ABSTRACT

TRYBY, MICHAEL EUGENE. Contaminant Source Monitoring and Characterization in Water Distribution Systems. (Under the direction of S. Ranji Ranjithan and G. Mahinthakumar.)

Recent anxiety surrounding the security of the nation's critical water infrastructure has increased interest in monitoring and characterization problems in water distribution system contexts. This dissertation addresses the difficulties associated with solving source identification inverse problems in water distribution systems due to ill-conditioning. Specifically, novel solution and monitoring design methods are investigated. A description of the source identification problem is developed and it is shown that the problem can be formulated as a discrete linear inverse problem. Such problems are well understood and powerful tools exist for their analysis and solution. Regularization is a technique for stabilizing the solution of ill-conditioned inverse problems. Typically, an inverse problem is regularized by incorporating additional information into the problem prior to solution. The form of the problem is modified and an approximate solution is sought. The effect of regularization methods on the source identification solution is investigated. It is concluded that regularizing for sparse solutions is most meaningful for the contaminant source identification problems in water distribution systems.

A novel simulation optimization based solution approach for environmental monitoring and characterization problems is also investigated. The approach utilizes global search heuristics such as evolutionary algorithms as opposed to classical gradient based algorithms. Simulation optimization requires many evaluations of the simulation model as the search progresses making them computationally intensive. Evolutionary algorithms are amenable to parallelization and in this work they are combined with the computing power of computational grids making the solution approach tractable. A general framework for parallel evolutionary algorithms is developed with the specific intent of solving environmental monitoring and characterization problems. The solution and computational performance achieved using the framework were studied for representative environmental characterization problems. Results indicate that significant raw performance improvements are possible using the approach and that global search techniques identify high quality solutions for the characterization problems studied.

The structure of the errors associated with an inverse problem solution are a function of monitoring observations. Optimal inverse experiment design is investigated as an approach for improving solution quality. The approach involves the selection of monitoring locations that are best suited to the generation of a well-conditioned source identification inverse problem. The monitoring design problem is formulated as a non-linear combinatorial optimization problem and solved using the optimization framework developed previously. The monitoring designs generated exhibit an optimal substructure that may be exploited to develop more efficient methods of solution. An analysis is conducted to evaluate the source inversion performance of an optimized monitoring network relative to networks designed using different methods. The results of the analysis demonstrate conclusively that when the source identification problem is underdetermined the number of monitoring sensors installed in the network is more important than the method used to locate them.

Contaminant Source Monitoring and Characterization in Water Distribution Systems

by
Michael Eugene Tryby

A dissertation submitted to the Graduate Faculty of
North Carolina State University
in partial fulfillment of the
requirements for the Degree of
Doctor of Philosophy

Civil Engineering

Raleigh, North Carolina

2008

APPROVED BY:

Jeffrey A. Joines

E. Downey Brill, Jr.

John W. Baugh, Jr.

S. Ranji Ranjithan
Co-Chair of Advisory Committee

G. Mahinthakumar
Co-Chair of Advisory Committee

Dedication

This dissertation is dedicated to the memory of my grandparents Eugene Stanley and Martha Evelyn Tryby and my Uncle James Craig Hudelson who each in their own way inspired me to pursue my dreams.

Biography

The youngest of two children, Michael Eugene Tryby was born to Julia Carol and Felix Eugene Tryby in Cleveland, Ohio on November 21, 1968. After graduation from Mayfield High School, Mayfield Village, Ohio in 1987, he attended Kent State University in Kent, Ohio prior to transferring in 1990 to the University of Cincinnati School of Engineering in Cincinnati, Ohio. While studying there he participated in the cooperative education program working for the U.S. Environmental Protection Agency and Quantum Chemicals. He graduated from the University of Cincinnati in 1993 earning a B.S. in Civil Engineering with a concentration in water resources engineering.

Michael spent the lost years after graduation working odd jobs and traveling across the United States of America in a camper van. After discovering that testing laundry detergent, rough carpentry, operating ski lifts, crawling around chemical plants, and smoke testing sanitary sewers did not suit him, he returned to the University of Cincinnati for graduate school in 1995. There he studied drinking water treatment with advisors James G. Uber Ph.D. and R. Scott Summers Ph.D. focusing on booster disinfection in water distribution systems. He married Phoebe Elizabeth Acheson, a graduate student in the Department of Classics at the University of Cincinnati, in the spring of 1998. Later that year he was promoted to a Research Associate with the Department of Civil Engineering Institute for Water Treatment Optimization and ultimately completed his M.S. in Environmental Engineering in 2000.

Prior to returning to graduate school in 2002 to pursue a doctorate in Civil Engineering at North Carolina State University in Raleigh, North Carolina, he worked as a software developer and water distribution simulation domain expert at Haestad Methods in Waterbury, Connecticut. At NC State University, Michael studied with advisors Ranji Ranjithan Ph.D. and Kumar Mahinthakumar Ph.D. in the areas of environmental systems analysis and computer aided engineering. Michael became the proud father of Evelyn Claire Tryby in 2003 and Peter Hawkins Tryby in 2006. His ultimate career goal is to continue to learn and grow as he pursues a research career in that fascinating union between the disciplines of environmental engineering, environmental systems analysis, and computational science.

Acknowledgments

I would like to thank my advisors Dr. Ranjithan and Dr. Kumar for the freedom they allowed me to pursue my research interests and the patience they expended in the process. I owe both of them a debt of gratitude. I thank my committee for diplomatically coaxing me to focus my work and helping me to get on track.

I want to extend deep thanks to Dr. Baha Mirghani with whom I worked very closely on the development of the simulation optimization framework and groundwater inverse problems. Our working relationship suffered as we tried to complete our research on time while maintaining a high level of quality; for my role in that I extend a sincere apology. I also must thank Dr. Marco Propato for the many interesting conversations we shared about inverse problems and monitoring design in water distribution systems. It is hard to to find colleagues that are as genuinely interested in one's work as he was in mine, and indeed, the result was a very fruitful collaboration.

I must acknowledge the sacrifices my family has made enabling me to finish this work. My children have spent many weekends and evenings without their father and the task of watching them fell upon my spouse Phoebe. This degree has taken longer to earn than it should have and Phoebe held the family together during some difficult times. I am utterly insolvent in the currency of "husband points" and must declare bankruptcy; I hope that as we move into the post graduate school phase of our lives we can negotiate terms of repayment and she will allow me to reestablish my credit.

The camaraderie among colleagues in graduate school is something I cherish and relied quite heavily on for support. Special thanks must be extended to all the students that passed through Mann Hall Room 431 during my extended stay here. I would especially like to thank Rajasooriyar Partheepan, Dr. Jason Dorn, Pam Schooler and her husband Rubin, Dr. Pervin "Ozge" Kaplan, Dr. Xin Jin, Li Liu, Dr. Yung Jung, and Matthew Clayton for their friendship.

And lastly, I want to thank my mom and dad, my mother and father in-law, and especially my Sister Mary Jo for their love and support.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

The main thrust of this dissertation is to investigate ill-posed environmental monitoring and characterization problems in water distribution system contexts, a problem closely aligned with recent developments in water distribution system security. In particular, novel solution and monitoring design techniques are studied. The development of a generic simulation optimization framework enables the investigation of the environmental monitoring and characterization problems studied. In this introductory chapter the motivations for this work are described, research objectives guiding it are laid out, and the organization of this document are presented.

1.1 Introduction

Environmental characterization describes an important class of inverse problems that occur in the problem domains studied by environmental engineers, such as groundwater and surface water hydrology, air pollution, and urban water systems. In practice, environmental characterization is an important component of environmental monitoring, remediation, and restoration activities, as well as resource management and regulatory enforcement. This research will focuses on potable water distribution systems, where environmental characterization problems are frequently encountered in system management and operations. Until recently most research on water distribution characterization problems has centered on model calibration and water quality monitoring. In the wake of the September 11, 2001 terrorist attacks, however, new energy has been focused on security related water distribution characterization problems, primarily the detection and identification of malicious sources of contamination in a system. Other related water security research has focused on quantifying exposure and the resulting adverse health effects of such an attack and the development of frameworks for risk analysis and mitigation.

An essential aspect of all environmental characterization problems regardless of the specific problem domain in which they reside is the process of deducing information from sparse monitoring data. Problems where unknown system parameters or inputs are resolved from observational data characterizing a system's outputs are known as inverse problems. Such problems in environmental contexts frequently involve the identification of system parameters that describe chemical kinetics or transport phenomena and system inputs such as initial conditions, source location, source concentration or mass flux, and source release schedules. Environmental characterization problems can range from simple calibration problems requiring the estimation of a few system parameters to complex problems requiring the identification of many unknown system parameters and inputs.

Solution of an environmental characterization problem often requires solution of an inverse problem, as the monitoring data is sparse and contains only the signatures of the desired system information. It is reasonable to infer that the ability to solve such problems is, in part, a function of monitoring data characteristics. Indeed, it can be shown that the statistical qualities of error in an inverse problem solution are a function of monitoring data characteristics; and as such, monitoring data quality and the ability to accurately solve characterization problems are related to one another.

Monitoring data is rather difficult and expensive to collect and as a result is sparse and considered valuable. Monitoring data collection requires careful planning so that, in the end, for the purpose in which it was intended. An environmental monitoring plan is used to describe data characteristics such as the spatial and temporal distribution of samples to be taken in the monitoring domain, the analytical methods to be used for data analysis, and quality control and assurance procedures. The development of monitoring plans for environmental characterization activities is complex, requiring the consideration of goals, strategies, and methods in conjunction with an understanding of the physical, chemical, and biological variables and processes active in the monitoring domain [2]. The scope of monitoring activities and desires for coverage, precision, and accuracy are often at odds with budgetary and other practical constraints. Thus, the process of developing environmental monitoring plans is considered a difficult problem with multiple and conflicting objectives that are not easily resolved. Practitioners typically apply existing knowledge of the monitoring domain as the basis of an ad hoc process for developing monitoring plans with little emphasis on coordinating the monitoring plan with characterization analysis requirements. **One objective of this research is to develop a formal method for designing monitoring plans to improve the solution quality of environmental characterization activities in water distribution systems.**

To further illustrate how the aspects of environmental monitoring and characterization described above apply in a water distribution security problem context consider the following example. One can imagine a scenario where an unknown toxic contaminant is maliciously injected at an unknown location in a municipal water distribution system. As the source enters and is transported through

the pipe network the resulting distribution of contaminant concentrations would contain the "finger print" of the original source release (desired characteristics such as location, time, and concentration). A network of monitoring stations would continuously monitor water quality for the presence of toxic contamination. As monitoring data was collected it could be used to infer the source characteristics of the attack — a classic inverse problem. As mentioned previously, monitoring data quality and the solution of the characterization problem are related. Working with a monitoring plan, in this case a monitoring network design, should improve the quality of the characterization problem solution.

There are several approaches for solving the inverse problems arising from environmental characterization analysis. Here the focus is on an "simulation optimization" approach. The simulation optimization approach relies on a forward model, usually a system of partial differential equations, that describes the dynamic processes of the environmental system and defines the relationship between model inputs and outputs. Forward models are coupled with formal mathematical or heuristic search procedures to determine the model inputs that best approximate the observed data. The solution of inverse problems is several orders of magnitude more computationally challenging than solution of the corresponding forward model, since thousands of forward model evaluations are typically required.

In general, the solution complexity of such problems is proportional to the number of system parameters to be determined. Inverse problems are difficult to solve due to ill-posedness. Depending on the particular problem context, such as groundwater systems, solution of inverse problems in the environmental domain are particularly challenging due the characteristics of the coupled large scale non-linear PDE systems that describe the dynamic process present in environmental systems.

Gradient-based search techniques represent the state-of-the-art search procedure for solving inverse problems using the optimization approach. Gradient-based search techniques, while efficient for well formed problems, are often poorly suited for ill-poised environmental problems with multiple optima and non-linear, discontinuous, and discrete features. Further, gradient-based techniques can converge erroneously to local optima, missing the global optimal solution of the problem. One alternative is the use of global optimization techniques, such as evolutionary algorithms, which can provide a more robust search of the decision space. Yet another approach is combining global search techniques to identify favorable regions and then applying local search heuristics to fine tune the solution. Such hybrid approaches have the potential to outperform gradient based approaches for solution of these types of ill-posed inverse problems. **Another objective of this research is to investigate the application of evolutionary algorithms for the solution of environmental characterization problems.**

Coupling the optimization approach for solution of environmental characterization problems with evolutionary algorithms as proposed here makes the investigation of distributed computing on high

performance hardware a necessity. Investment in high speed networking infrastructure has allowed the aggregation of geographically distributed high-performance computing resources into what are referred to as computational grids. On top of this hardware infrastructure, computational grids are constructed from a software middle-ware which provides distributed computing, communication protocols, scheduling, security, and policy mechanisms. In part because computational grids promote reliable and economical access to and sharing of high-end computing resources, they have emerged as a new paradigm in scientific and engineering computation. The emergence of computational grids has created new possibilities for the solution of environmental characterization problems. **The final objective of this work is the development of a prototype grid-enabled simulation optimization framework suitable for the solution of environmental characterization problems.**

1.2 Research Objectives

The overall objective of this research is to investigate the solution of ill-posed environmental monitoring and characterization problems, employing novel solution techniques and monitoring design methods to improve solution quality. The proposed research has three main components as identified previously and described in greater detail in the following paragraphs.

1. *Development of an environmental monitoring and characterization problem formulation:* an environmental monitoring and characterization problem is studied and a problem formulation for its solution is developed. More specifically, the problem focuses on the identification of malicious contamination source locations in municipal water distribution networks. The problem has two parts: 1) the problem of locating monitoring sensors in the network for the accurate and robust identification of contamination sources; 2) the identification of a malicious source given the data provided by the monitoring network. The major questions addressed include, How best to locate monitoring sensors for accurate and robust identification of malicious sources? and, How best to formulate the source identification problem for timely and accurate solution?

2. *Development of a prototype grid-enabled simulation optimization framework:* a prototype search framework is developed and deployed on a computational grid. The objective is to develop a flexible framework with a modular component architecture. The framework is designed for deployment on a computational grid composed of geographically distributed super computing clusters. The framework facilitates the investigation of environmental monitoring and characterization problem formulations, search procedures, and computational performance

and efficiency. The research question addressed here is, What software design approaches and tools are best suited to the development of distributed search framework components and computational grid deployments?

3. *Investigate the use of evolutionary algorithms for solution of environmental monitoring and characterization problems:* an evolutionary algorithmic (EA) approach is investigated for the solution of the above mentioned problems. In particular, genetic algorithms and evolutionary strategies are studied. The research questions addressed include, Are EAs suitable for the solution of the monitoring sensor location problem with its combinatorial solution characteristics? and, Are EA solution approaches suitable for ill-posed environmental characterization problems like source identification in water distribution or groundwater systems?

1.3 Dissertation Organization

The chapters that follow document the investigation of each of the three proposed research objectives. Chapter 2 develops an environmental characterization problem, specifically the contaminant source identification problem in water distribution systems. The problem is described mathematically and the effect of regularization on solution quality is discussed. In Chapter 3, the design of the grid-enabled simulation optimization framework is documented. The solution framework developed is efficient and scalable, and this is demonstrated with a performance analysis for a several representative environmental characterization problems. The development of an environmental monitoring problem formulation related to water distribution system security can be found in Chapter 4. The formulation represents the monitoring objectives associated with source identification in water distribution security applications. Finally, an executive summary, conclusions, and suggestions for future work are presented in Chapter 5.

Chapter 2

Sparse Solutions for Source Identification in Water Distribution Networks

This chapter is an exposition on the water distribution source identification problem studied in this dissertation. In it, contaminant transport processes in water distribution systems are described, and the source identification problem is formulated as a discrete linear inverse problem. The difficulties associated with inverse modeling in this problem context are illustrated through an extended example. Lastly, the effect of regularization on the source identification problem is demonstrated and discussed.

Looking ahead, the monitoring requirements associated with source identification imply that the source identification and monitoring design problems are coupled. In a later chapter, the source identification problem formulated here serves as the basis of a monitoring design formulation where these requirements are expressed.

2.1 Introduction

Accidental contamination events have occurred sporadically in water distribution systems (WDS) for as long as they have been operated. Over time, multiple barriers against contamination have been engineered, most notably modern disinfection and filtration practices, and as a result the frequency and severity of accidental contamination events has decreased. In the post September 11, 2001 security environment new energy has been directed towards emergency planning and preparedness for accidental and deliberate contamination events.

Preparing for contamination events, whether accidental or deliberate, is complicated by WDSs themselves. "Water distribution networks are complex large scale engineered systems serving populations situated in large service areas. Such networks by their nature offer many potential uncontrolled entry points for contamination. Once contaminates have entered a system, transport can occur over

multiple flow pathways dictated by system hydraulics, which in turn are driven by stochastic user demands and dynamic system operations. [56]" In part, because of their large spatial extent and the complexity of their transport and water quality dynamics, and complicated by the scarcity of monitoring data in the WDS, water utilities typically do not know with a high degree of certainty the quality of the water as it is distributed and consumed in the system. Therefore, one important aspect of a utility's preparedness for a contamination event is the development of a strategy for system water quality monitoring and contaminant source identification.

This chapter is concerned with the formulation of a source identification problem for water distribution networks. If a malicious or accidental contamination event was detected a critical aspect of a utility's response would involve characterization of the source. Source characterization involves identifying the location and quantifying the release of the contaminant occurring there from the monitoring data collected after the detection of the event. Identifying a source location would prove useful as a utility took countermeasures in response to an attack such as source isolation or bleeding contaminated water from the system. Further, quickly identifying a source location could aid in the apprehension and ultimate prosecution of the party responsible in the case of a malicious attack. Source identification in WDSs, however, is a challenging technical problem.

Source identification problems of this type are classified as inverse problems as the state of the system is being inferred from the monitoring observations. Inverse problems are difficult to solve for several reasons including ill-posedness, ill-conditioning, and computational tractability. Well-posedness was first defined by Hadamard to classify problems describing physical phenomena [24]. Its converse, ill-posedness, is regularly used to describe the issues making the solution of inverse problems difficult:

- existence — the lack of a solution;
- non-uniqueness — multiple solutions;
- and, instability — sensitivity to noisy data observations.

Some ill-conditioning issues are a function of monitoring design. For example, model and measurement errors can be important factors contributing to parameter identification uncertainties. Source identification inversion problems in WDSs are likely to be under-determined, because the monitoring data is sparse and there are more potential contaminant injections than monitoring observations. When the inverse problem is linear this leads to a rank deficient problem description where there are an infinite number of potential solutions and inherent solution non-uniqueness. Thus, the monitoring design and source identification problems are coupled with one another not only because the source identification problem requires data from a sensor network for solution, but more importantly, because the structure of the source identification solution itself and the errors associated with it are a function of monitoring design. The question of how best to design monitoring network for the

purpose of source identification, however, is a topic addressed later in Chapter 4.

In the sections that follow the difficulties associated with solving source identification problems in WDSs are explained. First, numerical models for water quality transport are described. Next, an input/output (I/O) water quality model is developed and the source identification problem is formulated. Elementary discrete linear inverse theory is briefly reviewed. An extended example is used to illustrate the issues related to solution of the inverse problem. Finally, the effect of regularization on problem solution is demonstrated and discussed.

2.2 Methodology

The objective of the source identification problem is to identify the location where contamination is entering the WDS using data from a monitoring network. Thus, the intrinsic state of the system must be inferred from external observations. This type of problem is categorized as an inverse problem and is solved using various techniques dependent on problem structure. The general approach taken here consists of using linear system dynamics to describe the water quality transport behavior in WDSs, and hence, allowing the source identification problem to be described as a discrete linear inverse problem where the model parameters being identified describe the contamination source location and release history. First, however, a comprehensive mathematical description of water quality transport modeling in WDSs is developed.

2.2.1 Water Distribution Transport Modeling

EPANET is a popular public domain hydraulic and water quality solver developed by the U.S. Environmental Protection Agency for water distribution network modeling [49]. The modeling process abstracts the physical components of a network — pipes, pumps, valves, and storage facilities — into a network of links joined at nodes and configured to represent the topology of the physical system being modeled. Water distribution hydraulic dynamics are described by time varying nodal heads, reservoir volumes, and link flow rates that are simulated as a sequence of steady-state hydraulic solutions (solved using the Hybrid Node-Loop Method or alternately as the Global Gradient Algorithm [54]) related through periodic hydraulic boundary condition updates, a technique referred to as extended period simulation [47]. Water quality transport through each pipe in a water distribution system is modeled using the plug flow reactor (PFR) equation, a special case of the classical one-dimensional advection dispersion reaction equation where longitudinal dispersion is neglected [50]. The transport governing equation is a hyperbolic partial differential equation that describes the advective transport and reactions of a chemical solute in link i traveling its length $\mathcal{L}_i$ from the upstream end $x = 0$ to the downstream end $x = \mathcal{L}_i$;

$$\frac{\partial c_i(x,t)}{\partial t} + u_i(t)\frac{\partial c_i(x,t)}{\partial x} + R(c_i(x,t)) = 0, \tag{2.1}$$

$$c(x=0,t) = c_{j_i}(t), \tag{2.2}$$

$$c(x,t=0) = c_0(x), \tag{2.3}$$

$$\forall i \in \mathcal{P}.$$

Where, $c_i(x,t)$ is the concentration of the solute c in pipe i, as a function of longitudinal position x and time t, $u_i(t)$ is the time varying fluid flow velocity, and $R(c_i(x,t))$ is a general reaction term for the chemical solute. The PDE has accompanying boundary conditions describing solute concentration at position $x = 0$ as $c_{j_i}(t)$ the concentration exiting the upstream node j_i of link i, and the concentration profile in the pipe at time $t = 0$. An equation is written for each link i in the set of links $\mathcal{P}$ comprising the network.

Schemes for the numerical solution of Eqn. (2.1) can be characterized as Eulerian or Lagrangian in nature. In the Eulerian scheme fluid flow is observed from points in a fixed reference frame, while in the Lagrangian scheme the fluid is observed from a reference frame moving with the bulk fluid flow (Eqn. (2.1) is the Eulerian form of the PFR equation). Typically, the PDE is reduced to an ODE by replacing the spatial derivative with a finite difference approximation such as the Lax-Wendroff scheme [29]. Eulerian based transport algorithms, however, are known to exhibit numerical dispersion compared to Lagrangian based algorithms [23].

The method of characteristics can be applied to change the coordinates of the problem from the Eulerian (x,t) to a Lagrangian reference frame (x_0,s) where the new variable x_0 is a constant and s is a distance along a curve in the x,t plane, see Figure 2.1. In doing so, Eqn. (2.1) is converted to a system of ODEs to be solved along characteristic curves in the new coordinate system. The scalar concentration then becomes $c(x(s),t(s))$, and taking the directional derivative along s yields,

$$\frac{dc_i(x(s),t(s))}{ds} = \frac{dx}{ds}\frac{\partial c_i(x(s),t(s))}{\partial x} + \frac{dt}{ds}\frac{\partial c_i(x(s),t(s))}{\partial t}. \tag{2.4}$$

Note that the derivatives along lines in the new reference frame are ordinary. Comparing Eqns. (2.1) and (2.4), if one chooses

$$\frac{dx}{ds} = u_i(t(s)), \tag{2.5}$$

$$\frac{dt}{ds} = 1, \tag{2.6}$$

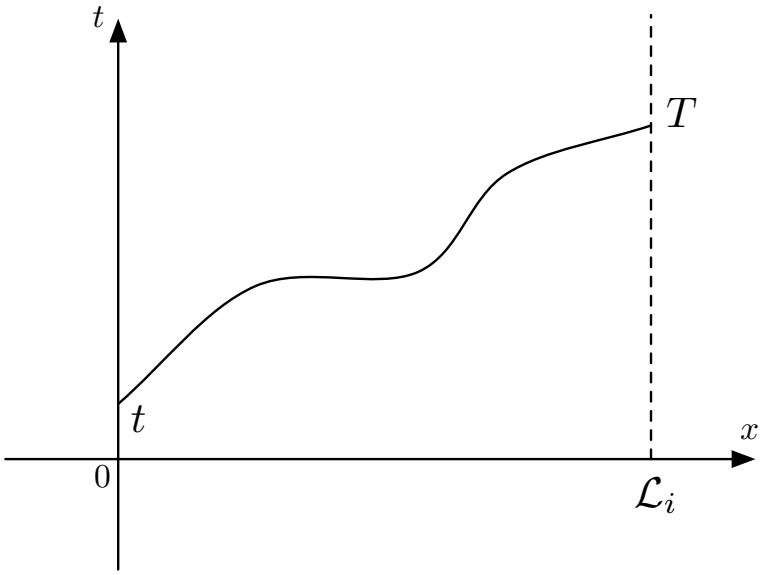

Figure 2.1: Characteristic curve for link i with flow velocity $u_i(t) > 0.0$ [14].

then along the characteristic curves Eqn. (2.1) becomes,

$$\frac{dc_i(x(s),t(s))}{ds} + R(c_i(x(s),t(s))) = 0. \tag{2.7}$$

The solution to the original PDE Eqn. (2.1) can then be obtained by first solving Eqns. (2.5) and (2.6) (known as the characteristic equations) with the initial conditions $x(0) = 0$ and $t(0) = t_0$ for transformation between (x_0, s) and (x, t). Then solving Eqn. (2.7) with boundary condition Eqns. (2.2) and (2.3) for the change in concentration due to reactions.

Integrating Eqn. (2.6) by inspection $t = s$. Furthermore, when fluid flow is positive during a period of time the reference frame is propagated along the pipe in the downstream direction at a rate equal to the fluid velocity and Eqn. (2.5) can be solved accordingly [14],

$$\int_0^{\mathcal{L}_i} dx = \int_t^T u_i(s)ds. \tag{2.8}$$

Thus, when the system hydraulic dynamics are known *a priori* (*i.e.* the flow velocities $u_i(t)$ are known), the distance along a link x can be eliminated by calculating the time when a fluid parcel enters a link t and the time it exits T. As noted the case briefly described here is for positive unsteady flow velocities Constans *et. al.* describes a method for solving for times t and T for the general case when flow reversals occur [14].

Each link in a network model is represented as a PFR and modeled with Eqns. (2.5), (2.6), and (2.7). The individual link equations are joined together with shared boundary conditions written at upstream and downstream nodes. Nodes can represent either pipe junctions or storage reservoirs, the essential difference being the volume associated with the node. Pipe junctions are assumed to be completely mixed and have no volume associated with them. Under these assumptions the time varying solute concentration exiting a junction node $c_j(t)$ can be written as,

$$c_j(t) = \frac{\sum_{i \in IN_j} q_i(t) c_i(x(t) = \mathcal{L}_i, t) + m_j(t)}{\sum_{i \in IN_j} q_i(t) + q_j(t)}, \forall j \in \mathcal{N}. \tag{2.9}$$

Where, the total mass entering the node is the sum over the set IN_j of all links i entering node j. The numerator is the mass rate entering from a link is simply the product of the link flow rate q_i and the time varying concentration exiting the link $c_i(x(t) = \mathcal{L}_i, t)$ plus $m_j(t)$ the mass rate injected directly into node j if it happens to be a source for the chemical solute. The denominator is the total flow rate entering the node where, $q_j(t)$ are unknown flows into the node associated with solute injections (typically these are assumed negligible relative to the flow rate entering the node). The junction node equation is written for each node j in the set of junction nodes $\mathcal{N}$ in the network.

Links can also have storage nodes as their boundary conditions. Storage nodes differ from junction nodes in that they can have a significant volume associated with them making the calculation of mixing phenomena and reactions within the storage volume necessary. Assuming that the storage volume is completely and instantaneously mixed the total mass in the storage volume is,

$$M_j(t) = V_j(t) c_j(t), \tag{2.10}$$

where $V_j(t)$ is the time varying volume and $c_j(t)$ is the time varying solute concentration in storage node j. Expanding the expression for $M_j(t)$ by the chain rule and accounting for reactions in the volume yields the equation for the change in the mass of solute in storage with time,

$$\frac{dM_j(t)}{dt} = V_j(t)\frac{dc_j(t)}{dt} + c_j(t)\frac{dV_j(t)}{dt} + V_j(t)R(c_j(t)). \tag{2.11}$$

Solving for $\frac{dc_j(t)}{dt}$ yields,

$$\frac{dc_j(t)}{dt} = \frac{1}{V_j(t)}\frac{dV_j(t)}{dt}(c_i(x(t) = \mathcal{L}_i, t) - c_j(t)) + m_j(t) + R(c_j(t)) \tag{2.12}$$

$$\frac{dV_j(t)}{dt} = Q_j(t) \tag{2.13}$$

$$c_j(t = 0) = 0 \tag{2.14}$$

$$\forall i \in IN_j, \forall j \in \mathcal{S}.$$

Thus, the equations for the change in solute concentration within the storage volume Eqn. (2.12) and the change in that volume itself Eqn. (2.13) are linked with one another. The differential equations for concentration and volume are written along with the accompanying initial conditions Eqn. (2.14) for all nodes j in the set of storage nodes $\mathcal{S}$.

2.2.2 Input/Output Water Quality Model

It has been shown that the response of the WDS system to a contaminant mass injection can be described using linear dynamical systems theory under the following assumptions [9]:

1. Hydraulic dynamics are known *a priori*;
2. The contaminant is conservative, or reactive in the zero-order or first-order; and
3. The observed reaction rate is independent of the dosage applied.

Previous researchers have used linear superposition to describe historical exposure reconstruction [25], and booster disinfection scheduling [9] and location [55]. The idea was developed further into a general linear input/output model by [65, 51, 46] and used as the basis of booster disinfection design, [46] automatic system calibration [65, 51], and intelligent adaptive control [59]. The general linear I/O model proposed by Zierolf and developed further by Shang differs from the one proposed by Propato in that it computes system response in reverse time, though the resulting descriptions of a system are equivalent [46].

The contaminant concentration dynamics $c_i(t)$ at node i can be expressed as the linear summation of responses to individual dosages $u_j(k)$ at node j having occurred at previous time steps using the general linear I/O model. Using the notation of [46, 45] this relationship can be more formally stated as,

$$c_i(t) = \sum_{j=1}^{n} \sum_{k=1}^{t} \theta_{ij}^{k}(t) m_j(k), \tag{2.15}$$

where, the summation extends over all potential contamination sources $j = 1, \ldots, n$ and over the discretized injection time steps $k = 1, \ldots, t$. Taking the sampling time step $\Delta t = 1$, the response coefficient $\theta_{ij}^{k}(t)$ is the concentration at node i at time t resulting from a contaminant injection occurring at node j between time $t = k$ and $t = k + 1$ with all other injections being equal to zero. The response coefficients $\theta_{ij}^{k}(t)$ take a value greater then or equal to zero for all i, j, k, t.

Equation (2.15) provides an approach for expressing the source identification problem mathematically as a linear system of equations. Considering a set of n_s monitoring sensors and monitoring

time window $[0, T]$, in compact matrix notation the linear I/O model becomes,

$$\mathbf{c} = \mathbf{A}\mathbf{m}. \tag{2.16}$$

Where in expanded form $\mathbf{c}$ is a length $n_s T$ vector of contaminant concentrations,

$$\mathbf{c} = (c_1(1)c_1(2)\dots c_1(T)c_2(1)\dots c_{n_s}(T))';$$

$\mathbf{m}$ is a length nT vector of contaminant mass injections,

$$\mathbf{m} = (m_1(1)m_1(2)\dots m_1(T)m_2(1)\dots m_n(T))';$$

and, $\mathbf{A}$ is a size $n_s T \times nT$ matrix of response coefficients,

$$\mathbf{A} = \begin{pmatrix} \theta^1_{11}(1) & 0 & \dots & 0 & \theta^1_{12}(1) & \dots & 0 \\ \theta^1_{11}(2) & \theta^2_{11}(2) & \dots & 0 & \theta^1_{12}(2) & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ \theta^1_{11}(T) & \theta^2_{11}(T) & \dots & 0 & \theta^1_{12}(T) & \dots & \theta^T_{1n}(T) \\ \theta^1_{21}(1) & 0 & \dots & 0 & \theta^1_{22}(1) & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ \theta^1_{n_s 1}(T) & \theta^2_{n_s 1}(T) & \dots & \theta^T_{n_s 1}(T) & \theta^1_{n_s 2}(T) & \dots & \theta^T_{n_s n}(T) \end{pmatrix}.$$

Given the connectivity and transport characteristics typical of water distribution networks, the response matrix $\mathbf{A}$ is very sparse. Generally, monitoring observations are relatively difficult and expensive to gather. Thus, for all practical problems $\mathbf{A}$ is a rectangular system with $n_s < n$, and therefore, the system of Eqns. (2.16) is underdetermined with the unknowns outnumbering the equations.

Given a system response matrix $\mathbf{A}$ and nodal contaminant concentrations $\mathbf{c}$, solving Eqn. (2.16) for the vector of unknown contaminant injections $\mathbf{m}$ is the essence of the source identification problem. The author has shown that the problem can be formulated using a linear system of equations, and thus can be categorized as a discrete linear inverse problem. This is convenient as discrete linear inverse problems are well understood and powerful tools exist for their analysis and solution.

2.2.3 Linear Discrete Inverse Theory

The response matrix $\mathbf{A}$ is principal to linear inverse problems as it provides the mapping $A : M \rightarrow C$ provided by the forward model Eqn. (2.16). The objective of the inverse problem is to determine $\mathbf{m}$ from observations $\mathbf{c}$ from the forward mapping A. The naive method of solving the problem is simply to solve for zero;

$$\mathbf{c} - \mathbf{Am} = 0. \tag{2.17}$$

Unfortunately, for most practical problems this solution does not exist due to measurement and modeling errors. The approach then becomes one of finding the model $\mathbf{m}$ that minimizes the misfit between the data set and the predicted values from the forward model. Typically, some measure of distance such as a vector norm is used as the misfit function,

$$\min_{\mathbf{m}} \|\mathbf{c} - \mathbf{Am}\|_p; \tag{2.18}$$

where, the vector p-norm is equal to,

$$\|\mathbf{c} - \mathbf{Am}\|_p = \left[\sum_{i=1}^{n_s T} (c_i(t) - (\mathbf{Am})_i(t))^p\right]^{1/p}. \tag{2.19}$$

The vector norm equation is valid for $p \geq 1$, though even in this form the problem is frequently ill-posed and difficult to solve. The most familiar formulation is least squares problem with $p = 2$. The L_2 norm is the Euclidean length of the misfit associated with the model. Direct solution of the the least squares problem is possible via the normal equations

$$\mathbf{m}_{L_2} = (\mathbf{A}'\mathbf{A})^{-1}\mathbf{A}'\mathbf{c}. \tag{2.20}$$

This solution is guaranteed to be the unique minimum when $(\mathbf{A}'\mathbf{A})^{-1}$ exists (*i.e.* $\mathbf{A}$ is full column rank, $rank(\mathbf{A}) = n_s T$). A matrix is said to be rank-deficient when $rank(\mathbf{A}) < n_s T$. When $\mathbf{A}$ is rank-deficient, solution of discrete linear inverse problems becomes difficult or sometimes impossible due to non-uniqueness and stability issues, and special techniques are required for their solution.

Singular Value Decomposition (SVD) is a powerful numerical technique for analyzing conditioning and resolution in discrete linear inverse problems. SVD is a factoring method, analogous to eigenvalue analysis for non-square matrices, applied to the forward model matrix $\mathbf{A}$ which is decomposed to,

$$\mathbf{A} = \mathbf{USV}' \tag{2.21}$$

where $\mathbf{U}$, size $nT \times nT$, and $\mathbf{V}$, size $n_sT \times n_sT$, are orthogonal matrices spanning the date space and model space respectively, and $\mathbf{S}$, size $nT \times n_sT$, is the matrix of singular values.

SVD can be used to form the Moorse-Penrose pseudo-inverse which has useful properties when solving rank deficient problems. Applying the pseudo-inverse the solution of the least-squares problem becomes:

$$\mathbf{m}^\dagger = \mathbf{A}^\dagger \mathbf{c}, \tag{2.22}$$

where $\mathbf{A}^\dagger = \mathbf{V_r S_r^{-1} U_r}'$. The pseudo-inverse $\mathbf{A}^\dagger$ always exists, and thus, the solution $\mathbf{m}^\dagger$ also always exists, unlike the inverse associated with the normal equations $\mathbf{A}'\mathbf{A}^{-1}$ which can become singular and non-invertible when $\mathbf{A}$ is rank deficient.

The pseudo-inverse also exhibits useful properties related to solution stability. When the singular values of $\mathbf{S}$ are small an inverse solution becomes sensitive to noise and round-off errors. The matrix $\mathbf{S}$ is a diagonal matrix with positive elements ordered from greatest to least:

$$s_1 \geq s_2 \geq \ldots \geq s_{min(_r)} \geq 0. \tag{2.23}$$

In general, the singular value elements of $\mathbf{S}$ will decay and can eventually equal zero. The r^{th} element of the diagonal is the last non-zero singular value. The pseudo-inverse enhances solution stability by using only the r largest singular values to form the inverse. This is often referred to as singular value truncation and is a simple form of "regularization" — a technique for stabilizing and making a solution unique. Singular value truncation, however, can degrade the level of detail present in the recovered model, *i.e.* the model resolution.

Solution non-uniqueness is related to the null-space of the system matrix $NS(\mathbf{A})$. Partitioning the SVD representation of $\mathbf{A}$ into non-zero and zero sub-matrices and expanding yields;

$$\mathbf{A} = [\mathbf{U}_r, \mathbf{U}_0] \begin{bmatrix} \mathbf{S}_r & 0 \\ 0 & 0 \end{bmatrix} [\mathbf{V}_r, \mathbf{V}_0]', \tag{2.24}$$

where $\mathbf{U}_p$ and $\mathbf{U}_0$ denote the first p and the last $nT - r$ columns of $\mathbf{U}$ respectively, and $\mathbf{V}_r$ and $\mathbf{V}_0$ denote the first p and the last $n_sT - p$ columns of $\mathbf{V}$ respectively. In compact notation the SVD expansion becomes $\mathbf{A} = \mathbf{U}_r \mathbf{S}_r \mathbf{V}_r{}'$. Thus, the matrix $\mathbf{A}$ carries no information about the portions of $\mathbf{U}$ and $\mathbf{V}$ that are multiplied by zero singular values (*i.e.* $\mathbf{U}_0$ and $\mathbf{V}_0$). Furthermore, the rank of $\mathbf{A}$ is equal to r and the sub-matrices $\mathbf{U}_0$ and $\mathbf{V}_0$ can be shown to form the orthonormal basis for the data null space $NS(\mathbf{A}')$ and the model null space $NS(\mathbf{A})$ respectively [3].

The model null space of the system matrix $\mathbf{A}$ is defined as the set of $n_sT - r$ column vectors $\mathbf{v_0}$ such that $\mathbf{A v_0} = 0$, expressed in set notation the definition becomes $NS(\mathbf{A}) = \{\mathbf{v_0} \in \Re^{n_sT} | \mathbf{A v_0} = 0\}$. Correspondingly, each of the orthogonal columns of $\mathbf{V_0}$ satisfies the definition. Assuming that

a vector $\mathbf{m}^*$ satisfies Eqn. (2.16) it is referred to as a particular solution. The complete solution containing all vectors satisfying Eqn. (2.16) can be expressed as a particular solution plus any vector from the null space as follows:

$$\begin{cases} \mathbf{m} = \mathbf{m}^* + \mathbf{V_0 q} \\ \mathbf{q} = \{\mathbf{q} \in \Re^{n_s T} | \mathbf{m} > 0\} \end{cases} \tag{2.25}$$

It can easily be shown that this relationship is true for the projection of any arbitrary $\mathbf{q}$, size $n_s T - r$, into the null space $\mathbf{V_0}$. When solving an underdetermined system, such as Eqn. (2.16), once a particular solution has been identified it is possible to formulate the complete solution that contains an infinite set of particular solutions that satisfy the system of equations.

2.3 Example

In this section, an example WDS source identification and a resolution and stability analysis is performed using the Example 3 network that comes as part of the EPANET software distribution. The network model has been modified to simulate a malicious contamination scenario to demonstrate the solution of the source identification inverse problem using classical discrete linear inverse theory. The network is modest in size, containing two sources, three tanks, 97 nodes, and 116 links, see Figure 2.2.

