SANDIA REPORT

SAND2018-10178

Unlimited Release

Printed September 2018

Developing Fugitive Emissions Sensor Networks

Katherine A. Klise, Bethany Nicholson, Carl D. Laird, Tatiana Flanagan, Arvind Ravikumar, Sindhu Sreedhara, Adam Brandt

Prepared by
Sandia National Laboratories
Albuquerque, New Mexico 87185 and Livermore, California 94550

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology and Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International, Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

Issued by Sandia National Laboratories, operated for the United States Department of Energy by National Technology and Engineering Solutions of Sandia, LLC.

Available to DOE and DOE contractors from

U.S. Department of Energy
Office of Scientific and Technical Information
P.O. Box 62
Oak Ridge, TN 37831

Telephone: (865) 576-8401
Facsimile: (865) 576-5728
E-Mail: reports@osti.gov
Online ordering: http://www.osti.gov/scitech

Available to the public from

U.S. Department of Commerce
National Technical Information Service
5301 Shawnee Rd
Alexandria, VA 22312

Telephone: (800) 553-6847
Facsimile: (703) 605-6900
E-Mail: orders@ntis.gov
Online order: https://classic.ntis.gov/help/order-methods/

SAND2018-10178
Printed September 2018
Unlimited Release

Developing Fugitive Emissions Sensor Networks

Katherine A. Klise, Bethany Nicholson, Carl D. Laird, Tatiana Flanagan
Sandia National Laboratories
P. O. Box 5800
Albuquerque, New Mexico 87185-1138

Arvind Ravikumar, Sindhu Sreedhara, Adam Brandt
Stanford University
367 Panama Street
Stanford, CA 94305

Abstract

This document summarizes research performed under the Laboratory Directed Research and Development (LDRD) project titled Developing Fugitive Emissions Sensor Networks: New Optimization Algorithms for Monitoring, Measurement and Verification. The purpose of this project is to develop methods and software to enhance detection programs through optimal design of the sensor network. This project includes both software development and field work. While this project is focused on methane emissions, the sensor placement optimization framework can be applied to a wide range of applications, including the placement of water quality sensors, surveillance cameras, fire and chemical detectors. This research has the potential to improve national security by improving the way sensors are deployed in the field.

ACKNOWLEDGMENTS

This work was supported by Sandia National Laboratories' LDRD funds, sponsored by the Geoscience Investment Area. Sandia National Laboratories is a multimission laboratory managed and operated by National Technology and Engineering Solutions of Sandia, LLC., a wholly owned subsidiary of Honeywell International, Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525. Field work and the testing of detection technologies was performed in collaboration with Stanford University and Environmental Defense Fund.

TABLE OF CONTENTS

FIGURES

TABLES

1 PURPOSE AND ACCOMPLISHMENTS

The oil and natural gas industry relies on a complex network of infrastructure for production, processing, transmission, and storage. This infrastructure experiences routine emissions that are part of normal operations as well as unplanned operational failures that can lead to fugitive emissions. These emissions cost operators lost revenue and cause environmental impacts, so significant efforts are underway to limit natural gas emissions. Both normal and fugitive emission rates can vary greatly over time and from site to site. Furthermore, individual sites are subject to complex wind patterns that transport emissions offsite, and the configuration of sites varies depending on equipment needs and layout. Reliable low-cost sensors are needed to accurately characterize emissions, alert operators to problematic equipment, and plan and prioritize repairs. In an era of low natural gas prices and rapid technological innovation, it is crucial to develop detection programs that make the best use of available and affordable leak detection technologies.

Recent development of low cost sensors has made it possible to monitor the environment more readily. However, even with low-cost sensors, only a limited number of sensors can be placed on any given site. Given the large number of potential leak sites across the natural gas sector, along with uncertain and rapidly changing atmospheric conditions, it is not feasible to place sensors downwind of every possible leak. The physical placement of sensors and the sensor technology that is used at a particular site can have a large impact on the performance of a sensor network. Furthermore, sensors can be placed for different objectives, including maximum coverage, early detection, or to quantify emissions. Different monitoring objectives may require different monitoring strategies, which need to be evaluated by stakeholders before sensors are placed in the field.

The purpose of this project is to develop methods and software to enhance detection programs through optimal design of the sensor network. This project includes both method development and field work. The methods make use of mixed-integer linear programming (MILP) formulations to optimize sensor locations and sensor type that maximize the effectiveness of the sensor network. The methods integrate atmospheric transport models with sensor characteristics, including fixed and mobile sensors, sensor cost and use site specific conditions which capture differences in meteorology, terrain, gas concentration, and emission characteristics. Field work was designed to quantify sensor detection capabilities and validate sensor models. The sensor models can be integrated into sensor placement optimization to provide realistic sensor behavior. While this project is focused on methane emissions, the sensor placement optimization framework can be applied to a wide range of applications, including the placement of water quality sensors, surveillance cameras, fire and chemical detectors. This research has the potential to improve national security by improving the way sensors are deployed in the field.

Project accomplishments are noted below, including journal articles and reports, open source software, field work, conference presentations, and partnerships. As part of this project, the following papers and reports were published (or submitted for publication), an additional paper is in preparation:

- J Liu and C.D Laird, (2018), A global stochastic programming approach for the optimal placement of gas detectors with nonuniform unavailabilities, Journal of Loss Prevention in the Process Industries, 51, 29-35. [1]

Abstract: Optimal design of a gas detection systems is challenging because of the numerous sources of uncertainty, including weather and environmental conditions, leak location and characteristics, and process conditions. Rigorous CFD simulations of dispersion scenarios combined with stochastic programming techniques have been successfully applied to the problem of optimal gas detector placement; however, rigorous treatment of sensor failure and nonuniform unavailability has received less attention. To improve reliability of the design, this paper proposes a problem formulation that explicitly considers nonuniform unavailabilities and all backup detection levels. The resulting sensor placement problem is a large-scale mixed-integer nonlinear programming (MINLP) problem that requires a tailored solution approach for efficient solution. We have developed a multitree method which depends on iteratively solving a sequence of upper-bounding master problems and lower-bounding subproblems. The tailored global solution strategy is tested on a real data problem and the encouraging numerical results indicate that our solution framework is promising in solving sensor placement problems. This paper was selected for the special issue in JLPPI from the 2016 International Symposium of the MKO Process Safety Center.

- K.A. Klise, B. Nicholson, and C.D. Laird, (2017). Sensor Placement Optimization using Chama, Sandia Report SAND2017-11472, Sandia National Laboratories. [2]

Abstract: Continuous or regularly scheduled monitoring has the potential to quickly identify changes in the environment. However, even with low-cost sensors, only a limited number of sensors can be deployed. The physical placement of these sensors, along with the sensor technology and operating conditions, can have a large impact on the performance of a monitoring strategy. Chama is an open source Python package which includes mixed-integer, stochastic programming formulations to determine sensor locations and technology that maximize monitoring effectiveness. The methods in Chama are general and can be applied to a wide range of applications. Chama is currently being used to design sensor networks to monitor airborne pollutants and to monitor water quality in water distribution systems. The following documentation includes installation instructions and examples, description of software features, and software license. The software is intended to be used by regulatory agencies, industry, and the research community. It is assumed that the reader is familiar with the Python Programming Language. References are included for additional background on software components. Online documentation, hosted at http://chama.readthedocs.io, will be updated as new features are added. The online version includes API documentation.

- K.A. Klise, B. Nicholson, C.D. Laird, A.R. Ravikumar, and A.R. Brandt, (submitted for publication), Sensor placement optimization for site-scale methane emissions monitoring, Environmental Science and Technology. [3]

Abstract: Advances in sensor technology have made it possible to monitor changes in air quality more readily. However, even as the cost of sensors declines, only a limited number of sensors can be installed at any given site. The physical placement of sensors, along with the sensor technology and operating conditions, can have a large impact on the performance of a monitoring strategy. This paper demonstrates methods to determine optimal sensor placement and sensor technology to improve the effectiveness of monitoring strategies for site-scale methane emission applications. The methods use site specific emission scenarios which capture uncertainty in wind conditions and emission characteristics. Mixed-integer linear programming formulations are then used to determine sensor locations and detection thresholds that maximize detection of the emission scenarios. Results show that even optimally placed high sensitivity sensors are not able to detect all emission scenarios. When compared to sensors placed near emission sources, the optimized sensor networks consistently increase the ability to detect leaks.

In an effort to make the methods developed in this research available to a wide audience, the research team released open source software for sensor placement optimization. The software is

built on Sandia developed R&D100 award winning Pyomo software [4] and can be used to design sensor networks that minimize impact (e.g. time to detection, plume extent, population impacted) or maximize coverage (e.g. detection of possible scenarios, detection of geographic regions). The software includes functionality to define point and camera sensors that can be stationary or mobile. The software, called Chama [2], is copyright through National Technology & Engineering Solutions of Sandia (SCR# 2146.0) and released under a revised BSD license. The software is hosted on Sandia National Laboratories GitHub site at https://github.com/sandialabs/chama. More information on Chama is available online at http://chama.readthedocs.io.

To develop and validate sensor models, the research team undertook three weeks of field trials using 10 mobile sensors (drones, trucks, and airplanes) at two locations. The sensors were used in a series of single-blind controlled release experiments that mimicked methane emissions at natural gas facilities. These experiments were designed to evaluate basic technology parameters including the ability to identify leak location, rate of false positives and false negatives, leak quantification accuracy, and the ability to resolve two closely spaced leak. Technology specifications from these experiments can inform sensor placement optimization to further improve cost-effectiveness of methane emissions detection.

As part of this project, the research team presented work at several conferences and workshops, including the Mary Kay O'Connor Process Safety Center International Symposium, American Geophysical Union Fall Meeting, Workshop on Convergent Data Science Solutions to Safer Systems (invited), PyomoFest, CoDA (Conference on Data Analytics for the Department of Energy), Environmental and Water Resources Institute Meeting, and American Institute of Chemical Engineers Annual Meeting.

Several partnerships were developed or continued as part of this research, including partnerships with Stanford University Energy Resources Engineering department; Environmental Protection Agency National Risk Management Research Laboratory, including sensor placement optimization for drinking water (see Section 7.2), Sandia's MELCOR Accident Consequence Code Systems (MACCS) group, including preliminary investigation into sensor placement optimization for nuclear response (see Section 7.4), industry representatives at the Purdue Process and Safety Assurance Center, including consultants at Kenexis (CRADA started as part of this project, see Section 7.5); Chemical Engineering graduate students at Purdue University; Texas A&M Chemical Engineering department, including a joint proposal written for the Gulf of Mexico Research Institute; and Sandia's Water Power Technologies group, including a joint proposal written to the Department of Energy Water Power Technologies Office. The research team is using the software and expertise gained from this project to promote the use of sensor placement optimization within a wide range of applications.

2 METHANE EMISSIONS FIELD WORK

Methane is a powerful greenhouse gas, 28-36 times more efficient at trapping radiation than carbon dioxide over a 100 year period [5]. A recent study of methane emissions reported that the Environmental Protection Agency estimate of 2015 oil and gas supply chain emissions is a significant underestimation of true emission levels [6]. This underestimation is attributed to methane released during abnormal operating conditions which are unaccounted for in conventional emissions estimation methods. Such fugitive methane emissions can erode the greenhouse gas emissions benefits of natural gas over coal. Regulations involving sensors for periodic leak detection and repair strategies can significantly reduce fugitive methane emissions. However, these technologies are often expensive (an infrared camera costs around $100,000) and slow to detect leaks.

There are several key questions that must be answered when designing effective sensor networks for methane emissions, including which type of sensors to deploy, how many sensors to use, and where to place the sensors. In addition, a sensor network can be designed for different objectives such as minimizing the time to detection or maximizing the number of potential leaks that might be detected. These objectives can be formulated as a cost-constrained optimization problem. Cost-effectiveness becomes a critical parameter especially considering the low natural gas prices in the global market. Therefore, it is crucial to determine the most reliable cost-effective leak detection technologies available.

Over the past several years, numerous methane detection technologies have been developed and are actively in use [7]. Satellite images can display daily large scale methane anomalies across the country, infrared cameras can be used to view leaks at the equipment scale in real-time, and highly precise $CH_4/CO_2/H_2O$ measurements can be taken from a moving vehicle linked to GPS. These new approaches promise faster, cheaper, and more effective leak detection than current methods. However, ensuring mitigation targets are achieved requires demonstrating that these new technologies are at least as effective in reducing emissions as current approaches. To do this, the research team undertook three weeks of field trials with 10 new mobile sensors (drones, trucks, and airplanes) at two locations. The sensors were used in a series of single-blind controlled release experiments that mimicked methane emissions at natural gas facilities. These experiments were designed to evaluate basic technology parameters including the ability to identify leak location, rate of false positives and false negatives, leak quantification accuracy, and the ability to resolve two closely spaced leak. The parameters obtained in this study can be used to compare performance with conventional leak detection approaches such as optical gas imaging. Furthermore, technology specifications from these experiments can inform sensor placement optimization models to further improve cost-effectiveness of methane emissions detection.

2.1 Methane Emissions Technology Evaluation Center

The Methane Emissions Technology Evaluation Center (METEC) is a controlled release facility designed to test Department of Energy-funded methane detection technologies through the MONITOR program. METEC is also open to other users. The site contains equipment typically found at a natural gas production site, including wellheads, separators, and tank batteries. Each piece of equipment has multiple leak points made from ¼ inch steel tubing. Methane is sourced from a centrally-located cylinder at 2500 psi and flow is controlled by a combination of pressure regulation plus three parallel discrete choked-flow orifices that in

combination allow for eight flow rates from each source. The pressure-dependent flow rate is calibrated for each leak source across a range of input pressures. Across the site, hundreds of possible leak locations exist.

The METEC facility is often used to test new sensor technologies through single-blind controlled releases. Given the widespread use of infrared cameras for methane leak detection, our first study at METEC aimed to improve our understanding of optical gas imaging (OGI) systems. These OGI systems form a "baseline" technology against which other detectors should be compared. Over five days in July 2017, we conducted multiple controlled release tests at different environmental and operating conditions. The results were used to determine the probability of detection as a function of leak size and imaging distance.

