EURO Journal on Computational Optimization manuscript No.
(will be inserted by the editor)

Applying Ranking and Selection Procedures to Long-Term Mitigation for Improved Network Restoration

Emily A. Heath · **John E. Mitchell** · **Thomas C. Sharkey**

Received: date / Accepted: date

Abstract In this paper we consider methods to determine the best single arc mitigation plan for improving rapid recovery of a network with a given level of statistical certainty. This problem is motivated by infrastructure managers interested in increasing the resilience of their systems through costly long-term mitigation procedures. Our problem is two-stage, where we consider a small number of pre-event decisions for mitigation, with a large second-stage integer programming problem to capture the restoration process for each damage scenario and each mitigation plan. We consider a ranking and selection (R&S) procedure and compare its performance against a brute force method using standard statistical testing on problems with low, medium, and high damage levels. These comparisons are made by using the same number of integer programs for each method, and comparing the level of confidence achieved to determine a best single arc mitigation plan. We find that R&S procedures perform as well or better than brute force procedures in all cases, and significantly outperform the brute force procedure in almost all cases (five out of six). Having developed a general framework for determining the best single arc mitigation plan for any network, we conclude with thoughts and challenges on how this framework can be expanded and applied to different problems.

Keywords ranking and selection · network resilience · mitigation · restoration

The work of these authors was supported in part by the U.S. National Science Foundation under grants number CMMI-1314350 and CMMI-1254258.

Emily A. Heath
Department of Mathematical Sciences
Rensselaer Polytechnic Institute
110 8th St., Troy, NY 12180
Tel.: +1203-379-7622
Fax: +1518-276-4824
E-mail: heathe@rpi.edu

John E. Mitchell
Department of Mathematical Sciences
Rensselaer Polytechnic Institute
110 8th St., Troy, NY 12180

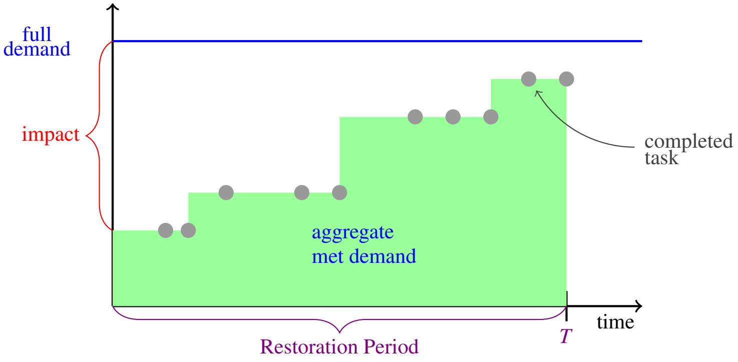
Thomas C. Sharkey
Deparment of Industrial and Systems Engineering
Rensselaer Polytechnic Institute
110 8th St., Troy, NY 12180

1 Introduction

Network resilience is an important topic for managers of critical infrastructures and supply chains, and it is particularly important in light of recent natural disasters, like Hurricane Sandy, that caused catastrophic damage in New Jersey and New York in the United States. The U.S. Department of Energy Office of Electricity Delivery & Energy Reliability (2012) reported that as of the morning of November 7, 2012, seven to eight days after the hurricane hit, 650,416 customers were still without power in the affected states in the mid-Atlantic and Northeast [14]. The damage also impacted several petroleum and natural gas refineries, whose reduced operating capacities disrupted the supply of these products to consumers and left 24% of gas stations in the New York Metropolitan area without gasoline for sale as of November 7, 2012 [14]. Improving the resilience of these affected networks is an important step not only in restoring services more rapidly, but also in reducing the impact caused by another natural disaster. In the U.S. Presidential Policy Directive on Critical Infrastructure Security and Resilience, resilience is defined as "the ability to prepare for and adapt to changing conditions and withstand and recover rapidly from disruptions. Resilience includes the ability to withstand and recover from deliberate attacks, accidents, or naturally occurring threats or incidents [37]." This paper considers how to improve network resilience through long-term mitigation activities and focuses on the rapid recovery of the network from an event.

For a graphical representation of the resiliency measure used in this work, consider the following graph:

Fig. 1 Resiliency Measures



In this graph, the event occurs at time 0 and causes a disruption in the services so that full demand can no longer be met. Full demand is labeled on the vertical axis and marked with the straight line. The immediate drop in the amount of met demand is labeled with a brace as "impact", and can be viewed as the immediate impact. The filled-in portion of the graph labeled "aggregate met demand" indicates the amount of met demand in the network over time. Improvements in the met demand are made in steps, as restoration activities are performed that recover parts of the network. Completed restoration activities are the circles, one of which is labeled as "completed task" for an example. The restoration period is labeled with a brace along the horizontal axis, and covers the time of the event until a fixed time horizon T. There is a substantial body of research and literature that focuses on minimizing the immediate impact to the network from a disruption. In the graph, this type of focus can be interpreted as minimizing the "impact" brace over a variety of scenarios. This type of focus can help improve the resiliency of the network, but it does not consider all aspects of network resiliency. In this paper, the focus is on maximizing the met demand in the network over a fixed restoration time period, i.e. until time T. Note that this restoration period does not necessarily cover the time until full demand is met again. In Cimellaro et al. (2010), the authors define the recovery time as the time needed to restore the functionality of the system to a desired level. As the authors point out, many managers may use the recovery time from a disruptive event to improve their overall network to obtain higher levels of performance than before the event. In this work, the restoration time period is designed to capture only the recovery time associated with the existing network, and not the transition to rebuilding and redesigning the network. This focus corresponds to maximizing the filled-in area in the graph, which covers this restoration time period. This type of resiliency measure covers not only the immediate impact of the event but also the evolution of the state of the network as it recovers from the event. To the best of the authors' knowledge, this type of resiliency measure, i.e. rapid recovery, has not been used for network mitigation prior to this paper.

Mitigation refers to activities that are undertaken before a disruption occurs and are designed to reduce the impact of any expected damage. There are short-term and long-term mitigation activities. An example of a short-term mitigation activity is the pre-positioning of supplies or work crews to a critical area likely to be damaged by an event. An example of a long-term mitigation activity is burying a power line in an electrical network or locating an industrial back-up generator at a production facility or warehouse in a supply chain. Mitigation decisions contribute to a network's resilience by improving its ability both to withstand and recover from disruptions. Short-term mitigation activities will improve the resilience of the network for one particular event, while long-term mitigation activities will improve the resilience of the network for many different events. This paper focuses on long-term mitigation decisions, as these decisions involve high costs that prohibit managers from being able to undertake several long-term activities at once. For example, consider the decision to bury a power line. The U.S. Energy Information Administration has found that on average, underground lines can cost five to ten times more than overhead lines, which does not include the cost of dismantling the system if an existing overhead line is selected to be buried [15]. As a result, utility managers consider burying an entire power system to be cost prohibitive, but they may be able to select a few key components to bury. Similarly, locating an industrial generator is also a very expensive long-term mitigation decision for managers of a supply chain network to make. These generators are expensive themselves, and they also have maintenance costs that can be very high. In this paper, we consider these types of long-term mitigation decisions that have very high costs but can contribute to the resilience of the network in a significant way.

The focus in this paper is on how mitigation can improve the rapid recovery of the network over time, where recovery is measured through restoration decisions. Restoration decisions are designed to recover a system from an event by repairing damaged components of a network and bringing services back to affected areas. Examples of restoration decisions include sending work crews to clear roads with downed trees or to repair damaged electrical components like downed power lines or impacted substations. In a supply chain network, examples of restoration decisions include testing and assessing equipment for damage as well as making repairs on equipment. Restoration decisions contribute to a network's resilience by improving how rapidly disrupted services recover. Focusing on mitigation or restoration alone can help to improve the resilience of a network, but considering these two aspects together can allow managers of critical infrastructures and supply chains to make better mitigation decisions that impact the restoration process.

Decisions on mitigation affect the restoration by either eliminating the need to restore certain components of the network, or by improving the speed with which restoration activities take place. A natural question arises: what mitigation decisions can a manager make that will most improve the restoration of the network over time? This paper develops a method for determining with statistical significance the best long-term mitigation decision to completely protect a single component of the network. Statistical significance is established by evaluating the contribution of the mitigation plans to the restoration of the network over time for a given damage scenario, and sampling across several different damage scenarios. The mitigation plans are ranked according to their individual contributions so that the best plan is selected with probability $1-\alpha$, where α is the user-defined level of statistical significance. This paper is novel in its development of a performance measure and procedure that accomplishes two things not seen in previous work on network mitigation: first, it determines the impact of mitigation decisions on the large-scale second-stage integer program of the restoration of the network over time; and second, it selects the best mitigation plan out of several with a given level of statistical certainty.

The remainder of the paper is organized as follows: Section 2 surveys related literature and discusses how the work in this paper differs. Section 3 describes the problem formulation and gives the general framework used for testing mitigation plans. In Section 4, the data set used in this paper and the implementation details are described, and results are presented. Section 5 discusses how this framework can be used to consider sequential mitigation decisions, and presents results for the same data set. The paper concludes with Section 6 with remarks on the performance of the general framework and how it can be improved and expanded for future work.

