MATHEMATICAL PROGRAMMING FORMULATIONS FOR THE OPTIMAL PLACEMENT OF IMPERFECT DETECTORS WITH APPLICATIONS TO FLAMMABLE GAS DETECTION AND MITIGATION SYSTEMS

A Dissertation

by

ALBERTO JOSE BENAVIDES SERRANO

Submitted to the Office of Graduate and Professional Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Chair of Committee,	M. Sam Mannan
Co-Chair of Committee,	Carl D. Laird
Committee Members,	Mahmoud El-Halwagi
	Martin Wortman
Department Head,	M. Nazmul Karim

December 2014

Major Subject: Chemical Engineering

ABSTRACT

The placement of detectors in mitigation systems is a difficult problem usually addressed in the industry via qualitative and semiquantative approaches. Simplifications are used to circumvent difficulties regarding problem size, parameter uncertainty, and lack of information concerning leak development. Given recent improvement of consequence modeling tools, the use of a stochastic Mixed-Integer Linear Programming (MILP) formulation (SP) was previously proposed to quantitatively approach this problem. This formulation minimizes the expected damage over a large set of gas leak scenarios while assuming perfect detectors. In reality gas detectors are prone to false positives and false negatives. Two solutions are usually implemented in the process industries. First, additional confirmation from several detectors (i.e., voting) is required before emergency actions are triggered in order to avoid false positives. Second, in order to avoid false negatives, the unavailability of the detectors is considered in the placement strategy. Unavailability corresponds to the probability that the detector will not be able to perform its intended function when required.

In the first part of this dissertation, two problem formulations were developed and validated to address the issue of imperfect detectors: minimization of expected damage considering unavailability (SP-U) and minimization of the expected damage considering unavailability and voting (SP-UV). SP-U and SP-UV placement results were compared with those obtained assuming perfect detectors. Results demonstrate that explicit consideration of unavailability and voting effects alters the final detector placement. Quantitative risk can be significantly higher if we neglect these issues when solving for the optimal placement. Furthermore, SP-UV placement results were compared with those of four existing approaches for gas detector placement using three different performance metrics in

accordance to the objectives of gas detection systems. Results provide further evidence on the effectiveness of the use of dispersion simulations, and mathematical programming, to supplement the gas detector placement problem.

Formulation SP-U assumes a uniform unavailability across all detector types and locations. In the second part of this work, this assumption is relaxed via formulation SPqt, which considers non-uniform dynamic detector unavailabilities. Relaxing this assumption results in a Mixed-Integer NonLinear Programming (MINLP) formulation. SPqt, being an extension of SP-U, explicitly considers different backup detection levels, allowing an approximation where the maximum degree of the nonlinear products considered can be determined by the modeler. The effect of reducing the number of detection levels was analyzed. For the problem, results shown that two detection levels are sufficient to find objective values within 1% of the optimal solution. Considering two detection levels reduces the MINLP formulation to a zero-one quadratic formulation (SPqt-Q). A solution quality comparison between SPqt-Q and approximate solution strategies previously proposed in the literature demonstrates its suitability to obtain approximate answers for the general nonlinear problem. Two exact linear reformulation strategies (SPqt-L_1 and SPqt-L_2) were proposed for SPqt-Q and validated from the computationally efficiency perspective.

All the results presented were obtained by using four real data sets provided by GexCon. The data corresponds to FLACS CFD dispersion simulations including the full geometric features of an offshore facility and capturing the uncertainty in the leak characteristics. Additionally, real unavailability values were obtained from industry gas detector reliability databases.

The work presented here constitutes a step forward toward the achievement of a realistic detector placement formulation that includes current industrial practice for these important safety systems.

En memoria de mi abuelita Leonor. Gracias por cuidar de mí; en el jardín infantil y en la vida entera.

A mi familia.

ACKNOWLEDGEMENTS

These past few years have been an amazing journey of learning and self-discovery. The persons below, amongst several others, made it possible.

I would like to start by thanking my parents and my grandmother. Their selfless support and love have always been the main motors of my development. I deeply admire the sacrifice they put themselves through every day in order to ensure better opportunities for my brother and me. Their determination is my inspiration to overcome my personal challenges. To my parents, brother, and family: thank you for encouraging me to set high goals for myself, for believing in me when I pursued those goals, and being there every step of the way to support me. In more than one way, graduate school was our communal effort and I never would have made it without you. I love you.

I am also profoundly grateful to Dr. Carl Laird. All his mentoring, and the development opportunities he provided, made graduate school a far better experience than I have ever hoped to be. Not only he went the extra mile in terms of providing me with the tools necessary to advance my knowledge, he also taught me how to be a better person. I am specially grateful to him for sharing his passion for knowledge and research, encouraging me to love what I do, and being a caring and inspiring leader that teaches by example. He is a terrific advisor and human being, it has been my utmost honor working with him these years and having him as an example to follow.

I extend my most sincere gratitude and appreciation to Dr. Sam Mannan. His guidance and support have been crucial during my graduate school years, and even before. I first came to Texas A&M as an exchange undergraduate student. He took me under his wing and introduced me to a whole new world of exciting ideas and opportunities. My decision to pursue a graduate degree was determined by his involvement in my education. As my advisor, on my graduate school years, he continued to stood by my side to help me and correct me to guarantee that I always went further. For all of his influence in my life, I can not thank him enough.

I thank my group members, both from Dr. Laird's group and the Mary Kay O'Connor Process Safety Center. I am specially thankful to Dr. Sean Legg, Arpan Seth, Dr. Daniel Word, Gabe Hackebeil, Adewale Awoniyi, Camilo Rosas, and Josh Richardson. It is a pleasure to be part of such great groups of individuals. Thanks for your friendship, mentoring, collaboration, and guidance. It means a lot to me.

I would also like to thank Valerie Green, Donna Startz, and Mary Cass, for their extensive support in these years. The Mary Kay O'Connor Process Safety Center provided the funding for this research. Dr. Mannan, Valerie, Donna, Mary, and the rest of the team, made this possible for me with their constant effort.

I would like to acknowledge the members of my committee, Dr. Mahmoud El-Halwagi and Dr. Martin Wortman. Thanks for your sincere advice. I appreciate your willingness to always help me.

I also appreciate all the help from GexCon, and specially Dr. Scott Davis and Are Bratteteig, for their support in providing dispersion simulation data for this work. Also, I want to thank DNV's collaboration, and particularly Dr. Robin Pitblado, for the invaluable technical and field experience input provided.

Finally, I would like to thank Dr. Richart Vázquez-Román from the Instituto Tecnológico de Celaya and Dr. Hans Pasman from the Mary Kay O'Connor Process Safety Center.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1. INTRODUCTION: MOTIVATION, CHALLENGES, AND DATA *

Gas mitigation systems usually constitute the last line of defense against health, safety, security, and environmental disasters. While gas mitigation systems differ depending on the wide variety of gases they can be designed to mitigate, they share a common condition: the requirement to efficiently acknowledge the hazards before issuing executive actions to minimize the hazard consequences. This condition causes the proper placement of gas detectors to be a major concern in their design.

A key shortfall of gas detector placement approaches to date is their inability to rigorously handle the uncertainties quantitatively. The overwhelming amount of information and uncertainties to consider, in conjunction with the huge number of placement possibilities, presents a challenging problem. This difficulty is often circumvented by the usage of simplified qualitative and semi-quantitative approaches. When detailed gas dispersion data is available, it is typically used only as a guideline for the placement, not taking advantage of the full extent of the information that this data can provide. To fill this gap, while considering the inherent uncertainty associated with gas detection, the use of a stochas-

* Part of this section is reprinted with permission from "Optimal Gas Detector Placement Under Uncertainty Considering Conditional-Value-at-Risk" by Legg S. W., Wang C., Benavides-Serrano A. J., and Laird C. D., 2012. Journal of Loss Prevention in the Process Industries, Copyright 2012 by Elsevier Ltd.

Part of this section is reprinted with permission from "A Stochastic Programming Approach for the Optimal Placement of Gas Detectors: Unavailability and Voting Strategies" by Benavides-Serrano, A. J., Legg S. W., Vázquez-Román R., Mannan, M. S., and Laird C. D., 2014. Industrial & Engineering Chemistry Research, Copyright 2013 by American Chemical Society. ACS Articles on Request author-directed link: `http://dx.doi.org/10.1021/ie401369v`.

Part of this section is reprinted with permission from "Effect of Backup Detection Levels in P-Median Formulations for Optimal Placement of Detectors in Mitigation Systems" by Benavides-Serrano, A. J., Hackebeil, G., Mannan, M. S., and Laird, C. D., 2014. Proceedings of the 8th International Conference on Foundations of Computer-Aided Process Design (FOCAPD), Washington, USA, July 2014, Copyright 2014 by Elsevier Ltd.

Part of this section is reprinted with permission from "A Quantitative Assessment on the Placement Practices of Gas Detectors in the Process Industries" by Benavides-Serrano A. J., Mannan, M. S, and Laird C. D., 2014. To appear in the Journal of Loss Prevention in the Process Industries.

tic programming formulation (SP) equivalent to the P-median Problem (PMP) (Hakimi, 1965; ReVelle and Swain, 1970) was proposed and validated by Legg et al. (2012a,b); Legg (2013); Legg et al. (2013). Results demonstrated the potential and suitability of mathematical programming for the gas detector placement problem in mitigation systems while rigorously considering its inherent uncertainties. This work was motivated on the proposal and validation of formulation SP by Berry et al. (2005), for the detection of contamination scenarios in municipal water networks. The results obtained in the gas detector placement context constitutes unequivocal evidence of the wide range of applications that formulation SP and its extensions have. The formulations presented in this work, being generalizations of formulation SP, can be applied to a wide range of facility location and mitigation system detector placement problems. This work will focus on the flammable gas detection and mitigation case.

The optimization-based approaches discussed above assume that detectors are perfect. Therefore, they do not consider two key features associated with flammable gas detector equipment and policies: detector unavailability and voting. As recognized throughout the industry and outlined in sources like Stiftelsen for industriell og teknisk forskning (SINTEF) (2002), detectors are prone to a list of failure modes that include erratic or failed output, failure to function on demand, spurious signal, and others. These failure modes can manifest as false positives or negatives, and both these aspects should be considered.

First, the potential unavailability of detectors should be considered when designing the system. Unavailability corresponds to the probability that the detector will not be able to perform its intended function when required, i.e., the probability of a false negative. Detector unavailability is a broader concept than detector reliability. Unavailability not only includes situations like random failure, but also considers other aspects like the detector being offline due to preventive maintenance and testing, or the absence of the detector due to repairs or replacement. A complete discussion on the methods and considerations for the unavailability calculation is presented by Modarres et al. (2010). Placement approaches should consider the probability that a detector will be unavailable during a hazardous event.

Second, detection redundancy is often required before emergency actions are triggered. A voting logic scheme is often utilized to require confirmation by k detectors before the existence of a hazard is acknowledged. This k-out-of-p detectors confirmation requirement, where p is the total number of detectors, is commonly stated as k-o-o-p. The 2-o-o-p voting logic is the most widely used. The purpose of the implementation of these redundant schemes is to create a system that is robust against false signals resulting from electrical and mechanical failures, inappropriately selected set points, exposure to non-target contaminants, and negligible emissions from external sources (Center for Chemical Process Safety (CCPS), 2009). More precisely, this voting logic will shield the system against costly executive actions issued in response to false positives. In practice, initial warning actions like alarms, are often assigned to 1-o-o-p detections while further actions like emergency shutdown systems are assigned to k-o-o-p detections.

In this work, we extend the concepts and formulations of Berry et al. (2005) and Legg et al. (2012a) to include detector unavailability, i.e. the possibility of false negative cases, and voting logic, in order to avoid false positive cases. These work constitutes a significant step forward in the realism of the problem formulation.

The remainder of this work is divided into seven sections. The final part of this section focuses on introductory concepts for flammable gas detection and mitigation systems, current flammable gas detector placement approaches and their flaws, the motivation behind the use of mathematical programming formulations to solve these flaws, and the data employed for the work presented. Finally, a summary of the hardware and software employed for this work is presented. Section 2 provides a review of the treatment of imperfection concepts in the facility location context. The theoretical and modeling concepts addressed by the facility location literature can be generalized and applied to a broader range of placement problems, see for example, Malcolm and ReVelle (2005). In our particular case, the concept of lack of service in facility location problems is the analog of detector unavailability due to reliability and maintainability considerations. Section 3 presents formulation SP-U, an improved formulation of the Reliability PMP (RPMP) of Snyder and Daskin (2005), which includes the concept of detector unavailability into formulation SP. This formulation adapts the concept of backup facilities by redefining the original SP decision variables to account for different detection levels. This redefinition, and the assumption that all detectors have the same unavailability, allowed the use of a binomial distribution in the objective function to model detection failure, resulting in a mixed-integer linear programming (MILP) formulation. Section 4 presents formulation SP-UV, a further generalization of formulation SP-U that makes use of the negative binomial distribution to model detector failure and voting policies while still resulting in an MILP formulation. In both sections, results are presented and compared with optimal placement results from previous formulations that ignore the extended formulation features. Unavailability and voting logic considerations result in changes to the optimal detector placement and significant improvements in the expected time to detection when false positives and false negative alarms are considered. In Section 5, four existing approaches for gas detector placement were implemented and compared with the previously proposed quantitative optimization-based approach using three different performance metrics in accordance to the objectives of gas detection systems. Results provide further evidence on the effectiveness of the use of dispersion simulations, and mathematical programming, to supplement the detector placement problem.

The uniform unavailability assumption used in Sections 3 and 4 (Formulations SP-U and SP-UV, respectively) is reasonable for gas detection and mitigation systems that use the same type of detectors under the same process, environmental, maintenance and repair conditions. However, if this is not the case, a natural extension can be used in order to account for the non-uniform detector unavailabilities. This extension introduces nonlinearities due to the multiplication of probabilities in the objective function. These nonlinearities result in a Mixed-Integer Non-Linear Programming (MINLP) formulation. Motivated by this perspective, Section 6 uses real facility data for the optimal gas detector placement problem to determine the impact of changing the number of detection levels (i.e., the number of factors in the objective probability multiplications) and select a level of redundancy that gives a reasonable accuracy while reducing the complexity of the MINLP. Results show that it is reasonable to consider two detection levels in order to obtain a zero-one Quadratic Programming (QP) formulation that can be solved to optimality with minimal deterioration of the optimal objective. In Section 7 we present a general formulation, SPqt, that considers non-uniform dynamic detector unavailabilities. This formulation, based on the SP-U formulation, explicitly considers detection levels. This feature, and the results presented in Section 6, allowed us to propose a truncated version of SPqt, SPqt-Q, to efficiently obtain approximate answers for the general nonlinear problem. The computational efficiency of two exact linear reformulation strategies (SPqt-L_1 and SPqt-L_2) for formulation SPqt-Q is analyzed in section 7.3. Finally, formulation SPqt-Q solution quality is compared against current solution strategies for the full nonlinear problem. A summary, conclusions, and future work are presented in Section 8.

1.1 Gas Detector Placement: Importance, Current Performance, and Challenges

Gas detectors constitute the key component of the flammable gas detection system, an important safety system with interfaces to several other safety safeguards. Incidents like the Buncefield fire are tangible and harsh reminders of this importance and the need for proper detection. The Buncefield fire (Buncefield Major Incident Investigation Board, 2008a,b) was a major conflagration caused by a series of explosions at the Hertfordshire

Oil Storage Terminal, an oil storage facility. The incident took place when a high-level switch failed to operate during a normal night filling operation. According to the calculations, on the morning of December 11, 2005, between 5:20 am and 5:30 am, a tank overflowed causing a pool formation and the subsequent formation of a vapor cloud. From 5:30 am until 6:00 am the vapor cloud thickened and spread. The first and largest explosion occurred at 6:01 am. This led to a domino effect, which eventually overwhelmed 20 large storage tanks. Forty minutes were available to avoid this incident if appropriate detection and mitigation would have been in place. As part of the conclusions and recommendations, the investigation report (Buncefield Major Incident Investigation Board, 2008a,b) that followed this incident stated that improvements were necessary in the design and siting of the systems for detection of flammable vapors.

Despite receiving widespread media and general public attention due to third party damages, the property damage value of the Buncefield incident was small compared to other catastrophic incidents experienced by the hydrocarbon industry. From a review of the 100 largest property damage losses, around 70 are attributed to fires, explosions, and/or vapor cloud explosions (Marsh, 2012). These are all incidents where the fire and gas detection system played, or could have played, an important role in preventing further damages after loss of containment. The number of incidents remains high, and the data do not indicate a decreasing trend. Bureau of Safety and Environmental Enforcement (BSEE) (2012) data for the US outer continental shelf attributed a total of 1612 incidents to fires and explosions from 1996 to 2011 (Not including 2006), 649 of them in the period from 2007−2011. Health and Safety Executive (HSE) (2007) data from 1980 to 2005 for floating offshore units attributed a total of 296 incidents to fires and explosions, 235 of them in the period from 1990−2005. The Petroleum Safety Authority (2012) reported that there is not significant statistical evidence to support the idea that there has been a reduction in the number of leaks per facility year in the Norwegian continental shelf. This conclusion was obtained for leak rates greater than 0.1 kg/s, and compared data from 2011 against the average for the period 2003−2010. Furthermore, the Health and Safety Executive (HSE) (1997, 2003) reported that less than 50% of the known releases in offshore facilities are detected by the facility's gas detection system. If unknown releases are considered, the actual fraction of releases detected is even lower.

To a great extent this poor detection performance is attributed to the problem of sub-optimal detector placement. While effective technology exists for gas detection, several difficulties make the problem of gas detector placement in the process industry challenging. Leak location, size, and duration are generally unknown, leading to a large uncertainty space and a large number of potential leak scenarios to consider. Second, formal quantification of the risk for any given leak scenario is difficult. The gas leak dispersion development and transport depend on fluid properties, environmental factors, and facility geometry. Reliable gas dispersion simulations are needed to accurately assess leak development. Personnel, assets and production, environmental, and business image costs should be considered. For flammable releases, explosion and fire consequences need to be assessed, requiring the formal quantification of ignition probabilities, structural damage, personnel location patterns, and human response. Finally, even if all this data is consolidated with the highest quality, due to the combinatorial aspects of the problem, exhaustive search is not an option. For example, assuming a detector placement study identifies 1000 candidate detector locations, the number of possible placement combinations will be $2^{1000} \approx 10^{300}$.

The correct placement of detectors in mitigation systems not only impacts safety, but it also has an effect on the proper allocation of resources. According to Bratteteig et al. (2011), the cost of adding to or modifying existing detector layouts in facilities can be prohibitive. Given the high cost of the detector system and the importance of this key layer of protection, it is imperative that we make the best use of these economic resources by developing an improved quantitative approach for detector placement.

1.2 Gas Detector and their Placement Practices in the Process Industries

Catalytic and infrared gas detectors are the most commonly used detectors for the detection of combustible gas clouds (Health and Safety Executive (HSE), 2011; Nolan, 2010). Catalytic gas detectors provide an accurate measurement of contaminant concentration in air through an oxidation-reduction reaction on a catalyst. While these detectors have the benefit of a low unit cost, they are susceptible to catalyst poisoning and will only reveal a failure of the unit when inspected through regular maintenance. Conversely, infrared detectors have a higher unit cost and lower maintenance requirement. These detectors operate by detecting the absorption of infrared energy by the surrounding contaminant cloud. This means that the detectors can detect contaminants over larger distances, but provide lower overall accuracy in terms of concentration quantification. These detectors can operate as point detectors and line-of-sight detectors. Line-of-sight detectors can detect a contaminant cloud crossing a beam over extremely long distances. They also possess the added benefit of failing positive, allowing for immediate detection of a failed detector. Unfortunately, these detectors provide a concentration measure that is integrated over distance, so therefore do not provide a precise concentration of the contaminant cloud. Either fixed or portable detectors are available; however, for the applications presented in this work, only fixed detectors are considered.

Regulations, standards and recommended practices for gas detection systems mostly provide general guidelines regarding the placement of fixed gas detectors. Recommendations and requirements are focused on installation, testing and performance, calibration, detection technologies and the type of actions expected in response to a confirmed gas leak. Most of them do not provide guidelines regarding the number of detectors or the placement strategies that should be used. Examples include: FM Global (2001), American

Petroleum Institute (API) (2001) (Section C.1.3.2) , International Society of Automation (ISA) (2003) (IEC 61779-6 Mod, Section 6), National Fire Protection Association (NFPA) (2007) (Section 6.5.2.7.1), Canadian Standards Association (2001), Health and Safety Executive (HSE) (2001) (Section 4), International Organization for Standardization (ISO) (2003), International Organization for Standardization (ISO) (1999) (Appendix B.6), Russian agency on technical regulating and metrology (GOST) (1981), Det Norske Veritas (DNV) (2008) (Section 4.D), and Oil & Gas UK (UKOOA) (2003). More recently the use of dispersion studies has gained recognition as a tool to better understand the behavior of the releases, e.g., EC 60079-29-2 (Section 8) (International Electrotechnical Commission (IEC), 2007), and NORSOK STANDARD S-001 (Sections 12 and 13) (Norsk Sokkels Konkuranseposisjon (NORSOK), 2008). However, in the above-mentioned sources, methods for determining gas detector placement using data provided by dispersion studies are not specified, and common industry practice considers only a limited set of high-impact scenarios.

The generality of these regulations, standards and recommended practices in conjunction with the inherent challenges of the gas detector placement problem enumerated in Section 1.1, has resulted in a widespread use of prescriptive and qualitative detector placement approaches. These strategies rely upon the identification of key process equipment, development of credible release scenarios, and are based upon the properties of the particular gases being studied. Qualitative methods for the placement of gas detectors are outlined in the guidelines set forth by Center for Chemical Process Safety (CCPS) (2009). These methods can be categorized by their main intended purpose. These categories include source monitoring, volumetric monitoring, enclosure monitoring, perimeter monitoring, and path of travel and target receptor monitoring. Examples include NFPA 15 (Section 6.5.2.7.1) (National Fire Protection Association (NFPA), 2007), API RP 14C (Section C.1.3.2) (American Petroleum Institute (API), 2001), ANSI/ISA-RP12.13.02 (IEC

61779-6 Mod) (International Society of Automation (ISA), 2003), IEC 60079-29-2 (International Electrotechnical Commission (IEC), 2007), Health and Safety Executive (HSE) (1993) (Section 6), Health and Safety Executive (HSE) (2001) (Section 4.6), Oil & Gas UK (UKOOA) (1995), Nolan (2010) (Section 17.6), and ISA-TR84.00.07 (Annex A.2, Step 7) (International Society of Automation (ISA), 2010).

Dispersion studies are generally perceived to be of significant importance, though methods for determining gas detector placement do not always utilize the full value of the data provided by these studies. It was not until recently that standardization entities started assessing the use of these metrics in performance-based designs. ISA-TR84.00.07-2010 (International Society of Automation (ISA), 2010) is the state of the art in this body of literature. Scenarios and geographical coverage quantification is proposed as metrics to achieve a desired risk reduction in the design of fire and gas detection systems.

In order to increase the effectiveness of many of the approaches mentioned previously, risk and programming approaches have been coupled with the semi-quantitative approaches. Strøm and Bakke (1999) proposed a performance-based algorithm for the detector placement where potential detectors locations are ranked according to an overall efficiency metric and selected by the ranking. The grid of potential detector locations is defined within the volume of the facility and the overall efficiency is calculated for each potential detector. ISA-TR84.00.07 (International Society of Automation (ISA), 2010) proposed an iterative approach to detector placement. First, a candidate placement is chosen and a coverage-based mitigated risk assessment is performed. If the desired risk threshold is not met, the gas detector placement is modified and the process is repeated. DeFriend et al. (2008) proposed a risk-based methodology to determine the maximum gas cloud size that must be detected to maintain a tolerable risk level for the facility. Additionally, methodologies with optimization concepts were also considered. In Dhillon and Chakrabarty (2003), probabilities of missed detection and coverage considerations were incorporated into two general purpose algorithms. A genetic algorithm with elitist selection was proposed by Obenschain et al. (2004), where population members were assigned fitness scores and the members with the highest scores were carried into the next population. Gencer et al. (2008) proposed mathematical models to optimally locate chemical detector systems and alarm systems to provide effective interaction between both systems. Lee and Kulesz (2008) developed an iterative dynamic programming algorithm designed to create a detector placement that minimizes a risk-based objective function.

While these approaches strive for performance based quantitative designs, and they represent improvements over qualitative techniques, they fail to fully overcome the difficulties described earlier. Lack of probabilistic scenario analysis, oversimplification in metrics and risk considerations, and the use of non-optimal methods to achieve the design objective are their most common flaws. A more systematic approach is still desired to address several key issues. First, the estimated performance of a particular gas detector placement is highly dependent on important outputs such as point concentrations, cloud size, and detection times. Rigorous simulation of gas dispersion is needed to accurately calculate these outputs, and several key variables such as process conditions and geometry, leak locations and gas properties, and weather conditions all have significant impact on these dispersion simulations. The influence of these key variables is reviewed in the works by Kelsey et al. (2002), Kelsey et al. (2005), Bratteteig et al. (2011), and Marx and Cornwell (2009). Furthermore, much of the existing work and industry practice is based on heuristics or analyses considering only a limited set of high-impact scenarios. These methods do not provide a rigorous treatment of the high levels of uncertainty associated with variables like leak location, leak characteristics, and weather conditions. Because the uncertainty space for these variables is large, a high number of plant-specific leak scenarios are required. With these simulations however, it is possible to evaluate key performance metrics like the expected detection time. Additionally, using simulation data from many scenarios, statistics on these metrics can be calculated to provide confidence in detector placements. Finally, many of the programming techniques previously discussed provide no guarantee of global optimality. Even when guarantees against suboptimality are provided, the detector placement metrics do not consider the important key variables necessary. An optimization-based approach that considers uncertainty while using the valuable dispersion simulation data provided by rigorous dispersion models and providing guarantees of optimality is preferred.

1.3 Input Data for Formulations

This subsection presents the data employed for the generation of the results in the following sections. Two main data types and sources were employed: Dispersion simulation data provided by GexCon US and real gas detector unavailability data from reliability databases.

1.3.1 CFD Dispersion Simulations

Dispersion simulations for this study were performed following standards and recommendations specially designed for capturing the gas dispersion problem uncertainties (Norsk Sokkels Konkuranseposisjon (NORSOK), 2001, 2008). The procedure is outlined in Hansen and Middha (2008) and Hansen et al. (2013). The availability of this high quality data was the main motivation behind the particularization of this study to flammable gas detection and mitigation systems. This data is a superset of the data previously employed by Legg et al. (2012a,b); Legg (2013); Legg et al. (2013).

An initial assessment was carried out in order to determine the candidate detector locations (L). This assessment was based on considerations regarding variable costs such as wiring, detection likelihood, accessibility and technical viability for installation, inspection and maintenance activities. Subsequently, a set S of possible leak source locations was assessed. Commonly assessed leak situations include material deterioration (e.g. corrosion, erosion, fatigue failure), inadequate sealing and welding, collision, overfilling, overpressures, and runaway reaction. Based on the set of possible leak source locations S, a set A of leak scenarios was simulated using the CFD code FLACS (GexCon, 2011). The use of CFD simulations is a recommended practice in order to guarantee the quality of the input data, especially for facilities with intricate geometries. Nevertheless, when the geometry and other relevant considerations allow for the use of simpler models, these might be used in order to reduce the computational requirements of scenario generation. Scenario modeling and simulation involved the examination of inventories, process conditions, type of releases, air movement and ventilation patterns, and geometric details of the given module. To acknowledge the stochasticity of the leak scenario a combination of different leak characteristics (size, flow, etc), wind velocities and wind directions were included in the analysis for each leak source location. Scenarios were generated for a wide range of leak conditions, excluding those that are so small as to be negligible or undetectable, and those that are so large as to be noticed by changes in process variables.

Results were generated using 4 independent data sets. Simulations were performed on a real, medium-scale, proprietary offshore facility geometry capturing the full process features, i.e. equipment, piping, support structures, etc. The gas composition consist of a mixture of light alkanes, mostly methane. The different data sets correspond to 4 different modules in the same facility. Data set A is composed of 270 release scenarios and 994 potential point detector locations. Data set B is composed of 145 scenarios and 943 potential point detector locations. Data set C is composed of 78 scenarios and 607 potential point detector locations. Data set D is composed of 314 scenarios and 768 potential point detector locations. The volumes to monitor were $21,000\,\mathrm{m}^3$ ($28\,\mathrm{m} \times 60\,\mathrm{m} \times 12.5\,\mathrm{m}$), $10,548\,\mathrm{m}^3$ ($21\,\mathrm{m} \times 48\,\mathrm{m} \times 10.5\,\mathrm{m}$), $3,570\,\mathrm{m}^3$ ($17\,\mathrm{m} \times 30\,\mathrm{m} \times 7\,\mathrm{m}$), and $2,520\,\mathrm{m}^3$ ($12\,\mathrm{m} \times 30\,\mathrm{m} \times 7\,\mathrm{m}$), respectively. The smaller length corresponds to the height in all cases.

1.3.2 Gas Detector Unavailability in the Process Industries

Typical gas detector unavailability values in the process industries were employed for all the formulation validation and testing. A review is presented below. Notation is summarized in Table A.1 (Appendix A).

First, a distinction must be make between the instantaneous unavailability, $q(t)$, and the time-averaged unavailability, $\bar{q}$. The instantaneous unavailability, $q(t)$, is the unavailability at a given time $t > 0$. The time-averaged unavailability, $\bar{q}$, is the average value of the instantaneous unavailability over a length of time. These two values are related by Equation (1.1).

$$\bar{q} = \frac{1}{t_2 - t_1} \int_{t_1}^{t_2} q(t)\,\mathrm{d}t \tag{1.1}$$

For gas detectors, failure to function on demand is most likely detected upon testing and inspection. Modarres et al. (2010) denotes this type of equipment as periodically inspected (tested) systems with instantaneous and time-averaged unavailabilities given by Equations (1.2) and (1.3), respectively. Parameters λ and T_O represent the detector constant failure rate (h^{-1}) and the detector operating time (up time) (h), respectively. Equations (1.2) and (1.3) assume that testing and repair are perfect and small in comparison with the detector operation time, and therefore can be neglected in the unavailability calculation.

$$q(t) = \lambda t \tag{1.2}$$

$$\bar{q} = \frac{1}{2}\lambda T_O \tag{1.3}$$

One of the most recognized sources available for the determination of parameter λ in

Equations (1.2) and (1.3) is the Offshore REliability DAta (OREDA) database (Stiftelsen for industriell og teknisk forskning (SINTEF), 2009). The OREDA project is an initiative of several of the world's major oil and gas companies to collect and analyze real offshore equipment maintenance and reliability data in order to provide input to reliability analyses. According to this database, the constant failure rate, λ, of infrared hydrocarbon gas detectors (most commonly used type of flammable gas detector) undergoing a failure to function on demand was estimated to have an average value of $1.03 * 10^{-6}(h^{-1})$ with a 90% confidence interval of $[0.06 * 10^{-6}(h^{-1}), 3.62 * 10^{-6}(h^{-1})]$. These values were computed based on a population of 221 detectors surveyed on 6 different offshore facilities over a total operational time of $6.05 * 10^6$ hours. The multiple facility sample was aggregated and fitted to a Gamma distribution following the considerations provided by Spjøtvoll (1985). Based on this data, and making use of Equations (1.2) and (1.3), the expected values of the instantaneous and time-averaged unavailabilities and their 90% confidence intervals are presented in Figures 1.1 and 1.2, respectively.

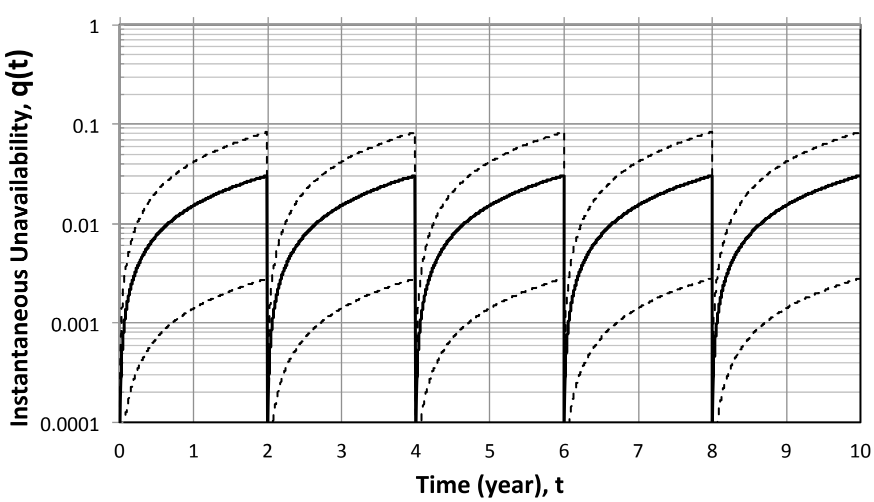


Figure 1.1: Gas detector instantaneous unavailability (—) and its 90% confidence interval (···) as a function of time, t, for T_O=2 years.

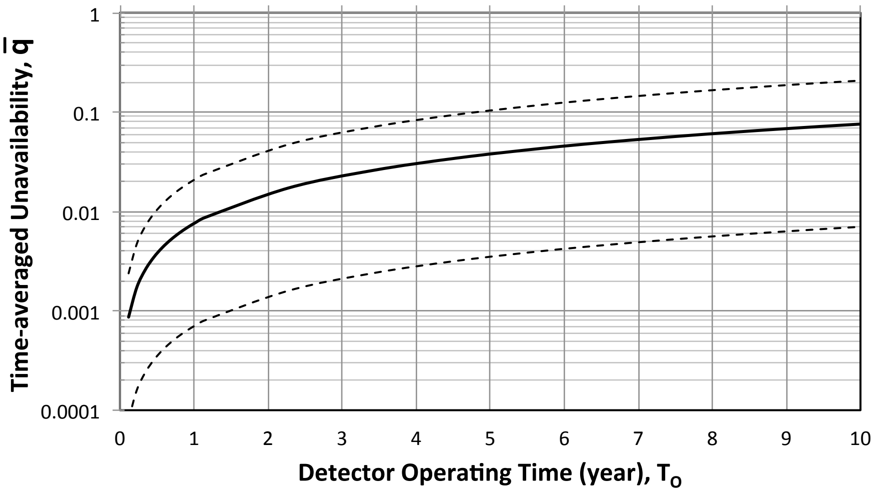


Figure 1.2: Gas detector time-averaged unavailability (—) and its 90% confidence interval (┄) as a function of detector operating time (up time), To.

In the case of Figure 1.1, a detector operating time before testing and repair (T_O) of 2 years was assumed. However, gas detectors in the industry are expected to be tested and repaired more regularly. Under the assumption of perfect repairs, every time the detector is maintained/repaired it is assumed that it goes back to an "as good as new" condition (i.e., q=0). For a detector operating time of 2 years the expected value of the instantaneous unavailability is $\sim$ 0.03, and the upper bound of the 90% confidence interval never exceeds 0.1. Figure 1.2 presents time-averaged unavailability values for different values of detector operating time (up time), T_O. For the extreme case of a detector undergoing $\sim$ 10 years without testing and repair, the expected value of the time-averaged unavailability does not exceed 0.08, and the upper bound of the 90% confidence interval is close to 0.2. A gas detector in a facility with standard maintenance and repair intervals (i.e., T_O<2 years) can be expected to have time-averaged unavailabilities below 0.05 (the upper bound of the 90% confidence interval for an operating time of 2 years is 0.032).

Typical industry rules of thumb assign unavailability values up to 0.1 to the whole detection system equipment unavailability. However, this number includes factors like emergency shutdown system actuation failure and detector coverage considerations for small leaks. Based on this consideration, the real gas detector unavailability values presented in Figures 1.1 and 1.2, and assuming standard repair and maintenance practices, it can be conservatively assumed that actual instantaneous and time-averaged unavailability values for gas detectors (i.e. including wiring and additional equipment failure considerations) are between 0.01 and 0.1.

1.4 Software and Hardware

All the problem formulation presented in this work were formulated in Pyomo and solved using either CPLEX 12.2 (Sections 3-6), CPLEX 12.5 (Section 7), or Gurobi 5.6 (Section 7). The Python Optimization Modeling Objects (Pyomo) software package (Hart et al., 2011, 2012) is an open source tool for the definition and solution of optimization problems within the high-level programming language Python. Pyomo supports the representation of linear, mixed-integer linear, nonlinear, and nonlinear mixed-integer models, while providing the user with the capability to access the wide set of Python supporting libraries. Pyomo is a package within the Coopr (COmmon Optimization Python Repository) software library.

A dual quad-core Intel(R) Xeon(R) CPU X5482 with a clock speed of 3.2GHz and 18 GB RAM was used to solve the problem instances presented in Sections 3-6. The timing results presented in Section 7 were obtained using an Intel Xeon CPU E5-2697 v2 with a clock speed of 2.7 GHz and 264 GB of RAM.