For this study, methane was release from three locations at the METEC site: a wellhead flange, a separator vent, and a wellhead valve. These leaks exhibit both diffuse and point-source characteristics. A set of eight unique leak sizes were randomly release at each location, including a null leak rate (0 g/h) and imaged at six distances. Each experiments was repeated in randomized fashion 10 times. The experiments and were broken down into two categories:

- Small short-distance (SSD): Small leaks measured at short distances with leaks in the range 3.3 – 83 g/h measured at 1.5, 3, and 6 m.
- Large long-distance (LLD): large leaks measured at long distances with leaks in the range 17 – 332 g/h measured at a distance of 6, 9, 12, and 15 m.

These two categories were used to explore the full parameter space of OGI technologies. Given the discretization in leak sizes, imaging distances, and repeat sets, each piece of equipment was imaged over 550 times for a total of 1680 binary yes/no observations across all experiments.

Figure 1 shows the empirical detection probability curves at different imaging distances for the SSD and LLD experiments, along with corresponding error bars. The median detection limit, defined as the leak rate at which the probability of detection is 50%, is 3, 6, and 20 g/h for imaging distances of 1.5, 3, and 6 m, respectively. Similarly, the median detection limits for the LLD experiments are 20, 50, 115, and 140 g/h for imaging distances of 6, 9, 12, and 15 m, respectively. Data taken across different atmospheric conditions over the week of field studies were aggregated in these figures and all experiments were randomized in time.

These experimentally-derived probability detection curves can be directly used in the sensor optimization models for OGI-based leak detection. Compared to the physics-based modeling framework used in prior optimization frameworks, the probability of detection curves derived here come from experiments at typical natural gas facilities and therefore better represents real-world performance. Furthermore, these probability detection curves can be derived for different environmental and operating conditions, rendering it flexible for use in a variety of field conditions. By extracting a functional relationship between median detection limits and imaging distance, one can simulate a variety of OGI-based leak detection configurations including truck-mounted, fence-line, and aerial-mounted OGI. Lastly, using such empirical probability detection curves significantly reduces the computational complexity in the sensor model by removing the need to model atmospheric transmission.

Figure 1. Experimentally-derived leak detection probability curves for (a) SSD and (b) LLD experiments. The horizontal error bars in (a) correspond to the accuracy of the mass flow controller, while the error bars in (b) correspond to 1 standard deviation of the mean in observed flow rates. The vertical error bars correspond to 95% confidence interval associated with the finite sample size.

2.2 Stanford/Environmental Defense Fund Mobile Monitoring Challenge Site

Members of this research team led the Stanford/Environmental Defense Fund (EDF) Mobile Monitoring Challenge, which sought to test the effectiveness of innovative mobile approaches to methane leak detection. The call for applications resulted in 28 technologies submitting to participate in the challenge. The research team selected 12 technologies for participation based on various criteria such as technology readiness, scientific capability, applicability to real-world methane leak detection issues, and ability to deploy at scale on short notice. Ten technologies ultimately participated in this test. Out of the 10 technologies, 3 were truck-based systems (Heath, Aeris, and University of Calgary), 6 were drone-based (Picarro, SeekOps, Advisian, ABB, BHGE, and University of Calgary), and 1 was plane-based (Ball Aerospace).

These technologies were tested over three weeks at two locations, METEC in Colorado and Northern California Gas Yard in California, between April and May 2018. The technologies were split-up by detection sensitivities, determined initially by detailed conversations with individual teams. The first week at METEC involved two technologies (Heath and Picarro) with detection sensitivities in the 0 – 2 standard cubic feet per hour (scfh) range. The second week at METEC involved 5 technologies (Aeris, Advisian, SeekOps, ABB, and BHGE) with detection sensitivities in the 5 – 10 scfh range. The final week in California involved 3 technologies (Ball Aerospace, and University of Calgary truck and drone systems) with detection sensitivities in the 100 – 1000 scfh range. At each of the test sites, we conducted controlled single-blind releases to evaluate basic technology parameters as shown in Table 1 below.

The data from this study will be made publicly available by early Fall 2018. Here, we present some representative results, without divulging company names. For each technology, approximately 100 leaks were tested over the course of one week. The technology operators reported binary results (yes/no) for detection of each leak event. The leak events included a significant number of null leak rates (0 g/h) to test for false positives. Table 2 gives a summary

of the performance from one team, which we will call Team A. Over a 1 week period, Team A tested 104 leaks, of which 63 were non-zero release rates. Out of the 63 leaks that were tested, the team correctly identified 59 of them with a true positive rate of 94% and false negative rate of only 6%. Furthermore, Team A was able to correctly identify the location of the leak in 50 of the 59 correctly-identified cases, corresponding to a location accuracy of 85%. Team A had a zero false positive rate, meaning that they identified all 41 of the 'zero' leaks correctly. This team exhibited the best in class performance for both detection and quantification. In addition to the binary yes/no data, Team A was also able to provide real time initial estimates of leak quantification – very few teams were able to provide such information during the day of the test.

Table 1. List of parameters tested in the Stanford/EDF Mobile Monitoring Challenge.

Parameter	**Description**
Location identification	Equipment level (both type and number) Component level
Binary Yes/No detection	True/False positive percentage True/False negative percentage Detection probability curves (based on data availability)
Speed	Time taken for data acquisition on the field
Quantification accuracy	Parity chart of controlled leak tests
Ability to resolve leaks	Small vs. large leaks at close proximity Multiple similar leaks in close proximity Multiple leaks on the same pad

Table 2. Performance summary from "Team A" in the Stanford/EDF Mobile Monitoring. Challenge

Total number of leaks			104
Leaks with non-zero release rate			63
Leaks with zero release rate			41
	Detected	**Not detected**	**Total**
Leak	59 (94%)	4 (0.6%)	63
No Leak	0 (0%)	41(100%)	41

In addition to binary detection characteristics, we also asked the teams to quantify leaks on 2 of the 5 days of testing. On the first quantification day, teams were given only one leak to quantify. This helped test the accuracy of their algorithms. On the second quantification day, we gave the teams multiple leaks (up to 3) to quantify. The teams had the option to evaluate facility-level leak rate estimates, or individually measure each of the leaks. Figure 2 shows the accuracy of that estimate for Team A. The measured leak rate is typically about twice that of the actual leak rate, with an error of about 50% in each individual measurement. The red line corresponds to the 1:1 line, where the measured leak rate would be identical to the actual leak rate.

Quantification of emissions rates from atmospheric measurements, in general, is a very difficult inverse modeling problem. While a given concentration profile is measured by a technology, a

number of possible real leaks could give similar atmospheric readings. Given the number of variables involved in atmospheric dispersion and turbulence that can affect point concentration measurements, it is remarkable that new technology can come within two times the actual leak rate. Not surprising, Team A exhibited best-in-class performance we observed of all technologies tested. Such controlled release tests comparing different technologies are critical to understand the benefits and limitations of different types of technologies. Future work will involve incorporating some of the technologies tested as part of this challenge into the sensor placement optimization software developed in this project. With empirical results such as those generated in the Mobile Monitoring Challenge, optimization routines can be used to optimally place or deploy such detectors.

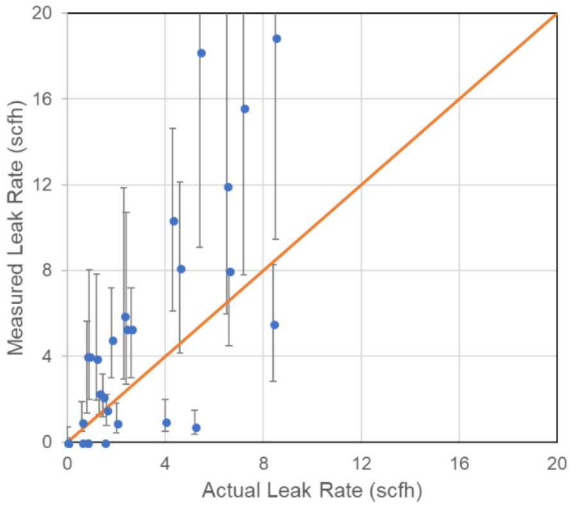


Figure 2. Quantification parity chart (actual vs. measured leak rate in scfh) for “Team A” in the Stanford/EDF Mobile Monitoring Challenge.

3 ATMOSPHERIC DISPERSION MODELING

Sensor placement optimization requires information about detectability from a wide range of sensors under different field conditions. This information is typically obtained by combining data from system models and sensor models. For air quality applications, the system model is typically an atmospheric dispersion model that is used to simulate uncertainty in contaminant concentrations subject to emission sources, site geometry, and local atmospheric conditions.

Depending on the region of interest and the complexity of the system, very detailed or coarse atmospheric dispersion models can be used to simulate emissions. The Weather Research and Forecasting model [8], the Hybrid Single Particle Lagrangian Integrated Trajectory model [9], and the Global Chemistry Transport model [10] are all intended to model atmospheric conditions and transport from the national to global scale. At the local scale (10s of km), simple Gaussian plume dispersion models are commonly used to model air quality. The steady state Gaussian plume model, AERMOD [11], and the transient Gaussian puff model, CALPUFF [12], are recommended by the Environmental Protection Agency for regulatory purposes at this scale. These models are simple to run and computationally efficient. On the other end of the spectrum, computational fluid dynamics (CFD) models can be used to simulate detailed site and transport characteristics and "building aware" models can be used to model transport through urban settings [13]. Between these two extremes, there is a surprising lack of models that are complex enough to account for effects of turbulence and atmospheric variation (i.e., varying wind speeds, calm conditions), but not so complex as to be impractical for simulation and optimization workflows. Because optimization workflows typically require many thousands of simulations, methods like CFD will likely remain too computationally expensive for practical use. New models are especially needed where pollutant dispersion is dominated by local turbulence instead of wind-assisted dispersion.

3.1 Modified Gaussian Plume Dispersion Model

In this project, the research team developed a modified Gaussian plume dispersion (m-GPD) model that builds on the standard Gaussian plume dispersion (GPD) model and incorporates turbulent "clumping" effects on plume concentration profiles at a parametric level. In a natural gas facility, dispersion is often influenced by localized turbulence because of the presence of equipment in the path of the plume. This turbulence causes local eddies to form within the plume, often making the simple GPD model unreliable in predicting flux rates. The turbulence results in plumes forming localized regions of higher or lower than expected pollutant concentration. Given sufficient input data and computational resources, localized turbulence can be modeled by solving complex fluid dynamic equations or Lagrangian particle-tracing algorithms which require detailed 3-D maps of local wind fields, terrain data, surface roughness, and atmospheric boundary layer conditions. These parameters are not commonly available at remote natural gas sites. To overcome this data availability issue, we develop the m-GPD model that is computationally fast and does not require detailed parameterization. The m-GPD strikes a balance between the simplistic GPD that is not applicable at many oil and gas facilities because of turbulence and other complex models that require extensive parameterization and computation time. The m-GPD model can be readily integrated into other simulation and optimization frameworks for improved sensor performance characterization.

The standard GPD model calculates methane concentration downwind of a given leak rate. The plume concentration, $\Phi(X;u)$, at a downwind distance, x, from a leak is given by,

$$\Phi(X;u) = \frac{Q}{2\pi u \sigma_y(x)\sigma_z(x)} exp\left(\frac{(y-y_0)^2}{2\sigma_y^2(x)}\right)\left[\exp\left(\frac{(z-z_m(x))^2}{2\sigma_z^2(x)}\right) + exp\left(\frac{(z+z_m(x))^2}{2\sigma_z^2(x)}\right)\right] \quad (3\text{-}1)$$

where u is the wind-speed, z is measured above the ground, y is measured perpendicular to the downwind direction, y_0 is the y-axis position of the leak source, Q is the leak flux, and $\sigma_y(x)$ and $\sigma_z(x)$ are the dispersion coefficients of the plume concentration extracted using analytic approximations of published curves based on Pasquill-Gifford atmospheric stability classes [14]. $z_m(x)$ tracks the vertical position of the center of the plume based on buoyancy of the gas (a small effect compared to wind-driven plume transport) and follows standard treatment as shown by Beychok [15].

The m-GPD model represents turbulence by introducing random concentration distortions into the Gaussian plume along the dominant wind direction. Here, the concentration path-length from the GPD model is modified using parameters that describe location, size and magnitude of distortions. These parameters are described below in Sections 3.1.1, 3.1.2, and 3.1.3. Each of these parameters introduces variations in the smooth concentration path length profile of GPD. The parameters are illustrated in Figure 3. The figure shows a simulation of a 1 g/s leak originating at a height of 2 m under a wind velocity of 1.5 m/s. The atmospheric stability class is assume to be C, and the dashed line denotes the center line of the plume. The m-GPD model represents turbulence by introducing random concentration variations 'μ' with a circle of radius 'r' at a randomly chosen point P(x_0,z_0).

After the m-GPD model has been generated, the entire plume is scaled by the total concentration in the GPD model. This normalization ensures that mass is conserved and that the total mass is identical in the GPD and m-GPD cases. We empirically calibrate model parameters using recorded plume videos of controlled releases as discussed in below in Section 3.2.

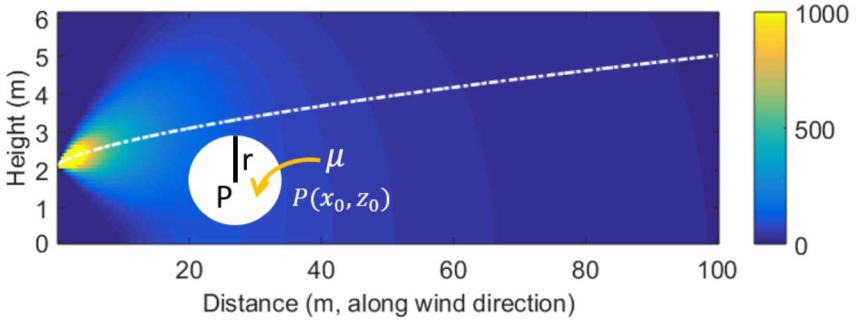


Figure 3. Illustration of parameters in m-GPD model: μ, r, and P(x_0,z_0).