2 Literature Review

As such an important issue, improving network resilience by means of mitigation and restoration has been the subject of much research. Previous work in network restoration points to the use of an integer programming (IP) formulation and IP algorithms for the problem, see Cavdaroglu et al. (2013), Lee et al. (2007), Matisziw et al. (2010), and Nurre et al. (2012) for examples. For a recent overview of network restoration literature, see Nurre and Sharkey (2014). These problems involve determining a schedule for repair tasks and assigning a work crew to complete these tasks. Integer variables are required to capture the scheduling decisions, work crew assignments, and network operations. These formulations are also frequently time-indexed, with a restoration time horizon over which these tasks are completed, so the integer variables must be indexed for each time period to capture the restoration process. These restoration problems are often NP-complete, see Nurre and Sharkey (2014) for proofs of certain classes. They are, therefore, difficult to solve even for relatively simple restoration problems. Time-indexed integer restoration models, although difficult to solve, are better able to accurately capture the restoration process for networks. The restoration models cited in the above papers, however, are not integrated with mitigation decisions. One contribution of this paper is to integrate restoration models like the ones discussed with models that also consider mitigation decisions.

A common approach for considering network mitigation or restoration is to use a mathematical model that considers the services provided by them as flows in the network. For an overview of network flows, the reader is invited to consider Abuja et al. (1993). With a network flow approach, a two-stage stochastic programming model is one method used to solve problems in network mitigation. For a review of stochastic programming, see Birge and Louveaux (1997). Some examples of two-stage stochastic programming applied to problems in network mitigation can be seen in Barbarosoglu and Arda (2004), Hentenryck et al. (2010), Mete and Zabinsky (2010), Rawls and Turnquist (2010), Salmon and Apte (2010), and Shen (2013). One feature of these two-stage stochastic programming approaches is that they consider a relatively simple second-stage problem focused on the immediate state of the network after the disruption occurs. In reference to Figure 1, these types of approaches focus on minimizing the brace labeled the "impact." These approaches determine a mitigation plan that will allow the network to better withstand a disruption. Additionally, these approaches also have second-stage decision variables that are typically continuous, which keeps the problem size from increasing to a level where it is computationally intractable. In contrast, the second-stage problem in this research is to consider the restoration process of the network over a given time horizon.

The second-stage problem in this work involves many integer variables, and is difficult to solve in a two-stage stochastic programming model with a large number of scenarios. There has been previous work that considers how to achieve stochastic convergence, the most popular of which involves Monte Carlo simulation and the sample average approximation (SAA) technique, see Kleywegt et al. (2002). Unlike the method in this paper, the SAA approach is unable to provide a statistical guarantee of selecting the best solution from many candidates (Kleywegt et al. (2002)). The SAA approach is better suited for problems where the first-stage problem is complicated and the second-stage problem is continuous, see Chang et al. (2007) for an example of its application to this type of problem. This paper, however, considers a complicated second-stage problem, the restoration of the network over a given time horizon, and focuses on being able to select the best mitigation plan from many candidates with statistical certainty.

Additional areas that have considered topics related to mitigation and restoration include work in network interdiction. These problems allow one to test the current resilience of the network against a "smart" disruptor. In interdiction research, the focus is on selecting the components of the network to damage in order to disrupt services. Much of the research in interdiction has focused on "maximum flow interdiction," where the interdictor (agent causing damaging to the network) seeks to minimize the maximum possible flow in the network. Wood (1993) formulates an IP model for the maximum flow interdiction problem, and shows it to be NP-complete. Cormican et al. (1998) extend this maximum flow interdiction problem to stochastic variants involving uncertainty in arc capacities. Additional relevant papers on interdiction include Church et al. (2004) and Losada et al. (2012). Interdiction work is important in understanding the current vulnerabilities of a network, but it is important to note that it is not directly applicable to mitigation on its own. Scaparra and Church (2008) consider the problem of selecting a subset of facilities to protect in order to minimize an interdiction strike, and point out that the subset of facilities to protect is not equal to the subset of facilities that would be selected for interdiction alone. Although papers like Scaparra and Church (2008) consider fortifying components of the network, the primary focus is still on maximum flow interdiction problems and the immediate state of the network following the attack. To relate these approaches back to Figure 1, they still emphasize minimizing the “impact” brace. Therefore, these works still do not consider a complicated second-stage problem of the restoration of the network as the work in this paper considers.

Another related area of research is defender-attacker-defender (DAD) work. For a tutorial on DAD models, the reader is invited to consider Alderson et al. (2014). In DAD models, a tri-level problem is considered. In the first stage of the model, the defender selects components of the network to protect. In the second stage of the model, the attacker observes the fortifications and selects components of the network to damage. In the third stage of the model, the defender determines how the network will operate based on the damage. The DAD model is yet another method for determining how to improve the resilience of a network, see Alderson et al. (2011) for example. However, the approach in DAD models still focuses on a relatively short time horizon after the attack/disruption, and does not consider the restoration of the network over time as this paper considers. As in other related areas, this focus can be interpreted as minimizing the impact of the event, labeled with the “impact” brace in Figure 1. Additionally, some DAD work is more concerned with designing a new network, rather than protecting existing components of the network (see Smith et al. (2006) for an example). In this paper, the design of a new network or the addition of new components is not considered because the focus is on determining how the manager can make a single mitigation decision to improve the resilience of the current network.

One final area of related work to consider is facility location research. The facility location problem is well-studied, and primarily addresses how to design a new network or place new facilities. Daskin (1995) is a comprehensive text on the subject, and Owen and Daskin (1998) is a popular comprehensive review paper on the subject. For more recent reviews on discrete and stochastic facility location work, the reader is directed to Current et al. (2004) and Snyder (2006) respectively. As mentioned previously, facility location work focuses on how to improve network resilience through design considerations. While these considerations are relevant in cases where a manager is planning a new network or seeking to add components to an existing network, these cases are not the focus of this paper. Instead, this paper is concerned with improving the resilience of a network by protecting an existing component that will contribute to the rapid recovery after an event.

In light of the differences in focus between these related areas of research and the question this paper seeks to address, this work utilizes a ranking and selection (R&S) procedure. The use of a R&S procedure is motivated by the desire to be able to select the best mitigation plan with a given level of statistical certainty. As described previously, the mitigation plans being considered in this paper are single long-term investments in the network. These long-term investments come with substantial costs, so it is important that a manager have confidence in the decision made. R&S procedures offer a statistical guarantee on selecting the best system, and are well-suited for problems where the number of systems being considered is large. In this paper, the systems being tested are the mitigation plans for the network, and they are evaluated using a performance measure constructed to determine the best mitigation plan for increasing the rapid recovery of the network. Evaluating a system according to a performance measure is referred to as sampling a system, and R&S procedures are designed to work well when this sampling is computationally expensive. In this work, the performance measures for mitigation plans will require the solution of large integer programming problems, and is therefore computationally expensive to compute. R&S procedures have a user-specified guaranteed level of statistical certainty, meaning they will select the best or a group of the best systems with a given probability. For a broad overview of R&S procedures and the underlying mathematics, see Kim and Nelson (2006). These procedures work generally by first sampling each system a fixed number of times, referred to as the first-stage sample size. A screening procedure is next used to eliminate some of these competing systems. Typically, a R&S procedure is a two-stage procedure or a sequential procedure, where the stages refer to the number of times an individual system may be sampled. If the procedure is two-stage, it will conclude by allocating a second-stage sample size to remaining systems and using a selection procedure to determine the best or a subset of the best systems. If the procedure is sequential, sampling and screening may continue over many rounds until a stopping criterion is met whereby selection of the best can take place. For examples of sequential procedures, see Frazier (2013), Kim and Nelson (2001), or Pichitlamken et al. (2006). For examples of two-stage procedures, see Kim and Nelson (2006) or Nelson et al. (2001). The advantage of R&S procedures is that they are designed to minimize computationally expensive sampling while still ensuring that the best system can be determined with a given level of statistical certainty. For these reasons, R&S procedures appear to be well-suited to determine the best mitigation plan for contributing to the rapid recovery of a network with a given statistical guarantee. The next section outlines the exact procedure used to test and select a best mitigation plan.

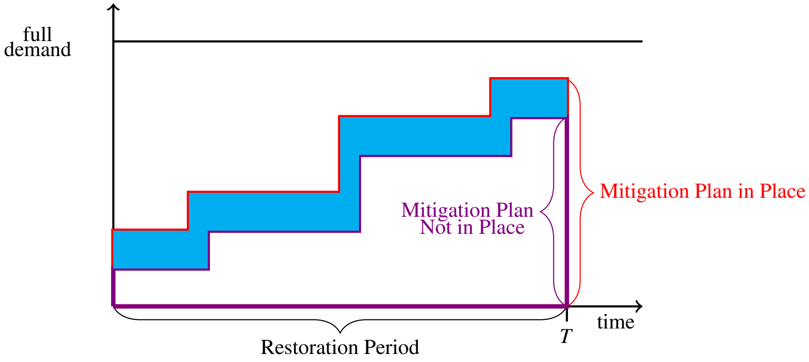
3 Problem Formulation and Solution Approaches

This paper addresses how to determine the best mitigation plan for contributing to the rapid recovery of a network over time. First, it is important to determine the mitigation plans being considered. As mentioned previously, the focus of this work is on selecting one component of the network to protect, as this protection decision is a long-term investment that has high costs. To model the protection of one component in the network, the set of possible mitigation plans being considered is an appropriate set of arcs. Here, the term “appropriate” will refer to any set of components a manager may consider as reasonable choices for protection, which can include arcs that are not currently part of the network. Additionally, it is important to note that this set of arcs is not limited to actual arcs in the network but can include arcs that represent nodes as well. Any node can be modeled as an arc by using a standard network expansion technique that splits the node into one arc and two nodes (see Nurre et al. (2012)). In the following subsections, we define how we will measure the impact of using a particular mitigation plan and the R&S procedure used to test mitigation plans.