The Pyomo and Python files necessary for the generation of the results presented in this work are presented in Appendices B-P.

2. BACKGROUND: PREVIOUS WORK AND LITERATURE REVIEW *

In this section, a review is presented outlining the concepts and previous developments on which the work at hand is built. First, the main mathematical programming formulation, SP, is discussed. The formulation is outlined, along with the motivations and specific advantages behind its use in the detector placement problem for mitigation systems. Previous extensions to formulation SP, which originated from within our research group, are also presented. Finally, a literature review of the treatment of the concept of imperfect detectors is presented. The use of mathematical programming formulations, and in particular formulation SP, to address the detector placement problem in mitigation systems is quite recent, making the literature directly dealing with it scarce. However, several optimal placement problems considering redundancy, backup coverage, and unavailability have been previously developed and studied in operations research, under the umbrella of optimal facility location. Once it is recognized that the detector placement problem is an optimal facility location problem, it is possible to extrapolate several of the concepts presented in these previous formulations to the problem at hand. This literature review will be focused in these formulations.

---

* Part of this section is reprinted with permission from "Effect of Backup Detection Levels in P-Median Formulations for Optimal Placement of Detectors in Mitigation Systems" by Benavides-Serrano, A. J., Hackebeil, G., Mannan, M. S., and Laird, C. D., 2014. Proceedings of the 8th International Conference on Foundations of Computer-Aided Process Design (FOCAPD), Washington, USA, July 2014, Copyright 2014 by Elsevier Ltd.

Part of this section is reprinted with permission from "Optimal Gas Detector Placement Under Uncertainty Considering Conditional-Value-at-Risk" by Legg S. W., Wang C., Benavides-Serrano A. J., and Laird C. D., 2012. Journal of Loss Prevention in the Process Industries, Copyright 2012 by Elsevier Ltd.

Part of this section is reprinted with permission from "A Stochastic Programming Approach for the Optimal Placement of Gas Detectors: Unavailability and Voting Strategies" by Benavides-Serrano, A. J., Legg S. W., Vázquez-Román R., Mannan, M. S., and Laird C. D., 2014. Industrial & Engineering Chemistry Research, Copyright 2013 by American Chemical Society. ACS Articles on Request author-directed link: `http://dx.doi.org/10.1021/ie401369v`.

2.1 Formulation SP and Previously Addressed Extensions

Given the large uncertainty space and the combinatorial problem of selecting detector locations, numerical optimization is a promising quantitative approach for the problem of detector placement in mitigation systems. In order to determine an optimal detector placement while rigorously considering inherent detection and mitigation system variables and uncertainties, Berry et al. (2006b) proposed the use of a Mixed Integer Linear Programming (MILP) formulation. This formulation (SP), based on the P-Median Problem (PMP) (Hakimi, 1965; ReVelle and Swain, 1970), is a stochastic programming formulation that determines a detector placement that minimizes the expected value of a damage coefficient across a large number of hazardous scenarios. This initial work by Berry et al. (2006b) successfully applied the formulation to the detection of contaminants in water networks. Motivated by these results, Legg et al. (2012b) applied formulation SP to flammable gas mitigation systems. Point detectors used in water networks detect contamination only in the instance where the contaminant passes directly through the location of the detectors (Berry et al., 2005). In this sense, the detectors act in the same manner as the fixed infrared (point and line-of-sight) and catalytic gas detectors typically used in process facilities (Fire & Safety World Online, 2011). The significant difference between detector placement formulations for water network problems and those for open-air dispersion problems are the simulation frameworks used to create the contamination scenario data. In the water network problems (Berry et al., 2006b), the network model EPANET is used to generate the data necessary. Because our work considers process facilities, flammable gas releases, rather than water quality simulations, are the scenarios of interest. Therefore, gas dispersion software (CFD in this case) is needed to generate rigorous simulations for the wide range of process conditions, leak locations, and weather conditions possible. By making use of data set A described in Section 1.3.1, Legg et al. (2012b) further validated the potential and general applicability of formulation SP to approach the detector placement problem in mitigation systems while rigorously considering its inherent uncertainties.

2.1.1 Formulation SP

Formulation SP from Berry et al. (2005), i.e., the PMP for detector placement, is presented in Problem (2.1). Data requirements and notation for the formulations in this paper is summarized in Table A.1. The Pyomo model file containing formulation SP is presented in Appendix F.

$$\min \quad \sum_{a \in A} \alpha_a \sum_{i \in \mathcal{L}_a} d_{a,i} \boldsymbol{x_{a,i}} \tag{2.1a}$$

s.t.

$$\sum_{i \in \mathcal{L}_a} \boldsymbol{x_{a,i}} = 1 \qquad \forall\, a \in A \tag{2.1b}$$

$$\sum_{l \in L} \boldsymbol{s_l} \leqslant p \tag{2.1c}$$

$$\boldsymbol{x_{a,i}} \leqslant \boldsymbol{s_i} \qquad \forall\, a \in A,\ i \in \mathcal{L}_a \backslash \mathcal{D}_a \tag{2.1d}$$

$$\boldsymbol{s_l} \in \{0, 1\} \qquad \forall\, l \in L \tag{2.1e}$$

$$0 \leqslant \boldsymbol{x_{a,i}} \leqslant 1 \qquad \forall\, a \in A,\ i \in \mathcal{L}_a \tag{2.1f}$$

Here, the set L represents the set of all potential detector locations, and the set A represents the set of hazardous scenarios considered. Since each hazardous scenario does not necessarily affect all potential detector locations, subsets $\mathcal{L}_a$, are defined such that $\mathcal{L}_a$ contains all the detector locations that can detect hazardous scenario a. The probability of occurrence of a particular hazardous scenario a is represented by the parameter α_a. The parameter $d_{a,i}$ is the damage coefficient associated with hazardous scenario a, if that scenario is first detected by a detector at location i. The maximum number of detectors that can be allocated is given by p. Two variables are used in the problem, s_l and $x_{a,i}$. Variable s_l is a binary decision variable that signals the existence of a detector at location l (s_l=1, and 0 otherwise). Variable $x_{a,i}$ indicates whether a detector is the first to detect a scenario a at location i ($x_{a,i}$=1, and 0 otherwise). While $x_{a,i}$ is a continuous variable in the PMP, under reasonable assumptions it is guaranteed to converge to an integer solution (Berry et al., 2006b)

The goal of the objective function in Eq. (2.1a) is to minimize the expected value of the desired consequence metric. Constraint (2.1b) enforces the requirement that each hazardous scenario needs to be detected by at least one detector. To account for scenarios that go undetected, this constraint can be relaxed by adding dummy variables ($\mathcal{D}_a$) to $x_{a,i}$, i.e., augmenting $\mathcal{L}_a$ with an index i for a dummy variable. Dummy variables are associated with damage coefficient values, d_{max}, that penalize any undetected hazardous scenarios. Constraint (2.1c) provides an upper limit, p, on the number of detectors allowed. Constraint (2.1d) ensures that location i can only be the first to detect hazardous scenario a if there is a detector placed at location i.

Several features of the the PMP formulation make it suitable for the placement of detectors in mitigation systems. First, it offers a direct coupling between the detector placement and the detection and mitigation system risk minimization objective. As presented by Crowl and Louvar (2011), risk is defined as a measure of the health, environmental, or economic losses in terms of both loss scenario likelihood and loss scenario magnitude. The ultimate goal of objective function (2.1a) is the minimization of risk, the probability of the loss scenario multiplied by the scenario resulting consequence. Particular risk metrics are used for different types of detection and mitigations systems. For instance, in the case of water networks risk metrics include the population exposed and the mass of toxic agent removed from the network via demand. In flammable gas detection and mitigation systems, risk metrics include the time to detection for a predetermined percentage of the Lower

Flammability Limit (LFL) concentration and the total volume of the flammable gas cloud. The PMP formulation can exactly accommodate these relevant detection and mitigation system risk metrics by changing the definition of the damage coefficient. Furthermore, if the loss scenario development is well understood, and its final consequences can be fully quantified, the damage coefficient can ultimately represent the aggregated health, safety, security, and environmental losses. Second, damage coefficients are determined from a preprocessing step in which relevant hazardous scenarios are assessed and simulated. This enables the use of the wide variety of hazard-specific modeling software available for the assessment of scenario probabilities and consequences. The complex non-linearities and uncertainty related to the hazardous scenario development are captured in a traceable manner by this step without impacting the formulation solution time. Since the computational effort associated with the PMP solution is often relatively small in comparison to the effort associated with scenario generation, designers can reuse the simulation data many times in order to test different problem extensions, objectives, numbers of detectors, tolerances, etc. Third, the PMP is a well-reviewed problem in facility siting with an extensive list of solution strategies and problem extensions that can be readily extrapolated to detector placement problems. This list continues to grow, supported by the detector placement community, as presented in the sections below. Two formulation extensions, SP-C and SP-CVaR, are presented below.

2.1.2 Formulation SP-C: Coverage Constraints

Legg et al. (2012a) and Legg (2013), presented and assessed a modified form of the SP formulation (SP-C) that incorporated an additional coverage constraint requiring that some level of spatial coverage be maintained while still attempting to optimize the detector placement. This additional coverage constraint is presented in Equation (2.2) below.

$$\sum_{c \in C_l} s_c \geqslant 1, \quad \forall\, l \in L \tag{2.2}$$

Set C_l represents the set of all locations that provide coverage to candidate location l. For this purpose, a coverage radius is initially determined and subsequently, for each location l, the set C_l is defined in a preprocessing step as all the candidate detector locations contained within the volume predetermined by the coverage radius. Constraint (2.2) requires every candidate location l in the facility to be provided coverage by at least one detector placed within the predetermined volume. This additional constraint embeds the volumetric approach logic, a common detector placement scheme in the process industries, within formulation SP. The volumetric approach was presented in Section 1.2 and is further discussed and assessed in Section 5. Following the work by Mak et al. (1999), a Monte-Carlo sampling procedure was used to determine confidence intervals on the probability distributions of expected time to detection (objective function) and fraction of covered scenarios.

The addition of the coverage constraint improves the resilience of both detector placement metrics to unforeseen scenarios, a specially important result for cases in which the uncertainty space has not been extensively or properly sampled. Furthermore, when compared against a coverage-only placement, both formulation SP and SP-C significantly outperform this approach. This provides further demonstration of the value of quantitative hazardous scenario data (i,e., CFD-dispersion simulations) and the use of mathematical programming to post-process this data.

2.1.3 Formulation SP-CVaR: Improved Tail Behavior

The minimization of formulation SP objective function, that is, the minimization of the expected value of the damage coefficient, is not a guarantee of satisfactory worst-case scenario performance. The solution of formulation SP objective might have associated with it hazardous scenarios with damage coefficients greater than what is acceptable. To address this issue, Legg et al. (2013) and Legg (2013) presented and evaluated a further extension to the model, SP-CVaR, which improves the tail-behavior of the distributions of detection times by considering the Conditional-Value-at-Risk (CVaR) in the optimization formulation.

The CVaR metric is based on the Value-at-Risk (VaR) metric. VaR corresponds to the maximum expected damage coefficient within a predetermined level of confidence. CVaR corresponds to the mean value of the damage coefficient, given the condition that the damage coefficient is higher than the VaR. That is, VaR corresponds to a threshold value for the tail of the probability distribution, while CVaR corresponds to a mean value of the tail of the probability distribution. However, as discussed by Krokhmal et al. (2011), VaR lacks key properties suitable for optimization and control applications, like convexity and subadditivity. CVaR is not only easy to compute, but also possesses several of these attractive mathematical features (Artzner et al., 1999). CVaR, to improve tail behavior, was implemented by adding Equations (2.3) below to the original SP formulation.

$$\beta + \frac{1}{1-\theta}\sum_{a \in A} z_a \alpha_a \leqslant CVaR^* \tag{2.3a}$$

$$z_a \geqslant 0 \qquad \forall a \in A \tag{2.3b}$$

$$z_a \geqslant \sum_{i \in \mathcal{L}_a} d_{a,i} x_{a,i} - \beta \qquad \forall a \in A \tag{2.3c}$$

Parameter θ corresponds to the desired confidence level. Parameter $CVaR^*$ is a pre-computed upper bound on the CVaR value of formulation SP solution.

Legg et al. (2013) and Legg (2013) applied formulation SP-CVaR to data set A, and data sets A, B, and C, respectively. CVaR considerations into formulation SP results in improved tail-behavior with little penalization to the expected time to detection. This result is of special interest when put in the context of detector placement for mitigation systems design, it indicates that it is possible to mitigate the effect of Low-Probability High-Consequence (LPHC) scenarios without compromising the system performance against the common low-consequence scenarios. The LPHC scenario risk (likelihood of the incident $\times$ expected loss in case of the incident), while being equivalent or higher than that of most common events, can be easily disregarded leading to catastrophic events.

2.2 Literature Review: Optimal Placement of Imperfect Facilities

As discussed before, a literature review of the treatment of imperfect facilities in the operations research context is presented in this section. The main goal of this body of literature has been the optimization of service facility placement, particularly emergency medical services (EMS). The development of these problem formulations was principally driven by a desire to extend previously developed problems to allow for the disruption of service. These considerations were initially applied to set covering problems, such as the Location Set Covering Problem (LSCP) (Toregas et al., 1971) and the Maximal Coverage Location Problem (MCLP) (Church and ReVelle, 1974), and later extended to the the PMP. Set covering problems (LSCP, MCLP) and PMPs correspond to discrete facility location models extensively used in EMS allocation (Daskin, 2008). Besides a common taxonomy, these problems also share several theoretical links as presented by Church and Weaver (1986) and reaffirmed in the efforts to unify location-allocation formulations (Lei, 2010). Despite these similarities, backup and facility unavailability considerations into the PMP are more recent and focused in the supply chain context.

Categorization of facility location models is provided by Owen and Daskin (1998), ReVelle and Eiselt (2005), ReVelle et al. (2008) and Daskin (2008). Applications, extensions, and solution methods of the LSCP, MCLP and PMP have been studied by Marianov and Serra (2004), ReVelle and Williams (2004), Daskin and Dean (2005), Reese (2006), Jia et al. (2007), Church and Murray (2009), Marianov and Serra (2009) and Farahani et al. (2012). It is worth mentioning that another body of literature addressed the backup and unavailability issues through queuing (Berman et al., 1987; Larson, 1974, 1975). This approach is valid for facilities, but not for detectors.

2.2.1 Covering Models: Redundant Coverage Formulations

The first approaches to the maximization of backup coverage were modifications of the LSCP. Both Berlin (1972) and Daskin and Stern (1981) proposed a bi-objective problem. A first objective follows the LSCP and minimizes the number of emergency vehicles needed to cover a given area, while the second objective function seeks to maximize the extent of the multiple coverage for the area. Benedict (1983), Hogan and ReVelle (1986), and Eaton et al. (1986) extended the backup coverage considerations from the LSCP to the MCLP. The Backup Coverage Problem (BACOP) proposed by Hogan and ReVelle (1986) seeks to maintain a more uniform level of service by avoiding situations where no ambulance is available when service is demanded. Two formulations are provided, BACOP1 and BACOP2. BACOP1 requires initial coverage at each demand node while BACOP2 allows trading between initial and backup coverage. Pirkul and Schilling (1989) expanded the MCLP in order to include the workload capacities on facilities and multiple coverage levels. Gendreau et al. (1997) proposed the Double Standard Model (DSM) which seeks to maximize the total demand that is covered by at least two service facilities. Erdemir et al. (2010) extended the coverage definition to allow for response time and total service time in complex coverage situations, such as those with air and ground EMS vehicles. Extensions for both the LSCP and the MCLP were presented: the Set Cover with Backup Model (SCBM) and the Maximal Cover for a Given Budget Model (MCGBM), respectively.

Applications of the MCLP with backup considerations outside the EMS literature have been pursued by Malcolm and ReVelle (2005) and Curtin et al. (2010). Malcolm and ReVelle (2005) presented the Maximal Species Backup Coverage (MSBC) for the protection of endangered species. Curtin et al. (2010) introduced the Police Patrol Area Covering (PPAC), where GIS and crime data is integrated to determine the efficient distribution of police patrols.

2.2.2 Covering Models: Unavailability Formulations

Two different approaches have been applied that include unavailability considerations in covering formulations. In this body of literature, unavailability and reliability are treated as equivalent concepts. In the first approach, chance constraints are included. Chapman and White (1974) and Aly and White (1978) used structured chance constraints to integrate the unavailability of service into the LSCP by accounting for the busy fraction of vehicles and a service reliability factor. The busy fraction is analogous to the detector unavailability. ReVelle and Hogan (1988, 1989a) extended this via the Probabilistic Location Set Covering Problem (PLSCP) by using a methodology to estimate zone specific busy fractions. PLSCP concepts were integrated with the MCLP in the Maximum Availability Location Problem (MALP) (ReVelle and Hogan, 1989b). Backup and probabilistic versions of the Facility Location, Equipment-Emplacement Technique (FLEET) are presented by Schilling et al. (1979). ReVelle (1989) provides a review of the formulations previously mentioned. Based on the PLSCP, Ball and Lin (1993) proposed the REL-P formulation, a binary integer programming problem where the stochasticity of the EMS allocation is more explicitly handled.

The second approach employs the unavailability of the facilities as a weighting factor in the objective function. Daskin (1982, 1983) formulated and solved the Maximum Expected Covering Location Problem (MEXCLP), an extension of the MCLP. Assuming the reliability of the facilities in service follow a binomial distribution, the problem seeks to maximize the expected value of demand coverage. Batta et al. (1989) revisited the MEXCLP and proposed the Adjusted MEXCLP (AMEXCLP) in order to relax some of the assumptions regarding the independence of busy probabilities. Repede and Bernardo (1994) presented a MEXCLP with time variation (TIMEXCLP). MEXCLP and probabilistic concepts were applied as well to the FLEET model by Bianchi and Church (1988), ReVelle and Marianov (1991) and Marianov and ReVelle (1992).

Multiple, excess, backup, and expected coverage, and the relation between them, are reviewed by Daskin et al. (1988). Li et al. (2011) provide a review of the LSCP, MCLP, DSM, MEXCLP and MALP, as well as extensions and optimization techniques.

2.2.3 PMP: Redundant Coverage and Unavailability Formulations

Initial considerations regarding facility unavailability in median problems were presented by Berman and Larson (1982). The problem was approached from a queuing point of view. The probability that all facilities are busy is added as a weighting term to modify the traditional PMP objective. Work regarding the availability of individual facilities in the PMP was presented by Drezner (1987) (Unreliable PMP). The probability that a given facility r was active, given that the $r-1$ closest facilities were not, was included as a weighting factor in the objective function. Heuristic solution methods for the Unreliable PMP were proposed and tested by Lee (2001). Weaver and Church (1985) and Pirkul (1989) presented extensions to the PMP and the Uncapacitated Fixed-Charge Location Problem (UFLP), where customers demands are satisfied by multiple facilities providing fixed partial coverage. The UFLP is a variation of the PMP where the number of facilities is a decision variable, not a constraint.

Drezner's approach has been extended to explicitly include the probability of facility failure (Krass et al., 2003; Menezes et al., 2003a,b; Snyder, 2003; Snyder and Daskin, 2002, 2003, 2005). In this work, we build from the previous work of Snyder and Daskin

(2005), which proposed and tested the Reliability PMP (RPMP) and the Reliability Fixed-Charge Location Problem (RFLP). The RFLP is the unavailability version of the UFLP. Unlike the UPMP, the RPMP explicitly considers the unavailability of each facility by making use of the binomial assumption proposed by Daskin (1982, 1983). The concept of backup facilities was also integrated into the formulation; customers are assigned to backup facilities when closer facilities have failed. An optimal Lagrangian relaxation algorithm was proposed to solve both problems.

The binomial distribution allows for the assignment of an individual unavailability probability to each facility, but assumes that all of them are equal. Later work has been focused mainly on relaxing this assumption via nonlinear formulations (Berman et al., 2007; Berry et al., 2009a; Cui et al., 2010; Shen et al., 2011). In the context of facility location, Berman et al. (2007) presented the Median Problem with Unreliable Facilities (MPUF) along with a discussion of nodal optimality and asymptotic results for this problem. Cui et al. (2010) developed the Reliable Uncapacitated Fixed Charge Location Problem (RUFL). This formulation was linearized into the LRUFL and solved via a Lagrangian relaxation algorithm. Furthermore, a Continuum Approximation (CA) model was developed and tested for large instances of the problem. Extensions to the RFLP have been proposed as well (Lim et al., 2010; Shen et al., 2011). Reviews on supply chain disruption and unavailability, including PMP extensions, are provided by Snyder and Daskin (2007) and Snyder et al. (2010). Berry et al. (2009a) extended the idea of imperfect detectors into water networks. A nonlinear formulation, impSP, along with six solution strategies were proposed and tested. Further discussion of the work by Berry et al. (2009a) and Berman et al. (2007) is presented in Section 7.

3. MIXED-INTEGER LINEAR PROGRAMMING FORMULATION INCLUDING UNAVAILABILITY (SP-U) *

Previous work in the gas detector placement problem (Legg et al., 2012a,b; Legg, 2013; Legg et al., 2013) did not considered a key feature associated with gas detector equipment: The possibility of detection failure. Gas detectors are prone to a number of failure modes that include failure to function on demand and no output signal. Additionally, due to maintainability considerations, gas detectors can be be offline due to preventive maintenance and testing, or absence due to repairs or replacement. In this section, we extend our initial formulation (SP) to include these considerations. An MILP formulation is presented in the next section. This formulation, SP-U, explicitly accounts for the possibility that the detectors are not able to perform the intended function when service is demanded. This is achieved by including the concept of detection levels and detector unavailability. Formulation SP-U is based on the Reliability P-Median Problem (RPMP) proposed by Snyder and Daskin (2005) for facility location models. Results for the proposed formulation are presented and discussed in Section 3.2, and they are compared with those previously obtained by Legg et al. (2012a,b); Legg (2013); Legg et al. (2013). The explicit treatment of unavailability in the formulation results in changes to the optimal detector placement. The possibility of false negative cases is a common concern in the industry, and this modification constitutes a step forward in the realism of the problem formulation. This work can be found in the papers by Benavides-Serrano et al. (2012) and Benavides-Serrano et al. (2014). Section 3.3 provides a summary of the section.

---

* Part of this section is reprinted with permission from ”A Stochastic Programming Approach for the Optimal Placement of Gas Detectors: Unavailability and Voting Strategies” by Benavides-Serrano, A. J., Legg S. W., Vázquez-Román R., Mannan, M. S., and Laird C. D., 2014. Industrial & Engineering Chemistry Research, Copyright 2013 by American Chemical Society. ACS Articles on Request author-directed link: `http://dx.doi.org/10.1021/ie401369v`.

3.1 SP-U Formulation

The RPMP extended the PMP formulation by taking into account the probability that a small distributor will occasionally defect from the company or go out of business, thereby requiring a shift in facility assignment. In an analogous way, our formulation incorporates the probability that a detector will not be able to perform its intended function, i.e., the detector unavailability. It is important to keep in mind that since the dynamic behavior of the detector unavailability is not incorporated, the pertinent unavailability values to use in the formulation are the time-averaged values ($\bar{q}$). Since detectors are prone to failure, the primary detector may fail, requiring a second detector to signal detection. The same concept applies for higher coverage levels. If we assume that the time-averaged unavailability, $\bar{q}$, of all detectors is the same, the probability that the detector in a given detection level is responsible for signaling will follow a binomial distribution. That is, it will follow the discrete probability distribution of the number of failures (r in our case) in a sequence of Bernoulli trials. This corresponds to the failure of detectors in the first r levels, followed a successful detection in the next level. The probability mass function for the binomial distribution is presented in Equation 3.1.

$$w_1(r, \bar{q}) = \bar{q}^r(1 - \bar{q}) \tag{3.1}$$

The MILP formulation (SP-U) is presented below. Notation for the formulation is provided in Table A.1. The Pyomo model file containing formulation SP-U is presented in Appendix G.

$$\min \quad \sum_{a\in A}\alpha_a \sum_{i\in\mathcal{L}_a}\sum_{r\in\mathcal{R}_{a,i}} d_{a,i} w_1(r,\bar{q})\boldsymbol{x_{a,i,r}} \tag{3.2a}$$

s.t.

$$\sum_{i\in\mathcal{L}_a}\boldsymbol{x_{a,i,r}} = 1 \qquad \forall\, a\in A, r\in R \tag{3.2b}$$

$$\sum_{l\in L}\boldsymbol{s_l} \leqslant p \tag{3.2c}$$

$$\sum_{r\in\mathcal{R}_{a,i}}\boldsymbol{x_{a,i,r}} \leqslant \boldsymbol{s_i} \qquad \forall\, a\in A, i\in\mathcal{L}_a\backslash\mathcal{D}_a \tag{3.2d}$$

$$\boldsymbol{s_l}\in\{0,1\} \qquad \forall\, l\in L \tag{3.2e}$$

$$0\leqslant\boldsymbol{x_{a,i,r}}\leqslant 1 \qquad \forall\, a\in A, i\in\mathcal{L}_a, r\in\mathcal{R}_{a,i} \tag{3.2f}$$

The problem has two sets of decision variables: s_l and $x_{a,i,r}$. The first one, s_l, indicates if a detector is allocated at location l (s_l=1, and 0 otherwise). The second, $x_{a,i,r}$, indicates that scenario a is detected at coverage level r by location i ($x_{a,i,r}$=1, and 0 otherwise). Formulation SP-U and its extensions could accommodate co-location effects (i.e., multiple detectors in the same candidate detector location) by allowing s_l to take any non-negative integer value. However, for the results presented in this work, it is going to be assumed that only one detector is allowed per candidate detector location. The damage coefficient, $d_{a,i}$, corresponds to the consequence associated with scenario a prior to its detection at location i. The coverage level r indicates the detection sequence. For example, consider the case where the damage coefficients correspond to detection time. If a detector is selected at location i, and it is the first to encounter scenario a, this detector is assigned coverage level 0. The second selected detector to encounter scenario a is assigned coverage level 1, and so forth. Three sets are defined for the problem: the set L of N potential detector locations, the set A of M potential hazardous scenarios and the set R of C coverage levels. The number of coverage levels can never be larger than the number of detectors. Ideally, to ensure all potential coverage levels are included, the formulation should consider as many coverage levels as there are detectors. However, as indicated by Snyder and Daskin (2005), when the time-averaged unavailability is reasonably low, the probabilities associated with higher coverage levels quickly tend to zero, there is often no need to consider more than a few levels. In this work, we considered 5 coverage levels (C=4). Validation of the previous statement for our data sets, and justification of the number of coverage levels chosen, is presented in Section 6. Sets $\mathcal{L}_a$ and $\mathcal{D}_a$ are defined for each scenario. The sets $\mathcal{L}_a$ are defined as the list of locations that are able to detect scenario a. To account for scenarios that go undetected, dummy variables are added to $x_{a,i,r}$ with damage coefficient values that penalize any undetected leak scenarios. The sets $\mathcal{D}_a$ contain indices corresponding to these dummy variables. Enough dummy variables are added to allow for detection failure of each scenario at all the coverage levels. In spirit, these are similar to dummy detector locations that are always selected. Finally, subsets $\mathcal{R}_{a,i}$ are defined for each pair of scenario a and location i. These correspond to the set of backup levels in R that location i can occupy given its associated damage coefficient $d_{a,i}$, where the initial detection level is 0, the second is 1, and so on. For the results presented, the dummy damage coefficient (d_{max}) was set to be 10 seconds greater than the largest damage coefficient for the data set. The maximum number of detectors that can be allocated is given by p. The probability of a given scenario a is represented by α_a. The value of $\bar{q}$ corresponds to the time-averaged unavailability of the given type of detector, i.e., the probability that the detector will not perform the intended function when needed.

The objective function (3.2a) minimizes the expected value of the overall damage coefficient considering the probability of failed detection. The product of the damage coefficient $d_{a,i}$ and the probability of scenario α_a would result in a measure of risk. The first constraint (7.2b) guarantees that every scenario a is detected by a detector at each coverage level in R (where the addition of dummy variables relaxes this constraint). Equation (7.2c) limits the number of detectors that can be deployed. Equation (7.2d) links the existence of a detector to the coverage levels for a given scenario a. Implicitly, this constraint serves two purposes. First, it requires that a given scenario a can be counted as detected by one, and only one, coverage level r for a particular detector location. Secondly, the constraint ensures that location i can only be the first to detect leak scenario a at coverage level r if there is a detector placed at location i. The objective function guarantees that, for each scenario a, the detector assigned to the coverage level r will always be a detector with a smaller damage coefficient than the detector assigned to coverage level $r + 1$. In this way, the objective function guarantees that the detectors are assigned coverage levels in a sequence of increasing damage coefficients. In our formulation constraint (7.2d) replaces two constrains from the formulation of Snyder and Daskin (2005) (equations 3 and 5 in the RPMP formulation). Constraint (7.2d) serves the same purpose but improves memory usage and solution times.

In the RPMP model proposed by Snyder and Daskin, a term in the objective function accounts for those facilities loyal to the firm, i.e., those that will not fail to supply their services to the firm under any circumstances. Their formulation also allows for the possibility of opening a non-failable emergency facility in the case that no facilities are available to serve the costumer. The addition of such a facility may result in a penalty such as a low sales cost or a cost of purchasing product from a competitor. In the detector layout problem, the detectors used are all susceptible to failure, and the concept of a non-failable location is not applicable. Therefore, these parts of the formulation were removed.

3.2 Numerical Results

In this section, results for the SP-U formulation for different time-averaged unavailability values (0, 0.1 and 0.2) are compared. These correspond to representative values of the real gas detector unavailabilities in the process industries (see Section 1.3.2). Results were generated using data set A previously presented in Section 1.3.1. This corresponds to the same data employed by Legg et al. (2012a,b); Legg (2013); Legg et al. (2013). For this data set, a typical instance of the formulation has 26792 variables and 6195 constraints. The time to solution was in all cases less than 10 seconds. Five coverage levels (C=4) were used for all the results reported. The expected time to detection results account for the low probability event that 5 detection levels are all present but each fails to detect the event. That is, $d_{max}\,(1 - \sum_{r\in R} w_1(r,\bar{q}))$ was added to the final objective function value.

A leak scenario was considered detected at a given detector location when the simulated gas cloud reached a concentration greater than 10% of the Lower Flammability Limit (LFL) value. Detection times for each of these scenarios and locations were recorded, and these times were used as the damage coefficients, $d_{a,i}$, resulting in problem formulations that seek to minimize the expected time to detection across all the scenarios. Ideally, the objective function to minimize should be the overall risk to the facility. This can be easily accommodated into our formulations by using $d_{a,i}$ as maximum risk incurred for scenario a prior to detection at location i. A significant amount of data is necessary to compute the consolidated risk to the facility. Given the data available for our analysis, the performance metric used in this work, our previous work (Legg et al., 2012a,b; Legg, 2013; Legg et al., 2013) considers the minimization of the expected detection time over all gas leak scenarios as the objective function. This approach is in accordance to the principal objective of gas detection systems. Although the evaluation of the risk reduction capability of the gas detection systems is the exception rather than the norm, it is possible to find wide agreement regarding the principal objective of the gas detection system: to provide fast and reliable detection of gas accumulations before they reach concentration and sizes which could pose a risk to the facility and it's occupants. That is, identifying accidental releases as fast as possible, so that proper countermeasures can be initiated (International Electrotechnical Commission (IEC), 2007; International Society of Automation (ISA), 2010; Norsk Sokkels Konkuranseposisjon (NORSOK), 2008). This point of view is shared by recent performance analyses where gas detection systems effectiveness is commonly evaluated in terms of time to and probability of detection Bratteteig et al. (2011); Kelsey et al. (2002, 2005). As previously mentioned, gas detection systems have interfaces with several other safety systems. These include the Emergency Shut Down (ESD), Blow Down (BD), Ignition Source Control (ISC), ventilation, Public Address (PA) and alarms system, and fire fighting systems (Norsk Sokkels Konkuranseposisjon (NORSOK), 2008). Minimizing the time to detection and guaranteeing reliable detection allows for effective corrective actions and emergency response, including ignition source control, containment, evacuation of personnel, or other actions appropriate to the specific situation.

The same probability of occurrence, $\alpha_a = 1/M$, was utilized for each dispersion scenario a in a given data set; M corresponds to the number of scenarios in the respective data set. Expert advice, databases or more rigorous approaches like Layer Of Protection Analysis (LOPA) and Quantitative Risk Assessment (QRA) should be used to obtain these scenario probabilities in real world applications.

The effect of the increasing number of detectors on the expected time to detection and the fraction of scenarios covered is presented in Figure 3.1. The fraction of scenarios covered represents the fraction of scenarios that have at least one of the selected detector locations in subset $\mathcal{L}_a$ (excluding the dummy locations). With $\bar{q}$=0, the SP-U formulation is equivalent to the initial SP formulation, yielding the same results presented in Legg et al. (2012a,b); Legg (2013).

As expected, when detector layouts with the same number of detectors and different time-averaged unavailability are compared, the layout with the smallest unavailability will have the lowest objective. An increase in the unavailability will result in an increased weighting of the backup coverage levels, which are associated with larger damage.

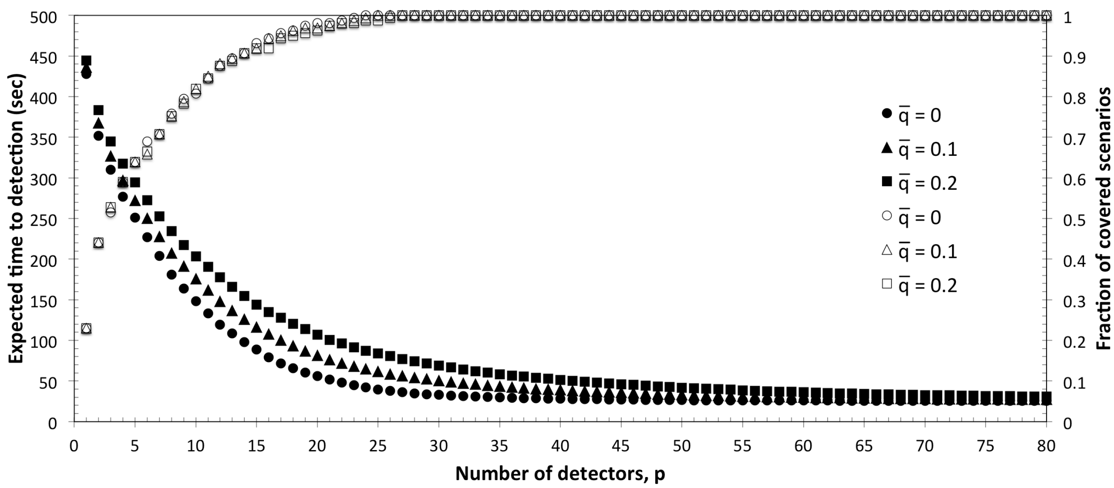


Figure 3.1: SP-U formulation results: Expected detection time (●, ▲ and ■) and fraction of leak scenarios covered (○, △ and □) as a function of the maximum number of allowed detectors, *p*.

An overhead view of the optimal placement of 25 detectors using the SP, SP-U with $\bar{q}$=0.1 and SP-U with $\bar{q}$=0.2 formulations is presented in Figure 3.2. Only 19 of the 25 detectors placed by the $\bar{q}$=0 case are used by the $\bar{q}$=0.1 case. Likewise, 15 of them are part of the solution of the $\bar{q}$=0.2 case. Furthermore, the results of the $\bar{q}$=0.1 and $\bar{q}$=0.2 cases share 16 detector locations. This shows that consideration of unavailability in the formulation alters the final detector placement. For this problem, all scenarios were detected when the number of detectors was greater than or equal to 24, 25, and 27 for $\bar{q}$=0, $\bar{q}$=0.1, and $\bar{q}$=0.2 respectively. This indicates that for our data set, and by means of our formulation, there will be little loss of overall scenario coverage due to the necessity of providing redundant coverage.