3.1.1 Location Parameters

Location parameters represent distortion in concentration along the x, y, and z directions with respect to the GPD model. In all cases, the x-axis is along the dominant wind direction, while the z-axis is perpendicular to the ground. The center-point (x_0,y_0,z_0) of a distortion in concentration path-length is given by:

$$x_0 = (1+\delta)x_c, \quad y_0 = \omega\sigma_y^c(x), \text{ and } \quad z_0 = z_m^c + \beta\sigma_z^c(x) \quad (3\text{-}2)$$

where parameters δ, ω, and β introduces randomness in the chosen (x_c, 0, z_m^c) so that distortions do not appear exactly along the center line ($-1 \leq \delta,\omega,\beta \leq 1$). x_c is the x-coordinate of a set of randomly chosen points along the x-direction to introduce the distortion, z_m^c is the location of the plume center line at x_c. The higher the absolute value of ω and β, the farther the distortion is from the centerline, along the y and z directions respectively. The value of δ is randomly chosen from a uniform distribution to vary plume location along the x-axis direction.

3.1.2 Size Parameters

The distortions are modeled as spheres where the concentration path-length is higher than the surrounding area. Experimental observations show that the characteristic size of these distortions tends to get larger as you move farther away from the source – i.e., the radius of the distortion is a function of downwind distance and "tighter" turbulent swirls grow in size as they move downwind. To preserve the overall Gaussian nature of the plume, the minimum and maximum radius of the distortions, r_{min} and r_{max}, are represented as a fraction of the dispersion on the z-axis at a particular location, $\sigma_z^c(x_c)$. That is,

$$r_{min} = \theta \times \min\left(\sigma_z^c\right) \; and \; r_{max} = \phi \times \max\left(\sigma_z^c\right) \tag{3-3}$$

where $r_{min} < r_{max}$. The radius of the distortion is linearly interpolated between these two values using the equation,

$$r = r_{min} + \left[\frac{r_{max} - r_{min}}{\max\left(x_c\right) - min(x_c)}\right](x - min(x_c)) \tag{3-4}$$

3.1.3 Magnitude Parameters

Given our model of spherical concentration enhancements, the magnitude of the concentration path-length within the region of distortion is by definition higher than the concentration in the surrounding region. Intuitively, these concentration enhancements should approach background concentration as the plume moves farther away from the source. This means that the disturbed turbulent plume should eventually relax to the Gaussian approximation far from the emission point. The concentration enhancement magnitude parameter, μ, is modeled as an exponential decay function dependent on the downwind coordinate, x, as shown below:

$$\mu = 1 + \alpha e^{-x/\tau} \tag{3-5}$$

where α is the intercept, or concentration enhancement for disturbances at the point of emission, and τ is a scaling factor that determines the distance at which the m-GPD model starts to resemble the GPD model. Large values of τ imply long distances for the magnitude parameter μ to relax to 1. For example, at non-dimensional length $x = 3\tau$, the strength of disturbance is reduced to e^{-3}, or ~1/20 of its initial value.

3.2 Field Calibration

To calibrate the m-GPD model, we use footage of methane leaks using a FLIR GF-320 infrared camera at the Methane Emissions Technology Evaluation Center (METEC) in Fort Collins, CO over 5 days in July 10 – 14, 2017 (see Section 2.1). In this study, we use videos from leaks emitted at separators to validate the m-GPD model. Leak-sizes ranged from 0 standard cubic feet

per minute (scfm) to about 140 scfm. This represents a typical range of leak-sizes observed at natural gas upstream facilities. Each run consists of 8 leak levels, including no leak, run for approximately 3 minutes each, at capture rates of 9 frames/second (limited by IR sensor capabilities).

Under steady state conditions, methane plumes average over time to an approximately Gaussian envelope [16]. That is, averaging multiple frames of a leak video will provide a real-time approximation the underlying Gaussian dispersion. The number of frames used in the average is a function of steady-state atmospheric conditions: if the wind changes direction, then the Gaussian characteristics are disrupted. Each frame can then be subtracted from the time-averaged Gaussian plume. The deviation from the average Gaussian plume represents turbulence and other atmospheric distortion that we would like to incorporate in the m-GPD model. Statistics are derived from this "deviation" image using feature recognition software to provide estimates of the location, size, and magnitude parameters used in this model.

Figure 4 illustrates this process using a 100 scfh leak from a separator vent, imaged at a distance of 10 ft. Note that released gas appears lighter than background sky in these weather conditions. Figure 4a is a single frame in a 100-frame video showing turbulence. Figure 4b shows the average of 46 frames of the leak. As expected, the dispersion of the "average" plume is Gaussian. These 46 frames were chosen based on visual observation where the atmospheric conditions did not change significantly to shift plume location. Figure 4c shows the result of subtracting the single frame (a) from the time-averaged Gaussian plume (b), allowing us to visualize "hot spot" regions in the plume that deviate from the Gaussian dispersion model by having enhanced concentration. Figure 4d shows the process of binarizing the differenced image (c) and using circular Hough transform to identify circles within the image.

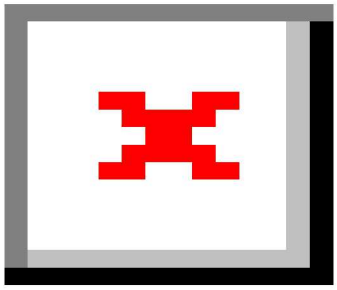
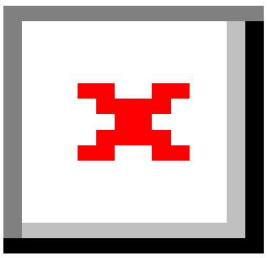


Figure 4. (a) A single frame of the leak video, (b) average of 46 frames of the leak video, exhibiting Gaussian dispersion characteristics, (c) deviation of the single frame from the average used to determine m-GPD model parameters, and (d) binarizing the image and using circular Hough transform to identify circles.

To extract the location and size parameters of the m-GPD model, the Gaussian deviation image (Figure 4c) is converted to a binary image using Otsu's thresholding technique [17]. We then use the circular Hough transform algorithm [18] to identify circular distortion regions in the plume (Figure 4d). The circular Hough transform selects the location and size of the circles by maximizing the fraction of the binarized object within each circle. Finally, the magnitude parameter is obtained by overlaying these circles onto each individual frame and calculating the ratio of the pixel intensity in that frame to that of the averaged frame (b), where more brightness corresponds to more gas concentration relative to background or time-averaged concentrations.

All the frames from a given video can then be analyzed to extract the location, size, and magnitude parameters. Each video corresponds to a single leak imaged at distances varying from about 5 m to 20 m from the source, each for a period of 3 minutes. Figure 5a shows the radius of circles as a function of downwind distance from the leak source. As expected, we see that the radius of the distortions increases the farther one moves from the source of the leak. The median rate of increase of the radius of the circles is about 0.016 m/m along the downwind direction. From Figure 5a, it is clear that a linear regression fit to the radius of the circle as a function of the distance from the leak source does not fully capture the dynamics of plume development. For example, even as the radius gets bigger as you move farther away from the source, there are still many turbulences with smaller radii at farther distances. Using the location and radii of the circles, next we estimate the magnitude of concentration enhancement within the distortions in the plume. To calculate this, we arbitrarily assign a concentration index of 1 to the circles closes to the leak source, and estimate the concentration ratio of all other circles in reference to this. In addition, we also normalize the concentration to the area of the circles. Based on this choice of normalization, the concentration ratio decreases as we move away from the leak source as shown in Figure 5b. The ratio is fit to an exponential decay curve, as expected in dispersion modeling.

(a) (b)



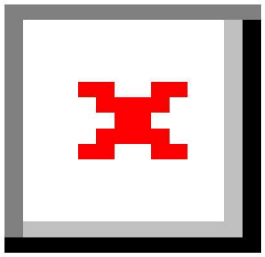


Figure 5. (a) Radius of the distortion circles as a function of downwind distance from a leak video of 100 scfh taken at an imaging distance of 10 ft from the leak source. (b) Ratio of concentration within the turbulent circles as a function of distance from the leak source.

Figure 6 shows an example simulation using the GPD model and m-GPD model. The simulation uses a 1 g/s methane leak at a constant wind-speed of 1.5 m/s under atmospheric stability class C. The m-GPD model plume is simulated using the size and magnitude parameters derived from field calibration, while the location of circles were chosen randomly about the Gaussian plume center line. Figure 7 shows the concentration profile along the downwind direction at three different vertical heights of a point sensor, using both the GPD and m-GPD models. As expected, the concentration is highest at the location of the leak with a value around 10,000 parts per million (ppm) – typical values seen very close to the leak source. The total methane concentration under both dispersion models are identical. Including the effects of turbulence breaks the monotonic behavior of leak detection probability as a function of downwind distance. Because total mass of the pollutant is conserved, there is not a large difference in leak detection probability across aggregate sample sizes. Therefore, while this model can better predict whether individual leaks are detected by a point sensor, it will replicate the GPD model in the aggregate.

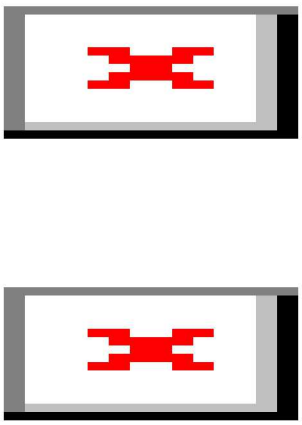


Figure 6. Simulation of a 1 g/s methane leak at a constant wind-speed of 1.5 m/s under atmospheric stability class C using (top) GPD model, and (bottom) m-GPD model.

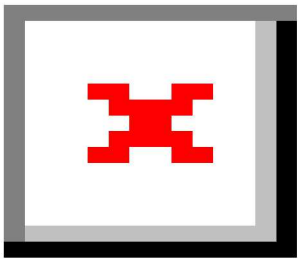


Figure 7. Downwind methane concentration from the GPD (solid lines) and m-GPD (dotted lines) models at three different point sensor heights: 2 m (blue), 1 m (red), and 5 m (green).

4 SENSOR TECHNOLOGIES AND MODELS

Many different sensor technologies exist. In the context of gas detection, sensors can monitor the concentration at a fixed point, along a line (path integral detectors), or create a 2-d measurement of concentration (as in optical gas imaging technology described above). Sensors can monitor continuously or take measurements at defined sampling times. Sensors can also be mounted on vehicles or drones and move through a specified region. Furthermore, sensors can have different operating conditions which can change detectability. In order to understand the tradeoffs between different sensor technologies and operating conditions and to select an optimal subset of sensors, these different options should be considered simultaneously within sensor placement optimization. In this section, we discuss the sensor technologies modeled in this study – namely point sensors and imaging sensors. These sensors are also included in the optimization framework described in the next section. We conclude with a brief discussion on new mobile sensor technologies that are being developed for methane leak detection applications including trucks, drones, and plane-based systems.

4.1 Point Sensors

Point sensors measure direct atmospheric concentrations at the location of the sensor. High sensitivity sensors (e.g., Picarro) can measure concentrations down to parts per billion (ppb) level, while hand-held methane sensors typically have sensitivities in the 10 – 1000 parts per million (ppm) range. Typically, these sensors are based on laser spectroscopy, flame ionization systems, or electro-chemical detection methods. Given the safety concerns at large natural gas facilities, laser spectroscopy based systems are only used in situations where the sensor is deemed inherently safe. The cost of these sensors are typically proportional to their sensitivity, with high-precision laser-based sub-ppm detectivity sensors costing upwards of $200,000 per unit. While point sensors are most ideally suited to detect the smallest of leaks, they are also highly sensitive to location and atmospheric conditions. Given the high cost of detectors, some technologists are developing mobile platforms that allow the same sensor to measure many sites per day.

To model point sensors, concentration measurements at the sensor location are simply extracted from the simulation of methane concentration. If concentration is above the sensor detection threshold, a detection is recorded. If a point sensor is placed on a mobile unit (truck, drone, plane, etc.), the position of that sensor can be modeled as a function of time. When modeling point sensors, it is important to consider the resolution at which the system is modeled. For example, if the atmospheric dispersion model uses a 10 m by 10 m grid, concentrations in a single cell is generally averaged over that region and the detection threshold should be adjusted accordingly.

4.2 Imaging Sensors

Optical gas imaging (OGI) sensors measure a concentration path-length along the field of view of the camera. OGI sensors can monitor a larger area, but generally at a higher cost as compared to point sensors. An infrared camera can cost $100,000. Typically, OGI sensors are brought to a site periodically (i.e. seasonally or once a year) and positioned such that a plume would be perpendicular to the camera view. This position maximizes the ability of the camera to detect a leak and requires knowledge of wind conditions. As mentioned in Section 2, the research team completed field work to categorize detection parameters of OGI technologies. The research team

then extended previous developed OGI models developed at Stanford [16] based on a FLIR's GasFind-IR cameras (GF-320).

Modeling OGI based leak detection requires modeling of two related physical processes: 1) infrared optical absorption by methane, and 2) optical properties of the surrounding environment. OGI relies on infrared absorption by gas molecules, as described by the Beer-Lambert's law:

$$\frac{I_t}{I_0} = e^{-K_\alpha \cdot \rho} \tag{4-1}$$

where K_α is the molar absorption cross-section, ρ is the concentration path-length, and I_t and I_0 are the transmitted and incident light intensity, respectively. K_α is calculated using the high-resolution transmission (HITRAN) molecular absorption database using the spectral line-widths, and line-strengths of the molecular transitions. The uncertainties in the spectral line-strengths in the HITRAN database is significantly smaller than the uncertainty from other operator-controlled parameters, and can therefore be neglected. The optical properties an infrared camera are influenced by four components: 1) the direct radiance from the methane plume, 2) the transmitted radiance from the background, 3) the background-reflected cold-sky radiance, and 4) direct atmospheric radiance. Each of these components are affected by environmental properties such as temperature of the background and the plume, emissivity of the background, humidity, and apparent temperature of the atmosphere and its emissivity. At short imaging distances (within 100 ft of the leak), atmospheric transmission is approximately one, and so direct atmospheric radiance can be neglected. Furthermore, except under cases where the background emissivity is low (dark, absorbing backgrounds like soil, forests, etc.), the background-reflected cold-sky radiance is at least an order of magnitude larger than the background radiance transmitted through the plume. Finally, we are left with two components to the radiance observed by the OGI system: 1) the direct radiance, and 2) the transmitted radiance. This change in observed radiance, $\Delta L_{obs}(\nu)$ with and without the methane plume can be written as,

$$\Delta L_{obs}(\nu) = (1 - e^{-K_\alpha \cdot \rho})(B(T_p,\nu) - \epsilon_b B(T_b,\nu) \tag{4-2}$$

where ρ is the concentration path-length, and $B(T,\nu)$ is the Planck blackbody function for a body at temperature T and wavenumber ν. Here, the background emissivity corresponds to the composite emissivity of the 'clutter' present in the scene of the methane leak. Furthermore, we assume that the background emissivity is constant within the narrow spectral range of the filter used in the imaging process. A pixel registers an intensity change if this difference in radiance, ΔL_{obs}, is greater than the Noise Equivalent Power (NEP) of the IR camera. Detailed model equations can be found in [16]. For this project, the OGI model was modified to be integrated into the sensor placement optimization framework, this allows the analyst to specify the cameras location, orientation, field of view, and detection properties. As with point sensors, OGI sensors can be placed on mobile units (i.e., trucks, drones, planes) and the position of the camera can be modeled as a function of time.