An extended period simulation of the hydraulics and water quality transport was performed and contaminant concentrations were gathered at 12 monitoring locations distributed throughout the network. Sensor locations are identified in Figure 2.2. The sensor locations were located using an *ad hoc* process involving random choice and engineering judgement. The monitoring sensors gathered data on a 10 minute sampling interval. The total length of the simulation was 8 hours, from 12:00 AM until 8:00 AM. Thus, a total of 576 data points were recorded and used to formulate the source identification problem. Over the same observation interval, all of the 97 nodes in the network were regarded as potential sources. The injections at potential sources were discretized into 10 minute injection intervals. Thus, there are 4656 (6x8x97) variables describing the potential contaminant injections occurring in the network over the 8 hour observation window.

The sparse matrix density of $\mathbf{A}$ for the monitoring design under consideration is 3.63 percent with approximately 576 and 2524 non-zero rows and columns respectively (see Figure 2.3). The forward model matrix density and sparsity pattern are a function of the sensor locations and sampling frequency of the monitoring sensor network. Consider that the rows included in the matrix are a subset of the full linear I/0 model description of the system. Taken together the sensor locations and the sampling frequency of a particular monitoring design determine the filtered view of the system

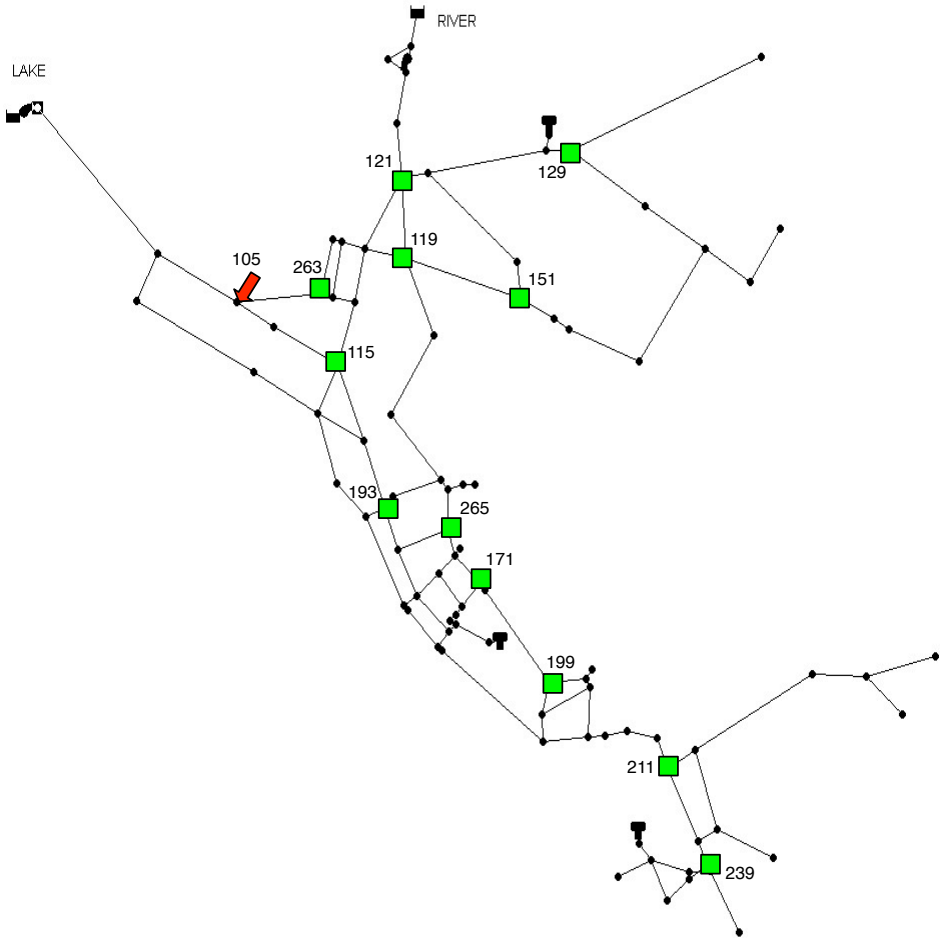

Figure 2.2: Example 3 network with monitoring sensor locations indicated with squares and the true source at arrow.

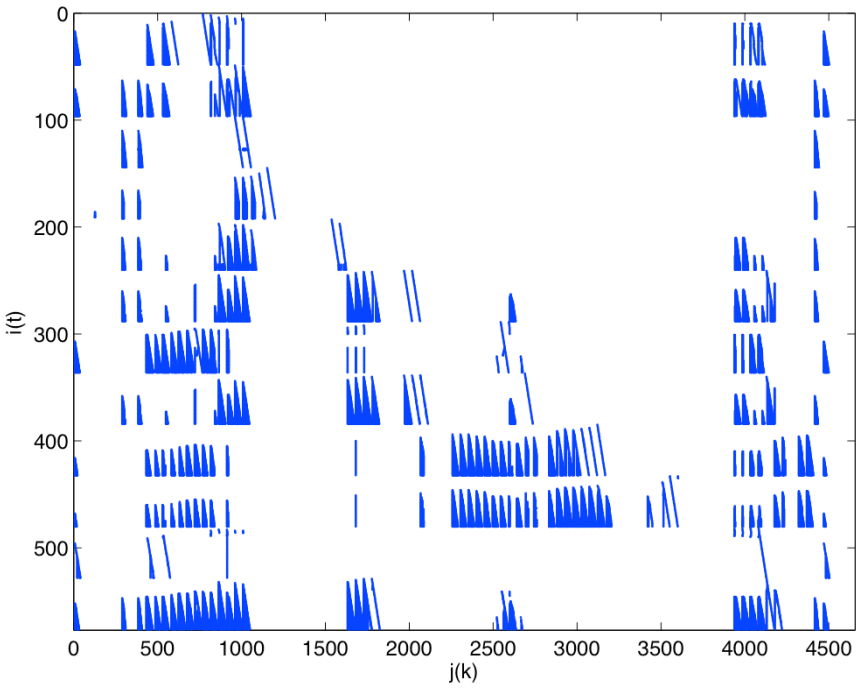

Figure 2.3: Sparsity pattern for forward model matrix $\mathbf{A}$, where $j(k)$ is the index for the elements of mass vector $\mathbf{m}$ and $i(t)$ is the index for elements of the concentration vector $\mathbf{c}$.

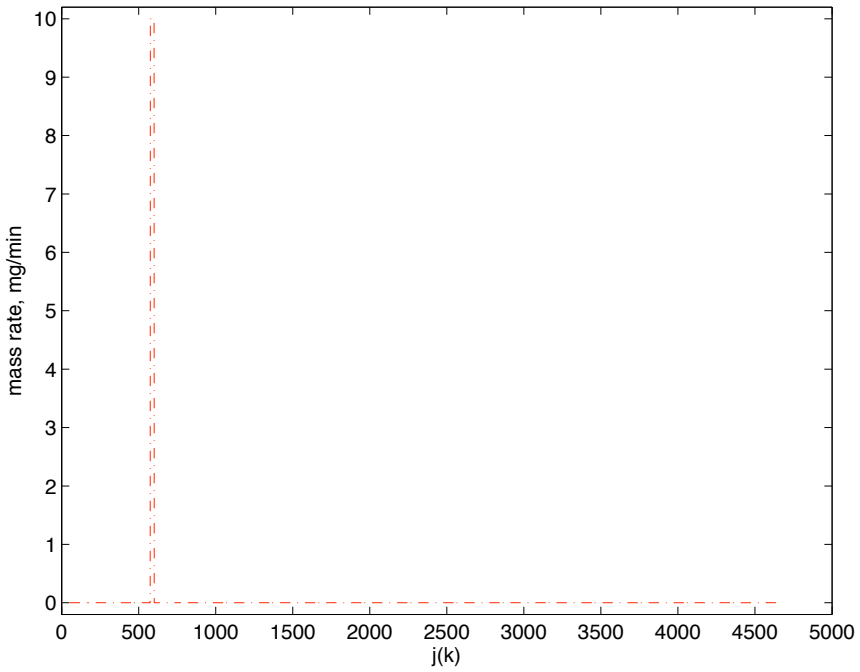

Figure 2.4: True vector of contaminant mass injections applied to the system, where $j(k)$ is the index for the elements of mass vector $\mathbf{m}$.

response that is carried in the forward model matrix. Rows containing only zero elements correspond to a monitoring location that does not register a response from any of the potential sources in the system, though this is rarely the case[1]. Columns containing only zero elements correspond to dependent variables (a source location and injection interval) that have gone undetected by the sensor network. These dependent variables will not be included in a particular solution to the problem, though as will be explained shortly, they reside in the null space of $\mathbf{A}$ and therefore may be a part of the problem's general solution. The sparseness pattern of $\mathbf{A}$ can be analyzed in conjunction with the vector of monitored concentrations $\mathbf{c}$ to eliminate rows and columns from the problem and thus reduce the size and the computational expense of its solution as described in [45].

The true source was a non-reactive contaminant injected into the system at node 105 at 12:00 AM for a duration of four hours (node 105 is indicated by the red arrow in Figure 2.2). The Figure 2.4 shows the true mass injection input $\mathbf{m}^*$ applied to the system. The profile displayed in the figure corresponds to all 4656 dependent variables describing all possible inputs into the system, with the inputs occurring at each node concatenated together. The true model (shown in red) applied to the

[1]Assuming perfect sensors and a constant contaminant injection, this is unlikely because a source can occur at any node and the sensors occupy a subset of the nodes; thus, a sensor should be able to detect at least one source as long as it is not located at a zero flow dead end node.

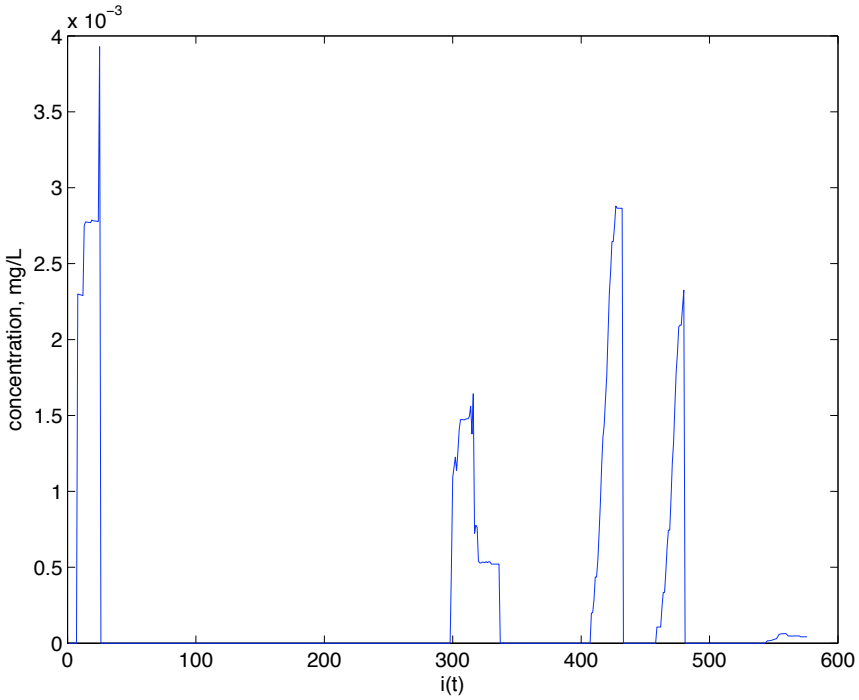

Figure 2.5: True vector of contaminant concentrations monitored in the system, where $i(t)$ is the index for elements of the concentration vector $\mathbf{c}$.

system is essentially a unit spike input occurring over a four hour time duration (twenty-four 10 minute injection intervals). A true zero residual solution does exist and can be identified as unique assuming the sensors and model are perfect, like in this synthetic example, and the null space is trivial. This is no longer true, however, even when very small measurement and/or modeling errors are present or when the problem is underdetermined.

The true system response is shown in Figure 2.5. Again the profile displayed in the figure corresponds to the concentration time series observed at each node concatenated together. For this example, the vector of true contaminant concentrations $\mathbf{c}^*$ was generated synthetically by solving the forward problem $\mathbf{c}^* = \mathbf{A}\mathbf{m}^*$. The system response indicates that most of the monitoring sensors in the network did not detect the presence of the contaminant.

A SVD analysis of the forward model matrix $\mathbf{A}$ was performed. The singular values are shown in Figure 2.6. All of the 576 singular values $\mathbf{A}$ are significantly greater than zero, hence A is full row rank. Theoretically, this is the greatest rank value possible, but it does not necessarily lead to the best possible problem conditioning. Despite being full row rank, the problem is significantly underdetermined and the inverse solution is likely to be non-unique and therefore difficult to interpret.

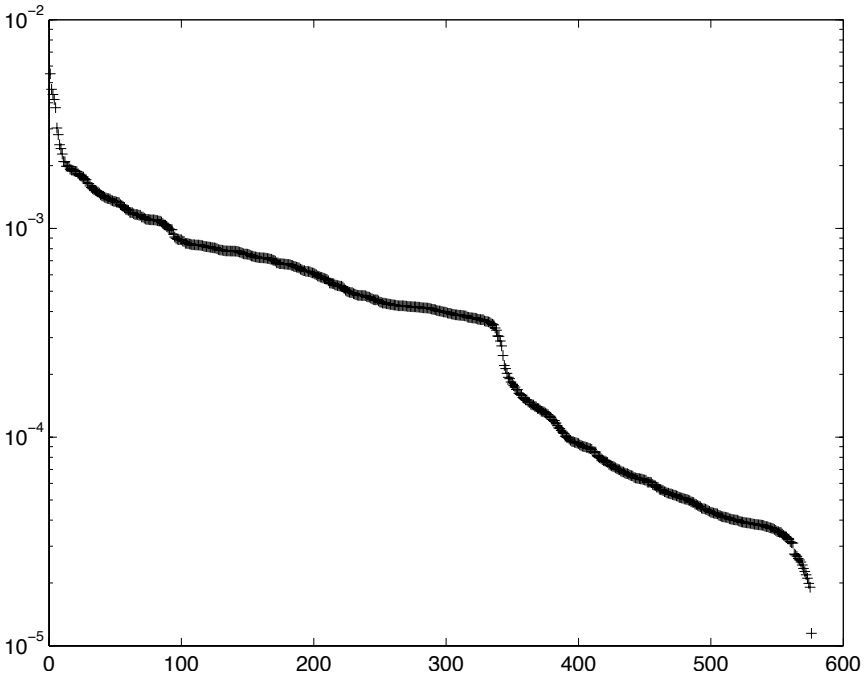

Figure 2.6: Singular value spectrum for forward model matrix, **A**.

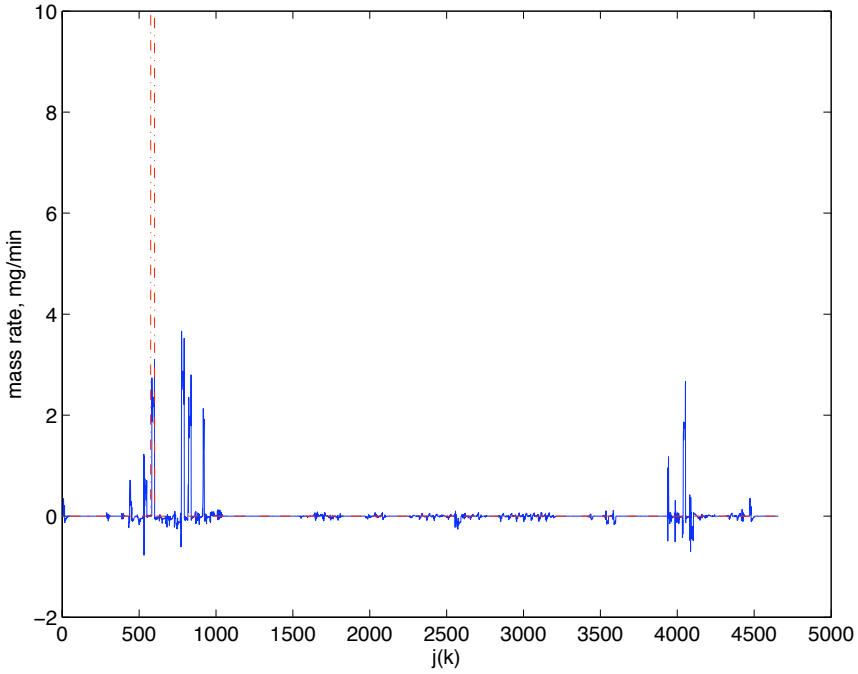

Figure 2.7: Model recovered by linear least squares, where $j(k)$ is the index for the elements of mass vector **m**.

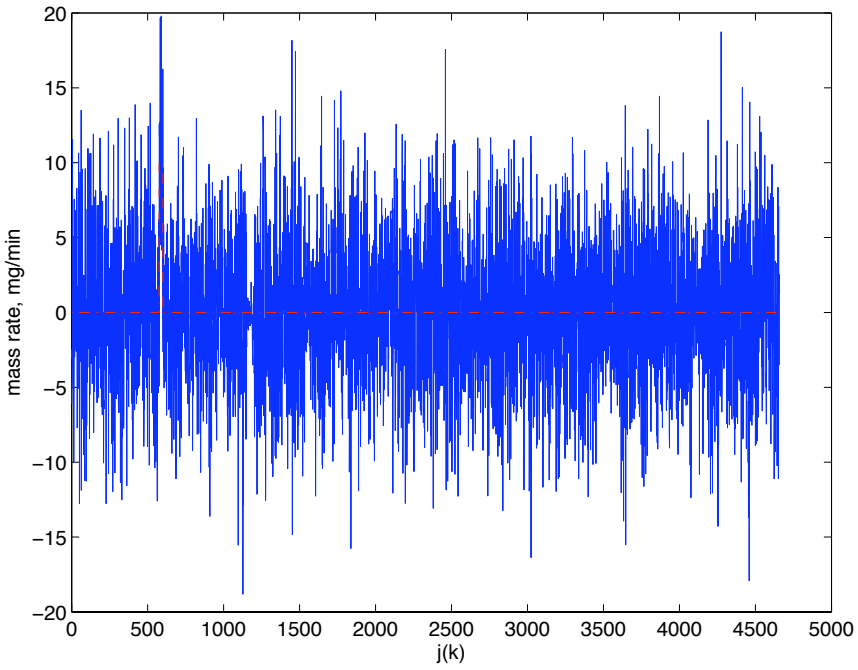

Figure 2.8: An instance of the generalized solution, where $j(k)$ is the index for the elements of mass vector $\mathbf{m}$.

The inverse solution is shown in Figure 2.7 and was generated using the Moorse-Penrose pseudo-inverse. This solution is stable; however, comparison with the true model inputs indicates it was poorly recovered. In our source identification problem context, the model recovered involves contaminant injections distributed both spatially and temporally throughout the distribution system. Intuitively, it is unlikely that an accidental or malicious contamination event would actually occur with a pattern of contamination injections such as this. Furthermore, the negative injections occurring in the solution are physically meaningless.

In the course of SVD analysis the orthonormal basis for the model null spaces are calculated. It is straightforward to demonstrate the significance of a non-trivial null space with respect to the problems solution. Figure 2.8 illustrates one instance of the generalized solution, $\mathbf{m_0} = \mathbf{m}^* + \mathbf{V_0 q}$. The vector $\mathbf{q}$ length $n_s T - p$ was randomly generated with a normal distribution, N(0, 1). The system response to the input $\mathbf{m_0}$ was calculated by solving the forward problem $\mathbf{c_0} = \mathbf{A m_0}$ and is displayed in Figure 2.9. Visual inspection of Figures 2.5 and 2.9 reveals that they are identical. This is significant for two reasons: 1) an infinite number of general solutions to the problem exists, and 2) the monitoring network is unable to detect contaminant injections occurring at many locations distributed throughout the network. It is important to note that the model null space is a property

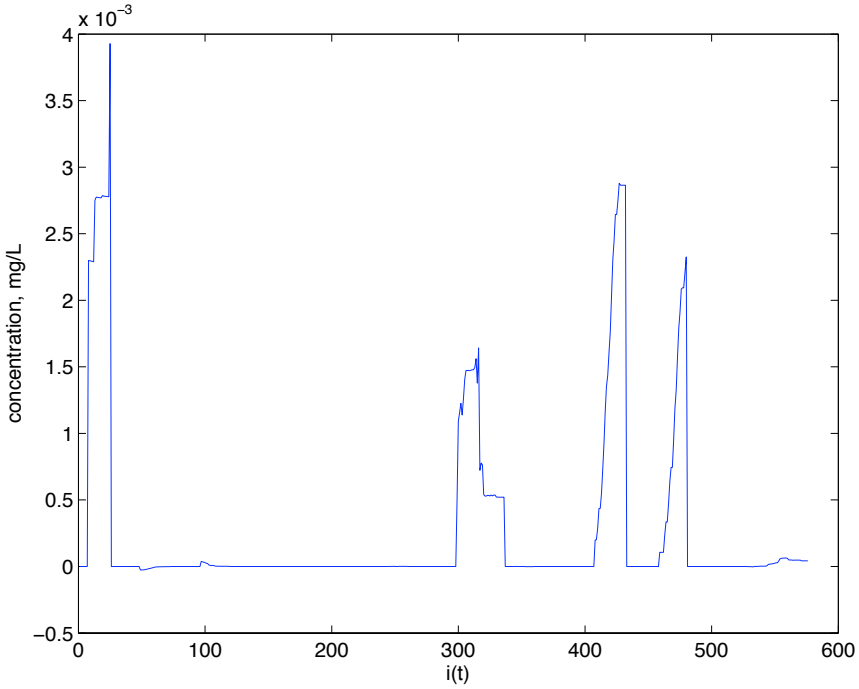

Figure 2.9: System response to the generalized solution, where $i(t)$ is the index for elements of the concentration vector $\mathbf{c}$.

of forward model matrix $\mathbf{A}$ and the mapping it defines; and further, it is a consequence of the problem being underdetermined.

2.4 Regularization

Regularization is a technique for stabilizing the solutions of ill-posed inverse problems. Frequently, the application of regularization results in an approximate solution of the original problem by incorporating prior information [58]. Prior information may include known values for the parameters being estimated, upper and lower bounds on the parameter values, or the expectation of a solution being of minimum length, smooth, or sparse (these concepts will be explained in greater detail below). In this section, two types of regularization are discussed, Tikhonov regularization, which has previously been applied to the WDS source characterization problem, and a method called Basis Pursuit that has recently emerged as a regularization technique for solving underdetermined systems of linear equations.

2.4.1 Tikhonov Regularization

Regularization techniques are commonly used to compensate for solution instability or to select a unique particular solution from the complete solution for a system of underdetermined linear equations. The most common technique is Zeroth-order Tikhonov regularization [53]. Zeroth-order Tikhonov regularization as well as its higher order forms incorporate the expectation of a smooth solution (prior information) into the problem formulation. In its simplest form a regularization term is added to a standard least squares problem to yield a damped least squares formulation,

$$\min_m \|\mathbf{A}\mathbf{m} - \mathbf{c}\|_2 + \alpha\|\mathbf{m}\|_2; \tag{2.26}$$

Employing Tikhonov regularization the resulting damped least squares problem can be formulated as a linear least squares problem with an augmented coefficient matrix,

$$\min_m \left\| \begin{bmatrix} \mathbf{A} \\ \alpha\mathbf{I} \end{bmatrix} \mathbf{m} - \begin{bmatrix} \mathbf{c} \\ 0 \end{bmatrix} \right\|_2^2. \tag{2.27}$$

where α is a regularization parameter and the solution norm added to the prediction error objective is the "length" of the solution vector. The standard least squares problem that results can be solved directly via the method of normal equations and SVD factorization [3].

Informally, the least squares term can be thought of as ensuring existence, and the term minimizing the solution norm ensures uniqueness [58]. The rationale for adding the solution norm for

regularization is to promote smoothing, especially when the problem contains errors in either $\mathbf{A}$ and / or $\mathbf{c}$, by discouraging potential solutions from over fitting the data. In the WDS source identification problem context minimizing the solution norm tends to select approximate solutions that minimize the mass of contaminant entering the system from potential sources.

The approximate solution selected by the formulation now becomes a function of the parameter α. When α is small the solution minimizes the prediction error. Conversely, when α is large emphasis is placed in minimizing the solution norm. By carefully selecting α an approximate solution can be selected which minimizes both prediction error and the solution norm. Several methods have been developed for selecting an optimal value for α. The L-curve method is a graphical technique where the prediction error and solution norm are plotted against each other, and thus, the tradeoff between the objectives of the damped problem is visualized. The α value on the L-curve that minimizes both the prediction error and the solution norm is presumed to be the optimal value.

2.4.2 Tikhonov Regularization Example

Tikhonov regularization does produce a unique approximate solution to the original problem. This solution, however, may not necessarily be the true solution of the original problem as the following example will illustrate. The example was prepared by solving the source identification problem from Section 2.3 using Tikhonov regularization and solved directly by the normal equations based on a SVD factorization. The problem was solved for a range of α values to illustrate the sensitivity of the solution to the regularization parameter. Note that the non-negativity constraint on the decision variables was removed to more clearly illustrate the effect of Tikhonov regularization on the solution. The results are illustrated in the composite Figure 2.10.

Subplot three is the L-curve generated for the example. Inspecting the L-curve, as α increases the residual norm increases and the solution norm stays constant and then decreases as $\|\mathbf{Am}-\mathbf{c}\| \rightarrow \|\mathbf{c}\|$. This is somewhat unusual, as the L-curve is transposed from its expected orientation, implying that in this problem context the residual and solution norm objectives are antagonistic, and thus a single α value that minimizes both the residual and solution norms does not exist. Subplots one, two, and four illustrate changes in the solution vector as α increases. The solution in subplot one is the least squares solution with no regularization (compare with least squares solution from previous example). As α increases in subplots 2 and 4 the solution vector is driven closer to the origin as $\|\mathbf{m}\| \rightarrow 0$. Furthermore, as α increases the solution vector becomes progressively more non-sparse as more variables take nonzero values. One explanation for this behavior is that as α increases and more emphasis is placed on minimizing the solution norm, more injection terms are activated in the solution to satisfy the residual norm more efficiently. Thus, Tikhonov regularization has the unintended side effect of selecting non-sparse solutions making the identification of contamination

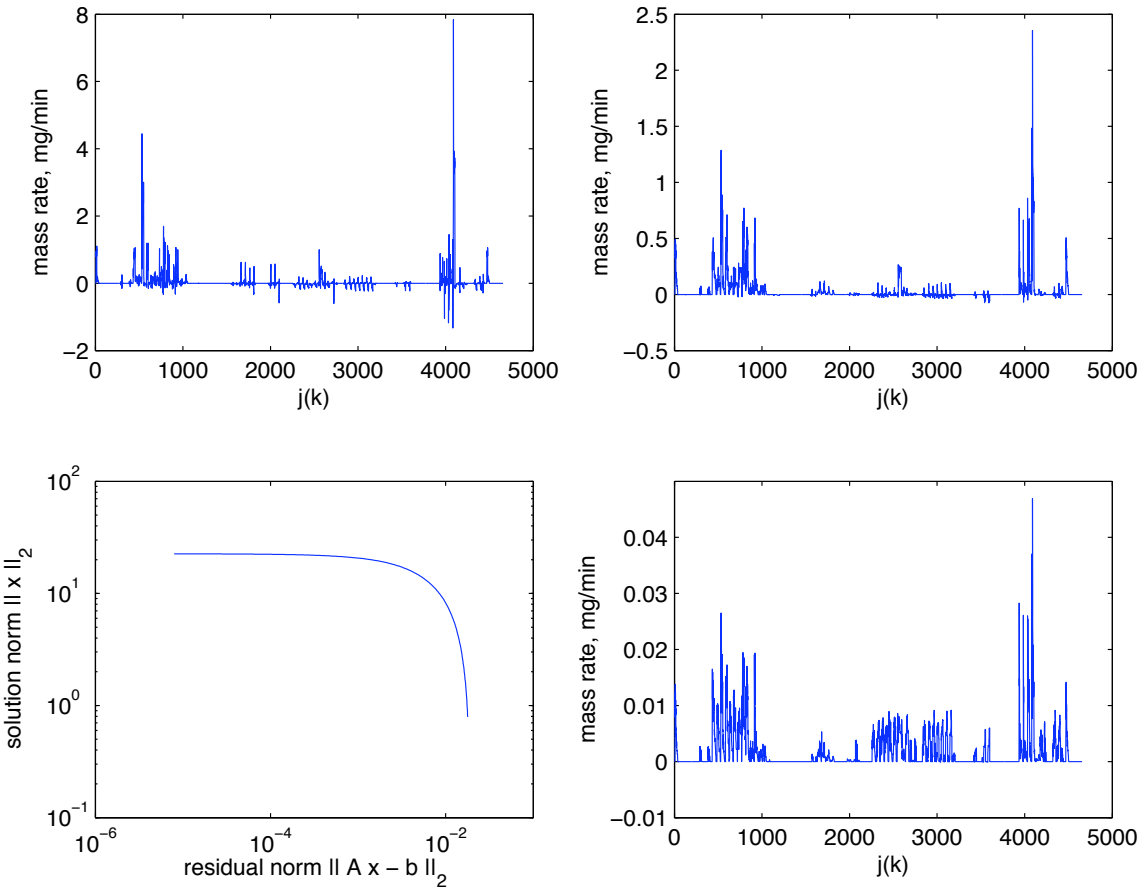

Figure 2.10: Tikhonov regularized solutions and l-curve, illustrating progressive non-sparse solutions. Sub-plots are numbered left to right, top to bottom.

sources inadvertently more difficult.

Aster *et al.* describes the appropriate conditions for applying Tikhonov regularization this way, "The accuracy of a Tikhonov regularized solution depends ... on the smoothness of the model ... If the true model is not smooth, then Tikhonov regularization will simply not give an accurate solution. [3]" Rather than erroneously seek a smooth minimum norm solution the author argues that a sparse solution to the underdetermined system should be sought. Consider for example a malicious contamination scenario. Contamination is likely to enter the system from a *few* or at most *several* source locations simultaneously in a coordinated attack scenario. The majority of injection terms in the source identification model therefore are zero and not involved, *i.e.* a sparse solution is anticipated. This is far more plausible than a solution that hypothesizes injections occurring at many different locations simultaneously. Put another way, applying the principle of Occam's Razor the simplest solution — involving the fewest non-zero coefficients — is the most likely to have generated the observed data [40]. Preference for a sparse solution can be thought of as *a priori* information added to the problem for the purpose of regularization. Thus, a regularization method which selects for sparse solutions would be more meaningful in this problem context.

2.4.3 Basis Pursuit

Basis pursuit is one method among several recently developed for identifying the sparse solution of systems of linear equations problem frequently occurs in the areas of statistics, image reconstruction, and signal processing (in particular wavelet analysis), and has applications in linear inverse theory. The idea behind basis pursuit, as applied in our problem context, is the representation of the sampled data with an "optimal" superposition of possible contaminant injections (columns of the forward model matrix). In this case, optimal means the smallest L_0 norm of the model vector $\mathbf{m}$ among all combinations that satisfactorily represent the data. Thus, the sparsest feasible solution of the linear system of equations is sought. Formally the basis pursuit problem is stated as,

$$\min_{m} \|\mathbf{m}\|_0 \tag{2.28}$$

$$s.t.$$

$$\mathbf{Am} = \mathbf{c}; \mathbf{m} \geq 0. \tag{2.29}$$

Where the objective is the L_0 solution norm subject to equality constraints written for the data observations and to non-negativity constraints on the solution vector. Due to the combinatorial nature of the objective function, solution of the basis pursuit problem has been shown to be NP-hard [40].

Employing a method referred to as convex relaxation, an L_1 norm is substituted for the objective in Eqn. (2.28) as follows,

$$\min_m \|\mathbf{m}\|_1 \tag{2.30}$$

$$s.t.$$

$$\mathbf{Am} = \mathbf{c}; \mathbf{m} \geq 0. \tag{2.31}$$

A convex objective is substituted for the combinatorial objective in Eqn. (2.28) that makes solution intractable. Convex relaxation depends on an equivalence between the L_0 and L_1 formulations of the problem which has been proven theoretically by Donoho and Elad, though specific assumptions are made regarding the forward model matrix as part of the proof [20, 21]. The two problem formulations are identical except for the objective functions; however, the problems are quite different mathematically as a result of this change allowing for different solution strategies to be employed. Donoho *et al.* show that sparse solution to Eqn. (2.30) can be found by linear programming when the true solution is sufficiently sparse [19]. The proof they develop is based on the theory of convex polytopes.

2.4.4 Basis Pursuit Example

Basis pursuit can identify a sparse solution to the linear system of equations as this example will demonstrate. These solutions, however, are not guaranteed to be unique as they are when employing Tikhonov regularization[2]. This example was prepared by solving the source identification problem from Section 2.3 using a primal dual interior-point algorithm for linear programming. Algorithms research has made interior-point methods very efficient for solving large-scale optimization problems like those that occur when seeking the sparse solution via basis pursuit. Results are shown on Figure 2.11.

Inspection of the figure reveals that the true solution was recovered. This is an interesting result, but how can LP produce such a result? Chen *et al.* explored the equivalence between linear programming and basis pursuit in their 1998 paper [13], a brief summary of which is provided here. The essential question in solving the basis pursuit problem is determining which elements of $\mathbf{m}$ are non-zero and which are zero. The non-zero elements are associated with the columns in $\mathbf{A}$ that make up the basis. The LP solution identifies the optimal basis for representing the observed data $\mathbf{c}$. Thus, Chen argues LP is essentially a basis pursuit process. Further, the basis selected dictates

[2]The non-uniqueness of sparse solutions generated by Basis Pursuit is demonstrated as part of the sensitivity analysis found in Section A.2.

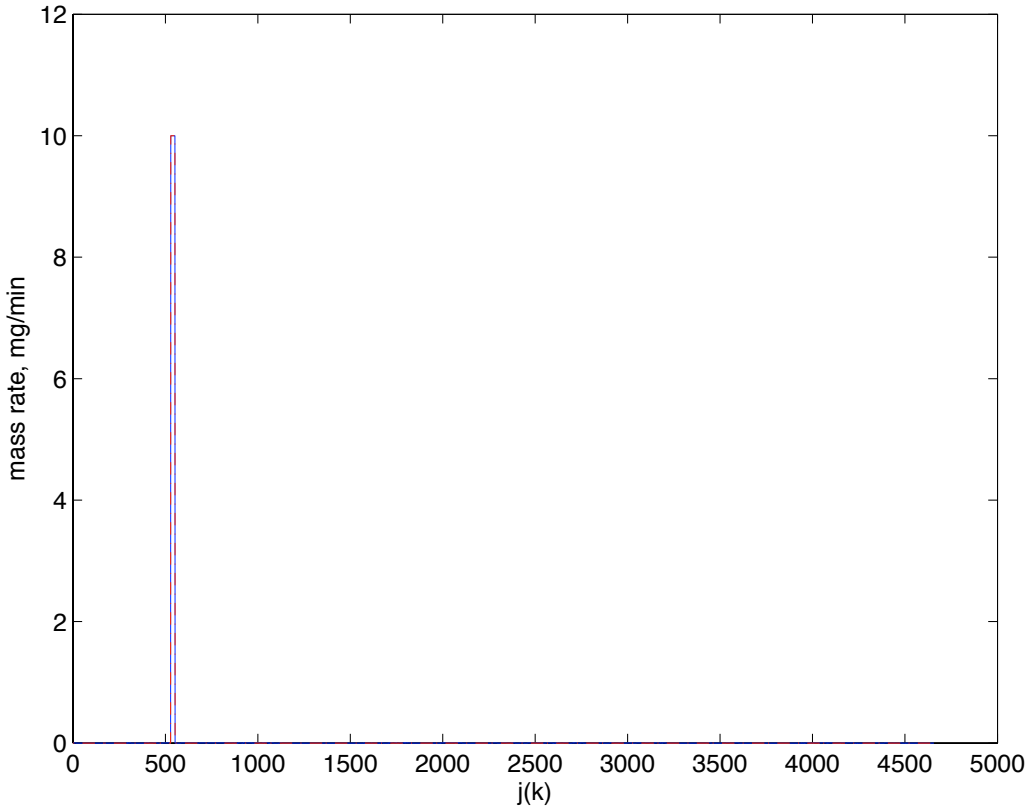

Figure 2.11: Sparse solution obtained by linear programming.

the solution and is not known *a priori*, rather it depends on the data supplied to the problem.

The simplex algorithm, a method for solving linear programming problems, is interpreted by Chen *et al.* from the perspective of basis pursuit offering further insights. The simplex algorithm is named after a "simplex" — a polyhedron in R^{nT} that defines the feasible region in decision space. Once a feasible solution is identified the algorithm iteratively improves the solution by exchanging columns that offer the greatest improvement in the objective function. A column exchange is the equivalent of a hop between vertices of the simplex polyhedron. Because the objective function is convex, there always exists a column exchange that improves the solution, except at the optimum solution. The simplex algorithm iteratively improves the basis until the optimal basis is identified. Thus, Chen concludes that the simplex algorithm is an explicit process of basis pursuit.

Basis pursuit and convex relaxation were originally developed for the solution of wavelet decomposition problems which differs from the source identification problem context in several ways. The purpose of wavelet decomposition is to optimally encode a signal using the atoms of a wavelet dictionary — a collection of parameterized wave forms. Wavelet dictionaries are designed *a priori* for this intended purpose, and thus, they are well suited to the solution of the basis pursuit problem. Typically, decomposition occurs using an overcomplete dictionary — a dictionary where the resulting signal encodings are unique. The decomposition problem is unique when the columns of the dictionary are linearly independent from one another and more strictly when they are orthogonal. Thus, an important measure of dictionary quality is coherence — a measure of the correlation between atom pairs. Dictionary encodings with low correlation between atom pairs are preferred. The best dictionary design would have orthonormal atom pairs, implying no correlation between atoms, though this requirement can be relaxed significantly. The forward model matrix associated with the source identification problem is incoherent with many columns highly correlated with one another. Incoherence makes the solution of the basis pursuit problem in our problem context more difficult. It may be possible to employ monitoring design techniques to better condition the basis pursuit problem for sparse solution. This, however, is left as a promising topic for future research.

2.5 Conclusions

The solution of source identification problems in water distribution systems is an important aspect of preparedness for accidental and deliberate contamination events. Source identification problems are more generally classified as inverse problems — a problem where system state is known and the parameters describing the system including boundary and initial conditions are unknowns. In this chapter the difficulties associated with solving source identification inverse problems in water distribution systems were discussed. The main contributions of this work were to propose the

linear I/O model as a mathematical framework in which to formulate the source identification problem, and the application of discrete linear inverse theory to understand the issues surrounding the solution of rank-deficient problems. Source identification in water distribution systems have not been extensively studied and the effects of regularization on the resulting solutions is poorly understood; this work suggests that regularizing for sparse solutions is more appropriate for water distribution source identification problems.

Contaminant transport in water distribution systems is governed by the one dimensional advection dispersion reaction equation. The equation describes longitudinal movement along the interconnected pipe segments of which the distribution network is composed. When the contaminant is conservative or reactive in the first-order the equations describing reaction dynamics are linear in concentration. Higher order reactions result in non-linear reaction dynamics.

When assuming linear reaction dynamics contaminant, transport in water distribution systems is governed by a linear homogeneous system of differential and algebraic equations. This system of equations can be written in state space form and solved analytically by integration. Transport within each hydraulic time step of a water quality simulation is governed by linear integral equation referred to as the Fredholm integral equation of the first kind. A surprising number of inverse problems are mathematically related in that they can be written in this form. Integration yields a discrete linear system of equations which can be solved using linear algebra. Thus, theoretically the source identification problem in water distribution systems fits squarely within the well understood theory for solution of linear discrete inverse problems.