To assess the true value of considering detector unavailability in the optimization formulation, the differences between the optimal SP-U solution and the value of the SP-U objective calculated with the solution placement from the SP formulation of Legg et al. (2012a); Legg (2013) are reported in Figure 3.3. Expected time to detection values are

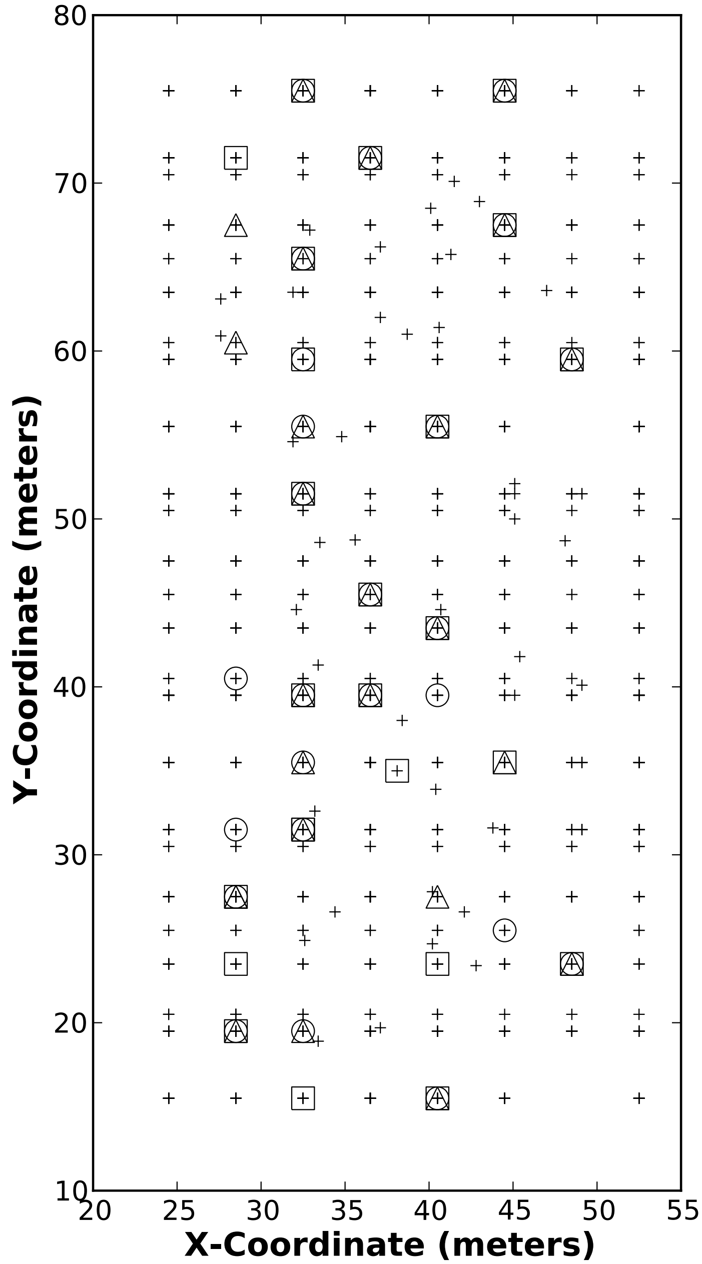


Figure 3.2: SP-U formulation results: Detector locations in the process facility, p=25. Potential detector locations are represented by $+$, ○ represent the detectors placed by the $\bar{q}$=0 case (SP formulation), △ represent the detectors placed by the $\bar{q}$=0.1 case, and □ represent the detectors placed by the $\bar{q}$=0.2 case.

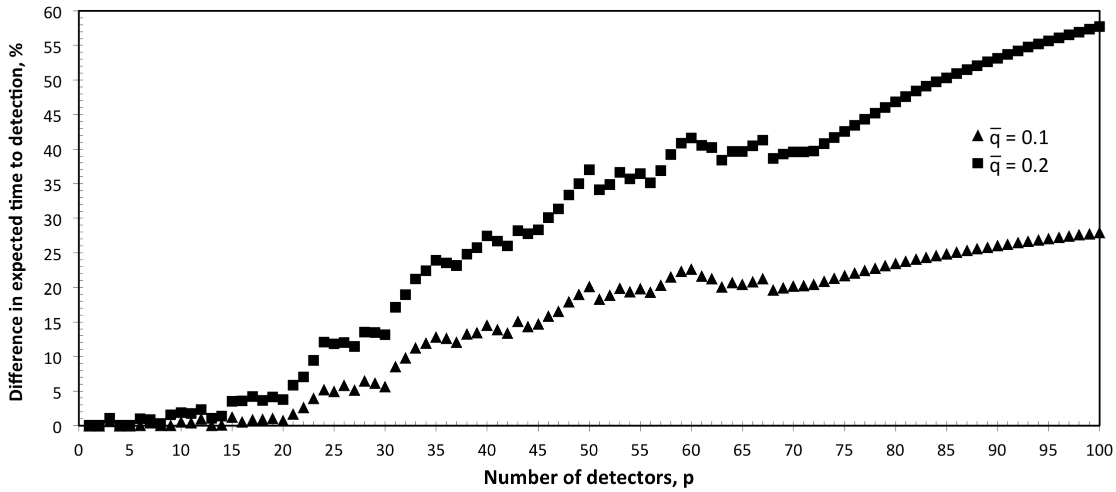


Figure 3.3: Comparison showing the value of SP-U formulation. This figure shows the expected time to detection from the placements produced by SP when unavailability is considered. This metric is displayed as percent difference from the SP-U results. Expected detection times for $\bar{q}$=0.1 and $\bar{q}$=0.2 are shown as ▲ and ■, respectively.

presented as percentage differences between the optimal SP-U ($\bar{q}$=0.1 and $\bar{q}$=0.2) solution, and the value of the SP-U objective calculated with the solution placement from the SP formulation ($\bar{q}$=0). Initially, for a low number of detectors, both the SP and SP-U formulations will result in similar placements. The effect of relaxing the assumption of perfect detectors becomes more noticeable as the number of detectors are increased. Note that the SP formulation makes no additional improvement in its objective for $p > 72$. Past this point the SP formulation will not benefit from the allocation of more detectors, while the SP-U formulation will continue to benefit from the allocation of new detectors by providing additional backup coverage. When 100 detectors are allocated, the expected time to detection is 28% higher for the $\bar{q}$=0.1 case and 57% higher for the $\bar{q}$=0.2 case. This figure shows that it is important to consider detector unavailability when performing the optimal placement.

3.3 Summary

An extension of a previously developed MILP approach for optimal gas detector placement was presented. Extension (SP-U) considers the effect of the detector not being able to perform its intended function by including the concept of detection levels and detector unavailability. This formulation guarantees that the increase in the expected time to detection due to false negative (non-detection) cases is minimal. For our case study there is little loss of overall scenario coverage when considering unavailability. However, the optimal placements are affected. Furthermore, if we neglect detector unavailability when optimizing the layout, and then use this layout in a real setting where unavailability exists, the actual expected time to detection is significantly deteriorated. The work presented here constitutes a step forward toward the achievement of a realistic detector placement formulation that includes current industrial practice for these important safety systems.

4. MIXED-INTEGER LINEAR PROGRAMMING FORMULATION INCLUDING UNAVAILABILITY AND VOTING EFFECTS (SP-UV) *

Previous work (Legg et al., 2012a,b; Legg, 2013; Legg et al., 2013) in the gas detector placement problem did not consider the common operating requirement for voting logic. In reality gas sensors are prone to false positives. The solution usually implemented in the process industries to tackle this issue is to require additional confirmation from several detectors before emergency actions are triggered. This additional confirmation requirement is known as voting. A discussion regarding voting logic schemes was presented in Section 1. In this section, we extend our previous formulation (SP-U) to include these considerations. An MILP formulation is presented in the next section. This formulation, SP-UV, generalizes SP-U by making use of the negative binomial distribution in order to take the voting logic effects into account. Results for the proposed formulation are presented and discussed in Section 4.2, and they are compared with those previously obtained by Legg et al. (2012a,b); Legg (2013); Legg et al. (2013) and in the previous section for formulation SP-U. When a voting policy for the detection and mitigation system is in place, explicitly including voting logic considerations result in changes to the optimal detector placement and significant improvements in the expected time to detection. This work can be found in the paper by Benavides-Serrano et al. (2014). Section 4.3 provides a summary of the section.

---

* Part of this section is reprinted with permission from "A Stochastic Programming Approach for the Optimal Placement of Gas Detectors: Unavailability and Voting Strategies" by Benavides-Serrano, A. J., Legg S. W., Vázquez-Román R., Mannan, M. S., and Laird C. D., 2014. Industrial & Engineering Chemistry Research, Copyright 2013 by American Chemical Society. ACS Articles on Request author-directed link: `http://dx.doi.org/10.1021/ie401369v`.

Part of this section is reprinted with permission from "A Quantitative Assessment on the Placement Practices of Gas Detectors in the Process Industries" by Benavides-Serrano A. J., Mannan, M. S, and Laird C. D., 2014. To appear in the Journal of Loss Prevention in the Process Industries.

4.1 SP-UV Formulation

Formulation SP-U is now extended to reflect the characteristics of a voting logic, i.e., requiring confirmation by k detectors before the existence of a gas leak is acknowledged. As before, the objective function (4.2a) seeks to minimize the expected value of the overall damage considering the probability of failed detection. Once again, since the dynamic behavior of the detector unavailability is not incorporated, the pertinent unavailability values to use in the formulation are the time-averaged values ($\bar{q}$). However, this formulation takes into account that k detectors must confirm the leak existence before action is taken. By incorporating this confirmation requirement, and assuming that the time-averaged unavailability of all detectors is the same, the probability that the detector in a given detection level will be responsible for signaling can be modeled by using a negative binomial distribution. That is, it can be modeled by the discrete probability distribution of the number of failures (r in our case) in a sequence of Bernoulli trials before a specified non-random number of successes (denoted k) occur. The probability mass function for the negative binomial distribution is presented in Equation 4.1).

$$w_2(r, k, \bar{q}) = \binom{r+k-1}{r} \bar{q}^r (1-\bar{q})^k \tag{4.1}$$

The MILP model proposed (SP-UV) is presented below. Notation for the formulation is provided in Table A.1. The Pyomo model file containing formulation SP-UV is presented in Appendix H.

$$\min \quad \sum_{a\in A}\alpha_a \sum_{i\in\mathcal{L}_a}\sum_{r\in\mathcal{R}_{a,i}} d_{a,i} w_2(r,k,\bar{q})\boldsymbol{x_{a,i,r+k-1}} \tag{4.2a}$$

s.t.

$$\sum_{i\in\mathcal{L}_a} \boldsymbol{x_{a,i,r}} = 1 \qquad \forall a \in A, r \in R \tag{4.2b}$$

$$\sum_{l\in L} \boldsymbol{s_l} \leqslant p \tag{4.2c}$$

$$\sum_{i\in\mathcal{L}_a} d_{a,i}\boldsymbol{x_{a,i,r}} \leqslant \sum_{i\in\mathcal{L}_a} d_{a,i}\boldsymbol{x_{a,i,r+1}} \qquad \forall a \in A, \{r|r \in R, r \leqslant k-1\} \tag{4.2d}$$

$$\sum_{r\in\mathcal{R}_{a,i}} \boldsymbol{x_{a,i,r}} \leqslant \boldsymbol{s_i} \qquad \forall a \in A, i \in \mathcal{L}_a \backslash \mathcal{D}_a \tag{4.2e}$$

$$\boldsymbol{s_l} \in \{0,1\} \qquad \forall l \in L \tag{4.2f}$$

$$0 \leqslant \boldsymbol{x_{a,i,r}} \leqslant 1 \qquad \forall a \in A, i \in \mathcal{L}_a, \{r|r \in \mathcal{R}_{a,i}, r \geqslant k\} \tag{4.2g}$$

$$\boldsymbol{x_{a,i,r}} \in \{0,1\} \qquad \forall a \in A, i \in \mathcal{L}_a, \{r|r \in \mathcal{R}_{a,i}, r < k\} \tag{4.2h}$$

Equations 4.2b-4.2c and 4.2e-4.2g are equivalent to those in the SP-U formulation. Two issues arise since the first k detection levels are no longer part of the objective function. First, the proper order of detection levels is no longer guaranteed (since the corresponding damage coefficient no longer appears in the objective), and an additional constraint (4.2d) was added to fulfill this requirement. Furthermore, there is no longer a guarantee that the continuous variables $x_{a,i,r}$ will converge to integer values, therefore, these variables are constrained to be $0-1$ in (4.2h).

4.2 Numerical Results

In this section, results for the SP-UV formulation with $\bar{q}$=0.1 and two different voting strategies (k=1 and k=2) are presented. These correspond to a representative value of the real gas detector unavailability and two of the most used voting policies in the process industries, respectively. Refer to Section 1 for a discussion on the reasoning behind these values. Results were generated using data set A previously presented in 1.3.1. This corresponds to the same data employed by Benavides-Serrano et al. (2014); Legg et al. (2012a,b); Legg (2013); Legg et al. (2013). Damage coefficients and scenario probabilities were calculated following the same considerations presented in Section 3.2.

For this data set, a typical instance of the formulation has 26792 variables and 6517 constraints. With respect to SP-U, the number of variables remains the same, while Equation (4.2d) added 322 constraints. Depending on the number of detectors allowed, the time to solution varied from a few minutes to a couple of days. Five coverage levels (C=4) were used for all the results reported. The expected time to detection results account for the low probability event that 5 detection levels are all present but each fails to detect the event. That is, $d_{max}\left(1-\sum_{r\in R} w_2(r,k,\bar{q})\right)$ was added to the final objective function value.

The effect of the increasing number of detectors on the expected time to detection and the fraction of scenarios covered is presented in Figure 4.1. The fraction of scenarios covered represents the fraction of scenarios that have at least k of the selected detector locations in $\mathcal{L}_a$ (excluding the dummy locations). The SP-UV formulation with k=1 is equivalent to the SP-U formulation, and yields the same results presented in Figure 3.1.

As anticipated, for detector layouts with the same number of detectors and probabilities of failure but different voting strategies, the case with the simpler voting logic, i.e. lower k, will have an smaller expected time to detection. Due to the redundancy requirement added by the SP-UV formulation, k detector locations must confirm the existence of a gas leak before an action is initiated. Therefore, the lowest achievable detection time for a scenario is that given by the k-th detector. Since more detectors are required before a leak scenario triggers an action, there is an increased likelihood of undetected scenarios for a given detector budget. For detector layouts with the same number of detectors, equal probabilities of failure, and different voting strategies, the placements corresponding to

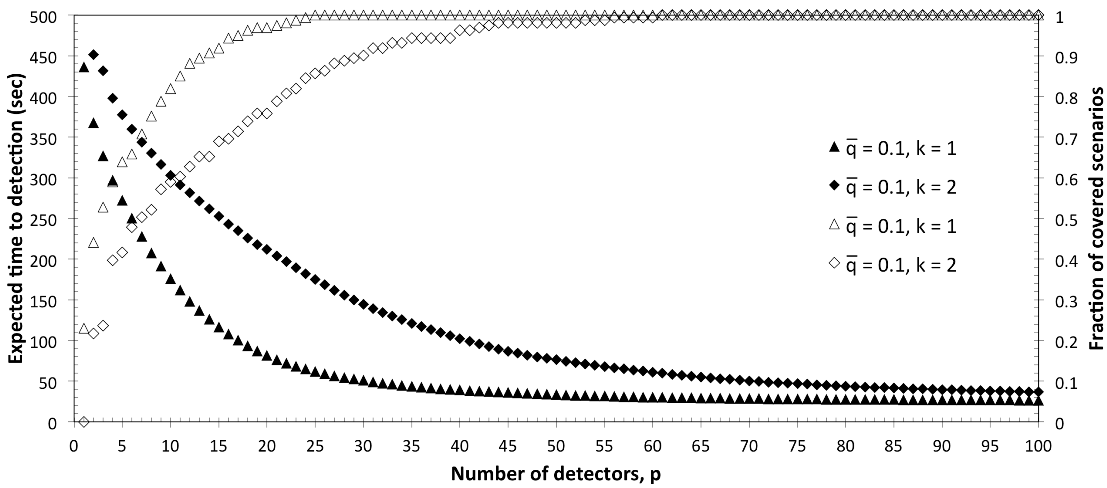


Figure 4.1: SP-UV formulation results: Expected detection time (▲, ◆) and fraction of leak scenarios covered twice (△, ◇) as a function of the maximum number of allowed detectors, p.

simpler voting logic will always have an equal or larger fraction of scenarios detected. Again, this is an expected consequence of the redundant confirmation requirement.

For the data set, the minimum number of detectors required in order to cover all of the scenarios for a 2-o-o-p voting logic is 59, which corresponds to an expected time to detection of 62 seconds. One might expect that the number of detectors required to cover all scenarios for the k=2 case would double, or close to double, that of the number of detectors required to cover each scenario for the k=1 case. However, the objective function in this case is to minimize the expected time to detection, not to maximize coverage, and the formulation may prioritize additional backup levels on high impact scenarios over coverage of all scenarios, meaning more than double the number of detectors may be required. Figure 4.2 presents an overhead view comparison between the optimal placement of 59 detectors using the SP, SP-U and SP-UV formulations. The number of optimal locations shared by the SP and SP-UV ($\bar{q}$=0.1, k=2) formulations is 18. For the SP-U ($\bar{q}$=0.1, k=1) and SP-UV ($\bar{q}$=0.1, k=2) formulations the number of shared locations is 28.

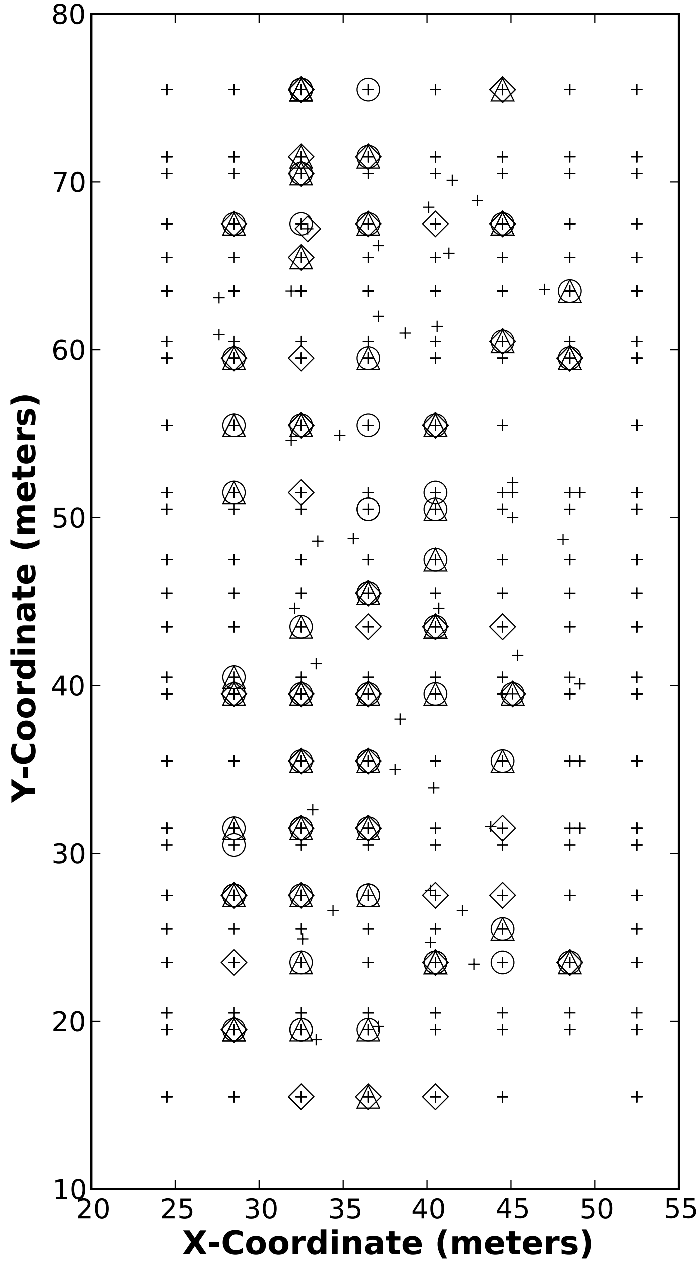


Figure 4.2: SP-UV formulation results: Detector locations in the process facility, p=59. Potential detector locations are represented by $+$, ○ represent the detectors placed by the SP formulation, △ represent the detectors placed by the SP-U formulation with $\bar{q}$=0.1, and ◇ represent the detectors placed by the SP-UV formulation with $\bar{q}$=0.1 and k=2.

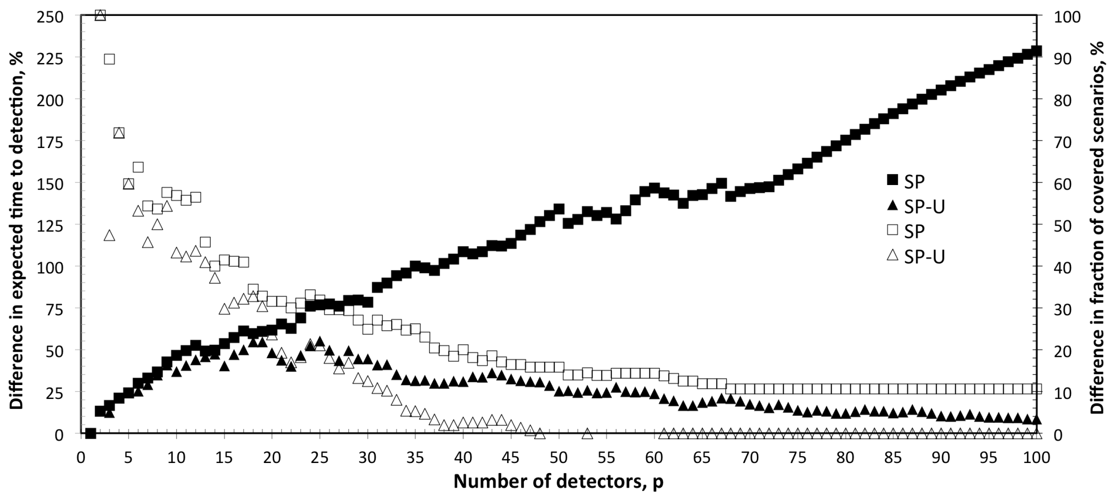


Figure 4.3: Comparison showing the value of SP-UV formulation. This figure shows the expected time to detection and fraction of covered scenarios from the placements produced by SP and SP-U when unavailability and voting is considered. These metrics are displayed as percent differences from the SP-UV results. Expected detection times for SP and SP-U are shown as ■ and ▲ respectively. Fraction of leak scenarios covered twice for SP and SP-U are shown with □ and △ respectively.

These correspond to differences in 70% and 53% of the allocated detectors, respectively. It is clear that considering the effect of voting strategies substantially changes the detector placement.

Actual process facilities regularly require voting strategies. However, formulation SP-UV is significantly more complex and computationally demanding than SP and SP-U. To asses the true benefit of the SP-UV formulation, we compare the optimal expected time to detection from SP-UV with the expected time to detection that arises if we determine optimal placements using the original SP formulation of Legg et al. (2012a) and SP-U formulation, but evaluate those placements considering unavailability and voting (using the objective from SP-UV). The same time-averaged detector unavailability ($\bar{q}$=0.1) was used for the SP-U and SP-UV cases. The resulting differences in the expected time to detection and the fraction of scenarios covered between the optimal SP-UV solution and the evaluation of the solutions from SP and SP-U on the SP-UV objective are reported in Figure 4.3. The expected time to detection differences are presented as a percentage of the optimal expected time to detection obtained with the SP-UV formulation. Differences in the fraction of scenarios covered are presented as a percentage of the fraction of scenarios covered obtained with the SP-UV formulation.

When the SP results are evaluated using a 2-o-o-p voting logic, the effects of not considering unavailability and voting explicitly become evident. As shown in Figure 4.3, the difference in the fraction of scenarios detected for the SP formulation is considerable. For our data set, for a number of detectors equal to 100, one out of ten scenarios will remain uncovered. This is again due to the fact that for the SP formulation, further addition of detectors past 72 will provide no improvement in the expected detection time. Due to this same reason, the expected time to detection follows the same pattern as Figure 3.3. Above 72 detectors the difference will again grow steadily. When 100 detectors are allocated, the expected time to detection arising from the SP placement is already 228% higher than that arising from the SP-UV placement.

When the single voting logic (1-o-o-p) results are evaluated using a 2-o-o-p voting logic, the difference between the fractions of scenarios covered will be considerable when few detectors are placed since formulation SP-UV favors backup coverage of high impact scenarios. However, for some placement results with a number of detectors ranging from 50 to 60, the difference will actually be negative (never less than -1%). This is again attributed to the fact that the SP-UV formulation prioritizes the placement of detectors to provide additional backup coverage of high impact scenarios. For the SP-U formulation, the curve for the time difference results increases initially and then decreases. Initially, the SP-U formulation will preferentially provide a primary coverage level of the scenarios. But, when the amount of detectors is sufficiently larger, detectors will be preferentially employed as backup coverage on high impact scenarios. At this point, SP-U and the SP-

UV formulations are both essentially improving the second coverage level. Because both formulations now share a similar goal, the difference in the expected time to detection between both solutions begins declining. However, this correction is not sufficient for formulation SP-U to provide answers that are close to the optimal answer obtained by using formulation SP-UV. This figure shows the importance of explicitly considering voting policies when performing the optimal placement.

On a final note, it is important to acknowledge the increase in the computational effort arising from the explicit consideration of voting into the formulation. Given the numerical results, this additional computational effort can be easily justified. First, this is a design problem and the solution time is negligible when compared to the amount of time already invested at this point in the detection and mitigation system design, the dispersion simulations assessment, and the simulations. Finally, the amount of economic losses for which these detection and mitigation systems are developed, and the cost of the system itself, easily exceed the cost of the additional computational effort required.

4.3 Summary

An extension of the previously developed SP-U formulation for optimal gas detector placement was presented. Formulation SP-UV extends SP-U to incorporate a voting scheme that explicitly requires k detectors to confirm the gas leak existence before actions are initiated. Voting logic shields the system against false alarms. As with the previous formulation, the optimal layouts from SP-UV differed significantly from those obtained with SP and SP-U. Neglecting the voting strategies in the optimization, that is, optimizing with SP or SP-U and applying that layout in a real setting with voting, causes a significant decrease in performance. These results clearly indicate the importance of explicitly including the detection and mitigation system voting policies into our formulations. Furthermore, while formulation SP-UV originates from the need of including detection and mitigation system voting, it is application can be easily extended by analogy to other facility siting problems in the operations research literature.

5. QUANTITATIVE ASSESSMENT AGAINST CURRENT GAS DETECTOR PLACEMENT PRACTICES IN THE PROCESS INDUSTRIES *

As introduced in Section 1.2, most detector placement strategies for gas detection systems are prescriptive approaches supported by qualitative considerations and rules of thumb rather than quantitative metrics based on the dispersion behavior of the possible leak scenarios. The work presented by Legg et al. (2012a,b, 2013) and Benavides-Serrano et al. (2014), and summarized in Sections 2-4, proposed, developed, and validated the use of stochastic programming formulations in order to take further advantage of the quantitative information provided by dispersion simulations. These formulations identify the gas detector layout that minimizes the expected value of the overall damage (i.e., the minimization of a risk metric) given a set of dispersion scenarios. Results demonstrated the potential and suitability of numerical optimization to approach the gas detector placement problem while rigorously considering its inherent uncertainties.

Motivated by this evidence, this section strives to answer the following questions: Are current practices effective at designing gas detection systems? What is the value of dispersion data and numerical optimization techniques in terms of detection system performance? The rest of this section is organized as follows. In Section 5.1 we describe our assumptions, and develop implementations of the placement approaches described in Section 1.2. Each of the approaches in Section 5.1 is then applied to the generated data sets. Their performance is presented and analyzed in Section 5.2. Section 5.3 provides general conclusions and a summary of the section.

* Part of this section is reprinted with permission from “A Quantitative Assessment on the Placement Practices of Gas Detectors in the Process Industries” by Benavides-Serrano A. J., Mannan, M. S, and Laird C. D., 2014. Submitted to the Journal of Loss Prevention in the Process Industries.

5.1 Models

For this work, four existing approaches for gas detector placement were implemented and compared with two quantitative optimization-based approaches. The approaches were selected based on their wide use in the process industry. As well, it was intended to include a broad range of qualitative/semi-quantitative approaches and methodologies being currently used. The four existing approaches studied were the Random Approach (RA), the Volumetric Approach (VA), the Minimum Source Distance Problem (MSDP), i.e., the minimization of the distance between the detectors and the leak sources, and a Greedy Coverage (GC) approach. These approaches were compared against the previously presented stochastic programming formulation considering unavailability and voting effects (SP-UV, Benavides-Serrano et al. (2014), Section 4). Additionally, a second mathematical programming formulation, the Maximum Coverage Location Problem (MCLP, Church and ReVelle (1974)), is proposed for comparison purposes. A discussion regarding the algorithms, their technical basis, and challenges is provided in the subsections below. Table A.1 summarizes the data requirements and notation used in each algorithm. For comparison and accountability reasons, equivalent notation was used.

5.1.1 Random Approach (RA)

The RA corresponds to the simplest possible approach for the layout of gas detectors, it only requires the specification of the set L of N candidate detector locations. This algorithm has minimal data requirements since detectors are placed randomly in candidate locations. The algorithm was run 1000 times, and mean values of the performance metrics are reported.

Is not uncommon to encounter sites were no study or analysis is performed to supplement the placement of gas detectors. Due to the lack of a placement strategy, it can be expected that the results obtained by this method will underperform those of the other algorithms and formulations presented in this section. This will provide an expected lower bound for the performance of the other approaches presented, and will help to answer a number of questions: Are other approaches better than a placement performed without any information? If they are, how much better are they? What is the value of the added information?

5.1.2 Volumetric Approach (VA)

In the volumetric approach, the goal of detector placement is to guarantee that a spherical gas cloud is detected before reaching a predetermined diameter. The number of gas detectors is determined based on the facility's volume, and the placement is carried out following a regular or staggered grid pattern. As with the RA, this algorithm only requires the specification of the set L of N candidate detector locations.

The Health and Safety Executive (HSE) (1993) outlines the theoretical background behind this approach. For confined volumes with blockage ratios up to 0.4, the point ignition of a flammable stoichiometric hydrocarbon cloud (methane or propane) will not achieve speeds greater than 125 m/sec if the acceleration distance is less than 6 m. The blockage ratio is defined as the fraction of the volume occupied by congestion elements (i.e. equipment, pipework, etc) to the plant volume under consideration. The resulting overpressures would be less than the threshold for major structural damage (150 mbar), reducing the structural risk to the facility. Based on this evidence, the placement of flammable gas detectors at 5 m intervals was established as a rule of thumb for offshore platforms (Oil & Gas UK (UKOOA), 1995). To date, this approach remains common practice for the determination of appropriate gas detection and placement criteria (International Society of Automation (ISA), 2010). Center for Chemical Process Safety (CCPS) (2009) describes the typical use in the process industries. The determination of the diameter to detect is based on the type and size of the space to be monitored: 4 m for small ventilated buildings, 5 m for other fully enclosed structures with volumes greater than $1,000 \text{ m}^3$ (if they have an inerting system in place), and 5 m for partially enclosed volumes greater than $1,000 \text{ m}^3$, if their blockage ratios are greater than 0.3. Partially enclosed and open volumes with blockage ratios under 0.3 should be analyze in a case by case basis, but usually will not require gas detection unless congested areas are present.

The gas detector placement was performed following Center for Chemical Process Safety (CCPS) (2009) guidelines. Our data sets correspond to partially enclosed volumes with volumes greater than $1,000 \text{ m}^3$, and blockage ratios greater than 0.3. Therefore, the chosen detector spacing was 5 m. The common practice for lighter than air gases is to use a single layer of detectors on an approximate 5 m by 5 m grid, close to the ceiling of the module. Therefore, for each of the models a base case was implemented where a single 5 m by 5 m grid layer of detectors was placed in the ceiling of the monitored module. This was used as a base case. When the height of the modules allowed for it, additional case studies with extra horizontal detector layers were assessed and compared.

5.1.3 Minimum Source Distance Problem (MSDP)

The placement of detectors near leak sources is a widespread rule of thumb in the process industries. The placement strategy behind this practice conjectures that the closer the detectors are placed to the potential leak sources, the more effective the placement will be. That is, the objective is to minimize the expected distance from the leak sources to the detectors. Some sources that explicitly recommend the location of gas detectors close to potential sources of gas releases include International Electrotechnical Commission (IEC) (2007), International Society of Automation (ISA) (2003), National Fire Protection Association (NFPA) (2007), and General Monitors (2013). Some of these standards explicitly suggest prioritizing the placement of gas detectors based on the proximity to the leak source, e.g. NFPA 15 (Section 6.5.2.7.1) (National Fire Protection Association (NFPA), 2007).

The required data for the implementation of this approach is minimal, only a set L of candidate detector locations, and a set S of potential leak source locations are necessary. While sensors could be manually placed near leak locations, to remove subjectivity when the number of sensors is smaller than the number of leaks, we implemented an optimal placement for this objective. This formulation MSDP, is a P-Median Problem (PMP) (Hakimi, 1965; ReVelle and Swain, 1970) that finds sensor locations that minimize the sum of distances between leak locations and the nearest detector. The MSDP is equivalent to formulation SP with a redefined damage coefficient. The damage coefficient definition is presented below. Notation for the formulation is provided in Table A.1. The Pyomo model file for formulation MSDP is equivalent to the SP file presented in Appendix F.

The MSDP minimizes the expected distance from the leak sources to the candidate detector locations (Equation 5.1a) by selecting at most (p) of the candidate detector locations (Constraint 5.1b). Two decision variables are used. Variable y_l indicates if a detector is allocated at location l ($y_l = 1$, and 0 otherwise). Variable $x_{s,l}$ stipulates if a leak source location s was considered covered by a detector at location l ($x_{s,l} = 1$, and 0 otherwise). Parameter $\delta_{s,l}$, corresponds to the Euclidean distance from leak source location s to candidate detector location l. The probability associated with a given leak location s is represented by α_s. Constraint (5.1c) relates the existence of a gas detector at location l with the coverage of leak source location s at that location. Equation (5.1d) specifies that leak source location s can only be considered covered by one detector location. This detector location corresponds to the closest placed detector to the leak source location. Constraint (5.1e) specifies decision variable $x_{s,l}$ as continuous belonging to the interval [0,1]. Equation (5.1e) stipulates decision variable y_l as binary.

$$\min \sum_{s \in S} \alpha_s \sum_{i \in L} \delta_{s,l} \boldsymbol{x_{s,l}} \tag{5.1a}$$

$$\sum_{l \in L} \boldsymbol{y_l} \leqslant p \tag{5.1b}$$

$$\boldsymbol{x_{s,l}} \leqslant \boldsymbol{y_l}, \quad \forall s \in S, l \in L \tag{5.1c}$$

$$\sum_{l \in L} \boldsymbol{x_{s,l}} = 1, \quad \forall\, s \in S \tag{5.1d}$$

$$0 \leqslant \boldsymbol{x_{s,l}} \leqslant 1, \quad \forall\, s \in S, l \in L \tag{5.1e}$$

$$\boldsymbol{y_l} \in \{0, 1\}, \quad \forall\, l \in L \tag{5.1f}$$

5.1.4 Scenario Coverage Approach (GC and MCLP)

Overall scenario coverage has a widespread use as a metric for the placement and evaluation of gas detector systems. Common objectives are the achievement of full scenario coverage with the fewest number of detectors, and the maximization of the overall scenario coverage given a fixed number of detectors, i.e. a predetermined budget.

Several categories of strategies are employed to achieve these objectives. In the basic case, detectors are placed according to the team's assessment of the dispersion patterns while keeping scenario coverage in mind. The number of detectors is obtained from budget constraints, cost-benefit analyzes, heuristics, standards, regulations, or following the team's intuition. For example, API RP 14C (Section C.1.3.2) (American Petroleum Institute (API), 2001) states: *"In enclosed areas containing flammable gas compressors, the minimum number of sensors is one per compressor unit, plus an additional sensor per three units or fractional part thereof"*. A second set of strategies comprises more sophisticated instances where dispersion simulations are used. For example, ISA-TR84.00.07 (International Society of Automation (ISA), 2010) (Annex A.2, Step 7-11 and Annex B.4) proposes the use of probabilistic dispersion simulations supplemented by a graphical output of the scenario coverage. Dispersion scenarios are overlaid on a grid containing information about the candidate detector locations, and detectors are added until a desired overall scenario coverage is achieved. The use of scenario dispersion simulations allows the quantification of the number of scenarios covered by one, two, or more detectors. Results from these approaches can be verified, and are expected to outperform those of the more basic strategies. While high quality quantitative data is used in this second set of strategies, an ad-hoc non-optimal algorithm is used for the detector placement. A third set of strategies would consider the problem from a formal quantitative perspective in order to guarantee the best possible placement given the scenario coverage objective.

Two different placement approaches based on scenario coverage were implemented: A greedy coverage algorithm (GC) and a Maximum Coverage Location Problem (MCLP). The data required for both algorithms is the same. Three sets of data are required. The set L of N potential gas detector locations, the set A of M potential leak scenarios, and sets $\mathcal{L}_a$. Sets $\mathcal{L}_a$ are defined for each scenario, and are defined as the list of all locations that are able to detect scenario a. This corresponds to the data proposed by ISA-TR84.00.07 (International Society of Automation (ISA), 2010) in Step 3. Notation for both approaches is summarized in Table A.1. The Python file corresponding to the implementation of algorithm GC is presented in Appendix I. The Pyomo model file for formulation MCLP is presented in Appendix J.