4.3 Future Sensing Technologies

With increasing focus on reducing methane emissions, a number of new sensing platforms are being developed. These sensors promise cheaper, faster, and more effective leak detection compared to existing sensors like OGI. While low-cost continuous monitoring point sensors have been developed in recent years (IBM or Xerox PARC), much innovation has been focused on

fast mobile approaches to leak detection. It is anticipated that future leak detection and repair programs at oil and gas facilities would evolve into a two-tiered process where *all* facilities are first "screened" with a fast-technology to identify those facilities that have anomalous emissions. Detailed leak detection and repair surveys with OGI or other hand-held instruments would then be conducted at those facilities flagged during the screening process. Because of the well-known issue of super-emitters, it is expected that rapid screening will flag only a small number of sites for further study, which would render the two-tiered program cost-effective. In this section, we briefly discuss some of the future sensing technologies that can be incorporated into an optimization framework. These include truck-based, drone-based, plane-based, and satellite-based technologies.

Truck-based systems: Truck-based systems typically measure concentration profiles over time and space and can back calculate leak location and flow rate using dispersion algorithms. Often, the effectiveness of truck-based systems depend on the effectiveness of their dispersion models and the complexity of the site. Nevertheless, these system can be used for fast fence-line screening. Indeed, many recent measurements in Alberta and British Columbia in Canada have used such methods to study methane emissions across oil and gas facilities. While fence-line studies lend themselves very well to the screening method described above, significant challenges exist in attributing individual plumes to specific sites, especially when the line-of-sight along the wind direction contains multiple potential leaks.

Drone-based systems: Drone-based systems are versatile in that various flying patterns can be programmed that can take into account prevailing winds and site geography. Flight paths could be optimized to gather data from locations that will help quantify methane leaks. However, drone systems are not intrinsically safe due to their use of Li-ion batteries and so cannot be used at high risk areas such as natural gas processing facilities. In addition, battery life of existing drone-based systems are limited to about 45 minutes of flying time, with many implementations challenged to reach these levels. Advances in battery life, sensor weight, and safety will help speed-up adoption.

Plane-based systems: Plane-based systems are well suited for surveying large areas in limited time. Indeed, such systems have recently been used to measure regional and basin-level methane emissions in the US and Canada. While the survey cost per site is significantly smaller than other methods described here, plane-based systems typically suffer from high detection limits. These detection limits are often two orders of magnitude or higher than truck- or drone-based systems. However, if emissions are driven by super-emitters, it could be advantageous to use a plane-based system for fast screening across large areas.

Satellite-based systems: Satellite-based systems are considered ideal for methane leak detection and monitoring systems. Despite their high capital costs, a single satellite can stay in orbit for five years and can map a particular geographic location every few days providing near continuous monitoring. Moreover, the operating costs of satellites are very low and can provide data to a large number of operators, regulators, and scientists. However, existing satellites do not have the spatial or detection threshold resolution to be useful in identifying individual facilities. For example, TROPOMI, the most advanced methane measuring satellite, has a spatial resolution of about 7 km by 7 km, and a detection threshold in the several tons per hour. While existing satellites cannot be used for leak detection applications at the facility level, there are many new developers in this field including GHGSat (Montreal, Canada) and EDF with promising ideas.

5 SENSOR PLACEMENT OPTIMIZATION

The physical placement of sensors, along with the sensor technology and operating conditions, can have a large impact on our ability to detect changes in the environment. In this project, the research team developed a framework to determine optimal sensor placement and sensor technology to improve the effectiveness of monitoring strategies. Sensors can be placed to minimize impact (e.g. time to detection, plume extent, population impacted) or maximize coverage (e.g. detection of possible scenarios, detection of geographic regions). This work was focused on sensor placement for site-scale methane emission applications. However, the methods are general and can be applied to a wide range of applications, including the placement of water quality sensors, surveillance cameras, fire and chemical detectors. For methane emissions, the optimization requires a set of simulations that model site specific emission scenarios which capture uncertainty in wind conditions and emission characteristics. Mixed-integer linear programming (MILP) formulations are then used to determine sensor locations and operating conditions that maximize detection of the emission scenarios.

Some of the mathematical approaches in Chama were originally developed by Sandia National Laboratories and the U.S. Environmental Protection Agency to design sensor networks to detect contamination in water distribution systems [19, 20, 21]. In this context, contamination scenarios are simulated using a water distribution system model, feasible sensor locations and thresholds are defined, and the optimization method selects a set of sensors to minimize a given metric. In this project, these methods were extended to include additional options such as sensor models, coverage and side constraints, and so that the methods could be applied to other applications.

5.1 Impact Assessment

Impact assessment extracts the impact of a particular sensor detecting a particular scenario, where that impact can be measured using a range of metrics. As described in Section 3, very detailed or coarse atmospheric dispersion models can be used to simulate methane emissions. Each scenario captures uncertainty in wind conditions, leak rate, and other variables. In the case of site-scale methane emissions, a Gaussian dispersion model is typically used. The ability of a sensor to detect a scenario depends on several factors, including the scenario environmental conditions, sensor location, and sensor operating parameters. While some scenarios might be detected multiple times by a single sensor, other scenarios can go undetected by all sensors.

Impact can be measured using metrics such as time to detection, population impacted, or volume of contaminant released before detection. These metrics can be used in impact-based optimization formulations. Additionally, these impact metrics can be used to define if a sensor is able to detect a particular scenario for use in coverage-based optimization formulations. These formulations are described below.

5.2 Impact Formulation

The Impact formulation is used to determine optimal sensor placement and sensor type that minimizes impact, where impact can be the sensor's detection time or some other measure of damage. The Impact formulation, which is based on the P-median facility location problem, is described below:

$$minimize \quad \sum_{\alpha \in A} \alpha_a \sum_{i \in L_a} d_{ai} x_{ai} \tag{5-1}$$

$$\text{subject to} \quad \sum_{i \in L_a} x_{ai} = 1 \qquad \forall a \in A \tag{5-2}$$

$$x_{ai} \leq s_i \qquad \forall\, \alpha \in A, i \in L_a \tag{5-3}$$

$$\sum_{i \in L} c_i s_i \leq p \tag{5-4}$$

$$s_i \in \{0,1\} \qquad \forall\, i \in L \tag{5-5}$$

$$0 \leq x_{ai} \leq 1 \qquad \forall\, \alpha \in A, i \in L_a \tag{5-6}$$

where:

- A is the set of all scenarios
- L is the set of all candidate sensors
- L_a is the set of all sensors that are capable of detecting scenario a
- α_a is the probability of occurrence for scenario a
- d_{ai} is the impact assessment, and represents some measure of the impact or damage that will be incurred if scenario a is first detected by sensor i
- x_{ai} is an indicator variable that will be 1 if sensor i is installed and that sensor is the first to detect scenario a (where first is defined as the minimum possible impact, usually defined as time to detection)
- s_i is a binary variable that will be 1 if sensor i is selected, and 0 otherwise
- c_i is the cost of sensor i
- p is the sensor budget

The size of the Impact formulation is determined by the number of binary variables. Although x_{ai} is a binary indicator variable, it is relaxed to be continuous between 0 and 1, and yet it always converges to a value of 0 or 1. Therefore, the number of binary variables that needs to be considered by the solver is a function of the number of candidate sensors alone, and not the number of scenarios considered. This makes the formulation scalable to large systems.

This formulation has been used to place sensors in large water distribution networks [19, 20, 21] and for gas detection in petrochemical facilities [22]. In this research, the Impact formulation was used to optimize sensors location along with sensor technology. While the formulation is the same, the input data includes the impact of detecting scenarios using different sensor technologies. This allows for side by side comparison of different sensor technologies given a set budget. Additional criteria could be considered to ensuring redundancy (requiring observation by at least N cameras).

5.3 Coverage Formulation

The Coverage formulation is used to place sensors that maximize the coverage of a set of entities, where an entity can be a scenario, the times at which a scenario is detected, or geographic location. The Coverage formulation is described below:

$$minimize \quad \sum_{a \in A} \alpha_a x_a \tag{5-7}$$

$$subject\ to \quad x_a \leq \sum_{i \in L_a} s_i \qquad \forall a \in A \tag{5-8}$$

$$\sum_{i \in L} c_i s_i \leq p \tag{5-9}$$

$$s_i \in \{0,1\} \qquad \forall\, i \in L \tag{5-10}$$

$$0 \leq x_{ai} \leq 1 \qquad \forall\, \alpha \in A \tag{5-11}$$

where:

- A is the set of all entities
- L is the set of all candidate sensors
- L_a is the set of all sensors that cover entity a
- α_a is the objective weight of entity a
- x_a is an indicator variable that will be 1 if entity a is covered
- s_i is a binary variable that will be 1 if sensor i is selected, and 0 otherwise
- c_i is the cost of sensor i
- p is the sensor budget

This formulation is similar to the Impact formulation, however there is no benefit to early detection, the entity is either detected or not. As with the Impact formulation, the number of binary variables is a function of the number of candidate sensors and not the number of entities considered. This means that the Coverage formulation is scalable to large problems.

This formulation can be used to design a sensor network that maximizes the chance of detecting a wide range of scenarios. This is useful when optimizing sensors for continuous monitoring of a particular site where site conditions change frequently. This formulations can also be used to maximize geographic coverage. This is most useful when using cameras. Example applications include placing security guards or cameras observe assets, and placing fire detectors in buildings, manufacturing sites, or process facilities. These problems can be formulated with several different objectives and constraints, including maximizing the area or volume coverage (assuming that resources may not be sufficient for full coverage), or minimizing the number of cameras required to achieve a certain level of coverage.

5.4 Grouping Constraints

Constraints can be added to both the Impact and Coverage formulations in order to enforce or restrict the number of sensors allowed from certain sets or groups. We refer to these constraints as grouping constraints and they take the following general form:

$$g_{min} \leq \sum_{i \in L_g} s_i \leq g_{max} \tag{5-12}$$

where:

- L_g is a subset of all candidate sensors
- s_i is a binary variable that will be 1 if sensor i is selected, and 0 otherwise
- g_{min} is the minimum number of sensors that must be selected from the subset L_g
- g_{max} is the maximum number of sensors that may be selected from the subset L_g

Grouping constraints can be used to ensure that an optimal sensor placement follows required policies or meets practical limitations. For example, one might want to determine the optimal sensor placement, while also ensuring that there is at least one sensor in every 10 m x 10 m subvolume of the space. This can be formulated by defining sensor subsets L_g containing the candidate sensors within each subvolume and adding a grouping constraint over each of these subsets with g_{min} set to 1.

Another example where grouping constraints might be used is when you have different categories of sensors and you want to make sure that an optimal placement has a certain number of each category. In this case you would define a sensor subset L_g for each category of sensor and then set g_{min} and g_{max} according to how many sensors you want in each category. For example, grouping constraints could be used to ensure that at least one sensor provides visualization to enable more rapid response by repair teams.

There are many other examples of how these grouping constraints can be used in conjunction with the optimal placement formulations described above to give the user more control over the placements returned by the optimizer. However, it should be noted that while there is no limit on the number of these constraints that can be added to the problem, it is possible to formulate infeasible optimization problems if these constraints are not formulated carefully.

5.5 Sensor Failure

For some applications the probability of sensor failure can be higher than desired (e.g, because of the nature of the detection technology, turbulence in the environment, etc.). There are several ways to adapt the formulations above to consider sensor failure while performing the optimization.

The Coverage formulation includes a parameter that allows additional redundancy. For example, you can place sensors such that an entity is not counted as detected unless at least 3 sensors can detect the scenario. This is a straightforward approach for adding redundancy to the solution and ensuring increased robustness in the face of potential sensor failure.

Other formulations are possible that explicitly consider the probability of sensor failure in the problem formulation and use this probability formally when computing the expected value in the objective function [1]. These formulations can be significantly more difficult to solve. Including a constraint that describes the probability of detection across the selected sensors introduces nonlinearity and shifts the problem class from a mixed-integer linear programming (MILP) problem to a mixed-integer nonlinear programming problem (MINLP). Off-the-shelf tools exist that can efficiently solve large-scale MILP problems, however, given the increased challenges in MINLP, tailored, problem specific solution strategies are often required to obtain solutions.

6 OPEN SOURCE SOFTWARE

As part of this project, open source software was developed to make the sensor placement optimization framework available to a wider audience. The software is built on Sandia developed R&D100 award winning Pyomo software [4] and can be used to design sensor networks for several objectives. The software, called Chama [2], is copyright through National Technology & Engineering Solutions of Sandia (SCR# 2146.0) and released under a revised BSD license. The software is hosted on Sandia National Laboratories GitHub site at https://github.com/sandialabs/chama. More information on Chama is available online at http://chama.readthedocs.io.

6.1 Chama

Chama is a Python package which includes mixed-integer linear programming (MILP) formulations to determine sensor locations and technology that maximize monitoring effectiveness. The optimization formulations in Chama can be used to design sensor networks that minimize impact (e.g. time to detection, plume extent, population impacted) or maximize coverage (e.g. detection of possible scenarios, detection of geographic regions). Chama includes functionality to define point and camera sensors that can be stationary or mobile. Furthermore, results from third party simulation tools can be imported to integrate site specific conditions and uncertainty. The software is intended to be used by regulatory agencies, industry, and the research community to design sensor networks and understand the tradeoff between sensor technologies.

The methods in Chama are general and can be applied to a wide range of applications. Several applications are described in Section 7, including placing sensors to monitor methane emissions, water quality, surveillance, and nuclear incidents. For each application, an appropriate model must be selected to represent the system. For example, atmospheric dispersion models can be used to determine optimal sensor placement to monitor oil and gas emissions, while water distribution system models can be used to determine optimal sensor placement to monitor drinking water quality.