3.1 Problem Formulation: INDS Framework

Given the set of possible mitigation plans, the term “best” must be defined. In order to prescribe a performance measure to a mitigation plan, restoration measures in integrated network design and scheduling (INDS) problems, as described in Nurre et al. (2012), are considered. Since the focus of this work is on considering how a mitigation plan contributes to the ability of a network to recover rapidly, the performance measure chosen is the improvement in the restoration of the network over time given that mitigation plan for a damage scenario. For a given mitigation plan, a single performance measure is found with the following procedure: First, a damage scenario is generated, which can be done in a problem-specific or application-specific way, by determining a set of damaged arcs. The arc considered for mitigation is not in the set of damaged arcs, as the set of damaged arcs is selected from all remaining arcs. Next, the INDS problem is solved twice, with a small change in the set of the damaged arcs. The INDS problem is first solved with the original set of damaged arcs, where the arc considered as a mitigation plan is not part of that set. This INDS problem corresponds to having the mitigation plan in place. It is then solved again, where the arc considered as a mitigation plan is part of the set of damaged arcs. This INDS problem corresponds to not having the mitigation plan in place. In each solution for the INDS problem, the objective function value gives the restoration performance from that particular damage scenario. The performance measure is the difference between these two objective function values, specifically the objective function value for the INDS problem where the mitigation plan is in place minus the objective function value for the INDS problem where the mitigation plan is not in place. For illustrative purposes, consider the following figure:

Fig. 2 Performance Measure Illustration



In this illustration, the area enclosed by the top line denotes the objective function value for the INDS problem with the mitigation plan in place, labeled in part by the brace "Mitigation Plan in Place." The area enclosed by the lower line denotes the objective function value for the INDS problem with the mitigation plan not in place, labeled in part by the brace "Mitigation Plan Not in Place." The performance measure developed in this work takes the difference between these two objective function values. In this illustration, this difference corresponds to the filled-in portion. The objective function value for the INDS problem where the mitigation plan is in place at least as large as the objective function value for the INDS problem where mitigation plan is not in place because the set of damaged arcs is otherwise the same. Therefore, the performance measure is non-negative, and it determines the improvement in the restoration of the network when the mitigation plan is in place for that damage scenario. A high performance measure indicates that using a particular mitigation plan has a significant contribution to how well the network can be restored for that damage scenario. In this illustration, the particular mitigation plan does have a positive contribution to how well the network can be restored for this damage scenario, so its performance measure is a positive value that corresponds to the filled-in area. Following the INDS formulation, the performance measure will be made concrete.

Before giving the complete formulation of the INDS problem, we first define the general features of the problem. Let $G = (N,A)$ denote the original network and any arcs considered for mitigation that may not be part of the network, where N is the set of nodes and A is the set of all relevant arcs in the network. All arcs in the network are directed, so an arc (i, j) denotes a connection from node i to node j. Without loss of generality, we will assume that damage to the network damages arcs only, as it has already been explained that only arcs are protected and nodes can be represented as arcs using a standard network expansion technique. Let T be the number of time periods in the horizon, which can be tailored to the specific problem or application. Let $S \subseteq N$ be the set of supply nodes, each with a supply s_i. Let $D \subseteq N$ be the set of demand nodes, each with a demand d_i. Each arc (i, j) has a capacity u_{ij}, which denotes the maximum amount of flow that can be carried from node i to node j, and a processing time p_{ij} that denotes how long it would take for the arc to be repaired if it were damaged. In the case of arcs that are not part of the original network, their processing time is longer than the time horizon of the problem. Each unit of flow to a demand node $i \in D$ has weight w_i. Let μ_t be the weight of the performance of the network at time t. The network flow variable x_{ijt} gives the flow on arc (i, j) at time period t. The demand met variable v_{it} gives the amount of demand met at node $i \in D$ in time period t.

With these general features of the problem, we can now define the features of the problem specific to a damage scenario. Let $\mathscr{S}$ denote a particular damage scenario. Let $\bar{A}(\mathscr{S}) \subseteq A$ denote the set of damaged arcs. The arcs in $\bar{A}(\mathscr{S})$ are arcs from the original network that have been damaged in $\mathscr{S}$ and are unable to carry any flow. $\bar{A}(\mathscr{S})$ will also include all arcs considered for mitigation that are not part of the original network, as these arcs are not available in the network.To repair damaged arcs, there are K identical work groups, denoted by the subscript k. A work group can be thought of as any resource that can be deployed to repair arcs in the network. In this INDS formulation, the travel time for a work group to move from one component of the network to the next is not explicitly included, see Nurre and Sharkey (2015). The definition of a time period in the INDS formulation is problem and application specific, and can therefore implicitly consider travel time. There are two sets of binary variables that relate the set of arcs in $\bar{A}(\mathscr{S})$ and the work groups. The first set of binary variables, denoted by β_{ijt}, are used to indicate whether an arc (i,j) from the set $\bar{A}(\mathscr{S})$ is available in the network at time t. A value of 1 indicates that the arc is now available in time period t, and a value of 0 indicates it is not yet available. The "first" β value of 1 for an arc in $\bar{A}(\mathscr{S})$ indicates the first time period at which it becomes available, meaning its first β indicates when its restoration was completed. The second set of binary variables, denoted by γ_{kijt}, are used to indicate whether a work group k has completed an arc (i,j) in $\bar{A}(\mathscr{S})$ at time t. A value of 1 indicates the work group number k restored the arc (i,j) in time period t. It is important to note that these sets of binary variables, β and γ are only defined for arcs in $\bar{A}(\mathscr{S})$. With these parameters and variables, the INDS problem formulation is as follows:

$$\phi(\bar{A}(\mathscr{S})) := \max \sum_{t=1}^{T} \sum_{i \in D} \mu_t w_i v_{it}$$

$$\text{s.t.} \sum_{(i,j)\in A} x_{ijt} - \sum_{(j,i)\in A} x_{jit} \le s_i, \qquad \forall i \in S, t = 1,...,T \quad (1)$$

$$\sum_{(i,j)\in A} x_{ijt} - \sum_{(j,i)\in A} x_{jit} = 0, \qquad \forall i \in N \setminus \{S \cup D\}, t = 1,...,T \quad (2)$$

$$\sum_{(i,j)\in A} x_{ijt} - \sum_{(j,i)\in A} x_{ijt} = -v_{it}, \qquad \forall i \in D, t = 1,...,T \quad (3)$$

$$0 \le v_{it} \le d_i \qquad \forall i \in D, t = 1,...,T \quad (4)$$

$$0 \le x_{ijt} \le u_{ij} \qquad \forall (i,j) \in A, t = 1,...,T \quad (5)$$

$$0 \le x_{ijt} \le u_{ij}\beta_{ijt} \qquad \forall (i,j) \in \bar{A}(\mathscr{S}), t = 1,...,T \quad (6)$$

$$\sum_{(i,j)\in \bar{A}(\mathscr{S})} \sum_{r=t}^{\min\{T, t+p_{ij}-1\}} \gamma_{kijr} \le 1 \qquad \forall k = 1,...,K, t = 1,...,T \quad (7)$$

$$\beta_{ijt} - \sum_{r=1}^{t} \sum_{k=1}^{K} \gamma_{kijr} \le 0 \qquad \forall (i,j) \in \bar{A}(\mathscr{S}), t = 1,...,T \quad (8)$$

$$\sum_{t=1}^{p_{ij}-1} \beta_{ijt} = 0 \qquad \forall (i,j) \in \bar{A}(\mathscr{S}) \quad (9)$$

$$\sum_{k=1}^{K} \sum_{t=1}^{p_{ij}-1} \gamma_{kijt} = 0 \qquad \forall (i,j) \in \bar{A}(\mathscr{S}) \quad (10)$$

$$\alpha_{kijt}, \beta_{ijt} \in \{0,1\} \qquad \forall (i,j) \in \bar{A}(\mathscr{S}), k = 1,...,K, t = 1,...,T \quad (11)$$

The objective of this INDS problem is to maximize the weighted met demand in the network over the time horizon, so it captures how restoration efforts improve the state of the network over time. It is based on the given damage scenario, so it is denoted by $\phi(\bar{A}(\mathscr{S}))$. Constraints (1), (2), and (3) are flow balance constraints for supply, transshipment, and demand nodes respectively. Constraints (4) give the upper bound for the variable demand met to be no greater than the actual demand at that node. Constraints (5) limit the flow on each arc in the network to be no greater than its capacity. Constraints (6) are the analog of constraints (5) for the set of damaged arcs, where the flow must be 0 if the arc is not available and is constrained by its capacity if it is available. Constraints (7) ensure that no more than one task is being processed by each work group in time period t. Constraints (8) make sure that an arc is not considered available until after it has been completed by a work group. Constraints (9) and (10) are logical constraints that prevent an arc from being available in a time period earlier than its processing time. Constraints (11) give the binary constraints on the appropriate variables. Since the objective function in this formulation gives the maximum weighted met demand in the network over the time horizon, this formulation is used to provide a measure of the restoration performance in the network.