Algorithm GC places detectors following a simple rule: at each stage, choose the candidate detector location that covers the largest number of uncovered leak scenarios. Once all scenarios are covered once, it continues to add redundancy by selecting the detector that covers the largest number of uncovered leak scenarios twice, thrice, and so on, until each scenario is covered k times. With k depending on the voting scheme. Again, this corresponds to the coverage algorithm embedded in ISA-TR84.00.07 (International Society of Automation (ISA), 2010). While this greedy approach is intuitive and straightforward to implement, it is not guaranteed to be optimal (e.g., an equivalent coverage may be possible with fewer detectors, or a higher coverage may be possible using the same number of detectors).

The MCLP, a mathematical programming formulation, guarantees maximum scenario coverage given a number of detectors. Initially proposed by (Church and ReVelle, 1974), the MCLP maximizes the weighted sum of covered scenarios (Equation 5.2a) by selecting at most (p) of the candidate locations (Constraint 5.2b), therefore avoiding the sub-optimality of greedy placement approaches. Two decision variables are used; y_l and x_a. The first one, y_l, indicates if a detector is allocated at location l ($y_l = 1$, and 0 otherwise). The second, x_a, indicates if a scenario a was detected ($x_a = 1$, and 0 otherwise). Finally, parameter k corresponds to the number of detector confirmations required before emergency actions are triggered, that is, a k-out-of-p voting scheme. Constraint (5.2c) links the existence of a detector impacting scenario a to it's detection by requiring that k detectors must confirm the leak existence before the scenario outcome is flagged as covered. Constraint (5.2d) and (5.2e) specify decision variables x_a and y_l as binary.

$$\max \sum_{a \in A} \boldsymbol{x_a} \tag{5.2a}$$

$$\sum_{l \in L} \boldsymbol{y_l} \leqslant p \tag{5.2b}$$

$$\sum_{i \in \mathcal{L}_a \setminus \mathcal{D}_a} \boldsymbol{y_i} \geqslant k \boldsymbol{x_a}, \quad \forall a \in A \tag{5.2c}$$

$$\boldsymbol{x_a} \in \{0, 1\}, \quad \forall\, a \in A \tag{5.2d}$$

$$\boldsymbol{y_l} \in \{0, 1\}, \quad \forall\, l \in L \tag{5.2e}$$

5.2 Results and Analysis

Results were generated using the four independent data sets and the data generation procedure previously presented in Section 1.3.1. Three different performance metrics were used in accordance to the objectives of gas detection systems, i.e., fraction of covered scenarios at alarm level, fraction of covered scenarios at action level, and expected time to detection at action level. Since a 2-out-of-p voting scheme was used for the results presented, we consider two levels of detection. That is, the *alarm level* corresponds to the results when 1 detector confirms detection of the gas leak, while *action level* corresponds to the results when 2 detectors confirm detection of the gas leak.

Two different sets of results are presented for each data set. A first set of results is presented in Figures 5.1, 5.2, 5.3, and 5.4. These results were obtained by running the detector placement algorithms over each of the full sets of scenario data and then evaluating the generated placement using the three metrics mentioned above. A second set of results are presented in Figures 5.5, 5.6, 5.7, and 5.8. This second set of results were generated in order to test the resilience of the placement algorithms to unforeseen scenarios due to deficiencies in the scenario generation analysis. Here, the detector placement algorithms were run over a randomly selected subset of scenarios. In all cases the size of the subset used to selected the detector locations corresponded to 75% of the total number of dispersion scenarios. Subsequently, the remaining 25% of the scenarios were used to evaluate the generated placement via the three selected metrics. Results are analyzed below for each of the detector placement models. Note that for the RA and VA approaches the placement for both these sets of results is the same since scenario data is not used when determining the placement.

For the remaining approaches (MSDP, GC, MCLP, and SP-UV), required parameters were obtained from the data sets. For GC, MCLP, and SP-UV a given candidate detector

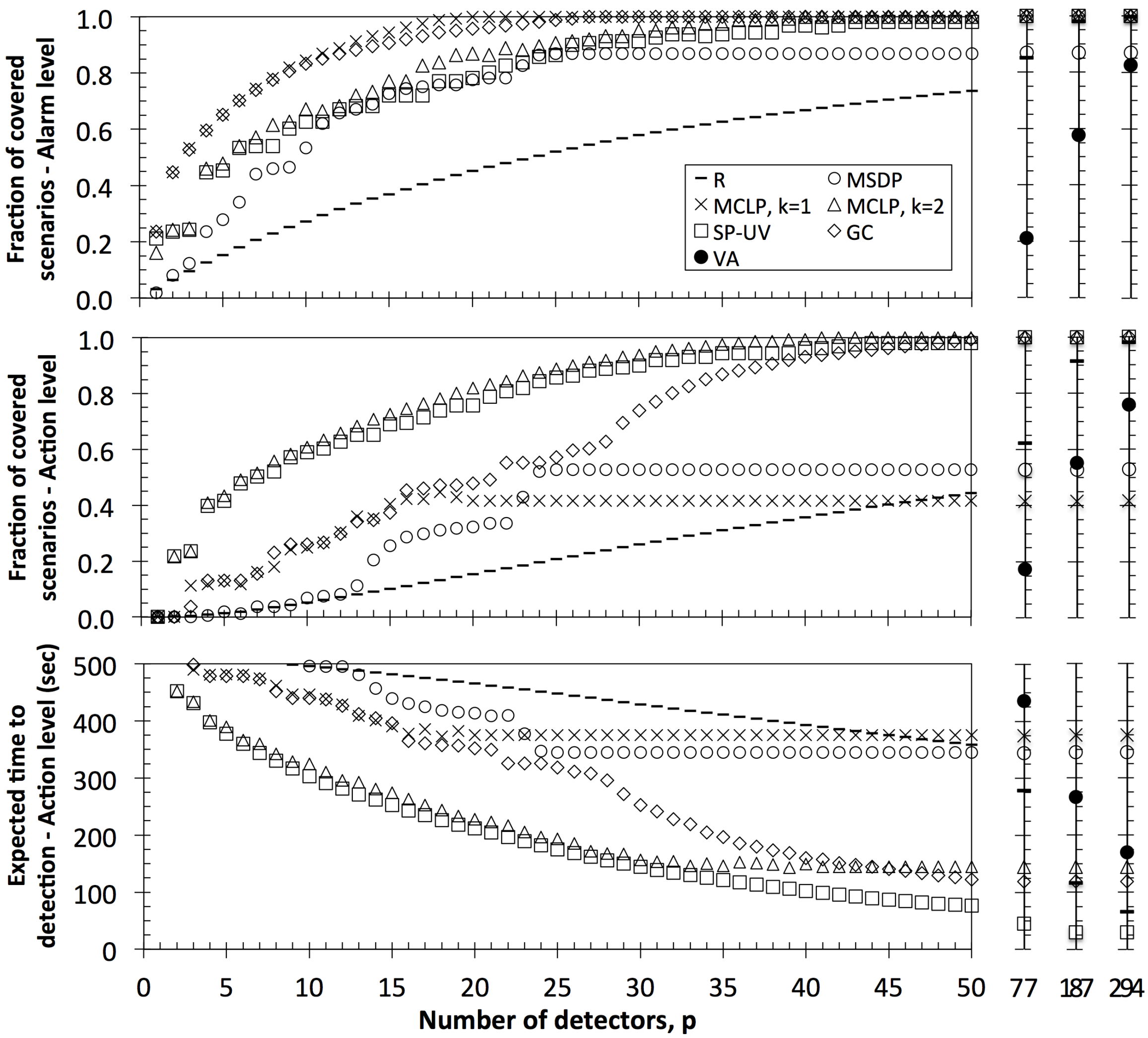


Figure 5.1: Quantitative comparison: Data set A results. Full set of scenarios

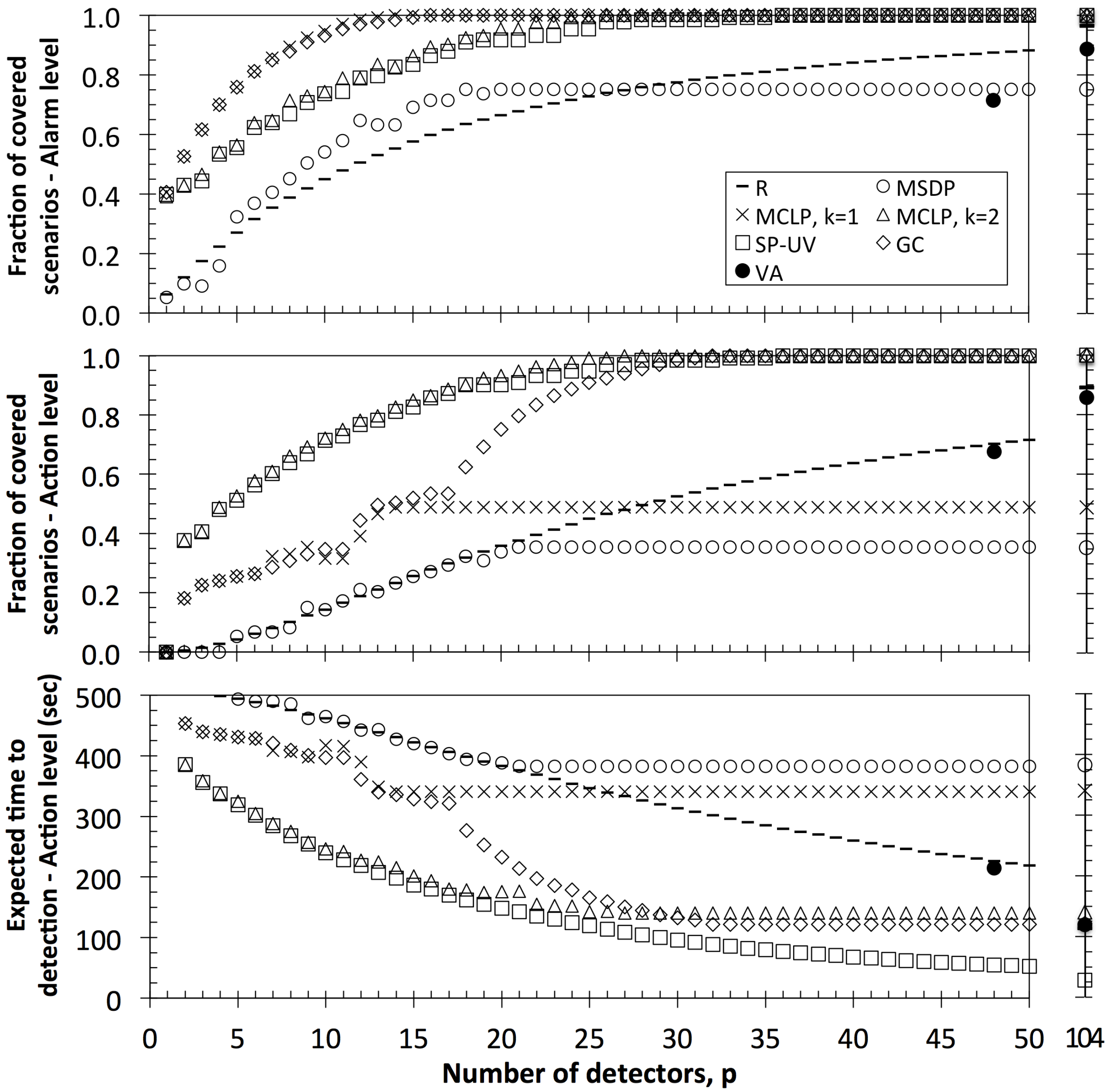


Figure 5.2: Quantitative comparison: Data set B results. Full set of scenarios

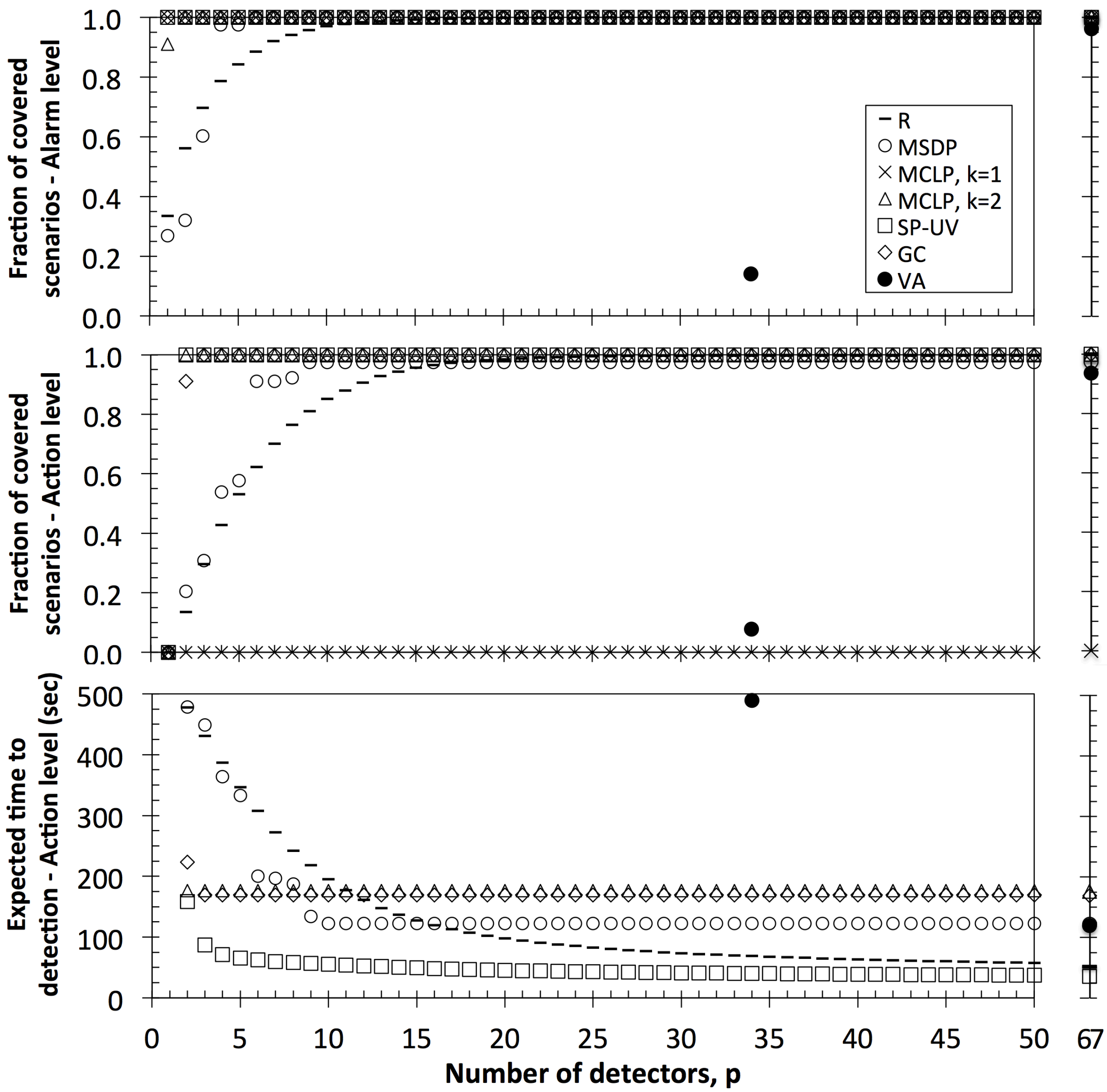


Figure 5.3: Quantitative comparison: Data set C results. Full set of scenarios

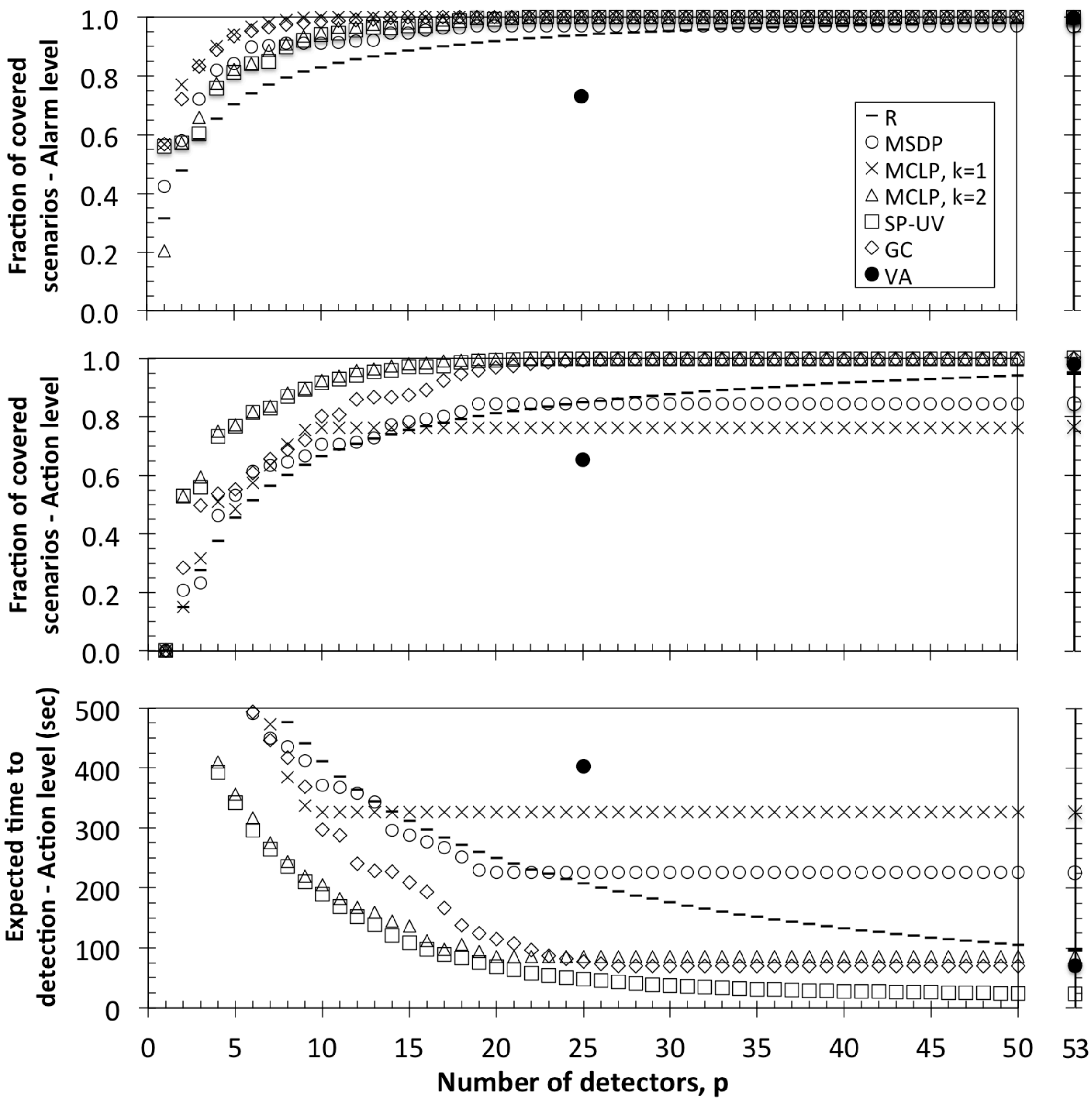


Figure 5.4: Quantitative comparison: Data set D results. Full set of scenarios

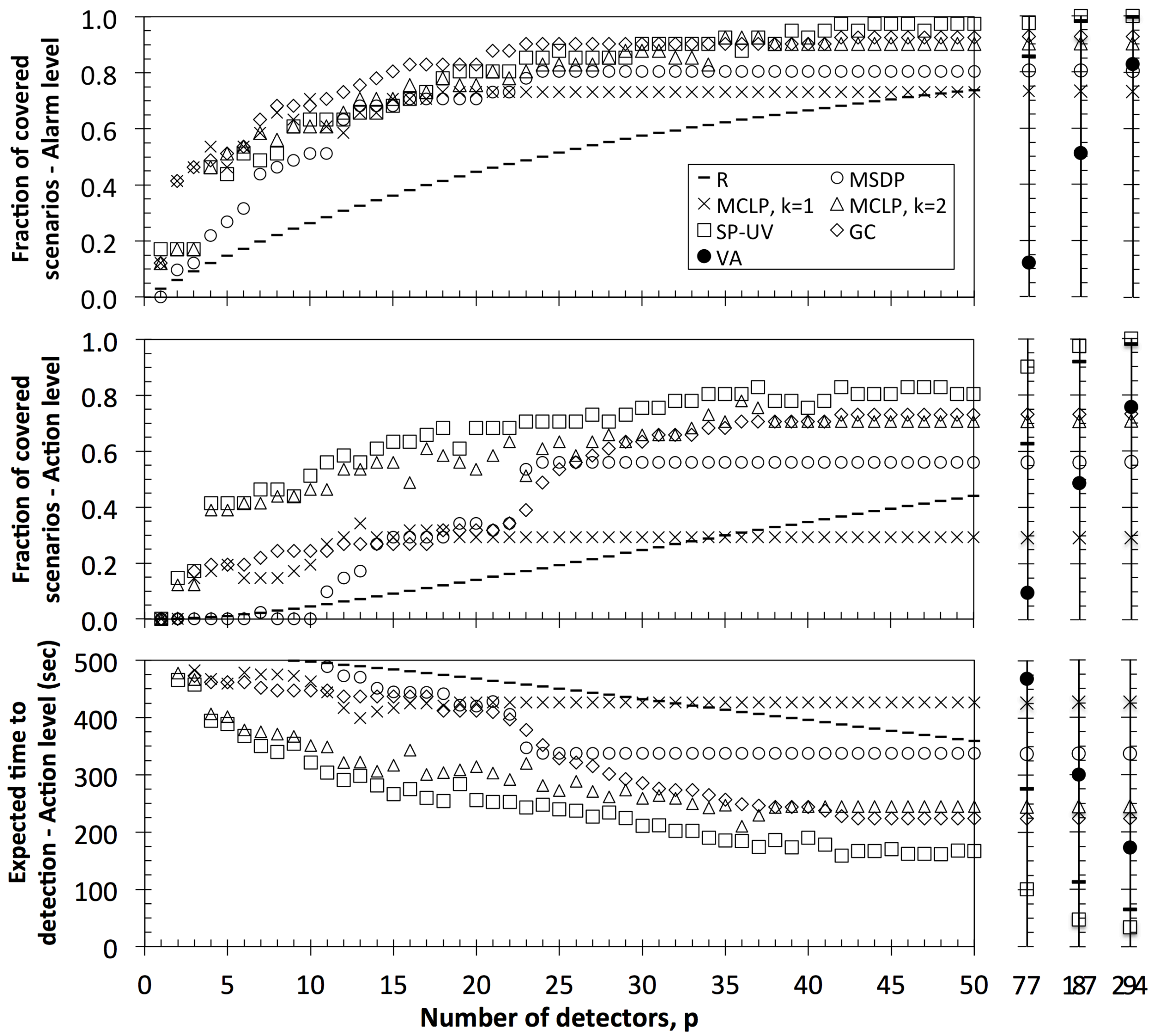


Figure 5.5: Quantitative comparison: Data set A results. Randomly selected subset of scenarios

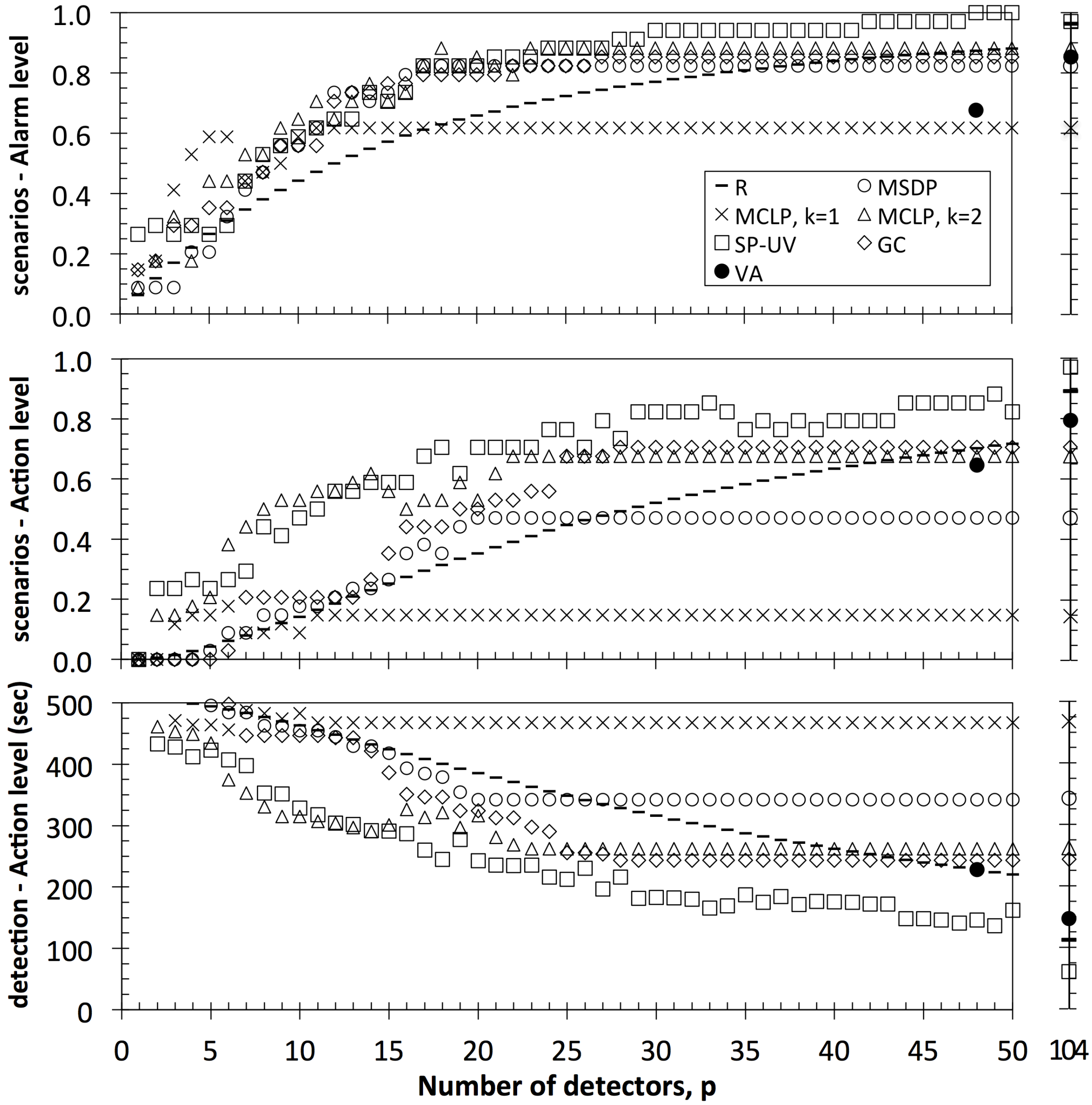


Figure 5.6: Quantitative comparison: Data set B results. Randomly selected subset of scenarios

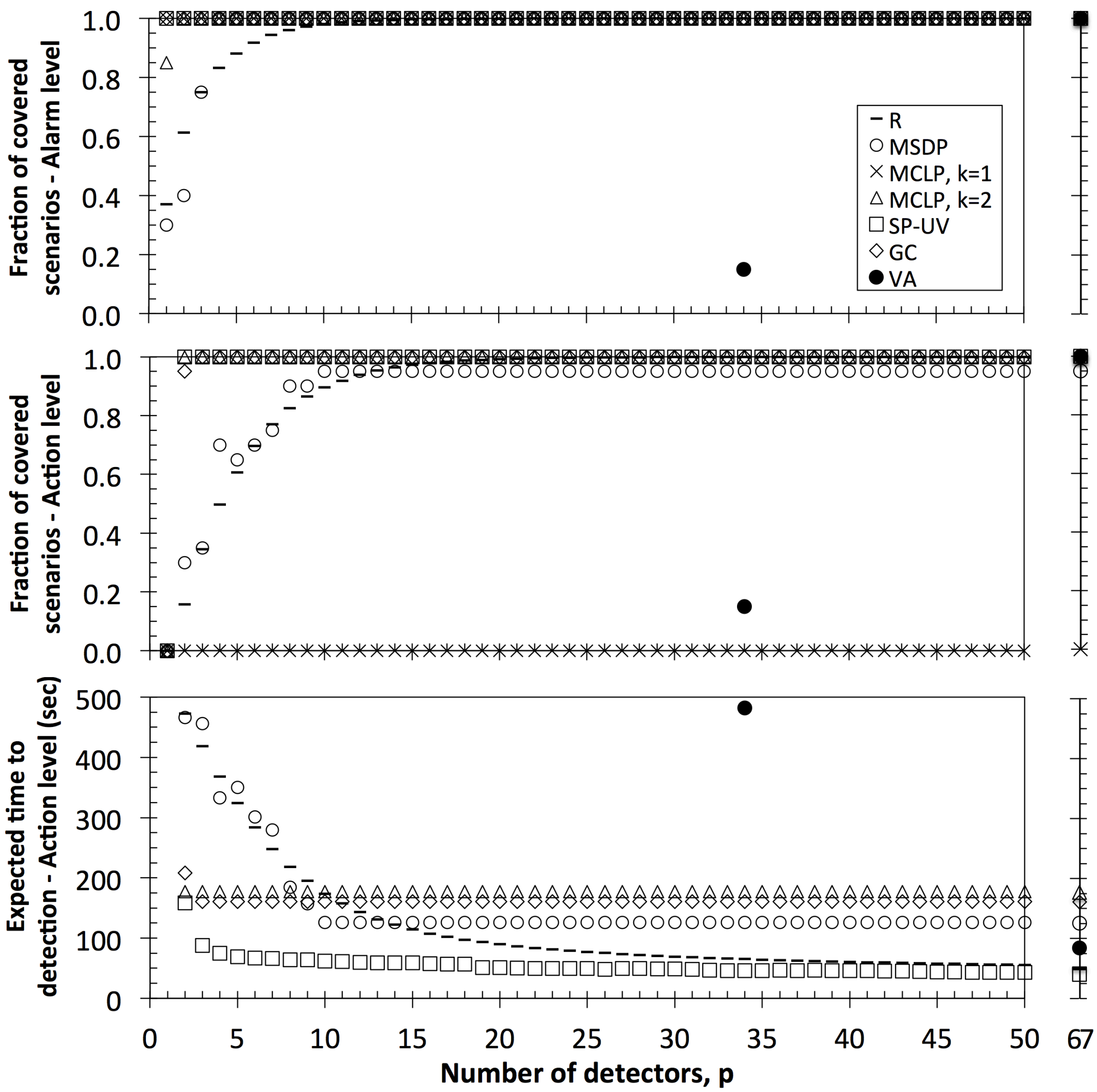


Figure 5.7: Quantitative comparison: Data set C results. Randomly selected subset of scenarios

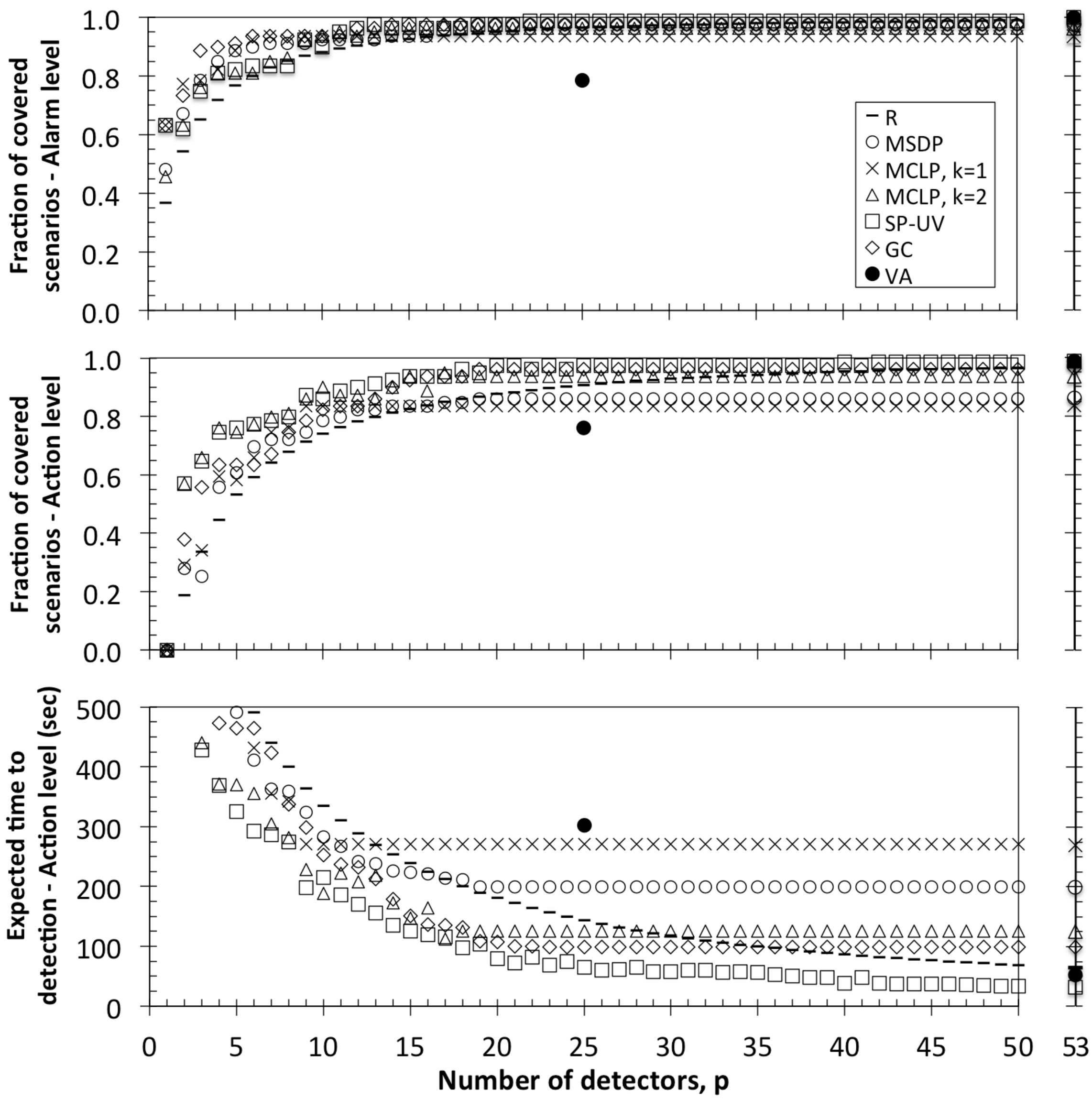


Figure 5.8: Quantitative comparison: Data set D results. Randomly selected subset of scenarios

location (l) was considered to affect a leak scenario (a), and therefore added to set $\mathcal{L}_a$, if the simulated gas cloud reached a concentration of the gas greater than 10% of the Lower Flammability Limit (LFL) value at the candidate detector location. This corresponds to the detection threshold value at which the gas detectors are set. Additionally, for formulation SP-UV discussed in Section 4, a damage coefficient $d_{a,i}$ was calculated for each location i that affected a scenario a. The damage coefficient $d_{a,i}$ is set to the time between the leak initiation for scenario a and the detection of that scenario at location i. The dummy damage coefficient (d_{max}) was set to be 10 seconds greater than the largest damage coefficient $d_{a,i}$ for the given data set. A probability of detector unavailability of q=0.1 was used. A discussion and data for unavailability calculations in the process industries are presented by Modarres et al. (2010) and Stiftelsen for industriell og teknisk forskning (SINTEF) (2002), respectively. Furthermore, two voting policies were assessed, k=1 and k=2. Five coverage levels were considered for all the results present. The same probability of occurrence $\alpha_a = 1/M$ was utilized for each dispersion scenario a in a given data set.

5.2.1 RA Results

As mentioned above, it was expected that the RA would underperform all other approaches. With the exception of the VA, this is true for a low number of detectors for most of the results presented. However, as the number of detectors increases some of the detector placement algorithms (MSDP, GC, MCLP (k=1), and MCLP (k=2)) are actually outperformed by the RA. Once these 4 approaches have all of the leak sources (MSDP) or scenarios (GC, MCLP (k=1) and MCLP (k=2)) covered, there is no secondary objective to intelligently place remaining detectors. With k=1, the MCLP approach stops when all scenarios are covered at alarm level. With k=2, the MCLP and the GC approach stop when all scenarios are covered at action level. In real life applications, past this point the users of these placement approaches may erroneously decide that there is no benefit in placing additional detectors and therefore stop placing detectors. If these approaches must be used, a secondary objective should be considered when the primary objective is completely satisfied. Surprisingly, the VA underperformed the RA results on almost all sets of results. Further discussion of these behaviors is presented in the results analysis section for those detector placement algorithms.

5.2.2 VA Results

The Volumetric Approach (VA) underperformed all other approaches, including the RA. This was an unexpected result, especially considering the widespread use of this approach. For all data sets, the single detector layer cases yielded worse results than those of the RA for both the full set of scenarios and the randomly selected subset of scenarios. Is important to take into account that the MSDP, GC, MCLP (k=1), and MCLP (k=2) algorithms stop intelligently placing detectors after fulfilling their main objective, and therefore past this point they can not be fairly compared. However, in most instances, and even using a much higher number of detectors, VA results are worse or at best in the same range as those obtained by these approaches prior to this point. Adding detector layers did not represent great improvement. When additional layers of detectors are used at several heights, absurd numbers of detectors are required, and the results achieved still continue to underperform those of the RA.