The resulting linear system of equations is a discrete input / output water quality model. The dimensionality of the system of equations is governed by the number of potential source and monitoring nodes in the system and the discretization time step. Due to technological and economic constraints the number of potential source injections is greater than the number of monitoring observations and the linear system of equations is in general underdetermined. The source identification problem can be solved directly utilizing methods for underdetermined systems of equations, such as linear least squares where the misfit between the observed data and model predictions is minimized. One consequence of working with an underdetermined system is the existence of a non-trivial null space. In practical terms, the null space is the set of potential injections occurring in the network that are not observable by monitoring locations and therefore not characterizable. An extended example was developed illustrating the use of direct methods for the solution of the WDS source identification problem.

Regularization is a technique for stabilizing the solution of ill-posed inverse problems by introducing *a priori* information about the anticipated solution. Zero-order Tikhonov regularization is

the most commonly used of such methods. Essentially, Tikhonov regularization introduces a new objective into the least squares problem — one that seeks to minimize the length of the solution vector — and the damping parameter controls the tradeoff among objectives. An example was used to illustrate an unintended consequence of the inclusion of this second objective, namely, that Tikhonov regularization tends to select for non-sparse solutions making the identification of contamination sources inadvertently more difficult.

Basis pursuit is one of several methods recently developed for the selection of sparse solutions of underdetermined systems of linear equations. A sparse solution is one where the objective is to minimize L_0 norm of the feasible solutions — minimize the number of non-zero elements in the solution vector. Applying convex relaxation the problem is reformulated such that the L_1 norm of the solution vector is minimized and the problem becomes a linear program. An example was prepared and the basis pursuit linear program was able to successfully identify the solution to the source identification problem.

Finally, the application of discrete linear inverse theory to the source identification problem offers a quantitative framework for studying solution existence, uniqueness, stability, and resolution as they relate to the monitoring sensor network design problem.

Chapter 3

A Solution Framework for Environmental Characterization Problems

In this chapter, a generic simulation optimization framework is developed and representative environmental characterization applications are performed[1]*. The solution approach taken here couples environmental simulation models with global search methods and requires high-performance computer resources for tractability. Results for computationally challenging groundwater source identification and release history reconstruction problems are presented*[2]*. Computational performance studies were conducted to better understand the speed-up characteristics associated with various framework configurations. Significant raw computational performance improvements were observed while deploying the framework on the TeraGrid.*

3.1 Introduction

Investment in high speed networking infrastructure has allowed the aggregation of geographically distributed high-performance computing resources into what are referred to as computational grids. On top of this hardware infrastructure, computational grids are constructed from a software middleware which provides distributed computing, communication protocols, scheduling, security, and policy mechanisms. In part, because computational grids promote reliable and economical access to, and sharing of, high-end computing resources, they have emerged as a new paradigm in scientific and engineering computation. The emergence of computational grids has created new possibilities

[1] *This work was conducted in close collaboration with Baha Mirghani on the framework development and subsequent groundwater applications.*

[2] *The reader may be asking, Why are groundwater applications being performed in a dissertation about water distribution systems? The reason is simply that the computational intensity of the water distribution transport model was not sufficient to exercise the capabilities of the solution framework developed herein.*

for the solution of environmental characterization problems. Given the complexity of computational grids, however, the development of grid-enabled applications is a non-trivial task [1].

In this chapter, the development of LArge Scale Simulation Optimization (LASSO), a generic framework for the solution inverse and combinatorial optimization problems, is discussed. An essential aspect of environmental characterization is the process of resolving system characteristics from sparse observational data. Problems of this nature are categorized as inverse problems. There are many approaches for solving inverse problems. This work investigates a "simulation optimization" approach.

In general, the solution complexity of inverse problems is proportional to the number of system parameters to be determined. Ill-posedness makes inverse problems difficult to solve in anyone of three ways; 1) non-existence is the lack of a solution, 2) non-uniqueness is the existence of multiple solutions, and 3) instability is a sensitivity to noisy data observations. Solving an inverse problem is several orders of magnitude more computationally intensive then solving the corresponding forward model, since thousands of forward model evaluations are typically required for solution. Furthermore, solving these problems in the environmental domain is particularly challenging due to the characteristics of the coupled large scale non-linear PDE systems that describe the dynamic processes commonly present in environmental systems.

The ill-posed nature of environmental inverse problems, combined with the computational requirements inherent in performing many thousand forward model evaluations, often renders the application of optimization to the solution of inverse problems intractable, limiting current solution practice to trial and error or oversimplification of the problem's complexities [44]. Given the computational resource demands of the optimization solution approach, grids have the potential to facilitate the solution of environmental inverse problems that previously would not have been possible.

In the remaining sections of this chapter the framework is described and applied to an environmental characterization problem. First, papers in the scientific literature most related to this work are briefly reviewed. Next, the elements composing the simulation optimization framework are described and an analytical performance model for the framework is derived. Next, the framework is applied to a computationally challenging groundwater source identification and a source history reconstruction problem. Finally, the solution and computational performance observed in these applications is reported.

3.2 Related Work

While researchers have been actively developing strategies for adapting many different types of applications to computational grids, the focus here is on grid-enabled engineering design and simulation optimization applications.

A simulation optimization framework documented by Parashar *et al.* [43] and Matossian *et al.* [36] applies grid-enabling technologies for solving oil reservoir management problems using a data-driven approach. The dynamic, data-driven systems (DDDAS) framework consists of several autonomic components that discover and interact with one another on a peer to peer basis to solve the optimization problem. The framework is composed of a parallel oil reservoir simulation model (IPARS) and an optimization service which provides heuristic search algorithms. The autonomic components are built on a communications substrate which facilitates their interactions. A service-oriented middleware (STORM) is utilized for distributed data querying and parameter setting, and an autonomic grid middle-ware (DISCOVER) uses the Globus toolkit and CORBA commodity grid for service composition, execution, and collaboration.

Another related project is a grid architecture for engineering optimization and design search documented by Cox *et al.* [16] and Xue *et al.* [63]. The researchers developed a grid-enabled simulation optimization framework using open standard communications technologies and have demonstrated applications in computational fluid dynamics. The framework creates grid services by wrapping existing components in web service containers. Their solution framework consists of four main components. The first is the application portal, which allows engineers to interact with the solution framework. The second component is the application service provider, which allows access to the design and analysis tools. The third is an optimization service provider, which provides different search algorithms. And finally, the forth component is a computation provider, which executes simulation models and returns objective function values. Condor is used for computational resources management and was also offered as a generic web service. They observe that simulation optimization frameworks require the integration of varied computational and data resources. Their use of web service containers combined with XML and SOAP based messaging allows differences between system environments and soft programming languages to be overcome.

More recently, research has focused beyond grid-enabling optimization frameworks and onto matching grid topologies with meta-heuristic search algorithms to improve computational performance. In one such project, Lim *et al.* [30] develop a distributed population genetic algorithm (GA) framework using grid-enabling technologies. The connectivity typical of computational grids is well suited to island model distributed GAs. The researchers develop a framework for hierarchical distributed genetic algorithms using the Globus toolkit. Upon practical application of their framework,

however, the design was modified to centralize the populations and distribute only their evaluation, thus improving performance on grids composed of heterogeneous computing resources.

The essential difference between the engineering optimization and simulation optimization applications reviewed above appears to be the communications substrate used to facilitate communications between services running on distributed grid resources. This choice is critical, as it dictates the interactions between framework components, effects the complexity of the framework design and implementation, and ultimately affects application performance. Further, when designing grid applications, design choices that appear practical and advantageous may not always turn out to be. Here a pragmatic approach for grid-enabling the application is adopted and a simple framework is developed.

3.3 Framework Architecture

Simulation optimization is a general term used to describe a family of optimization techniques which utilize simulation models for the evaluation of objective and constraint functions, or gradients. A wide variety of applications lend themselves to simulation optimization techniques, including engineering design optimization, optimization of stochastic systems, model calibration, and solution of inverse problems. In this section of the chapter the design and runtime behavior of the simulation optimization framework is described.

LASSO is a framework for parallel simulation optimization. The framework task flow is illustrated in Figure 3.1 and consists of a centralized optimization application that utilizes a master worker task distribution strategy. The optimization, master, and worker processes are executed on grid based computational resources. Despite its name, the TeraGrid is operated as a static set of computational resources. Co-scheduling a job — scheduling jobs at more than one site simultaneously — required human to human communication between the user and system administrator and advance scheduling. At the time this work was performed, the software stack present on all TeraGrid sites was not sufficient to run Globus services. Thus, staging and scheduling the application was performed manually.

The LASSO master process and workers are launched by the cluster scheduler, and the TeraGrid utilizes the PBS scheduler at all locations, using Message Passing Interface CH2 (MPICH2) `mpiexec` to launch the framework on the cluster nodes allocated. The worker processes interface with instances of the forward model for distributed task execution. Results are returned to the master for processing by the centralized optimization application.

The master, worker, and task pool implementations used in the framework were originally designed and written by Dr. John Baugh as part of Vitri [5], though they were heavily modified for the

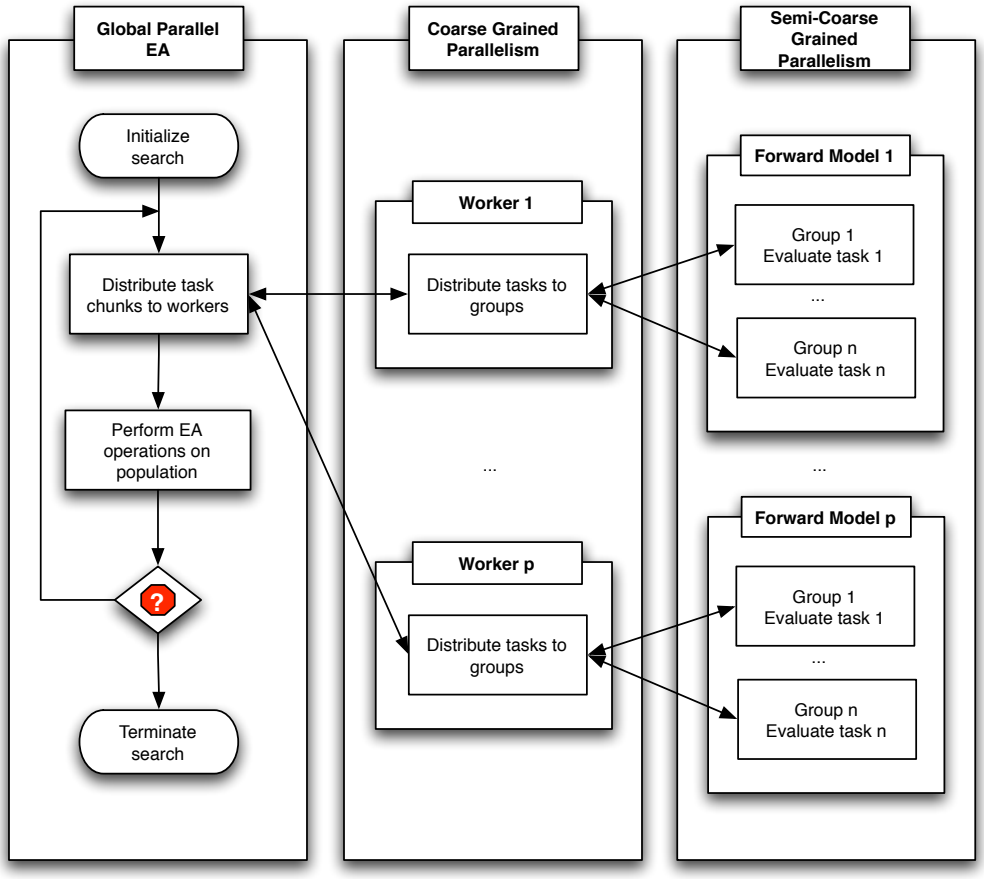

Figure 3.1: Diagrammatic representation of task flow in LASSO framework.

purposes of this work. The master maintains a pool of remote tasks — a chunk of individuals requiring evaluation. Aggregating individuals in this manner reduces communications overhead. Worker processes running on distributed grid resources, having established a TCP-IP socket connection with the master, signal their readiness and draw tasks from the task pool. The worker process transfers the remote task to the MPI zeroth processor of the forward model. From this point forward, standard MPI group communications are utilized.

The solution procedure adopted here involves two levels of parallel granularity exhibited by the search procedure and the forward model. Each iteration of the search procedure exhibits a coarse grained parallel structure that requires an uncoupled forward model evaluation for each individual in the population. The optimization application acts as the master process in the master worker task distribution strategy. The forward model manages multiple groups at the semi-coarse grained level, each group evaluates an individual in the task chunk. Within each forward model group at the fine grained level, the three dimensional groundwater simulation domain is discretized using finite element methods. The MPI library is used to group processors into computational domains and for fine grained message passing within each of these groups. A more detailed explanation of parallelism within and implementation of the forward model can be found in [35]. The results of these simulations are then aggregated into a result chunk and returned to the optimization application for processing by the search algorithm. Finally, the next generation of the search is initiated.

In the following sections of the chapter key components of the application architecture are described in greater detail.

3.3.1 Search Algorithm

Several different search procedures have been implemented in Java, making up the centralized optimization application. Evolutionary algorithms (EA) are stochastic population based search techniques that mimic Darwinian evolution to identify solutions for difficult non-linear and ill posed search problems. The optimization model representation of the inverse problems solved in this chapter were solved using an evolutionary strategy (ES) [4] and genetic algorithm (GA) [22].

A GA encodes a potential solution to the problem within an individual and starts with a collection of individuals, referred to as a population. The objective function of the optimization model is used to quantify an individual's fitness as a potential solution to the inverse problem. The forward model is executed for the calculation of each individual's fitness value. During the search process, the population is iteratively subjected to stochastic search operators analogous to natural selection, mating, and mutation. Each iteration of the algorithm constitutes a generation. This search process continues until some predefined convergence criteria is satisfied.

Evolutionary strategies were first developed by Rechenberg in 1963 [48]. Evolutionary strategies

are conceptually similar to GAs, but differs in the mechanics of some of the operators. The ES search was first applied to real encoded solution representations and uses mutation and selection as its primary exploration operators. Mutation, the most important operator in an ES search, is performed by randomly perturbing the decision vector elements by a normally distributed random value. The mutation strength is equal to the standard deviation of the normal distribution and is determined by any number of dynamic self-adaptation heuristics. Selection is carried out deterministically based on fitness rankings. The recombination operator plays a less significant role in ESs than in GAs, and traditionally, is eliminated altogether or serves as a repair operator.

The application of EAs to inverse problems is advantageous because of their robustness and global search characteristics. Several drawbacks, however, include the computational intensity of a typical EA search and slow final convergence prior to termination. Furthermore, no guarantee of optimality can be made regarding the most fit individual in the population at algorithm termination.

Evolutionary algorithms are naturally parallel, as the evaluation of populations and application of evolutionary operations can be performed concurrently. Research results indicate that parallelizing EAs results in decreased run times and improved numerical performance. Furthermore, some of these performance benefits can be attributed to spatially distributing or structuring the EA population. For example, an "island model" structures a single population into a set of interconnected sub-populations that is typically represented as a graph. A "grid model" (not to be confused with computational grids) embeds each individual into a graph, and evolutionary operations only occur between individuals adjacent to one another. In each example, the characteristics of the interconnection graph affect characteristics of the algorithms numerical performance. A panmictic EA (an EA which employs no population structuring) is the most straightforward to parallelize. This approach, referred to as global parallelism, decreases algorithm run times, but leaves the numerical performance improvements associated with population structuring unrealized. Population structuring, however, does not require parallelization as the associated benefits are realizable using serial implementations. Thus the population structuring model and the parallel implementation are largely decoupled.

The approach taken here is a panmictic population and global distributed population scheme. Maintaining a panmictic population is advantageous, because it simplifies the implementation of the EA. For example, centralizing the EA allows the same random number generator to be used throughout the algorithm. This means that the stream of random numbers generated is solely a function of the random seed. This would not necessarily be the case in a distributed EA where the population is running on heterogeneous computational resources. One ramification of this is that the results in a distributed EA may not be deterministically repeatable. This makes interpretation of the results more complicated. There are several potential solutions to this problem, but they all lead to more complicated implementations of the EA. Structuring the population also implies more

complicated population data structures, support for communications between sub-populations, and a host of other issues that arise when dealing with distributed computing applications.

3.3.2 Application Parallelism

Simulation optimization techniques typically require many forward model evaluations. Each forward model evaluation can be thought of as a task. The dependencies and sequencing of tasks depends on the type of optimization technique employed. A data decomposition analysis yields a straightforward algorithm decomposition. Essentially, the technique is data-parallel since each task is performing identical operations on different sets of input data.

The tasks are generated dynamically, as they are not known *a priori*. The tasks are uniform requiring roughly the same amount of time for execution. The individual tasks themselves are uncoupled (no data transfers are required between concurrently running tasks); however, depending on the type of optimization algorithm (for example, a synchronized generational GA) dependencies can be introduced which require all tasks to be completed before the search algorithm can continue. Different strategies can be applied to decrease the impact of these dependencies on performance.

Task mapping is the process of assigning a task to a process (or processor) for execution and is an important factor affecting overall efficiency of a parallel algorithm. Mapping can be complex because multiple competing objectives must be reconciled simultaneously. Mapping objectives include: maximizing concurrency, minimizing total completion time, and minimizing task interactions.

Mapping directly influences total execution time (the sum of time on task, time interacting with other tasks, and time idling). Mapping can be either static or dynamic depending on whether the mapping is performed a priori or during the execution of the algorithm. The structure and organization of simulation optimization algorithms generally require dynamic mapping. Dynamic mapping schemes can be either centralized or distributed. Dynamic mapping tasks to grids is complex due to the heterogeneous and dynamic nature of computational resource availability and due to communication latency issues. If one accesses the grid as a static set of resources, however, the complexity of the parallel algorithm model can be greatly reduced.

A straightforward approach for managing simulation optimization on grids is to centralize the search algorithm and distribute the tasks it generates. The tasks can be distributed using a task pool mapping strategy. Task pools dynamically map tasks to processes and can be centralized or distributed (the current implementation is a centralized task pool). The task pool itself is a data structure that stores references to tasks and distributes them on a first in first out (FIFO) basis. The task pool is managed by a master process. Worker processes connect to the master and request a task when they are available; when the task is completed the worker returns the result. One benefit of the task pool strategy is the implicit load balancing behavior of the approach. Workers contact

the task pool when they are available to perform work. On a grid with heterogeneous computational resources, workers running on faster machines contact the task pool more frequently and workers on slower machines less frequently. This is exactly the load balancing behavior desired, yet it is achieved very simply and without the complication of an explicit scheduling process.

Task pools perform best when the data, and therefore the communications overhead, associated with a task is small compared to computation associated with the task. The granularity, size of the data and the computation associated with a task, can be adjusted using chunking. In this case, a chunk would be several individual tasks bundled together. Sending more tasks in a chunk would increase the ratio of computation to communication associated with each chunk. One drawback of chunking, however, is that large chunk sizes can cause load imbalances on heterogeneous collections of machines.

3.3.3 TeraGrid Infrastructure

Supercomputing grids are composed of geographically distributed super computing resources connected via a high performance networking infrastructure. In general, the computing resources available on a grid are heterogeneous. One can safely assume that a grid constructed of distributed cluster resources behaves according to a partitioned memory address space model. Therefore, a message passing rather than a multi-threading paradigm is more suited to the grid testbed considered here. The complex topology of grids makes the design and development of distributed applications challenging.

The computational experiments presented here were performed on the National Science Foundation (NSF) TeraGrid — a distributed computing infrastructure providing peta scale data collection, analysis, and storage; and tera scale computational capabilities. The TeraGrid is a heterogeneous agglomeration of computational resources distributed across the United States and connected through a high speed network. A TeraGrid network schematic is shown in Figure 3.2.

Catlett et al. provide a detailed description of the Teragrid computing infrastructure which is summarized in the following paragraphs [12]. The Distributed Terascale Facility (DTF) was funded in 2001 anchored by four locations, NCSA, SDAC, UC/ANL, and Caltech. In 2002 the decision was made to combine the existing TeraScale Computing System (TCS-1) at PSC with the DTF, and thus the TeraGrid was born. ORNL, PU, IU, and TACC, as of September 2003, expanded the TeraGrid to 9 sites. In late 2005 Caltech decided to no longer participate in the Teragrid partnership.

The TeraGrid has a specialized interconnection network referred to as the "backplane" designed for high-band width data transfer. Computer room networking design guidelines were applied rather than a load averaging approach typical of general purpose communications networks. Thus, the TeraGrid backplane is designed for peak loads and can better accommodate data bursts between

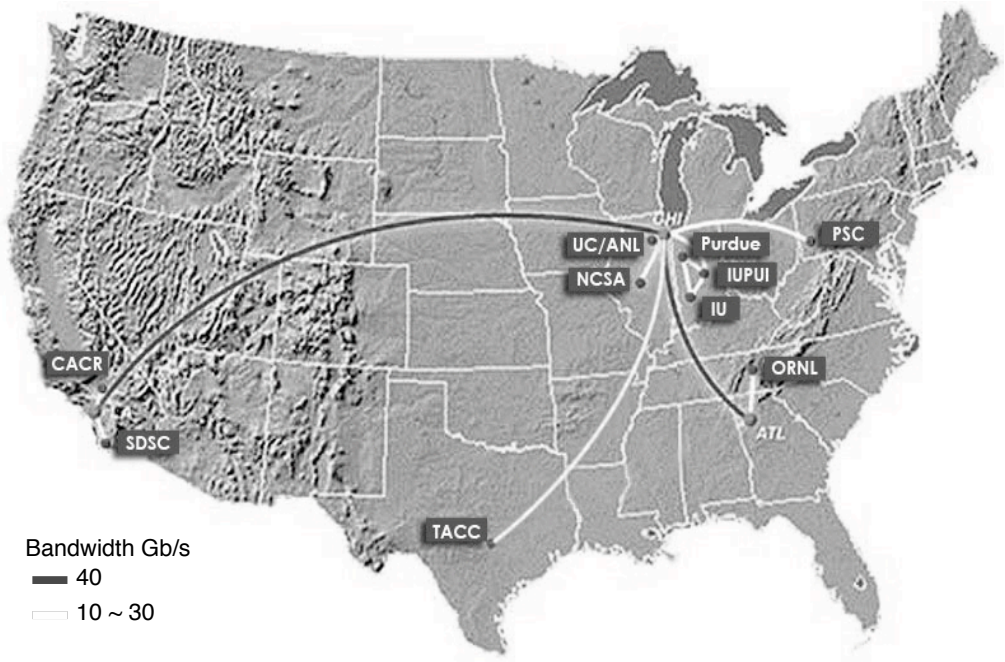

Figure 3.2: Map of TeraGrid Sites and ETF Network Segments, Note that CACR is no longer affiliated with TeraGrid, The Teragrid Project

Table 3.1: Summary of TeraGrid cluster resources for computational tasks.

Site	*CPU Type (Intel)*	*CPU Speed (GHz)*	*Nodes (num)*	*CPUs (num)*	*Performance (TFlops)*
NCSA	Itanium2	1.3, 1.5	256, 631	512, 1262	5.2, 6.0
SDSC	Itanium2	1.5	256	512	3.1
UC/ANL	Itanium2	1.3, 1.5	62	124	0.6
IU	Itanium2	1.3	8	32	0.166
ORNL	Xeon	3.06	28	56	-
TACC	Xeon	3.06, 3.20	384, 128	768, 256	5.2

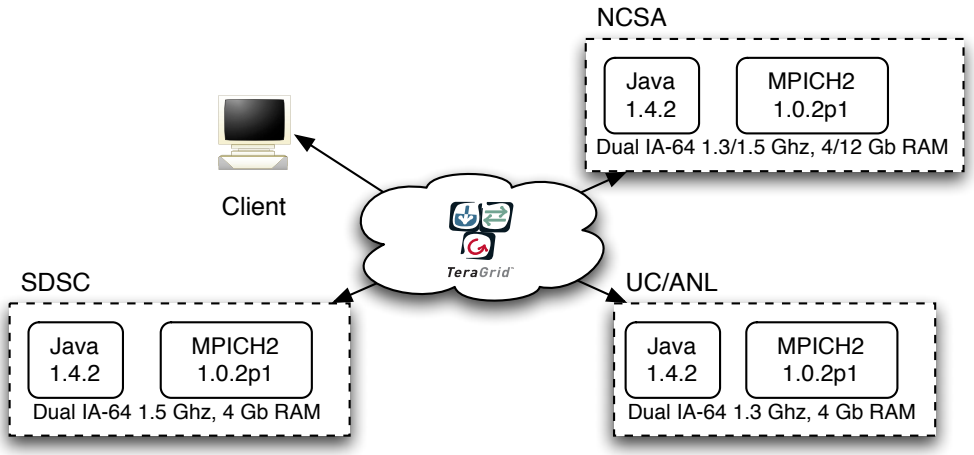

Figure 3.3: The TeraGrid testbed and software stack used for the computational experiments.

sites. A backbone, hub, and spoke network topology was adopted to accommodate future expansion easily. Backplane hubs, located in Chicago, Los Angles, and Atlanta, are connected with a 40 Gb/s dedicated fiber network. The anchor sites, NCSA, SDSC, UC/ANL, and PSC, are connected to the nearest hub with 30 Gb/s spokes. The sites added later are connected to nearest hub with 10 Gb/s spokes.

The TeraGrid sites themselves are a heterogeneous mixture of computational resources, consisting of cluster, SMP, and SP nodes. A summary of Linux cluster resources dedicated to computationally intensive tasks available on the TeraGrid is shown in Table 3.1. The four original sites composing the DTF are the core of the Linux cluster resources available on the TeraGrid. The DTF was designed with the objective of cross-site homogeneity, and thus the sites are composed of identical microprocessor technology. IU has a small site compatible with the core DTF configuration. TACC offers substantial resources based on different, but compatible microprocessor technology. Specifically, this study was conducted on the three original Distributed Terascale Facility (DTF) sites which anchor

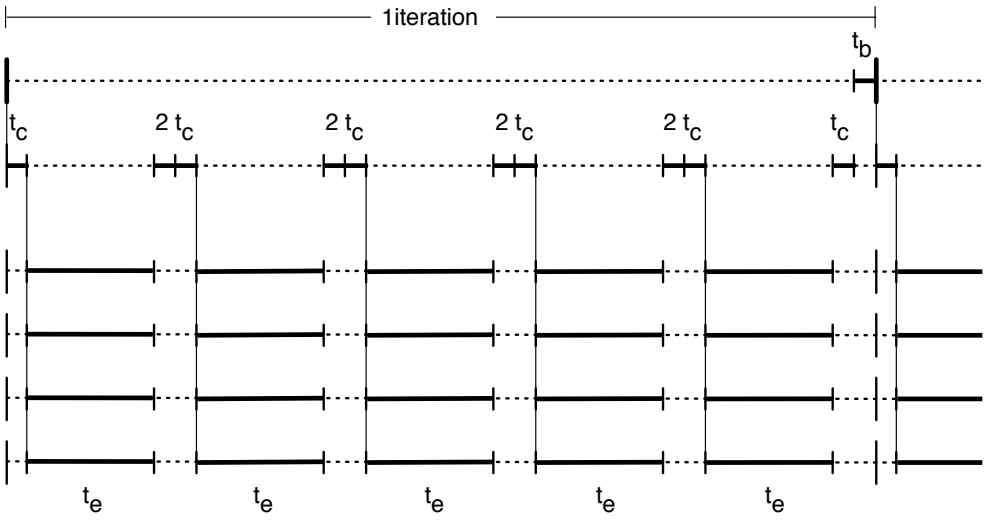

Figure 3.4: A schematic of the execution of a threaded global parallel GA, where t_b is the time spent performing search algorithm calculations, t_c is communication time, and t_e is forward model execution time.

the TeraGrid; NCSA, SDSC, and UC/ANL as shown in Figure 3.3.

3.4 Theoretical Performance Analysis

In this section of the chapter a theoretical performance analysis is documented. The analysis was conducted to better understand the performance characteristics of the framework and to help structure the computational experiments presented later in the chapter. This analysis is related to similar analyses for parallel GAs with different task decompositions that have been developed by [10, 62] and [30].

This analysis centers on the multi-threaded client and worker processes shown in Figure 3.4. The figure shows a timeline of steps occurring in the main and pool threads running in the client process and in each of the worker processes. One iteration of a typical search process is shown as it is the basic unit of execution that repeats until the search is terminated. From top to bottom the main and task pool threads in the master process appear followed by four worker processes, each running on a separate machine and having one thread of execution. During application startup the worker processes start, initialize, and set up a socket connection with the master process. Each time a socket connection is established between a worker and the master the master starts a new task pool thread. Thus, the master maintains a dedicated thread for communicating with each worker via a

shared socket connection.

The task pool works on a simple producer / consumer model, where by a remote worker communicates its readiness and its corresponding pool thread sends a task chunk. Sending a task chunk uses time t_c. The pool threads then wait for results to be returned by each respective worker. Each task chunk may contain one or several tasks for remote execution depending on the task chunk size selected *a priori*. Next, the tasks are executed asynchronously and concurrently by the worker processes using a time t_e. When task execution is complete a worker processes sends a result chunk back to its corresponding pool thread again using time t_c. Thus, it is assumed that the communication time sending task and receiving result chunks is equal. This process continues until all of the task chunks in the pool have been evaluated. Then the search algorithm performs its calculations preparing for the next iteration using a time t_b.

3.4.1 Speed-up Model

When the master and worker processes are operating on a homogeneous cluster of resources like that found at a single-site of the TeraGrid, one can reasonably assume that only minor variations in the observed t_c and t_e will occur. Modifying the notation of Xu [62] for our purposes, the total time for one iteration T_S can be expressed as;

$$T_S = T_B + T_C + T_E. \tag{3.1}$$

Where, T_S is the total time for the serial algorithm and is composed of T_B the time associated with the search algorithm calculation, T_C the communication time, and T_E the forward model execution time.

The main application thread is idle while task chunks remain in the pool, and then the search algorithm performs its calculations at the end of the iteration. For population based search procedures such as EAs, the time required to perform these calculations is generally a function of the population size n. Thus, T_B can be expressed as

$$T_B = t_b(n). \tag{3.2}$$

The main application thread idles while the pool thread is sending task chunks and while receiving result chunks. Assuming that the time, t_c, to send or receive are equal the total communication time for one iteration is the number of communication events times the average observed communication time,

$$T_C = \frac{2v}{p}\overline{t_c}. \tag{3.3}$$

Where, υ is the number of task chunks sent and the number of result chunks received, and S is the number of worker processes.

The main application thread also idles while task chunks are executed on remote workers. The total execution time during one iteration is equal to the number of tasks processed by each worker times the average observed execution time,

$$T_E = \frac{\upsilon}{S}\overline{t_e}. \tag{3.4}$$

Assuming the problem size remains constant, the application speed-up is equal to the total time for one worker divided by the total time for p workers plus overhead;

$$\begin{aligned} S(n,p) &= \frac{T_S}{T_p} && (3.5)\\ &= \frac{t_b(n) + 2\upsilon\overline{t_c} + \upsilon\overline{t_e}}{t_b(n) + \frac{2\upsilon}{p}\overline{t_c} + \frac{\upsilon}{p}\overline{t_e} + \Omega(p,n)}. && (3.6) \end{aligned}$$

Where $\Omega(p)$ is an overhead term that captures communication and parallel overheads and is assumed to be a function the number of workers p.

Amdahl's Law relates the theoretical maximum speed of serial algorithm when it is parallelized as follows;

$$S_{max} = \frac{1}{(1 - f_p) + \frac{f_p}{p}}, \tag{3.7}$$

where f_p is the fraction of the algorithm effectively parallelized. Equation (3.5) closely resembles Amdahl's Law, with:

$$\begin{aligned} fs &= \frac{t_b(n)}{T_S}; && (3.8)\\ fp &= \frac{\frac{2\upsilon}{p}\overline{t_c} + \frac{\upsilon}{p}\overline{t_e} + \Omega(p,n)}{T_S}, && (3.9) \end{aligned}$$

and noting that $f_s + f_p = 1.0$. Following Amdahl's Law, in the limit as p increases S_max approaches $\frac{1}{1-f_p}$. Furthermore, linear speedup is possible only when the algorithm can be totally distributed with no overheads *i.e.* $f_p = 1.0$. Realistically, it is expected that $T_B << T_C + T_E$ and that Ω will be small; nevertheless, sub-linear speedup is anticipated given the task decomposition strategy and implementation chosen for the framework.

3.4.2 Communications Model

Turning our focus to communications time, the analysis can extended to consider the impact of υ on total communication time. It is reasonable to consider communication time to be a function of the population and chunk size,

$$T_C = \frac{2\upsilon}{p} t_c(n, \upsilon). \tag{3.10}$$

Furthermore, one can assume that communication time is linear with respect to message size plus some fixed overhead associated with a message; thus $t_c = \alpha \frac{n}{\upsilon}\lambda + \beta$; where, α is the communication time per unit message length and β is the communication latency. Substituting and simplifying total communication time becomes,

$$T_C = \frac{2}{p}\left[\alpha n\lambda + \beta\upsilon\right]. \tag{3.11}$$

Upon inspection of the equation, one realizes that the variable costs of communicating are determined by the population size and that the fixed costs are a function of the number of chunks υ, and that communication time can be minimized by reducing the number of chunks to the greatest extent possible *i.e.* $\upsilon = p$.

The degree of performance improvement possible when parallelizing EAs is controversial. Speed-up performance can be categorized one of three ways: sub-linear, linear, or super-linear. Linear speed-up implies that each processor added to the computation is fully utilized, and furthermore that the communications overhead incurred is negligible. The theoretical limit on performance improvement when parallelizing algorithms structured like EAs is linear; furthermore, the author has shown that sub-linear speedup is anticipated in this investigation. Nevertheless, researchers have reported that super-linear speed up is possible when parallelizing EAs[3].

There is theoretical and empirical evidence that does support the super-linear performance claims. Population partitioning has the side effect of enhancing the diversity of the global population and therefore improving the global search characteristics of the algorithm and may contribute to faster convergence properties. Furthermore, this side effect on diversity may allow a reduction in the global population size. Since population partitioning enhances diversity, the probability of finding the optimal solution would stay roughly the same while modestly reducing the computational requirements. The effectiveness of this approach, however, is an open question and an area of active research.

[3]Assuming that parallel speed up is being measured correctly.

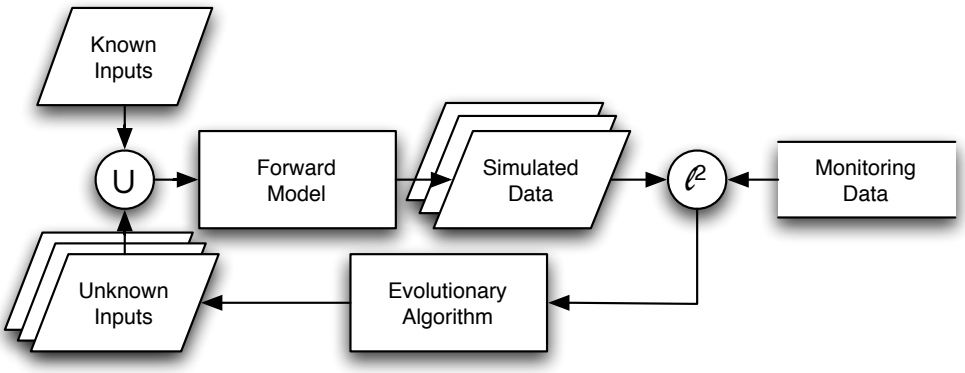

Figure 3.5: Inverse problem solution procedure flow chart (serial algorithm).

3.5 Applications

In this section the LASSO framework described previously is used to solve two groundwater inverse problems — a source identification problem and a source release history reconstruction problem. For both problems, this analysis focuses on the solution quality and application performance characteristics. In particular, the analysis evaluates the overall effectiveness of our solution approach in general and the framework in particular.

The serial form of the inverse problem solution procedure employed in this work is illustrated in Figure 3.5. The simulation optimization approach utilizes a forward model, usually a system of partial differential equations (PDEs), describes the dynamic processes of the environmental system, and defines the relationship between model inputs and outputs. Forward models are coupled with formal mathematical or heuristic search procedures to determine the model inputs that best approximate the observed data. A measure of difference between the modeled output and the observed data is the basis of the objective function. The objective value of the potential solution is computed, and using it, along with other criteria, the search procedure updates the decision variables to improve the approximation.

In the sections that follow the framework setup for the computational experiments is described in greater detail.

3.5.1 Parallel Groundwater Model

The simulation optimization framework developed here is generic and designed to be loosely coupled with the forward model. For this particular set of applications the framework was coupled with a groundwater hydraulics and transport simulation model — Parallel Groundwater REMediation 3D

Table 3.2: Parameter values for hypothetical groundwater domain used in simulation studies.

Parameter	*Value*
Problem size	500 x 300 x 10 m
Grid spacing	$dx = 10$, $dy = 10$, $dz = 1$ m
Grid size	51 x 31 x 11 nodes
Time step	20 days
Duration	22 yrs
Dispersion	$\alpha_L = 10$, $\alpha_{TH} = 5$, $\alpha_{TV} = 1$ m
Flow field	Heterogeneous
Velocity	Spatially variable
Execution time	19.2 sec
Source location	$[x_c = 9, y_c = 10, z_c = 4, s = 2, C_0 = 70]$

(PGREM3D) [32]. The simulator is based on finite element methods and is written in Fortran. The code is organized into flow and transport modules. The flow module solves the steady-state groundwater flow equations describing the head and flow field in the simulation domain. A detailed mathematical description of steady-state groundwater flow and solution techniques for the resulting system of equations is presented in [6]. The transport module solves the transport and chemical reaction equations in the simulation domain. Transport phenomena are described using the three dimensional advective dispersion reaction equation. For a detailed description of the transport and reaction model in PGREM3D refer to [32]. The transport module is parallelized using a two-dimensional domain decomposition. The MPI library is used to exchange information between domains. The code is written in Fortran using double precision arithmetic. The simulator has been tested extensively for scalability and performance on different parallel architectures [33] and [34].

3.5.2 Application Setup

As explained previously, the framework applications described here are concerned with the solution and computational performance achieved with the framework. Computational experiments were performed to measure solution quality using a hypothetical groundwater problem to synthetically generate contaminant concentration observations at monitoring wells. The hypothetical three-dimensional groundwater model domain considered in these applications is illustrated in Figure 3.6. The domain was 500m x 300m x 10m in size with a grid resolution for the simulation model of 17,391 (51x31x11) finite element nodes. The simulation duration was approximately 22 years, 8000 days or 400, 20 day time steps. A total of 18 monitoring wells were distributed evenly at the center and farthest end of the model domain as shown in Figure 3.6. Steady state flow with a heterogeneous velocity field was computed using a heterogeneous hydraulic conductivity field. The conductivity field was generated using the turning band method [52]. The execution time using a single node at TeraGrid-NCSA

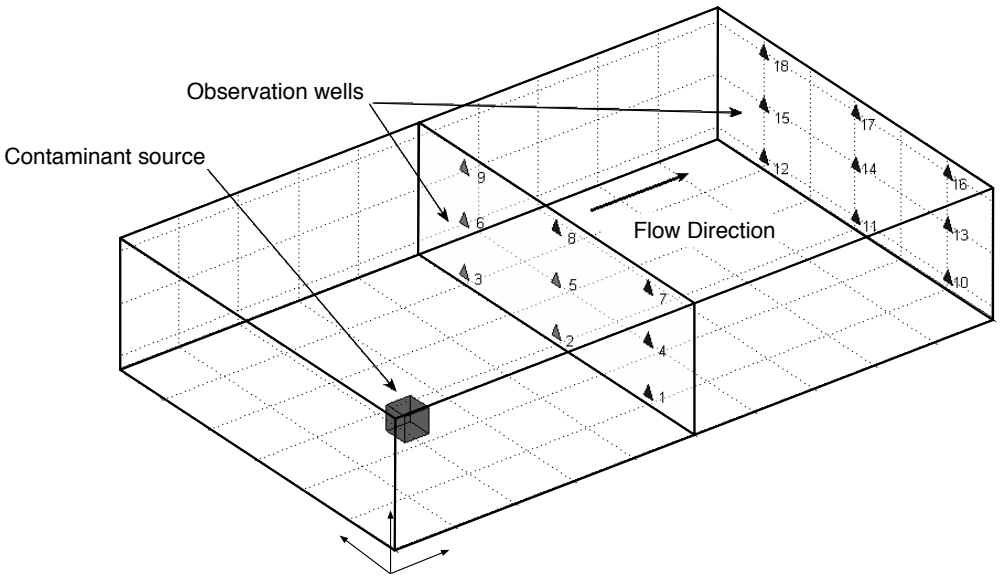

Figure 3.6: The hypothetical three dimensional groundwater simulation domain used for the inverse problem applications.

cluster was approximately 20 seconds. A summary of the geometrical and hydraulic parameters describing the groundwater model domain is presented in Table 3.2.