Using the INDS problem formulation, we can now formally give the definition of a performance measure for an arc considered as a mitigation plan. All relevant arcs in the network are in the set A. Let the arc considered for mitigation be denoted by a_1. In line with the INDS formulation, a particular damage scenario is denoted by $\mathscr{S}$ and the set of damaged arcs is denoted by $A'(\mathscr{S})$. The original set of damaged arcs will not include a_1, so the damaged arcs are selected from the remaining arcs. Recall also that the set of damaged arcs must also include any arcs in A that are not part of the original network and not being considered for mitigation. Let the objective function values for the different solutions to the INDS problem by denoted by $\phi(A'(\mathscr{S})) = f_1$ and $\phi(A'(\mathscr{S}) \cup a_1) = f_2$. The performance measure, denoted by PM_{a_1} is given by the following:

Algorithm 1 Performance Measure

```
1: Generate damage scenario $\mathscr{S}$ and select damaged arcs $A'(\mathscr{S})$, where $A'(\mathscr{S}) \subseteq \{A \setminus a_1\}$
2: Solve INDS with $A'(\mathscr{S})$, obtain objective function value $\phi(A'(\mathscr{S})) = f_1$
3: Solve INDS with $A'(\mathscr{S}) \cup a_1$, obtain objective function value $\phi(A'(\mathscr{S}) \cup a_1) = f_2$
4: Performance measure for $a_1$, $PM_{a_1} = f_1 - f_2$
```

3.2 Solution Approaches: GSP Framework

Given the description of the performance measure and how it is used with the INDS formulation, we next discuss the testing procedure used for mitigation plans. This procedure is a R&S procedure from Nelson et al. (2001), called the Group Screening Procedure (GSP). As GSP is discussed in detail in Nelson et al. (2001), we will only provide a brief overview of it and give its full steps in Algorithm 2. The first steps of GSP are to select and determine several parameters, a few of which we will describe here. First, the overall level of significance α is determined by the user. α must then be split into α_0 and α_1 such that $\alpha_0 + \alpha_1 = \alpha$. α_0 is used to determine the test statistic t, which impacts the overall test statistic in the procedure for two mitigation plans i and j, denoted by W_{ij}. The size of α_0 therefore impacts **Testing** in Algorithm 2. α_1 is used to determine the parameter h, which impacts the number of second-stage performance measures that must be computed for mitigation plans. GSP is a two-stage procedure where all mitigation plans will have a fixed first-stage number of performance measures computed, but the second-stage performance measures are allocated only to mitigation plans that survive testing. The number of these second-stage performance measures is determined in part by h. The size of α_1 therefore impacts **Sample 2** in Algorithm 2. The split of α to α_1 and α_0 is one of the aspects of GSP that can be used to minimize the number of performance measures that must be computed for mitigation plans, and it is discussed in more detail in Section 4. Along with determining the overall level of significance, the user also determines the indifference zone (IZ) parameter. The IZ parameter is the user-specified level of practical significance. It is the amount by which the average performance measure for mitigation plans must vary in order to be considered different. The final important decision made by the user is to group the mitigation plans, denoted by $G_1, G_2, ...$ Mitigation plans can be partitioned into any number of groups, so long as the first group has at least two members. Empirical results from Nelson et al. (2001) indicate that GSP will minimize the number of performance measures that must be computed by constructing groups so that the mitigation plans most likely to be best are in Group 1. Therefore, the user should not randomly select groups. Group construction is discussed in more detail in Section 4, where we describe how to determine which mitigation plans to put in Group 1.

With all of the appropriate parameters selected and determined, GSP proceeds in roughly two major steps. The first step is to allocate to all mitigation plans their first-stage number of performance measures. This first step is denoted by **Sample 1** in Algorithm 2, and it also requires computing the average and sample variance of performance measures for all mitigation plans. In Algorithm 2, the average of all performance measures computed for a particular mitigation plan i is denoted by $\overline{PM}_i$, and the sample variance is denoted by S_i^2. The second step is to test the mitigation plans in groups, denoted by **Testing**. Group 1 is tested first, and any surviving mitigation plans have a second-stage number of performance measures computed. This allocation of second-stage performance measures is denoted by **Sample 2**. As with **Sample 1**, **Sample 2** involves updating the average of performance measures and the sample variance over all performance measures that have been computed for a mitigation plan (**Sample 1** performance measures and **Sample 2** performance measures). Once the surviving mitigation plans from Group 1 have been allocated second-stage performance measures and had their averages and variances updated, these mitigation plans are added to Group 2. Group 2 is tested, with its additions. Again, new surviving mitigation plans from Group 2 proceed to **Sample 2** while old surviving mitigation plans from Group 1 will be moved to the next group without any additional performance measures. This process continues until all groups have been tested. The current group of mitigation plans being tested is denoted by I^{new} and the group of surviving mitigation plans from previous groups is denoted by I^{old}. At the end, the mitigation plan with the highest average is selected as the best from the final group of survivors. We now present the steps of GSP in Algorithm 2 as it applies to this work.

Algorithm 2 GSP

```
Select overall level of significance $\alpha$, and select $\alpha_0$ for testing and $\alpha_1$ for sampling such that $\alpha_0 + \alpha_1 = \alpha$.
Select the first-stage number of performance measures $n_0 \geq 2$, and number of mitigation plans $p$.
Set $t = t((1-\alpha_0)^{\frac{1}{p-1}}, n_0 - 1)$ where $t(\beta, \nu)$ denotes the $\beta$ quantile of the student $t$ distribution with $\nu$ degrees of freedom.
Set $h = h(1-\alpha_1, n_0, p)$ where $h$ is Rinott's constant.
Select the IZ parameter $\delta$, the user-specified level of practical significance.
Let $G_1, G_2, ..., G_m$ be groups of systems such that $G_1 \cup G_2 \cup ... \cup G_m = \{1,2,...,p\}, G_i \cap G_j = \emptyset$ for $i \neq j, |G_i| \geq 1$ for all $i$, and $|G_1| \geq 2$.
Set $I_0 = \emptyset$.
**Sample 1**: Do the following for all systems $i$:
    1. Repeat Algorithm 1 $n_0$ times to obtain $n_0$ performance measures.
    2. Compute $\overline{PM}_i$ (the sample average) and $S_i^2$ (the sample variance).
    3. Set $\tilde{N}_i = n_0$, where $\tilde{N}_i$ denotes how many performance measures have been computed for system $i$.
Do the following for round $l = 1, 2, ..., m$:
    1. **Testing**: $l$ is the current round of testing, and $G_l$ is the current group of mitigation plans to be tested.
       The test statistic for two mitigation plans $i$ and $j$ is $W_{ij}$, where $W_{ij} = t\left(\frac{S_i^2}{\tilde{N}_i} + \frac{S_j^2}{\tilde{N}_j}\right)^{\frac{1}{2}}$.
       $\tilde{N}_i = n_0$ if system $i$ has only received first-stage sampling and $N_i$ if system $i$ has received second-stage sampling.
       $I^{\text{new}}$ is the set of newly tested mitigation plans that make it into the next screening, while $I^{\text{old}}$ is the set of previously retained mitigation plans that survived another round.
       $I_l = I^{\text{new}} \cup I^{\text{old}}$ where $I^{\text{new}} = \{i : i \in G_l, \overline{PM}_i \geq \overline{PM}_j - (W_{ij} - \delta)^+, \forall j \in G_l$ and $\overline{PM}_i \geq \overline{PM}_j - (W_{ij} - \delta)^+, \forall j \in I_{l-1}\}$
       and $I^{\text{old}} = \{i : i \in I_{l-1}, \overline{PM}_i \geq \overline{PM}_j - (W_{ij} - \delta)^+, \forall j \in G_l$ and $\overline{PM}_i \geq \overline{PM}_j - (W_{ij} - \delta)^+, \forall j \in I_{l-1}\}$.
    2. **Sample 2**: Do the following for all $i \in I^{\text{new}}$:
       (a) Compute the second-stage sample size $N_i$ based on Rinott's procedure and using the sample standard deviation $S_i$:
           $$N_i = \max\left\{ n_0, \left\lceil \left(\frac{hS_i}{\delta}\right)^2 \right\rceil \right\}.$$
       (b) Repeat Algorithm 1 $N_i - n_0$ times to obtain $N_i - n_0$ additional performance measures.
       (c) Update $\overline{PM}_i$ and $S_i^2$ for all performance measures obtained.
       (d) Set $\tilde{N}_i = N_i$.
Select as best the mitigation plan $i \in I_m$ with the largest sample average $\overline{PM}_i$.
```

GSP is proven to have a probability of correct selection greater than or equal to $1 - \alpha$ under the condition that the true mean of the best system is greater than the true mean of the second best system by at least δ, the IZ parameter. The full proof for this claim is in the appendix of Nelson et al. (2001), so we will provide only a sketch. The event that the best system survives **Testing** the first time it is evaluated and is not eliminated by the first-stage sample means of any systems evaluated after it occurs with probably greater than or equal to

$1-\alpha_0$. The event that the best system's sample mean through its second-stage sampling is not eliminated by any other system's sample mean through second-stage sampling occurs with probability greater than or equal to $1-\alpha_1$ because Rinott's constant was used to determine the second-stage sample size. The probability of the joint event is greater than or equal to $1-(\alpha_0+\alpha_1)$. Since we selected α_0 and α_1 such that the sum of these two levels of significance is equal to the overall desired level of significance ($\alpha_0+\alpha_1=\alpha$), we can conclude that GSP has an overall probability of correct select greater than or equal to $1-\alpha$. We will use these general frameworks for the INDS problem and GSP to build a method for determining the best mitigation plan.

4 Data Set and Implementation

We can now discuss how the general INDS problem formulation applies to our research. In our research, we examine the power network for an artificial community, described in Nurre and Sharkey (2015). This artificial community was constructed based on real data, so it falls under the category of "realistic data." Not only does it mimic a power network for a real community, it is also similar to a supply chain network in the sense that we have a 'high-level' network of facilities and each demand point is connected to a single facility. For a complete discussion on how this network is similar to a supply chain network, see Nurre and Sharkey (2015). In the context of this research, we consider it as a power network. This power network has 74 transmission arcs and 775 distribution arcs. Transmission arcs in a power network connect power plants to electrical substations, whereas distribution arcs connect substations to customers. As such, transmission lines are considered more critical to the overall ability of the power network to meet demand. Therefore, the pool of arcs we consider for mitigation are the transmission arcs, and we do not consider any new arcs for the network for mitigation. The following two figures display the network used in this paper; in the first figure, the full network is given and in the second figure, the transmission network is given.