Kelsey et al. (2002, 2005) have previously assessed the performance of this approach on a simulated offshore facility, their findings outline to a great extent the reasons for the poor performance observed in Figures 5.1-5.8. In their study, infrared detectors (point and line of sight), and catalytic detectors were placed aiming to guarantee the detection of a 5 m diameter explosive cloud. For the base cases analyzed, 24 point detectors (two horizontal layers), and 4 beam detectors, were necessary. The proposed placements failed to detect all releases, however, infrared detectors outperformed catalytic detectors in almost every aspect. Three reasons were provided for undetected releases and long detection times. First, small leak rates ($\leqslant 1\,\mathrm{kg} \cdot \mathrm{s}^{-1}$) led to the formation of small and hard to detect clouds. Secondly, some leaks had concentrations below detector alarm and/or action signaling levels. Finally, high momentum horizontal releases missed the detector grid due to the false expectancy that buoyancy would transport releases up to the detectors. For releases forming 5 m diameter clouds, only the infrared point detector grid was successful in detecting all releases at alarm level, but not at action level. For this same grid to detect these releases at action level the cloud size would have to reach 7 m diameter. To address these issues, the common recommendation is to further reduce the spacing between detectors. In the case of Kelsey et al. (2002, 2005) this resulted only in a slight decrease in the mean detection times.

For our study, real leak dispersion behavior resulted in cloud shapes that were not spherical and lengths higher than the separation between detectors. Furthermore, for small clouds and slow leak rates, long times to detection were observed, and in some cases leaks were not detected at all. Also, parts of the geometry with higher congestion and/or confinement will still pose a structural risk to the facility. Additionally, our results provide quantitative evidence of a well-known deficiency of this approach. The number of detectors necessary, even for the standard 5 m cloud, can be prohibitive (Center for Chemical Process Safety (CCPS), 2009; International Society of Automation (ISA), 2010; Kelsey et al., 2002, 2005). Since this methodology was initially developed for offshore facilities, this is a special concern for onshore facilities, which usually have even larger volumes. In our studies, this approach was not only ineffective, but also impractical due to the unreasonable number of detectors required.

5.2.3 MSDP Results

From the perspective of the three metrics, the MSDP can be considered the detector placement approach with the second worst results. Since it was assumed that one detector was sufficient to cover a given leak source, i.e., voting requirements were ignored, after the number of detectors is equal to the number of leaks sources the MSDP stops intelligently placing detectors and can not be fairly compared. However, for the range of results where it can be fairly compared, the MSDP underperformed all of the other approaches with the exception of the RA and VA.

Furthermore, it was observed that for our data sets the detector closest to the leak source was usually not the one that detected the particular leak scenario at alarm level, frequently even the second closest detector was not the one that detected the particular leak scenario at action level. That is, a detector placed near a leak location is not a guarantee of leak detection. This is the basic reason behind the poor performance observed in the three metrics, and constitutes strong evidence against the use of this approach. Momentum effects due to the leak source and/or a strong wind can easily drive the gas cloud away from the detectors. These results are consistent with previously issued warnings regarding the use of this approach (Bratteteig et al., 2011; Center for Chemical Process Safety (CCPS), 2009). Furthermore, since the primary concern of the gas detection system must be the detection of gas leaks, and not the monitoring of leak sources, this approach can lead to nuisance alarms due to insignificant normal operation leakage. The use of this approach without supplementing it with proper dispersion studies and voting considerations may lead to a false sense of security.

5.2.4 GC and MCLP Results

From the set of industry common methodologies (R, VA, MSDP, and GC), the best results were obtained for the greedy scenario coverage algorithm (GC). This constitutes evidence of the advantages of supplementing detector placement practices using information from dispersion simulations.

The main reason for the implementation of the MCLP (k=1) algorithm was to provide an upper bound on the fraction of scenarios detected at alarm level in order to be able to better compare the other approaches. Nevertheless, these results also confirmed previous results presented in Benavides-Serrano et al. (2014) and Section 4, that neglecting voting strategies in the placement algorithms can lead to a significant decrease in performance. This is more evident when MCLP (k=1) and MCLP (k=2) results are compared. The optimal layouts obtained for MCLP (k=1) differed significantly from those obtained with MCLP (k=2).

In all cases, for the expected time to detection metric, the MCLP (k=2) initially outperformed the GC algorithm, but was eventually slightly outperformed by the GC algorithm. This is not surprising since the MCLP (k=2) aims for scenario coverage, not expected time to detection. The MCLP (k=2) is able to cover all scenarios at action level with a lower number of detectors and therefore stops intelligently placing detectors earlier than the GC algorithm. However, and for the range in which both algorithms are comparable, the improvement obtained by using an optimal algorithm becomes evident. Even though they both use the same data, the MCLP (k=2) provides a sensible improvement in the fraction of scenarios detected, and time to detection metrics, closely resembling the SP-UV results. Furthermore, after obtaining scenario data, there is no reason not to use an optimal method.

5.2.5 SP-UV Results

The SP-UV algorithm results confirm above-mentioned conclusions regarding the effectiveness of more informed detector placement approaches, and in particular, mathematical programming approaches. For all the results presented the SP-UV formulation outperformed all the industry common methodologies in both action level metrics. The fraction of scenarios covered at action level metric results were quite high, which was unexpected since this was not the objective considered. For the full set of scenarios results (Figures 5.1-5.4) they closely followed those of the MCLP (k=2) formulation. For the scenarios subset results (Figures 5.5-5.8) the SP-UV results outperformed all of the other approaches, including those of the MCLP (k=2) formulation. For the fraction of covered scenarios at alarm level metric the only algorithm that outperformed the SP-UV formulation was the GC algorithm. This result is not surprising, given that the GC algorithm prioritizes the coverage of scenarios at alarm level.

For the full set of scenarios results (Figures 5.1-5.4) the expected time to detection of the SP-UV formulation was at all times 30 seconds or less than the best industry common methodology (usually the GC algorithm). For the type of mitigation scenarios for which gas detection is designed, 30 seconds or more can represent the difference between low and catastrophic consequences. Furthermore, in most cases the amount of detectors necessary to achieve similar results with the SP-UV formulation are substantially lower than those necessary with the common industry approaches. A remarkable example is provided by data set A (the one associated with the largest geometry). For the results obtained with the full set of scenarios, the amount of detectors necessary to obtain the same expected time to detection and fraction of covered scenarios at action level with the SP-UV formulation was more than an order of magnitude lower that those of the VA. This is strong evidence of the performance improvement that can be achieved using dispersion data and mathematical programming formulations.

5.3 Summary

In this section, four existing approaches for gas detector placement were implemented and compared with the previously proposed quantitative optimization-based approach (SP-

UV, Section 4). The existing approaches compared correspond to the Random Approach (RA), the Volumetric Approach (VA), the minimum detector distance to leak source heuristic, and a Greedy scenario Coverage approach (GC). Apart from formulation SP-UV, two additional mathematical programming formulations were further proposed to capture the essence behind the placement heuristics and metrics presented. The first, formulation MSDP, embeds the minimum detector distance to leak source heuristic and was implemented to eliminate subjectivity in the analysis of this heuristic. The second, formulation MCLP, provides an upper bound on the maximum coverage possible given a number of detectors. It was proposed as a mean for assessing the full capabilities of scenario coverage considerations. In accordance to the objectives of gas detection systems, expected time to detection and scenario coverage metrics were assessed for real sets of CFD dispersion data.

Initially, results were reported using the full data sets for placement and evaluation (Figures 5.1-5.4). These figures yielded three main findings. First, for our study, the Volumetric Approach (VA) consistently performed poorly. This was a direct consequence of the real dispersion cloud shapes not being spherical, a basic assumption behind the use of this approach. Second, the approaches that do not make use of dispersion simulations (R, VA, MSDP) were compared against those that do (GC, MCLP, SP-UV). From this comparison, it is evident that quality dispersion simulations provide a sensible improvement in the performance of the detection and mitigation system. Finally, the approaches that rigorously optimize the objectives (MCLP and SP-UV) were compared versus those that do not (GC). This comparison showed that numerical optimization techniques provide a suitable approach to further improve the performance of gas detector placements.

A second set of results was presented to further validate previous results and analyze the performance of the approaches in the presence unforeseen scenarios not considered during placement. A randomly selected sample of 75% of the data was used to obtain detector placement results. These placements were then evaluated on the remaining 25% of the data. As expected, the results for the detector placement approaches that do not require dispersion scenario data (MSDP and VA) did not have a substantial change in behavior. Also expected, the performance of the approaches that make use of scenario data (MCLP, GC, and SP-UV) decreased since not all scenarios were considered during placement. However, these approaches continued to outperform the algorithms that do not require dispersion data (MSDP, VA). Furthermore, the approaches that rigorously optimize the objectives (MCLP and SP-UV) continued to outperform the ones that do not (GC). It is clear that the use of dispersion simulations and numerical optimization improves the performance of the gas detector placement.

6. BACKUP DETECTION LEVELS EFFECT IN P-MEDIAN FORMULATIONS FOR OPTIMAL PLACEMENT OF DETECTORS IN MITIGATION SYSTEMS *

The uniform unavailability assumption used in formulations SP-U and SP-UV (Sections 3 and 4, respectively) is reasonable for detection and mitigation systems that use the same type of detectors under the same process, environmental, maintenance and repair conditions. However, if this is not the case, the effect of individual detector unavailability needs to be taken into account. This consideration introduces nonlinearities due to the multiplication of probabilities in the objective function that are dependent on detector placement.

As indicated by Snyder and Daskin (2005) and Berman et al. (2007), when the failure probability is reasonably low, the probabilities associated with higher coverage levels quickly tend to zero, and there is often no need to consider more than a few detection levels. Real detectors are expected to have reasonably low unavailability values as presented in Section 1.3.2. However, previously proposed formulations for this problem do not explicitly consider individual detection levels and are not able to take advantage of this feature. The number of variables in the nonlinear product terms is determined by the number of locations capable of detecting individual scenarios. In contrast, our formulations explicitly consider detection levels. This opens then possibility to propose an extended formulation (formulation SPqt, Section 7) that considers a subset of the total possible number of levels, truncating the number of variables in the nonlinear products that form the objective function. In this formulation, the maximum number of variables

* Part of this section is reprinted with permission from "Effect of Backup Detection Levels in P-Median Formulations for Optimal Placement of Detectors in Mitigation Systems" by Benavides-Serrano, A. J., Hackebeil, G., Mannan, M. S., and Laird, C. D., 2014. Proceedings of the 8th International Conference on Foundations of Computer-Aided Process Design (FOCAPD), Washington, USA, July 2014, Copyright 2014 by Elsevier Ltd.

in the objective function products is given by $C-1$, where C is the number of backup detection levels considered. While the size of the problem would increase with the number of detection levels treated, the complexity of the problem would be dictated by the choice of the modeller rather than the problem data itself. Motivated by this perspective, Section 6 uses real facility data for the optimal gas detector placement problem to determine the impact of changing the number of detection levels and select a level of redundancy that gives a reasonable accuracy.

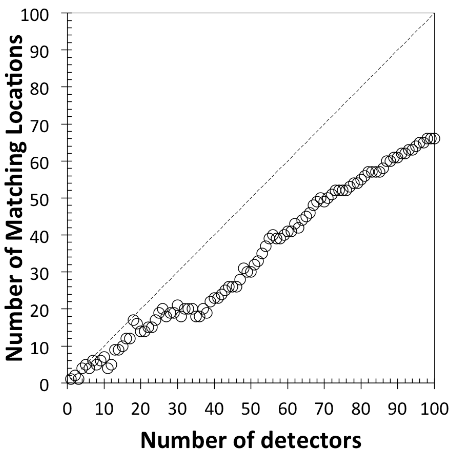
6.1 Effect of Backup Detection Levels

Based on representative unavailability values for the gas detector placement problem (Section 1.3.2), a sensitivity analysis was performed to test the resilience of formulation SPqt to the truncation of the objective function. This sensitivity analysis was carried out and validated via the four data sets presented in Section 1.3.1. Damage coefficients and scenario probabilities were calculated following the same considerations presented in Section 3.2. Two different metrics were implemented. For both metrics, the ideal case in which the maximum allowed number of detector levels is equal to the number of allowed detectors (i.e., $C=p$) was used as the base case. The first metric corresponds to the number of matching detector locations between the truncated objective solution and the base case solution. The second metric corresponds to the percent difference between the truncated objective solution and the base case.

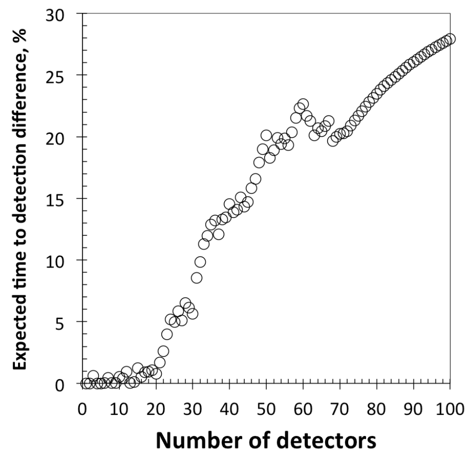
For data set A, three different unavailability values were assessed, q=0.05, q=0.1, and q=0.2. As presented in Section 1.3.2, q=0.05 corresponds to a conservative unavailability value for real gas detectors, from both the time-averaged and instantaneous perspective. The two additional values correspond to the common industry rule of thumb for gas detector time-averaged unavailability (q=0.1), and a representative value for highly substandard repair and maintenance practices (q=0.2). In every test, a uniform unavailability was assumed for all detectors. As presented in Section 7, under this assumption, formulation SPqt collapses back to formulation SP-U and the optimal solution to formulation SP-U is also provably optimal for formulation SPqt. This assumption enabled us to obtain sensitivity results for real size data sets, a task otherwise unachievable since it would have required the solution of the general non-linear formulation (or the complete enumeration of the detector placement combinations for a small subset of the real data as in Berman et al. (2007)).

6.2 Numerical Results Analysis

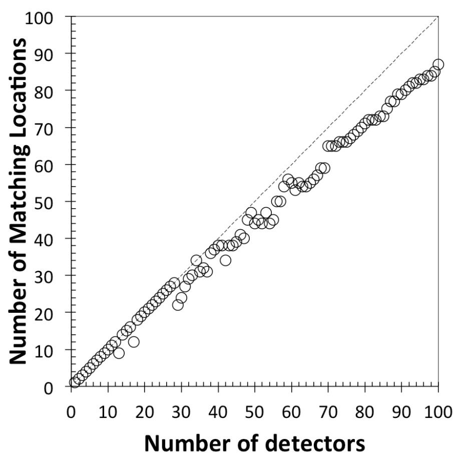
Figure 6.1 present the objective function truncation effect for q=0.1. Three different objective function truncations are presented, C=0, C=1, and C=2. As the maximum number of detector backup levels (C) in the formulation is increased the matching location metric will quickly follow the ideal line, while the percent difference will tend to zero. With $C \geqslant 5$ the base case and the truncated objective formulation will yield the same results for all the range of allowed detectors presented in the figures. In all cases, for a low number of detectors both solutions are the same, but as the number of detectors increases they diverge. Initially, for a low number of detectors, both formulations strive to cover the primary detection level. As the number of allowed detectors is increased, formulation focus changes and the extra detectors are preferentially employed to cover the first backup level. As the number of allowed detectors keeps increasing, the focus changes to the second backup level, and so on. With each successive change of focus the importance of the neglected objective function terms increases. The incremental effect of this lack of information results in the divergent behavior of the solutions. However, even with a high number of allowed detectors (up to 100), for C=1 the percent difference between the solutions is less than 1.4% for all the range of allowed detectors presented. Adding additional backup levels brings the gap to less than 0.05% and 0.001% for the



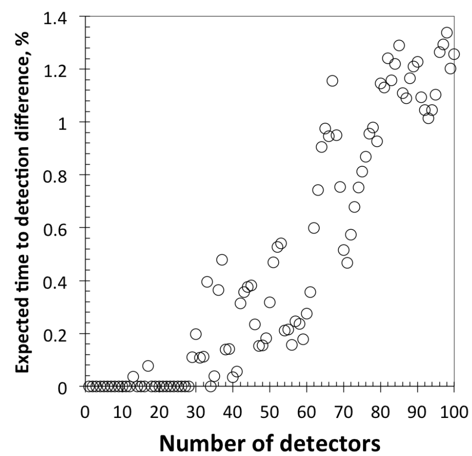
(a) q=0.1, C=0



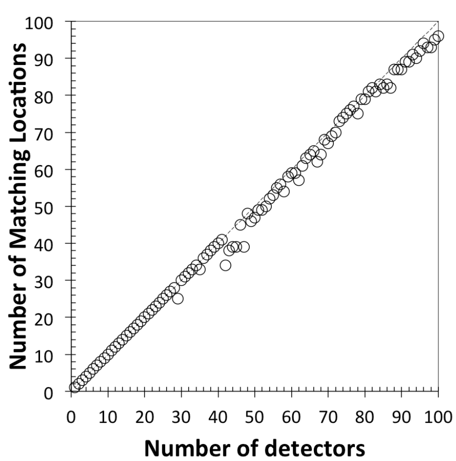
(b) q=0.1, C=0



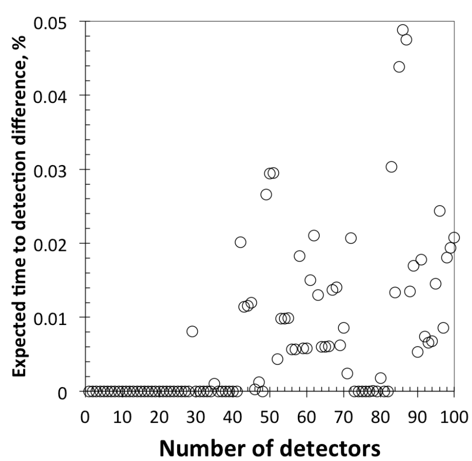
(c) q=0.1, C=1



(d) q=0.1, C=1

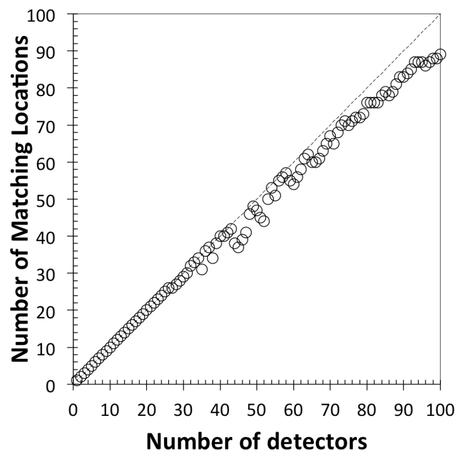


(e) q=0.1, C=2

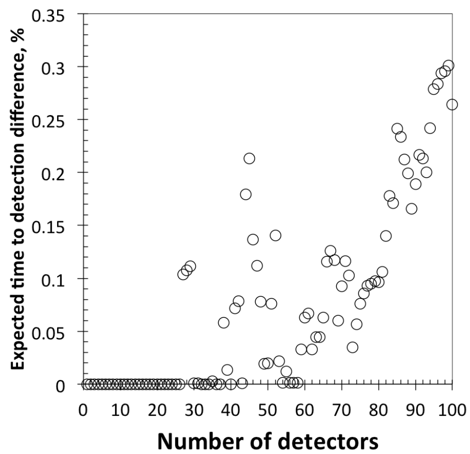


(f) q=0.1, C=2

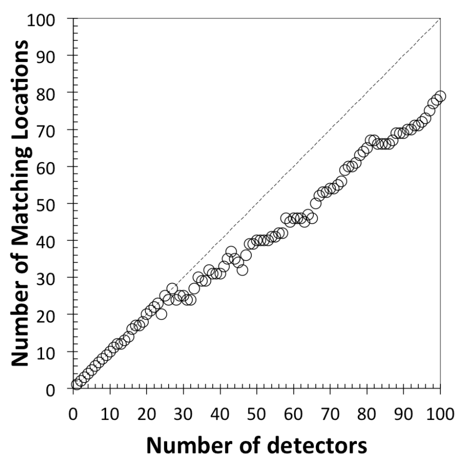
Figure 6.1: Data set A sensitivity analysis results. Numbers of backup levels effect. Objective functions truncated at 0, 1, and 2 backup levels (C=0, 1, 2) for $\bar{q}$=0.1. Matching locations results are presented in Figures 6.1a, 6.1c, and 6.1e. Expected time to detection percent difference results are presented in Figures 6.1b, 6.1d, and 6.1f.



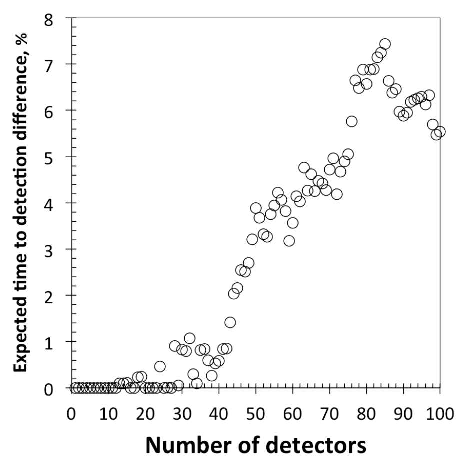
(a) q=0.05, C=1



(b) q=0.05, C=1

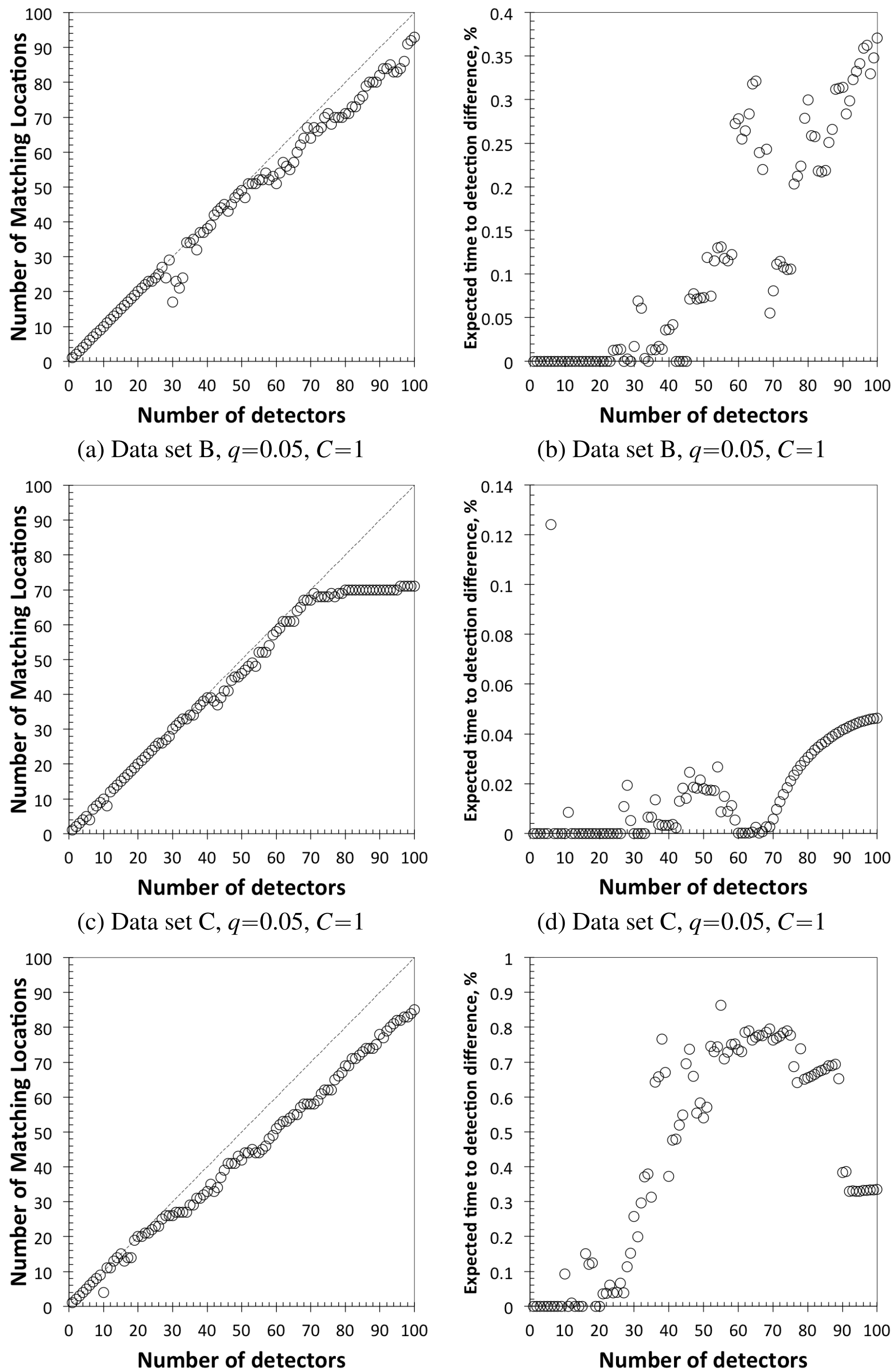


(c) q=0.2, C=1



(d) q=0.2, C=1

Figure 6.2: Data set A sensitivity analysis results. Time-averaged unavailability effect. Objective functions truncated at 1 backup level (C=1) for $\bar{q}$=0.05 and $\bar{q}$=0.2. Matching locations results are presented in Figures 6.2a and 6.2c. Expected time to detection percent difference results are presented in Figures 6.2b and 6.2d.



(a) Data set B, q=0.05, C=1 (b) Data set B, q=0.05, C=1

(c) Data set C, q=0.05, C=1 (d) Data set C, q=0.05, C=1

(e) Data set D, q=0.05, C=1 (f) Data set D, q=0.05, C=1

Figure 6.3: Data sets B, C, and D objective functions truncated at 1 backup level (C=1) for $\bar{q}$=0.05. Matching locations results are presented in Figures 6.3a, 6.3c, and 6.3e. Expected time to detection percent difference results are presented in Figures 6.3b, 6.3d, and 6.3f.

C=2 and C=3 cases, respectively. Moreover, above a certain number of detectors, placing additional detectors is not cost-effective due to the negligible improvement in the expected value of the damage. Optimal solutions above this number will not be of any interest due to their lack of real world applicability. The lower the time-averaged unavailabilities, the closer this number can be expected to be to the point at which the solutions start to diverge. According to the results presented in Benavides-Serrano et al. (2014) for the same case study (data set A and q=0.1), this point is located before 50 detectors. For the $1-50$ detector range the percentage gap is less than 0.7% and 0.03% for the C=1 and C=2 cases, respectively. Note both metrics behavior for $p > 71$ in Figures 6.1a and 6.1b. Past this point, the truncated formulation makes no additional improvement in its objective since all scenarios are already covered at primary coverage level (r=0). The truncated formulation will not benefit from the allocation of more detectors, while the base case formulation will continue to benefit from the allocation of new detectors by providing additional backup coverage.

Figure 6.2 present results for q=0.05 and q=0.2 for data set A with C=1. When compared to Figures 6.1c and 6.1d, these figures illustrate the unavailability effect on the formulation results. As expected, higher unavailabilities result in higher importance of the neglected objective function terms, and therefore in earlier, and higher, solution divergency. For this data set, and even for the wide range of allowed detectors presented, for a conservative time-averaged detector unavailability value (q=0.05) the use of one backup level (C=1) is enough to bring the gap down to less than 0.31%. Even in the case of highly substandard maintenance and repair practices (q=0.2) and real data sets, truncation at C=1 still yields results comparable to those reported for heuristic algorithms on small test instances.

Data set A results were validated using the three remaining data sets with the objective function truncated at one backup level (C=1) and an unavailability of q=0.05. Results are presented in Figure 6.3. The percent differences between the truncated objective solution and the base case were less than 0.38% (data set B), 0.13% (data set C), and 0.87% (data set D) for a range of 1−100 detectors. Note again both metrics behavior for $p > 69$ in Figures 6.3c and 6.3d (data set C). The reason for this behavior is analog to that of Figures 6.1a and 6.1b. Past this point, the truncated formulation makes no additional improvement in its objective since all scenarios are already covered at coverage levels r=0 and r=1.

Based on these results it is possible to conclude that for standard unavailability values, proposed formulation SPqt with C=1 provides a good trade-off between computational complexity and solution accuracy.

6.3 Summary

An analysis of the trade-off between the number of coverage levels considered in our formulations and their solution accuracy was presented. In process facilities, gas detectors can be expected to have unavailabilities below 0.05 (both instantaneous and averaged). Given these unavailability values, probabilities associated with higher coverage levels are expected to quickly tend to zero. We analyze this trade-off by using 4 real data sets for the gas detector placement problem. Two metrics were employed for this analysis, the matching detector locations between the truncated objective solution and the base case solution, and the percent difference between the truncated objective solution and the base case. Results in both metrics confirmed the feasibility of considering just a few backup detector levels under real industry gas detector unavailability values. Furthermore, our results show that two detection levels are sufficient to find objective values within 1% of the optimal solution.

This analysis was performed in order to exploit this feature on a future extension of our formulation. Based on the results obtained in this section, it is reasonable to consider two detection levels in order to obtain an MINLP formulation that can be solved to optimality with minimal deterioration of the optimal objective. Using two detection levels reduces the general nonlinear formulation to a quadratic formulation, opening the door for more efficient global optimization methods on this challenging problem.

7. EXTENDED FORMULATIONS CONSIDERING DYNAMIC NONUNIFORM UNAVAILABILITIES *

If the uniform unavailability assumption in the SP-U formulation (Section 3) is not valid, the modeler is forced to deal with nonuniform detector unavailabilities, and nonlinearities arise due to products of probabilities that depend on detector selection. Current solution strategies for these MINLPs rely on random sampling, heuristic and greedy algorithms, ignoring the detector imperfection, or lumping the locations into detection classes based on their associated unavailability. Several of these strategies can not guarantee high solution quality, while others depend on assumptions similar to that of uniform unavailability.

In this section, we present formulation SPqt, an extension to formulation SP-U that considers the dynamic and nonuniform characteristics of the detector unavailabilities. Formulation SPqt relaxes the assumption of an identical time-averaged unavailability for all detectors by allowing detector unavailabilities to differ by location and time. Formulation SPqt, like those previously presented in the literature, leads to a mixed-integer nonlinear programming formulation due to the need to model products of nonuniform unavailabilities. However, formulation SPqt explicitly considers individual detection levels, allowing the modeler to specify a predetermined number of detection levels and, hence, problem complexity.

This section is organized as follows. Section 7.1 discusses previous nonuniform unavailability extensions to the PMP and SP formulations. Section 7.2 introduces formula-

* Part of this section is reprinted with permission from "Effect of Backup Detection Levels in P-Median Formulations for Optimal Placement of Detectors in Mitigation Systems" by Benavides-Serrano, A. J., Hackebeil, G., Mannan, M. S., and Laird, C. D., 2014. Proceedings of the 8th International Conference on Foundations of Computer-Aided Process Design (FOCAPD), Washington, USA, July 2014, Copyright 2014 by Elsevier Ltd.

tion SPqt. Section 7.3 presents a quadratic version of formulation SPqt that provides a good compromise between accuracy and complexity. The truncation to a quadratic formulation is supported by the analysis of the effect that the backup detection levels have on the solution quality (Section 6). Three solution strategies are proposed for this zero-one quadratic problem: SPqt-Q, SPqt-L_1, and SPqt-L_2 (Sections 7.3.1, 7.3.2, and 7.3.3, respectively). The computational efficiency of these solution strategies is shown in Section 7.4. Section 7.5 presents a solution quality comparison between the quadratic formulation and approximate solution strategies previously proposed by Berry et al. (2009a) and Berman et al. (2007). A summary is presented in Section 7.6.

7.1 Nonuniform Unavailability Extensions to the PMP and SP Formulations

In the context of facility location, Berman et al. (2007) presented the Median Problem with Unreliable Facilities (MPUF) along with a discussion of nodal optimality and asymptotic results for this problem. Three versions of the MPUF are presented. The first version corresponds to a general MPUF objective function where nonuniform unavailabilities are assumed for each facility. The second version is the uniform-MPUF, where the facility unavailabilities are assumed to be uniform. The uniform MPUF formulation closely resembles the one proposed by Snyder and Daskin (2005). The third version presented, the uniform (r, m)-MPUF problem, is a variant of the uniform-MPUF formulation where only the r closest facilities (out of the total m) are used to model the failure/success to obtain service. A greedy heuristic and an improved version of this heuristic were proposed to solve the problems. By using explicit enumeration on small test instances of the general MPUF objective function, percentage difference errors of 8% and 5% were reported for the greedy heuristic and its improved version, respectively.

Along the lines of the general MPUF by Berman et al. (2007), Berry et al. (2009b) presented formulation impSP. This formulation treats variable $x_{a,i}$ in the original SP formulation as the probability that a detector placed at location i is the first to detect hazardous scenario a. This is achieved by replacing Equation (2.1d) with Equation (7.1), where $\mathcal{L}_{a,i}^{<} \subset \mathcal{L}_a$ is the set of locations with a damage coefficient lower than that of location i for hazardous scenario a. Parameter $\bar{q}_i$ represents the time-averaged unavailability of the detector to be placed at location i. The use of Equation (7.1) results in an Mixed Integer Nonlinear Programming (MINLP) formulation. Berry et al. (2009b) proposed six solution strategies to obtain approximate solutions.

$$\boldsymbol{x_{a,i}} = (1-\bar{q}_i)s_i \prod_{j \in \mathcal{L}_{a,i}^{<}} \left(1-(1-\bar{q}_j)s_j\right), \quad \forall a \in A,\ i \in \mathcal{L}_a \tag{7.1}$$

The concept of backup levels is not explicitly incorporated in formulation impSP. If a given scenario a is detected by a total of $|\mathcal{L}_a|$ candidate locations, the longest nonlinear term in the formulation will correspond to a product of $|\mathcal{L}_a|$ integer variables (Equation 7.1).

For small to moderate unavailabilities, the work by Snyder and Daskin (2005) and Berman et al. (2007) in the facility location context provides evidence of the convenience and viability of considering only the dominant terms in the probability products. This result has been validated for heuristic algorithms and exact uniform-unavailability formulations. Furthermore, analogous behavior was observed for the detector placement problem while obtaining the results presented in Sections 3 and 4. When the unavailabilities are reasonably low, the probabilities associated with higher coverage levels quickly tend to zero, and there is often no need to consider more than a few detection levels. In practice, this allows the modeler to specify the maximum degree of the nonlinear products based on the trade-off between computational complexity and accuracy. Due to its explicit treatment of backup detection levels, formulation SP-U provides a natural way of incorporating this idea for exact formulations with non-uniform unavailabilities. An extended formulation, SPqt, is presented in Section 7.2 for this purpose.

7.2 SP Formulation Including Individual Nonuniform Mean Unavailabilities (SPqt)

This formulation is a generalization of the SP-U formulation previously presented in Section 3. Formulation SPqt is presented in Equation (7.2). Notation is provided in Table A.1.

$$\min \quad \int_0^F \frac{E(t)}{F} dt \tag{7.2a}$$

s.t.

$$\sum_{i \in \mathcal{L}_a} \boldsymbol{x_{a,i,r}} = 1 \qquad \forall\ a \in A, r \in R \tag{7.2b}$$

$$\sum_{l \in L} \boldsymbol{s_l} \leqslant p \tag{7.2c}$$

$$\sum_{r \in \mathcal{R}_{a,i}} \boldsymbol{x_{a,i,r}} \leqslant \boldsymbol{s_i} \qquad \forall\ a \in A, i \in \mathcal{L}_a \backslash \mathcal{D}_a \tag{7.2d}$$

$$\sum_{i \in \mathcal{L}_a} d_{a,i} \boldsymbol{x_{a,i,r}} \leqslant \sum_{i \in \mathcal{L}_a} d_{a,i} \boldsymbol{x_{a,i,r+1}}, \qquad \forall a \in A, \{r | r \in R, r \leqslant C-1\} \tag{7.2e}$$

$$\boldsymbol{s_l} \in \{0,1\} \qquad \forall\ l \in L \tag{7.2f}$$

$$\boldsymbol{x_{a,i,r}} \in \{0,1\} \qquad \forall\ a \in A, i \in \mathcal{L}_a, r \in \mathcal{R}_{a,i} \tag{7.2g}$$

The objective function (7.2a) is the expected value of the overall damage considering the probability of detection failure/success. Term $E(t)$ (Equation 7.3) represents the expected value of the overall damage at a given point in time.

$$E(t) = \sum_{a \in A} \alpha_a \sum_{i \in \mathcal{L}_a} \sum_{r \in \mathcal{R}_{a,i}} d_{a,i} \left(1 - q_i(t)\right) \boldsymbol{x_{a,i,r}} \boldsymbol{\Psi_{a,i,r}}(\boldsymbol{x}) \tag{7.3}$$

Parameter F represents the projected life of the facility. Therefore, integral $\int_0^F E(t)/F\ dt$

represents the expected value of the damage over all the projected life of the facility. The trapezoidal rule (Equation 7.4) was used for the numerical integration of the objective function. The domain was discretized into n equally spaced samples.

$$\int_0^F \frac{E(t)}{F}\, dt = \frac{\left(\frac{E(0)}{2} + \sum_{k=1}^{n-1} E(\frac{kF}{n}) + \frac{E(F)}{2}\right)}{n} \tag{7.4}$$

The main feature inherited from formulation SP-U is the concept of detection levels. As in SP-U (Section 3.1), and in order to accommodate this concept, variable $x_{a,i,r}$ indicates that scenario a is detected at coverage level r by location i ($x_{a,i,r}$=1, and 0 otherwise). As well, a set R of C coverage levels was defined. The maximum size of the nonlinear products in the objective function is given by $C + 1$. Subsets $\mathcal{R}_{a,i}$ are defined for each pair of scenario a and location i. These correspond to the set of backup levels in R that location i can occupy given its associated damage coefficient ($d_{a,i}$). Parameter $q_i(t)$ corresponds to the instantaneous unavailability (at time t) of the detector to be placed at location i. For all dummy locations, $q_i(t) = 0 \;\; \forall t > 0$.