Using the hypothetical groundwater problem domain synthetic observations were generated by placing a conservative contaminant source with a constant release concentration in the aquifer at a predetermined location. The groundwater contaminant transport model was then used to generate the true concentration profiles at the monitoring wells. In turn, these profiles became the observed set of monitoring data used to formulate the inverse problems. It is important to note that these inverse problem applications assume the absence of measurement errors in the observed data.

Depending on the problem solved (which will be described in greater detail in the sections that follow) the true solution and the solution recovered by the inverse problem were used to calculate the solution error. The solution error metric used here was based on the L_2 norm of the normalized differences between the solution vectors as follows,

$$\epsilon_s = \frac{1}{m} \frac{\|\mathbf{x} - \hat{\mathbf{x}}\|_2}{\|\mathbf{x}\|_2}, \tag{3.12}$$

where $\mathbf{x}$ and $\hat{\mathbf{x}}$ are vectors describing the identified and true sources respectively, and m is the number of elements in $\mathbf{x}$ each vector. It is intuitive that solution error can only be calculated when the true solution is known; solution error can not be calculated for "real world" problems. Thus, the use of the hypothetical groundwater problem and synthetic data generated from a known source is useful when evaluating the performance of the proposed solution procedure, but otherwise unrealistic.

Considering the computational burden associated with inverse problems in general and with the proposed solution procedure in particular, the computational performance of the framework was a critical factor. Therefore, experiments were also performed to measure computational performance during solution of the applications. Computational performance was evaluated using runs performed on the TeraGrid with single-site and cross-site configurations.

3.5.3 Source Characterization Problem

The specific environmental inverse problem considered in this section is a groundwater source identification problem. The solution of source identification problems are important for environmental forensics and when characterizing contaminate releases for the purposes of accessing liability. In this problem context, parameters for the source coordinates, size, and concentration describe the source locations and mass flux of contaminant released. These parameters are the unknown model inputs which are resolved from spatially and temporally distributed observational data collected at monitoring wells located in the groundwater field domain.

For the purpose of solution, the inverse problem is formulated and solved as an optimization

model, the objective of which is to identify the forward model inputs which minimize the error between the simulated and observed monitoring data. The relationship between contaminant releases at the source and concentrations observed at monitoring wells is defined by the groundwater forward model. There are several ways to parameterize the contaminant source; here, it is assumed to be a compact cubical shape with a constant release concentration. The source location and contaminant mass flux is completely described by five parameters, the Cartesian coordinates of its centroid (x_c, y_c, z_c), the length of an edge s, and the release concentration C_0.

The contaminant concentration at monitoring well i at time t resulting from the contaminant release is $Cm_i(t)$. Concentration measurements are assumed to occur at the monitoring wells at discrete sampling times $k = 1, 2, \ldots, t$ with a sampling interval Δt. Measurements are taken over a finite time horizon extending from $[0, T]$. The start of the monitoring period time $tm_0 = 0$ is assumed to occur when the contaminant is detected at one of the n_w wells. The vector $\mathbf{Cm}$ in expanded form then becomes;

$$\mathbf{Cm} = (Cm_1(1)Cm_1(2)\ldots Cm_1(T)Cm_2(1)\ldots Cm_2(T)\ldots Cm_{n_w}(T))'. \tag{3.13}$$

The objective function is the prediction error, here the L_2 norm of the difference between the simulated $\mathbf{Cm}$ and observed $\hat{\mathbf{Cm}}$ concentrations at the wells,

$$\min_{\mathbf{X_c}, \mathbf{Y_c}, \mathbf{Z_c}, \mathbf{S}, \mathbf{C_0}} \|\mathbf{Cm} - \hat{\mathbf{Cm}}\|_2. \tag{3.14}$$

The objective is minimized over the decision variables describing the source parameterization. Constraints are written imposing bounds on the decision variables;

$$x_{min} \leq x_c \leq x_{max} \tag{3.15}$$

$$y_{min} \leq y_c \leq y_{max} \tag{3.16}$$

$$z_{min} \leq z_c \leq z_{max} \tag{3.17}$$

$$s_{min} \leq s_c \leq s_{max} \tag{3.18}$$

$$0 \leq C_0 \leq C_{max}. \tag{3.19}$$

Taken together Eqns. (3.14) through (3.19) constitute the source identification problem formulation. For this computational study, the minimum and maximum values bounding (x_c, y_c, z_c) were set to the extents of the modeled domain $(1, 51)$, $(1, 31)$, and $(1, 11)$ respectively. The cubical source edge length s was bounded to $(1, 10)$ and C_0 was bounded to $(0, 100)$.

Several preliminary trials were conducted to fine-tune ES search parameters (population size,

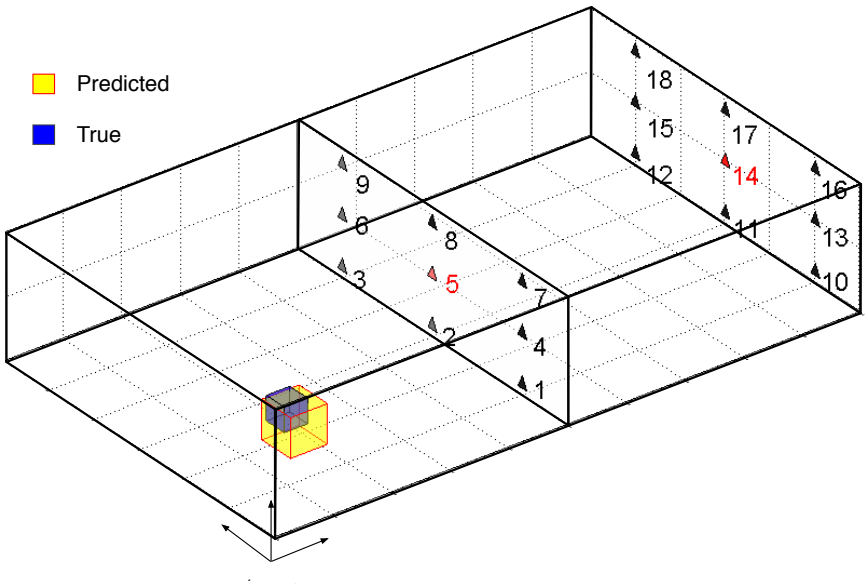

Figure 3.7: True and predicted source locations for the hypothetical groundwater source identification problem.

number of generations, and mutation rate). An $ES(\mu+\alpha)$ search with real decision variable encoding was used in the study with $\mu = 100$, $\alpha = 100$, and the mutation parameter $\alpha = 1.0$. Convergence criteria were based on a fixed quantity of computational effort. The search was terminated after 100 generations, thus the theoretical maximum number of forward model evaluations was 10,000. Like all EAs an ES-based search is a probabilistic method, thus, thirty random trials were performed to evaluate the robustness of the search.

The best result obtained from the thirty random trials was $(x_c, y_c, z_c) = (9.35, 9.59, 3.11)$, $s = 3.17$, and $C_0 = 69.52$ (note that the true solution is displayed in Table 3.2). The objective error and solution error for the source identification problem were 0.0276 and 0.1252 respectively. This result was typical of the other solutions identified in the random trials. Comparison of predicted versus observed concentration values at two representative monitoring wells are shown in Figure 3.8 illustrating the goodness of fit of the solution in objective space. The estimated source locations and sizes are shown in Figure 3.7 illustrating the goodness of fit of the solution in solution space. The source characterization procedure was able to identify solutions with relatively low prediction errors;

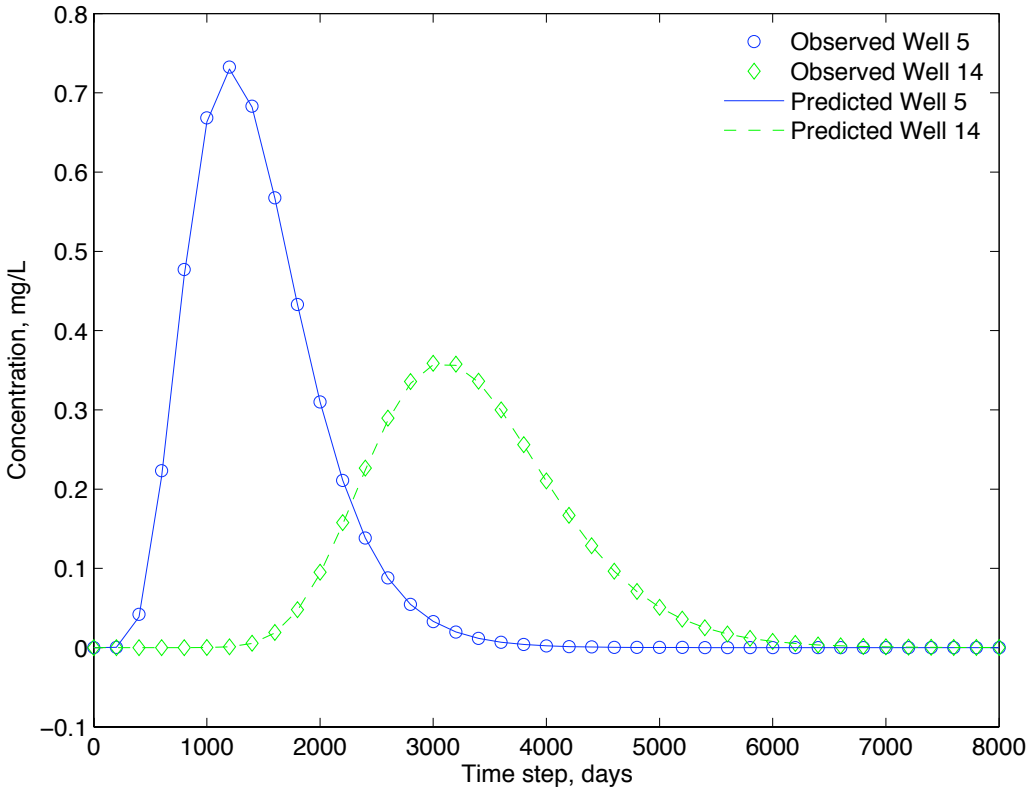

Figure 3.8: True and estimated source profiles at monitoring wells 5 and 14.

Table 3.3: Fitted model parameters for single-site performance runs.

Run	L_2 *Norm Residuals*	f_p
NCSA Semi-coarse	3.3865e+00	9.9073e–01 ± 7.8816e–04
NCSA Coarse	6.6387e–01	9.7890e–01 ± 1.9539e–03

however, the solution error was relatively high (in this case approximately 13 percent) with most of the error associated with the estimates for the z_c and s parameter values. This may be attributed to the lack of sensitivity the monitoring well locations have with respect to these variables and the solution non-uniqueness associated with the problem.

Single-Site Performance

Timing studies were conducted to investigate the framework's computational performance as a function of semi coarse and coarse grained parallelism. The timing runs differ from the solution performance runs described previously in that the population size was set to 128 individuals and and the search was terminated after 10 generations. Based on preliminary fine grained scaling runs the group size was set to a single processor for the performance runs. Improvement in fine grained parallelism is associated with problem size, and since the problem size used in this application study was fixed and relatively small, performance improvements associated with scaling to exploit fine grained parallelism were negligible.

The framework configuration used to conduct the timing studies were based on preliminary performance results and resource availability at the TeraGrid NCSA site. For each timing scenario, tasks were distributed among four servers and forward model instances were configured with 32 groups and a single processor per group as mentioned previously. The node allocation for the runs breaks down as follows: one node for the optimization application, four nodes for servers, and 68 nodes (134 processors) for the groups with masters; thus, the computational performance study utilized a total of 73 nodes and 146 processors.

The first set of runs was used to investigate the timing issues associated with semi-coarse grained parallelism. Here the wall time was measured as the number of groups increased and the other parameters including the population size, the number of processors per group, and the number of tasks per group were kept unchanged. Speed-up results for the single-site semi-coarse grained runs are shown in 3.9. Based on the theoretical performance analysis performed in Section 3.4.1 sub-linear speed-up is anticipated. The figure illustrates linear speed-up (in magenta), and the speed-up observed (blue circles). Indeed, the results indicate sub-linear speed-up. Further, the speed-up observed exhibits a diminishing return as the number of groups is increased characteristic

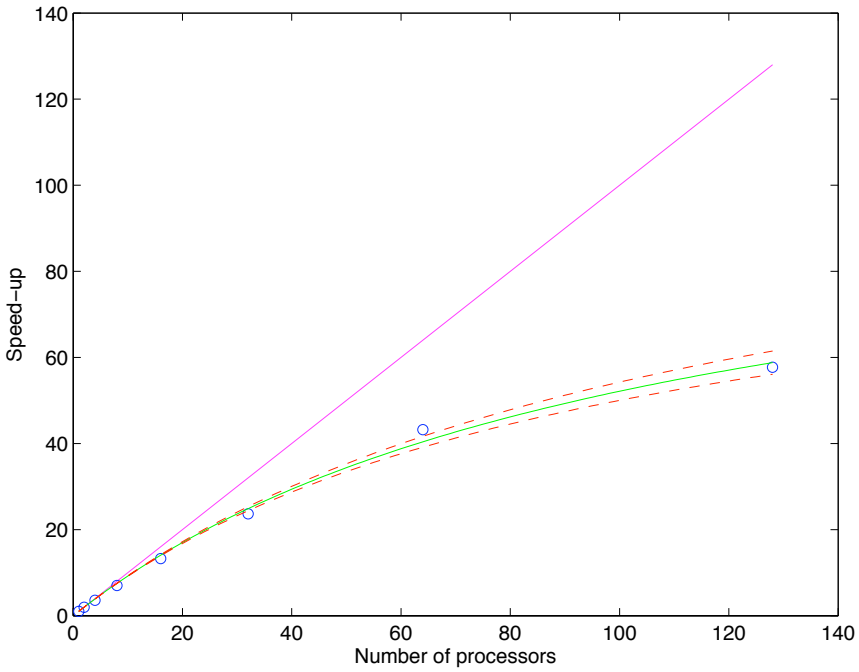

Figure 3.9: Semi-coarse grained parallelism (Procs/Group = 1:1, Tasks/Group = 1:1)

of Amdahl's Law.

To better understand the observed behavior and verify the speed-up model developed in Eqn. (3.7), the data was used to estimate the model parameter f_p using non-linear least squares. The model parameters estimates with 95 percent confidence intervals are shown in Table 3.3 and in Figure 3.9 the speed-up predicted using the model (green) with the 95 percent confidence interval (dashed red) are illustrated. The parallel fraction f_p predicted by the model was approximately 99 percent with a high degree of certainty. Overall, the model fit the data well, exhibiting a small residual norm and a narrow confidence region.

The second set of runs was used to investigate the computational time associated with coarse grained parallelism. Here the wall time was measured while the number of workers available to the computation was successively doubled. The other parameters including the population size, the number of processors per group, the number of groups per worker, and number of tasks per group were kept unchanged. Speed-up results for the single-site coarse grained runs are shown in Figure 3.10. Again, sub-linear speed-up results were anticipated given the theoretical analysis preformed. Equation (3.7) was able to fit the observed speed-up data well. The parallel fraction of the algorithm f_p was approximately 98 percent, slightly less when compared to the semi-coarse runs. Furthermore, this difference is likely to be significant. These results suggest that the framework

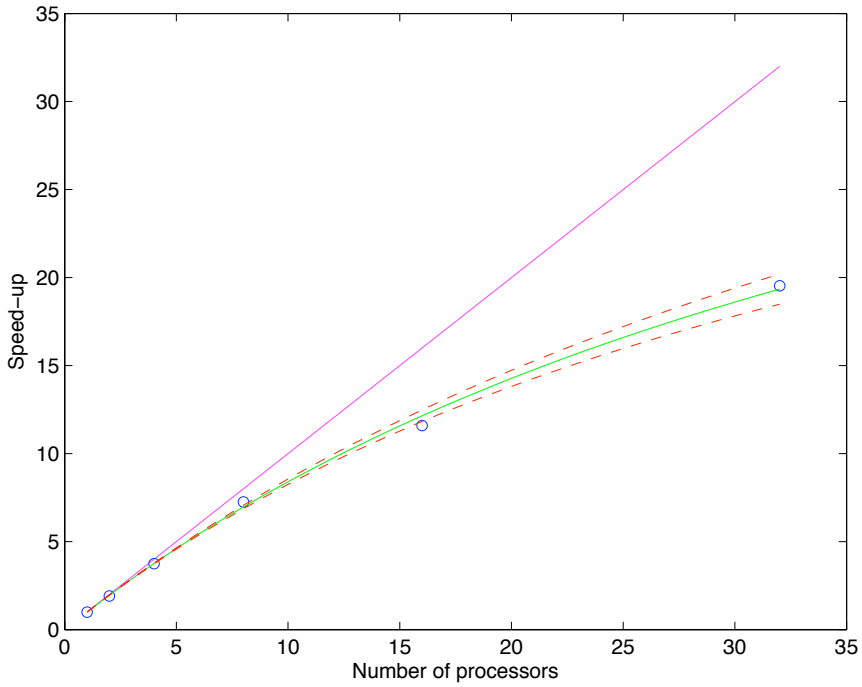

Figure 3.10: Scaling study, coarse grained parallelism (Groups/Worker = 1:1, Tasks/Group = 1:1, Population size = 128, Procs/Group = 1:1)

scales less efficiently at the coarse grained level compared to the semi-coarse grained level.

For both the semi-coarse and coarse grained levels of parallelism, the speed-up observed exhibits a diminishing return as the number of groups is increased characteristic of Amdahl's Law. This behavior is a consequence of the small but finite serial fraction of the algorithm. The serial fraction for the framework configuration tested was approximately 1 - 2 percent. Indeed, the assumption that $T_B << T_C + T_E$ was reasonable; nevertheless, a monotonic decrease in speed-up was observed and diminishes the utility of each additional processor in accordance with Amdahl's Law.

Viewing these results from another perspective, the ratio of the theoretical wall time using a single node to the wall time utilizing the framework was used to calculate speed-up ratios. For the scenario evaluated here, the theoretical wall time on a single processor was 191,900 seconds. With the framework the wall time was approximately 1744.41 seconds; thus, a speedup of approximately 110:1 was achieved utilizing 146 processors. Put another way, the total wall time for the job was reduced from approximately 2 days to 29 minutes. These speedup results show that significant and meaningful raw performance improvements are achievable using the framework[4].

3.5.4 Release History Reconstruction Problem

The specific environmental inverse problem considered in this section is a groundwater release history reconstruction problem. The release history reconstruction problem is closely related to, but is less general than, the source identification problem considered in the previous section. In this problem context, historical contaminate release schedules at source locations are the unknown model inputs which are resolved from spatially and temporally distributed observational data collected at monitoring wells located in the groundwater field domain. Once again the inverse problem is formulated and solved as an optimization model, the objective of which is to identify the forward model inputs which minimize the error between the simulated and real observational data. The simulated concentrations are generated using the forward simulation model, which provides the relationship $\mathbf{Cm} = f(\mathbf{Cr})$, where $\mathbf{Cr}$ is the source release concentration time series and $\mathbf{Cm}$ is the time series of monitored concentrations.

The source release history is reconstructed over a finite time horizon $[0, T]$ extending from the time in the past when the monitoring activities started tr_0 towards the present time T, and referred to the as the release history reconstruction period. The contaminant concentration released at source $j = 1, 2, \ldots n_s$ at time t is $Cr_j(t)$. The contaminant release schedule at the source is discretized into intervals $k = 1, 2, \ldots, t$ with an interval length Δt_r over which the concentration is assumed to remain constant. The release schedule vector $\mathbf{Cr}$ in expanded form then becomes;

[4]For those readers interested, a more detailed evaluation of solution and computational performance for the source identification problem performed by Mirghani can be found in [39].

$$\mathbf{Cr} = (Cr_1(1)Cr_1(2)\dots Cr_1(T)Cr_2(1)\dots Cr_2(T)\dots Cr_{n_s}(T))'. \tag{3.20}$$

The time series of concentrations detected at monitoring wells is parameterized as stated previously in Eqn. (3.13). With no loss of generality it is assumed that $\Delta t_r = \Delta t_m$; further, in general tm_0 and tr_0 do not correspond, however, for the problem being studied here it is assumed that $tm_0 = tr_0 = 0$. The objective of the optimization model here is to minimize the 2-norm between the simulated and observed concentrations at the wells as follows,

$$\min_{\mathbf{Cr}} \|\mathbf{Cm} - \hat{\mathbf{Cm}}\|_2, \tag{3.21}$$

where the concentration time series $\mathbf{Cm}$ and $\hat{\mathbf{Cm}}$ are the simulated and observed concentrations respectively. The vector of released concentrations $\mathbf{Cr}$ are the set of decision variables over which the objective is minimized. Constraints are written to enforce decision variable positivity,

$$0 \leq \mathbf{Cr_j^k} \leq c_{max} \tag{3.22}$$

$$\forall j = 1, \dots, n_s; \forall k = 1, \dots, t \tag{3.23}$$

The number of decision variables is equal to the number of sources times $(T - tr_0/\Delta t_r)n_s$, where n_s is the number of sources.

The source history reconstruction was formulated using a single contaminant source with 18 monitoring wells sampled at a frequency, Δt_m, equal to 10 times the simulation model time step. Thus, a total of 1800 observations are used corresponding to 18 wells and 100 periodic samples. The release history reconstruction period, $[0, T]$, was equal to the simulation duration, and the source release schedule discretization, Δt_r, was equal to 100 times the simulation model time step. Thus, the optimization problem contained 10 decision variables.

A generational GA with elitism and real decision variable encoding was used to solve the optimization model developed immediately above. Several preliminary trials were conducted to estimate GA parameters. The GA population size was set to 128 individuals, and the crossover and mutation rates were set to 0.70 and 0.2 respectively. Termination criteria were based on a fixed set of computational effort of 100 generations. As in the previous application, the results for the source history reconstruction problem were recorded for 30 random trials.

The framework configuration used to solve the source release history reconstruction problem was based on the results of preliminary performance runs and the availability of computational resources at TeraGrid sites. The configuration utilized three TeraGrid sites as follows; the optimization

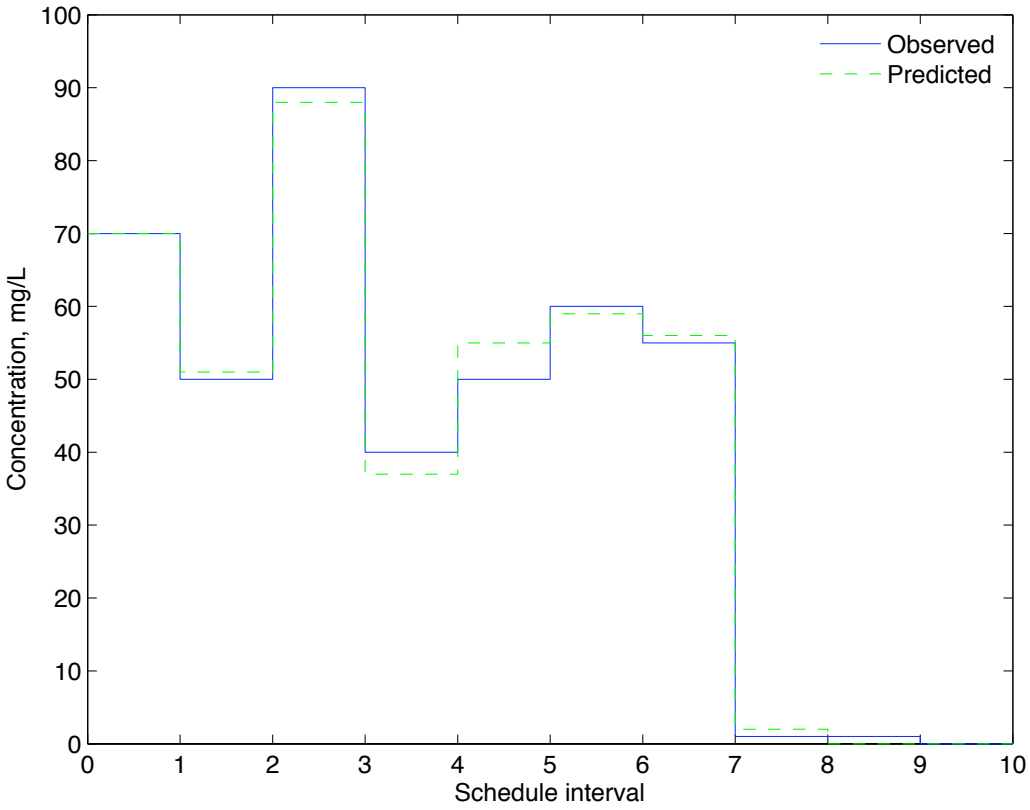

Figure 3.11: Observed and predicted release histories occurring at source.

application (master) was run on the ANL/UC site, and the forward models were executed on both the NCSA and SDSC sites. The theoretical maximum number of forward model evaluations performed during the search is equal to the population size (128) multiplied by the number of generations (100); thus, the maximum number of evaluations was 12,800. These evaluations were distributed across the workers running on the NCSA and SDSC sites, with each site running two servers. Furthermore, each server was running 32 processor groups within the forward model with one processor per group. Thus, a total of 73 computing nodes distributed across three sites were utilized breaking down as follows; a single node assigned for the optimization application at the ANL/UC site, and two nodes for the servers with 34 nodes (68 processors) for the groups with masters in each NCSA and SDSC sites.

The best predicted time series of source release concentrations generated over the random trial are shown with the true releases in Figure 3.11. A typical example of the observed concentration profile compared to the predicted profile at Observation Well 5 is shown in Figure 3.12. The GA performed well and was able to estimate the source release history with an objective function error

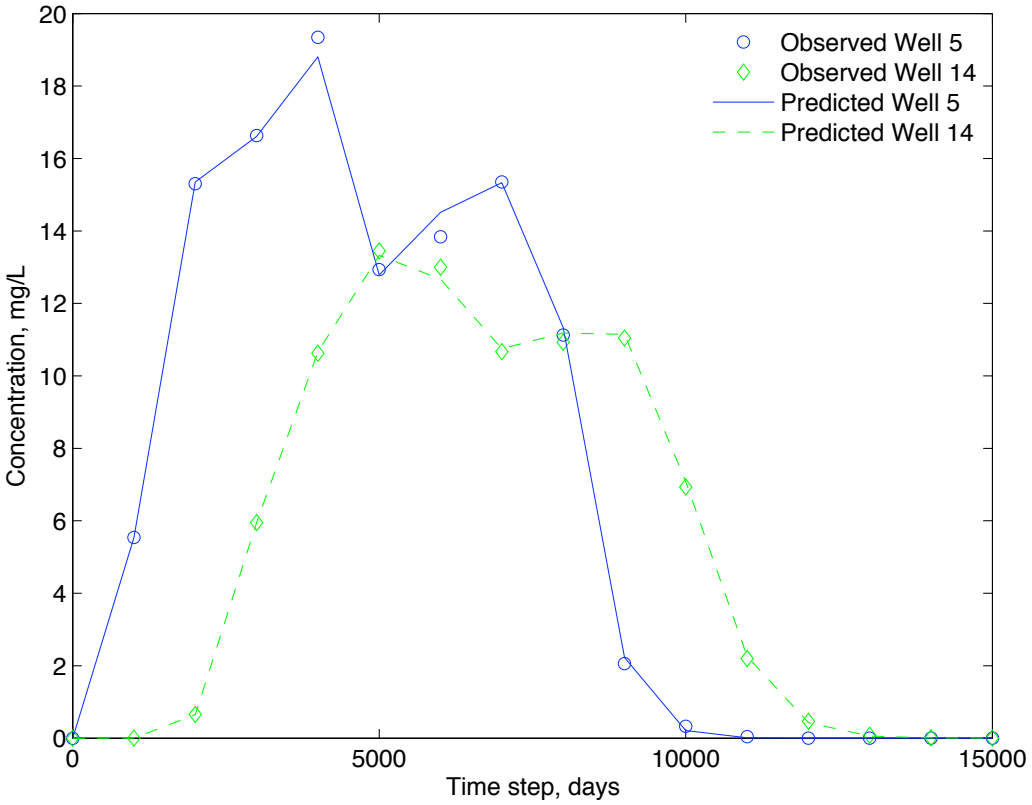

Figure 3.12: True and estimated source profiles at monitoring well 5 and 14.

equal to 2.103 and a solution error equal to 0.247 percent.

Cross-Site Performance

The solution of the source history reconstruction problem presented immediately above was obtained via a cross-site framework configuration. In this section, the computational performance characteristics associated with cross-site configurations on the TeraGrid are studied in greater detail. Two sets of timing studies were performed with the optimization application (master) and forward model (workers) distributed across the ANL/UC, NCSA, and SDSC TeraGrid sites. The first set of runs was developed to study communications costs for cross-site TeraGrid configurations of the framework. The second set of runs were used to better gauge cross-site effects on application speedup. Both of the timing studies presented here facilitate the characterization of communication cost and latency associated with the various TeraGrid interconnects in particular and cross-site TeraGrid performance in general.

The first timing study focusing on communications costs included three sets of runs — a single-site and two cross-site configurations. Master and worker processes were both located at ANL/UC for the single-site configuration, while the master was run at ANL/UC and the workers at NCSA and SDSC for the cross-site configurations respectively. While making the runs the population size, number of servers, and processors per group were held constant. Furthermore, the ratio of tasks/group was held constant at 1:1; however, as the number of groups was increased the number of tasks per chunk and correspondingly the message size also increased. The data from the runs was analyzed to validate the linear communications cost assumption presented in Section 3.4.2) and estimate model parameters quantifying these costs.

Results for the communications cost model verification are shown in Figure 3.13. The Figure shows the measured data and the results of fitting the linear communications model. When interpreting the linear model, the intercept with the y-axis is the communication latency β, while the slope is a measure of communication cost per unit message size α. Parameter values for each of the three runs were estimated by linear least squares with their corresponding values shown in the legend of Figure 3.13.

These results indicate that fixed communication time is almost negligible between ANL/UC and NCSA, however, it is quite significant between ANL/UC and SDSC. Furthermore, the variable component of communication time is very similar for the single-site and ANL/UC and NCSA cross-site configuration, and significantly greater for the ANL/UC and SDSC configuration. Surely these result were anticipated, but the magnitude of the differences and their impact on the cross-site speed-up model is of primary interest. The results also suggest that the linear communications model is a reasonable approximation within a single-site and over short Teragrid interconnects. The linear

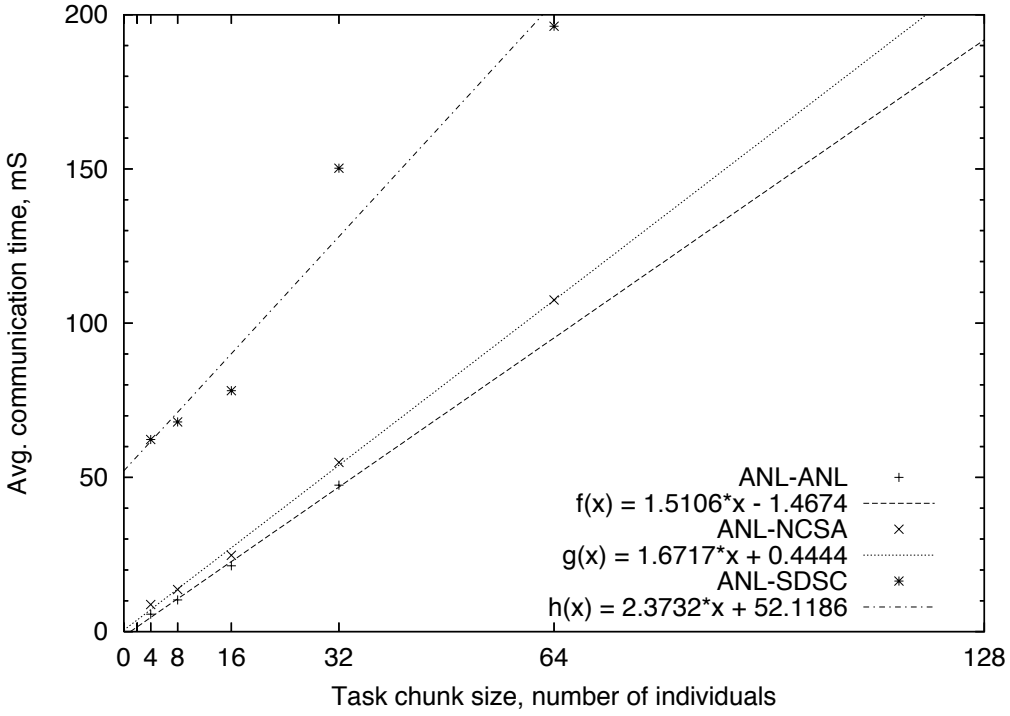

Figure 3.13: Average communication time as a function of task chunk size.

Table 3.4: Fitted model parameters for cross-site performance runs.

Run	*L2 Norm Residuals*	f_p
ANL/UC - ANL/UC	1.7701e+00	9.9171e–01 ± 9.5069e–04
ANL/UC - SDSC	1.7128e+01	9.8960e–01 ± 4.4872e–03
ANL/UC - NCSA	1.3219e+01	9.8795e–01 ± 4.0624e–03
ANL/UC - NCSA, SDSC	6.6635e+00	9.8769e–01 ± 2.0970e–03

model fits the cross-site data between ANL/UC and SDSC rather poorly. The poor fit may have occurred due to errors in the performance data due to high network traffic, or more significantly, the linear model may not adequately describe communication time over very long Teragrid interconnects.

Having gained insight into cross-site communication costs from the first set of timing studies, a second set of studies was performed to better understand the effect cross-site communication costs have on the speedup observed for practical applications. This second timing study was conducted in much the same way as the first. However, in addition to the single-site ANL/UC runs and the ANL/UC to NCSA, and ANL/UC to SDSC runs, cross-site runs were performed with the master located at ANL/UC and workers distributed across both the NCSA and SDSC locations. Again, for each set of runs, the wall-time was measured as the number of groups was increased (parallelism at the semi-coarse grained level), while the population size, number of processors per group, and number of tasks per group were kept unchanged. The speed-up data was then modeled using Eqn. (3.7) with model parameters determined using robust non-linear least squares.

Performance data for the single-site configuration (ANL/UC to ANL/UC) is shown in Figure 3.14 and parameter values for the speed-up model are shown in Table 3.4. When viewing Figure 3.14 make note that sufficient computational resources were not available at ANL/UC to perform the 128 group run. The best speed-up performance was observed for the single-site runs. Indeed, the speed-up produced by the 64 group single-site configuration was greater than that observed for any of the other configurations including the 128 group runs.

As with the performance study in the previous section, the speed-up observed for these single-site runs were well represented using Amdahl's Law. The model predictions and model parameters both had small confidence regions and the norm of the residuals for the model fit was small. The parallel fraction f_p determined by the speed-up model fit was approximately equal to 99 percent. Viewing the confidence interval for the parameter estimates (see Table 3.4), this value was significant relative to some of the other configurations.

Performance data for the ANL/UC to SDSC and ANL/UC to NCSA configuration are illustrated in Figures 3.15 and 3.16 respectively. The speed-up observed for these cross-site configurations was smaller than the other configurations. Applying the speed-up model to the data yielded an acceptable

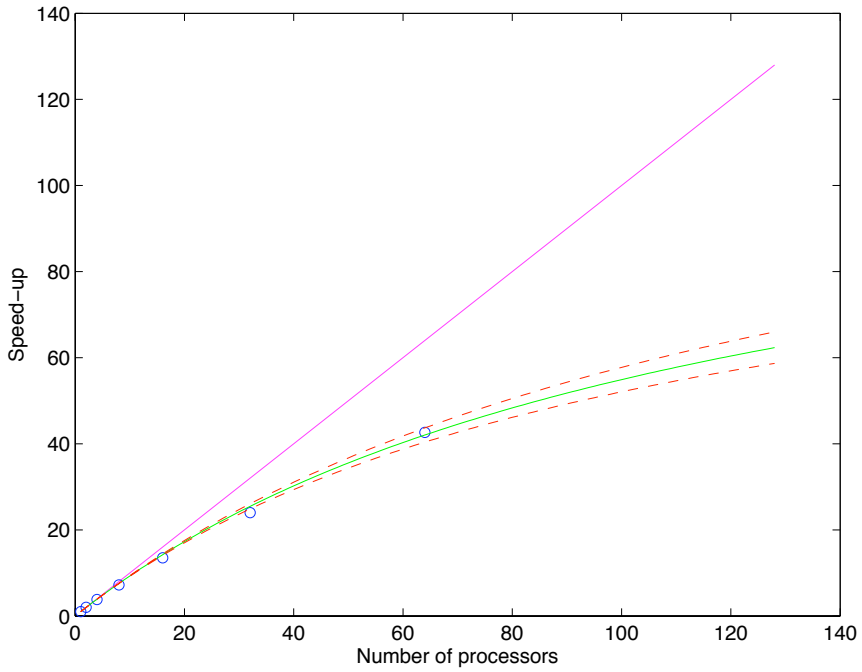

Figure 3.14: Cross-site scaling study, ANL/UC-ANL/UC (Proc/Group = 1:1, Tasks/Group = 1:1).

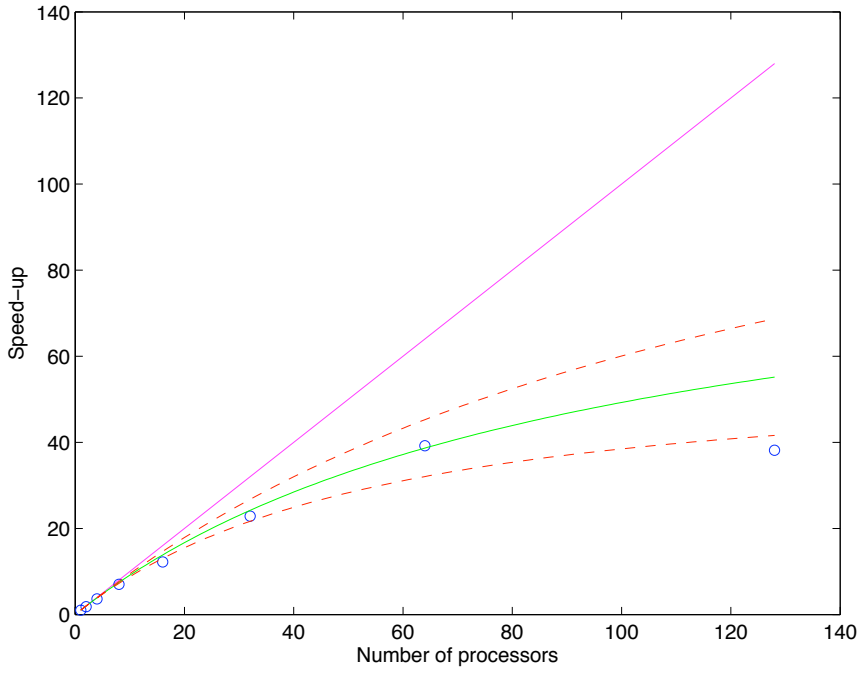

Figure 3.15: Cross-site scaling study, ANL/UC-SDSC (Proc/Group = 1:1, Tasks/Group = 1:1).

model fit. The confidence region for the model predictions and parameter estimates, however, were larger. Thus, a higher degree of uncertainty is associated with these estimates. The magnitude of the parallel fraction f_p estimates were smaller for these cross-site configurations by a fraction of a percent compared to the other configurations. Referring to the parameter estimate confidence values in Table 3.4, surprisingly, these differences, though small, were significant in some cases.

Assuming the robust model fit is appropriate for the data, significant overheads $\Omega(n, p)$ were observed between 64 and 128 groups for both the ANL/UC to SDSC and ANL/UC to NCSA runs. This is evident from the high model residuals repeatedly present for the 128 group runs across the different configurations. This phenomena is most evident for the ANL/UC to SDSC configuration and can be seen in Figure 3.15.