Fig. 3 Full Network

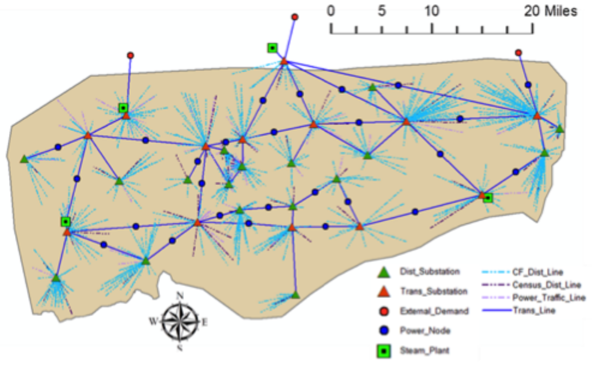
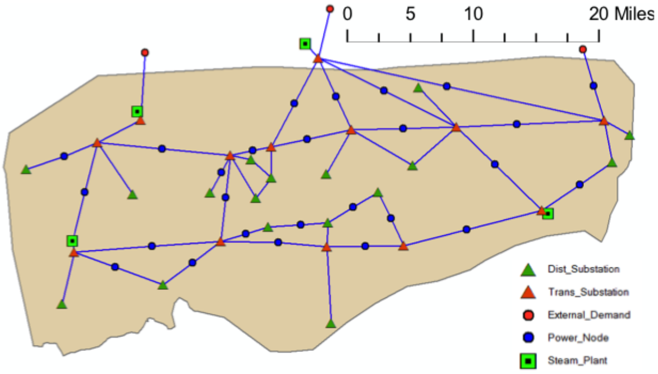


Fig. 4 Transmission Network



For the parameters of our INDS problem formulation, we use $T = 40$ time periods. In this setting, a time period corresponds to six hours, so the time horizon covers 10 days. This time horizon is meant to capture the majority of the restoration activity occurring after a disruption. As described previously, many times after a disruptive event, the network will move into a rebuilding phase. Therefore, by setting $T = 40$, we are considering only the restoration phase and not the rebuilding phase. We use a weight of one for every demand node and a performance weight of one for each time period. For our work, we did not consider any demand nodes to be more important than others, or the performance of the network at any time period to be more important than at other time periods. However, it is reasonable that under different applications these weights would not all be equal, and this method would accommodate those changes. There are three work groups in this work, which was determined based on the size of the network. Every arc has a processing time of two, which is another simplifying assumption that is not required for this method, as many different processing times could be considered in this framework. Additionally, this work does not consider any new arcs as arcs for mitigation plans, so no arcs need to have a processing time larger than 40. As described in the INDS formulation, the set of damaged arcs $\bar{A}(\mathscr{S})$ is determined based on each damage scenario.

We can next discuss our method for damage scenario generation. We consider three damage levels in this work: low (10%), medium (25%), and high (40%). These damage levels are determined based on the immediate loss in met demand caused by the event. Our damage scenarios are generated randomly based on the percentage of damaged arcs in the network that corresponds to desired damage level, and are defined by the arcs in $\bar{A}(\mathscr{S})$. Transmission arcs and distribution arcs are damaged separately. For example, consider a scenario with 20% damage in the network. With a 20% damage level in the network, every transmission arc has a 20% chance of being damaged, and every distribution arc has a 20% chance of being damaged. We round the decimal value of 20% of the 74 transmission arcs and 20% of the 775 distribution arcs to the nearest integer to determine the number of each type of arc that is damaged. As stated previously, however, one of the selected arcs to be damaged cannot be the arc that is currently considered as a mitigation plan. Therefore, this arc is removed from consideration for $\bar{A}(\mathscr{S})$. The arcs to be damaged are selected randomly from the remaining arcs in each group using a Mersenne Twister method. An arc that is selected to be damaged is put in $\bar{A}(\mathscr{S})$. This arc is considered to be nonoperational and unable to carry any flow until repaired. At the end, $\bar{A}(\mathscr{S})$ will consist of all of the arcs that were selected to be damaged (transmission arcs and distribution arcs), and will not include the arc considered for mitigation.

For the implementation of GSP, we used an overall confidence level of 95%, with $\alpha = 0.05$. As described previously in Section 3 and Algorithm 2, α is split between the α_0, which impacts the t statistic used in **Testing** in the procedure, and the α_1 statistic, which impacts the h statistic used in **Sample 2** in the procedure. If more of the 0.05 is allocated to α_0, then the t test statistic will be smaller and **Testing** will be tighter. However, at the cost of this tighter testing, α_1 will be smaller and we will have a larger h statistic, which means larger numbers of second-stage performance measures in **Sample 2**. Conversely, allocating less of α to α_0 will mean a larger t test statistic and looser **Testing**, but a larger α_1 and a smaller h statistic, which means smaller numbers of second-stage performance measures in **Sample 2**. Our primary interest is in reducing the computational effort required, as measured by the number of IPs that must be solved. Every evaluation of a performance measure requires solving two IPs, so we wish to find a good split of α that will reduce this effort. To determine a good split, we first consider GSP with one group, referred to as 1-GSP. In 1-GSP, we perform all of the steps of GSP as denoted in Algorithm 2, but all of the mitigation plans are in Group 1 and there are no other groups. We compute n_0 performance measures for each mitigation plan, as in **Sample 1**. For various splits of α to α_0 and α_1, we first determine the relevant t statistic that impacts **Testing** and the h statistic that impacts **Sample 2**. Next, we use this t for **Testing** with all mitigation plans to determine the number of surviving arcs, and determine the number of IPs that would be needed in **Sample 2** by using h to compute the number of second-stage performance measures for all surviving arcs. We consider the following splits of $\alpha = 0.05$ to (α_0, α_1) respectively: (0.01,0.04), (0.02,0.03), (0.025, 0.025), (0.03, 0.02), (0.02, 0.03), and (0.01, 0.04). Of the splits we consider, we select as a good split the one that requires the minimum number of IPs to be solved in Sample 2. We will use this split in our implementation of GSP because its performance in terms of the computational effort should be at least similar to the results in 1-GSP, which is a result from Nelson et al. (2001).

The remainder of our parameters for our implementation of GSP are as follows: the IZ parameter δ is set to be $22 * 40 = 880$ because the smallest level of demand in the power network is 22, and 40 is the number of time periods we use. We chose this parameter because our performance measure uses the maximum weighted met demand in the network. The IZ parameter is the user-defined level of practical significance, or the smallest difference we care about in the performance of our systems. Since our systems are mitigation plans that should improve the met demand in the network, we define this smallest difference to be between a plan that can meet even the smallest level of demand in every time period and a plan that cannot meet this level of demand. Our first-stage sample size n_0 is 10, and we have $p = 74$ mitigation plans being compared. Our t statistic is found using standard methods, and our h statistic is found using MATLAB code translated from FORTRAN code found in Bechhofer et al. (1995).

As mentioned previously, the empirical results for GSP in Nelson et al. (2001) indicate that computational savings are greatest when the best system is in the first group. Therefore, we wish to construct our groups so that Group 1 contains the arcs that seem most likely to be the best. Previous research by Cavdaroglu (2012) developed a measure of an arc's importance called its modified average starting time (MAST). In solving the INDS problem for a given damage scenario, an optimal schedule is found that determines when damaged arcs are repaired. Using this schedule, we can determine when a damaged arc (i, j) was scheduled for repair by finding the minimum time t where β_{ijt} is equal to 1, and subtracting its processing time. The minimum time t where β_{ijt} is equal to 1 is the first time the arc is operational. It is logical that critical arcs are scheduled to be repaired early in the time horizon, while less important arcs are scheduled later, and arcs that are insignificant may not even be scheduled at all. To determine the MAST for the 74 transmission arcs, we randomly generated 1,000 damage scenarios at a particular damage level and solved one INDS problem for each scenario (one IP per damage scenario). For each damage scenario, we recorded what arcs were damaged. We used the optimal schedule to find when each damaged arc was scheduled to be repaired, and we used a time of timeHorizon+1 as the start time for repair for an arc that was damaged but not scheduled to be repaired within the time horizon. For each of the 74 transmission arcs, we summed all of the arc's starting times and divided by the number of times it was damaged over the 1,000 scenarios to find its MAST. After ranking the arcs according to their MAST from smallest to largest, we formed groups based on roughly similar MASTs. Group 1 contains the arcs with the lowest MASTs, and so should contain within it the most important arc. We will add the 1,000 IPs that were solved to find the MAST to our total number of IPs when we are determining the computational effort of GSP.

In order to evaluate the computational performance of our approach, we compare results with a brute force approach. We are interested in comparing how GSP performs with respect to standard statistical methods. In order to do so, we determine how many performance measures should be computed for each system so that roughly the same total number of IPs are being solved by each method. For each damage level, we determine the total number of IPs that must be solved to use GSP. We take this number of IPs, divide it by the number of mitigation plans, and then divide it by two (since a performance measure requires the solution of two IPs) and take the ceiling. This number will tell us how many performance measures to compute for each mitigation plan. We then use Tukey's Honestly Significant Difference (HSD) Test to determine at what confidence level a single best mitigation plan is found.

