach observed which produces redundancy, this configuration yields a mean-squared error for travel time of 12.66 once sensor failures are considered, which is a significant improvement when compared with the mean-squared error of 300.72 found with the original configuration. To ascertain this configuration as the best, the solution algorithm had to explore to the 504th most optimal no-failure configuration before the termination criteria was met. The convergence results of this search are shown in Fig. 8.

When comparing the no-failure configuration to the configuration with sensor failures, it is interesting to note how sensors have been reallocated. With sensor failures, the model places many between Hennepin Avenue and I-35W South, whereas the no-failure configuration only has one. Similarly, the configuration with failures deploys fewer sensors between Riverside Avenue and T.H. 280. While it is difficult to make any statement about this without overly focusing on the traffic conditions present, several causes can be proposed. First, the clustering of sensors near Downtown Minneapolis suggests that a sensor is critically needed in this area, perhaps to capture a variable traffic condition or bottleneck. This need is so critical that a sensor failure with only one sensor would increase error dramatically. Allocating many sensors to that region guarantees that only under slim chances will there be no sensor coverage. Second, it can be suggested that the increased chance of sensor failure away from the data hub has influence on the model, encouraging a configuration that provides redundancy in higher areas of failure versus lower areas. Of course, this would depend on whether a sensor is critically needed in these high failure regions. But, all things being equal, the model appears to be analyzing risk and reward, taking away sensors from some critical locations to ensure that more critical locations are provided supplemental coverage in the event of a sensor failure.

The configuration with sensor failures may not seem intuitive compared with the no-failure configuration, but it is important to reiterate that the sensor failure model is solving the problem differently than the no-failure model. This can be observed in the differences between the two configurations, which contrasts the most optimal no-failure configuration versus the 346th optimal no-failure configuration. But when sensors fail, the 346th optimal no-failure configuration yields a better accuracy than all other configurations, making it the best.

How would the result differ if sensors failed at a higher rate? To explore this, a case study similar to the one above is done using the same assumed data hub and concerns over a severed communications cable, but instead assuming the furthest eligible sensor site from the hub has a 15% chance of failure (85% of time being fully operational) rather than a 3% rate. Similar to the previous example, between these two sites, the occurrence of failure decreases at a linear rate, such that the eligible sensor site adjacent to the assumed data hub has no chance of failure due to a severed cable (100% of time being fully operational).

Using the model proposed in this paper, the 926th most optimal configuration yields the lowest error once failures are considered. Whereas this configuration normally yields a mean-squared error of 15.36 without failures, it yields a mean-squared error of 16.23 with failures. The model had to explore up to the 1533rd most optimal configuration to determine this result as valid. The deployment is shown in Fig. 9.

The convergence results of this search are shown in Fig. 10.

It is interesting to note the differences between the 3% and 15% failures. Whereas the 3% failure case placed most deployments near downtown Minneapolis, the 15% failure case placed more deployments closer to the data hub. This is to be expected, as locations closer to the hub are more reliable than locations further away. However, the disparity in...
Fig. 8. Algorithm convergence (3% max. failure). Termination occurred at \( k = 504 \) (for clarification purpose, only objective values after iteration \( k = 30 \) are shown).

Fig. 9. Proposed loop detector configuration based on the model's recommendations for 16 loop detectors with probabilistic failure (maximum failure rate of 15%). Data hub is located near the Huron Blvd. ramp.

Fig. 10. Algorithm convergence (15% max. failure). Termination occurred at \( k = 1533 \) (for clarification purpose, only objective values after iteration \( k = 30 \) are shown).
configurations comes as no surprise either. Since these sensors have a greater chance of failing, configurations that may have seemed the best for a 3% failure case are too problematic for a 15% failure case. Table 6 summarizes the results from this case study, identifying each of the optimal configurations determined by this model under a given set of failure conditions.

Although sensor failure rates may be higher or lower in reality, this case study has demonstrated the sensitivity of the sensor location problem. Even with a maximum 3% failure rate, the configuration has changed drastically, reflecting that sensor failures have much more than a trivial impact on an optimized configuration. Thus, sensor allocation models that do not account for these sensor failures will likely fail to find the truly best configuration in reality.

6. Conclusions

Despite the best of efforts, sensor failures are a serious problem that will not likely cease to exist anytime in the near future. Even though the consequences of these failures are widespread today, none of the existing sensor location models have attempted to directly address this problem, simply assuming that sensor deployments are always fully operational or can be repaired in quick time. To bridge the gap in existing literature, this paper proposes a probabilistic sensor location model to minimize the overall error of performance monitoring by addressing all possible failure scenarios for a certain sensor configuration. The proposed model focuses on a one-directional corridor, such as a freeway, and is applicable for point sensors, such as inductance loop detectors or remote traffic microwave sensors (RTMS), given that point sensors have been widely used on transportation networks.

A mixed-integer linear program is formulated to represent a deployed sensor configuration with all possible scenarios of failure, where each scenario contains a different combination of operational and non-operational sensors and a certain probabilistic likelihood of occurrence. By transforming a sensor failure scenario into a binary string with a distinct numeric value, a set of rigorous constraints is proposed to ensure that all applicable failure scenarios are uniquely and completely defined for the problem. A heuristic search solution algorithm is proposed that can explore no-failure configurations until an optimal configuration with failures taken into consideration can be identified. This model is then applied to a case study network, revealing that sensor failure rates, even when small, can have dire consequences that an optimal no-failure configuration becomes suboptimal if these failures are not considered. Therefore, accounting for sensor failures can change the proposed configuration when compared with prior sensor location models where failure is not considered.

The proposed model is applicable to any sensor deployment problem that uses point sensors along a one-directional corridor, which is a common case for many sensor deployment problems in the transportation field. The proposed model can serve as a decision-support tool for long-term planning situations by exploring all potential alternative configurations and recommending the truly optimal one after considering sensor failures. For situations where new sensors are being proposed as additions to an existing configuration, this model can help answer where these new sensors should be allocated or if these existing sensors should be replaced with more modern, less problematic options. For corridors that lack sensor infrastructure, such as roadway arterials that are converted into freeways, this model can assist in determining a good configuration. Lastly, for existing sensor configurations that suffer from recurrent widespread sensor failures, this model can help practitioners decide an affordable solution of either repairing such sensors or deploying additional newer sensors instead.

However, this model can still be generalized in several ways. First, as mentioned in the algorithm part, enumerating all possible failure scenarios can be tedious. In this paper, several possible methods are suggested but a more efficient approach of reducing computational requirements is needed. Second, the objective of the sensor location problem proposed in this paper is to minimize the expected estimation error. Stochasticity among sensor failure scenarios is not taken into consideration by optimal solutions from the developed model. A stochastic program will be able to incorporate such randomness. Lastly, in this paper, we assume a failed sensor would not provide any data. In reality, however, loop detectors with failure may generate random measurement. We will incorporate random measurement errors in future work.

Acknowledgements

This research was partially funded by the California Department of Transportation (Caltrans). Sincere thanks go to Mr. JD Margulici of University of California, Berkeley and Dr. Jeff Ban of Rensselaer Polytechnic Institute for their contributions and collaboration in this project.
References