The performance data for the ANL/UC to SDSC, NCSA cross-site configuration are shown in Figure 3.17. The speed-up observed for this configuration was not significantly different from the other cross-site configurations. The confidence intervals associated with the model predictions and parameter estimates, however, were smaller, suggesting that the model fit was slightly better than those for the other cross-site configurations. The parallel fraction f_p was approximately 98.7 percent and not significantly different than that estimated for the other cross-site configurations. Again, computational overheads $\Omega(n, p)$ were observed with a significant difference existing between the model prediction and the speed-up measurement for the 128 group run.

Why do the differences in speed-up performance evident in the performance data and corroborated by the speed-up model occur? There are generally three causes of poor scaling efficiency, the inherent sequentiality of an algorithm, communication overheads, and load imbalances. The speed-up predicted by Amdahl's Law for the performance study runs was sensitive to the serial fraction of the algorithm. Even though the serial fraction estimated for these runs was consistently between 1 to 3 percent, this is sufficient to bound computational performance. Recall from the communications model and analysis performed previously, that the ANL/UC to ANL/UC run had the lowest latency and variable communications costs. The relative differences between configurations in this part of the performance analysis may also be attributed to the differences in fixed and variable communication costs observed in the communication study runs. It is important to note, however, that there is a level of uncertainty not represented in these data sets as communication costs are a function of network traffic known to be governed by stochastic processes. When workers and the master process are running on heterogeneous computational resources via networks with heterogeneous communications costs the framework, as it was designed, adaptively balances loads to minimize computation times. This is accomplished passively as slower workers and workers connected via higher latency communication paths tend to process fewer tasks and faster workers and

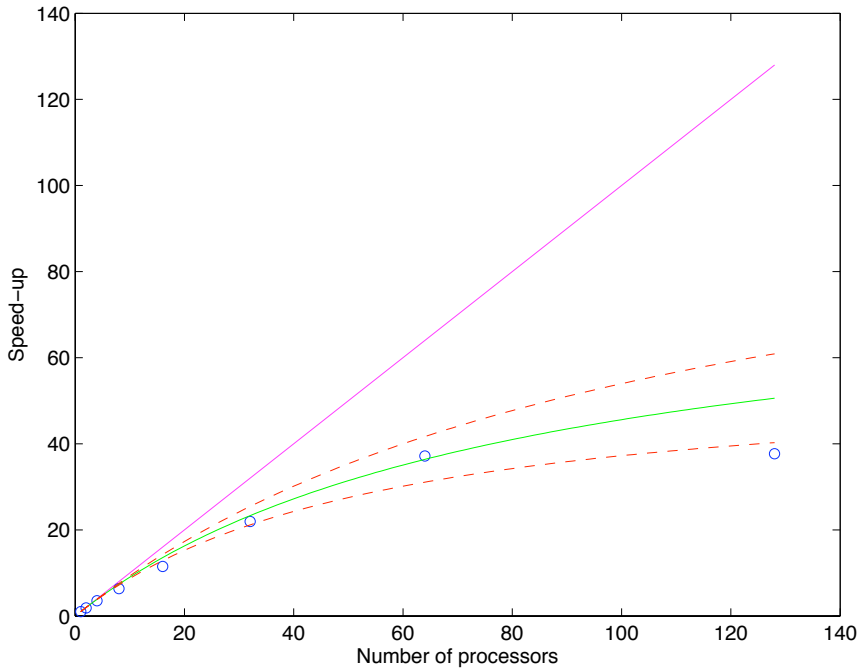

Figure 3.16: Cross-site scaling study, ANL/UC-NCSA (Proc/Group = 1:1, Tasks/Group = 1:1).

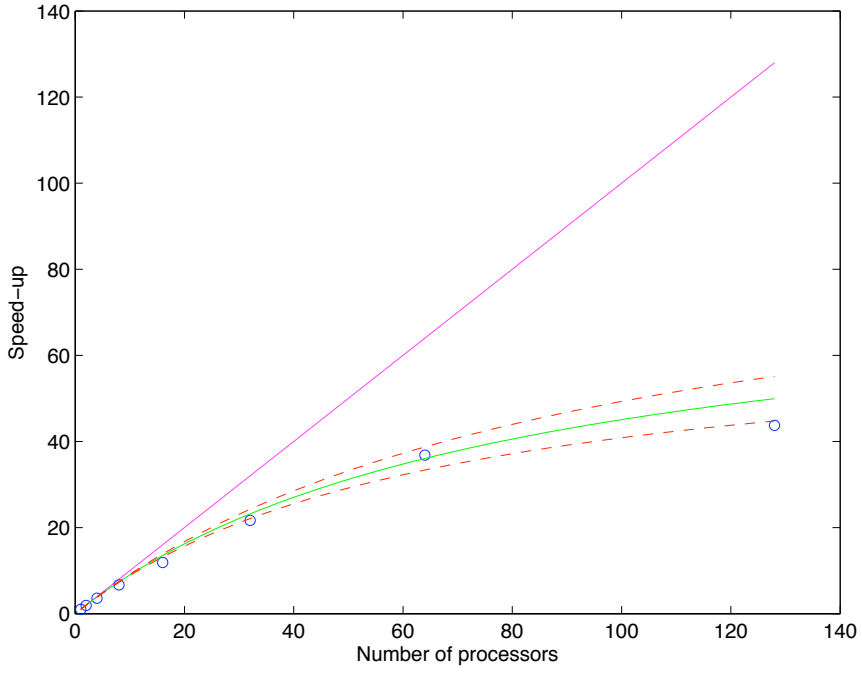

Figure 3.17: Cross-site scaling study, ANL/UC-NCSA, SDSC (Proc/Group = 1:1, Tasks/Group = 1:1).

workers with lower communications costs tend to process more. Based on the framework configuration employed for performance study, however, load imbalance can be eliminated as the cause of the performance degradation observed. The performance model and the data set it was applied to, did not have the resolution to distinguish between these different factors which effected speed-up.

The results were more favorable, when considering raw performance improvements. The normalized computation time for a single evaluation with the framework settings described above took approximately 0.210 second, versus 18.878 seconds on a single processor, i.e., solving the problem took approximately 35 minutes instead of approximately 50 hours. The speedup results show that the performance of the framework for this particular problem was approximately 90 times faster compared to the single processor results[5].

3.6 Discussion

There are, however, several shortcomings with the current task pool implementation. The results of the computational performance studies indicate that the manager process is a performance bottleneck when the number of worker processes is large. Another performance issue is that the optimization algorithms used herein execute in a synchronized manner where all tasks in the task pool are performed before the next set of tasks is added. This implies that the processors are periodically idle while the optimization algorithm generates a new set of tasks. Performance issue of this nature, however, are best addressed in a detailed analysis of the optimization algorithm itself.

There are several alternatives for addressing the performance bottlenecks of the current task-pool implementation. One alternative would be to decentralize the task pool. Different implementation approaches would result in task pools with varying degrees of decentralization. In this approach, the master would manage several task pools simultaneously. The worker processes would then have multiple task pools to connect with, spreading out communications over several socket connections and hopefully decreasing contention. Another more radical alternative would be to completely distribute the task pools to the workers themselves. Such an approach would be completely decentralized. To prevent load imbalances, the workers could communicate amongst themselves with the faster workers requesting tasks from slower worker's task pools. Several load balancing schemes exist for organizing these communications — asynchronous round robin, global round robin, and random polling — each with its own advantages and disadvantages.

The framework design considered here assume that the resources available to the computation on the grid are static. It is more general to consider the case where grid resources are dynamic. Unfortunately, this increases the complexity of the application considerably. An explicit scheduling

[5]For those readers interested, a more detailed evaluation of computational performance for the LASSO framework performed by Mirghani can be found in [39].

process would be required that could manage the dynamic resources of the grid. Such a process would be responsible for managing resources required by the computation. This would involve acquiring available resources and transferring unfinished tasks when resources become unavailable. Such a scheduling process could be combined with the existing task pool strategy, by simply launching workers on new resources as they become available. An approach for transferring tasks between workers no longer available and new workers would be required. Such a strategy would be more in keeping with the "grid" model of distributed computing; however, it is not within the scope of the current work.

3.7 Conclusions

This chapter describes the development of a distributed simulation optimization framework. The framework was designed with a centralized optimization application and a simple master worker task pool distribution strategy. A theoretical performance analysis was performed on the master worker task distribution strategy and multi-threaded task pool implementation. Assuming a fixed problem size, an expression for application speed-up was derived and shown to be a special instance of Amdahl's Law. Further, a simple communications model was developed assuming a linear function for communications cost. Inspection of the resulting expressions shows that communication cost can be minimized when aggregating tasks such that the number of task chunks is equal to the number of workers.

Illustrative applications were performed to demonstrate the effectiveness of the framework. The framework was coupled with a parallel 3 dimensional groundwater hydraulics and transport model for solution of a source identification and a source history release reconstruction problem. The source identification problem was formulated and solved as an optimization problem using an evolutionary strategy (ES). The ES was able to successfully identify the source location and approximate the contaminant mass flux with an acceptable degree of error. A battery of runs was conducted to measure computational performance for the single-site framework configuration at the semi-coarse and coarse grained levels of parallelism. The parallel fraction of the framework was estimated at approximately 99 percent using the data resulting from the runs and the theoretical speed-up model developed previously. The framework was found to scale more efficiently at the semi-coarse grained rather than the coarse grained level.

Similarly, the source history reconstruction problem was also formulated as a optimization problem and solved using a genetic algorithm (GA). The GA was able to reconstruct the historical mass fluxes occurring at a known source location within an acceptable degree of error. The framework was

configured for cross-site runs with worker processes at various cluster on the TeraGrid. The theoretical communication cost model was used to interpret data from a subset of the runs performed. The linear cost model was found to represent the observed data accurately. Further, it was evident from the results that the fixed and variable communication cost associated with a message were a function of the leg of the backplane interconnect the message traversed. A second set of runs were conducted to better understand the effect that heterogenous communication costs would have on practical applications of the framework. The data was analyzed using the theoretical speed-up model, and again the parallel fraction of the framework configuration was estimated at approximately 99 percent. Minor differences in speed-up efficiency were observed among the different framework configurations tested. Analysis of the results revealed a degradation in speed-up efficiency when scaling from 64 to 128 groups that was not explained by the model. This loss of efficiency was attributed to parallel and communication overheads but the data set and speed-up model lacked the resolution to provide a specific characterization of the overhead.

The efficacy of a simple application architecture and task distribution strategy for grid-enabling a simulation optimization framework was demonstrated. The results indicate that significant and meaningful raw computational performance improvements were achieved without sacrificing solution performance when applying the framework to representative environmental characterization problems.

Chapter 4

Monitoring Design for Source Identification in Water Distribution Systems

This chapter documents the formulation of an environmental monitoring problem and its solution. Using the linear I/O water quality model developed in Chapter 2, a monitoring sensor network design problem is formulated that expresses the conditioning of the source identification inverse problem as its objective. The formulation is based on methods for optimal inverse experiment design and results in a non-linear combinatorial optimization problem. The design problem is solved using the distributed simulation optimization framework developed in Chapter 3. Results for example networks seen previously in the literature are presented. The results illustrate the tradeoff among the location and number of monitoring stations and source identification inverse problem conditioning. Such information could most directly be utilized to help locate monitoring sensor in water distribution networks to improve the solution quality of source identification inverse problems.

4.1 Introduction

Accidental contamination events have occurred sporadically in water distribution systems (WDS) for as long as they have been operated. Over time, multiple barriers against contamination have been engineered, most notably modern disinfection and filtration practices, and as a result the frequency and severity of accidental contamination events has decreased. In the security environment created in the aftermath of the attacks on September 11, 2001 new energy has been directed towards emergency planning and preparedness for accidental and deliberate contamination events. One important aspect of a utility's response during contamination events is system water quality monitoring. In a WDS contamination scenario, water quality monitoring results would provide crucial information such as confirmation of a contamination event, the nature of the event, and the extent of contamination, all

of which are critical when rapidly planning and executing a mitigating response. The problem of how best to perform monitoring activities under such circumstances is currently an active area of research.

The difficulties associated with monitoring activities are compounded by water distribution network topology and dynamics.

> Water distribution networks are large scale engineered systems serving populations situated in large service areas. Such networks by their nature offer many potential uncontrolled entry points for contamination. Once contaminates have entered a system, transport can occur over multiple flow pathways dictated by system hydraulics, which in turn are driven by stochastic user demands and dynamic system operations. Furthermore, the detection of a malicious source or accidental contamination in such networks involves the real-time monitoring of sensors strategically placed in an existing system [56].

The implementation of such a sensor network would require the solution of many technical problems as well; most notably, the design of the sensors themselves, the integration of sensor arrays with SCADA systems and algorithms for analyzing the time series of measurements generated, and the design of sensor arrays (placement and sampling frequencies) within the WDS. Complex system topography, dynamics, and modeling uncertainties make locating sensors a difficult problem. Modeling uncertainties specific to the monitoring design problem include the probability of attack at any given location and time and variability in population densities [among others] [11].

Inverse problems are difficult to solve for several reasons including rank deficiency and ill-posedness, both of which can cause solution uniqueness and stability issues, and computational tractability. Some of these issues are a function of monitoring design which dictates the amount and quality of information available to formulate and solve the problem. Model and measurement errors can be an important factor contributing to identification uncertainties. Generally, source inversion problems are under-determined, because the data is sparse and there are more unknown variables than observations. This leads to an inverse problem description where there are an infinite number of solutions and inherent non-uniqueness. Thus, the monitoring design and source identification problems are coupled with one another not only because the source identification problem requires data from a sensor network for solution, but more importantly, because the structure of the source identification solution itself and the errors associated with it are a function of monitoring design.

This chapter is organized as follows. First, the relevant monitoring design literature is reviewed. Then, the linear input/output (I/O) water quality model developed in Chapter 2 is reintroduced along with a brief review of elementary theory of discrete linear inverse problems. Next, methods for optimal inverse experiment design are introduced, followed by a discussion of the proposed

methodology and solution procedure. Finally, applications of the methodology are prepared to better understand the solutions generated and illustrate their usefulness.

4.2 Literature Review

The water security dimension of the monitoring network design problem is relatively new. This is understandable considering that work began in earnest in 2002. The Water Initiative at Sandia National Laboratory and the Water Security Group at the U.S. Environmental Protection Agency have been most active in this area of research. The current literature reflects how the monitoring network design problem has been addressed by separating it into two distinct yet related problems — the source detection and source identification problems. The source detection problem is concerned with locating an array of sensors in a network capable of detecting the presence of a contaminant and raising a warning if an event occurs. The source identification problem is concerned with identifying the location of a contaminant source once an event has occurred using the data provided by the monitoring sensors. These problems are closely related to monitoring design research conducted in groundwater contexts [31, 38].

4.2.1 Source Detection Problem

Recent work on the source detection problem has centered on designing sensor networks for the protection of public health. As such, the monitoring design objectives frequently expressed include minimizing time to detection, lethal exposures, spatial extent of contamination, and others.

Berry *et al.* [7] formulated a source detection problem whose objective was to minimize the fraction of population at risk. The formulation, however, was based on a steady-state hydraulic simulation of the WDS and failed to capture the dynamic transport characteristics of the system. The problem was cast as an integer programming problem for efficient solution. Again using integer programming, Berry *et al.* [8] presented a successful reformulation of the problem taking hydraulic dynamics into account.

Complex system topography, dynamics, and modeling uncertainties make locating the sensors a non-trivial problem. Carr *et al.* [11] develop a solution to the source detection problem that takes into account uncertainty in system hydraulics and contaminate concentrations. The specific modeling uncertainties identified include the probability of attack at any given location in the system and variability in population densities and water demands. The problem was formulated as a mixed integer programming problem.

The source detection problem is also multi-objective in nature. Watson *et al.* [60] perform a multi-objective analysis of the problem and conclude that a robust sensor array must satisfy

multiple, disparate design objectives. Specific objectives identified included minimizing exposure, time of detection, volume consumed, failed detections, false detections, extent of contamination, and the number of sensors as a surrogate for construction and operational costs.

Branch and bound procedures are generally employed to solve the source detection problem owing to the integer and mixed integer problems formulated. Several studies, however, have used search heuristics to solve the problem. Ostfeld and Salomon [41, 42] solved the problem under dynamic hydraulic and water quality conditions, using genetic algorithms. Uber *et al.* [56] cast the problem as a set coverage formulation and solved it using greedy heuristic methods.

4.2.2 Source Identification Problem

Recent work on the source identification problem has focused on the computational efficiency of the solution procedure. Waanders *et al.* [57] formulated the source identification problem as a non-linear programming problem. The problem is solved by coupling EPANET for dynamic water quality simulation with the DAKOTA optimization library. This approach, however, was found to be computationally inefficient. Laird *et al.* [28] present a solution to the problem that can identify the location and source history. They have developed an origin tracking algorithm to formulate an efficient solution procedure where the ODEs describing water quality transport are embedded within the non-linear programming problem.

Separating the source detection and identification problems forces the assumption of a monitoring network configuration to formulate and solve the source identification problem. The accuracy of the solution, however, is a function of the data provided by the aforementioned network. Thus, despite being very different, the source detection and identification problems are coupled with one another. Little work, however, has been performed on how to represent the monitoring design objectives associated with the source identification problem.

4.3 Methodology

The objective of the source identification problem is to identify the location where contamination is entering the WDS using data from a monitoring network. Thus, the intrinsic state of the system is inferred from external observations. Problems of this type are categorized as an inverse problems and are solved using various techniques dependent on their structure. The approach taken here uses linear system dynamics to describe the water quality transport behavior in WDSs, and hence, the source identification problem can be described as a discrete linear inverse problem where the model parameters being identified describe the contamination source location and release history. The proposed methodology is described in greater detail in the sections that follow.

4.3.1 Input/Output Water Quality Model

It has been shown that the response of the WDS system to a contaminant mass injection can be described using linear dynamical systems theory under the following assumptions [9]:

1. that the contaminant is conservative, or reactive in the first-order; and
2. that the observed reaction rate constant is independent of the dosage applied.

Previous researchers have used linear superposition to describe historical exposure reconstruction [25], and booster disinfection scheduling [9] and location [55]. The idea was developed further into a general linear input/output model by [65, 51, 46] and used as the basis of booster disinfection design, [46] automatic system calibration [65, 51], and intelligent adaptive control [59]. The general linear I/O model proposed by Zierolf [65] and developed further by Shang [51] differs from the one proposed by Propato [46] in that it computes system response in reverse time, though the resulting descriptions are equivalent [46].

When the assumptions for linearity are enforced and the network hydraulic dynamics are known the contaminant concentration $c_i(t)$ at node i can be expressed as the linear summation of responses to individual dosages $u_j(k)$ at node j having occurred at previous time steps. Using the notation of [46, 45] this relationship can be more formally stated as,

$$c_i(t) = \sum_{j=1}^{n} \sum_{k=1}^{t} \theta_{ij}^{k}(t) u_j(k), \tag{4.1}$$

where the summation extends over all potential contamination sources $j = 1, \ldots, n$ and over the discretized injection time steps $k = 1, \ldots, t$. Taking the sampling time step $\Delta t = 1$, the response coefficient $\theta_{ij}^{k}(t)$ is the concentration a node i at time t resulting from a contaminant injection occurring at node j between time $t = k$ and $t = k + 1$ with all other injections being equal to zero. The response coefficients $\theta_{ij}^{k}(t)$ take a value greater then or equal to zero for all i, j, k, t.

Equation (4.1) provides an approach for expressing the source identification problem mathematically as a linear system of equations. Considering a set of n_s monitoring sensors and monitoring time window $[0, T]$, in compact matrix notation the linear I/O model becomes,

$$\mathbf{Am} = \mathbf{c}, \tag{4.2}$$

where, $\mathbf{A}$ is a $n_s T \times nT$ matrix of response coefficients, $\mathbf{m}$ is a nT vector of contaminant mass injections, and $\mathbf{c}$ is a $n_s T$ vector of contaminant concentrations. Given the connectivity and transport characteristics typical of water distribution networks, the response matrix $\mathbf{A}$ is very sparse. Generally, monitoring observations are relatively difficult and expensive to gather. Thus, for all practical

problems $\mathbf{A}$ is a rectangular system with $n_s < n$, and therefore, the system of equations (4.2) is underdetermined with the unknowns outnumbering the equations.

Given a system response matrix $\mathbf{A}$ and nodal contaminant concentrations $\mathbf{c}$, solving Eqn. (4.2) for the vector of unknown contaminant injections $\mathbf{m}$ is a canonical discrete linear inverse problem. This formulation is valid when the assumptions for linearity are enforced, otherwise, a more general non-linear inverse problem formulation would be required. The linear formulation, however, is convenient as discrete linear inverse problems are well understood and powerful tools exist for their analysis and solution.

4.3.2 Discrete Linear Inverse Problems

In this section a method for conditioning inverse experiments is introduced. The method is based on an eigenvalue positivity concept that improves inverse problem conditioning by maximizing a quality measure Θ computed using the eigenvalue spectra of the forward model matrix $\mathbf{A}$. The quality measure applied here was selected among many such metrics found in the literature, because it is computationally efficient to evaluate — an important consideration given the search procedure employed and problem sizes explored in this work.

Given a system where an explicit linear mapping exists between the model space $\mathbf{m}$ and the data space $\mathbf{c}$, this mapping is captured in the forward model matrix $\mathbf{A}$, and takes the form of a linear system of equations (4.2). This is called the "forward problem" where a data vector $\mathbf{c}$ can be generated given a model vector $\mathbf{m}$. The inverse problem is essentially the reverse of the forward problem: given a data vector $\mathbf{c}$ estimate the model vector $\mathbf{m}$ that produced it. A straightforward approach for solving a discrete linear inverse problem is to find the model $\mathbf{m}$ that minimizes the misfit between the data set and the predicted values from the forward model. Typically, some measure of distance is used as the misfit function[1];

$$\|\mathbf{c} - \mathbf{Am}\|_2 = \left[\sum_{i=1}^{n_s T} (c_i(t) - (\mathbf{Am})_i(t))^2\right]^{1/2}. \tag{4.3}$$

Other norms can be used to formulate the inverse problem as well, though the L_2 solution is most common and is referred to as the "least squares solution." Solution of the the least squares problem is possible via the normal equations,

$$\mathbf{m}_{L_2} = (\mathbf{A}'\mathbf{A})^{-1}\mathbf{A}'\mathbf{c}. \tag{4.4}$$

Equation (4.4) is the direct solution of Eqn. (4.3)[2].

[1] Here the inverse problem has been formulated using the L_2 norm.

[2] Equation (4.4) can be derived either one of two ways. Applying elementary calculus Eqn. (4.3) is expanded and

The solution of Eqn. (4.4) is guaranteed to be the unique minimum of the least squares problem when $(\mathbf{A}'\mathbf{A})^{-1}$ exists (*i.e.* $\mathbf{A}$ is full column rank, $rank(\mathbf{A}) = nT$). Therefore, when $n_s \ll n$ the problem will always be rank-deficient. When $\mathbf{A}$ is rank-deficient, solution of the inverse problem can become difficult or sometimes impossible due to non-uniqueness and stability issues.

Factoring techniques such as singular value decomposition (SVD) or eigenvalue decomposition can be employed *a priori* to analyze the solution stability, resolution, variance, and uniqueness characteristics of the inverse problem. Solving the standard eigenvalue decomposition problem on the real symmetric matrix $\mathbf{A}'\mathbf{A}$ of size $nT \times nT$;

$$\mathbf{A}'\mathbf{A}\mathbf{V} = \mathbf{V}\mathbf{\Lambda}. \tag{4.5}$$

Yielding, $\mathbf{V}$, size $nT \times nT$, the matrix of eigenvectors and $\mathbf{\Lambda}$, size $nT \times nT$ the diagonal matrix of eigenvalues, $\{\lambda_i : \forall i = 1, \ldots, nT\}$. Since $\mathbf{A}$ is real all eigenvalues in Λ are ≥ 0. When $\mathbf{A}$ is ill-conditioned or singular, however, some eigenvalues can be very small or equal to zero.

The inverse of $\mathbf{A}'\mathbf{A}$ can be computed via the eigenvalue decomposition as follows,

$$(\mathbf{A}'\mathbf{A})^{-1} = \mathbf{V}\mathbf{\Lambda}^{-1}\mathbf{V}^{-1}. \tag{4.6}$$

The inverse exists, however, only when $\mathbf{V}$ and Λ are invertible[3]. When $\mathbf{A}$ is not full column rank (*i.e.* singular) other methods such as singular value decomposition (SVD) can be applied to calculate the Moore-Penrose psudo-inverse, making the problem tractable [3]. Nevertheless, analyzing the eigenvalue spectrum of $\mathbf{A}'\mathbf{A}$ can provide important information regarding the conditioning of the underlying inverse problem.

For example, when $\mathbf{A}'\mathbf{A}$ is near singular and small eigenvalues are present, the inverse solution may become unstable as measurement errors become amplified within the solution. This can be seen clearly by substituting Eqn. (4.6) into Eqn. (4.4) and performing a column wise expansion of the matrix multiplications;

$$\mathbf{m} = (\mathbf{V}\mathbf{\Lambda}^{-1}\mathbf{V}^{-1})\mathbf{A}'\mathbf{c} = \sum_{i=1}^{nT} \frac{(\mathbf{A}_{.,i})'\mathbf{c}(\mathbf{V}_{.,i})^{-1}}{\lambda_i}\mathbf{V}_{.,i}. \tag{4.7}$$

Any measurement errors in $\mathbf{c}$ with a component parallel to the eigenvectors $V_{.,i}$ becomes amplified by a factor of $1/\lambda_i$ and incorporated into the solution. Further, from Eqn. (4.7) it can be seen that the eigenvectors $V_{.,i}$, form the basis — a linearly independent set of vectors that spans the model

the partial derivative taken with respect to the model is set equal to zero [37]. Applying basic vector space theory from linear algebra, Eqn. (4.4) can also be derived by setting the columns of $\mathbf{A}$ orthogonal to $\mathbf{A}\mathbf{m} - \mathbf{c}$ (i.e. multiplying by $\mathbf{A}'$) [3].

[3] *i.e.* when the eigenvectors $\mathbf{V}$ are linearly independent and the eigenvalues $\lambda_i > 0, \ \forall\, i$.

space — of the solution $\mathbf{m}$. Put another way, the model $\mathbf{m}$ is expressed solely as a summation of scaled eigenvectors $V_{.,i}$. The extent that the eigenvector span the model space determines the resolution of the resulting solution — the degree to which details in the model can be resolved.

Assuming the the concentration measurements $\mathbf{c}$ are random due to measurement errors and have a multivariate normal distribution, then the model $\mathbf{m}$ is also multivariate normal with covariance. The measurement errors in the data vector $\mathbf{c}$ are mapped to the model via a general linear operator such as the matrix $\mathbf{G}$ given by [3];

$$cov(\mathbf{Gc}) = \mathbf{G}cov(\mathbf{c})\mathbf{G}'. \tag{4.8}$$

Applying this mapping to the right hand side of Eqn. (4.4), with $\mathbf{G} = \mathbf{A}'\mathbf{A}^{-1}\mathbf{A}'$, with independent, normal, and identical data measurement errors yields;

$$cov(\mathbf{m}) = \sigma_c^2(\mathbf{A}'\mathbf{A})^{-1}. \tag{4.9}$$

Thus, the model covariance matrix is proportional to $(\mathbf{A}'\mathbf{A})^{-1}$. In the absence of model parameter correlation the covariance matrix is diagonal; this, however, is rarely the case. As illustrated in Eqn. (4.7), the elements of model $\mathbf{m}$ are composed of linear combinations of the elements in data vector $\mathbf{c}$, and thus, the correlation present in the model should hardly be surprising [3].

4.3.3 Optimal Inverse Experiment Design

The effect of the eigenvalue spectrum on solution stability, resolution, and uniqueness makes it an important indicator of forward model matrix conditioning. Further, the condition of the forward model matrix is solely a function of the monitoring design (sampling location, frequency, and duration). This suggests a method for the optimal design of monitoring sensor networks, one where the objective is to maximize the eigenvalues of $\mathbf{A}'\mathbf{A}$ over all possible monitoring designs. Maximizing eigenvalues in this manner has the potential to reduce the propagation of noise, improve resolution, and reduce the degrees of freedom present in the null space by improving the conditioning of the resulting inverse problem.

Discrete linear inverse theory has been applied extensively in the geophysical problem domain and several functions for quantifying eigenvalue positivity exist in the literature related to optimal inverse experiment design. These functions were examined systematically by Curtis in the course of analyzing cross-borehole tomographic experiments [17]. The eigenvalue positivity functions identified by Curtis are displayed in Table 4.1.

The performance of a potential monitoring design could be quantified by one of the five proposed

Table 4.1: Eigenvalue positivity functions for designing geophysical inverse experiments compiled by Curtis [17]. Note that Curtis assumes that the eigenvalues are sorted greatest to least.

Function	*Comment*	*Citation*
$\Theta_1 = \sum_{i=1}^{N} \frac{\lambda_i}{\lambda_1}$	assuming $\mathbf{A}$ is real, then $\lambda_i > 0 \; \forall i$	Curtis and Sneider (1997, 1999)
$\Theta_2 = \log \lambda_k$	for a predefined index k	Barth and Wunsch (1990), and Smith *et al.* (1992)
$\Theta_3 = k$	such that $\lambda_k > \delta$ for a predefined δ	Curtis and Sneider (1999)
$\Theta_4 = \sum_{i=1}^{N} \frac{-1}{\lambda_1 + \delta}$		Maurer and Boerner (1998)
$\Theta_5 = \sum_{i=1}^{N} \gamma_i$	where $\gamma_i = \begin{cases} \log \lambda_i & \text{if } \lambda_i \geq \delta \\ \textit{penalty} & \text{if } \lambda_i < \delta \end{cases}$	Wald (1943), Box and Lucas (1959), John and Draper (1975), Kijko (1977), Rabinowitz and Steinberg (1990), and Steinberg *et al.* (1995)

positivity functions displayed in Table 4.1. Each function's effect on the *posteriori* model uncertainties is different because of their differing sensitivities to the eigenvalue spectrum [17]. Thus, the design problem is one of maximizing the best suited objective function Θ subject to constraints on the economically feasible number of sensors.

A monitoring sensor network design formulation based on the eigenvalue positivity concept is proposed. Given the problem size and solution procedures examined in this work, computational cost associated with evaluating the positivity metric became the most important factor in selecting the objective function. For these reasons, positivity metric Θ_1, developed by Curtis *et al.* [18, 17], was selected as the objective function. Positivity metric Θ_1 has the following form,

$$\Theta_1 = \frac{1}{nT\lambda_{max}} \sum_{i=1}^{nT} \lambda_i. \tag{4.10}$$

Where, λ_i is the eigenvalue of column i and λ_{max} is the largest eigenvalue of the nT eigenvalues in $\mathbf{\Lambda}$. Thus, Eqn. (4.10) is the sum of the normalized eigenvalues and its magnitude is proportional to the area under the eigenvalue curve. Normalizing the sum of eigenvalues by the term $1/nT\lambda_{max}$ makes Eqn. (4.10) sensitive to relative rather than absolute differences between the eigenvalue spectrums of the monitoring designs being evaluated and ensures that $0 < \Theta_1 \leq 1$ [17].

The optimal inverse experiment design problem introduced here selects a set of monitoring sensor locations that maximizes the normalized area under the eigenvalue spectra curve. Binary variables represent the decision to locate a sensor, or not, at a particular node in the network. The number of sensors placed in the network is related to the fixed cost of building and variable costs associated with operating the monitoring design. Constraints are written to ensure that the maximum allowable

number of sensors is not exceeded.

Mathematically, the optimal inverse experimental design problem takes the form,

$$\max_{\delta} \Theta_1(\delta). \tag{4.11}$$

Where the objective is to maximize Θ_1 over the vector of potential sensor locations δ. One binary variable δ_i is introduced for each node in the network denoting whether a sensor is constructed at node i, $\delta_i = 1$ or not $\delta_i = 0$. The maximization is subject to a constraint on the number of sensors allowed in the monitoring network;

$$\begin{gathered} \sum_{i=1}^{n_p} \delta_i \leq n_s, \\ \delta_i \in \{0,1\},\ \forall\, i = 1, \ldots n_p, \end{gathered} \tag{4.12}$$

where, n_p is the number of potential monitoring sensor locations, and n_s is the maximum number of sensors allowed in the network. Further, the variables δ_i are restricted to binary values. Taken together Eqns. (4.11) and (4.12) compose the optimal inverse experimental design formulation.

Preliminary studies of the proposed monitoring design problem indicated several tradeoffs worthy of exploration within the design decision space, the most noteworthy being the feasibility of a sensor network design and the condition of the inversion problem posed from the data it generates. Due to the lack of availability, high cost, and sensor response characteristics associated with current technologies, the resulting source identification problems are likely to be highly underdetermined and difficult to solve. Thus, accurate source identification problem solutions may be neither technically nor economically feasible, especially when a rapid response is required to achieve public health protection benefits.

4.3.4 Solution Procedure

The optimization model proposed was solved using a two step process. First, the sensitivity coefficients were calculated. A perturbation approach was employed for its simplicity, though a sensitivity model or adjoint system could also be used. Then as the optimization model was solved the sensitivity coefficient matrix calculated previously was used to evaluate the objective function (see Figure 4.1). The total coefficient matrix calculated *a priori* contains sensitivities for all possible source injection, monitoring location, and sample combinations. The coefficient matrix used to calculate the objective function for any particular monitoring design is essentially a filtered version of the total sensitivity coefficient matrix that only conveys the data intercepted by the subset of the

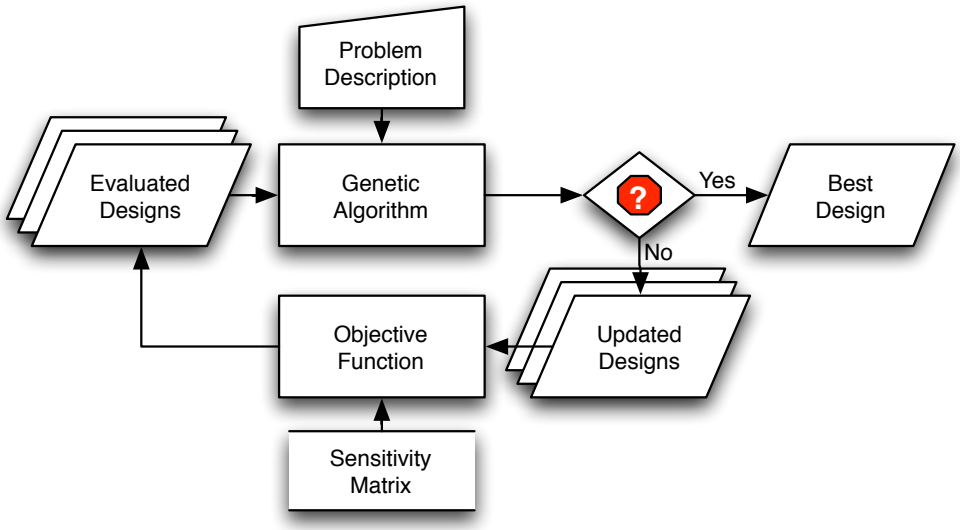

Figure 4.1: Solution procedure flow chart for serial algorithm.

monitoring sensor locations it specifies.

The objective function Θ_1 proposed for the monitoring design optimization problem is highly non-linear with respect to the decision variables, *i.e.* the monitoring designs. Genetic algorithms (GA) are stochastic population based search techniques that mimic Darwinian evolution to identify solutions of non-linear combinatorial problems like that developed here. An GA population is composed of individuals; encoded within each individual is a potential solution to the search problem. Individuals are evaluated and assigned a fitness value, via an objective function, proportional to its suitability as a solution to the problem. A canonical GA iteratively produces new populations of individuals through the application of stochastic evolutionary operators (selection, crossover, and mutation). This synchronized generational process continues until some stopping criteria is satisfied, ultimately producing many high performing individuals.

Genetic Algorithms are inherently parallel, as the evaluation of populations and application of evolutionary operations can in many cases be performed concurrently. One global population is sometimes referred to as "panmictic." A panmictic GA can be parallelized by computing the fitness values of the individuals in the population concurrently. This approach, referred to as global parallelism, decreases algorithm run times in proportion to the number of concurrent evaluations being performed. One advantage of maintaining a panmictic population compared to other parallelization strategies, is that it simplifies the implementation of the GA.

Encoding

The binary formulation of the monitoring design problem presented above requires a constraint Eqn. (4.12) on the number of allowable monitoring sensors n_s. Constraint handling in GAs is awkward compared to classical optimization techniques. Several techniques exist, however, for handling them: 1) a penalty function can be used to modify the objective space making individuals that violate the constraint less attractive from a fitness perspective; 2) feasibility based selection can be used to reduce selection pressure on infeasible individuals making them less likely to propagate; 3) a repair operator can be used to modify infeasible individuals such that they satisfy the constraint making them feasible; or 4) reformulate the problem to eliminate the constraint or handle it implicitly.

Decision variable encoding is an important factor when using GAs that is not straightforward and has many ramifications for the solution of the problem. While tuning the GA for the monitoring design problem several of these constraint handling techniques were explored. Based on these solution experiences, the problem was reformulated to handle the constraint implicitly. This was accomplished by modifying the encoding of the decision variables within each individual of the population. The monitoring design presented above uses an array of binary variables of length n_p to represent a monitoring design. The new encoding represents the decision variables as an array of integers of length n_s, each taking a node index value to represent the nodes selected for the monitoring design. After reformulation, the monitoring design problem is similar to a k-subset selection problem.

The monitoring design problem reformulation can be stated mathematically as,

$$\underset{\mathcal{S}\subseteq\mathcal{N}}{max}\ \Theta_1(\mathbf{s}). \tag{4.13}$$

The problem becomes one of selecting the subset of nodes that maximize the objective function. The subset $\mathcal{S}$,

$$\mathcal{S} = \{s_i | i = 1, \ldots, n_s; s_i \in \mathcal{N}, \} \tag{4.14}$$

are the indices of the best nodes for siting monitoring sensors. The number of members in $\mathcal{S}$ is equal to n_s the maximum allowable number of monitoring sensors. The members of $\mathcal{S}$ are members of the superset $\mathcal{N}$, the set of all potential monitoring sensor locations with $|\mathcal{N}| = n_p$. Thus, the constraint Eqn. (4.12) can be eliminated by handling it implicitly within the decision variable encoding.

Fitness Evaluation

The eigenvalue positivity functions displayed in Table 4.1 vary in their computational complexity; however, all require a full eigenvalue decomposition of $\mathbf{A}'\mathbf{A}$ except Θ_1 [17]. Modern numerical techniques for performing eigenvalue decompositions are efficient. Nevertheless, their computational expense makes them impractical for the large matrices and population sizes evaluated herein. A full eigenvalue decomposition can be avoided because the diagonal of a real square matrix is an invariant property of a similarity transform, and thus,

$$\sum_{i=1}^{nT} \lambda_i = trace(\mathbf{A}'\mathbf{A}). \tag{4.15}$$

Square matrices have the useful property that the spectral norm — the largest eigenvalue λ_{max} — is equal to the 2-norm;

$$\lambda_{max} = \|\mathbf{A}'\mathbf{A}\|_2. \tag{4.16}$$

The 2-norm of $\mathbf{A}'\mathbf{A}$, and thus λ_{max} can be efficiently and accurately estimated using a power method [26].

The substitution made in Eqn. (4.15) can be taken further by computing the Frobenius Norm of $\mathbf{A}$ in place of $trace(\mathbf{A}'\mathbf{A})$ accordingly;

$$trace(\mathbf{A}'\mathbf{A}) = \sum_i \sum_j |a_{ij}|^2 = \|\mathbf{A}\|_F^2. \tag{4.17}$$

From the 2-norm of a real matrix and its transpose it follows that $\|\mathbf{A}\| = \|\mathbf{A}'\|$; further the 2-norm is a sub-multiplicative norm and thus, $\|\mathbf{AB}\| \leq \|\mathbf{A}\|\|\mathbf{B}\|$. Thus, the spectral norm can be calculated directly from the norm estimate of $\mathbf{A}$ as follows,

$$\sqrt{\lambda_{max}} = \|\mathbf{A}\|_2. \tag{4.18}$$

Thus, the computational expense of calculating Θ_1 can be reduced further by eliminating the costly matrix matrix multiplication required to calculate $\mathbf{A}'\mathbf{A}$. In this form the objective function was relatively efficient to evaluate even for large matrices.