Chama relies on additional open source Python packages, including Sandia developed optimization modeling software Pyomo [4] and the data analytics software Pandas [23]. In addition to Python package dependencies, a Pyomo supported mixed-integer programming solver is required to solve the optimization problems. Example solvers that meet this requirement include GLPK [24], Gurobi [25], and CPLEX [26]. Chama is installed using standard Python package installation tools and Python distributions, such as Anaconda, can be used to manage the Python environment.

The basic steps required for sensor placement optimization using Chama are shown in Figure 8 and described below.

Figure 8. Basic steps in sensor placement optimization using Chama.

1. Simulation: Generate an ensemble of simulations representative of the system in which sensors will be deployed.
2. Sensor technology: Define a set of feasible sensor technologies, including stationary and mobile sensors, point detectors and cameras along with their location and other important properties.
3. Impact assessment: Extract the impact of detecting each simulation with each candidate sensor.
4. Optimization: Optimize sensor location and type given a sensor budget.
5. Graphics and post-processing: Generate maps of the site that include the optimal sensor layout and information about scenarios that were and were not detected.

The user can enter the workflow at any stage. For example, if the impact assessment was determined using other means, Chama can still be used to optimize sensor placement. The real benefit of Chama is that it provides a standard workflow for application-agnostic sensor placement problems. Instead of implementing one-off code for each new application, Chama generalizes common routines and facilitates the use of sensor placement formulations through a series of distinct modules with standardized data requirements.

6.1.1 Simulation

Chama uses an ensemble of simulations to extract the data needed for sensor placement optimization. In general, these simulations are run by third party simulation tools, with the exception of Gaussian plume and Gaussian puff dispersion models which are embedded in the software. The type of simulation needed for sensor placement depends on several factors including the application, scale of interest, and the sensor placement objective. In many cases, multiple scenarios should be generated to capture uncertainty in the system. Steady state or transient simulations can be used. While transient simulations are required to minimize time to detection, steady state simulations are sufficient to maximize scenario coverage. Figure 9 illustrates example simulations that can be used in sensor placement optimization for very different applications.

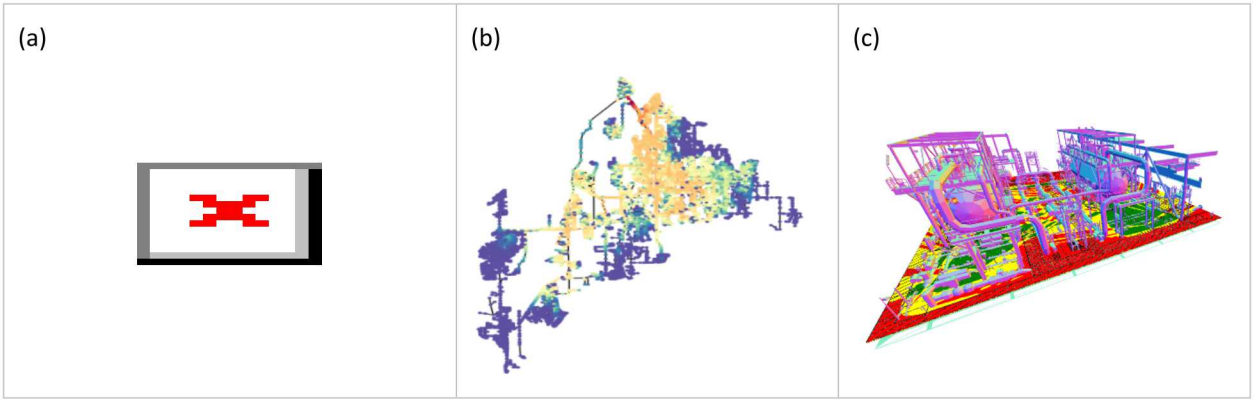


Figure 9. Example simulations that can be used for sensor placement optimization, including (a) atmospheric dispersion simulation showing methane emissions from a well pad (see Section 3.1),

(b) water contamination scenario in a water distribution network [27], and (c) fire scenario in a chemical facility [28].

For each scenario, the time, location, and signal of interest (e.g., methane concentration) are recorded and loaded into Chama. Several options exist to load data in different formats, including from text and Excel files. The points used to record time and location can be sparse to help reduce data size. Example simulation data is shown in Figure 10. In "XYZ format", the X, Y, and Z location and T time is stored for each entry. In "Node format", a Node index and T time is stored for each entry. Node format is particularly useful for networks and sparse systems when the physical location of the node is not needed to extract sensor detection. For both XYZ and Node format, multiple scenarios (S1, S2, S3, ….) are included.

(a)

	X	Y	Z	T	S1	S2	S3
0	1	1	1	0	0.00	0.00	0.00
1	1	1	1	10	0.00	0.00	0.01
2	1	1	1	20	0.00	0.00	0.00
3	2	1	1	0	0.25	0.21	0.20
4	2	1	1	10	0.32	0.14	0.25
5	2	1	1	20	0.45	0.58	0.61
6	1	2	1	0	0.23	0.47	0.32
7	1	2	1	10	0.64	0.12	0.15
8	1	2	1	20	0.25	0.54	0.24
9	2	2	1	0	0.44	0.15	0.45
10	2	2	1	10	0.25	0.28	0.68

...

(b)

	Node	T	S1	S2	S3
0	n1	0	0.00	0.00	0.00
1	n1	10	0.32	0.14	0.25
2	n1	20	0.25	0.54	0.24
3	n2	0	0.00	0.00	0.01
4	n2	10	0.45	0.58	0.61
5	n2	20	0.44	0.15	0.45
6	n3	0	0.00	0.00	0.00
7	n3	10	0.23	0.47	0.32
8	n3	20	0.25	0.28	0.68
9	n4	0	0.25	0.21	0.20
10	n4	10	0.64	0.12	0.15

...

Figure 10. Example simulation data using (a) XZY format and (b) Node format.

6.1.2 Sensor Technology

In Chama, sensors are defined by specifying a position and a detector. The following position options are available:

- Stationary: A stationary sensor is fixed at a single location.
- Mobile: A mobile sensor moves according to defined waypoints and speed. It can also be defined to repeat its path or start moving at a particular time. A mobile sensor is assumed to be at its first waypoint for all times before its starting time and is assumed to be at its final waypoint if it has completed its path and the repeat path option was not set. It is important to note that when using mobile sensors, the simulation's timestep must be sufficient to adequately capture the trajectory of the sensor.

The following detector options are available:

- Point: A point detector. This type of detector determines detection by comparing a signal to the detector's threshold.
- Camera: A camera detector using the camera model from [16]. This type of detector determines detection by collecting the signal within the camera's field of view, converting that signal to pixels, and comparing that to the detector's threshold in terms of pixels.

Additional sensor technologies could easily be incorporated. For example, the research team is working on ray tracing detectors, which detect anything within a camera's field of view or laser's line of sight. In Chama, every defined sensor includes a function that accepts a signal and returns the subset of that signal that is detected by the sensor. This information is then used to extract the impact of each sensor on each scenario, as described below. The sensor placement optimization uses this information to select sensors. Example code used to define a stationary point sensor is shown in Figure 11.

```
pos1 = chama.sensors.Stationary(location=(1,2,3))
det1 = chama.sensors.Point(threshold=0.001, sample_times=[0,2,4,6,8,10])
stationary_pt_sensor = chama.sensors.Sensor(position=pos1, detector=det1)
```

Figure 11. Example code used to define a stationary point sensor.

6.1.3 Impact Assessment

Impact assessment extracts information about detectability for each scenario-sensor combination. This information is used in sensor placement optimization to minimize impact or maximize coverage. The ability for a sensor to detect a scenario depends on several factors, including the scenario environmental conditions, sensor location, and sensor operating parameters. While some scenarios might be detected multiple times by a single sensor, other scenarios can go undetected by all sensors. Impact assessment starts by extracting the times when each sensor detects a scenario. Detection time is defined as the elapsed time from the start of the simulation. Detection times can be converted into other impact and coverage metrics to be used in sensor placement optimization. Example code to extract detection times, statistic on the detection times, and convert detection times to coverage is shown in Figure 12. Example data, including detection times, detection time statistics, and scenario coverage, is shown in Figure 13.

```
det_times = chama.impact.extract_detection_times(signal, sensors)
det_time_stats = chama.impact.detection_time_stats(det_times)
scenario_cov = chama.impact.detection_times_to_coverage(det_times)
```

Figure 12. Example code used to extract impact and coverage metrics.

(a)

	Scenario	Sensor	Detection Times
0	S1	A	[30]
1	S1	B	[30]
2	S1	C	[10, 20, 30, 40]
3	S2	A	[10, 20, 30]
4	S2	B	[20, 30]
5	S2	C	[10, 20, 30, 40]
6	S3	A	[20, 30]
7	S3	B	[20, 30]
8	S3	C	[20, 30, 40]

(b)

	Scenario	Sensor	Min	Mean	Median	Max	Count
0	S1	A	30	30.0	30.0	30	1
1	S1	B	30	30.0	30.0	30	1
2	S1	C	10	25.0	25.0	40	4
3	S2	A	10	20.0	20.0	30	3
4	S2	B	20	25.0	25.0	30	2
5	S2	C	10	25.0	25.0	40	

(c)

	Sensor	Coverage
0	A	[S1, S2, S3]

```
1       B   [S1, S2, S3]
2       C   [S1, S2, S3]
```

```
4
6          S3        A     20   25.0      25.0     30
2
7          S3        B     20   25.0      25.0     30
2
8          S3        C     20   30.0      30.0     40
3
```

Figure 13. Example metrics showing (a) detection times, (b) detection time statistics including minimum detection time, and (c) scenario coverage.

6.1.4 Optimization

Chama contains impact and coverage sensor placement optimization formulations along with the ability to include grouping constraints. The formulations are written in Pyomo [4] and solved using an open source or commercial solvers [24, 25, 26]. The open source GLPK solver is used by default. Additional optimization formulations could be added. The sensor failure formulation, described in Section 5.5, has not been added to Chama due to the nonlinearity introduced from the probability of detection. This could be added at a later time.

The Impact formulation is used to determine optimal sensor placement and type that minimizes impact, where impact can be the sensor's detection time or some other measure of damage. The Coverage formulation is used to place sensors that maximize the coverage of a set of entities, where an entity can be a scenario, the average amount of time each scenario is detected, or geographic location. Example code used to run sensor placement optimization is shown in Figure 14. In this example, the Impact formulation is used to place 200 sensors that minimize detection time. Note that sensor budget can be given in dollars and the cost of individual sensors can be specified.

```
impactform = chama.optimize.ImpactFormulation()
results = impactform.solve(impact=min_det_time, sensor_budget=200)
```

Figure 14. Example code used to run sensor placement optimization.

To use the Impact formulation, the user defines an impact assessment, sensor budget, and (optionally) sensor cost and the scenario probability. This input is described below:

- Impact assessment: A single value of impact (detection time or other measure of damage) for each sensor that detects a scenario.
- Sensor budget: The number of sensors to place, or total budget for sensors.
- Sensor characteristics: Sensor characteristics include the cost of each sensor.
- Scenario characteristics: Scenario characteristics include scenario probability and the impact for scenarios that go undetected. When minimizing detection time, the undetected impact value can be set to a value larger than time horizon used for the study. Individual scenarios can be given different undetected impact values.

To use the Coverage formulation, the user defines coverage, sensor budget, and (optionally) sensor cost and the entity weights, as described below:

- Coverage: A list of entities that are covered by each sensor.
- Sensor budget: The number of sensors to place, or total budget for sensors.
- Sensor characteristics: Sensor characteristics include the cost of each sensor.
- Entity characteristics: Entity characteristics include entity weights.

Results from sensor placement optimization include the following information:

- A list of selected sensors.
- The expected (mean) impact or coverage based on the selected sensors.
- The fraction of scenarios or entities that were detected.
- Total cost of the selected sensors.
- Assessment of the selected sensor placement. This information can be used to extract particular scenarios or entities that are difficult to detect and the contribution of individual sensors.

6.1.5 Graphics and Post-Processing

Chama includes several graphics and post-processing methods to analyze and compare sensor placements. Simulation data can be visualized, as shown in Figure 15. The figure shows cross-sections of three Gaussian plume scenarios along the X-Y, X-Z, and Y-Z plane. In this case, concentration is shown on a log-scale. This type of visualization is useful to verify that the simulation data was generated and loaded as expected, compare scenarios, and to better understand optimal sensor placement. The position of fixed and mobile sensors can also be visualized, as shown in Figure 16. This type of visualization is useful to view the location of feasible sensors and to view the location of optimal sensor placement.

After running a series of sensor placement optimizations with increasing sensor budget, a figure can be generated using the objective value and the fraction of detected scenarios for each placement. Figure 17a compares the expected time to detection and scenario coverage as the sensor budget increases. The impact of individual scenarios can also be analyzed using the assessment that is returned from the optimization. Figure 17b compares impact values from several scenarios, given an optimal placement. These types of figures are created using standard Python tools.

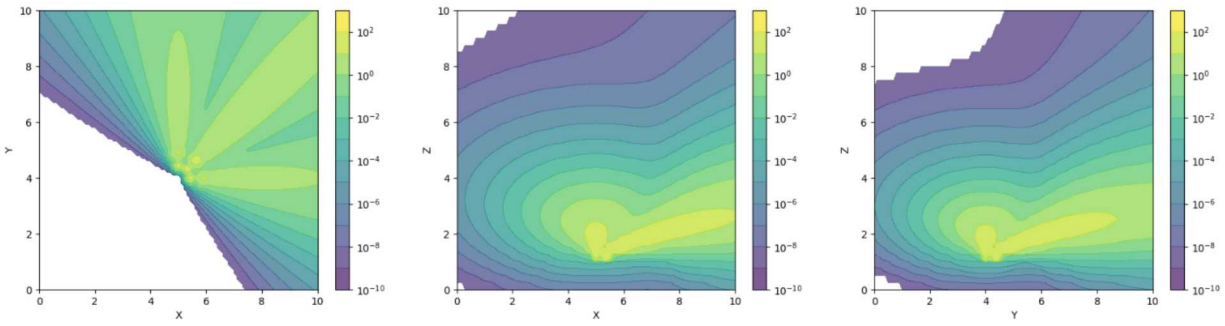


Figure 15. Cross-sections of multiple emission scenarios.