Our general plan for a given damage level is as follows: first, we find each arc's MAST in order to form groups. Second, we use 1-GSP to find a good split of α to α_0 and α_1. Next, we implement GSP with this split and these groups to find the best single arc mitigation plan. To compare results, we then use the brute force procedure described in the above paragraph. We compare results between GSP and the brute force procedure by considering the level of confidence achieved in each procedure in order to find one best single arc mitigation plan. We follow this general plan for low, medium, and high damage levels, and the results for each damage level are given in Section 5.

5 Results for Single Arc Mitigation Plans

5.1 10% Damage Results

We first consider what the best single arc mitigation plan is for a low damage level of 10% damage to the network. The average loss in met demand immediately following an event that constitutes 10% damage to the network is around 20%. Therefore, 10% damage to the network corresponds to a relatively low damage level. According to the damage scenario generation method, seven transmission arcs are damaged and 78 distribution arcs are damaged. With three work groups and a time horizon of 40, roughly 71% of the damaged arcs will be operational by the end of the time horizon. Each arc's MAST was computed using 1,000 randomly generated damage scenarios with 10% damage. The results are given in Table 5.1.

Table 1 10% MAST

Group	Number of Arcs	MAST Range
1	26	0 – 7.63
2	26	26.84 – 33.51
3	19	34.17 – 37.62
4	3	39

5.1.1 1-GSP Results: Split of α

As mentioned previously, 1-GSP is used next to determine a good split of $\alpha = 0.05$ to α_0 and α_1. As stated previously, α_0 impacts the t statistic used for **Testing** and α_1 impacts h used in **Sample 2**. With 10% damage, the different splits considered did not impact the number of surviving arcs: three arcs always survived. Therefore, it is best to make α_1 as large as possible to make h as small as possible in order to reduce the second-stage performance measures. The best split is therefore 0.1 to α_0 and 0.4 to α_1. The results for the various splits considered are presented in Table 5.1.1.

Table 2 10% 1-GSP Results

(α_0,α_1)	(t,h)	No. of Survivors	IPs in **Sample 2**
$(0.01,0.04)$	$(5.753,5.99)$	3	1494
$(0.02,0.03)$	$(5.214,6.228)$	3	1618
$(0.025,0.025)$	$(5.047,6.3809)$	3	1702
$(0.03,0.02)$	$(4.911,6.5707)$	3	1806
$(0.04,0.01)$	$(4.701,7.1904)$	3	2176

5.1.2 GSP Results

Following the framework given in Section 3, GSP is implemented using the groups formed by the MAST statistics and the t and h statistics found in 1-GSP. Three arcs survived from Group 1. These three arcs were given their second-stage performance measures, and added with Group 2 for the next round. In round two of **Testing**, only one arc survived, which was a previous surviving arc from Group 1. This arc continued to be the only surviving arc from the additional rounds of **Testing**. Thus at the end, this arc 1078 is declared to be the best single arc mitigation plan. An additional 1,494 IPs needed to be solved due to the second-stage performance measures allocated to the three surviving arcs. A total of 1,480 IPs were solved in allocating the first-stage performance measures (10 performance measures for each of the 74 single arc mitigation plans), and 1,000 IPs were solved to determine the MAST. Therefore, a final total of 3,974 IPs needed to be solved to implement GSP, and we find one best single arc mitigation plan with 95% confidence. The results from testing in each group are given in Table 5.1.2. Note that the totals in the last row indicate the overall number of different arcs that survive in any round, not the sum of the arcs surviving in each round, and the total number of IPs required as part of **Sample 2** in the algorithm. We expect in all cases that the total number of IPs required for **Sample 2** will be no larger than the number of IPs in Sample 2 from the row in the 1-GSP results corresponding to the split of α to α_0 and α_1 selected. We also expect in all cases that the number of total arcs surviving in the implementation of GSP will be no larger than the number of arcs that survive in the corresponding results for 1-GSP. In this case, the results in the last row of Table 5.1.2 exactly match the results indicated by the corresponding row in Table 5.1.1 of the 1-GSP results.

Table 3 10% GSP Results

Round of Testing, l	No. of Survivors	IPs in **Sample 2**
1	3	1494
2	1	0
3	1	0
4	1	0
Total	3	1494

5.1.3 Brute Force Results

For the brute force procedure, we use the total number of IPs required for GSP to determine how many performance measures to compute for each mitigation plan. GSP required a total of 3,974 IPs to be solved, which is approximately equal to 27 performance measures per mitigation plan (3,996 IPs). With 27 performance measures per mitigation plan, Tukey's HSD Test is unable to distinguish between the top four mitigation plans with a confidence level of 95%. With a confidence level between 90% and 60%, Tukey's HSD test is unable to distinguish between the top two mitigation plans. With a confidence level lower than 60%, Tukey's HSD Test is unable to converge. Since we cannot determine the best single arc mitigation plan at the 10% damage level using Tukey's HSD Test, we conclude that the brute force procedure failed.

5.1.4 Conclusions

In comparing our results from GSP and the brute force procedure, we find that we were unable to use the brute force method with the same number of IPs to achieve the same solution we found in GSP. Using GSP, we are able to determine the best single arc mitigation plan at the 10% damage level with 95% confidence by solving 3,974 IPs. With roughly the same number of IPs solved, we were unable to find the best single arc mitigation plan using the brute force procedure. Specifically, Tukey's HSD test could not distinguish between the top four mitigation plans until the confidence level was lowered. Therefore, we find that GSP significantly outperformed the brute force procedure in this case.

5.2 25% Damage Results

We next consider the best single arc mitigation plan when there is a moderate damage level of 25% damage to the network using the same procedure. With an event that causes 25% damage to the network, the average loss in met demand immediately following the event is 43%. Therefore, 25% damage to the network corresponds to a medium damage level. At this damage level, 18 transmission arcs are damaged and 194 distribution arcs are damaged, so roughly 28% of these damaged arcs can be repaired in the time horizon. We solved the INDS problem for 1,000 randomly generated 25% damage scenarios to determine each arc's MAST and created the groups given in Table 5.2.

Table 4 25% MAST

Group	Number of Arcs	MAST Range
1	21	0.03 – 5.18
2	5	10.74 – 20.9
3	4	26.66 – 28.09
4	14	29 – 32.46
5	13	32.64 – 34.4
6	14	34.85 – 37.11
7	3	39

5.2.1 1-GSP Results: Split of α

The split of $\alpha = 0.05$ to α_0 and α_1 is considered using 1-GSP. Unlike the 10% damage results when t and h were changed, changes in t alter the number of arcs that survive testing. The optimal allocation is 0.03 to α_0 and 0.02 to α_1. At this split, more arcs are eliminated in **Testing**, so even though individual numbers of second-stage performance measures are larger, an overall savings in the number of IPs required is achieved because the number of arcs left is smaller. Increasing the split to α_0 beyond this point, however, did not result in any more savings. Unlike the 10% damage results, the best split is not to make α_1 large to reduce h as much as possible, but instead to tighten the testing procedure. Our results are summarized in Table 5.2.1.

Table 5 25% 1-GSP Results

(α_0, α_1)	(t, h)	No. of Survivors	IPs in **Sample 2**
$(0.01, 0.04)$	$(5.753, 5.99)$	5	2140
$(0.02, 0.03)$	$(5.214, 6.228)$	5	2322
$(0.025, 0.25)$	$(5.047, 6.3809)$	5	2442
$(0.03, 0.02)$	$(4.911, 6.5707)$	3	1222
$(0.04, 0.01)$	$(4.701, 7.1904)$	3	1474

5.2.2 GSP Results

GSP is implemented using the groups from MAST, and the t and h statistics found in 1-GSP. In Group 1, three arcs survived. These arcs were given their second-stage performance measures and added with Group 2 for the next round of **Testing**. In the second round of **Testing**, only one arc (a surviving arc from Group 1) survived. In the remaining rounds, this arc continued to be the only survivor. We can declare this arc, arc 1078, to be the best single arc mitigation plan at the 25% damage level. We had to solve an additional 1,222 IPs in allocating second-stage performance measures in GSP. Including the IPs required to obtain first-stage performance measures and the 1,000 IPs required to determine MAST, we solved a total of 3,702 IPs in our implementation of GSP. We found one best single arc mitigation plan with 95% confidence. The results from testing for each group are presented in the Table 5.2.2. Note that our results from GSP in the last row of Table 5.2.2 are the same as the results indicated by the corresponding row in Table 5.2.1 of our 1-GSP results.

Table 6 25% GSP Results

Round of Testing, l	No. of Survivors	IPs in **Sample 2**
1	3	1222
2	1	0
3	1	0
4	1	0
5	1	0
6	1	0
7	1	0
Total	3	1222

5.2.3 Brute Force Results

For the brute force procedure, we use the total number of IPs required for GSP to determine how many performance measures to take of each mitigation plan. GSP required a total of 3,702 IPs to be solved, which is roughly equal to 26 performance measures per mitigation plan (3,848 IPs). With 26 performance measures per mitigation plan, Tukey's HSD Test is unable to distinguish between the top three mitigation plans with a confidence level between 95% to 60%. With a confidence level lower than 60%, Tukey's HSD Test fails to converge. Since we cannot determine the best single arc mitigation plan at the 25% damage level with Tukey's HSD Test, we conclude that the brute force procedure failed.