Having made evaluation of the objective function as efficient as possible, the overall computational effort associated with the proposed methods should be noted. The calculation of the forward model matrix by the perturbation approach used here can be time consuming as it requires the performance of many water quality simulations. The forward model matrix can become very large,

even for small to medium sized networks like the ones studied here, because of the temporal dimension of the source identification problem. The forward model matrix, however, is very sparse due to the connectivity typical of WDS topologies. The matrix computations discussed, especially eigenvalue decomposition, can be intensive for large sparse matrices. Therefore, the problem formulation developed here is viewed as a model suited to the investigation of the source identification related monitoring requirements, but one that may not scale to large network models with long monitoring time windows.

An unintended side effect of the monitoring design problem reformulation presented above, Eqns. (4.13) and (4.14), is the possibility of duplicate node indexes occurring within an individual's encoded decision variables. This is undesirable from cost and performance perspectives in that it corresponds to a extra sensor being placed within the network that does not collect any new information. It was determined that the most effective way to handle duplicate indexes was to eliminate them and calculate the objective function for the resulting unique set of indexes. The objective function is implicitly penalized by removing the duplicate nodes, since nearly all n_s sensor designs perform better than any of the $n_s - 1$ sensor designs. Thus, individuals with duplicate node indexes tend to have lower fitness values, are less likely to propagate, and tend to be eliminated from within the population over successive generations when they happen to occur.

GA Operators and Settings

A standard generational genetic algorithm (GA) with elitism was used to solve the optimization model presented above [22]. Furthermore, the global parallelism strategy was employed to speed up fitness evaluations. One drawback associated with GAs is that they require fine tuning of many parameters that effect search effectiveness and performance. In the course of tuning the GA to the problem formulation the solution experiences indicated that crossover was disruptive in that a crossover of two individuals frequently yielded an offspring of lower fitness. A one point crossover operator was selected to reduce the disruption of the search associated with crossover processes. Also, the speed of convergence and population diversity metrics were observed as being a strong function of selection pressure. A binary tournament selection operator was utilized to moderate selection pressures and maintain adequate diversity in the population over the course of the search. A standard mutation operator set for a low probability of mutation was also used.

Implementation

A prototype software application was developed in Java and Matlab to set up and solve the monitoring sensor design problem. Matlab scripts were written to call the EPANET Programmer's Toolkit and run the hydraulic and water-quality simulations necessary to compute the response coefficient

matrix **A**. The simulation optimization framework developed previously in Chapter 3 was redeployed for solution of the design problem. Starting the framework within Matlab on a separate thread and passing it a reference to Matlab JVM class loader enabled the framework and Matlab objects to operate concurrently and to reference each other.

Calling Java from Matlab is a feature supported by The Mathworks. This implementation, however, relies on calling Matlab commands *from* Java, an unsupported operation that is possible with the API implemented in the `MatlabControl` Java object developed by Whitehouse [61]. Leveraging the `MatlabControl` object, the task pool in the original framework design was replaced by an object that interfaces with the Distributed Matlab Toolbox job manager to request computational resources, create jobs, add tasks to a job, submit the job for evaluation, and gather results. The Matlab job manager processes tasks using a batch strategy — each individual in the population considered a separate task, the population constituting a job, and a new job created each generation of the search. A callback function was used to convert between Java and Matlab data types as tasks are being added to a job. The tasks in a job are distributed between the Matlab workers registered with the job manager at the time of execution. Errors returned from the Matlab environment are encapsulated in Java Throwable objects and thrown up the call stack to facilitate robust error handling.

The resulting prototype software implementation was successfully used to set up and solve the monitoring design optimization problem. The author found the Matlab Distributed Toolbox easy to configure and use. One drawback, however, was the amount of overhead associated with running a job. Jobs requiring only a small amount of computational effort per task may not scale efficiently because of these overheads.

4.4 Case Studies

A method for designing monitoring sensor networks with a source identification objective has been proposed. In the sections that follow, two applications — one for a hypothetical network and the other for a realistic network — are presented. The "Hypothetical Network" application was used to validate the solutions generated using the proposed method against solutions generated using enumeration. In the "Realistic Network" application monitoring designs generated using different methods and the conditioning of the source identification inversion problems that resulted were evaluated.

4.4.1 Case Study I — Hypothetical Network

The Hypothetical Network shown in Figure 4.2 appears in the published literature. Laird *et al.* analyzed it to demonstrate a method for method for solving WDS source identification problems [28]. Monitoring design, however, was not part of their study; as such, the source identification problem was formulated with monitoring sensors placed at even nodes. Summarizing Laird's description, the network and its hydraulic solution are symmetric about the diagonal extending from nodes 1 to 25. Nodal demands occur at boundary nodes only with water supplied from a single reservoir located at node 26. Time delays due to transport ranged from 0.5 to 5 hours.

Experimental Design

The proposed monitoring design method was used to generate the tradeoff between the problem conditioning metric Θ_1 and the number and locations of monitoring stations. The tradeoff curve was generated using the constraint method[4]; thus, each point along the tradeoff curve was generated by solving the monitoring design problem. Then the results were validated using enumeration to identify the globally optimal monitoring designs for each of the points along the tradeoff curve.

For the purposes of calculating sensitivity coefficients, all nodes in the network were considered potential source locations and potential sensor locations $n = n_p = 26$. The duration of the simulation was 12 hours with the sampling time step $\Delta t = 5$ minutes. The monitoring window was set equal to 4 hours, and a 10,000 (mg/L) first-order reactive contaminant was simulated at each of the potential source locations. The injection time step had a 30 minute interval and thus $k = 1, \ldots, 24$ discretized injection periods.

To generate the tradeoff curve, the optimization problem was solved for monitoring designs with $n_s = 2, 4, 6, 8, 10, 12$ respectively. As stated previously GAs are a stochastic search procedure; thus, the results are a function of the seed used to generate the stream of random numbers used by the algorithm. Each problem configuration was solved for 30 random trials to gauge the sensitivity to the random seed value and robustness of the proposed GA search configuration.

Results and Discussion

The analysis in this section explores the effect of the number and location of monitoring sensors on the Θ_1 objective function Eqn. (4.13) — the area under the normalized eigenvalue spectra curve. The relationship between the design variables and the objective function is shown in Figure 4.3. The tradeoff curve makes the comparison of designs possible by making a metric of the inverse problems conditioning an explicit function of the number of sensors. This enables the direct and rational

[4]The constraint method was chosen for its simplicity; more efficient methods such as multi-objective GAs could also be used.

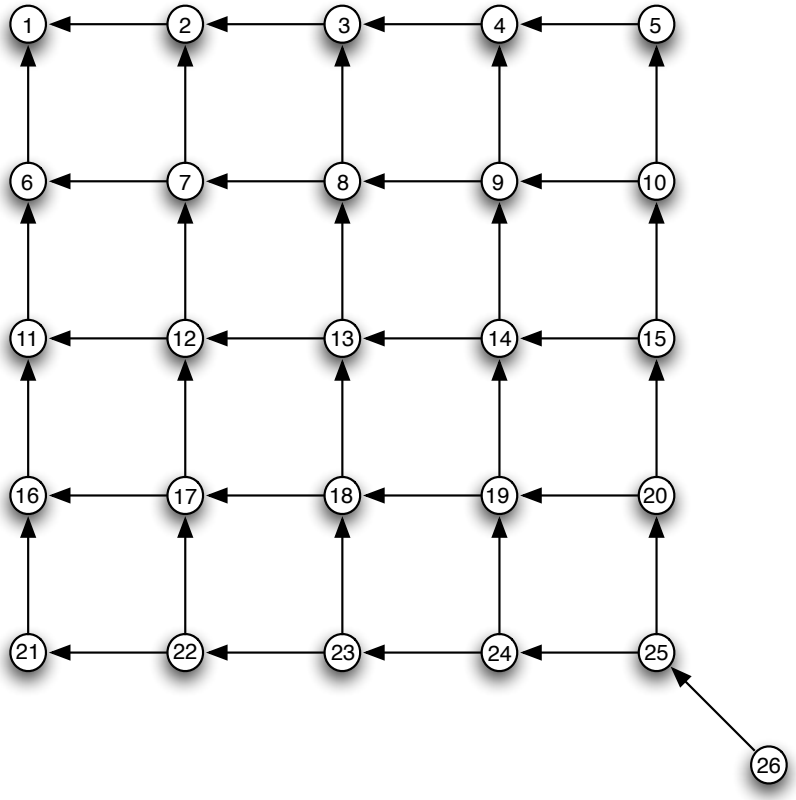

Figure 4.2: Schematic of Grid Network, Laird *et al.* [28].

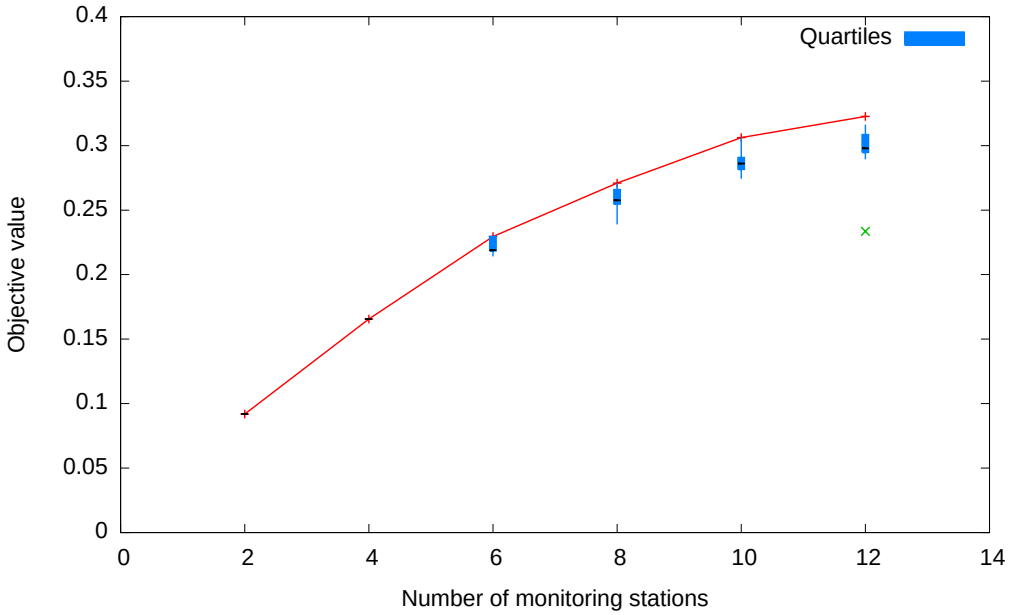

Figure 4.3: Tradeoff curve for Hypothetical Network example. The quartiles of the 30 random trials are indicated with blue bars, the designs generated by enumeration are indicated by "+", and the 12 sensor even node design is indicated with a "×."

comparison of different monitoring designs and gives additional insight into questions such as, "How many sensors are necessary for source identification in the network?" or "Given our sensor budget where should sensors be placed?"

The tradeoff curve was generated by varying n_s from 2 to 12, with the number of sensors increased by 2 for each successive design. The points along the curve are an approximation of non-dominated set of solutions for the problem. Furthermore, the curve increases as a function of the number of monitoring sensors while the marginal improvement accompanying each additional sensor pair decreases, implying that there is a limit to the number of sensors that are useful to deploy within a network for the purposes of source identification.

The monitoring designs located along the pareto front are described in more detail in Table 4.2. As a basis of comparison the 12 sensor even node design is also included in the Table and in Figure 4.3. Examination of the results indicate that the even node sensor network was dominated by all designs with more than 6 sensors — only performing slightly better than the 6 sensor design.

Trends are present in the pattern of sensor locations selected as part of optimal designs. Following the designs from 2 to 12 nodes the monitoring designs reveal an optimal substructure (see Figure 4.4). A problem solution has an optimal substructure when an optimal solution of the problem contains optimal solutions to its sub-problems [15]. As a consequence, the problem may be amenable to

Table 4.2: Results obtained by GA search for Hypothetical Network example.

Num. Sensors	*Objective* Θ_1	*Monitoring Design*	*Trace*	*Norm Estimate*	*Rank*	*Num. Samples*
2	9.1843e-02	4, 16	7.8479e-05	4.1081e-06	64	192
4	1.6556e-01	4, 6, 16, 17	1.4197e-04	4.1227e-06	150	384
6	2.2956e-01	2, 4, 6, 9, 16, 17	2.0546e-04	4.3028e-06	182	576
8	2.7099e-01	2, 4, 6, 9, 15, 16, 17, 23	2.4270e-04	4.3058e-06	192	768
10	2.9691e-01	2, 3, 4, 6, 9, 11, 16, 17, 19, 23	2.8533e-04	4.6201e-06	200	960
12	3.1478e-01	2, 3, 4, 6, 11, 13, 14, 15, 16, 17, 23, 24	3.0807e-04	4.7052e-06	200	1152
Even	2.3358e-01	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24	3.2094e-04	6.6059e-06	200	1152

Table 4.3: Results obtained by enumeration for Hypothetical Network example.

Num. Sensors	*Objective* Θ_1	*Monitoring Design*	*Trace*	*Norm Estimate*	*Rank*	*Num. Samples*
2	9.1843e-02	4, 16	7.8479e-05	4.1081e-06	64	192
4	1.6556e-01	4, 6, 16, 17	1.4197e-04	4.1227e-06	150	384
6	2.2956e-01	2, 4, 6, 9, 16, 17	2.0546e-04	4.3028e-06	182	576
8	2.7099e-01	2, 4, 6, 9, 15, 16, 17, 23	2.4270e-04	4.3058e-06	192	768
10	3.0623e-01	2, 3, 4, 6, 9, 11, 15, 16, 17, 23	2.9428e-04	4.6201e-06	200	960
12	3.2261e-01	2, 3, 4, 6, 9, 11, 15, 16, 17, 19, 23, 24	3.1001e-04	4.6199e-06	200	1152

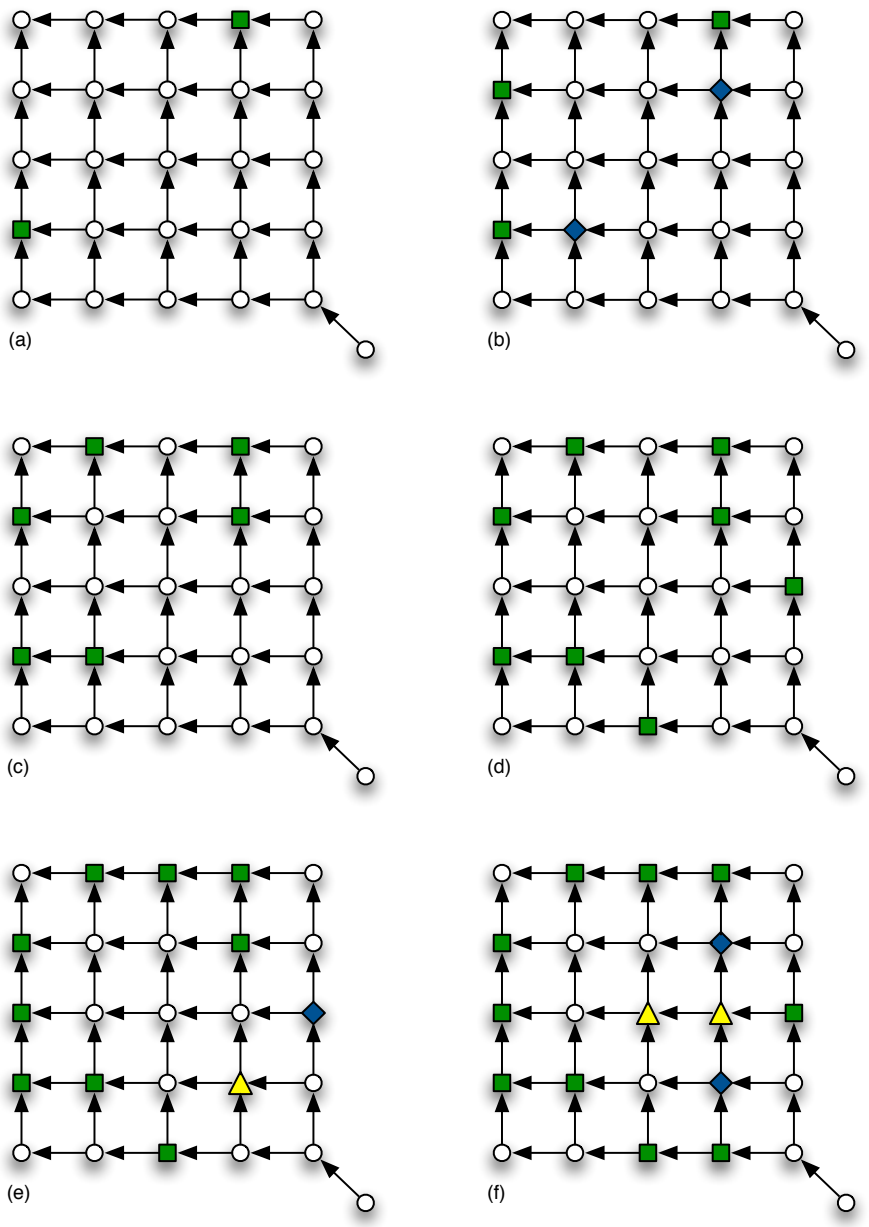

Figure 4.4: Monitoring sensor locations generated using enumeration and GA search for Hypothetical Network example. Sensor locations marked with squares were generated by both GA and enumeration. Sensor locations marked with diamonds appear in designs generated by enumeration. Sensor locations marked with triangles appear in GA generated design. Note that in sub-figure (b) alternate optima were identified.

solution by greedy algorithm search heuristics or other methods. The authors' solution experiences with the monitoring design problem, however, have suggested that solution methods which ignore correlation and covariance effects occurring between sensor locations produce sub-optimal designs.

There is no guarantee of optimality when using GAs. Comparing Tables 4.2 and 4.3, the GA was able to identify the optimal solution for every design except the 10 and 12 sensor designs. Indeed, the sub-optimal designs were very near the global optimal ones in objective space and only differ slightly in decision space. These results suggest an acceptable level of performance by the GA. Recalling that the monitoring design problem is combinatorial in nature, it is important to note the expansion of the decision space as the limit on the number of sensors was relaxed; the decision space for the 2 sensor designs contains 325 possible combinations. That number increased to $5.3e+06$ and $9.6e+06$ for the 10 and 12 sensor designs respectively. For realistic sized networks the decision space for the problem expands astronomically.

4.4.2 Case Study II — Realistic Network

In this section, the conditioning of the resulting source identification inversion problem is evaluated for different monitoring designs. Three different methods were used to generate the monitoring designs, the monitoring design formulation developed previously, a method based on nodal demands, and an *ad hoc* design. These alternate monitoring design methods will be explained in greater detail in the sections that follow. The application was prepared using the Realistic Network example illustrated in Figure 4.5. For each of the sampling designs, the resulting source identification problem was solved for an ensemble of contamination events. A statistical analysis was performed on the distribution of source identification solutions to better understand the performance of the designs.

The Realistic Network shown in Figure 4.5 comes bundled with the EPANET software distribution [49] as an example demonstrating source tracing. It has been modified slightly for modeling the monitoring and characterization problems studied here. The network has 97 nodes, 3 tanks, 1 reservoir, and 1 pump. The network hydraulics exhibit an atypical diurnal demand pattern and spatial clustering of high demand nodes.

Experimental Design

All nodes in the network were considered potential source locations and potential sensor locations $n = n_p = 97$. Sensitivity coefficients were calculated over a simulation duration of 24 hours with the sampling time step $\Delta t = 10$ minutes. The monitoring window was set equal to 4 hours, and a 10,000 (mg/L) conservative contaminant was simulated at each potential source location. The injection time step was equal to 10 minutes and thus $k = 1, \ldots, 144$ discretized injection periods. The travel times in the network were longer than the 4 hour monitoring time window.

Three different methods were used to create sensor networks. The first was an optimized monitoring design using the formulation developed previously in the chapter. The second was a heuristic method which assumes that a node's fitness as a sensor location is proportional to its total demand. The third was an *ad hoc* network studied recently in the literature [64]. Sensor networks with 3, 6, and 12 nodes were prepared using each of the design methods. For the duration of the chapter, these three design methods optimized, demand ranking, and *ad hoc* will be referred to as "Opt", "Dmd", and "Ad Hoc" respectively and the number of sensors in the networks will be referred to as "S3", "S6", and "S12" respectively.

The sensor locations for the Opt-S12, S6, and S3 designs are illustrated in Figure 4.5 and listed in Table 4.4[5]. The problems were solved using the GA solution procedure developed and verified previously in Case Study I. The decision space for the Opt design problems were on the order of $1.0e+14$ k-subsets, precluding total enumeration of the decision space to identify the globally optimal monitoring design with certainty. Therefore, the Opt results can not in general be considered globally optimal monitoring designs, but rather are the best designs generated using the proposed monitoring design procedure.

The sensor locations for the Dmd-S3, S6, and S12 designs are illustrated in Figure 4.6 and listed in Table4.4. The total demand ranking method was developed as a simple design heuristic for the purpose of generating competing sensor network designs. The method selects nodes as sensor locations by ranking them in descending order by the total volume demanded per day. The idea is that the transport pathways in the network connecting contamination sources to sensor locations tend to flow towards or through high demand nodes. As mentioned previously, the Realistic Network exhibits an atypical spatial distribution of demands. One consequence of this is the spatial clustering evident in the Dmd sensor locations relative to the other designs. Spatially clustered sensors are undesirable because they tend to have correlated input / output responses, and thus, supply redundant information.

The sensor locations for the Ad Hoc - S3, S6, and S12 designs are illustrated in Figure 4.7 and listed in Table4.4. The Ad Hoc sensor locations were selected using engineering judgement with the objective of achieving a uniform spatial distribution of sensors along key transport pathways within the network.

Focusing attention on Table 4.4, Θ_1 values were calculated for all of the designs to serve as a basis of comparison. Objective values for all the Opt designs were significantly greater, by an order of magnitude or more, compared to the Dmd and Ad Hoc designs. Further, the Dmd designs had larger objective values than the Ad Hoc designs though these differences were not as large. Thus,

[5]The designs generated using the Opt design method tended to exhibit an optimal substructure. Thus, the Opt-S12 design is an approximate superset containing the locations for the Opt-S6 and S3 designs within it.

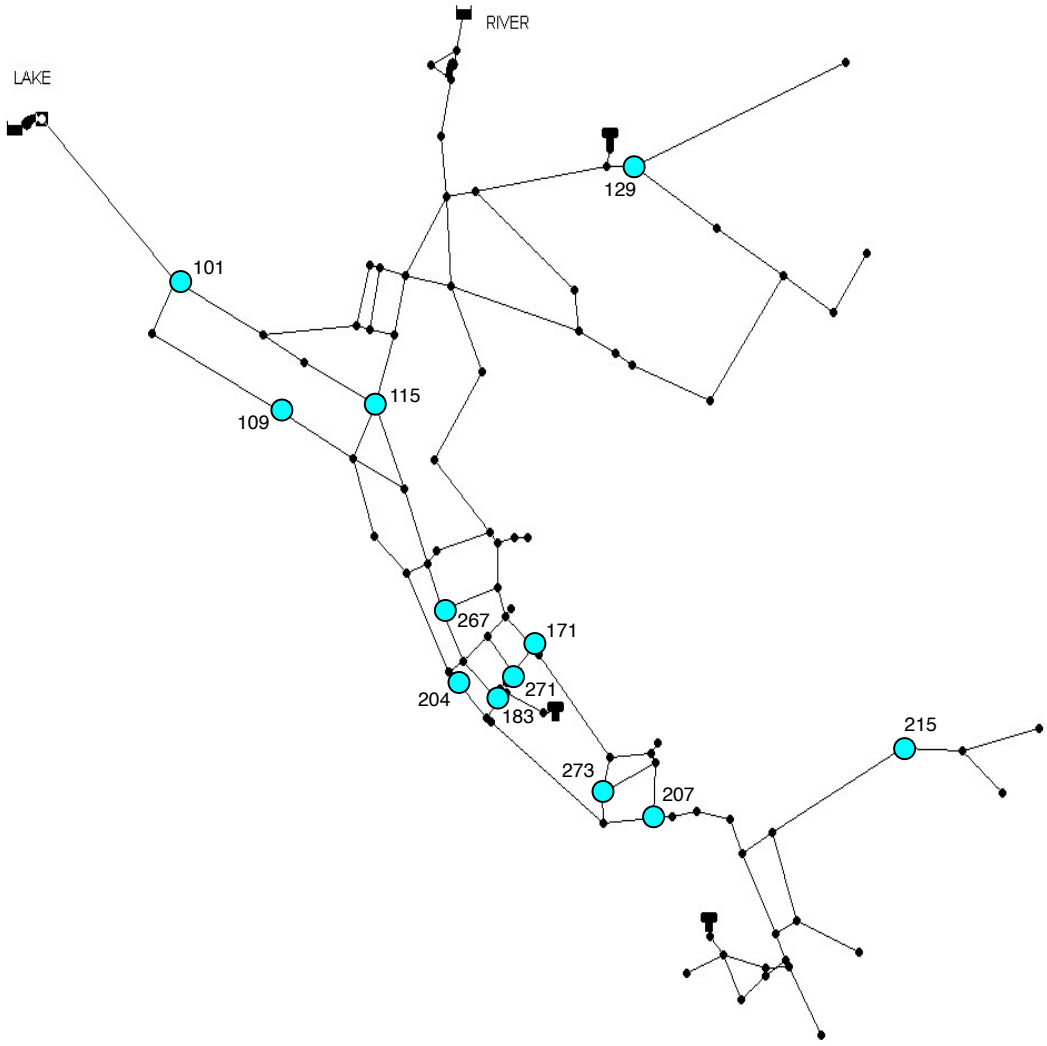

Figure 4.5: Sensor locations generated using the Opt design method.

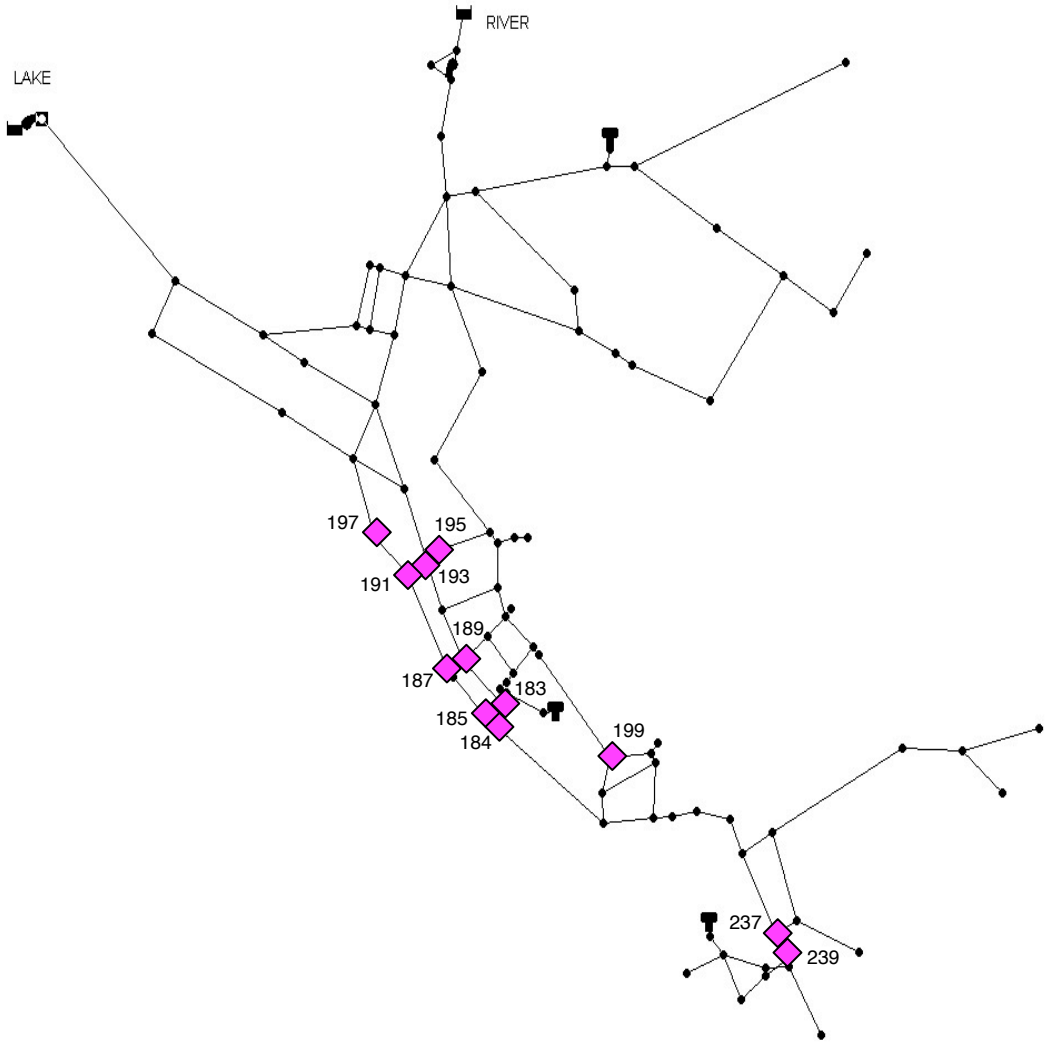

Figure 4.6: Sensor locations generated using the Dmd design method.

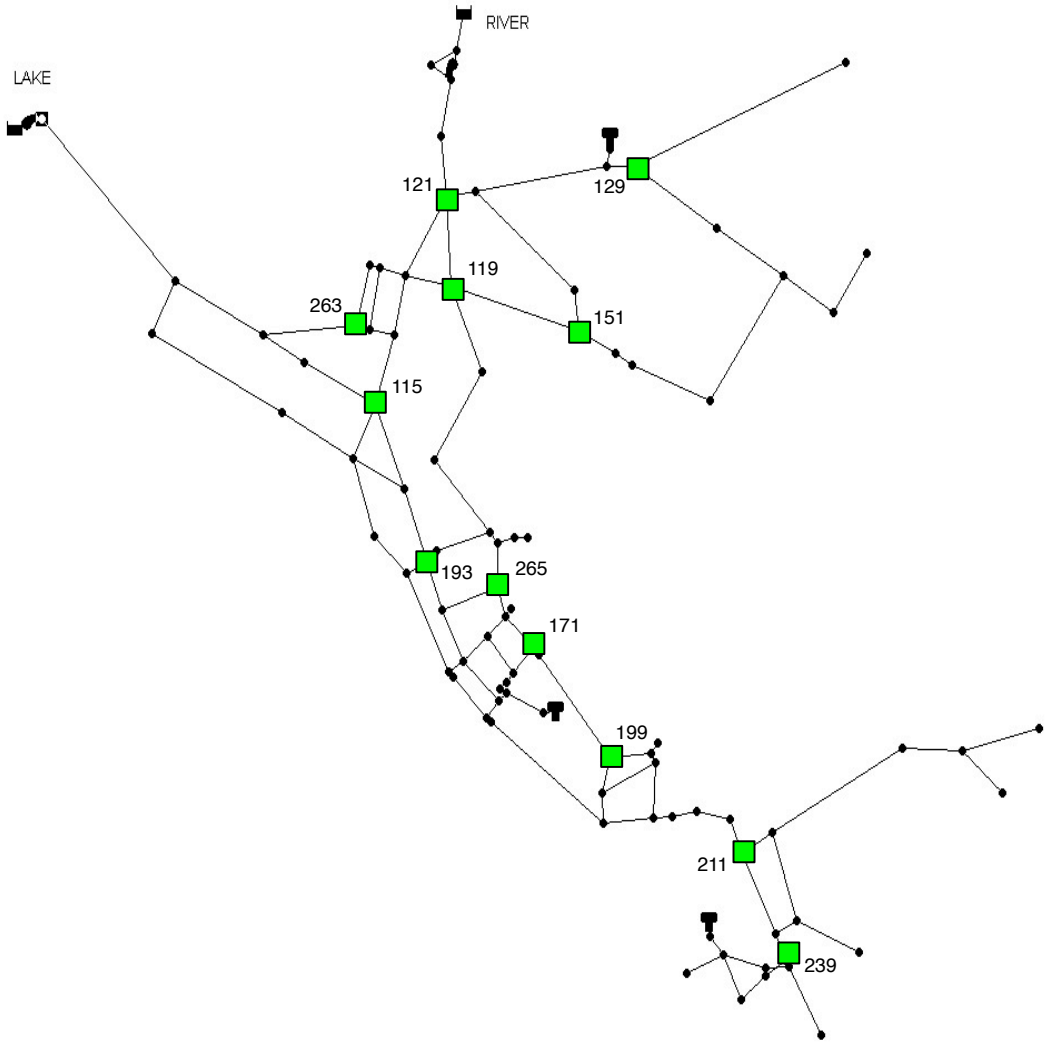

Figure 4.7: Ad Hoc sensor locations.

Table 4.4: Eigenvalue positivity objective for each of the nine designs being evaluated.

Design Factor	*Number Factor*	*Objective* Θ_1	*Monitoring Design*	*Trace*	*Norm Estimate*	*Rank*	*Num. Smpls*
Opt	S3	1.4759e-02	129, 215, 273	1.4193e-03	6.8849e-06	504	504
	S6	1.8699e-02	129, 183, 189, 215, 269, 273	1.8118e-03	6.9367e-06	1008	1008
	S12	2.2556e-02	101, 109, 115, 129, 171, 183, 204, 207, 215, 267, 271, 273	2.3968e-03	7.6072e-06	2015	2016
Dmd	S3	1.1825e-03	189, 187, 197	8.0129e-04	4.8514e-05	500	504
	S6	1.4860e-03	189, 187, 197, 199, 193, 195	1.9109e-03	9.2062e-05	1004	1008
	S12	2.8561e-03	189, 187, 197, 199, 193, 195, 191, 239, 237, 185, 184, 183	4.7468e-03	1.1898e-04	2011	2016
Ad Hoc	S3	9.7223e-04	119, 199, 239	1.6158e-03	1.1898e-04	504	504
	S6	1.0155e-03	115, 119, 121, 199, 239, 265	1.6878e-03	1.1898e-04	1004	1008
	S12	1.7411e-03	115, 119, 121, 129, 151, 171, 193, 199, 211, 239, 263, 265	2.8938e-03	1.1898e-04	2004	2016

the Opt, Dmd, and Ad Hoc designs were different from each other, by varying degrees, when viewed in objective space. The reader may ask, "How well do these differences in objective space translate into improvements in the solvability of the underlying inverse problem?"

To answer this question a Monte Carlo simulation based analysis was performed. A Monte Carlo analysis propagates the uncertainty associated with random inputs through a deterministic model, like the one used here, generating a distribution of outputs. The major sources of uncertainty in the source identification problem are the spatial and temporal distribution of nodal demands and the contamination source scenarios. This work, however, assumed that nodal demands were deterministic; thus, only one of the many potential demand realizations that are possible was being considered. Contamination source scenarios were modeled using random variables to represent the source location, and the contaminant injection start time, duration, and magnitude. The number of potential source realizations precluded enumerating them; thus, Latin Hypercube Sampling was used to generate a meaningful ensemble of realizations. The Monte Carlo analysis was performed using the resulting 200 uniformly distributed random source realizations. For each realization, the source identification inverse problem was solved using non-negative least squares (NNLS) and the objective and solution errors were recorded for post analysis.

The experimental setup for the post analysis consisted of a 3x3 factorial design[6]. The two

[6]A 3x3 factorial experimental design has 2 treatment factors with each factor having 3 distinct values.

Table 4.5: Raw data from the Monte Carlo simulation of 200 contaminant source realizations.

Design	*Number*	*Detect*	*Ident*	*Solution Error*		*Objective Error*	
Factor	*Factor*	*Freq*	*Freq*	*Mean*	*Std Dev*	*Mean*	*Std Dev*
Opt	S03	61	6	9.3298e-01	3.2953e-01	8.8927e-06	4.8070e-05
	S06	87	39	5.1864e-01	4.9491e-01	2.4169e-05	9.0034e-05
	S12	106	73	3.0294e-01	4.8239e-01	1.6746e-05	9.2775e-05
Dmd	S03	54	17	6.5907e-01	5.1287e-01	3.7973e-04	2.6305e-03
	S06	72	36	5.1749e-01	5.5983e-01	9.4691e-05	7.7264e-04
	S12	99	56	4.1039e-01	5.0636e-01	7.5763e-05	6.5981e-04
Ad Hoc	S03	98	11	9.3896e-01	3.8644e-01	1.4011e-05	7.2239e-05
	S06	102	48	5.5136e-01	5.4717e-01	1.3868e-05	7.1098e-05
	S12	118	78	3.4349e-01	5.0097e-01	9.7988e-06	5.2734e-05

treatment factors were the method used to generate the monitoring design and the number of sensors in the monitoring design and referred to as the "Design" and "Number" factors respectively. The 3 distinct values of the Design Factor were the Opt, Dmd, and Ad Hoc, and those for the Number Factor were S3, S6, and S12. Group statistics were calculated including the mean and the 95th percentile confidence interval about the mean based on the sample standard deviation. Group means and confidence intervals were compared to determine if the differences between them was significant given the variability present in the Monte Carlo simulation output.

Results and Discussion

The raw data from the Monte Carlo simulation is displayed in Table 4.5. For each of the nine designs in the experiment the detection and identification frequencies over the 200 source realizations were calculated. A source was considered detected if the concentration response for a source realization was $L_2(\mathbf{c} \geq 1.0e-06)$. The raw detection results indicate that the Ad Hoc designs had the best detection frequency, detecting approximately 50 - 60 percent of the source realizations, while the Dmd designs had the worst, with the Opt designs performing slightly better than the latter. A source was considered accurately identified if the solution error for a realization was $L_2(\mathbf{m}-\mathbf{m}') < 1.0e-06$. The raw identification results are also shown in Table 4.5, and again the raw results indicate that the Ad Hoc designs had the best identification frequency. Another way to view the data, however, is to normalize the number of realizations identified by the number detected. Viewed in this fashion the Opt-S12 design performed slightly better than the other S12 designs with a normalized identification frequency of approximately 70 percent. The Opt-S3 and S6 designs, however, performed slightly worse then the other S3 and S6 designs.

The raw detection and identification data from Table 4.5 was grouped by treatment factor values and statistics were calculated on the resulting grouped data sets. The group statistics for the

Table 4.6: Summary of group statistics for detection data.