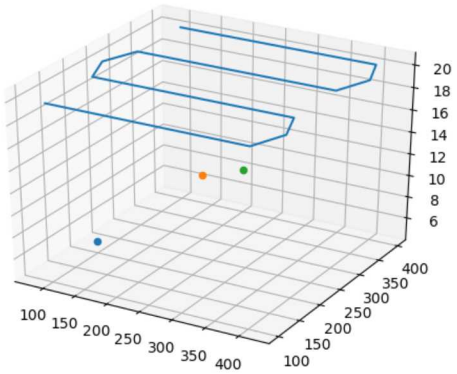


Figure 16. Sensor locations of stationary and mobile sensors.

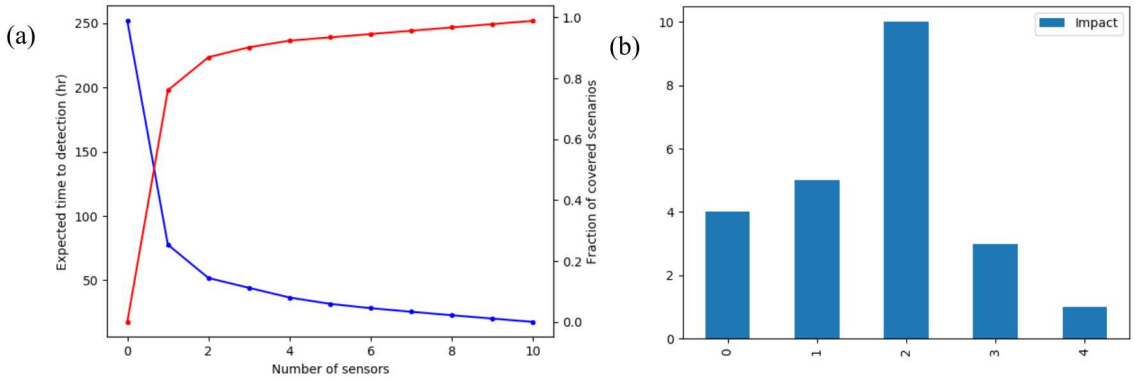


Figure 17. (a) Sensor placement optimization results showing expected time to detection and fraction of covered scenarios as the number of sensors increases and (b) impact values per scenario based on an optimal sensor placement.

6.2 Software Quality Assurance

Chama is maintained in a version controlled software repository hosted on Sandia Laboratories GitHub organization at https://github.com/sandialabs/chama. Chama includes continuous integration software tests that are run automatically each time changes are made to the repository. The tests cover a range of unit and integration tests designed to ensure that the code is performing as expected. The software includes an online user manual that is automatically rebuilt each time changes are made to the code. The documentation is publicly available at http://chama.readthedocs.io and as a SAND report [2]. The user manual includes an overview, installation instructions, simple examples, and information on the code structure and functions. Feature requests, issues, and software contributions can be submitted to the GitHub website. All contributions are expected to follow standard software practices, including documentation and testing. The core development team reviews all software changes and prioritizes feature requests and issues.

7 APPLICATIONS

This section describes several case studies that use Chama for sensor placement optimization, including placing sensors to monitor methane emissions, water quality, surveillance, nuclear incidents, and process safety.

7.1 Methane Emissions

This case study demonstrates how Chama is used to design sensor networks for methane production facilities. Here, we consider a hypothetical 100 m by 100 m by 10 m region in Fort Worth, Texas and use emission rates and weather conditions [29, 30] from the area to simulate leak scenarios from 10 leak locations. For the sensor network, we consider a grid of candidate sensor locations and three types of sensors with different detection thresholds and costs. Figure 18 shows a 2-dimensional site map for this study, where black Xs are the potential leak locations and the blue circles are the candidate sensor locations.

The main goal when generating scenarios for sensor placement is to accurately capture the full range of conditions that the sensors are likely to encounter when deployed. Therefore, the leak scenarios used in this case study consider a range of wind conditions, leak rates, and leak locations. We use the Gaussian plume dispersion model built into Chama to simulate methane concentrations over the entire region caused by a single leak source over a 1 day period. We divide our region into 1 m^3 cells and our time period into 1 hour time steps resulting in 2,400,000 data points for each simulation. Each simulation updates the wind speed and direction every hour using data from NOAA for Fort Worth in 2015 [29]. At this site, wind is predominantly from the South. In order to account for wind conditions seen throughout the year, we randomly selected 1 representative day from each month. Leak rates were selected from a distribution published in the Fort Worth Natural Gas Air Quality Study [30]. Sampling from this distribution ensured that we had a realistic proportion of large and small leaks in our collection of scenarios. Finally, the leak height was selected from a uniform distribution between 0 and 2 m. Ten realizations, with different leak rate and height, were generated for each day and leak location. This resulted in 1200 leak scenario simulations.

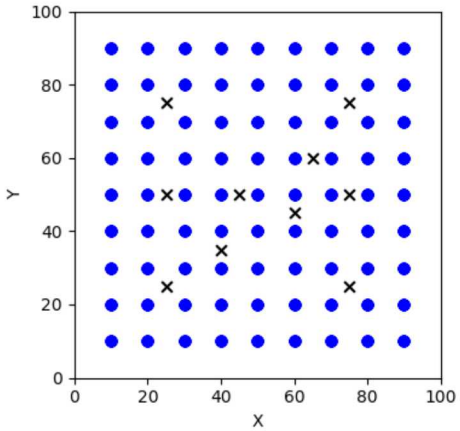


Figure 18. 2D site map, black X's are the potential leak locations, blue circles are candidate sensor locations.

Three types of candidate sensors were included in this case study to determine the optimal sensor placements at different price points. Table 3 lists the sensor characteristics for the three types of sensors. A regular grid is used to define the candidate sensor locations with 10 m spacing in the x and y directions and 1 m spacing in the z direction. This results in 2430 candidate sensor locations, 810 in each sensor category.

Table 3. Sensor characteristics

Category	**Detection Threshold**	**Cost**
Low resolution/cost	1 ppm	$1,000
Moderate resolution/cost	0.1 ppm	$10,000
High resolution/cost	0.01 ppm	$100,000

For each leak simulation and sensor detection threshold, Chama is used to extract detection times. These detection times are then used to optimize sensor placement. For this case study, we use the Coverage formulation in Chama to select sensors to maximize the coverage over the set of leak scenarios; where "coverage" is defined as a selected sensor detecting a scenario at any time. Of the 1200 scenarios used in this study, 67 of them were not detected by any of the candidate sensors. This is important to note as we will compare sensor placements in terms of the fraction of scenarios they detect. The upper bound on the fraction of detected scenarios given the candidate sensor layout is (1200-67)/1200 = 0.94.

We first look at optimal sensor placements for maximizing scenario coverage where the sensor budget is the number of sensors. Figure 19 shows results for the three categories of candidate sensors when they are considered separately. Each point on the plot is the result of solving an optimization problem with the indicated sensor budget and where all scenarios are equally weighted in the objective function. Note that the high sensitivity sensors detect ~70% of the scenarios with 10 sensors while the moderate and low sensitivity sensors detect far fewer scenarios. A 2D projection of the optimal sensor placements with 10 sensors is shown in Figure 20 for each category of sensor. Notice that the high sensitivity sensors are placed further from the leak locations in the dominant wind direction compared to the moderate and low sensitivity sensors which are placed very close to the leak locations. Also notice that the placements do not follow the heuristic of placing a sensor close to every leak location; which might be the first placement strategy an operator or regulator would consider.

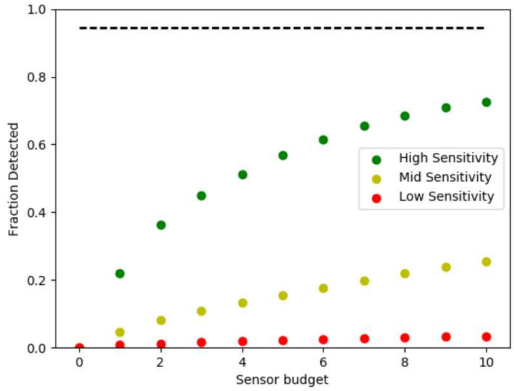


Figure 19. Fraction of scenarios detected as the number of sensors increases when considering one sensor category at a time. The dashed black line is an upper bound on the detection fraction.

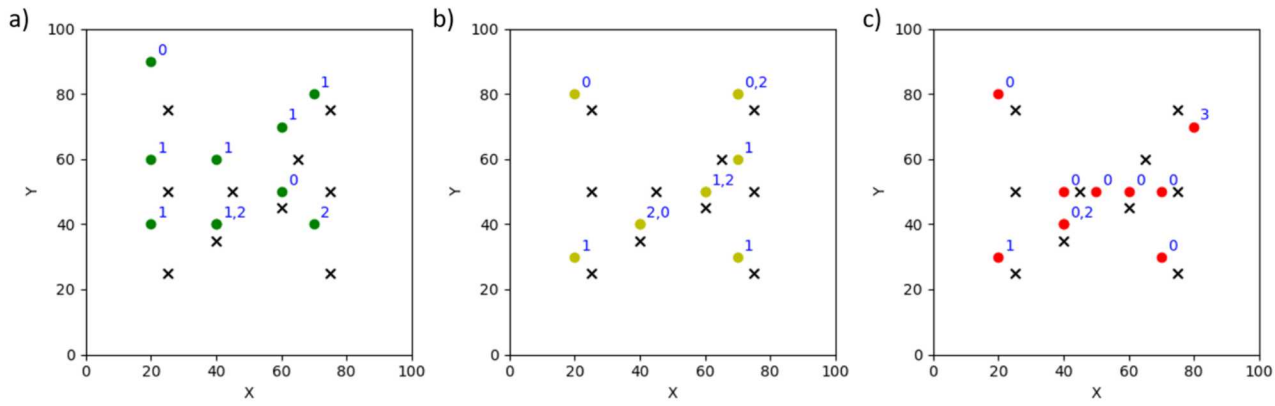


Figure 20. Optimal sensor placements with 10 sensors. Black X's are the leak locations, circles are the selected sensors, and blue text gives the height of each sensor (a) placement with high sensitivity sensors (b) placement with moderate sensitivity sensors (c) placement with low sensitivity sensors.

Additional results for this case study can be found in [3] including a comparison of sensor placements when using sensor costs, comparison with heuristic-based placements, and validation with scenarios sets that were not used in the optimization. The main takeaway from this study is that the optimal sensor placement is not always intuitive and there are advantages to using the mathematical formulations.

7.2 Water Distribution Contamination

The Impact formulation has been used extensively for security applications in water distribution systems [19]. Sandia National Laboratories has embedded these methods in software tools developed for the U.S. Environmental Protection Agency [20, 21]. As part of this project, these methods were extended and made available in the software package Chama. The updated software is more modular and offers a wider set of optimization and analysis options to the user. When porting these methods into the new software package, the research team developed case studies to make sure that the new software could be used for water security applications, as described below.

This case study demonstrates how Chama is used to design sensor networks for water quality applications. The Water Network Tool for Resilience (WNTR), a Python package developed at Sandia to analyze resilience of water distribution systems [27], was used to model contamination scenarios in a water distribution system and sensor placement optimization was carried out using Chama to minimize the extent of contamination and maximize scenario coverage. The contamination scenarios track contaminant concentrations throughout the water distribution system over a 7-day period, where each scenario injects contaminant at a different location and time. Each node is defined as a feasible sensor location with a detection threshold set to 1e-7 mg/L sampled at 1-hour intervals. The simulation data is used to compute the extent of contamination (pipe feet that are contaminated) at each timestep. The optimization methods are used to place sensors which minimize the extent of contamination or maximize scenario coverage in two separate optimization problems. Figure 21 illustrates results from a series of sensor placement optimizations where the objective is plotted on the Y-axis and the number of

sensors (increasing from 0 to 20) is plotted on the X-axis. The sensor location for a 5-sensor design is shown as inset to each graphic. Note that sensors placed to minimize extent of contamination tends to place sensors along the main distribution lines in the network, sensors placed to maximize scenario coverage tends to place sensors along the leaf nodes. Additional information can be extracted from the sensor placement results, such as scenarios that lead to the highest extent of contamination and sensors that detect the highest number of scenarios, given the optimal sensor layout.

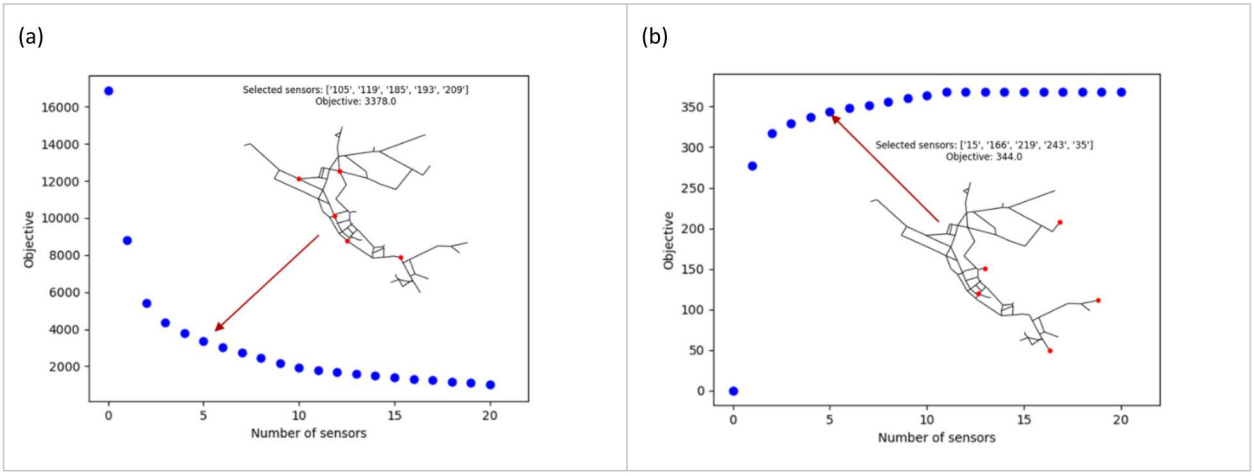


Figure 21. Sensor placement optimization in a water distribution system showing (a) sensors placed to minimize extent of contamination and (b) sensors placed to maximize scenario coverage.