5.2.4 Conclusions

Comparing our results from GSP and the brute force approach, we find that the brute force procedure was unable to determine the best single arc mitigation plan using the same number of IPs required for GSP. With GSP, we solved 3,702 IPs to determine the best single arc mitigation plan at the 25% damage level with 95% confidence. Using the same number of IPs to implement our brute force procedure, we are unable to replicate this result. With all confidence levels where Tukey's HSD converged, the brute force procedure could not do any better than to identify the top three mitigation plans. Therefore, we conclude that GSP significantly outperformed the brute force procedure in this case.

5.3 40% Damage Results

The final damage level considered is a high damage level of 40%, as this level would be substantial damage to the network. An event that causes 40% damage results in an average loss of 81% of the met demand in the network immediately following the event. At this damage level, 29 transmission arcs are damaged, and 310 distribution arcs are damaged. Just under 18% of these damaged arcs can be repaired in the time horizon. We solved 1,000 INDS problems for randomly generated damage scenarios to find each arc's MAST and determined the following groups given in Table 5.3.

Table 7 40% MAST

Group	Number of Arcs	MAST Range
1	16	$0.358-7.335$
2	5	$8.769-11.218$
3	14	$21.194-28.647$
4	36	$29.47-37.473$
5	3	39

5.3.1 1-GSP Results: Split of α

The split of $\alpha = 0.05$ to α_0 and α_1 is then considered. The results obtained with these changes are the same as the results for 25% damage level testing. Allocating 0.03 to α_0 and 0.02 to α_1 is the best split. With this split, the fewest total IPs are required and only five arcs survive testing. As with the 25% level results, a tighter screening process allows for more arcs to be eliminated at the cost of slightly higher individual numbers of second-stage performance measures. However, allocating more than 0.03 to α_0 did not allow us to eliminate any more arcs so it was not beneficial to tighten testing beyond this point. The results we obtained for the various splits we examined are given in Table 5.3.1.

Table 8 40% 1-GSP Results

(α_0,α_1)	(t,h)	No. of Survivors	IPs in **Sample 2**
$(0.01,0.04)$	$(5.753,5.99)$	6	7458
$(0.02,0.03)$	$(5.214,6.228)$	6	8068
$(0.025,0.025)$	$(5.047,6.3809)$	6	8476
$(0.03,0.02)$	$(4.911,6.5707)$	2	328
$(0.04,0.01)$	$(4.701,7.1904)$	2	398

5.3.2 GSP Results

GSP is implemented using the groups formed from MAST and the t and h statistics found in 1-GSP. In Group 1, two arcs survived. These two arcs were given their second-stage performance measures and added for **Testing** with Group 2. In round two of **Testing**, only one arc, a previous surviving arc from Group 1, survived. In round three, this same arc survived again along with a new arc from Group 3. The new arc was given its second-stage performance measures, and both were added with Group 4 for the next round of **Testing**. In the fourth round of **Testing**, two arcs survived: the survivor from Group 1 and a new arc from Group 4. The new arc had its second-stage performance measures computed, and both arcs were added with Group 5 for another round of **Testing**. In round five, only one arc, the previous surviving arc from Group 1, survived. Thus, this arc, arc 1078, can be declared the best single arc mitigation plan at the 40% damage level. From second-stage performance measures, GSP required an additional 4,910 IPs to be solved. In addition to the IPs required for first-stage performance measures and the 1,000 used to determine MAST, we had to solve a total of 7,390 IPs. We found the best single arc mitigation plan with 95% confidence. The results from each group's testing are presented in Table 5.3.2. Note that surviving arcs in later rounds required a substantial number of IPs to be solved for their second-stage performance measures. This result is due to their large standard deviations, which is problematic in eliminating arcs in **Testing** and in keeping the computational effort low through the number of performance measures needed. We also note that our results for GSP in the last row of Table 5.3.2 are worse than the results indicated by the corresponding row in Table 5.3.1 of our 1-GSP results. This result is due to the fact that additional performance measures for the eventual winning arc, arc 1078, actually lowered its original average. As a result, arcs that did not survive **Testing** in 1-GSP were able to survive **Testing** in GSP.

Table 9 40% GSP Results

Round of Testing, l	No. of Survivors	IPs in **Sample 2**
1	2	328
2	1	0
3	2	1670
4	2	2912
5	1	0
Total	4	4910

5.3.3 Brute Force Results

For the brute force procedure, we use the total number of IPs required for GSP to determine how many performance measures to compute for each mitigation plan. GSP required a total of 7,390 IPs to be solved, which is approximately equal to 50 performance measures per mitigation plan (7,400 IPs). With 50 performance measures per mitigation plan, Tukey's HSD Test is unable to distinguish between the top nine mitigation plans with a confidence level between 95% and 85%. With a confidence level between 80% and 75%, Tukey's HSD Test is unable to distinguish between the top eight mitigation plans.With 70% confidence, Tukey's HSD Test is unable to distinguish between the top seven mitigation plans. Tukey's HSD Test is unable to distinguish between the top six mitigation plans with a confidence level between 65% and 60%. With a confidence level lower than 60%, Tukey's HSD Test is unable to converge. Since we are unable to use Tukey's HSD Test to determine the best single arc mitigation plan at the 40% damage level, we conclude that the brute force procedure failed.

5.3.4 Conclusions

Comparing our results for GSP with our results for the brute force procedure, we find that we were unable to use the brute force procedure to obtain the same results we obtained for GSP. With GSP, we are able to find the best single arc mitigation plan at the 40% damage level with 95% confidence by solving 7,390 IPs. By solving approximately the same number of IPs in our brute force procedure, we are unable to find the best single arc mitigation plan. Tukey's HSD test could not distinguish between many of the mitigation plans until the confidence level was dropped significantly, and it still could not distinguish between the top six mitigation plans.Therefore, we conclude that GSP significantly outperformed the brute force procedure in this case.

5.4 Overall Conclusions

There are two primary conclusions to draw from these results. First, the brute force procedure failed in all three cases. In the 10% damage case, it could not distinguish between the top four mitigation plans until the confidence level was dropped. It could not distinguish between the top three mitigation plans in the 25% damage case. In the 40% damage case, there were nine top mitigation plans until the confidence level was dropped. It is clear that the brute force procedure could not determine the top mitigation plan with the same level of confidence as GSP, and GSP significantly outperformed the standard statistical method overall.

The second primary conclusion to draw from these results is that the same arc, arc 1078, was selected as the winner in all three cases. From an application perspective, this result would be reassuring to a network manager. As we have previously described, this method is for long-term, high-cost mitigation plans. For each case individually, the winning single arc mitigation plan has a statistical guarantee that it will contribute the most to the restoration for that damage level. While this statistical guarantee for one case can help reassure a manager, this confirmation of the same winning single arc mitigation plan in all three cases would make an even stronger argument to select this arc as a mitigation plan. Because this mitigation plan can contribute the most to the restoration of the network in low, medium, and high damage levels, it will significantly improve the resilience of the network for all future events. Therefore, using this method for multiple damage levels would be a good way for a network manager to select a long-term mitigation plan by seeing if the same mitigation plan is the winner in all cases considered.

6 Sequential Mitigation Plans

Because of the types of long-term mitigation plans on which this research focuses, it is unlikely that a manager will have a sufficient budget to make two long-term mitigation decisions at the same time. However, these managers may be able to determine that in a later time in the future, they will again have a sufficient budget to make a long-term mitigation decision. Therefore, this section focuses on how a manager could make a sequential mitigation decision. To return to the results presented previously, it was determined that arc 1078 was the best single arc mitigation plan at all damage levels. The next relevant question that arises is what is the next best long-term mitigation plan, given that arc 1078 is already protected? Determining a sequential mitigation decision can help a manager to more efficiently plan for the future. The procedure outlined in this work can be modified to determine the next sequential mitigation decision. In all damage scenarios, the arc that has been selected as the first mitigation decision would not be allowed to be one of the damaged arcs and would always be available. With the results presented, this first mitigation decision corresponds to protecting arc 1078. In this section, the results for the next best arc to protect, given that arc 1078 is protected, is presented. We follow the same framework used in the original single arc testing. We use the same MAST groups, perform 1-GSP to find a good split of α to α_0 and α_1, and implement GSP with the same parameters as before to determine the best arc to protect with arc 1078. We also use the same brute force procedure described previously, where we used the total number of IPs required for GSP at that damage level to determine how many performance measures of each mitigation plan to take, and use Tukey's HSD Test to determine a single best mitigation plan. The results for all of the same damage levels considered previously are given in the next subsections.

6.1 10% Damage Results with Arc 1078 Protected

6.1.1 1-GSP Results: Split of α

For the low level damage of 10%, we first consider changes in the split of $\alpha = 0.05$ to α_0 and α_1. Since there are 73 systems now, the t and h statistics will both be smaller. Unlike the previous results at the 10% damage level, it is not best to simply make α_1 as large as possible to make h as small as possible. Instead, the ideal split is 0.03 to α_0 and 0.02 to α_1. With this split, the tighter testing eliminates all but one arc. Even though this arc individually has a higher number of second-stage performance measures, there are no other arcs that survive. Clearly, allocating more than 0.03 to α_0 would not allow any more arcs to be eliminated, so such a split would only mean the remaining arc has a higher number of second-stage performance measures. The results are displayed in the Table 6.1.1.