Due to the relaxation of the uniform unavailability assumption, the binomial distribution can not be used anymore to model detector failure, therefore, formulation SP-U objective function was modified to accommodate the products of nonuniform unavailabilities. Consider a minimal example where a single scenario a (i.e., α_a=1) can be detected by 4 detector locations. If $d_{a,1} < d_{a,2} < d_{a,3} < d_{a,4}$, and a total of 4 detection levels are allowed (i.e., C=3 and R=$\{0, 1, 2, 3\}$), term $E(t)$ (Equation 7.3) would correspond to:

$$
\begin{aligned}
&d_{a,1}\,(1-q_1(t))\,\boldsymbol{x_{a,1,0}}+\\
&d_{a,2}\,(1-q_2(t))\,\boldsymbol{x_{a,2,0}}+\\
&d_{a,3}\,(1-q_3(t))\,\boldsymbol{x_{a,3,0}}+\\
&d_{a,4}\,(1-q_4(t))\,\boldsymbol{x_{a,4,0}}+\\
&d_{a,2}\,(1-q_2(t))\,\boldsymbol{x_{a,2,1}}\;q_1(t)\boldsymbol{x_{a,1,0}}+\\
&d_{a,3}\,(1-q_3(t))\,\boldsymbol{x_{a,3,1}}\,(q_2(t)\boldsymbol{x_{a,2,0}}+q_1(t)\boldsymbol{x_{a,1,0}})+\\
&d_{a,4}\,(1-q_4(t))\,\boldsymbol{x_{a,4,1}}\,(q_3(t)\boldsymbol{x_{a,3,0}}+q_2(t)\boldsymbol{x_{a,2,0}}+q_1(t)\boldsymbol{x_{a,1,0}})+\\
&d_{a,3}\,(1-q_3(t))\,\boldsymbol{x_{a,3,2}}\;q_2(t)\boldsymbol{x_{a,2,1}}\;q_1(t)\boldsymbol{x_{a,1,0}}+\\
&d_{a,4}\,(1-q_4(t))\,\boldsymbol{x_{a,4,2}}\,(q_3(t)\boldsymbol{x_{a,3,1}}\;q_2(t)\boldsymbol{x_{a,2,0}}+q_2(t)\boldsymbol{x_{a,3,1}}\;q_2(t)\boldsymbol{x_{a,1,0}}+q_1(t)\boldsymbol{x_{a,2,1}}\;q_2(t)\boldsymbol{x_{a,1,0}})\\
&d_{a,4}\,(1-q_4(t))\,\boldsymbol{x_{a,4,3}}\;q_3(t)\boldsymbol{x_{a,3,2}}\;q_2(t)\boldsymbol{x_{a,2,1}}\;q_1(t)\boldsymbol{x_{a,1,0}}
\end{aligned}
\tag{7.5}
$$

For example, if we assume that p=2, and the selected candidate locations are i=1 and i=4, then the scenario will be detected by location 1 (with probability $1-q_1(t)$) or, location 1 will fail, and it will be detected by location 4 (with probability $(1-q_4(t))\ q_1(t)$). With the selected detector locations 1 and 4, and the damage coefficients ordered as indicated, $\boldsymbol{x_{a,1,0}}=1$ and $\boldsymbol{x_{a,4,1}}=1$, while all other $\boldsymbol{x_{a,i,r}}=0$, Equation (7.5) gives the result stated above. Likewise, if p=3 and the selected locations are i=2, i=3, and i=4, then the expected value is given by $(1-q_2(t))+(1-q_3(t))\ q_2(t)+(1-q_4(t))\ q_3(t)\ q_1(t)$, which is captured correctly by Equation (7.5) where $\boldsymbol{x_{a,2,0}}=1$, $\boldsymbol{x_{a,3,1}}=1$, and $\boldsymbol{x_{a,4,2}}=1$, while all other $\boldsymbol{x_{a,i,r}}=0$

By using this example, it is possible to better understand the significance of term $\Psi_{a,i,r}(x)$ in Equation (7.3). $\Psi_{a,i,r}(x)$ represents the summation of all the possible probability combinations associated with detector failure that will result in the detection of scenario a at coverage level r by the detector at candidate location i. All the possible combinations need to be included in $\Psi_{a,i,r}(x)$ since we do not know a priori which candidate detector locations will be selected. For the example presented in Equation (7.5), the corresponding expressions for $\Psi_{a,i,r}(x)$ are presented in Equation Set (7.6).

$$
\begin{aligned}
&\Psi_{a,1,0}(x) = \Psi_{a,2,0}(x) = \Psi_{a,3,0}(x) = \Psi_{a,4,0}(x) = 1 \\
&\Psi_{a,2,1}(x) = \bar{q}_1 \boldsymbol{x_{a,1,0}} \\
&\Psi_{a,3,1}(x) = \bar{q}_2 \boldsymbol{x_{a,2,0}} + \bar{q}_1 \boldsymbol{x_{a,1,0}} \\
&\Psi_{a,4,1}(x) = \bar{q}_3 \boldsymbol{x_{a,3,0}} + \bar{q}_2 \boldsymbol{x_{a,2,0}} + \bar{q}_1 \boldsymbol{x_{a,1,0}} \\
&\Psi_{a,3,2}(x) = \bar{q}_2 \boldsymbol{x_{a,2,1}}\ \bar{q}_1 \boldsymbol{x_{a,1,0}} \\
&\Psi_{a,4,2}(x) = \bar{q}_3 \boldsymbol{x_{a,3,1}}\ \bar{q}_2 \boldsymbol{x_{a,2,0}} + \bar{q}_3 \boldsymbol{x_{a,3,1}}\ \bar{q}_1 \boldsymbol{x_{a,1,0}} + \bar{q}_2 \boldsymbol{x_{a,2,1}}\ \bar{q}_1 \boldsymbol{x_{a,1,0}} \\
&\Psi_{a,4,3}(x) = \bar{q}_3 \boldsymbol{x_{a,3,2}}\ \bar{q}_2 \boldsymbol{x_{a,2,1}}\ \bar{q}_1 \boldsymbol{x_{a,1,0}}
\end{aligned}
\tag{7.6}
$$

It is important to notice that for a given scenario a and detection level r, only one $\Psi_{a,i,r}(x)$ will be different from zero. Also, for any given $\Psi_{a,i,r}(x)$, only one term in the summation will be different from zero. This is due to the relation between $\Psi_{a,i,r}(x)$ and $\boldsymbol{x_{a,i,r}}$ variables.

Contraints 7.2b - 7.2d correspond to those previously presented for formulation SP-U. The first constraint (7.2b) guarantees that every scenario a is detected by a detector at each coverage level r (dummy variables relax this constraint). Equation (7.2c) provides an upper limit, p, on the number of detectors allowed. Equation (7.2d) links the existence of a detector to the coverage levels for a given scenario a. Equation (7.2e) guarantees the proper order of the detection levels. nonlinear

It is important to notice that under the assumption of a uniform time-averaged detector unavailability, formulation SPqt is equivalent to formulation SP-U. First, at all points in time, the probability assigned to the detection of scenario a at coverage level r by the detector at candidate location i will be the same (i.e., $\boldsymbol{\Psi}_{a,i,r} = \bar{q}^{\,r} \;\; \forall a \in A, i \in \mathcal{L}_a, r \in \mathcal{R}_{a,i}$). Second, constraint (7.2e) becomes redundant and can be removed from the formulation. As demonstrated by Snyder and Daskin (2005), if all detectors have the same unavailability, the objective function alone will guarantee that for each scenario a, the detector assigned to the coverage level r will always be a detector with a smaller damage coefficient than the detector assigned to coverage level $r+1$. Applying these two simplifications to formulation SPqt will result in formulation SP-U previously presented by Benavides-Serrano et al. (2014).

The size of formulation SPqt instances increases with the number of detection levels (C) treated. When compared to formulation impSP by Berry et al. (2009b), the number of variables in formulation SPqt will lineally increase with the number of detection levels used. As well, the number of nonlinear terms will be increased due to the necessity of computing each of the possible failure possibilities independently. However, the complexity of the problem is dictated by the degree of the polynomial objective (i.e., by the choice of the modeler) rather than the problem data itself.

7.3 Quadractic SPqt Formulation (SPqt-Q): Solution Strategies

Based on the results in Section 6, is expected that two detection levels are sufficient to find objective values within 1% of the optimal solution for the full formulation. Based on this analysis, formulation 7.2 can be truncated at the first backup level (i.e. C=1 and R=$\{0,1\}$) with little loss of solution quality. The resulting formulation is a linearly constrained zero-one Quadratic Programming (QP) formulation. This formulation is far less computationally expensive than the full initial nonlinear formulation due to the reduction in the number of variables and the reduction in the order of the products summed in the objective function. Note that this order reduction greatly improves the problem conditioning (i.e. reduces the ratio between the largest and the smallest coefficient in the objective function). However, solution of general QP problems is still challenging and this does not guarantee that solution times will be reasonable for the typical problem sizes arising in the placement of gas detectors. Three solution strategies, SPqt-Q, SPqt-L_1, and SPqt-L_2, are assessed below.

7.3.1 SPqt-Q

The first solution strategy assessed was to solve the actual QP problem directly with an off-the-shelf solver. For this purpose, the expected value of the damage at time t, $E(t)$ (Equation 7.3), was truncated at the first backup level as presented in Equation (7.7). The integral definition in Equation (7.4) remains the same. Notation for the formulation is provided in Table A.1. The Pyomo model file containing formulation SPqt-Q is presented in Appendix K.

$$E(t) \approx \sum_{a \in A} \alpha_a \sum_{i \in \mathcal{L}_a} \left(d_{a,i} \left(1 - q_i(t)\right) \boldsymbol{x_{a,i,0}} + \sum_{j \in \mathcal{L}^{>}_{a,i}} d_{a,j}\, q_i(t) \left(1 - q_j(t)\right) \boldsymbol{x_{a,i,0}}\, \boldsymbol{x_{a,j,1}} \right) \tag{7.7}$$

7.3.2 SPqt-L_1

The second solution strategy corresponds to a standard exact reformulation strategy for zero-one QP problems. The bilinearities in the objective function involve binary variables only, it is possible to reformulate these bilinearities to obtain a linear objective. For this purpose, Equation set (7.8) was incorporated into formulation SPqt. Notation is provided in Table A.1. The Pyomo model file containing formulation SPqt-L_1 is presented in Appendix L.

$$E(t) \approx \sum_{a \in A} \alpha_a \sum_{i \in \mathcal{L}_a} \left(d_{a,i} \left(1 - q_i(t)\right) \boldsymbol{x_{a,i,0}} + \sum_{j \in \mathcal{L}^{>}_{a,i}} d_{a,j}\, q_i(t) \left(1 - q_j(t)\right) \boldsymbol{w_{a,i,j}} \right) \tag{7.8a}$$

$$\boldsymbol{w_{a,i,j}} \geqslant \boldsymbol{x_{a,i,0}} + \boldsymbol{x_{a,j,1}} - 1 \quad \forall\ a \in A, i \in \mathcal{L}_a, j \in \mathcal{L}^{>}_{a,i} \tag{7.8b}$$

$$0 \leqslant \boldsymbol{w_{a,i,j}} \leqslant 1 \quad \forall\ a \in A, i \in \mathcal{L}_a, j \in \mathcal{L}^{>}_{a,i} \tag{7.8c}$$

Equation (7.8a) defines $E(t)$ (previously defined by Equation 7.3). Equations (7.8b) and (7.8c) are supplementary sets of constraints added to formulation SPqt for reformulation purposes. Constraint (7.8c) stipulates continuous variable $w_{a,i,j}$. Quadratic terms $x_{a,i,0}\ x_{a,j,1}$ in Equation (7.7) are replaced with variable $w_{a,i,j}$ resulting in a linear objective function. Equation (7.8b) guarantees that $w_{a,i,j}$ will be able to take the value of 1, If, and only if, both $x_{a,i,0}$ and $x_{a,j,1}$ are equal to 1. Otherwise, it will take the value of 0.

A drawback of this approach is the number of additional variables introduced in the reformulation. The maximum number of $w_{a,i,r}$ variables for each of the M scenarios is $4N^2$ (assuming the number of necessary $x_{a,i,r}$ variables for each detection level is N). This would result in a total of $4MN^2$ additional (all $w_{a,i,j}$) variables. However, not every scenario a is detected by every candidate detector location l. Furthermore, for a given scenario a, variable $w_{a,i,j}$ will always be zero unless $\{i, j\} \in \mathcal{L}_a$ and $i < j$. By preprocessing the formulation with the specific problem data, the number of variables and constraints (arising from Equations (7.8b) and (7.8c), respectively) can be reduced several orders of magnitude to $\sum_{a\in A} \sum_{i\in\mathcal{L}_a} |\mathcal{L}^{>}_{a,i}|$.

7.3.3 SPqt-L$_2$

The third solution strategy corresponds to a exact linear reformulation strategy for zero-one QP problems proposed by Sherali and Smith (2007). For this purpose, Equation set (7.9) was incorporated into formulation SPqt (Equation set 7.2). Notation is provided in Table A.1. The Pyomo model file containing formulation SPqt-L$_2$ is presented in Appendix M.

$$\min \quad \sum_{a \in A} \alpha_a \left(\int_0^L \frac{E_1(a,t)}{F} dt + \sum_{i \in \mathcal{L}_a} \boldsymbol{z_{a,i}} \right) \tag{7.9a}$$

$$\int_0^L \frac{E_2(a,t)}{F} dt = \boldsymbol{y_{a,i}} + \boldsymbol{z_{a,i}} \qquad \forall\ a \in A, i \in \mathcal{L}_a \tag{7.9b}$$

$$\boldsymbol{y_{a,i}} \leqslant d_{max} \left(1 - \boldsymbol{x_{a,i,0}}\right) \qquad \forall\ a \in A, i \in \mathcal{L}_a \tag{7.9c}$$

$$0 \leqslant \boldsymbol{y_{a,i}} < \infty \qquad \forall\ a \in A, i \in \mathcal{L}_a \tag{7.9d}$$

$$0 \leqslant \boldsymbol{z_{a,i}} < \infty \qquad \forall\ a \in A, i \in \mathcal{L}_a \tag{7.9e}$$

With,

$$E_1(a,t) \approx \sum_{i \in \mathcal{L}_a} d_{a,i} \left(1 - q_i(t)\right) \boldsymbol{x_{a,i,0}} \tag{7.10}$$

$$E_2(a,t) \approx \sum_{j \in \mathcal{L}_{a,i}^{>}} d_{a,j}\ q_i(t) \left(1 - q_j(t)\right) \boldsymbol{x_{a,j,1}} \tag{7.11}$$

Equation (7.9a) replaces objective function (Equation 7.2a in SPqt). At a given point in time t, and for a given scenario a, term $E_1(a,t)$ (Equation 7.10) represents the primary detection level contributions to the expected value of the damage. Likewise, at a given point in time t and for a given scenario a, term $E_2(a,t)$ (Equation 7.11) represents the first backup detection level contributions to the expected value of the damage. Therefore, integral $\int_0^F \frac{E_1(a,t)}{F} dt$ represents the primary detection level contributions to the expected value of the damage over all the projected life of the facility (F) for a given scenario a. Integral $\int_0^F \frac{E_2(a,t)}{F} dt$ represents the first backup detection level contributions to the expected value of the damage over all the projected life of the facility (F) for a given scenario a. Again, the trapezoidal rule was used to evaluate these integrals.

Equations (7.9b) - (7.9e) are supplementary sets of constraints added to formulation

SPqt for reformulation purposes. Contraint (7.9e) specifies continuous variable $z_{a,i}$. This variable represents scenario a first backup detection level (r=1) interactions given the selection of location i at coverage level primary detection level (r=0). Constraints (7.9d) stipulates dummy continuos relaxation variables $y_{a,i}$ for the pair of scenario a and location i. Equations (7.9c) and (7.9d) guarantee that the quadratic and the linear reformulation of the problem are equivalent. For a given scenario a, if variable $x_{a,i,0}$=0 in Constraint (7.9e), dummy continuos variable $y_{a,i}$ will be allowed to take positive real values up to d_{max}. Therefore the objective function will force $z_{a,i}$=0. On the other hand, if variable $x_{a,i,0}$=1 in Constraint (7.9e), then $y_{a,i}$=0, and the objective function will force $z_{a,i}$ to be minimized based on the selection of the detector at the first backup level (r=1). The main feature of this approach is its linear size with respect to the number of initial $x_{a,i,0}$ variables in the original quadratic problem.

7.4 Comparison of Formulations: Computational Efficiency

Data set A, previously presented in Section 1.3.1, was employed for this analysis. Relevant problem size metrics for each of the test instances are presented in Table 7.1 and computational results are presented in Table 7.2. For a given data set, the parameter that affects the solution efficiency the most is the number of allowed detectors (p). To take into account the solution efficiency sensitivity to this parameter, the solution strategies were compared for a wide range of number of detectors. An upper bound of 64 GB of RAM or 100000 seconds was used for the results.

The optimization problems were formulated in Pyomo (Hart et al., 2011, 2012). Two different solvers were employed. Formulation SPqt-Q results were obtained using CPLEX 12.5.1.0 (default parameters and one thread). Formulations SPqt-L_1 and SPqt-L_2 results were obtained using Gurobi 5.6 (default parameters and one thread). CPLEX was more efficient for the quadratic problem while Gurobi was superior for the linearized versions of the problem. All the problems were run on an Intel Xeon CPU E5-2697 v2 with a clock speed of 2.7 GHz and 264 GB of RAM.

Instantaneous unavailability values were obtained by using Equation (1.2) along with the average value for parameter λ presented in Section 1.3.2 ($1.03 * 10^{-6}(h^{-1})$). The projected life of the facility, F, was assumed to be 20 years. To test the formulation under a wide range of possible ill-conditioning, 3 inspections and repair were randomly generated for each of the possible detector locations. At these randomly generated points in time, the unavailability of the detectors was reset back to 0 (perfect repair assumption). Numerical integration was carried out with n=100. At the discrete points evaluated in the numerical integral, the minimum and maximum instantaneous unavailability values obtained were $6.92*10^{-8}$ and 0.277, respectively. The minimum value reported ignores the unavailabilities that are equal to zero. That is, the ones associated with dummy variables and repairs/start-ups.

		Solution Strategy		
Data Set	**Object**	SPqt-Q	SPqt-L_1	SPqt-L_2
	Integer	11577	11577	11577
A	Continuous	0	130705	9488
	Constraints	5873	136574	16645

Table 7.1: Solution Strategies: Problem sizes

The results demonstrate the efficacy of solving this problem by means of the linear reformulation strategy SPqt-L_2. Even an order of magnitude of improvement can be achieved when SPqt-Q and SPqt-L_2 are compared for a small number of detectors (p). As well, for all the instances compared in Table 7.2, the RAM usage by SPqt-L_1 and

		Solution Strategy		
Data Set	***p***	SPqt-Q	SPqt-L_1	SPqt-L_2
A	1	37.20	10.27	2.68
	5	69.62	158.02	18.50
	10	302.64	342.47	100.87
	15	360.09	643.06	217.75
	20	11436.12	23540.79	1963.97
	25	5.47%†	78014.36	20194.54
	30	11.13%†	4.22%†	5.05%†
	35	13.54%†	5.04%†	6.05%†
	40	11.23%†	5.30%†	5.78%†
	45	11.53%*	5.54%†	5.10%†
	50	9.33%*	4.29%†	3.89%†

*(g%) indicates that the branch-and-bound was using more than 64 GB of RAM and was stopped with a gap of g%

†(g%) indicates that the branch-and-bound was stopped after 100000 seconds with a gap of g%

Table 7.2: Solution Strategies: Time to Solve (s)

SPqt-L_2 was only a small fraction of that required by SPqt-Q.

Our results agree with those presented by Sherali and Smith (2007). For the special case of zero-one quadratic programs with $Q \geqslant 0$, and a single linear knapsack constraint, Sherali and Smith (2007) reported higher efficiency of strategy SPqt-L_2 when compared to SPqt-L_1. Additionally, Sherali and Smith (2007) recognized that when the parameter in the right hand side of the constraint was increased (the analogous of p) the instances became harder to solve, and it was no longer clear which of the formulations was the best one. Our results support the same conclusion for our formulation.

Weighting the performance of the solution strategies over all the range of number of detectors (p), we can recommend the use of linear reformulation strategy SPqt-L_2 for the solution of SPqt-Q. It is important to notice that the tightness of formulation SPqt-L_2 is heavily dependent on the term multiplying $(1 - \boldsymbol{x_{a,i,0}})$ in the right hand side of Equation (7.9c). The maximum damage coefficient (d_{max}) was used for this purpose in the computational studies above. However, as noted by Sherali and Smith (2007), a preprocessing step can be implemented to identify a value that provides the maximum possible tightness (i.e. The maximum possible value for Integral $\int_0^F \frac{E_2(a,t)}{F} dt$). The improve in computational efficiency of this pre-processing step is an interesting area for future work.

7.5 Quadractic SPqt Formulation (SPqt-Q): Solution Quality Comparison

In this section, we compare the solutions quality obtained by our quadratic formulation with the placements produced by previous heuristic approaches from the literature. Based on the results presented in Section 7.4, and taking again into account that this is a design problem, it is computationally reasonable to solve the SPqt with $C = 1$ for real size problem instances. However, the solution quality improvement that the formulation can provide, when compared to previous heuristic approaches for the PMP considering nonuniform unavailabilities, remains unknown. For this purpose, some of the approximate approaches presented in Berry et al. (2009b) and Berman et al. (2007) were reproduced and compared with the results obtained for the quadractic SPqt Formulation (SPqt-Q).

Four approaches were reproduced from Berry et al. (2009b): Random sampling (R), ignoring imperfection (formulation SP), formulation SP-U (dcSP), and the one imperfect witness formulation (oiwSP). Random sampling comprises generating random detector placements, evaluating them using the full objective function (Equation 7.2a), and choosing the best one as problem solution. The number of random detectors placements used for this exercise was 10^5 for each data point. Ignoring imperfection neglects the unavailabilities and makes use of the SP formulation (Section 2.1.3) to find an approximate solution. The details of formulation SP-U are presented in Section 3.1. To apply this formulation, the average unavailability value for the data set ($\bar{q}$=0.061) was assigned to all candidate locations. Under this assumption, formulation SP-U results are equivalent to formulation dcSP (Berry et al. (2006a, 2009b)). Formulation dcSP assigns candidate locations to location classes. All locations within a location class have the same average unavailability, this allows to linearize the initial nonlinear formulation. Assigning the same average unavailability value to all the candidate locations is equivalent to assigning all the locations to one location class. Formulation oiwSP corresponds to formulation SP with the original damage coefficient ($d_{a,i}$) replaced for a weighted damage coefficient ($d'_{a,i}=(1-\bar{q}_i)d_{a,i}+\bar{q}_i d_{max}$). The Pyomo model file containing formulation oiwSP is presented in Appendix N.

Two approaches were reproduced from Berman et al. (2007): Greedy Heuristic(GH) and Improved Greddy Heuristic (IGH). At each iteration, algorithm GH chooses the candidate detector location that minimizes the full objective function (Equation 7.2a). As many iterations as allowed detectors, p, are performed. Algorithm IGH uses GH results and refine them by relocating one detector at a time. The Python files corresponding to the implementation of algorithms GH and IGH are presented in Appendices O and P, respectively.

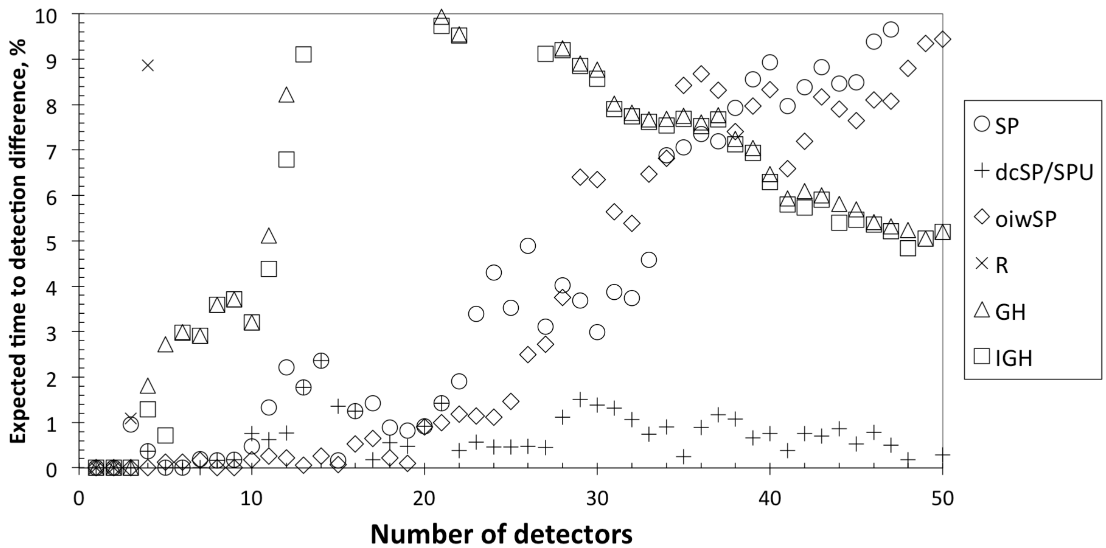


Figure 7.1: Comparison showing the value of SPqt-Q formulation. This figure shows the expected time to detection from the placements produced by SPqt-Q when compared to placements produced by previous heuristic approaches from the literature. This metric is displayed as percent difference from the SPqt-Q results. That is, the percentage differences between the value of the SPqt objective calculated with the optimal SPqt-Q placement and the value of the SPqt objective calculated with the placement provided by the heuristic (R, SP, SP-U, oiwSP, GH, or IGH).

These approaches were compared via the expected time to detection percentage difference metric presented in Section 3. The base case solution corresponds in all cases to the SPqt results obtained with $C=1$. Data set A was used for this analysis. Results are presented in Figure 7.1.

The comparison results presented in Figure 7.1 corroborate the results previously presented in Section 6 and demonstrate the higher solution quality can be obtained by solving the quadratic version of formulation SPqt to optimality. For all, but one point (dcSP/SP-U with 49 detectors), placements from SPqt-Q outperform all of the other approaches. As expected, from the set of compared approaches, the random sampling (R) approach underperformed all of the others. For most of the data range displayed in Figure 7.1, the expected time to detection percentage difference (between SPqt-Q and the random sampling) was higher than 100%. In terms of solution quality, GH and IGH algorithms followed random sampling. Algorithm IGH did not constitute a sensible improvement when compared to algorithm GH results. In term of solution quality, the fourth and fifth place corresponded to SP and oiwSP, respectively. Again, redefining the damage coefficient via formulation oiwSP did not provide a significant improvement when compared to formulation SP. These two formulations provided good solution quality until around 20 detectors. Initially, SP, oiwSP, and SPqt-Q will preferentially provide a primary coverage level of the scenarios. However, when the amount of detectors is sufficiently larger, SPqt-Q will preferentially employ detectors as backup coverage on high impact scenarios. Formulation SPqt-Q will be essentially improving the second coverage level while SP and oiwSP will continue to improve the first coverage level. At this point, SP and oiwSP solution quality starts to dramatically decline. This demonstrates the importance of explicitly considering the second coverage level when performing the optimal placement. These results corroborate previous results on the solution quality improvement that can be obtained when comparing data set A instances with $C=0$ and $C=1$ (Figures 6.1a-6.1d). Finally, SP-U formulation outperformed all of the other approaches with the exception of formulation SPqt-Q. Given the low computational cost of formulation SP-U, its results appear as a promising alternative for the preliminary assessment of the detector placement problem including non-uniform unavailability considerations. It is important to notice that these results are true for the unavailabilities values presented in Section 1.3.2, but may not be true if the unavailabilities are larger (i.e., quadratic truncation may be poor).

7.6 Summary

In Section 3, we developed an MILP formulation, SP-U, for the optimal placement of gas detectors in process facilities. Formulation SP-U assumed an uniform detector unavailability. In this section, we extended formulation SP-U to relax this assumption while additionally including the dynamic unavailability behavior. Relaxing the uniform unavailability assumption leads to nonlinear terms (and MINLPs) due to the need to model products of individual probabilities. However, and unlike previous nonuniform unavailability formulations in the literature, our extended formulation, SPqt, explicitly incorporates the concept of detection levels. This feature, inherited from SP-U, allows the modeler to easily determine the maximum degree of the nonlinear products (i.e. number of coverage levels) to be used based on the trade-off between computational complexity and solution accuracy. By making use of this feature, and leveraging the results obtained in Section 6, a truncated (quadratic) version of formulation SPqt was proposed as an alternative to find high-quality solutions to the full MINLP problem. Two linearized versions of SPqt-Q, formulations SPqt-L_1 and SPqt-L_2, were presented and assessed from the computational efficiency perspective. Results demonstrated the convenience of the use of these linear reformulation strategies for the solution of SPqt-Q. Finally, results from formulation SPqt-Q were compared with a set of approximate solution strategies previously available in the literature for the full MINLP problem. These results, obtained under real unavailability considerations, provide two main conclusions. First, random (R) greedy (GH and IGH) algorithms are outperformed by mathematical programming formulations (SP, oiwSP, SP-U and SPqt-Q). Second, explicit consideration of backup levels in the mathematical programming formulations (SP-U and SPqt-Q) will provide sensible solution quality improvement.

8. SUMMARY, CONCLUSIONS, AND FUTURE WORK *

To this day, gas detector placement in the process industries remains a major concern for which the qualitative and semi-quantitave techniques proposed have significant room for improvement. In this work, we focused on the extension and validation of quantitative solutions to this problem. Specifically, we extended and validated previously developed stochastic mathematical programming formulations in order to include the detector imperfections in the placement problem.

Section 1 presented the motivation behind the research in the optimal placement of gas detectors. The wide impact that proper mitigation can have in the consequences of a loss of containment scenario, the current state of detector placement approaches, and the successful outcomes obtained in both the gas detection and water network communities, make this an interesting and valuable problem to study and solve. Due to the availability of CFD data, our research has focused on the optimal placement of gas detectors. The data acquisition procedure was presented in Section 1.3.1. The data was obtained from four independent data sets corresponding to a real, medium-scale, proprietary offshore facility geometry capturing the full process features. Additionally, a discussion regarding gas detector unavailability in the process industries was presented in Section 1.3.2 to pave

---

* Part of this section is reprinted with permission from ”A Stochastic Programming Approach for the Optimal Placement of Gas Detectors: Unavailability and Voting Strategies” by Benavides-Serrano, A. J., Legg S. W., Vázquez-Román R., Mannan, M. S., and Laird C. D., 2014. Industrial & Engineering Chemistry Research, Copyright 2013 by American Chemical Society. ACS Articles on Request author-directed link: `http://dx.doi.org/10.1021/ie401369v`.

Part of this section is reprinted with permission from “Effect of Backup Detection Levels in P-Median Formulations for Optimal Placement of Detectors in Mitigation Systems” by Benavides-Serrano, A. J., Hackebeil, G., Mannan, M. S., and Laird, C. D., 2014. Proceedings of the 8th International Conference on Foundations of Computer-Aided Process Design (FOCAPD), Washington, USA, July 2014, Copyright 2014 by Elsevier Ltd.

Part of this section is reprinted with permission from “A Quantitative Assessment on the Placement Practices of Gas Detectors in the Process Industries” by Benavides-Serrano A. J., Mannan, M. S, and Laird C. D., 2014. To appear in the Journal of Loss Prevention in the Process Industries.

the way for the discussions in the subsequent sections.

A significant amount of work has been previously conducted by the operations research community, the detector placement community, and our research group. Section 2 presents a literature review of the previous developments on which the formulations presented are based. The review covers the years of previous work invested by the operations research community in furthering the modeling and solution of set covering and P-Median Problems (PMP), how this advancements resulted in formulation SP for the optimal placement of detectors in water networks, and how formulation SP was further adopted and extended for gas detector placement in mitigation systems.

Section 3 extended formulation SP to consider the possibility of detector failure, a key feature associated with gas detector equipment and policies. Gas detectors are prone to a number of failure modes that include failure to function on demand and no output signal. Formulation SP was extended by using a binomial distribution to include the detector unavailability i.e. the possibility of false negative cases. Results were presented for the proposed formulation (SP-U), and then compared with those previously obtained by Legg et al. (2012a,b). The explicit treatment of detector unavailability in the formulation resulted in changes to the optimal detector placement and significant improvements in the expected time to detection when false negative alarms are are considered in the optimal placement formulation.

Section 4 further extended formulation SP-U to include the requirement for a voting logic. A voting logic scheme is commonly utilized in the industry to shield the detection and mitigation system against costly false positive alarms. Formulations SP-UV was proposed to address this issue by explicitly including voting logic considerations into the formulation via a negative binomial distribution. Formulation SP-UV results were presented and compared with optimal placement results from our previously proposed formulations that ignore the voting logic considerations (SP and SP-U). As with the unavailability considerations in formulation SP-U, voting logic considerations in formulation SP-UV resulted in changes to the optimal detector placement, and tangible improvements in the expected time to detection when false positives alarms are considered in the optimal placement formulation.

While Section 3 and 4 results demonstrated again the potential and suitability of numerical optimization to solve the gas detector placement problem while rigorously considering its inherent uncertainties, further validation was presented in Section 5 to demonstrate their potential. Four existing industry approaches for gas detector placement were implemented and compared with the previously proposed quantitative optimization-based approach SP-UV using three different performance metrics in accordance to the objectives of gas detection systems. Results provided evidence on the effectiveness of the use of dispersion simulations, and mathematical programming, to supplement the gas detector placement problem.

Until this point, the formulations presented (SP-U and SP-UV) assumed that the detector unavailability was uniform across all the candidate detector locations. Considering independent detector failure probabilities leads to a mixed-integer stochastic programming formulation with nonlinear terms. However, since the nonlinear terms arise due to the need of considering products of probabilities associated with detection levels, it is possible to select a level of redundancy that gives a reasonable accuracy while reducing the complexity of the MINLP. In Section 6 we analyzed the effect of reducing the number of detection levels considered using real unavailability data for the gas detector placement problem. For the problem, our results show that there is minimal deterioration of the optimal objective as a result of this reduction. Furthermore, the results shown that objective values within 1% of the optimal solution can be achieved by using only two detection levels.

Formulation SPqt was presented in Section 7. Formulation SPqt further extends formulation SP-U by explicitly considering the dynamic and nonuniform characteristics of the detector unavailabilities. However, unlike previous formulations in the literature for the nonuniform unavailability problem, formulation SPqt explicitly considers different backup detection levels, a feature inherited from formulation SP-U. This feature allows an approximation where the maximum degree of the nonlinear products considered can be determined by the modeler, allowing her/him to manipulate the balance between solution accuracy and problem complexity. Making use of this feature, and leveraging the results presented in Section 6, formulation SPqt was truncated at the second detection level to obtain a quadratic formulation (SPqt-Q) that provides a good compromise between accuracy and complexity. This formulation was solved directly as an MIQP and via two proposed exact linear reformulation strategies. Finally, the quadratic formulation results were compared with those of other approximate solution strategies for the full MINLP. Formulation SPqt-Q outperformed all of the other approaches, demonstrating its potential to obtain near-optimal solutions to the non-uniform unavailability detector placement problem for real non-uniform unavailability values.