Group	*Number of Realizations*	*Detection Frequency*		*95 Percent Confidence Interval*	
		Mean	*Std. Error*	*Lower Bound*	*Upper Bound*
Opt	600	4.2333e-01	2.0188e-02	3.8369e-01	4.6298e-01
Dmd	600	3.7500e-01	1.9781e-02	3.3615e-01	4.1385e-01
Ad Hoc	600	5.3000e-01	2.0393e-02	4.8995e-01	5.7005e-01
S03	600	3.5500e-01	1.9552e-02	3.1660e-01	3.9340e-01
S06	600	4.3500e-01	2.0256e-02	3.9522e-01	4.7478e-01
S12	600	5.3833e-01	2.0369e-02	4.9833e-01	5.7834e-01

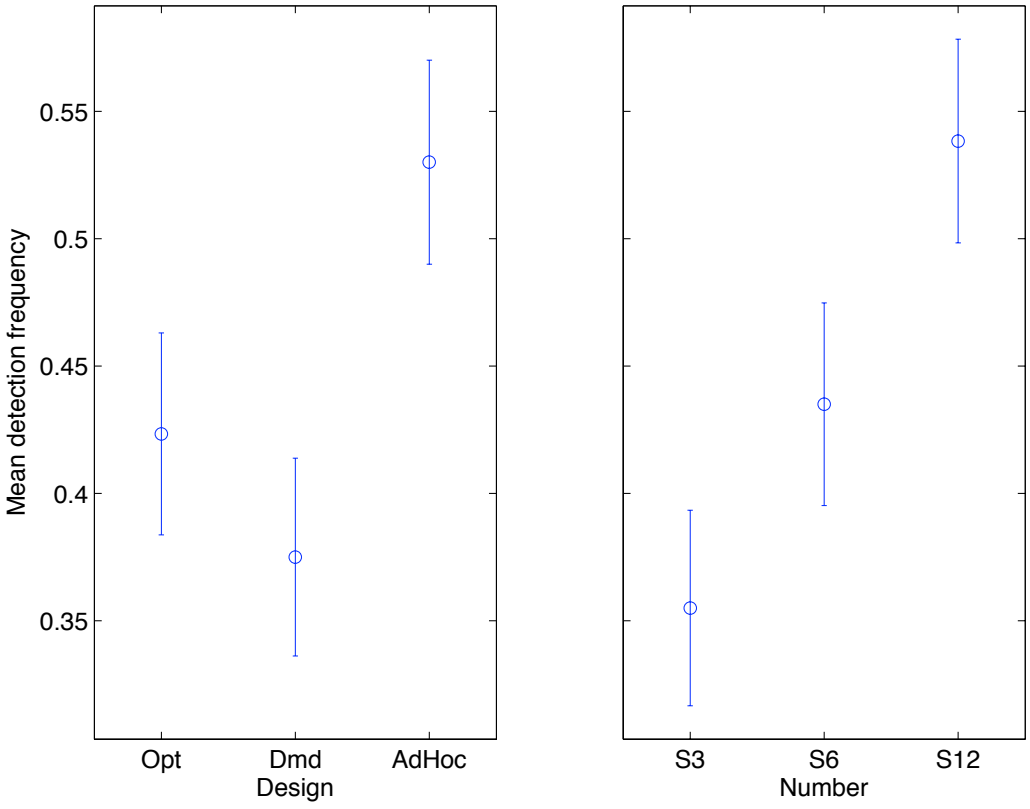

Figure 4.8: Statistics for the grouped detection data, group mean and 95 percentile confidence interval shown.

Table 4.7: Summary of group statistics for identification data.

Factor	*Number of*	*Identification Frequency*		*95 Percent Confidence Interval*	
	Realizations	*Mean*	*Std. Error*	*Lower Bound*	*Upper Bound*
Opt	254	4.6457e-01	3.1356e-02	4.0282e-01	5.2632e-01
Dmd	225	4.8444e-01	3.3391e-02	4.1864e-01	5.5025e-01
Ad Hoc	318	4.3082e-01	2.7813e-02	3.7610e-01	4.8554e-01
S03	213	1.5962e-01	2.5155e-02	1.1004e-01	2.0921e-01
S06	261	4.7126e-01	3.0957e-02	4.1031e-01	5.3222e-01
S12	323	6.4087e-01	2.6735e-02	5.8827e-01	6.9346e-01

detection data are summarized in Table 4.6 and a graphical representation of the key statistics can be found in Figure 4.8. The figure shows the mean value and 95 percent confidence interval for each group and for both factors. Overall the detection frequency for both factors was low, ranging from 36 to 54 percent. The difference between the mean detection frequency for the Ad Hoc and the other designs was found to be significant — note that the 95 percent confidence intervals for the means don't overlap (also see Table 4.6). In contrast to the Design factor, the differences between group mean detection frequencies for the Number factor were all significant. The range of the group means for the two factors are roughly equal; from this the author concludes that the design and the number of sensors are both important factors that influence detection frequency.

The statistics for the grouped identification data are summarized in Table 4.7 and shown in Figure 4.9. The mean identification frequency for the Design factor was less than 50 percent for all of the groups. Indeed, the differences between the means for the designs were found to be insignificant — note that the confidence intervals for all groups overlap one another. Once again, however, the differences between means for the Number factor were all found to be significant. Further, 64 percent of the sources detected were correctly identified by the S12 designs when taken as a group. When considering all source realizations, however, the S12 designs had a mean identification frequency of approximately 35 percent. This frequency is relatively low, attesting to the difficulty of the source identification problem.

Intuitively, source detection is necessary but not sufficient for source identification. It was hardly surprising then, that the detection data and identification data were highly correlated. In other words, the variability present in the identification data accounted for a significant amount of variability in the identification data. This may be an indication that the detection and identification objectives were not as orthogonal as was postulated by the author.

Raw solution error and objective error data are displayed in Table 4.5. Both the solution and objective error data was prepared by excluding realizations that were not detected. The solution error for each source realization was equal to $\|\mathbf{m} - \mathbf{m}'\|_2$. The Dmd designs had the best overall mean normalized solution error, while the Ad Hoc designs had the worst. The lowest mean was

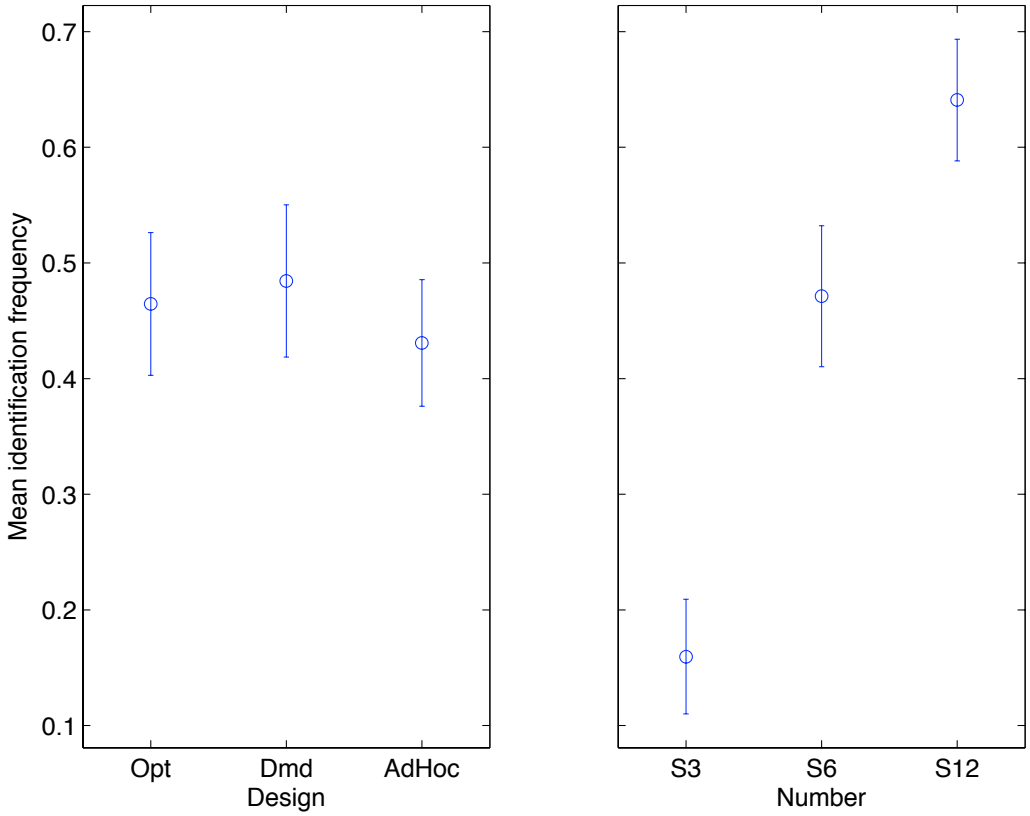

Figure 4.9: Statistics for the grouped identification data, group mean and 95 percentile confidence interval shown.

Table 4.8: Summary of group statistics for solution error data.

Factor	*Number of*	*Solution Error*		*95 Percent Confidence Interval*	
	Realizations	*Mean*	*Std. Error*	*Lower Bound*	*Upper Bound*
Opt	254	5.2813e-01	3.2384e-02	4.6435e-01	5.9191e-01
Dmd	225	5.0434e-01	3.5495e-02	4.3440e-01	5.7429e-01
AdHoc	318	5.9368e-01	3.0429e-02	5.3381e-01	6.5354e-01
S03	213	8.6629e-01	2.9028e-02	8.0907e-01	9.2351e-01
S06	261	5.3111e-01	3.2938e-02	4.6625e-01	5.9597e-01
S12	323	3.5069e-01	2.7652e-02	2.9629e-01	4.0509e-01

Table 4.9: Summary of group statistics for objective error data.

Factor	*Number of*	*Objective Error*		*95 Percent Confidence Interval*	
	Realizations	*Mean*	*Std. Error*	*Lower Bound*	*Upper Bound*
Opt	254	1.7402e-05	5.2154e-06	7.1313e-06	2.7674e-05
Dmd	225	1.5477e-04	9.5057e-05	-3.2549e-05	3.4209e-04
AdHoc	318	1.2402e-05	3.6505e-06	5.2196e-06	1.9584e-05
S03	213	1.0526e-04	9.0866e-05	-7.3856e-05	2.8438e-04
S06	261	3.9598e-05	2.5435e-05	-1.0487e-05	8.9682e-05
S12	323	3.2297e-05	2.0607e-05	-8.2443e-06	7.2838e-05

recorded for the Opt-S12 design. The performance of the Opt-S3 design, however, was poor relative to the other designs. The objective error for each source realization was equal to $\|\mathbf{c} - \mathbf{c}'\|_2$. The objective errors for all designs were very small, with means on the order of $1.0e-05$. The lowest objective error being recorded for the Opt-S3 design, and the largest for the Dmd-S3 design.

The statistics for the grouped solution error data are displayed in Table 4.8 and in Figure 4.10. The mean normalized solution error for all of the design groups was greater than 0.50, and furthermore, the differences observed between the means were not significant. Again, the Number factor had a significant influence on the mean normalized solution errors that were observed. The S12 group had a mean value of 0.35 and its marginal improvement relative to the other groups was significant.

The statistics for the grouped objective error data are shown in Table 4.9 and Figure 4.11. The mean objective errors for all groups were small; given that the source identification problems being solved were underdetermined to the extent that they were, this was hardly surprising. Review of the group statistics reveals that differences between means for all groups and both factors were not significant; however, the means of the Opt and Ad Hoc was lower than that for the Dmd group. Further, the Dmd factor designs possessed a higher degree of variability than either of the other two groups. Indeed, the variability present in the raw objective error data shown in Table 4.5 is an order of magnitude greater than the other designs. This is especially true for the Dmd, S3 and S6 designs. If a real problem were being solved one consequence of this large variability is that

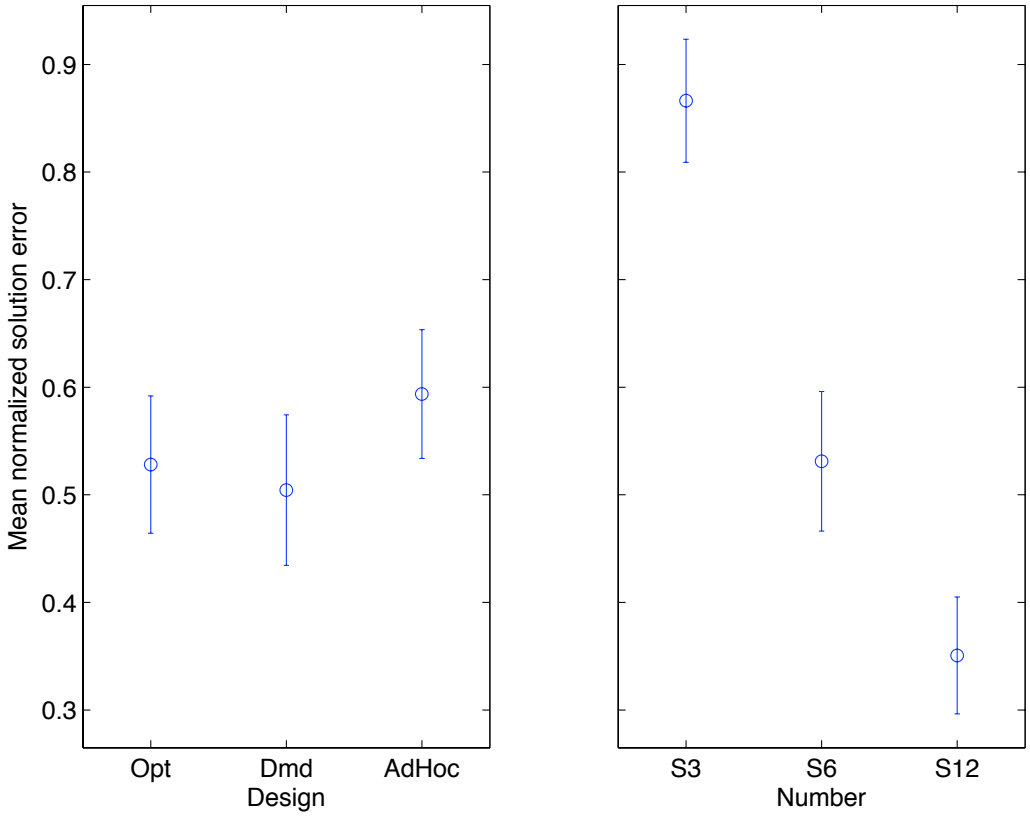

Figure 4.10: Statistics for the grouped solution error data, group mean and 95 percentile confidence interval shown.

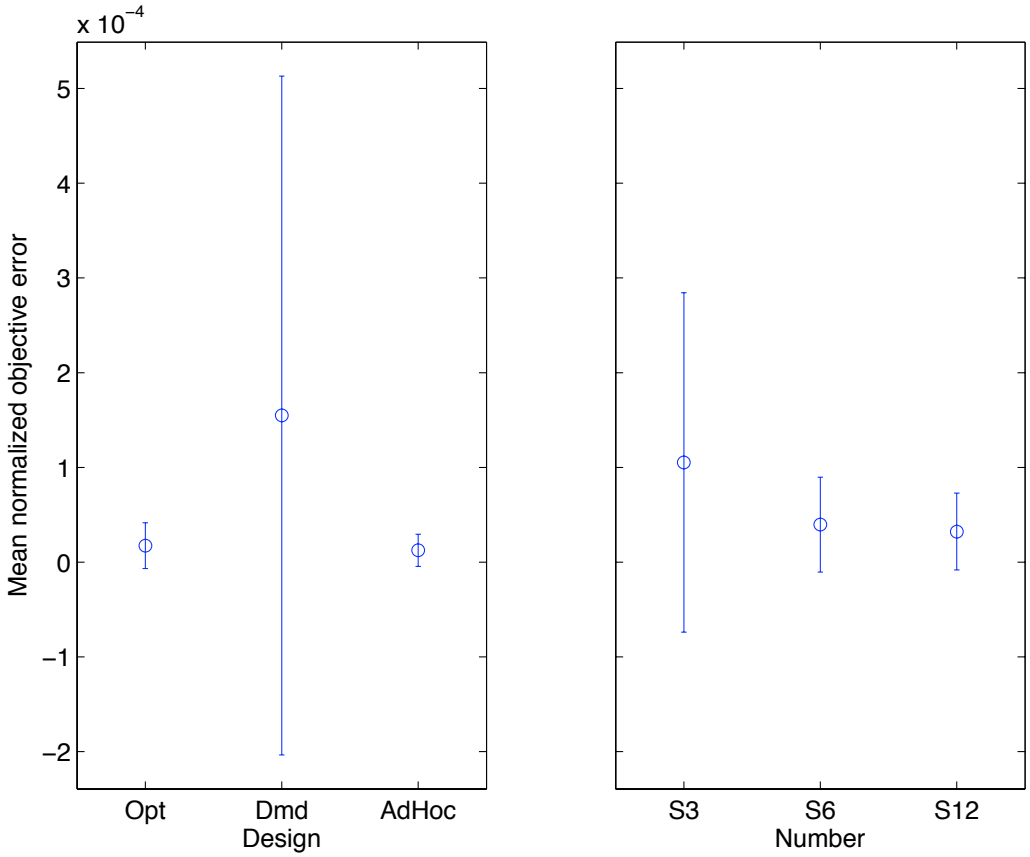

Figure 4.11: Statistics for the grouped objective error data, group mean and 95 percentile confidence interval shown.

greater uncertainty would be associated with the source identification solutions generated using the Dmd monitoring designs. Furthermore, this may be an indication that the Dmd designs do not yield source identification inverse problems that are as well conditioned as either the Opt or Ad Hoc methods.

A recurring pattern in the data is seen to emerge, where the differences between the Design factor groups are not as large in magnitude or significance compared to those for the Number factor groups. Figures 4.9 and 4.10 illustrate that the significance of the Number factor decreased as the number of sensors was increased. Case Study II was set in a regime of designs that were underdetermined by a factor ranging from 28:1 for the S3 group to 7:1 for the S12. Thus, the marginal benefit associated with adding sensors was observed to decrease as the degree to which the problem was underdetermined decreased. This effect can also be observed in the objective error date, see Figure 4.11. It was not significant, though, given the variability present in the sample. The number of sensors was anticipated as being an important monitoring design variable. What is interesting, however, was *how* important it was relative to other factors. It proved to be very important for this particular example and is suspected to be in general.

It is unclear why the demand rank heuristic Dmd designs performed as well as they did for this particular case. In Figure 4.12, the total volume demanded is displayed for each node in the Example 3 network. The figure shows that total demand in the network are heterogeneous, varying from 0.0 to approximately $6.5e-06$ gal/day. In Figure 4.6, the large demands are shown to occur in clusters about the lower two thirds of the network. The spatial clustering present in the Dmd designs was assumed to detrimentally affect source identification performance. On the contrary, the Dmd designs appear to have performed well if not better than either of other groups. It is unlikely, however, that this approach would perform as well when demands are less heterogeneous.

4.5 Conclusions

If a deliberate or accidental WDS contamination event occurred, a monitoring sensor network would provide important information to the system operators allowing them to characterize the source and execute an effective mitigating response. In this chapter an optimization model was developed for designing monitoring sensor networks with the objective of improving the conditioning of the source identification inverse problem.

Utilizing the input / output water quality model developed in Chapter 2 the water quality transport dynamics of a WDS can be represented in discrete form as a system of linear algebraic equations, called the forward model matrix. In general, this system of equations is underdetermined with the

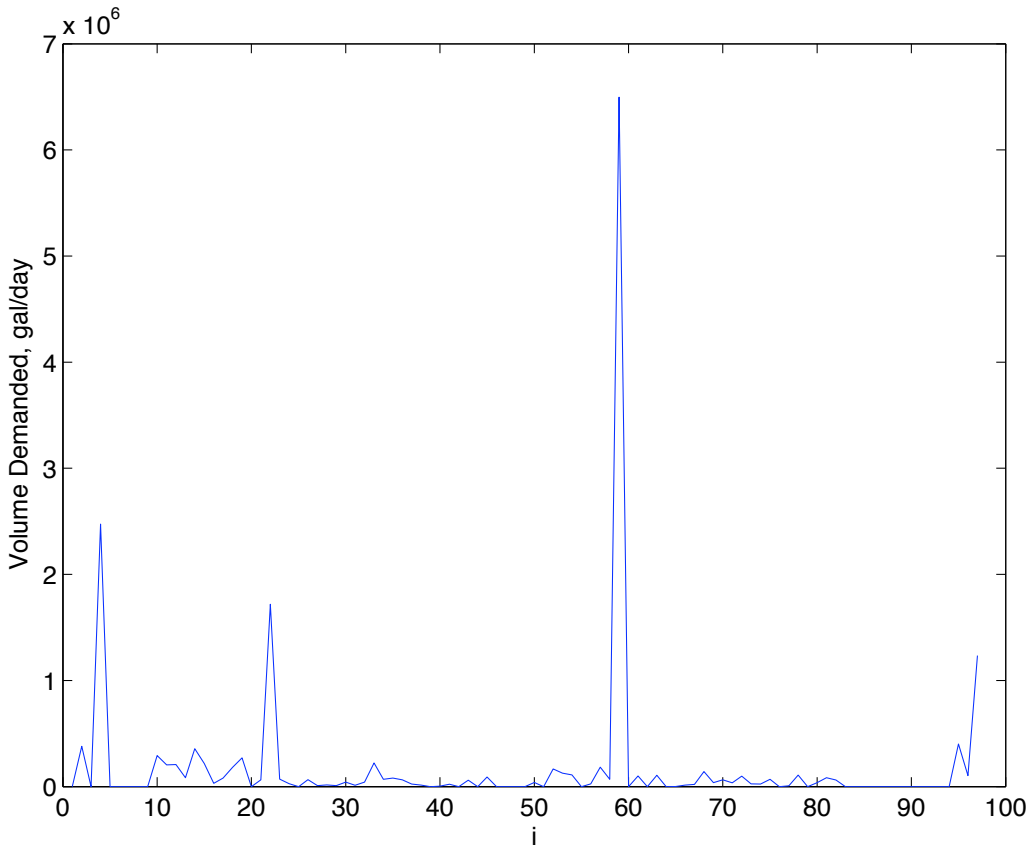

Figure 4.12: Total volume demanded per day for Realistic Network.

monitoring observations being greatly outnumbered by the potential contaminant sources. Underdetermined inversion problems are frequently ill-posed and difficult to solve. Factoring techniques such as eigenvalue decomposition can be used to analyze the stability of an inverse problem *a priori*. Analysis of the eigenvalue spectra, a product of the decomposition, provides important information on the conditioning of the underlying inverse problem.

The eigenvalue spectra of the forward model matrix is a function of the monitoring sensor network design — sensor location, sampling frequency, and sampling period length. A computationally efficient monitoring design objective that maximizes the normalized sum of the eigenvalue spectra was selected for this study. Maximizing this objective over a set of potential sensor locations selects for those locations that improve the conditioning of the underlying source identification inverse problem. The monitoring sensor network design problem was formulated as a non-linear combinatorial optimization model with binary decision variables and solved using the optimization framework developed in Chapter 3. Preliminary solution experience with the optimization model motivated a reformulation of the model that resembles a k-subset selection problem.

Two applications were developed using the proposed optimization model. The first utilized the model to generate the tradeoff between the conflicting inverse problem conditioning and sensor cost objectives for a hypothetical WDS network found in the literature. Results generated using the GA were verified by enumeration. Solving the optimization model using the GA for 30 random trials the optimal sensor locations were correctly generated for all points along the tradeoff curve excluding the 10 and 12 sensor designs. In general there are no guarantees of optimality associated with solutions generated using GAs; however, in this study the GA was able to reliably generate high performing designs.

The second was a Monte Carlo simulation based analysis to evaluate the effect of monitoring sensor network design on source identification solution performance, where the random variables considered were contaminant source location, start time, and duration. Three different methods were used to generate monitoring sensor networks — the monitoring design formulation, total demand ranking, and an *ad hoc* approach — and 3, 6 and 12 sensor designs were generated using each method. The source identification problem was solved using non-negative least squares for 200 random source realizations and each of the 9 sensor network designs in turn. Raw results were processed using a 3x3 factorial experimental design and statistical analysis to determine if statistically significant differences were present between treatment factors.

The *ad hoc* generated designs had the best detection frequency, detecting between 50 to 60 percent of the source realizations, while the total demand rank based designs performed the worst, only detecting between 27 to 50 percent of the realizations. Results of the statistical analysis for the detection date indicated that the *ad hoc* generated designs performed better by a statistically

significant margin than either the optimized designs or the total demand rank designs. In contrast, the differences in the mean detection frequency present between the 3, 6, and 12 sensor designs were all statistically significant. Thus, both the method of design and the number of sensors were determined to be important factors affecting source detection frequency.

The raw detection and identification data were highly correlated. This is hardly surprising as detection is necessary but not sufficient for identification. Identification frequencies were therefore normalized by the number of source realizations detected. No significant differences between means for the sensor network design methods were present in the normalized identification frequency data. On the contrary, clear and significant differences between means were present for the 3, 6, and 12 monitoring sensor network designs. Thus, when the source identification problem is highly underdetermined the number of sensors in the monitoring network is an important factor influencing identification frequency.

The distribution of source identification errors was also analyzed. The magnitude of the objective errors for all the experimental runs was small, on the order of $1.0e-04$ to $1.0e-06$. The variability of objective error values was much smaller for the optimized and *ad hoc* sensor designs compared to the total demand ranking design. Furthermore, variability in objective error values decreased as the number of sensors was increased. The differences in variability observed in the data may result in less confidence being attached to the results identified using the demand ranked monitoring designs.

The application of optimal inverse experimental design techniques to the source identification problem offers a quantitative framework for studying source inversion problem conditioning as it relates to the monitoring sensor network design. The application results consistently indicated that the method used to locate sensors was not as significant as the number of sensors installed on the basis of detection and identification performance. Furthermore, this significance decreased as the number of sensors was increased. It was anticipated that the number or sensors would be an important variable in monitoring sensor network design. The magnitude of its significance relative to other design variables, however, was not anticipated. These results cannot be easily extrapolated beyond the network used for this study. It is suspected, however, that the number of sensors is an important design variable in the general case.

Chapter 5

Summary and Conclusions

5.1 Executive Summary

The main contribution of this research is the analysis of monitoring and characterization problems in water distribution systems. Recent anxiety regarding the security of the nation's critical water infrastructure has renewed interest in these problems and reinvigorated research. This work has focused on addressing the difficulties associated with solution of source identification problems in water distribution systems due to ill-conditioning of the underlying inversion problem. Specifically, novel solution and monitoring design methods were investigated.

The solution of source identification problems in water distribution systems is an important aspect of preparedness for accidental and deliberate contamination events. Source identification problems are classified as inverse problems — a problem where system state is known and the parameters describing the system including boundary and initial conditions are unknowns. In Chapter 2 the difficulties associated with solving source identification inverse problem in water distribution systems are discussed.

Contaminant transport in water distribution systems is governed by the one dimensional advection dispersion reaction equation. The equation describes longitudinal fluid flow along the interconnected pipe segments of which the distribution network is composed. When the contaminant introduced into the system is conservative or reactive in the first order the equations describing reaction dynamics are linear in concentration. Higher order reactions result in non-linear reaction dynamics. When assuming linear reaction dynamics, contaminant transport in water distribution systems is governed by a linear homogeneous system of differential and algebraic equations, the solution of which resembles the linear integral equation referred to as the Fredholm integral equation of the first kind. A surprising number of inverse problems are mathematically related in that they can be written in this form. Integration of the integral equation yields a discrete linear system of

equations which can be solved using linear algebra. Thus, theoretically the source identification problem in water distribution systems fits within the well understood theory for solution of linear discrete inverse problems.

The resulting linear system of equations can be interpreted as an input / output water quality model. The dimensionality of the system of equations is governed by the number of potential source and monitoring nodes in the system and the discretization time step. Due to technological and economic constraints the number of potential source injections is greater than the number of monitoring observations and the linear system of equations will in general underdetermined. The source identification problem can be solved directly utilizing methods for underdetermined systems of equations, such as linear least squares where the misfit between the observed data and model predictions is minimized. One consequence of working with an underdetermined system is the existence of a non-trivial null space. In practical terms, the null space is the set of potential injections occurring in the network that are not observable by monitoring locations and therefore not characterizable. An extended example was developed illustrating the use of direct methods for the solution of the WDS source identification inverse problem.

Regularization is a technique for stabilizing the solution of ill-posed inverse problems by introducing *a priori* information about the anticipated solution. Zero-order Tikhonov regularization is the most commonly used of such methods. Applying Tikhonov regularization to discrete linear inverse problems results in a damped least squares formulation. Essentially, Tikhonov regularization introduces a new objective into the least squares problem — one that seeks to minimize the length of the solution vector — and the damping parameter controls the tradeoff between objectives. An example was used to illustrate an unintended consequence of the inclusion of this second objective, namely, that Tikhonov regularization tends to select for non-sparse solutions making the identification of contamination sources inadvertently more difficult. Indeed, a regularization method that selects for a sparse solution to the underdetermined system of linear equations would be more meaningful in this problem context.

Basis pursuit is one of several methods recently developed for the selection of sparse solutions of underdetermined systems of linear equations. A sparse solution is one where the objective is to minimize L_0 norm of the feasible solutions — minimize the number of non-zero elements in the solution vector. Formulation of the problem in this manner has been shown to be NP-hard owing to the combinatorial nature of the objective function. Applying convex relaxation the problem is reformulated such that the L_1 norm of the solution vector is minimized and the problem becomes a linear program. When the forward model matrix conforms to a specific set of assumptions a one to one correspondence exists between solutions of the L_0 and L_1 formulations of the problem. A sensitivity analysis was conducted to better understand the influence of monitoring sensor network

design on the recovery of sparse solutions by basis pursuit.

The emergence of computational grids has created new possibilities for the solution of environmental characterization problems. Chapter 3 describes the development of a grid-enabled simulation optimization framework. The framework is generic and can be utilized for the solution of problems in many different contexts. The framework, however, was developed with the specific intent of solving environmental monitoring and characterization problems — combinatorial optimization and inverse problems. Using the framework a novel simulation optimization-based approach combining global search heuristics with the readily available computing power of computational grids was investigated.

The framework was designed with a centralized optimization application and a simple master worker task pool distribution strategy. Standard communications protocols and the message passing interface 2 (MPI2) APIs were used to achieve a connection between the application framework and a legacy forward model. The resulting design is elegant in its simplicity and has beneficial load balancing characteristics without requiring an explicit task scheduling process. Task chunking was utilized to aggregate tasks with small evaluation times to control the ratio of evaluation to communication time and improve scaling performance. Within the centralized optimization application various EA methods have been implemented. EAs are embarrassingly parallel and readily amenable to the master worker task distribution strategy employed herein.

A theoretical performance analysis was performed on the master worker task distribution strategy and multi-threaded task pool implementation. Assuming a fixed problem size an expression for application speed-up was derived and shown to be a special instance of Amdahl's Law. Further, a simple communications model was developed assuming a linear function for communications cost. Inspection of the resulting expressions shows that communication cost can be minimized when aggregating tasks such that the number of task chunks is equal to the number of workers.

Illustrative applications were performed to demonstrate the effectiveness of the framework. The main focus of this research was monitoring and characterization problems in water distribution systems. Representative environmental characterization problems found in groundwater contexts, however, were also studied. The framework was coupled with a parallel three dimensional groundwater hydraulics and transport model for solution of a source identification and a source history release reconstruction problem. Runs were performed to evaluate the solution and computational performance achieved using the framework. The source identification problem was formulated and solved as an optimization problem and solved using an ES. The ES was able to successfully identify the source location and approximate the contaminant mass flux with an acceptable degree of error.

A battery of runs was conducted to measure computational performance for the single-site framework configuration at the semi-coarse and coarse grained levels of parallelism. The parallel fraction of the framework was estimated at approximately 99 percent using the data resulting from the runs

and the theoretical speed-up model developed previously. The framework was found to scale more efficiently at the semi-coarse grained than the coarse grained level.

Similarly, the source history reconstruction problem was also formulated as a optimization problem and solved using a GA. The GA was able to reconstruct the historical mass fluxes occurring at a known source location within an acceptable degree of error. The framework was configured for cross-site runs with worker processes at various clusters on the TeraGrid. The theoretical communication cost model was used to interpret data from a subset of the runs performed. The linear cost model was found to represent the observed data accurately. Further, it was evident from the results that the fixed and variable communication costs associated with a message were a function of the leg of the backplane interconnect the message traversed.

A second set of runs were conducted to better understand the effect that heterogenous communication costs associated with different legs of the interconnect would have on practical applications of the framework. The data was analyzed using the theoretical speed-up model, and again the parallel fraction of the framework configuration was estimated at approximately 99 percent. Minor differences in speed-up efficiency were observed among the different framework configurations tested. Analysis of the results revealed a degradation in speed-up efficiency when scaling from 64 to 128 groups that was not explained by the model. This loss of efficiency was attributed to parallel and communication overheads but the data set and speed-up model lacked the resolution to provide a specific characterization of the overhead.

The efficacy of a simple application architecture and task distribution strategy for grid-enabling a simulation optimization framework was demonstrated. The results indicate that significant and meaningful raw computational performance improvements were achieved without sacrificing solution performance when applying the framework to representative environmental characterization problems.

Research on monitoring sensor network design has focused on early detection and warning of contamination events with the expression of public health objectives. In Chapter 4 an optimization model was developed for designing monitoring sensor networks with the underlying objective of producing a well posed source identification inverse problem. The structure of a source identification solution and the errors associated with it are a function of monitoring design. Thus, monitoring design for source identification is an example of the coupling present between environmental monitoring and characterization problems.

Utilizing the input / output water quality model developed in Chapter 2 the water quality transport dynamics of a WDS can be represented in discrete form as a system of linear algebraic equations, called the forward model matrix. In general, this system of equations is underdetermined with the

monitoring observations being greatly outnumbered by the potential contaminant sources. Underdetermined inversion problems are frequently ill-posed and difficult to solve. Factoring techniques such as eigenvalue decomposition can be used to analyze the stability of an inverse problem *a priori*. Analysis of the eigenvalue spectra, a product of the decomposition, provides important information on the conditioning of the underlying inverse problem.

The eigenvalue spectra of the forward model matrix is a function of the monitoring sensor network design — sensor location, sampling frequency, and sampling period length. A theory for optimal inverse experimental design based on the concept of eigenvalue positivity has been developed and applied in the geophysical inverse problem domain. A computationally efficient monitoring design objective that maximizes the normalized sum of the eigenvalue spectra was selected for this study. Efficiency was improved further with the development of a method that eliminates a costly sparse matrix, matrix multiplication required for its calculation. Maximizing this objective over a set of potential sensor locations selects for those locations that improve the conditioning of the underlying source identification inverse problem.

The monitoring sensor network design problem was formulated as a non-linear combinatorial optimization model with binary decision variables and solved using the distributed simulation optimization framework developed in Chapter 3. Preliminary solution experience with the optimization model was unsatisfactory and motivated a reformulation of the model that resembles a k-subset selection problem. After reformulation, the GAs recombination and selection operator characteristics were fine tuned to suit the solution behavior observed in preliminary application runs.

Two applications were studied using the proposed optimization model. The first utilizes the model to generate the tradeoff between the conflicting inverse problem conditioning and sensor cost objectives for a hypothetical WDS network found in the literature. Results generated using the GA were verified by enumeration. Over the range of 2 to 12 sensors the decision space of the problem expands from approximately 300 to 9.7 million design combinations. Indeed, the size of the decision space even for modestly sized problems becomes astronomical. In solving the optimization model using the GA for 30 random trials the optimal sensor locations were correctly generated for all points along the tradeoff curve except for the 12 sensor design. Examination of the solutions revealed that the monitoring sensor designs exhibited an optimal substructure; as a consequence, the problem may be amenable to solution by greedy algorithm search heuristics or other methods. In general there are no guarantees of optimality associated with solutions generated using GAs; however, in this study the GA was able to reliably generate high performing designs.

The second was a Monte Carlo analysis to evaluate the effect of monitoring sensor network design on source identification solution performance, where the random variables considered were contaminant source location, start time, and duration. Three different methods were used to generate

monitoring sensor networks — the monitoring design formulation, total demand ranking, and an *ad hoc* approach — generating for each method a each 3, 6 and 12 sensor design respectively. Latin Hypercube Sampling was used to generate 200 uniformly distributed random source realizations. The source identification problem was solved using non-negative least squares for each of the 200 source realizations and each of the 9 sensor network designs in turn. A 2x3 factorial experimental design was employed and the raw results were analyzed to determine if statistically significant differences were present between treatment factors. The design treatments were evaluated on their ability to both detect and identify the source realizations. Further, the source identification results were evaluated on the basis of solution and objective errors.

The *ad hoc* generated designs had the best detection frequency, detecting between 50 to 60 percent of the source realizations, while the total demand rank based designs performed the worst, only detecting between 27 to 50 percent of the realizations. Results of the statistical analysis of the detection data indicated that the *ad hoc* generated designs performed better by a statistically significant margin than either the optimized designs or the total demand rank designs. In contrast, the differences in the mean detection frequency present between the 3, 6, and 12 sensor designs were all statistically significant. Thus, both the method of design and the number of sensors were determined to be important factors affecting source detection frequency.

The raw detection and identification data were highly correlated. This is hardly surprising as detection is necessary but not sufficient for identification. Identification frequencies were therefore normalized by the number of source realizations detected. No significant differences between means for the sensor network design methods were present in the normalized identification frequency data. In contrast, clear and significant differences between means were present for the 3, 6, and 12 monitoring sensor network designs. Thus, when the source identification problem is highly underdetermined the number of sensors in the monitoring network is an important factor influencing identification frequency.

The distribution of source identification errors when the source was not correctly identified was also analyzed. The magnitude of the objective errors for all the experimental runs was small, on the order of $1.0e-04$ to $1.0e-06$. The variability of objective error values was much smaller for the optimized and *ad hoc* sensor designs compared to the total demand ranking design. Furthermore, variability in objective error values decreased as the number of sensors was increased. Regardless of the differences in variability that were observed in the data, neither the method used to locate sensors nor the number of sensors in the network were significantly different from one another when comparing them on the basis of mean objective error.

The application results consistently indicated that the method used to locate sensors was not as significant as the number of sensors installed on the basis of detection and identification performance.

Furthermore, this significance decreased as the number of sensors was increased. It was anticipated that the number of sensors would be an important variable in monitoring sensor network design. The magnitude of its significance relative to design factors, however, was not anticipated. These results can not be easily extrapolated beyond the network used for this study. It is suspected, however, that the number of sensors is an important design variable in the general case.

5.2 Final Remarks

The results of this research can best be used to improve the solution quality of contaminant source identification problems in water distribution systems. As with all research, this work has generated several questions requiring further investigation.

This research has demonstrated that assuming *a priori* a sparse solution of the source identification problem is more meaningful than other regularization methods in this problem context. This work has generated some promising initial results. The development of a source identification problem formulation that selects for the sparse solution would make a useful topic for future research. This could be accomplished by explicitly incorporating a secondary objective or penalty function that tracks the number of unique source locations involved in the present solution. Or it could be accomplished implicitly, by specifying an adaptive encoding mechanism that seeks to simplify the complexity of the solution representation, where complexity is proportional to the number of unique source locations involved in the solution.

Sparse solution techniques hold promise to improve the usefulness of source identification solutions. The opportunity exists, however, to develop sparse solution methods for general problems. The problem of identifying the sparse solution of a underdetermined linear system of equations arises in many different application areas and has been shown to be NP-hard. The problem is well posed only when the columns of the matrix forming the linear system of equations are orthogonal. Though this requirement can be significantly relaxed, many practical applications do not conform to this set of assumptions. Heuristic methods have been developed for the identification of sparse solutions, but the application of evolutionary algorithms for this purpose has not been investigated, despite being well suited to the problem's characteristics. Thus, another promising topic for future research would be the development of a specialized evolutionary algorithm for identifying the sparse solution of underdetermined systems of linear equations.

The assumption of known water distribution hydraulic dynamics is fundamental to the linear input / output water quality model used to formulate the source identification and monitoring network design problem investigated in this work. In reality, demands in water distribution systems are governed by random variables and thus stochastic in nature. Ultimately, the source identification

problem must be solved in real time to be capable of protecting public health if a contamination event were to occur. Therefore, the hydraulics of the system must also be estimated at the time the source identification inversion problem is solved. Problems with dependencies of this type are complex and referred to as coupled inverse problems. Study of real-time hydraulics estimation and coupled formulations of the inversion problem have the potential to make a significant contribution to the ultimate goal of solving real-world source identification problems.

The monitoring sensor network design problem developed here is a planning model. As such, assumptions are necessary as the state of the system over the planning horizon is unknowable. Some sources of uncertainty, however, are more characterizable than others. The stochasticity of water system demands is one such source uncertainty. It would be meaningful to investigate the sensitivity of the proposed design method when subject to random demands to develop a robust method for locating sensor in water distribution networks.

List of References

[1] Gabrielle Allen, Tom Goodale, Michael Russell, Edward Seidel, and John Shalf. *Classifying and Enabling Grid Applications*, chapter 23, pages 601–614. Wiley Series in Communications Networking & Distributed Systems. John Wiley and Sons, Ltd, 2000.

[2] Janic F. Artiola, Ian L. Pepper, and Mark L. Brusseau. *Environmental Monitoring and Characterization*. Elsevier Academic Press, Amsterdam, Boston, 2004.

[3] Richard A. Aster, Brian Borchers, and Clifford H. Thurber. *Parameter Estimation and Inverse Problems*, volume 90 of *International Geophysics*. Elsevier Academic Press, 2005.