As described in Section 5.4, side constraints can be added to both the Impact and Coverage formulations in order to include additional conditions on the sensors that are selected. These constraints can be used to ensure that an optimal sensor placement follows required policies, requirements, or meets practical limitations. For example, sampling requirements on water distribution systems might require samples to follow geographic constraints, such as ensuring that at least one sample is taken from each geographic region. For water distribution systems, these regions can be based on a regular grid, neighborhoods, pressure zones, or districted metered areas.

Consider the two examples shown in Figure 22. In the figure on the left, the area is divided into a 5x5 grid, and we might want to include constraints to ensure that there are at least N sensors within each of the areas. In the figure on the right, the sets of interest are determined by community clusters (and could be pressure zones, etc.), and again, we may want to ensure there are at least N sensors within each community. Both of these examples can easily be addressed using grouping constraints. Grouping constraints can also be used to include additional properties into the optimization function. Looking again at water distribution systems, we might want to ensure that the distribution of water age from our sensor locations is representative of the distribution of water age across the entire system. We can categorize the sensors according to water age using simulations, and ensure that there are a sufficient number of sensors within each of these categories (i.e., ranges in the histogram of water age distribution). Grouping constraints can again be used in this context.

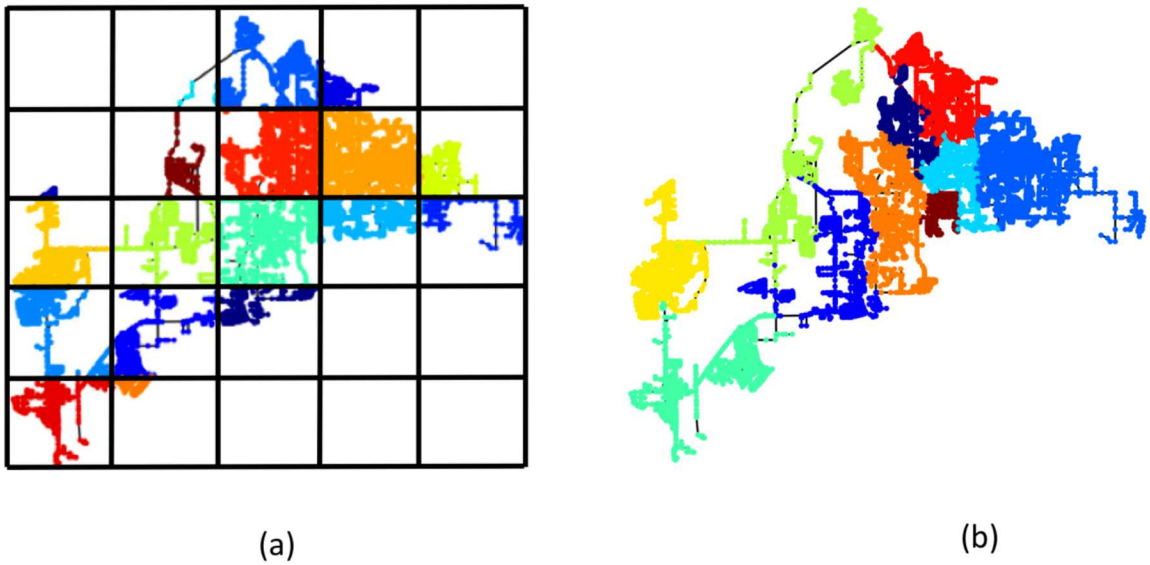


Figure 22: Examples of grouping constraints for water distribution systems: (a) 5x5 grid and (b) community clusters. Both these restrictions can be enforced with grouping constraints to ensure there are at least N sensors within each colored subset.

7.3 Surveillance

The following case study uses the Coverage formulation in Chama to place cameras that maximize geographic coverage. While this example focuses on surveillance, the example could be used for many applications, including placing fire detectors in buildings. In this example, the space under consideration is described by a set of areas or volumes (called entitites in the Coverage formulation). These represent the physical geographic spaces that we would like to observe by our cameras. Then, we define a set of potential cameras from which to select. A camera can be characterized by a location in the physical space, but may also be differentiated by other factors. For example, you may have two potential cameras at the same location, but they may differ in terms of their angle of orientation, technology type, or cost. In Chama, all potential technology options can be represented by distinct sensors.

In this application we need to determine the set of geographic locations (or entities) that can be observed for each potential camera. For example, one can specify that a potential camera has a location, a horizontal and vertical orientation (angle), and a field of view. Then if we divide the region into a set of 2D areas or 3D volumes, we can compute the set of observable areas or volumes for each of the potential cameras. In our example, we calculate this data with ray casting computations. Ray casting accounts for the field of view and physical barriers that determine if a particular camera can see a particular gridded area or volume. For each camera, a set of angles is defined (a discretization of the field of view). For each of these angles, we compute points along the ray from the camera and determine which squares in the 2D grid are observed by the camera along that angle. This excludes squares after the ray hits a solid (wall). The complete set of observable squares for each camera is used to form the input needed for sensor placement optimization in Chama. For other applications, determining the set geographic locations for each

camera could be more complicated and may involve physical simulations regarding the interaction of the sensor with the entity being observed. For example, in the case of fire detection, an accurate analysis may require more detailed modeling of the type and size of the fire, along with modeling the physics of the detector and its ability to observe these fires.

For this example, we consider the 2D space based on the layout in [31], a classic "art gallery" problem, where the goal is to minimize the number of cameras that can observe the whole gallery. Figure 23 shows the 2D space, which is discretized into a 100 by 100 grid, and potential cameras at each of the corners of the area and the walls. For this example, we allow four different camera directions at 45°, 135°, 225°, and 315°, each with a 60° field of view. We then run the ray casting analysis to determine which squares in the 100 by 100 grid are viewable by each potential camera location and direction. This is used to form the input for the Coverage formulation, and we run the optimization for 0 to 20 cameras.

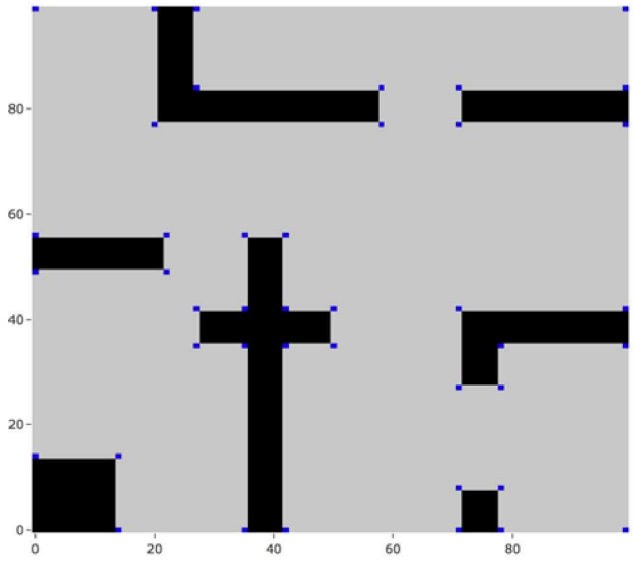


Figure 23. 2D layout of the surveillance example from [31]. Black areas are walls, and blue squares are potential cameras. The goal is to place cameras to maximize the area that can be observed.

Figure 24 shows the optimal area observed as a function of the number of cameras allowed in the optimization. Note that each of these columns is a separate optimization, maximizing the area covered with a constraint on the number of cameras placed. With 11 cameras, over 95% of the area is observed. Also note that each optimization is independent. That is, the optimal placement for a particular number of cameras need not contain the same cameras selected with the lower budget. For example, Table 4 shows the optimal placements with a sensor budget of 6 cameras and 7 cameras. This table shows that optimal placement for 7 cameras is not a superset of that for 6 cameras, though there is some overlap. This means that the optimization is finding results that would not be possible using a greedy algorithm that focused on adding one additional sensor at each iteration.

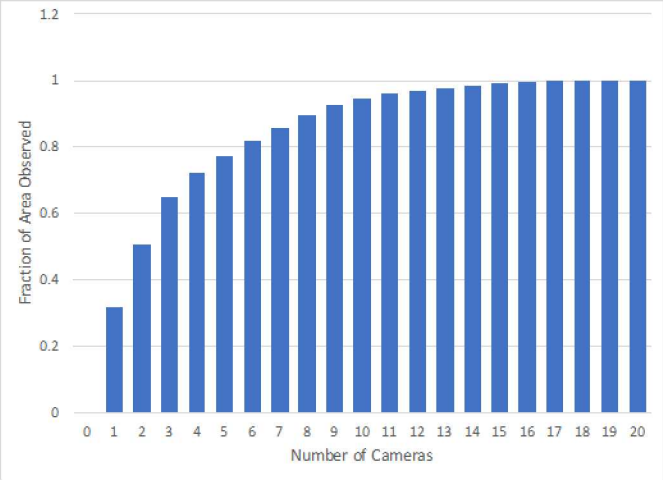


Figure 24. Optimal area observed as a function of the number of cameras placed. Here, 95% of the area is observed with 11 cameras.

Table 4. Placement results for 6 and 7 cameras.

Optimal solution placing 6 cameras	Optimal solution placing 7 cameras
(0, 49, 285, 345)	(0, 49, 285, 345)
(0, 99, 285, 345)	-
-	(35, 42, 105, 165)
(42, 0, 15, 75)	(42, 0, 15, 75)
-	(42, 35, 285, 345)
(99, 0, 105, 165)	(99, 42, 105, 165)
(99, 42, 105, 165)	(99, 77, 195, 255)
(99, 84, 105, 165)	(99, 84, 105, 165)

Figure 25 shows the optimal placement results for 4 sensor placements, using 1, 5, 10, and 15 cameras. The cameras are indicated by small squares (green for cameras selected in the optimization, and blue for cameras that were not selected). Yellow indicates areas that are not visible by any selected camera and green indicates the area that is visible given the selected cameras. We see that 10 cameras covers a large fraction of the area. Figures like this can be used to assess the quality of the optimization solution and, if not satisfactory, weights of individual areas (entities) can be adjusted to encourage observability of particular high importance areas.

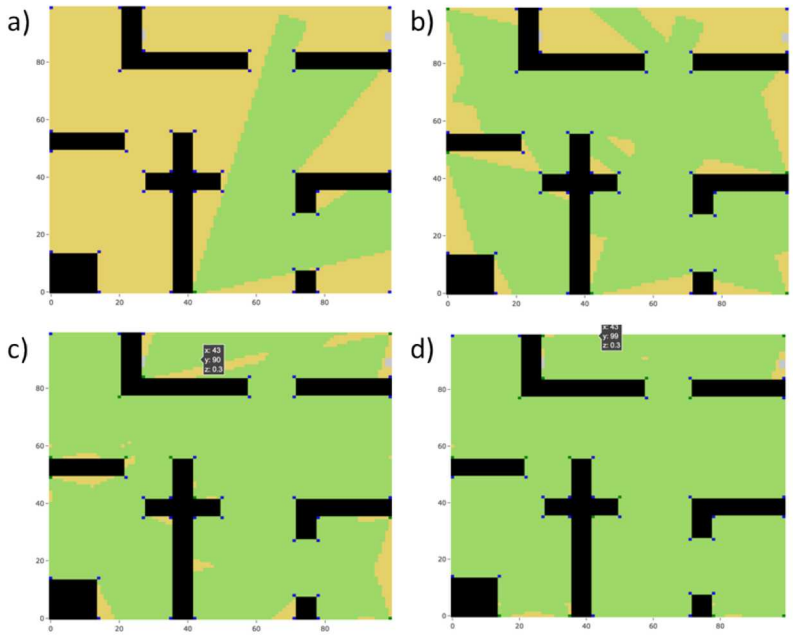


Figure 25. Optimal sensor placement results using (a) 1, (b) 5, (c) 10, and (d) 15 cameras. Yellow indicates areas that are not visible by any selected camera and green indicates the area that is visible given the selected cameras.

7.4 Nuclear Incident Response

The following preliminary case study demonstrates how Chama can be used to design sensor networks to monitor the area surrounding a nuclear reactor. The case study is similar in nature to the Methane Emissions case study in Section 7.1, however, the potential consequence and region of interest for a nuclear incident is much larger. For this case study, the research team partnered with the Sandia MACCS team [32] to integrate nuclear release simulations into sensor placement optimization. The simulations are computed using the Hybrid Single Particle Lagrangian Integrated Trajectory model (HYSPLIT) [9] to model a 1 hour release of nuclear material into the atmosphere. Multiple scenarios are generated to account for uncertainty in wind conditions throughout the year. For each scenario, nuclear material is transported and deposited over the course of several days due to local wind conditions. Figure 26 illustrates a single nuclear release scenario at a single timestep. The source is in the center of the figure and the plume extends 60 km to the northeast.

In this case study, a subset of the HYPLIT simulations is used to demonstrate the use of nuclear release simulations in sensor placement optimization. 50 scenarios are selected at random and only surface concentrations are used. Candidate sensors locations are defined on a regular grid surrounding the reactor with 5 km spacing. The candidate sensors cover the region which is 5 to 50 km from the source. This setup, where candidate sensors are not defined as the source, is appropriate for “fence-line” monitoring or monitoring by third parties that do not have access to the site. The Coverage formulation in Chama is then used to select sensors that maximize the average amount of time each scenario is detected. Figure 27 shows the fraction of time scenarios are covered as a function of the number of sensors that are optimally placed in the surrounding region. Even with 100 sensors, the average amount of time each scenario is detected is 60%. Example sensor placements are shown in Figure 28.

While this case study demonstrates the use of complex large scale dispersion simulations within the optimization framework, this case study is preliminary. More research is needed to determine the best use of sensor placement optimization for this application and to integrate the entire simulation data set into the optimization framework. This study uses simulations from a single reactor, and places sensors in the surrounding area to maximize the chance of detecting nuclear material. In reality, numerous reactors exist across the US, and a similar analysis could integrate multiple sources and design a sensor network to monitor a larger region. This type of analysis would help identify reactors that require extra monitoring stations and scenarios that are difficult to detect even with high sensitivity sensors. The optimal sensor locations could be evaluated using different scenario sets and compared to random or existing placements to determine the benefit of optimal placement.