8.1 Future Work

For the future work, from the modeling perspective there is interest in the analysis of larger and more complete data sets. The burden in these approaches is the identification and computation of the set of dispersion simulations. Statistical analysis of larger data sets will allow one to properly determine the size of the uncertainty space that must be considered. We are specially interested in finding good solutions with fewer scenarios. For the particular case of gas detectors, this uncertainty space includes leak rate, leak location, leak direction, and ventilation rate and direction. What is the relation between the solution obtained with a representative size of scenarios and that of the original, full stochastic problem? How are the parameters in the uncertainty space related to the optimal scenario sample size? As well, sensitivity analysis of more complete sets of data will result in the identification of the relevant metrics to include in this type of analysis. This work considered time to detection as the metric to minimize in the objective function. This study can include other metrics like cloud volume, cloud shape, and cost (e.g., wiring costs, maintenance costs, repair activities and ultimately consequence costs), and how they related to the damage coefficient, $d_{a,i}$. Finally, we are also interested in testing the proposed formulations for other type of detection and mitigation systems (e.g. toxic, flame, and smoke detection systems). This will further validate the general applicability of the formulation presented, and its future extensions, to the general mitigation system detector placement problem.

From the computational point of view, several interesting research questions arise. First, a stochastic problem such as the gas detector placement problem can be solved by decomposing the problem by scenarios. Decomposing the problem by scenarios and using parallel computing strategies can be specially useful for large problem instances. Progressive Hedging (PH) (Rockafellar and Wets, 1991) is one of such decomposition strategies. It is an iterative method which solves scenario sub-problems separately and then enforces implementability. Based on preliminary results, using the PH module in Pyomo, this stands as a good alternative for large sets of data. However, further testing is still required. Second, in the case of formulation SPqt-Q, it is worth exploring other linear reformulation strategies and improve the ones already proposed (e.g., improving the tightness of formulation SPqt-L_2 as suggested in Section 7.4). Finally, the development and assessment of specialized solution algorithms for the proposed formulations is also an area of interest. For example, for full-sized problems in water networks, Berry et al. (2009b) found that a variation of the Greedy Randomized Search Procedure (GRASP) for the PMP (Resende and Werneck, 2004) always provided the best solution when compared against five other approaches implemented in the paper. Are these results valid for large instances of the gas detector placement under realistic detector unavailabilities? If yes, for which instances is recommended to use mathematical programming approximations and for which should approaches like the GRASP be used?

Ultimately, the goal is for the industry to be able to benefit from the use of these formulations. All of the above mentioned and future efforts need to improve the approach and aim to bring optimization-based gas detector placement into practice.

REFERENCES

Aly, A. A. and White, J. A. (1978). Probabilistic formulation of the emergency service location problem. *The Journal of the Operational Research Society*, 29(12):1167–1179.

American Petroleum Institute (API) (2001). *Recommended Practice 14C, Recommended practice for analysis, design, installation, and testing of basic surface safety systems for offshore production platforms, 7th ed.* API Publishing Services, 1220 L Street, N. W, Washington, D.C. 20005.

Artzner, P., Delbaen, F., Eber, J., and Heath, D. (1999). Coherent measures of risk. *Mathematical Finance*, 9(3):203–228.

Ball, M. O. and Lin, F. L. (1993). A reliability model applied to emergency service vehicle location. *Operations Research*, 41(1):18–36.

Batta, R., Dolan, J. M., and Krishnamurthy, N. N. (1989). The maximal expected covering location problem: Revisited. *Transportation Science*, 23(4).

Benavides-Serrano, A. J., Legg, S., Mannan, M., and Laird, C. (2012). A stochastic programming approach for optimal gas sensor layout considering sensor reliability. In *15th Annual Symposium, Mary Kay O'Connor Process Safety Center*.

Benavides-Serrano, A. J., Legg, S. W., Vázquez-Román, R., Mannan, M. S., and Laird, C. D. (2014). A stochastic programming approach for the optimal placement of gas detectors: Unavailability and voting strategies. *Industrial & Engineering Chemistry Research*, 53(13):5355–5365.

Benedict, J. M. (1983). Three hierarchical objective models which incorporate the concept of excess coverage for locating e.m.s. vehicles or hospitals. Master's thesis, Department of Civil Engineering, The Northwestern University, Evanston, Ill.

Berlin, G. N. (1972). *Facility location and vehicle allocation for provision of an emergency service*. PhD thesis, The Johns Hopkins University, Baltimore, Md.

Berman, O., Krass, D., and Menezes, M. B. C. (2007). Facility reliability issues in network p-median problems: Strategic centralization and co-location effects. *Operations Research*, 55(2):332–350.

Berman, O. and Larson, R. C. (1982). The median problem with congestion. *Computers & Operations Research*, 9(2):119–126.

Berman, O., Larson, R. C., and Parkan, C. (1987). The stochastic queue p-median problem. *Transportation Science*, 21(3):207.

Berry, J., Carr, R., Hart, W., Leung, V., Phillips, C., and Watson, J. (2006a). On the placement of imperfect sensors in municipal water networks. In *Proceedings of the 8th Symposium on Water Distribution Systems Analysis. ASCE.*

Berry, J., Carr, R., Hart, W., Leung, V., Phillips, C., and Watson, J. (2009a). Designing contamination warning systems for municipal water networks using imperfect sensors. *Journal of Water Resources Planning and Management*, 135(4):253–263.

Berry, J., Carr, R. D., Hart, W. E., Leung, V. J., Phillips, C. A., and Watson, J.-P. (2009b). Designing contamination warning systems for municipal water networks using imperfect sensors. *Journal of Water Resources Planning and Management-Asce*, 135(4):253–263.

Berry, J., Fleischer, L., Hart, W., Phillips, C., and Watson, J. (2005). Sensor placement in municipal water networks. *Journal of Water Resources Planning and Management*, 131(3):237–243.

Berry, J., Hart, W., Phillips, C., Uber, J., and Watson, J. (2006b). Sensor placement in municipal water networks with temporal integer programming models. *Journal of water resources planning and management*, 132:218–224.

Bianchi, G. and Church, R. L. (1988). A hybrid fleet model for emergency medical service system design. *Social Science & Medicine*, 26(1):163–171.

Bratteteig, A., Hansen, O. R., Gavelli, F., and Davis, S. (2011). Using CFD to analyze gas detector placement in process facilities. In *14th Annual Symposium, Mary Kay O'Connor Process Safety Center*.

Buncefield Major Incident Investigation Board (2008a). *"The Buncefield Incident, 11 December 2005", The final report of the Major Incident Investigation Board*, volume 1. The Office of Public Sector Information, Richmond, Surrey, United Kingdom.

Buncefield Major Incident Investigation Board (2008b). *"The Buncefield Incident, 11 December 2005", The final report of the Major Incident Investigation Board*, volume 2. The Office of Public Sector Information, Richmond, Surrey, United Kingdom.

Bureau of Safety and Environmental Enforcement (BSEE) (2012). Incident statistics and summaries 1996-2012 ytd. `www.bsee.gov/Inspection-and-Enforcement/Accidents-and-Incidents/Listing-and-Status-of-Accident-Investigations.aspx` [Online; accessed April 8, 2013].

Canadian Standards Association (2001). *Combustible gas detection instruments Cana-*

dian electrical code part 2, industrial products, standard CSA-C22.2 No 152–M1984 (R2006).

Center for Chemical Process Safety (CCPS) (2009). *Continuous monitoring for hazardous material releases*. Wiley-AIChE, Hoboken, New Jersey.

Chapman, S. C. and White, J. A. (1974). Probabilistic formulations of emergency service facilities location problems. In *ORSA/TIMS Conference*, San Juan, Puerto Rico.

Church, R. and ReVelle, C. (1974). The maximal covering location problem. *Papers of the regional science association*, (32):101–118.

Church, R. and Weaver, J. (1986). Theoretical links between median and coverage location problems. *Annals of Operations Research*, 6:1–19. 10.1007/BF02034236.

Church, R. L. and Murray, A. T. (2009). *Business site selection, location analysis and GIS*. Wiley, New York.

Crowl, D. and Louvar, J. (2011). *Chemical Process Safety: Fundamentals With Applications*. Prentice Hall International Series in the Physical and Chemical Engineering Sciences. Prentice Hall PTR.

Cui, T., Ouyang, Y., and Shen, Z.-J. M. (2010). Reliable facility location design under the risk of disruptions. *Operations Research*, 58(4):998–1011.

Curtin, K. M., Hayslett-McCall, K., and Qiu, F. (2010). Determining optimal police patrol areas with maximal covering and backup covering location models. *Networks & Spatial Economics*, 10(1):125–145.

Daskin, M. S. (1982). Application of an expected covering model to emergency medical service system design*. *Decision Sciences*, 13(3):416–439.

Daskin, M. S. (1983). A maximum expected covering location model: Formulation, properties and heuristic solution. *Transportation Science*, 17(1).

Daskin, M. S. (2008). What you should know about location modeling. *Naval Research Logistics (NRL)*, 55(4):283–294.

Daskin, M. S. and Dean, L. K. (2005). Location of health care facilities. In Brandeau, M. L., Sainfort, F., and Pierskalla, W. P., editors, *Operations Research and Health Care*, volume 70 of *International Series in Operations Research and Management Science*, pages 43–76. Springer US.

Daskin, M. S., Hogan, K., and ReVelle, C. (1988). Integration of multiple, excess, backup, and expected covering models. *Environment and Planning B: Planning and Design*, 15(1):15–35.

Daskin, M. S. and Stern, E. (1981). A hierarchical objective set covering model for emergency medical-service vehicle deployment. *Transportation Science*, 15(2):137–152.

DeFriend, S., Dejmek, M., Porter, L., Deshotels, B., and Natvig, B. (2008). A risk-based approach to flammable gas detector spacing. *Journal of hazardous materials*, 159(1):142–151.

Det Norske Veritas (DNV) (2008). *Offshore Standard D301 - Fire protection*.

Dhillon, S. and Chakrabarty, K. (2003). *Sensor placement for effective coverage and surveillance in distributed sensor networks*, volume 3. IEEE.

Drezner, Z. (1987). Heuristic solution methods for two location problems with unreliable facilities. *The Journal of the Operational Research Society*, 38(6):509–514.

Eaton, D., Sanchez, H., Lantigua, R., and Morgan, J. (1986). Determining ambulance deployment in santo-domingo, dominican-republic. *Journal of the Operational Research Society*, 37(2):113–126.

Erdemir, E. T., Batta, R., Rogerson, P. A., Blatt, A., and Flanigan, M. (2010). Joint ground and air emergency medical services coverage models: A greedy heuristic solution approach. *European Journal of Operational Research*, 207(2):736–749.

Farahani, R. Z., Asgari, N., Heidari, N., Hosseininia, M., and Goh, M. (2012). Covering problems in facility location: a review. *Computers and Industrial Engineering*, 62(1):368–407.

Fire & Safety World Online (2011). Gas detection. `http://www.fs-business.com/FAQ/FAQGasDetection.asp`. [Online; accessed 19-July-2011].

FM Global (2001). *FM 6310/6320: Approval Standard for Combustible Gas Detectors*.

Gencer, C., Aydogan, E., and Soydemır, A. (2008). Chemical agent detector placement methodology. *Applied Mathematics and Computation*, 195(2):542–557.

Gendreau, M., Laporte, G., and Semet, F. (1997). Solving an ambulance location model by tabu search. *Location Science*, 5(2):75–88.

General Monitors (2013). Fundamentals of combustible gas detection. `http://s7d9.scene7.com/is/content/minesafetyappliances/Combustible%20Gas%20Detection%20White%20Paper`. [Online; accessed 15-June-2014].

GexCon (2011). FLACS CFD disperson modeling and explosion software. `http://gexcon.com/FLACSoverview`. [Online; accessed 15-January-2012].

Hakimi, S. L. (1965). Optimum distribution of switching centers in a communication network and some related graph theoretic problems. *Operations Research*, 13(3):462–475.

Hansen, O. R., Gavelli, F., Davis, S. G., and Middha, P. (2013). Equivalent cloud methods used for explosion risk and consequence studies. *Journal of Loss Prevention in the Process Industries*, 26(3):511 – 527. Papers presented at the 2011 Mary Kay O'Connor Process Safety Center International Symposium.

Hansen, O. R. and Middha, P. (2008). CFD-based risk assessment for hydrogen applications. *Process Safety Progress*, 27(1):29–34.

Hart, W., Watson, J., and Woodruff, D. (2011). Pyomo: modeling and solving mathematical programs in python. *Mathematical Programming Computation*, 3(3):219–260.

Hart, W. E., Laird, C., Watson, J.-P., and Woodruff, D. L. (2012). *Pyomo - Optimization Modeling in Python*. Springer.

Health and Safety Executive (HSE) (1993). Offshore detector siting criterion investigation of detector spacing by LLoyds Register, in Offshore Technology Report OTO 93002. Technical report.

Health and Safety Executive (HSE) (1997). Offshore hydrocarbon release statistics. Technical Report OTO 97 950.

Health and Safety Executive (HSE) (2001). Framework for HSE guidance on gas detectors (on-line checking of flammability monitoring equipment). Technical report.

Health and Safety Executive (HSE) (2003). Offshore hydrocarbon releases statistics and analysis, 2002. HID Statistics HSR 2002 002.

Health and Safety Executive (HSE) (2007). Accident statistics for floating offshore units on the UK Continental Shelf 1980-2005. Technical Report RR567, Det Norske Veritas (DNV).

Health and Safety Executive (HSE) (2011). Fire and gas detection. Technical report. [Online; accessed 19-July-2011].

Hogan, K. and ReVelle, C. (1986). Concepts and applications of backup coverage. *Management Science*, 32(11):1434–1444.

International Electrotechnical Commission (IEC) (2007). *IEC 60079, Explosive atmospheres - Part 29-2: Gas detectors - selection, installation, use and maintenance of detectors for flammable gases and oxygen*.

International Organization for Standardization (ISO) (1999). *ISO 13702 - Petroleum and natural gas industries - Control and mitigation of fires and explosions on offshore production installations - Requirements and guidelines*.

International Organization for Standardization (ISO) (2003). *ISO 10418, Petroleum and natural gas industries – Offshore production installations - Basic surface process safety systems*.

International Society of Automation (ISA) (2003). ANSI/ISA-RP12.13.02 (IEC 61779-6 Mod) - Recommended practice for the installation, operation, and maintenance of combustible gas detection instruments.

International Society of Automation (ISA) (2010). ISA-TR84.00.07-2010: Technical Report Guidance on the Evaluation of Fire, Combustible Gas and Toxic Gas System Effectiveness.

Jia, H., Ordóñez, F., and Dessouky, M. (2007). A modeling framework for facility location of medical services for large-scale emergencies. *IIE Transactions*, 39(1):41–55.

Kelsey, A., Hemingway, M., Walsh, P., and Connolly, S. (2002). Evaluation of flammable gas detector networks based on experimental simulations of offshore, high pressure gas releases. *Process Safety and Environmental Protection*, 80(2):78–86.

Kelsey, A., Ivings, M., Hemingway, M., Walsh, P., and Connolly, S. (2005). Sensitivity studies of offshore gas detector networks based on experimental simulations of high pressure gas releases. *Process Safety and Environmental Protection*, 83(3):262–269.

Krass, D., Berman, O., and Menezes, M. (2003). Reliability issues in median models. In *EURO/INFORMS Joint International*, Istanbul, Turkey.

Krokhmal, P., Zabarankin, M., and Uryasev, S. (2011). Modeling and optimization of risk. *Surveys in Operations Research and Management Science*, 16(2):49–66.

Larson, R. C. (1974). A hypercube queuing model for facility location and redistricting in urban emergency services. *Computers & Operations Research*, 1(1):67–95.

Larson, R. C. (1975). Approximating the performance of urban emergency service systems. *Operations Research*, 23(5):845–868.

Lee, R. and Kulesz, J. (2008). A risk-based sensor placement methodology. *Journal of hazardous materials*, 158(2):417–429.

Lee, S.-D. (2001). On solving unreliable planar location problems. *Computers & Operations Research*, 28(4):329–344.

Legg, S., Benavides-Serrano, A., Siirola, J., Watson, J., Davis, S., Bratteteig, A., and Laird, C. (2012a). A stochastic programming approach for gas detector placement using CFD-based dispersion simulations. *Computers & Chemical Engineering*, 47:194–201.

Legg, S., Siirola, J., Watson, J., Davis, S., Bratteteig, A., and Laird, C. (2012b). A stochastic programming approach for gas detector placement in process facilities. In *Proceedings of the FOCAPO Conference, Savannah*. FOCAPO.

Legg, S. W. (2013). *Stochastic programming approaches for the placement of gas detectors in process facilities*. PhD thesis, Texas A&M.

Legg, S. W., Wang, C., Benavides-Serrano, A. J., and Laird, C. D. (2013). Optimal gas detector placement under uncertainty considering conditional-value-at-risk. *Journal of Loss Prevention in the Process Industries*, 26(3):410–417.

Lei, T. (2010). *Location modeling utilizing closest and generalized closest transport/interaction assignment constructs*. PhD thesis, University of California, Santa Barbara.

Li, X., Zhao, Z., Zhu, X., and Wyatt, T. (2011). Covering models and optimization techniques for emergency response facility location and planning: A review. *Mathematical Methods of Operations Research*, 74(3):281–310. cited By (since 1996) 0.

Lim, M., Daskin, M. S., Bassamboo, A., and Chopra, S. (2010). A facility reliability problem: Formulation, properties, and algorithm. *Naval Research Logistics (NRL)*, 57(1):58–70.

Mak, W., Morton, D., and Wood, R. (1999). Monte carlo bounding techniques for determining solution quality in stochastic programs. *Operations Research Letters*, 24(1):47–56.

Malcolm, S. A. and ReVelle, C. (2005). Models for preserving species diversity with backup coverage. *Environmental Modeling and Assessment*, 10:99–105. 10.1007/s10666-004-5101-3.

Marianov, V. and ReVelle, C. (1992). A probabilistic fire-protection siting model with joint vehicle reliability requirements. *Papers in Regional Science*, 71(3):217–241.

Marianov, V. and Serra, D. (2004). *Facility Location: Applications And Theory*, chapter 4. Springer, 1st edition.

Marianov, V. and Serra, D. (2009). Median problems in networks. *Available at SSRN: http://ssrn.com/abstract=1428362 or http://dx.doi.org/10.2139/ssrn.1428362*.

Marsh (2012). The 100 largest losses 1972-2011. London, United Kingdom.

Marx, J. and Cornwell, J. (2009). The importance of weather variations in a quantitative risk analysis. *Journal of Loss Prevention in the Process Industries*, 22(6):803–808.

Menezes, M., Berman, O., and Krass, D. (2003a). The median problem and reliability of facilities. In *INFORMS Annual Meeting*, Atlanta, GA.

Menezes, M., Berman, O., and Krass, D. (2003b). Minisum with imperfect information. In *INFORMS Annual Meeting*, Atlanta, GA.

Modarres, M., Kaminskiy, M., and Krivtsov, V. (2010). *Reliability engineering and risk analysis*. Taylor and Francis, 2nd edition.

National Fire Protection Association (NFPA) (2007). *NFPA 15, Standard for water spray fixed systems for fire protection*. National Fire Protection Association.

Nolan, D. (2010). *Handbook of fire and explosion protection engineering principles: for oil, gas, chemical and related facilities*. William Andrew Publishing.

Norsk Sokkels Konkuranseposisjon (NORSOK) (2001). *Standard Z-013: Risk and emergency preparedness analysis*. Norway.

Norsk Sokkels Konkuranseposisjon (NORSOK) (2008). *Standard S-001: Technical safety*. Norway, 4th edition.

Obenschain, K., Boris, J., and Patnaik, G. (2004). Using ct-analyst to optimize sensor placement. In *Proceedings of SPIE*, volume 5416, page 14.

Oil & Gas UK (UKOOA) (1995). Guidelines for fire and explosion hazard management. Technical report.

Oil & Gas UK (UKOOA) (2003). Fire and explosion guidance. Part 0: Fire and explosion hazard management.

Owen, S. H. and Daskin, M. S. (1998). Strategic facility location: A review. *European Journal of Operational Research*, 111(3):423–447.

Petroleum Safety Authority (2012). Trends in risk level in the petroleum activity. summary report 2011. norwegian continental shelf. Technical report, Stavanger, Norway.

Pirkul (1989). The uncapacitated facility location problem with primary and secondary facility requirements. *IIE Transactions*, 21(4).

Pirkul, H. and Schilling, D. (1989). The capacitated maximal covering location problem with backup service. *Annals of Operations Research*, 18:141–154. 10.1007/BF02097800.

Reese, J. (2006). Solution methods for the p-median problem: An annotated bibliography. *Networks*, 48(3):125–142.

Repede, J. F. and Bernardo, J. J. (1994). Developing and validating a decision support system for locating emergency medical vehicles in louisville, kentucky. *European Journal of Operational Research*, 75(3):567–581.

Resende, M. G. C. and Werneck, R. F. (2004). A hybrid heuristic for the p-median problem. *Journal of Heuristics*, 10(1):59–88.

ReVelle, C. (1989). Review, extension and prediction in emergency service siting models. *European Journal of Operational Research*, 40(1):58–69.

ReVelle, C. and Eiselt, H. A. (2005). Location analysis: A synthesis and survey. *European Journal of Operational Research*, 165(1):1–19.

ReVelle, C., Eiselt, H. A., and Daskin, M. S. (2008). A bibliography for some fundamental problem categories in discrete location science. *European Journal of Operational Research*, 184(3):817–848.

ReVelle, C. and Hogan, K. (1988). A reliability-constrained siting model with local estimates of busy fractions. *Environment and Planning B: Planning and Design*, 15(2):143–152.

ReVelle, C. and Hogan, K. (1989a). The maximum reliability location problem and α-reliable p-center problem: Derivatives of the probabilistic location set covering problem. *Annals of Operations Research*, 18:155–173. 10.1007/BF02097801.

ReVelle, C. and Hogan, K. (1989b). The maximum availability location problem. *TRANSPORTATION SCIENCE*, 23(3):192–200.

ReVelle, C. and Marianov, V. (1991). A probabilistic fleet model with individual vehicle reliability requirements. *European Journal of Operational Research*, 53(1):93–105.

ReVelle, C. and Swain, R. (1970). Central facilities location. *Geographical Analysis*, 2(1):30–42.

ReVelle, C. and Williams, J. (2004). *Facility Location: Applications And Theory*, chapter 10. Springer, 1st edition.

Rockafellar, R. and Wets, R. (1991). Scenarios and policy aggregation in optimization under uncertainty. *Mathematics of Operations Research*, 16(1):119–147.

Russian agency on technical regulating and metrology (GOST) (1981). *State Standard 13320-81: Gazoanalizatory promy shlennye avtomaticheskie. Obshchie tekhnicheskie usloviya (OTU) (Automatic industrial gas analyzers. Common technical conditions (CTC))*. Moscow.

Schilling, D. A., Elzinga, D. J., Cohon, J., Church, R. L., and ReVelle, C. S. (1979). The team/fleet models for simultaneous facility and equipment siting. *Transportation Science*, 13(2):163–176.

Shen, Z. J. M., Zhan, R., and Zhang, J. (2011). The Reliable Facility Location Problem: Formulations, Heuristics, and Approximation Algorithms. *INFORMS Journal on Computing*.

Sherali, H. D. and Smith, J. C. (2007). An improved linearization strategy for zero-one quadratic programming problems. *Optimization Letters*, 1(1):33–47.

Snyder, L. (2003). *Supply chain robustness and reliability: Models and algorithms*. PhD thesis, Northwestern University, Department of Industrial Engineering & Management Sciences, Evanston, IL.

Snyder, L. and Daskin, M. S. (2002). A reliability-based p-median model. In *INFORMS*, San Jose, CA.

Snyder, L. and Daskin, M. S. (2003). Reliability models for facility location. In *INFORMS*, Atlanta, GA.

Snyder, L. and Daskin, M. S. (2005). Reliability models for facility location: The expected failure cost case. *Transportation Science*, 39(3):400–416.

Snyder, L. V., Atan, Z., Peng, P., Rong, Y., Schmitt, A. J., and Sinsoysal, B. (2010). OR/MS Models for Supply Chain Disruptions: A Review. *SSRN eLibrary*.

Snyder, L. V. and Daskin, M. S. (2007). Models for reliable supply chain network design. In Murray, A. T. and Grubesic, T. H., editors, *Critical Infrastructure*, Advances in Spatial Science, pages 257–289. Springer Berlin Heidelberg.

Spjøtvoll, E. (1985). Estimation of failure rate from reliability data bases. In *Proceedings of the Society of Reliability Engineers, Trondheim, Norway*.

Stiftelsen for industriell og teknisk forskning (SINTEF) (2002). *OREDA, Offshore reliability data handbook*. Det Norske Veritas OREDA Participants, Norway, 4th edition.

Stiftelsen for industriell og teknisk forskning (SINTEF) (2009). *OREDA, Offshore reliability data handbook*, volume 1 – Topside equipment. Det Norske Veritas OREDA Participants, Norway, 5th edition.

Strøm, Ø. and Bakke, J. (1999). Gas detector location. *Safety on offshore installations*, pages London, UK: 3.3.1–3.3.12.

Toregas, C., Swain, R., ReVelle, C., and Bergman, L. (1971). The location of emergency service facilities. *Operations Research*, 19(6):1363–1373.

Weaver, J. R. and Church, R. L. (1985). A median location model with nonclosest facility service. *Transportation Science*, 19(1):58–74.

APPENDIX A

NOTATION AND DATA REQUIREMENTS

Symbol	Meaning
L=$\{1, 2, \ldots, N\}$	Set of candidate detector locations (Indexed by l)
A=$\{1, 2, \ldots, M\}$	Set of hazardous scenarios (Indexed by a)
R=$\{0, 1, \ldots, C\}$	Set of coverage levels (Indexed by r)
$\mathcal{D}_a$=$\{1, 2, \ldots, C\}$	Set of dummy locations for scenario a
$\mathcal{L}_a$=$\{1, 2, \ldots, N_a\}$	Set of candidate detector locations affected by scenario a (Indexed by i)
$\mathcal{R}_{a,i}$=$\{0, 1, \ldots, C_{a,i}\}$	Set of coverage levels for the pair of location i and scenario a (Indexed by r)
$\mathcal{L}^{<}_{a,i}$=$\{1, 2, \ldots, i-1\}$	Set of locations in $\mathcal{L}_a$ with a damage coefficient lower than that of location i (Indexed by j)
$\mathcal{L}^{>}_{a,i}$=$\{i+1, i+2, \ldots, N_a\}$	Set of locations in $\mathcal{L}_a$ with a damage coefficient higher than that of location i (Indexed by j)
s_l	Indicator for an installed detector at location l
$x_{a,i}$	Indicator for location i that first detect scenario a
$y_{a,i}$	Continuos dummy relaxation variable for the pair of location i and scenario a
$z_{a,i}$	Continuos variable representing scenario a r=1 interactions given the selection of location i at r=0
$x_{a,i,r}$	Indicator for location i that first detect scenario a at a coverage level r
$w_{a,i,j}$	Indicator for detection of scenario a by locations i and j at coverage levels 0 and 1, respectively.
$\Psi_{a,i,r}(x)$	Summation of failure combinations resulting in scenario a detection at coverage level r by a detector at location i
$d_{a,i}$	Damage coefficient for scenario a at location i
$\delta_{s,l}$	Euclidean distance from leak source location s to candidate detector location l
α_a	Probability of scenario a
α_s	Probability associated with leak source s
q	Detector unavailability for the given type of detector
$\bar{q}_i$	Time-averaged unavailability for detector at location i
d_{max}	Damage coefficient for undetected scenarios
p	Maximum number of detectors allowed
k	Number of detectors required for leak detection confirmation

Table A.1: Formulations Notation

APPENDIX B

DATA FILES EXAMPLES

This appendix includes simplified examples of each of the three data files which are used in the stochastic programming formulations and algorithms presented in this work.

The first data file contains the leak positions within the volume of study. The name of this file is `DeckLeakPos.py`.

```
# leak_coordinates: Dictionary indexed by scenario number that \
returns a list of the x,y,z leak coordinates
leak_coordinates = {'1':list([25,15,0]),'2':list([55,30,0]),\
'3':list([25,30,0]),'4':list([55,15,0])}
```

The second data file contains the candidate detector locations positions within the volume of study. The name of this file is `DeckLocations.py`.

```
# location_coordinates: Dictionary indexed by candidate detector\
location that returns a list of the x,y,z location coordinates
location_coordinates = {'1':list([30,15,0]),'2':list([40,15,0]),\
'3':list([50,15,0]),'4':list([25,20,0]),'5':list([45,20,0]),\
'6':list([55,20,0]),'7':list([25,25,0]),'8':list([40,25,0]),\
'9':list([55,25,0])}
```

The third data file contains location indexes, scenario indexes, and damage coefficient data parsed from the original GexCon files. The name of this file is `Deck-p-LFL10.py`.

```
# This module contains the following data arrays necessary \
```

to run the model:

```
# all_locations: List of locations indexes in the Gexcon files
# all_scenarios: List of scenario indexes in the Gexcon files
# useful_locations: Set of locations that actually detect
# detected_scenarios: Set of scenarios that are actually detected
# detection_dict: Dictionary indexed by scenarios that return a \
set of all sensor locations that impact that scenario
# damage_dict: Dictionary indexed by scenarios that maps a \
tuple of (scenario_idx: sensor_id, damage_coefficient)
all_locations =  ['1','2','3','4','5','6','7','8','9']
all_scenarios =  ['4','3','2','1']
useful_locations =  set(['1','2','3','4','5','6','7','8','9'])
detected_scenarios =  set(['4','3','2','1'])
detection_dict =  {'4':set(['2','3','4','5','6']),\
'3':set(['4','7']),'2':set(['6','9']),'1':set(['1','2','3','5'])}
damage_dict =  {'4':{'3':5,'2':10,'6':9,'5':15,'4':25},\
'3':{'7':6,'4':12},'2':{'9':4,'6':11},\
'1':{'1':5,'2':9,'5':14,'6':19,'3':23}}
```

APPENDIX C

MAIN DETECTOR PLACEMENT RUN FILE

This is the main Python file. This file performs the following tasks: 1. Query and manipulates the user input, 2. Based on the user inputs, calls the pre-processing file (Appendix D) to process the data, 3. Calls the model files necessary to build the stochastic detector placement problems or runs the random/greedy placement algorithms, 4. Calls the solving routine in the post-processing file (Appendix E) if neccesary, 5. And returns the placement results. This file is named `main.py`.

```
import sys, random, gc, post, pre, plot_locations, layout_paper, \
matplotlib, time, numpy, os, string, coopr.environ

researchDir = string.replace(os.getcwd(),'concrete','')

sys.path.append(researchDir+'/data/testExample/')
sys.path.append(researchDir+'/data/testExample2/')
sys.path.append(researchDir+'/data/interDeck/')
sys.path.append(researchDir+'/data/upperDeck/')
sys.path.append(researchDir+'/data/mainDeck/')
sys.path.append(researchDir+'/data/mezzaDeck/')
sys.path.append(researchDir+'/models/')

if len(sys.argv) != 8:
    print 'Command line should read:'
```

```
    print '~/$ python main.py <seed> <Number of scenarios> \
    <Model> <Number of detectors string> <Number of co-location \
    levels> <Model Test> <Data Set> \n'
    print '<Number of detectors string>: 1,2,3,5-10,100-97'
    print '<Model>: R, GC, VA, LSCP, MCLP, MSDP, SP, SPU, SPUV, \
    oiwSP, tSPqQ, tSPqL1 or tSPqL2'
    print '<Model Test>: SP, SPU, or SPUV.'
    print '<Data Set>: testExample, testExample2, \
    interDeck-p-LFL10, interDeck-p-LFL30, mainDeck-p-LFL10, \
    mainDeck-p-LFL30, upperDeck-p-LFL10, upperDeck-p-LFL30, \
    mezzaDeck-p-LFL10 and mezzaDeck-p-LFL30'
    sys.exit()

seed_number = sys.argv[1]
num_scenarios = int(sys.argv[2])
model_to_use = str(sys.argv[3])
max_sensors = str(sys.argv[4])
colocation = int(sys.argv[5])
model_to_test = str(sys.argv[6])
data_set = str(sys.argv[7])

data = __import__(data_set)
all_locations = data.all_locations
detected_scenarios = data.detected_scenarios
useful_locations = data.useful_locations
detection_dict = data.detection_dict
```

```
damage_dict = data.damage_dict
gap = 0.0

fail_prob = float()
backup_levels = int()
voting_logic = int()
if model_to_use in ('R','C','VA','LSCP','MRP','MCLP',\
'MSDP','SP','oiwSP'):
    fail_prob = 0.0
    backup_levels = 0
    voting_logic = 0
if model_to_use in ('SPU','SPq'):
    fail_prob = 0.1
    backup_levels = 4
    voting_logic = 1
if model_to_use in ('SPqQ','SPqL1','SPqL2','tSPqQ',\
'tSPqL1','tSPqL2'):
    backup_levels = 1
if model_to_use == 'SPUV':
    fail_prob = 0.1
    backup_levels = 4
    voting_logic = 2

max_sensors_list = pre.numberSensors(max_sensors)
num_scenarios,valid_scenarios = pre.validScenarios(num_scenarios,\
data_set,detected_scenarios,detection_dict)
```

```
scenarios_to_use = scenarios_to_use_test = scenarios_available = \
pre.scenariosToUse(seed_number,valid_scenarios,num_scenarios)
useful_locations_plus_dummies,detection_dict,damage_dict,\
max_damage = pre.addDummy(damage_dict,useful_locations,\
detection_dict,backup_levels)
sensor_impact_tuples = pre.sensorImpactTuples(scenarios_to_use,\
useful_locations,detection_dict)
sorted_list_dict,sorted_damage_dict = pre.sortByDamage(damage_dict)
sensor_backup_tuples, Rai, objective_terms = pre.sensorBackupTuples\
(scenarios_to_use,sorted_damage_dict,backup_levels,sensor_impact_\
tuples)
MSDP_damage_dict = pre.MSDPDamageDict(scenarios_to_use,\
location_coordinates,leak_coordinates,colocation_levels)
facility_life = 175316
steps = 100
repairs = 3
repairTimes, qti, qAverage = dict(), dict(), dict()
repairTimes = pre.randomRepairTimes2(useful_locations_plus_dummies\
,facility_life,repairs,1)
qti = pre.qti(useful_locations_plus_dummies,facility_life,steps,\
repairTimes)
qAverage = pre.qAverage(useful_locations_plus_dummies,repairs,\
repairTimes,facility_life)
location_coordinates,leak_coordinates = plot_locations.\
getData(data_set)
```

```
t0model = time.time()


if model_to_use == "R":
    from R import *
if model_to_use == 'VA':
    from VA import *
if model_to_use == 'MSDP':
    from SP import *
    model_instance = SP(scenarios_to_use,all_locations_after_\
    colocation,MSDP_damage_dict,sensor_impact_tuples)
if model_to_use == 'SP':
    from SP import *
    model_instance = SP(scenarios_to_use,useful_locations,\
    detection_dict,damage_dict,sensor_impact_tuples)
if model_to_use == 'SPU':
    from SPU import *
    model_instance = SPU(scenarios_to_use,useful_locations,\
    detection_dict,damage_dict,fail_prob,backup_levels,\
    sensor_impact_tuples,sensor_backup_tuples,max_damage,\
    colocation)
if model_to_use == 'SPUV':
    from SPUV import *
    model_instance = SPUV(scenarios_to_use,useful_locations,\
    detection_dict,damage_dict,fail_prob,backup_levels,\
    voting_logic,sensor_impact_tuples,sensor_backup_tuples,\
    max_damage,Rai,objective_terms,colocation)
```

```
if model_to_use == "GC":
    from GC import *
if model_to_use == "MCLP":
    from MCLP import *
    model_instance = MCLP(scenarios_to_use,useful_locations,\
    detection_dict_before_dummies,voting_logic)
if model_to_use == 'oiwSP':
    from oiwSP import *
    model_instance = oiwSP(scenarios_to_use,useful_locations,\
    detection_dict,damage_dict,qAverage,sensor_impact_tuples,\
    max_damage)
if model_to_use == 'tSPqQ':
    from tSPqQ import *
    model_instance = tSPqQ(scenarios_to_use,useful_locations,\
    detection_dict,damage_dict,qti,sensor_impact_tuples,\
    sensor_backup_tuples,max_damage,Rai,sorted_list_dict,\
    objective_terms,colocation,facility_life)
if model_to_use == 'tSPqL1':
    from tSPqL1 import *
    model_instance = tSPqL1(scenarios_to_use,useful_locations,\
    detection_dict,damage_dict,qti,sensor_impact_tuples,\
    sensor_backup_tuples,max_damage,Rai,sorted_list_dict,\
    objective_terms,colocation,facility_life)
if model_to_use == 'tSPqL2':
    from tSPqL2 import *
    model_instance = tSPqL2(scenarios_to_use,useful_locations,\
```

```
    detection_dict,damage_dict,qti,sensor_impact_tuples,\
    sensor_backup_tuples,max_damage,Rai,sorted_list_dict,\
    objective_terms,colocation,facility_life)
if model_to_use == 'GH':
    from GH import *
if model_to_use == 'IGH':
    from IGH import *


if model_to_use == "R":
    sensors_used = R(scenarios_to_use,useful_locations,\
    sorted_list_dict,damage_dict,max_damage,max_sensors,\
    repairTimes,facility_life)
if model_to_use == "C":
    sensors_used = C(max_sensors,detection_dict,\
    useful_locations,scenarios_to_use)
if model_to_use == 'VA':
    sensors_used = VA(all_locations,location_coordinates,data_set)
if model_to_use == 'GH':
    sensors_used = GH(scenarios_to_use,useful_locations,\
    sorted_list_dict,damage_dict,max_damage,max_sensors,\
    repairTimes,facility_life)
if model_to_use == 'IGH':
    sensors_used = IGH(scenarios_to_use,useful_locations,\
    sorted_list_dict,damage_dict,max_damage,max_sensors,\
    repairTimes,facility_life)
if model_to_use not in ('R','VA','C','GH','IGH'):
```

```
model_instance.max_sensors = max_sensors
model_instance.preprocess()
t1model = time.time() - t0model
t0results = time.time()
(instance,results) = post.solve_instance(model_to_use,\
model_instance,gap)
t1results = time.time() - t0results
(results,objective,sensors_used) = post.print_results\
(instance,results)
```