[4] Thomas Back, David B. Fogel, and Zbigniew Michalewics, editors. *Handbook of Evolutionary Computation*. Institute of Physics and Oxford University Press, London, UK, 1997.

[5] John W. Baugh. Vitri 2.0, 2003.

[6] Bear. *Dynamics of Fluids in Porous Media*. Elsevier, 1972.

[7] Jonathan Berry, Lisa Fleischer, William Hart, and Cynthia Phillips. Sensor placement in municipal water networks. In *World Water and Environmental Resources Congress*, Philadelphia, PA, United States, Jun 23-26 2003. American Society of Civil Engineers.

[8] Jonathan Berry, W. E. Hart, Phillips C. A., and Uber J. G. A general integer-programming-based framework for sensor placement in municipal water networks. Symposium on Water Distributions Analysis, March 16 2004.

[9] Dominic L. Boccelli, Michael E. Tryby, James G. Uber, Lewis A. Rossman, Michael L. Zierolf, and Marios M. Polycarpou. Optimal scheduling of booster disinfection in water distribution systems. *Journal of Water Resources Planning and Management*, 124(2):99–111, 1998.

[10] Erick Cantu-Paz. Designing efficient master-slave genetic algorithms. IlliGAL Report 97004, Illinois Genetic Algorithms Laboratory, May 1997.

[11] R.D. Carr, H.J. Greenberg, W.E. Hart, and Phillips C.A. Addressing modeling uncertainties in sensor placement for community water systems. In *World Water and Environmental Resources Congress*, Salt Lake City, Utah, June 27 – July 1 2004. American Society of Civil Engineers.

[12] C. Catlett. The teragrid: A primer, 2002.

[13] Scott Shaobing Chen, David L. Donoho, and Michael A. Saunders. Atomic decomposition by basis pursuit. *SIAM Journal on Scientific Computing*, 20(1):33–61, 1998.

[14] Sophie Constans, Bernard Bremond, and Paul Morel. Simulation and control of chlorine levels in water distribution networks. *Journal of Water Resources Planning and Management*, 129(2):135–145, 2003.

[15] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. *Introduction to Algorithms*. MIT Press, Cambridge, MA, 2nd. edition, 2001.

[16] S. J. Cox, M. J. Fairman, G. Xue, J. L. Wason, and A. J. Keane. The grid: Computational and data resource sharing in engineering optimisation and design search. In *IEEE Proceedings of the 2001 ICPP Workshops*, pages 207–212, Valencia, Spain, September 2001. IEEE.

[17] Andrew Curtis. Optimal experiment design: cross-borehole tomographic examples. *Geophysical Journal International*, 136(3):637–650, March 1999.

[18] Andrew Curtis and Roel Snieder. Reconditioning inverse problems using the genetic algorithm and revised parameterization. *Geophysics*, 62(5):1524–1532, 1997.

[19] David L. Donoho and Jared Tanner. Sparse nonnegative solution of underdetermined linear equations by linear programming. *Proceedings of the National Academy of Sciences of the United States of America*, 102(27):9446–9451, July 2005.

[20] D.L. Donoho and X. Huo. Uncertainty principles and ideal atomic decomposition. *Information Theory, IEEE Transactions on*, 47(7):2845–2862, 2001.

[21] M. Elad and A.M. Bruckstein. A generalized uncertainty principle and sparse representation in pairs of bases. *Information Theory, IEEE Transactions on*, 48(9):2558–2567, Sept. 2002.

[22] David E Goldberg. *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison Wesley, 1989.

[23] W. G. Gray and G. F. Pinder. An analysis of the numerical solutions of the transport equation. *Water Resources Research*, 12(3):547–555, June 1976.

[24] J. Hadamard. Sur les problmes aux drives partielles et leur signification physique. *Bull. Univ. Princeton*, 13:49–52, 1902.

[25] Benjamin L. Harding and Thomas M. Walski. Long time-series simulation of water quality in distribution systems. *Journal of Water Resources Planning and Management*, 126(4):199–209, 2000.

[26] Nicholas J. Higham. Estimating the matrix p-norm. *Numerische Mathematik*, 62(1):539–555, 1992.

[27] Debra M. Knopman and Clifford I. Voss. Behavior of sensitivities in the one-dimensional advection-dispersion equation: Implications for parameter estimation and sampling design. *Water Resources Research*, 23(2):253–272, February 1987.

[28] Carl D. Laird, Lorenz T. Biegler, Bart G. Van Bloemen Waanders, and Roscoe A. Bartlett. Contaminant source determination for water networks. *Journal of Water Resources Planning and Management*, 131(2):125–134, March/April 2005.

[29] P. D. Lax and B. Wendroff. Systems of conservation laws. *Communications on Pure and Applied Mathematics*, 13:217 – 237, 1960.

[30] Dudy Lim, Yew-Soon Ong, Yaochu Jin, Bernhard Sendhoff, and Bu-Sung Lee. Efficient hierarchical parallel genetic algorithms using grid computing. *Future Generation Computer Systems*, 23(4):658–670, May 2007.

[31] P. S. Mahar and B. Data. Optimal monitoring network and ground-water-pollution source identification. *Journal of Water Resources Planning and Management*, 123(4):199–207, 1997.

[32] Mahinthakumar. Pgrem3d: Parallel groundwater transport and remediation codes. Users Guide, 1999.

[33] G. Mahinthakumar and F. Saied. Implementation and performance analysis of a parallel multicomponent groundwater transport code. In *Proceedings of the 1999 SIAM Parallel Processing Meeting*, San Antonio, TX, 1999. SIAM.

[34] G. Mahinthakumar and F. Saied. A hybrid mpi-openmp implementation of an implicit finite-element code on parallel architectures. *International Journal of High Performance Computing Applications*, 16(4):371–393, Winter 2002.

[35] G. Mahinthakumar and M. Sayeed. Hybrid genetic algorithm local search methods for solving groundwater source identification inverse problems. *Journal of Water Resources Planning and Management*, 131(1):45–57, January/February 2005.

[36] V. Matossian and M. Parashar. Autonomic optimization of an oil reservoir using decentralized services. In *Proceedings of the 1st International Workshop on Heterogeneous and Adaptive Computing*, pages 2–9, Seattle, WA, USA, June 2003. Computer Society Press.

[37] William Menke. *Geophysical Data Analysis: Discrete Inverse Theory*, volume 45 of *International Geophysics*. Academic Press, San Diego, CA, 1989.

[38] Philip D. Meyer, Albert J. Valocchi, and Wayland J. Eheart. Monitoring network design to provide initial detection of groundwater contamination. *Water Resources Research*, 30(9):2647–2659, 1994.

[39] BahaEldin Yousif Ahmed Mirghani. *Evolutionary Algorithms-based Parallel Simulation-Optimization Framework for Solving Inverse Problems*. PhD thesis, North Carolina State University, 2007.

[40] B. K. Natarajan. Sparse approximate solutions to linear systems. *SIAM Journal on Computing*, 24(2):227–234, 1995.

[41] Avi Ostfeld and Elad Salomons. Optimal layout of early warning detection stations for water distribution systems security. *Journal of Water Resources Planning and Management*, 130(5):377 – 385, September / October 2004.

[42] Avi Ostfeld and Elad Salomons. A stochastic early warning detection system model for drinking water distribution systems security. In *World Water and Environmental Resources Congress*, Salt Lake City, Utah, June 27–July 1 2004. American Society of Civil Engineers.

[43] M. Parashar, H. Klie, U. Catalyurek, T. Kurc, W. , Bangerth, V. Matossian, J. Saltz, and M. F. Wheeler. Application of grid-enabled technologies for solving optimization problems in data-driven reservoir studies. *Future Generation of Computer Systems*, 2004.

[44] Eillen.P. Poeter and Mary C. Hill. Inverse modeling, a necessary next step in ground-water modeling. *Ground Water*, 35(2):250–260, 1997.

[45] M. Propato, P. B. Cheung, and O. Piller. Sensor location design for contaminant source identification in water distribution systems. In *Proceedings of the 8th Annual Water Distribution Systems Analysis Symposium*, Cincinnati, Ohio, USA, August 27–30 2006.

[46] Marco Propato and James G. Uber. Linear least-squares formulation for operation of booster disinfection systems. *Journal of Water Resources Planning and Management*, 130(1):53–62, 2004.

[47] H. S. Rao and D. W. Bree Jr. Extended period simulation of water systems — part a. *Journal of the Hydraulics Division, ASCE*, 103(2):97–108, March 1977.

[48] I. Rechenberg. *Evolutionsstrategie – Optimierung technisher Systeme nach Prinzipien der biologischen Evolution*. Frommann-Holzboog, Stuttgart, GER, 1973.

[49] Lewis A. Rossman. *EPANET 2 Users Manual*. United States Environmental Protection Agency, Water Supply and Water Resources Division, National Risk Management Research Laboratory, Cincinnati, OH 45268, September 2000.

[50] Lewis A. Rossman and Paul F. Boulos. Numerical methods for modeling water quality in distribution systems: A comparison. *Journal of Water Resources Planning and Management*, 122(2):137–146, 1996.

[51] Feng Shang, James G. Uber, and Marios M. Polycarpou. Particle backtracking algorithm for water distribution system analysis. *Journal of Environmental Engineering*, 128(5):441–450, 2002.

[52] A. F. B. Thompson, R. Aboubu, and L. W. Gelhar. Implementation of the three-dimensional turning bands random field generator. *Water Resources Research*, 25(10):2227–2243, Oct 1989.

[53] Andrey N. Tikhonov and Vasiliy Y. Arsenin. *Solutions of Ill-Posed Problems*. Scripta Series in Mathematics. V. H. Winston and Sons, Washington, D.C., 1977.

[54] E. Todini and S. Pilati. A gradient method for the analysis of pipe networks. In *International Conference on Computer Applications for Water Supply and Distribution*, Leicester Polytechnic, UK, Semptember 8 - 10 1987.

[55] Michael E. Tryby, Dominic L. Boccelli, James G. Uber, and Lewis A. Rossman. Facility location model for booster disinfection of water supply networks. *Journal of Water Resources Planning and Management*, 128(5):322–333, 2002.

[56] James Uber, Robert Janke, Regan Murray, and Philip Meyer. Greedy heuristic methods for locating water quality sensors in distribution systems. In *World Water and Environmental Resources Congress*, Salt Lake City, Utah, June 27–July 1 2004. American Society of Civil Engineers.

[57] Bart G. Van Bloemen Waanders, Roscoe A. Bartlett, Lorenz T. Biegler, and Carl D. Laird. Nonlinear programming strategies for source detection of municipal water networks. In *World Water and Environmental Resources Congress*, Philadelphia, PA, United States, Jun 23–26 2003. American Society of Civil Engineers.

[58] Curtis R. Vogel. *Computational Methods for Inverse Problems*. Frontiers in Applied Mathematics. SIAM, 2002.

[59] Zhong Wang, M. M. Polycarpou, J. G. Uber, and Feng Shang. Adaptive control of water quality in water distribution networks. *Control Systems Technology, IEEE Transactions on*, 14(1):149–156, 2006.

[60] Jean-Paul Watson, J. Greenburg, Harvey, and Hart E. William. A multi-objective analysis of sensor placement optimization in water networks. In *World Water and Environmental Resources Congress*, Salt Lake City, Utah, June 27–July 1 2004. American Society of Civil Engineers.

[61] Kamin Whitehouse. Tinyos project, 2005.

[62] Kai Xu, Sushil J. Louis, and Roberto C. Mancini. A scalable parallel genetic algorithm for x-ray spectroscopic analysis. In *Proceedings of the 2005 Conference on Genetic and Evolutionary Computation*, pages 811–816, Washington DC, USA, 2005. ACM Press.

[63] G. Xue, W. Song, A. J. Keane, and S. J. Cox. Developing services for design optimization on the grid. *IEEE*, 2004.

[64] Emily M. Zechman, Jr. E. Downey Brill, G. Mahinthakumar, S. Ranjithan, and James Uber. Addressing non-uniqueness in a water distribution contaminant source identification problem. volume 247, pages 126–126. ASCE, 2006.

[65] M.L. Zierolf, M.M. Polycarpou, and J.G. Uber. Development and autocalibration of an input-output model of chlorine transport in drinking water distribution systems. *Control Systems Technology, IEEE Transactions on*, 6(4):543–553, 1998.

Appendices

Appendix A

Sensitivity Analysis

A.1 Experimental Design

In this section a sensitivity analysis is conducted to illustrate the effect of monitoring design on source identification solution quality. A monitoring design consists of the number and location or monitoring sensors, their sampling frequency, the length of time over which samples are taken, and the degree to which the contaminant injection at potential sources is discretized. The sensitivity analysis was performed using the basis pursuit formulation of the source identification problem for the Example 3 network analyzed in the previous sections. Two separate analysis were organized by logically grouping parameters; 1) number and location of sensors and length of the monitoring period, 2) sampling frequency and source discretization. The details of the experimental design used to conduct the sensitivity analysis are shown in Table A.1.

The length of the sampling period after a contamination event has been detected is an important factor influencing source identification quality, and is also a surrogate for contaminant exposure prior to a source identification. Thus, theoretically a tradeoff exists between source identification solution quality and population exposure. For this reason the shortest sampling period capable of producing an accurate source identification is desirable. Three different sampling period lengths were studied, 4, 8, and 12 hours. The 4 hour period representing the minimum time for contaminant transport capable of creating an observable contamination signature. While the 12 hour period represents the maximum tolerable time over which exposure could occur.

Monitoring sensors are likely to be expensive and sparsely located in a "real world" WDS; hence, the number of monitoring nodes in each design was chosen to reflect this. Three different monitoring sensor groups were used in the creation of the designs as shown in Figure A.1. Sets A, B, and C have 6, 3, and 3 monitoring nodes respectively. Further, the monitoring designs are supersets of each other — design A+B is a superset of A, and A+B+C is a superset of A+B — as indicated by

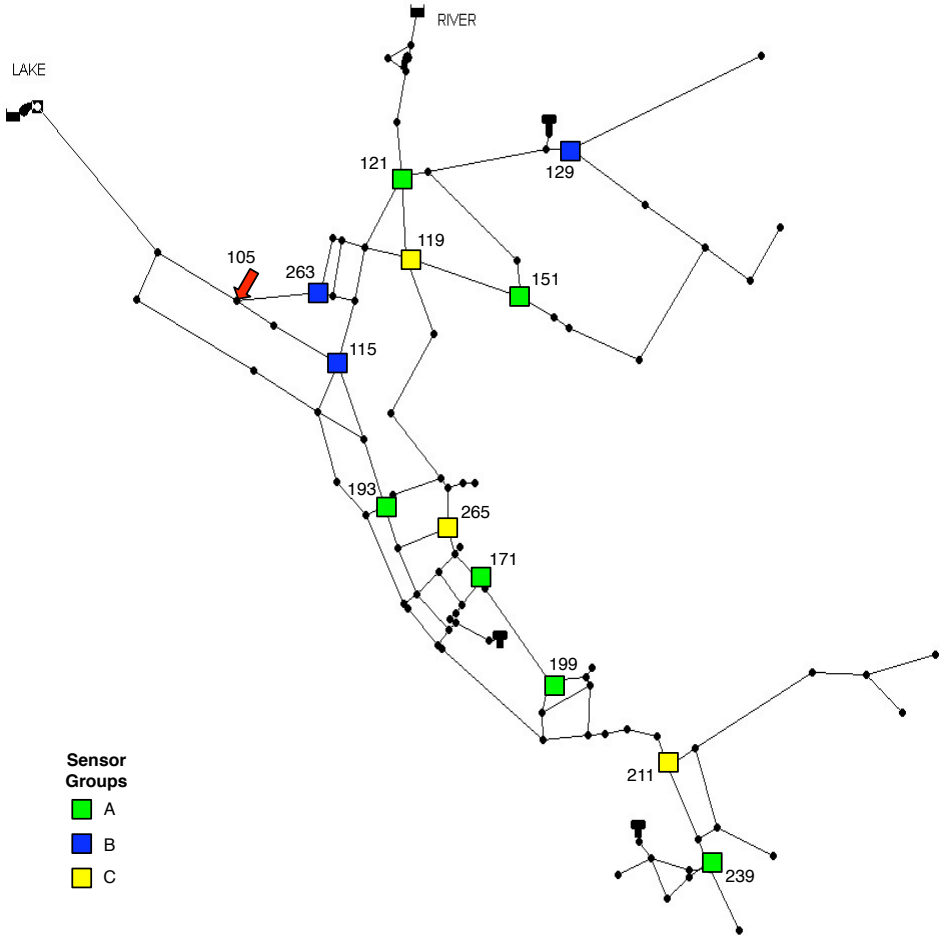

Figure A.1: Realistic network with sensor network broken down into 3 subgroups.

Table A.1: Sensitivity analysis experimental design.

Run (1)	*Monitoring Period Length (hrs)* (2)	*Monitoring Node Set* (3)	*Sampling Frequency (1/hrs)* (4)	*Source Discretization (1/hrs)* (5)	*Forward Matrix Dimensions* (6)
1.1	4	A	6	6	144 x 2328
1.2	4	A+B	6	6	216 x 2328
1.3	4	A+B+C	6	6	288 x 2328
1.4	8	A	6	6	288 x 4656
1.5	8	A+B	6	6	432 x 4656
1.6	8	A+B+C	6	6	576 x 4656
1.7	12	A	6	6	432 x 6984
1.8	12	A+B	6	6	648 x 6984
1.9	12	A+B+C	6	6	864 x 6984
2.1	8	A+B	6	6	432 x 4656
2.2	8	A+B	2	6	144 x 4656
2.3	8	A+B	1	6	72 x 4656
2.4	8	A+B	6	2	432 x 1552
2.5	8	A+B	2	2	144 x 1552
2.6	8	A+B	1	2	72 x 1552
2.7	8	A+B	6	1	432 x 776
2.8	8	A+B	2	1	144 x 776
2.9	8	A+B	1	1	72 x 776

their names. When monitoring nodes are added to form a superset, new data is being added to that already being collected from the previous set of monitoring nodes, and thus the amount of data for the source identification problem is incrementally increased. The locations of the monitoring nodes in each set were selected in an ad hoc fashion using judgement to avoid clustering of sensors in the network.

The size of the forward model matrix and therefore the degree to which the problem is underdetermined is influenced by the sampling frequency and source discretization selected for the source identification problem formulation. Sampling frequency is limited by the sensor technology deployed at a monitoring installation. Sensor development for contaminant detection is an active area of research that is beyond the scope of this work. For the sake of simplicity, perfect sensors capable of detecting and quantifying the contaminant injected into the system were assumed. Considering a range of sampling frequencies between 6 and 1 (samples/hour), however, was an attempted to bracket a realistic range of sampling frequencies. With 6 samples per hour being moderately optimistic and once per hour being pessimistic based on our general knowledge of sensor technologies. Source discretization determines the resolution of the contamination source release history that will be estimated as part of the source identification. Correctly identifying the location of the source is

Table A.2: Sensitivity analysis experimental results.

Run	*Zero Norm* $\|\mathbf{m}\|_0$	*One Norm* $\|\mathbf{m}\|_1$	*Residual Norm* $\|\mathbf{Am}-\mathbf{c}\|_2$	*Status*
(1)	(2)	(3)	(4)	(5)
1.1	33	162.0777	1.5921e-18	converged
1.2	24	258.9232	2.9964e-16	converged
1.3	24	258.9232	2.0625e-15	converged
1.4	930	206.7284	1.4006e-07	no feasible point found
1.5	24	240.0000	3.3395e-16	converged
1.6	24	240.0000	5.6170e-10	max iterations exceeded
1.7	2935	220.2214	4.9157e-06	no feasible point found
1.8	2623	240.5190	1.3970e-05	no feasible point found
1.9	1587	196.0198	5.1e-03	no feasible point found
2.1	24	240.0000	3.3395e-16	converged
2.2	1593	511.5725	5.2874e-06	no feasible point found
2.3	24	256.4861	9.1262e-13	max iterations exceeded
2.4	24	240.7320	3.3205e-16	max iterations exceeded
2.5	70	236.2535	6.4576e-18	converged
2.6	37	261.0946	1.8573e-18	converged
2.7	400	225.4615	1.9082e-02	no feasible point found
2.8	406	229.2239	5.8268e-06	no feasible point found
2.9	36	246.6738	5.2903e-18	converged

of primary importance. Determination of the release history is of secondary importance. Therefore, it may be possible to reduce the complexity of the source identification problem by reducing the number of parameters that require estimation, provided that source location identification accuracy is preserved.

A.2 Results and Discussion

As mentioned previously, two separate analysis were organized around a logical grouping of sampling parameters. The number and location of sensors was varied with the sampling period length to create a simple parametric study. The first study evaluates the equivalence between locating more sensors *a priori* or sampling for a longer period of time *a posteriori* and any effects these two variables may have on the resulting source identification solution. The second study was organized in a similar fashion, by varying the sampling frequency and source discretization. The second study illustrates how the source identification solution changes as the degree to which the system of equations is underdetermined is modified, by sampling less frequently at monitoring locations or by estimating fewer source injection parameters.

The run index, zero norm, one norm, residual norm, and LP convergence status are summarized

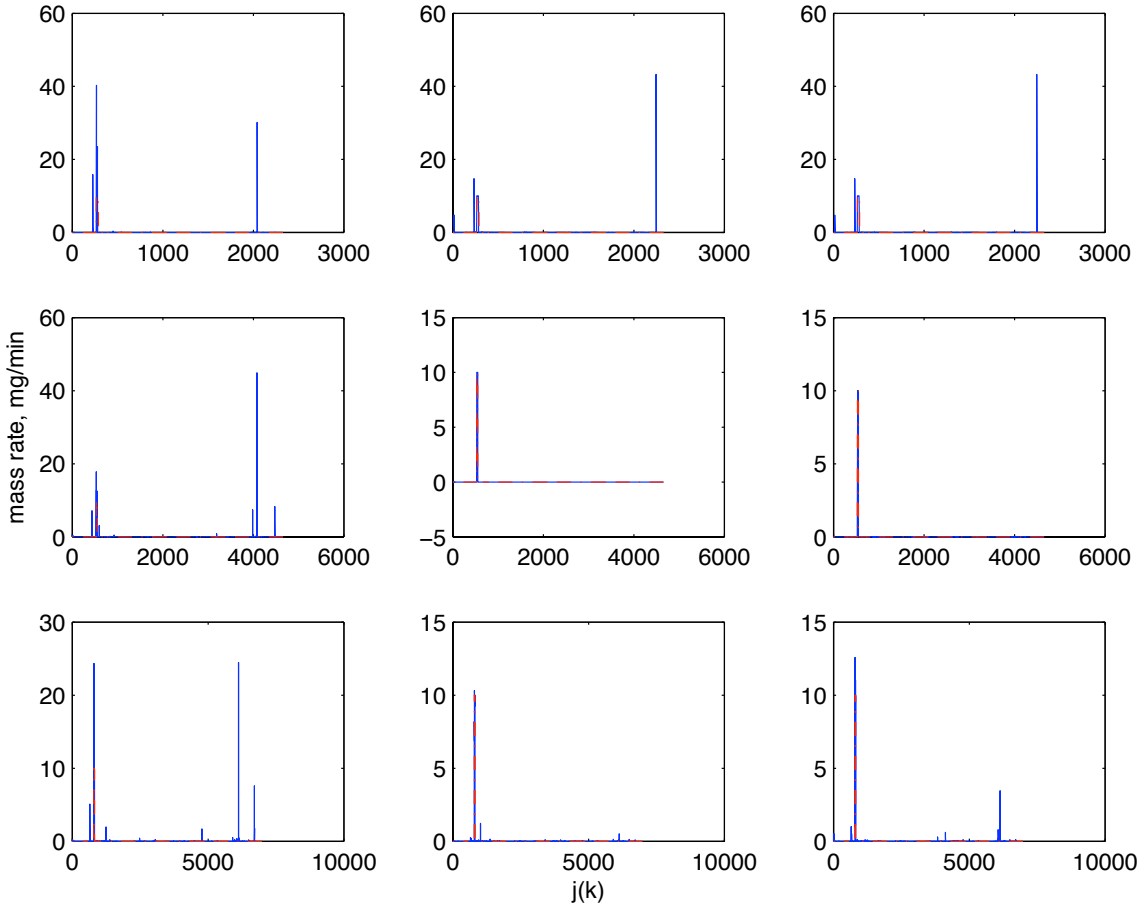

Figure A.2: Sensitivity analysis, number and location of sensors and sampling duration. Source identification solution shown in blue, true solution in red.

in Table A.2. It should be noted that the zero norm L_0 (see column (2)) of the source identification problem solution was computed by filtering out small elements of the solution vector as follows $L_0 = \sum_j^{nT} \mathbf{m_j} > \|\mathbf{m}\| * 1.0E-06$. The source identification solution vectors for each run are illustrated in Figures A.2 and A.4.

Figure A.2 is a 3x3 composite plot, where the number or monitoring sensors is increasing from 6 to 12 from right to left and the sampling period is increasing from 4 to 12 hours from top to bottom. Thus, the plot in the upper left is the source identification solution obtained from the fewest samples and likewise the plot in the lower right has the most samples. Broadly speaking, the results indicate that solution quality did not improve uniformly as the number of samples taken, either by locating more sensors or by extending the sampling period, was increased. Further, the results appear most sensitive to the number and location of sensors.

Viewing the results from left to right, the solutions tend to become more sparse, this is also apparent from inspecting Table A.2 column (2). In all cases, the solutions become more sparse as the number of sensors increased. Increasing the number of monitoring sensors also appears to have improved the solution accuracy of the source identification problems. The improvement in solution accuracy was most pronounced as the monitoring design went from 6 to 9 sensors, monitoring sets A to A+B respectively. This may be attributable to the increase in the number of sensors, or more likely, the proximity of the set B sensor locations to the true source at node 105. The design of the experiment prohibits us from concluding whether this can be attributed simply to the number of sensors or to the sensors proximity to the contamination source. Rather a comprehensive quantitative method for locating monitoring sensors would be required to make that determination. The results do suggest, however, that monitoring sensor location is an important factor in solution quality and therefore problem conditioning.

Viewing the results from top to bottom, there appears to be little change in the solutions. Extending the sampling period from 8 to 12 hours, however, had a detrimental effect on problem conditioning. Indeed, all of the source identification problems where the sampling period was 12 hours were poorly conditioned to an extent that they failed to converge to a feasible solution. Though in some cases the solutions do appear serviceable, *i.e. runs 1.8 and 1.9*, there is no way to determine this without the benefit of knowing the true solution. The observation that extending the sampling period does not necessarily improve the solution is counter intuitive; an improvement in solution quality was anticipated.

The strategy of extending the sampling period to acquire more information may fail for two reasons, 1) it is a zero sum game, and 2) additional samples are not always valuable for parameter estimation. Extended sampling is a zero sum game because the number of unknown injection terms increases in proportion to the number of additional samples taken. This is confirmed by examining the ratio of rows:columns in Table A.2 column (6). Thus, despite the extension of the sampling period the degree to which the problem is underdetermined stays constant. Secondly, samples taken later in an extended sampling period do not necessarily improve parameter estimates; rather, it is more important that the samples span the passing of a contamination front to yield parameter estimates with lower variance [27]. Once the contamination front has extend beyond the sensor network and into the periphery of the system continuing to sample does little to improve the solution of the source identification problem and may actually make the problem more difficult to solve. Considering that extending the sampling period also increases the exposure potential of a contamination event, it would appear that there is no advantage in pursuing it as a sampling strategy.

Examining the solution to sensitivity analysis run 1.1 (Figure A.2 upper left corner) in greater detail. The sources identified in the solution are indicated in Figure A.3. Following from left to

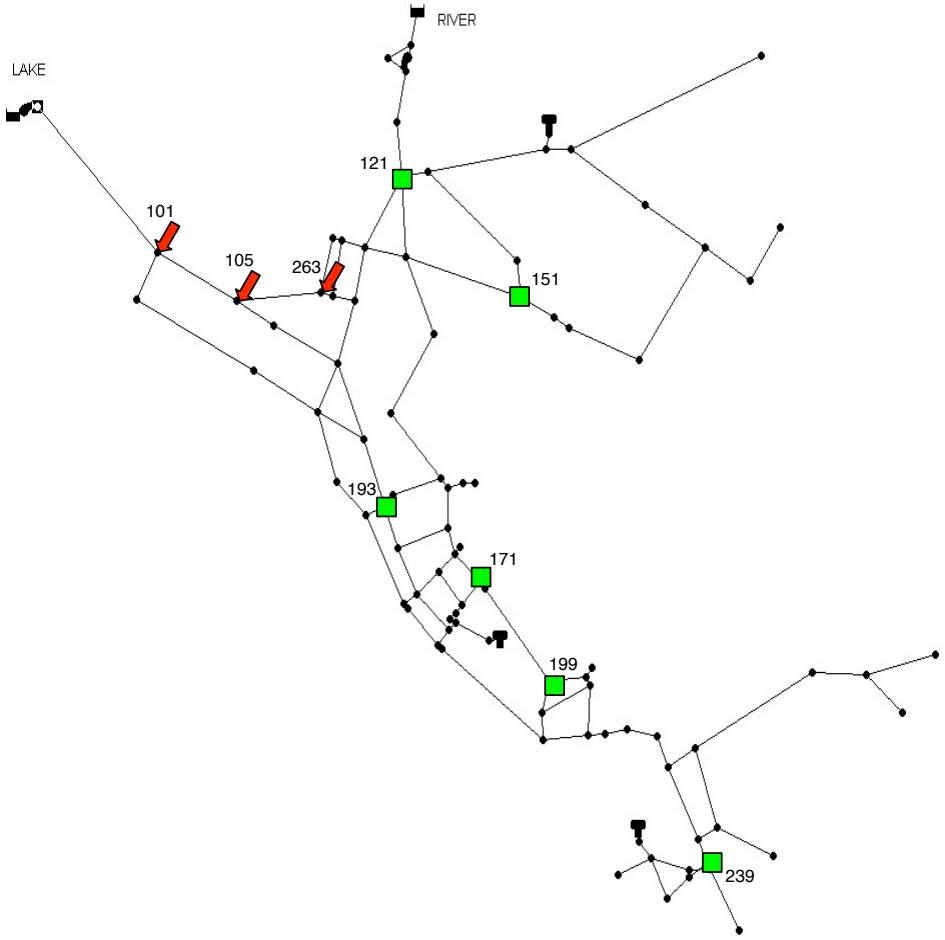

Figure A.3: Sources identified in solution for sensitivity analysis run 1.1.

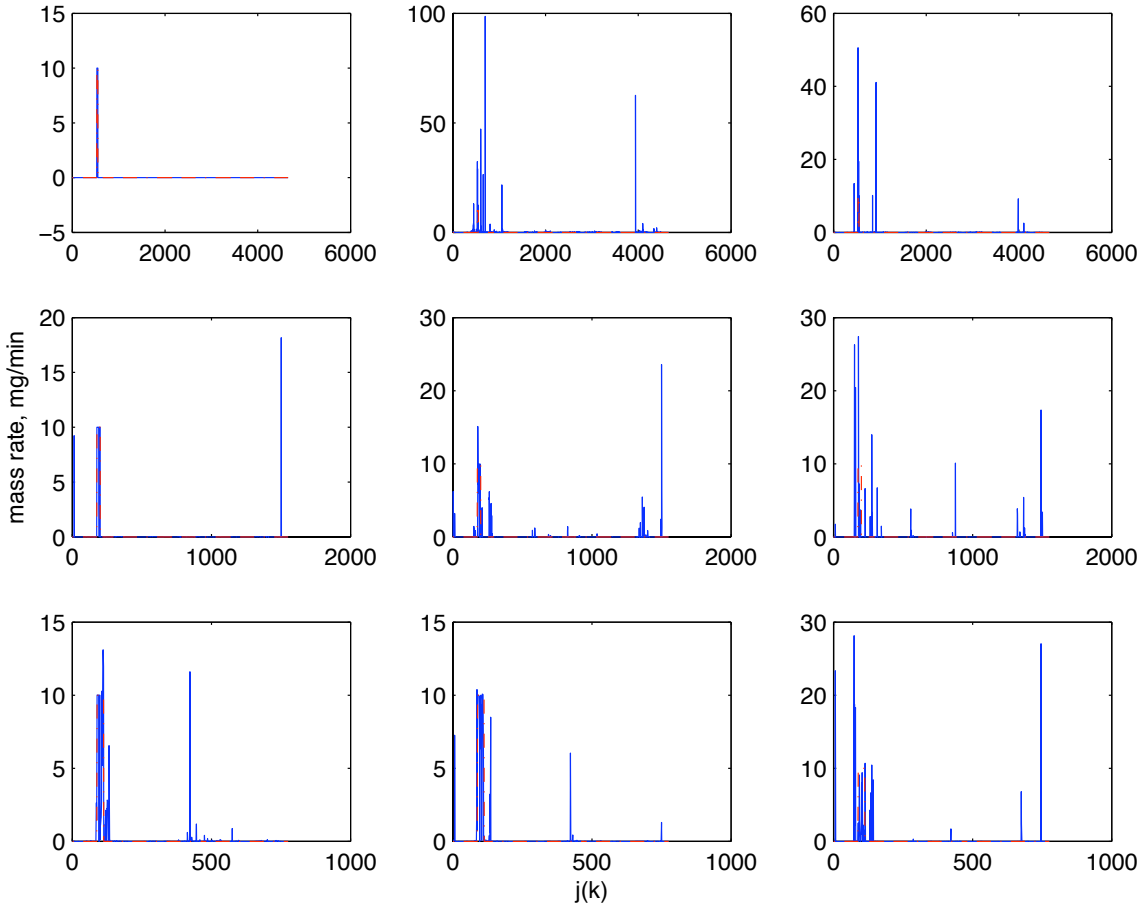

Figure A.4: Sensitivity analysis, sampling frequency and source discretization. Source identification solution shown in blue, true solution in red.

right, the three largest spikes in the solution vector plot correspond to nodes 101, 105, and 263 respectively on the network map. The large gap in the solution between the second and third peaks results from the indexing of the nodes when the network model was created. Hence, the solution exhibits a tight spatial clustering, despite appearing to the contrary in solution vector plot itself. Indeed, all of the solutions in Figure A.2 are similarly clustered, though some are more sparse than others. Given how few sensors were placed in the monitoring designs evaluated in the course of the sensitivity analysis and the degree to which the problems solved were underdetermined these are promising results.

As we observed in the previous set of results, the ratio of rows to columns is determined by the number of sensors located in the network, the sampling frequency, and the source discretization. In this the second part of the sensitivity analysis, the number of sensors and their locations are

unchanged and the sampling frequency and source discretization are modified to examine how the degree to which the problem is underdetermined effects problem solution and conditioning. Again, results of the sensitivity analysis are summarized in Table A.2 and illustrated in Figure A.4.

Figure A.4 is another 3x3 composite plot where the sampling frequency is reduced from 6 to 1 (1/hrs) from left to right and the source discretization is reduced from 6 to 1 (1/hrs) from top to bottom. Thus, the plot in the upper right corner is the solution for the source identification problem which was the most underdetermined and the plot in the lower left is the solution for the least underdetermined problem. Generally, the results indicate that solution quality did not improve uniformly as the degree to which the problem was underdetermined was reduced.

Viewing the results in Figure A.4 from left to right, the sampling frequency is reduced and a corresponding degradation in solution quality is observed. This result was anticipated. Reduction in the sampling frequency increases the degree to which the problem is underdetermined and thus the problem lacks sufficient quantity of data necessary for solution. Interpreting the results further, the sampling frequency is, in part, determined by the sensor technology deployed in the monitoring network. If sensor response is slow because of fundamental technological limitations, it may be difficult to produce useful source identification results using the solution techniques employed herein.

Considering the results from top to bottom, the source discretization is reduced and fewer contaminant injection terms require estimation. Corresponding to a reduction in the degree of underdeterminedness. This did not lead to an improvement of solution quality as was anticipated. Indeed, in most cases the solutions became more non-sparse as the number of injection terms was reduced. But why is this the case? It may be that a reduction in the degree of underdeterminedness does not necessarily lead to an improvement in problem conditioning.

Poor conditioning may be a contributing factor in other irregularities observed in the sensitivity analysis solutions. As mentioned previously, basis pursuit by LP is founded upon a correspondence between the L_0 and L_1 objective formulations of the problem. Inspection of the results, see Table A.2, indicates that a break down in this correspondence has occurred in several of the runs. Instances where the minimum L_0 norm does not correspond to the minimum L_1 norm and where the true solution is not recovered, include runs 1.1 and 2.5. This break down in objective correspondence is attributable to problem conditioning.

Further, the results for runs 1.1 and 2.5 imply that the solutions may not be unique. This was confirmed by inspecting the Lagrange multipliers of the problem constraints. All of the solutions of the sensitivity analysis runs contained some λ values equal to zero for the lower bound constraint. Injection terms where the relaxation of the lower bound constraint would have no effect on the objective function have $\lambda = 0$. These injection terms most certainly have projections into the null space of the forward model matrix. Using the results for sensitivity analysis run 1.1, this is illustrated

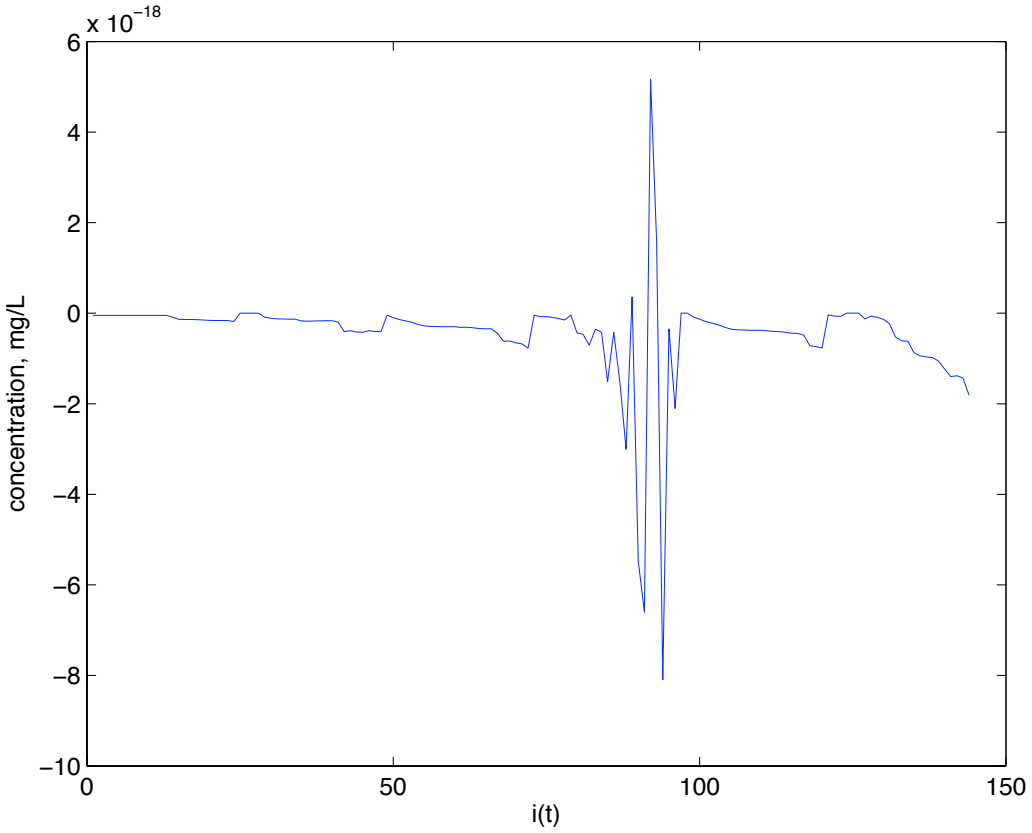

Figure A.5: Projection of the solution error for sensitivity analysis run 1.1 into the null space of the forward model matrix.

in Figure A.5. The figure was created by subtracting the recovered solution from the true solution and projecting the difference through the forward model matrix. Noting the scale of the y-axis, it is clear that the solution errors on the order of 1.0E10-18 are for all intensive purposes undetectable. Thus, basis pursuit does not provide a guarantee of solution uniqueness as Tikhonov regularization does.