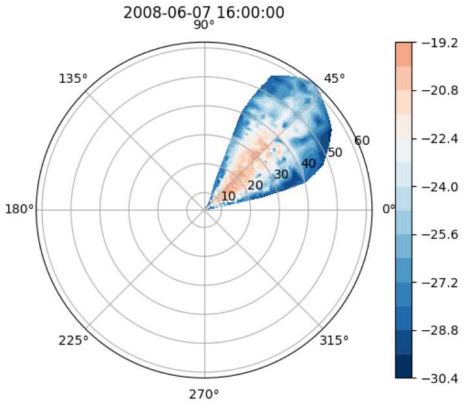


Figure 26. Nuclear release simulation using HYSPLIT.

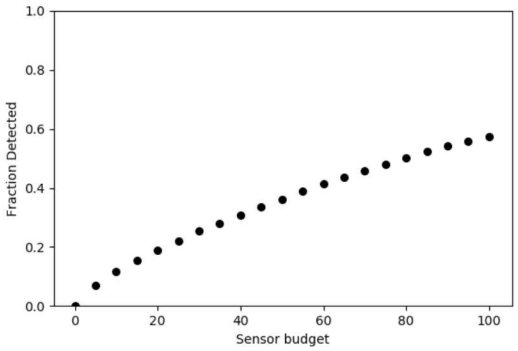


Figure 27. Fraction of time each scenario is covered as a function of the number of sensors.

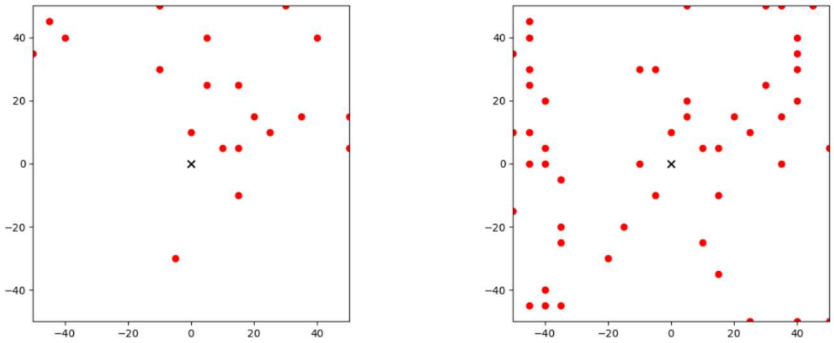


Figure 28. Sensor locations in a 20 (left) and 50 (right) sensor network. The red circles are the sensor locations and the black X is the reactor.

7.5 Process Safety

Chemical process facilities, such as refineries and oil and gas production sites, integrate process safety into their operations to prevent fires, explosions and accidental chemical releases. The physical placement of fire and gas detectors within these facilities can have a large impact on the ability to detect hazards. As part of this project, a CRADA was signed with Kenexis to advance simulation and optimization capabilities to improve process safety and assurance in chemical facilities. Kenexis is a leader in reliability engineering for process safety management. Kenexis has developed state-of-the-art simulation software, called Effigy [28], that maps fire and gas coverage based detailed site geometry, sensor models, and hazard scenarios. Figure 29 shows an example Effigy simulation of a process facility with fire scenarios and sensor coverage. As part of this project, the research team worked with Kenexis to integrate Effigy with Chama to provide optimal sensor placement for fire and gas detectors. For this integration, Effigy is used to model the site, define sensors and potential hazards, and extract information about sensor detection. This impact assessment is then reformatted to load into Chama, which runs geographic and scenario coverage and passes the optimal placement back to Effigy. This workflow is currently being tested and refined.

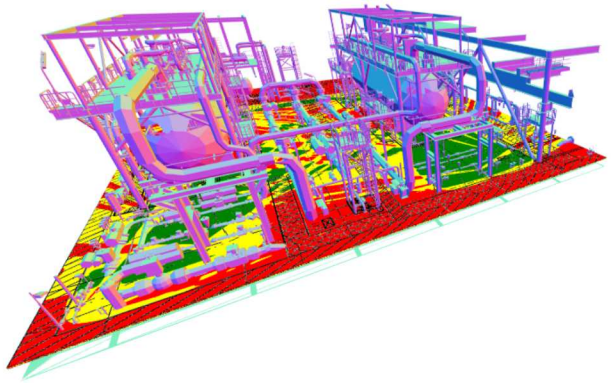


Figure 29. Fire scenario in a chemical facility [28].

8 CONCLUSIONS

This document summarizes research performed under the Laboratory Directed Research and Development (LDRD) project titled Developing Fugitive Emissions Sensor Networks: New Optimization Algorithms for Monitoring, Measurement and Verification. The purpose of this project is to develop methods and software to enhance detection programs through optimal design of the sensor network. Sensor placement optimization is an important field of research given the inherent variability of sensor performance based on environmental variables, sensors location, and the sensors operating conditions. Furthermore, sensor networks need to work together to maximize effectiveness and avoid collection of duplicate information. Different monitoring objectives may require different sensor network designs. Given the high cost to deploy sensors in the field, stakeholders need methods to help evaluate these decisions.

This project included field work to test detection technology, research on atmospheric dispersion modeling, and software development for sensor placement optimization. The research team completed three weeks of field work to evaluate sensor technology parameters including the ability to identify leak location, rate of false positives and false negatives, leak quantification accuracy, and the ability to resolve two closely spaced leak. This work defines empirical relationships for detection probability and leak identification for different sensor technologies. The results can be used to develop sensor models and identify technology attributes that are best suited for monitoring methane emissions. The data from this field work will be made publicly available.

Simple Gaussian plume dispersion models are commonly used to model air quality, however, these simplistic models do not account for effects of local turbulence and atmospheric variation that is seen in the field. Because optimization workflows typically require many thousands of simulations, complex simulation methods, like CFD, are often too computationally expensive for practical use. In this project, the research team developed a modified Gaussian plume dispersion model (m-GPD) that builds on the Gaussian plume dispersion model (GPD) and incorporates turbulent “clumping” effects on plume concentration profiles at a parametric level. The m-GPD strikes a balance between the simplistic GPD that is not applicable at many oil and gas facilities and other complex models that require extensive parameterization and computation time. The m-GPD model can be readily integrated into simulation and optimization frameworks for improved sensor performance characterization.

The physical placement of sensors, along with the sensor technology and operating conditions, can have a large impact on our ability to monitor a site. In this project, the research team developed a framework to determine optimal sensor placement and sensor technology to design sensor networks. In an effort to make this framework available to a wide audience, the research team released open source software which is accessible through the Sandia Laboratories GitHub site. The software package, called Chama, provides a standard workflow for application-agnostic sensor placement problems. While this project focused on methane emissions, the software can be applied to a wide range of applications, as outlined in this report. Instead of generating one-off code for each new application, the use of Chama reduces implementation time by allowing the user to take advantage of common routines, standard data formats, and tested sensor placement formulations. While Chama leverages sophisticated data science and optimization packages (e.g. Pandas and Pyomo), the software is packaged in a way that simplifies these routines. While many industry experts integrate simulation and analysis tools into their workflow, optimization is not always included. Chama allows users to quickly include

optimization in their workflow. For systems with complex geometry and complex interactions between simulations and sensors, integrating optimization can facilitate critical insight about the placement of sensors and the sensor operating conditions. The research team will continue to use the software for related projects and add features as needed. The software can be extended to include additional sensor placement optimization formulations, including methods to integrate sensor failure, sampling frequency, adaptive sensor networks, manual sampling, and additional sensor models.

9 REFERENCES

[1] J. Liu and C. Laird, "A global stochastic programming approach for the optimal placement of gas detectors with nonuniform unavailabilities," *Journal of Loss Prevention in the Process Industries,* vol. 51, pp. 29-35, 2018.

[2] K. Klise, B. Nicholson and C. Laird, "Sensor Placement Optimization using Chama," Sandia National Laboratories, SAND2017-11472, 2017.

[3] K. Klise, B. Nicholson, C. Laird, A. Ravikumar and A. Brandt, "Sensor Placement Optimization for Site-scale Methane Emissions," *submitted to Environmental Science and Technology,* 2018.

[4] W. Hart, C. Laird, J. Watson, D. Woodruff, G. Hackebeil, B. Nicholson and J. Siirola, Pyomo – Optimization Modeling in Python, Volume 67 of Springer Optimization and Its Applications, Springer Science & Business Media., 2017.

[5] U.S. Environmental Protection Agency, "Oil and Natural Gas Sector: Emission Standards for New and Modified Sources," Federal Registry, 80(181), 40 CFR Part 60, 2005.

[6] R. Alvarez, D. Zavala-Araiza, D. Lyon, D. Allen, Z. Barkley, A. Brandt, K. Davis, S. Herndon, D. Jacob, A. Karion and E. Kort, Assessment of methane emissions from the US oil and gas supply chain, New York, NY: Science, 2018.

[7] X. Liu, S. Cheng, H. Liu, S. Hu, D. Zhang and H. Ning, "A survey on gas sensing technology," *Sensors,* vol. 12, no. 7, pp. 9635-9665, 2012.

[8] W. Skamarock, J. Klemp, J. Dudhia, D. Gill, D. Barker, M. Duda, X. Huang, W. Wang and G. Powers, "A Description of the Advanced Research WRF Version 3," National Center for Atmospheric Research, Technical Note NCAR/TN-475+STR, Boulder, Colorado, 2008.

[9] A. Stein, R. Draxler, G. Rolph, B. Stunder, M. Cohen and F. Ngan, "NOAA's HYSPLIT Atmospheric Transport and Dispersion Modeling System," *American Meteorological Society,* vol. 96, no. 12, pp. 2059-2077, 2015.

[10] M. Krol, S. Houweling, B. Bregman, M. van den Broek, A. Segers, P. van Velthoven, W. Peters, F. Dentener and P. Bergamaschi, "The two-way nested global chemistry-transport zoom model TM5: algorithm and applications," *Atmospheric Chemistry and Physics,* vol. 5, no. 2, pp. 417-432, 2005.

[11] U.S. Environmental Protection Agency, "User's Guide for the AMS/EPA Regulatory Model (AERMOD)," Technical Report, EPA-454/B-18-011, 2018.

[12] J. Scire, D. Strimaitis and R. Yamartino, "A User's Guide for the CALPUFF Dispersion Model (Version 5)," Earth Tech, Inc., Concord, MA, 2000.

[13] M. Williams, M. Brown, B. Singh and D. Boswell, "QUIC-PLUME Theory Guide," Los Alamos National Laboratory, LA-UR-04-0561, 2004.

[14] A. Green, R. Singhal and R. Venkateswar, "Analytic Extensions of the Gaussian Plume Model," *Journal of the Air Pollution Control Association,* vol. 30, pp. 773-776, 1980.

[15] M. Beychok, Fundamentals of Stack Gas Dispersion, Milton R Beychok; 4th edition, 2005.

[16] A. Ravikumar, J. Wang and A. Brandt, "Are Optical Gas Imaging Technologies Effective For Methane Leak Detection?," *Environmental science and technology,* vol. 51, no. 1, pp. 718-724, 2016.

[17] N. Otsu, "A Threshold Selection Method from Gray-level Histograms," *Automatica,* vol. 11, pp. 285-296, 1975.

[18] T. Atherton and D. Kerbyson, "Size Invariant Circle Detection," *Image and Vision Computing,* vol. 17, no. 11, pp. 795-803, 1999.

[19] J. Berry, W. Hart, C. Phillips, J. Uber and J. Watson, "Sensor Placement in Municipal Water Networks with Temporal Integer Programming Models," *Journal of Water Resources Planning and Management,* vol. 132, no. 4, pp. 218-224, 2006.

[20] U.S. Environmental Protection Agency, "TEVA-SPOT Toolkit User Manual," Technical Report, EPA/600/R-08/041B, 2012.

[21] U.S. Environmental Protection Agency, "Water Security Toolkit User Manual," Technical Report, EPA/600/R-14/338, 187p, 2015.

[22] S. Legg, A. Benavides-Serrano, J. Siirola, J. Watson, S. Davis, A. Bratteteig and C. Laird, "A Stochastic Programming Approach for Gas Detector Placement Using CFD-Based Dispersion Simulations," *Computers and Chemical Engineering,* vol. 47, pp. 194-201, 2012.

[23] W. McKinney, Python for Data Analysis: Data Wrangling with Pandas, NumPy, and IPython, Sebastopol, CA: O'Reilly Media, 2012.

[24] A. Makhorin, "GLPK (GNU Linear Programming Kit)," [Online]. Available: https://www.gnu.org/software/glpk/. [Accessed 10 09 2018].

[25] Gurobi Optimization, "Gurobi Optimization," [Online]. Available: http://www.gurobi.com. [Accessed 11 July 2018].

[26] IBM, "CPLEX Optimizer," [Online]. Available: https://www.ibm.com/analytics/cplex-optimizer. [Accessed 11 July 2018].

[27] K. Klise, D. Hart, D. Moriarty, M. Bynum, R. Murray, J. Burkhardt and T. Haxton, "Water

Network Tool for Resilience (WNTR) User Manual," U.S. Environmental Protection Agency, Technical Report, EPA/600/R-17/264, Cincinnati, OH, 2017.

[28] Kenexis, "Effigy," [Online]. Available: https://www.kenexis.com/software/effigy/. [Accessed 24 August 2018].

[29] National Oceanic and Atmospheric Administration, "Automated Surface Observing System (ASOS)," [Online]. Available: https://www.ncdc.noaa.gov/data-access/land-based-station-data/land-based-datasets/automated-surface-observing-system-asos. [Accessed 17 8 2018].

[30] Eastern Research Group and Sage Environmental Consulting, "City of Fort Worth Natural Gas Air Quality Study, Final Report," http://fortworthtexas.gov/uploadedFiles/Gas_Wells/AirQualityStudy_final.pdf, 2011.

[31] Wikipedia, "Art Gallery Problem," [Online]. Available: https://en.wikipedia.org/wiki/Art_gallery_problem. [Accessed 2 December 2017].

[32] Sandia National Laboratories, "MACCS," [Online]. Available: https://maccs.sandia.gov/maccs.aspx. [Accessed 6 9 2018].

DISTRIBUTION

1	MS0359	D. Chavez, LDRD Office	1911
1	MS0735	Erik Webb	8860 (Electronic copy)
1	MS0750	Lori Parrott	8863 (Electronic copy)
1	MS0899	Technical Library	9536 (Electronic copy)
1	MS1124	Peter Kobos	8822 (Electronic copy)
1	MS1138	Stephanie Kuzio	8825 (Electronic copy)
1	MS9159	Jerry McNeish	8754 (Electronic copy)


Sandia National Laboratories