APPENDIX D

PRE-PROCESSING FILE

This is the pre-processing Python file. This file contains all the functions that are called in the main Python file (`main.py`, Appendix C) to prepare the data files to be used by the detector placement formulations and algorithms. This file is named `pre.py`.

```
def numberSensors(x):
    result = []
    for part in x.split(','):
        if '-' in part:
            a, b = part.split('-')
            a, b = int(a), int(b)
            if a < b:
                result.extend(range(a, b + 1,+1))
            if a > b:
                result.extend(range(a, b - 1,-1))
        else:
            a = int(part)
            result.append(a)
    return result

# Create the set of random scenarios.
def validScenarios(num_scenarios,data_set,detected_scenarios,\
detection_dict):
```

```
    valid_scenarios = set()
    for scen in detected_scenarios:
        if data_set not in ('testExample','testExample2'):
            if len(detection_dict[scen]) > 9:
                valid_scenarios.add(scen)
        else:
            valid_scenarios.add(scen)
    if len(valid_scenarios) < int(num_scenarios):
        num_scenarios = len(valid_scenarios)
    return num_scenarios,valid_scenarios

# Choose the scenarios that are going to be use from the \
pool of random scenarios available and Create a list of \
scenarios and locations to use
def scenariosToUse(seed_number,valid_scenarios,num_scenarios):
    import random
    random.seed(int(seed_number))
    return set(random.sample(valid_scenarios,num_scenarios))

#Add the dummy locations
def addDummy(damage_dict_before_dummies,\
locations_before_dummies,detection_dict_before_dummies,\
backup_levels):
    from copy import deepcopy
    useful_locations_plus_dummies = \
    set(locations_before_dummies)
```

```
detection_dict_after_dummies = \
deepcopy(detection_dict_before_dummies)
damage_dict_after_dummies = \
deepcopy(damage_dict_before_dummies)
max_damage = float()

for scen in damage_dict_after_dummies:
    for sens in damage_dict_after_dummies[scen]:
         max_damage = max(max_damage,float(\
         damage_dict_after_dummies[scen][sens]))
max_damage = max_damage + 9.0

for k,v in damage_dict_after_dummies.items():
    for l in range(0,backup_levels+1):
         useful_locations_plus_dummies.add(''.join\
         (['Failed_to_detect_scenario_',str(k),\
         '_in_level_',str(l)]))
         detection_dict_after_dummies[k].add(''.join\
         (['Failed_to_detect_scenario_',str(k),\
         '_in_level_',str(l)]))
         damage_dict_after_dummies[k][''.\
         join(['Failed_to_detect_scenario_',str(k),\
         '_in_level_',str(l)])] = max_damage+0.001*l

return useful_locations_plus_dummies,\
detection_dict_after_dummies,damage_dict_after_dummies\
```

```
    ,max_damage

#Create the sensor impact tuples
def sensorImpactTuples(scenarios_to_use,sensors_to_use,\
detection_dict):
    full_sensor_impact_tuples = [(scenario, sensor) for scenario \
    in scenarios_to_use for sensor in detection_dict[scenario]]
    sensor_impact_tuples = list()
    for (scenario,sensor) in full_sensor_impact_tuples:
        if scenario in scenarios_to_use:
            sensor_impact_tuples.append((scenario,sensor))
    return sensor_impact_tuples

#Sorting the damage coefficients, outer_keys = scenarios, \
inner_keys = locations affecting given scenario
def sortByDamage(damage_dict):
    sorted_damage_dict = dict()
    sorted_list_dict = dict()
    outer_keys = damage_dict.keys()
    for outer_key in outer_keys:
         inner_keys = damage_dict[outer_key].keys()
         sort_dict = [x for x in damage_dict[outer_key].\
         iteritems()]
         sort_dict.sort(key=lambda x:x[1])
         sorted_damage_dict[outer_key] = sort_dict
    for scen in sorted_damage_dict.keys():
```

```
        sorted_list_dict[scen] = [x[0] for x in \
        sorted_damage_dict[scen]]
    return sorted_list_dict, sorted_damage_dict

#Return sensor_backup_tuples, Rai, and objective_terms
#sensor_backup_tuples: Tuples of the possible scenario, \
location, and backup_level combinations
#Rai: Dictionary indexed by (scenario,location) that returns \
a set of the possible backup levels that the given \
(scenario,location) tuple can occupy
#objective_terms: Dictionary indexed by (scenario,backup_level) \
that returns a set of the locations that can occupy that \
backup_level for the given scenario

def sensorBackupTuples(scenarios_to_use,sorted_damage_dict,\
backup_levels,sensor_impact_tuples):
    sensor_backup_tuples = list()
    Rai = dict(((scen,sens),[]) for scen,sens in \
    sensor_impact_tuples)
    objective_terms = dict(((scen,r),[]) for scen in \
    scenarios_to_use for r in range(0,backup_levels+1))
    for scenario in scenarios_to_use:
        for x in sorted_damage_dict[scenario]:
            for r in range(0,backup_levels+1):
                if r<= sorted_damage_dict[scenario].index(x)\
                and not 'Failed' in str(x[0]):
```

```
                        sensor_backup_tuples.append((scenario,\
                        x[0],r))
                        Rai[scenario,x[0]].append(r)
                        objective_terms[scenario,r].append(x[0])
                   if r <= sorted_damage_dict[scenario].index(x)\
                   and 'Failed' in str(x[0]):
                        if ''.join(['_in_level_',str(r)]) in \
                        str(x[0]):
                             sensor_backup_tuples.append((scenario,\
                             x[0],r))
                             Rai[scenario,x[0]].append(r)
                             objective_terms[scenario,r].\
                             append(x[0])
    return sensor_backup_tuples, Rai, objective_terms

#Create a damage dictionary where the damage \
is the distance from the leak to the candidate detector location
def MSDPDamageDict(scenarios_to_use,location_coordinates \
,leak_coordinates,colocation_levels):
     MSDPDamageDict = dict((scen,{}) for scen in scenarios_to_use)
     for scenario in scenarios_to_use:
          scenarioDict = dict()
          x_scen = leak_coordinates[scenario[:-4]][0]
          y_scen = leak_coordinates[scenario[:-4]][1]
          z_scen = leak_coordinates[scenario[:-4]][2]
          for sens in location_coordinates:
```

```
            x_sens = location_coordinates[sens][0]
            y_sens = location_coordinates[sens][1]
            z_sens = location_coordinates[sens][2]
            dist = ((x_scen - x_sens)**2 + (y_scen - \
            y_sens)**2 + (z_scen - z_sens)**2)**0.5
            scenarioDict[sens] = dist
        MSDPDamageDict[scenario] = scenarioDict

    MSDPDamageDictAfterColocation = dict((scen,{}) \
    for scen in scenarios_to_use)
    for scen in MSDPDamageDict:
        colocationDict = dict()
        for k in range(1,colocation_levels+1):
            for sens, damage in MSDPDamageDict[scen].items():
                colocationDict['-'.join([str(sens),str(k)])]\
                = damage+1*(k-1)
        MSDPDamageDictAfterColocation[scen] = \
        colocationDict
    return MSDPDamageDictAfterColocation

def randomRepairTimes(sensors,\
facility_life,numberOfRepairs,seed):
    tr = dict()
    import random
    random.seed(seed)
    for sensor in sensors:
```

```
        tr[sensor] = sorted(random.sample(range(int(
        facility_life)),numberOfRepairs) + [0])
     return tr

def qt(sens,t,repairs):
     if 'Failed' in sens:
          return 0.0
     else:
          trleft = [i for i in repairs if i <= t]
          return 1.73 * 10**(-6) * (t - min(trleft, \
          key=lambda x:abs(x-t)))

def qti(sensors,L,n,tr):
    qti = dict()
    for k in range(0,n+1):
        t = k*L/float(n)
        q = dict()
        for sens in sensors:
            q[sens] = qt(sens,t,tr[sens])
        qti[t] = q
    return qti

def qAverage(sensors_used,numberOfRepairs,tr,facility_life):
    qAverage = dict()
    for sens in sensors_used:
         qArea = 0
```

```
        for repair in range(1,numberOfRepairs+1):
            qArea += (tr[sens][repair] - tr[sens][repair-1])*\
            qt(sens,tr[sens][repair]-0.000001,tr[sens]) / 2
        qArea += (facility_life - tr[sens][numberOfRepairs])*\
        qt(sens,facility_life,tr[sens]) / 2
        qAverage[sens] = qArea / facility_life
    return qAverage

def individualq(useful_locations_plus_dummies):
    q = dict((sens,()) for sens in useful_locations_plus_dummies)
    for sens in useful_locations_plus_dummies:
        if 'Failed' in sens:
            q[sens] = 0.05
        else:
            q[sens] = 0.05
    return q
```

APPENDIX E

POST-PROCESSING FILE

This is the post-processing Python file. This file contains the functions necessary to build the model instances. Also, it contains the functions necessary to analyze the results obtained from the main Python file (`main.py`, Appendix C). This file is named `post.py`.

```
from coopr.pyomo import *

from pyutilib.misc import Options

from coopr.opt import SolverFactory

import math, sys, itertools, coopr.environ


def solve_instance(model,instance,gap):

    if model == 'LSCP' or model == 'MRP'  or model == 'MCLP' \

    or model == 'MSDP' or model == 'SP' or model == 'SPU' \

    or model == 'SPUV' or model == 'oiwSP' or model == 'tSPqL1' \

    or model == 'tSPqL2':

        opt = SolverFactory('gurobi',solver_io='lp')

        opt.set_options(' '.join(['TimeLimit=100000','Threads=1']))

    if model == 'SPq':

        opt = SolverFactory('cplex',solver_io='lp')

    if model == 'SPqQ' or model == 'tSPqQ':

        opt = SolverFactory('cplex',solver_io='lp')

    results = opt.solve(instance,tee=True,keepfiles=False,\

    warmstart=False)
```

```
    return instance, results


def print_results(instance,results):
    stat = str(results.Solver.Termination_condition)
    sensors_used = set()
    if stat == 'optimal' or stat == 'maxTimeLimit':
        instance.load(results)
        objective = (value(instance.obj))
        for sens in instance.s.keys():
            if instance.s[sens].stale == False:
                if value(instance.s[sens]) > 0.01:
                    sensors_used.add(sens)
    elif stat == 'infeasible':
        objective = -1
        num_sens = -1
        fraction_detected = -1
        print 'Problem infeasible.'
    else:
        print 'Bad things are happening. Trust nothing.'
        sys.exit()
    return results,objective,sensors_used


# Calculate the fraction of scenarios covered
def fraction_covered(scenarios_to_use,sensors_to_use,\
detection_dict,sensors_used,voting_logic):
    scenarios_detected = float()
```

```
    for scen in scenarios_to_use:
        scenario_sensors = float()
        for sens in sensors_used:
            if sens in detection_dict[scen]:
                scenario_sensors = scenario_sensors + 1
        if scenario_sensors >= voting_logic:
            scenarios_detected = scenarios_detected + 1
    return scenarios_detected/len(scenarios_to_use)


def evaluation(scenarios_to_use,sensors_used,sorted_list_dict,\
damage_dict,qi,k,max_damage):
    expr = 0.0
    for scen in scenarios_to_use:
        ProbSum = 0.0
        ordered_used_in_La = [sens for sens in sorted_list_dict\
        [scen] if sens in sensors_used]
        for sensa in ordered_used_in_La:
            LaiLess = set(ordered_used_in_La[:ordered_used_in_La.\
            index(sensa)])
            if len(LaiLess) >= k-1:
                combinations = itertools.combinations(LaiLess,\
                len(LaiLess)-k+1)
                for combination in combinations:
                    if len(combination) <= 100:
                        ProbProd = 1.0
                        for sensb in combination:
```

```
                              ProbProd *= qi[sensb]
for sensc in LaiLess.difference(combination):
                              ProbProd *= 1.0-qi[sensc]
                          expr += damage_dict[scen][sensa]*(1.0 - \
                          qi[sensa]) * ProbProd
                          ProbSum += (1.0 - qi[sensa]) * ProbProd
        expr += max_damage * (1-ProbSum)
    return expr/len(scenarios_to_use)


def evaluationDynamic(scenarios_to_use_test,sensors_used,\
sorted_list_dict,damage_dict,repairTimes,k,max_damage,L):
    import pre
    n = 100
    expr = 0
    qti = pre.qti(sensors_used,L,n,repairTimes)
    expr += evaluation(scenarios_to_use_test,sensors_used,\
    sorted_list_dict,damage_dict,qti[0],k,max_damage)/2
    expr += evaluation(scenarios_to_use_test,sensors_used,\
    sorted_list_dict,damage_dict,qti[L],k,max_damage)/2
    for counter in range(1,n):
        expr += evaluation(scenarios_to_use_test,sensors_used,\
        sorted_list_dict,damage_dict,qti[counter*L/float(n)],k,\
        max_damage)
    return expr / n
```

APPENDIX F

FORMULATION SP: PYOMO MODEL FILE

This is the Pyomo file that contains detector placement formulation SP as described in Section 2.1. This file is named `SP.py`.

```
from coopr.pyomo import *

def SP(scenarios_to_use, useful_locations, sensor_impact_dict, \
damage_coeffs, sensor_impact_tuples):

    model = ConcreteModel()
    model.s = Var(useful_locations, within=Binary)
    model.x = Var(sensor_impact_tuples, bounds=(0.0,1.0))

    model.max_sensors = Param(mutable=True, initialize=0)

    def obj_rule (model):
        return 1.0/len(scenarios_to_use)*(sum(damage_coeffs[scen]\
        [sens]*model.x[(scen, sens)] for (scen,sens) in \
        sensor_impact_tuples))
    model.obj = Objective(rule = obj_rule)

    def max_sensors_rule(model):
        return sum(model.s[sens] for sens in useful_locations) \
```

```
    <= model.max_sensors
model.max_sensors_con = Constraint(rule = max_sensors_rule)


def sensor_binary_rule(model, scen, sens):
    if 'Failed' in str(sens):
        return Constraint.Skip
    else:
        return model.x[(scen, sens)] <= model.s[sens]
model.sensor_binary_con = Constraint(sensor_impact_tuples, \
rule = sensor_binary_rule)


def event_detection_rule(model, scen):
    return sum(model.x[(scen,sens)] for sens in \
    sensor_impact_dict[scen]) == 1
model.event_detection_con = Constraint(scenarios_to_use, \
rule = event_detection_rule)


return model
```

APPENDIX G

FORMULATION SP-U: PYOMO MODEL FILE

This is the Pyomo file that contains detector placement formulation SP-U as described in Section 3.1. This file is named `SPU.py`.

```
from coopr.pyomo import *
import math

def SPU(scenarios_to_use,location_s,sensor_impact_dict,\
damage_coeffs,q,backup_levels,sensor_impact_tuples,\
sensor_backup_tuples,max_damage,colocation):

    model = ConcreteModel()
    model.s = Var(location_s,bounds=(0.0,colocation),\
    within=Integers)
    model.R = RangeSet(0,backup_levels)
    model.x = Var(sensor_backup_tuples,bounds=(0.0,1.0))
    model.max_sensors = Param(mutable=True,initialize=0)

    def obj_rule (model):
        return 1.0/len(scenarios_to_use)*sum(math.pow(q,r)*\
        (1-q)*damage_coeffs[scen][sens]*model.x[(scen,sens,r)]\
        for (scen,sens,r) in sensor_backup_tuples)
    model.obj = Objective(rule = obj_rule)
```

```
def event_detection_rule(model,scen,r):
    return sum(model.x[(scen,sens,r)] for sens in \
    sensor_impact_dict[scen] if (scen,sens,r) in \
    sensor_backup_tuples) == 1
model.event_detection_con = Constraint(scenarios_to_use,\
model.R,rule = event_detection_rule)

def max_sensors_rule(model):
    return sum(model.s[sens] for sens in location_s)\
    <= model.max_sensors
model.max_sensors_con = Constraint(rule = max_sensors_rule)

def coverage_priority_rule(model,scen,sens):
    if 'Failed' in str(sens):
        return Constraint.Skip
    else:
        return sum(model.x[(scen,sens,r)] for r in model.R\
        if (scen,sens,r) in sensor_backup_tuples) <=\
        model.s[sens]
model.coverage_priority_con =\
Constraint(sensor_impact_tuples,
rule = coverage_priority_rule)

return model
```

APPENDIX H

FORMULATION SP-UV: PYOMO MODEL FILE

This is the Pyomo file that contains detector placement formulation SP-UV as described in Section 4.1. This file is named `SPUV.py`.

```
from coopr.pyomo import *
import math

def binomialCoefficient(n, k):
    if k < 0 or k > n:
        return 0
    if k > n - k:
        k = n - k
    c = 1
    for i in range(k):
        c = c * (n - (k - (i+1)))
        c = c // (i+1)
    return c

def SPUV(scenarios_to_use, sensors_to_use,sensor_impact_dict,\
damage_coeffs,q,backup_levels,k,sensor_impact_tuples,\
sensor_backup_tuples,max_damage,Rai,objective_terms,\
colocation):
```

```
def domain_rule(model,scenario,sensor,level):
    if level <= k-1:
        return Binary
    if level > k-1:
        return Reals

model = ConcreteModel()
model.s = Var(sensors_to_use,bounds=(0.0,colocation),\
within=Integers)
model.x = Var(sensor_backup_tuples, bounds=(0.0,1.0),\
domain=domain_rule)
model.R = RangeSet(0,backup_levels)
model.R5 = RangeSet(0,k-1)
model.max_sensors = Param(mutable=True, initialize=0)

def obj_rule (model):
    return  1.0/len(scenarios_to_use)*\
    sum(binomialCoefficient(r,r-k+1)*math.pow(q,r-k+1)*\
    math.pow(1-q,k) * damage_coeffs[scen][sens]*\
    model.x[scen,sens,r] for (scen,sens,r) in\
    sensor_backup_tuples)
model.obj = Objective(rule = obj_rule)

def event_detection_rule2(model, scen, r):
    return sum(model.x[scen,sens,r] for sens in\
    objective_terms[scen,r]) == 1
```

```
model.event_detection_con2 = Constraint(scenarios_to_use,\
model.R, rule = event_detection_rule2)

def max_sensors_rule(model):
    return sum(model.s[sens] for sens in sensors_to_use)\
    <= model.max_sensors
model.max_sensors_con = Constraint(rule = max_sensors_rule)

def coverage_priority_rule1(model,scen,r):
    return sum(damage_coeffs[scen][sens]*model.x[scen,sens,r]\
    for sens in objective_terms[scen,r]) <=\
    sum(damage_coeffs[scen][sens]*model.x[scen,sens,r+1] \
    for sens in objective_terms[scen,r+1])
model.coverage_priority_con1 = Constraint(scenarios_to_use,\
model.R5, rule = coverage_priority_rule1)

def coverage_priority_rule2(model, scen, sens):
    if 'Failed' in str(sens):
        return Constraint.Skip
    else:
        return sum(model.x[scen,sens,r] for r in\
        Rai[scen,sens]) <= model.s[sens]
model.coverage_priority_con2 = Constraint(sensor_impact_tuples,\
rule = coverage_priority_rule2)

return model
```

APPENDIX I

GREEDY COVERAGE (GC) ALGORITHM: PYTHON FILE

This is the Python file that implements the Greedy Coverage (GC) algorithm described in Section 5.1.4. This file is named `GC.py`.

```
import post

def GC(max_sensors,detection_dict,locations,scenarios):

    sensors_used = set()
    k = 1
    for i in range(1,max_sensors+1):
        maximum = 0.0
        for sens in locations:
            frac = post.fraction_covered(scenarios,locations,\
            detection_dict,sensors_used.union(set([str(sens)])),k)
            if frac > maximum:
                maximum = float(frac)
                newSens = str(sens)
        sensors_used.add(newSens)

        if frac == 1.0:
            k += 1
            print k
```

```
    if k == 3:
        print 'k=3 for ', len(sensors_used), 'sensors'
        break

return sensors_used
```

APPENDIX J

FORMULATION MCLP: PYOMO MODEL FILE

This is the Pyomo file that contains detector placement formulation MCLP as described in Section 5.1.4. This file is named `MCLP.py`.

```
from coopr.pyomo import *

def MCLP(scenarios_to_use, sensors_to_use, sensor_impact_dict, \
voting_logic):

    model = ConcreteModel()
    model.x = Var(scenarios_to_use, within=Binary)
    model.s = Var(sensors_to_use, within=Binary)

    model.max_sensors = Param(mutable=True, initialize=0)

    def obj_rule (model):
        return sum(model.x[scen] for scen in scenarios_to_use)
    model.obj = Objective(rule = obj_rule, sense = maximize)

    def sensor_coverage_rule(model, scen):
        return sum(model.s[sens] for sens in \
        sensor_impact_dict[scen]) >= voting_logic * model.x[scen]
    model.sensor_coverage_con = Constraint(scenarios_to_use,rule \
```

```
= sensor_coverage_rule)

def max_sensors_rule(model):
    return sum(model.s[sens] for sens in sensors_to_use) <= \
    model.max_sensors
model.max_sensors_con = Constraint(rule = max_sensors_rule)

return model
```

APPENDIX K

FORMULATION SPqt-Q: PYOMO MODEL FILE

This is the Pyomo file that contains detector placement formulation SPqt-Q as described in Section 7.3.1. This file is named `tSPqQ.py`.

```
from coopr.pyomo import *
import math
import numpy
import scipy

def tSPqQ(scenarios_to_use,sensors_to_use,sensor_impact_dict,\
damage_coeffs, qti, sensor_impact_tuples, sensor_backup_tuples,\
max_damage,Rai,sorted_list_dict,objective_terms,colocation,L):

    Q_tuples = list()
    backup_tuples_0 = list()
    for scen in scenarios_to_use:
        for sensa in objective_terms[(scen,0)]:
            backup_tuples_0.append((scen,sensa,0))
            indexa = sorted_list_dict[scen].index(sensa)
            for sensb in objective_terms[(scen,1)]:
                if sorted_list_dict[scen].index(sensb) > indexa:
                    Q_tuples.append((scen,sensa,sensb))
```

```
model = ConcreteModel()
model.s = Var(sensors_to_use, bounds=(0.0,colocation), \
within=Integers)
model.R = RangeSet(0,1)
model.R1 = RangeSet(0,0)
model.x = Var(sensor_backup_tuples, within=Binary)
model.max_sensors = Param(mutable=True, initialize=0)

def Et1(model,t):
    return (sum(damage_coeffs[scen][sens] * \
    (1-qti[t][sens]) * model.x[(scen,sens,r)] for \
    (scen,sens,r) in backup_tuples_0) + \
    sum(damage_coeffs[scen][sensb] * (1-qti[t][sensb]) * \
    qti[t][sensa] * model.x[(scen,sensa,0)] * \
    model.x[(scen,sensb,1)] for (scen,sensa,sensb) in \
    Q_tuples))/len(scenarios_to_use)

def obj_rule (model):
    return (Et1(model,0)/2 + sum(Et1(model,t) for t in qti \
    if t != 0 and t != L) + Et1(model,L)/2) / len(qti)
model.obj = Objective(rule = obj_rule)

def event_detection_rule(model, scen, r):
    return sum(model.x[scen,sens,r] for sens in\
    objective_terms[scen,r]) == 1
model.event_detection_con = Constraint(scenarios_to_use,\
```

```
model.R, rule = event_detection_rule)

def max_sensors_rule(model):
    return sum(model.s[sens] for sens in sensors_to_use)\
    <= model.max_sensors
model.max_sensors_con = Constraint(rule = max_sensors_rule)

def coverage_priority_rule2(model, scen, sens):
    if 'Failed' in str(sens):
        return Constraint.Skip
    else:
        return sum(model.x[scen,sens,r] for r in
        Rai[scen,sens]) <= model.s[sens]
model.coverage_priority_con2 = Constraint(\
sensor_impact_tuples, rule = coverage_priority_rule2)

def coverage_priority_rule1(model,scen,r):
    return sum(damage_coeffs[scen][sens]*\
    model.x[scen,sens,r] for sens in objective_terms\
    [scen,r]) <= sum(damage_coeffs[scen][sens]*\
    model.x[scen,sens,r+1]
    for sens in objective_terms[scen,r+1])
model.coverage_priority_con1 = Constraint(scenarios_to_use,\
model.R1, rule = coverage_priority_rule1)

return model
```

APPENDIX L

FORMULATION SPqt-L_1: PYOMO MODEL FILE

This is the Pyomo file that contains detector placement formulation SPqt-L_1 as described in Section 7.3.2. This file is named `tSPqL1.py`.

```
from coopr.pyomo import *
import math
import numpy

def tSPqL1(scenarios_to_use,sensors_to_use,sensor_impact_dict,\
damage_coeffs, qti, sensor_impact_tuples, sensor_backup_tuples,\
max_damage\,Rai,sorted_list_dict,objective_terms,colocation,L):

    Q_tuples = list()
    backup_tuples_0 = list()
    for scen in scenarios_to_use:
        for sensa in objective_terms[(scen,0)]:
            backup_tuples_0.append((scen,sensa,0))
            indexa = sorted_list_dict[scen].index(sensa)
            for sensb in objective_terms[(scen,1)]:
                if sorted_list_dict[scen].index(sensb) > indexa:
                    Q_tuples.append((scen,sensa,sensb))

    model = ConcreteModel()
```

```
model.s = Var(sensors_to_use, bounds=(0.0,colocation), \
within=Integers)
model.R = RangeSet(0,1)
model.R1 = RangeSet(0,0)
model.x = Var(sensor_backup_tuples, within=Binary)
model.w = Var(Q_tuples,bounds=(0.0,1.0))
model.max_sensors = Param(mutable=True, initialize=0)

def Et1(model,t):
    return (sum(damage_coeffs[scen][sens] * \
    (1-qti[t][sens]) * model.x[(scen,sens,r)] for \
    (scen,sens,r) in backup_tuples_0) + \
    sum(damage_coeffs[scen][sensb] * (1-qti[t][sensb]) * \
    qti[t][sensa] * model.w[(scen,sensa,sensb)] for \
    (scen,sensa,sensb) in Q_tuples))/len(scenarios_to_use)

def obj_rule (model):
    return (Et1(model,0)/2 + sum(Et1(model,t) for t in qti \
    if t != 0 and t != L) + Et1(model,L)/2) / len(qti)
model.obj = Objective(rule = obj_rule)

def additional_rule1(model,scen,sensa,sensb):
    return model.w[(scen,sensa,sensb)] >= \
    model.x[(scen,sensa,0)] + model.x[(scen,sensb,1)] - 1
model.additional_con1 = Constraint(Q_tuples, rule = \
additional_rule1)
```

```
def event_detection_rule(model, scen, r):
    return sum(model.x[scen,sens,r] for sens in \
    objective_terms[scen,r]) == 1
model.event_detection_con = Constraint(scenarios_to_use, \
model.R, rule = event_detection_rule)

def max_sensors_rule(model):
    return sum(model.s[sens] for sens in sensors_to_use) \
    <= model.max_sensors
model.max_sensors_con = Constraint(rule = max_sensors_rule)

def coverage_priority_rule2(model, scen, sens):
    if 'Failed' in str(sens):
        return Constraint.Skip
    else:
        return sum(model.x[scen,sens,r] for r in \
        Rai[scen,sens]) <= model.s[sens]
model.coverage_priority_con2 = Constraint(\
sensor_impact_tuples, rule = coverage_priority_rule2)

def coverage_priority_rule1(model,scen,r):
    return sum(damage_coeffs[scen][sens]*\
    model.x[scen,sens,r] for sens in \
    objective_terms[scen,r]) <= \
    sum(damage_coeffs[scen][sens]*\
```

```
    model.x[scen,sens,r+1] for sens in \
    objective_terms[scen,r+1])
model.coverage_priority_con1 = Constraint(scenarios_to_use, \
model.R1, rule = coverage_priority_rule1)

return model
```

APPENDIX M

FORMULATION SPqt-L_2: PYOMO MODEL FILE

This is the Pyomo file that contains detector placement formulation SPqt-L_2 as described in Section 7.3.3. This file is named `tSPqL2.py`.

```
from coopr.pyomo import *

import math

import numpy


def tSPqL2(scenarios_to_use,sensors_to_use,sensor_impact_dict,\
damage_coeffs, qti, sensor_impact_tuples, sensor_backup_tuples,\
max_damage\,Rai,sorted_list_dict,objective_terms,colocation,L):

    backup_tuples_0 = list()
    sensor_impact_tuples_0 = list()
    for (scen,i,r) in sensor_backup_tuples:
        if r == 0:
            backup_tuples_0.append((scen,i,r))
            sensor_impact_tuples_0.append((scen,i))

    scenario_ir = dict((scen,list()) for scen in scenarios_to_use)
    for scen in scenarios_to_use:
        for sens in sensor_impact_dict[scen]:
            for r in Rai[scen,sens]:
```

```
            scenario_ir[scen].append((sens,r))

model = ConcreteModel()
model.s = Var(sensors_to_use, bounds=(0.0,colocation), \
within=Integers)
model.R = RangeSet(0,1)
model.R1 = RangeSet(0,0)
model.x = Var(sensor_backup_tuples, within=Binary)
model.y  = Var(sensor_impact_tuples_0, within=PositiveReals)
model.s2 = Var(sensor_impact_tuples_0, within=PositiveReals)

model.max_sensors = Param(mutable=True, initialize=0)

def Et1(model,t):
    return sum(damage_coeffs[scen][i] * (1-qti[t][i]) * \
    model.x[(scen,i,r)] for (scen,i,r) in backup_tuples_0)

def obj_rule (model):
    return ((Et1(model,0)/2 + sum(Et1(model,t) for t in qti \
    if t != 0 and t != L) + Et1(model,L)/2) / len(qti) + \
    sum(model.s2[(scen,i)] for (scen,i) in \
    sensor_impact_tuples_0)) / len(scenarios_to_use)
model.obj = Objective(rule = obj_rule)

def Et2(model,scen,sensa,t):
    indexa = sorted_list_dict[scen].index(sensa)
```

```
    return sum(damage_coeffs[scen][sensb]*(1-qti[t][sensb]) * \
    qti[t][sensa] * model.x[(scen,sensb,1)] for sensb in \
    objective_terms[(scen,1)] if \
    sorted_list_dict[scen].index(sensb) > indexa)

def linear_rule_1(model,scen,sensa):
     return (Et2(model,scen,sensa,0)/2 + \
     sum(Et2(model,scen,sensa,t) for t in qti \
     if t!=0 and t!=L) + Et2(model,scen,sensa,L)/2) / len(qti)\
     == model.y[(scen,sensa)] + model.s2[(scen,sensa)]
model.linear_con_1 = Constraint(\
sensor_impact_tuples_0,rule = linear_rule_1)

def linear_rule_2(model,scen,i): #15c
    return model.y[(scen,i)] <= max_damage*(1-\
    model.x[(scen,i,0)])
model.linear_con_2 = Constraint(\
sensor_impact_tuples_0,rule = linear_rule_2)

def event_detection_rule(model, scen, r):
    return sum(model.x[scen,sens,r] for sens in \
    objective_terms[scen,r]) == 1
model.event_detection_con = Constraint(\
scenarios_to_use, model.R, rule = event_detection_rule)

def max_sensors_rule(model):
```

```
return sum(model.s[sens] for sens in sensors_to_use) \
<= model.max_sensors
    model.max_sensors_con = Constraint(rule = max_sensors_rule)

    def coverage_priority_rule2(model, scen, sens):
        if 'Failed' in str(sens):
            return Constraint.Skip
        else:
            return sum(model.x[scen,sens,r] for r in \
            Rai[scen,sens]) <= model.s[sens]
    model.coverage_priority_con2 = Constraint(\
    sensor_impact_tuples, rule = coverage_priority_rule2)

    def coverage_priority_rule1(model,scen,r):
        return sum(damage_coeffs[scen][sens]*\
        model.x[scen,sens,r] for sens in \
        objective_terms[scen,r]) <= \
        sum(damage_coeffs[scen][sens]*\
        model.x[scen,sens,r+1] for sens in \
        objective_terms[scen,r+1])
    model.coverage_priority_con1 = Constraint(\
    scenarios_to_use, model.R1, rule = \
    coverage_priority_rule1)

    return model
```

APPENDIX N

FORMULATION oiwSP: PYOMO MODEL FILE

This is the Pyomo file that contains detector placement formulation oiwSP as described in Section 7.5. This file is named `oiwSP.py`.

```
from coopr.pyomo import *

def oiwSP(scenarios_to_use,locations_before_dummies,\
sensor_impact_dict, damage_coeffs, qi, \
sensor_impact_tuples,max_damage):

    model = ConcreteModel()
    model.s = Var(locations_before_dummies, within=Binary)
    model.x = Var(sensor_impact_tuples, bounds=(0.0,1.0))
    model.max_sensors = Param(mutable=True, initialize=0)

    def obj_rule (model):
        return 1.0/len(scenarios_to_use) * \
        sum((damage_coeffs[scen][sens]*(1.0-qi[sens]) + \
        max_damage*qi[sens]) * model.x[(scen, sens)] for \
        (scen,sens) in sensor_impact_tuples)
    model.obj = Objective(rule = obj_rule)

    def max_sensors_rule(model):
```

```
    return sum(model.s[sens] for sens in \
    locations_before_dummies) <= model.max_sensors
model.max_sensors_con = Constraint(rule = max_sensors_rule)

def sensor_binary_rule(model, scen, sens):
    if 'Failed' in str(sens):
        return Constraint.Skip
    else:
        return model.x[(scen, sens)] <= model.s[sens]
model.sensor_binary_con = Constraint(sensor_impact_tuples, \
rule = sensor_binary_rule)

def event_detection_rule(model, scen):
    return sum(model.x[(scen,sens)] for sens in \
    sensor_impact_dict[scen]) == 1
model.event_detection_con = Constraint(scenarios_to_use, \
rule = event_detection_rule)

return model
```

APPENDIX O

GREEDY HEURISTIC (GH): PYTHON FILE

This is the Python file that implements the Greedy Heuristic (GH) described in Section 7.5. This file is named `GH.py`.

```
import post

def GH(scenarios_to_use,useful_locations,sorted_list_dict,\
damage_dict,max_damage,max_sensors,repairTimes,L):

    sensors_used = set()
    minimun = 10000.0
    for i in range(max_sensors):
        newSens = set()
        for sens in useful_locations.difference(sensors_used):
            value = post.evaluationDynamic(scenarios_to_use,\
            sensors_used.union(set([sens])),sorted_list_dict,\
            damage_dict,repairTimes,1,max_damage,L)
            if value < minimun:
                minimun = value
                newSens = sens
        sensors_used = sensors_used.union(set([newSens]))

    return sensors_used
```

APPENDIX P

IMPROVED GREEDY HEURISTIC (IGH): PYTHON FILE

This is the Python file that implements the Improved Greedy Heuristic (IGH) described in Section 7.5. This file is named `IGH.py`.

```
import post

def IGH(scenarios_to_use,useful_locations,sorted_list_dict,\
damage_dict,max_damage,max_sensors,repairTimes,L):

GH = {1: set(['45']),2:set(['45','137']),3:set(['59','45','137']),\
4: set(['137', '45', '59', '498']), 5: set(['59', '45', '137',\
'498','480']) #GH Results

    V = set(GH[max_sensors])
    minimum = post.evaluationDynamic(scenarios_to_use,\
    V,sorted_list_dict,damage_dict,repairTimes,1,\
    max_damage,L)

    for sens in GH[max_sensors]:
        newSens = set([sens])
        for loc in useful_locations:
            value = post.evaluationDynamic(scenarios_to_use,\
            V.difference(set([sens])).union(set([loc])),\
```

```
        sorted_list_dict,damage_dict,repairTimes,1,max_damage,L)
        if value < minimum:
            newSens = set([loc])
    V = V.difference(set([sens])).union(newSens)
return V
```