Table 10 10% 1078 in 1-GSP Results

(α_0,α_1)	(t,h)	No. of Survivors	IPs in **Sample 2**
$(0.01,0.04)$	$(5.742,5.9797)$	3	608
$(0.02,0.03)$	$(5.204,6.2175)$	3	662
$(0.025,0.025)$	$(5.036,6.3702)$	3	700
$(0.03,0.02)$	$(4.901,6.5598)$	1	96
$(0.04,0.01)$	$(4.692,7.1783)$	4	120

6.1.2 GSP Results

As previously stated, the same MAST groups from before are used but with arc 1078 removed. The t and h statistics found in 1-GSP are used to implement GSP with the same MAST groups. In Group 1, one arc survived. This arc continued to be the only survivor in remaining rounds of **Testing**. Thus, we can declare this arc, arc 1076, to be the best single arc mitigation plan at the 10% damage level with arc 1078 protected. We were required to solve an additional 96 IPs for second-stage performance measures. Adding to this number the 1,000 IPs needed to determine MAST and 1,460 IPs needed to allocate our initial 10 performance measures for all 73 arc mitigation plans, we needed to solve a total of 2,556 IPs. We found the best arc to protect with arc 1078 with a confidence level of 95%. Our results for testing in each group are presented in Table 6.1.2. Note that the last row of our results for GSP in Table 6.1.2 are the same as the results indicated by the corresponding row of Table 6.1.1 of our 1-GSP results.

Table 11 10% 1078 in GSP Results

Round of Testing, l	No. of Survivors	IPs in **Sample 2**
1	1	96
2	1	0
3	1	0
4	1	0
Total	1	96

6.1.3 Brute Force Results

For the brute force procedure, we use the total number of IPs required for GSP to determine how many performance measures to take of each mitigation plan. GSP required a total of 2,556 IPs to be solved, which is approximately equal to 18 performance measures per mitigation plan (2,628 IPs). With 95% confidence, Tukey's HSD Test is able to determine that arc 1076 is the best single arc mitigation plan with arc 1078 protected. This result agrees with the result from GSP. With a confidence level higher than 95%, Tukey's HSD Test is unable to distinguish between the top two mitigation plans. Therefore, we conclude that the brute force procedure is able to determine the best single arc mitigation plan with 95% confidence.

6.1.4 Conclusions

In comparing our results for GSP with our results for the brute force procedure, both GSP and the brute force procedure were able to determine the best single arc to protect with arc 1078 with 95% confidence. Tukey's HSD could not distinguish between the top two mitigation plans with higher confidence level. Therefore, we conclude that GSP and the brute force procedure performed the same.

6.2 25% Damage Results with Arc 1078 Protected

6.2.1 1-GSP Results: Allocation of α

For the medium level damage of 25%, we first consider the changes in the allocation of $\alpha = 0.05$ to α_0 and α_1. In line with previous results, the best split is to allocate 0.03 to α_0 and 0.02 to α_1. The tighter t statistic from this α_0 significantly reduces the number of surviving arcs from 14 to seven. With only seven arcs receiving second-stage sampling, the number of IPs that must be solved is significantly fewer than the number required with other splits. Allocating more than 0.03 to α_0 does not allow for any more arcs to be eliminated, so the four surviving arcs would have higher numbers of second-stage performance measures. Therefore, the best split with respect to number of IPs required to solve is 0.03 to α_0 and 0.02 to α_1. The results are presented in Table 6.2.1.

Table 12 25% 1078 in 1-GSP Results

(α_0, α_1)	(t,h)	No. of Survivors	IPs in **Sample 2**
(0.01, 0.04)	(5.742, 5.9797)	14	8082
(0.02, 0.03)	(5.204, 6.2175)	14	8760
(0.025, 0.025)	(5.036, 6.3702)	14	9206
(0.03, 0.02)	(4.901, 6.5598)	7	6482
(0.04, 0.01)	(4.692, 7.1783)	7	7788

6.2.2 GSP Results

The same MAST groups formed for regular 25% damage testing, and the t and h statistics found in 1-GSP are used to implement GSP. In Group 1, four arcs survived. These arcs received their additional second-stage performance measures, and were added with Group 2 for the next round of **Testing**. One of these two arcs survived round two, along with a new arc from Group 2. This new arc was given its second-stage performance measures, and both arcs were added with Group 3 for another round of **Testing**. In subsequent rounds of **Testing**, only the arc from Group 1 continued to survive. Therefore, we can declare this arc, arc 1076, to be the best single arc mitigation plan at the 25% damage level with arc 1078 protected. We needed to solve in total 2,532 more IPs as part of **Sample 2**. Including the 1,000 IPs required to determine MAST and the IPs required for the first-stage performance measures, we needed to solve a total of 4,992 IPs to implement GSP. We found the best arc to protect with arc 1078 at the 25% damage level with 95% confidence. Note the results for GSP given in the last row of Table 6.2.2 are better than the results indicated by the corresponding row of Table 6.2.1, the 1-GSP results. Also note that the surviving arc from Group 2 had a high standard deviation, as the number of IPs for **Sample 2** required for this single arc are higher than the number required for all four surviving arcs from Group 1. The results for testing of each group are presented in Table 6.2.2.

Table 13 25% 1078 in GSP Results

Round of Testing, l	No. of Survivors	IPs in **Sample 2**
1	4	1218
2	2	1314
3	1	0
4	1	0
5	1	0
6	1	0
7	1	0
Total	5	2532

6.2.3 Brute Force Results

For the brute force procedure, we use the total number of IPs required for GSP to determine how many performance measures to take of each mitigation plan. GSP required a total of 4,992 IPs to be solved, which is approximately equal to 35 performance measures per mitigation plan (5,110 IPs). With 35 performance measures per mitigation plan, Tukey's HSD Test is unable to distinguish between the first 13 mitigation plans with a confidence level between 95%. With a confidence level of 90% Tukey's HSD Test is unable to distinguish between the top nine mitigation plans. Tukey's HSD Test cannot distinguish between the top eight mitigation plans with a confidence level between 85% and 60%. Tukey's HSD Test fails to converge with a lower confidence level than 60%. Since we cannot use Tukey's HSD Test to determine the best single arc mitigation plan at 25% damage with arc 1078 protected, we conclude that the brute force failed.

6.2.4 Conclusions

In comparing the results from GSP with our results from the brute force approach, we find that we were unable to use the brute force procedure with the same number of IPs to obtain the same results from GSP. The brute force procedure can only get as close as determining the top eight mitigation plans, even as the confidence level falls to 60%. In this case, we conclude that GSP significantly outperformed the brute force method.

6.3 40% Damage Results with Arc 1078 Protected

6.3.1 1-GSP Results: Split of α

Changes in the split of $\alpha = 0.05$ are considered first. As indicated by previous results, the best split of α is 0.03 to α_0 and 0.02 to α_1. By using this split, only six arcs survive, as opposed to 13 surviving arcs using some of the other splits. The number of IPs required to solve is significantly lower with this split as a result. Allocating more than 0.3 to α_0, however, did not reduce the number of surviving arcs below seven and would therefore only increase the second-stage performance measures and the number of IPs that must be solved. The results for the various splits we examined are given in Table 6.3.1.

Table 14 40% 1078 in 1-GSP Results

(α_0, α_1)	(t, h)	No. of Survivors	IPs in **Sample 2**
$(0.01, 0.04)$	$(5.742, 5.9797)$	13	12390
$(0.02, 0.03)$	$(5.204, 6.2175)$	13	13414
$(0.025, 0.025)$	$(5.036, 6.3702)$	13	14096
$(0.03, 0.02)$	$(4.901, 6.5598)$	6	6944
$(0.04, 0.01)$	$(4.692, 7.1783)$	6	8336

6.3.2 GSP Results

To implement GSP, the same groups found in MAST and the t and h statistics found 1-GSP are used. In Group 1, three arcs survived. These arcs were given their second-stage performance measures and added with Group 2 for the next round of **Testing**. In the second round, only a previous surviving arc from Group 1 survived. This arc also survived in round three, along with two other arcs from Group 3. These new arcs were given their second-stage performance measures and all three arcs were added with Group 4 for round four of **Testing**. In the fourth round of **Testing**, this previous surviving arc from Group 1 survived with a new arc from Group 4. This new arc was given its second-stage performance measures and both arcs were added to Group 5 for the final round of **Testing**. In the final round of **Testing**, the only surviving arc was the original surviving arc from Group 1. Thus at the end we can declare this arc, arc 1076, as the best single arc mitigation plan at the 40% damage level with arc 1078 protected. As a result of the performance measures required in **Sample 2** for surviving arcs, we had to solve an additional 6,944 IPs. This number is substantially larger than the number of additional IPs required at any other damage level. This result is due to large standard deviations, similar with the results for regular 40% damage testing. Adding to this number the 1,000 IPs required to determine MAST and the IPs required for first-stage performance measures, implementing GSP required solving 9,404 IPs. We are able to determine the best arc to protect with arc 1078 at the 40% damage level with 95% confidence. Note that our results for GSP given in the last row of Table 6.3.2 match the results indicated by the corresponding row of Table 6.3.1 of our 1-GSP results. The results of testing each group are given in Table 6.3.2, where it is clear that arcs surviving in later groups had substantially higher numbers of second-stage performance measures.

Table 15 40% 1078 in GSP Results

Round of Testing, l	No. of Survivors	IPs in **Sample 2**
1	3	548
2	1	0
3	3	4606
4	2	1790
5	1	0
Total	6	6944

6.3.3 Brute Force Results

For the brute force procedure, we use the total number of IPs required for GSP to determine how many samples to take of each mitigation plan. GSP required a total of 9,404 IPs to be solved, which is approximately equal to 65 performance measures per mitigation plan (9,490 IPs). With 65 performance measures per mitigation plan, Tukey's HSD Test cannot distinguish between the top 12 mitigation plans with a confidence level of 95%. With a confidence level between 90% and 80%, Tukey's HSD Test cannot distinguish between the top 11 mitigation plans. For a confidence level between 75% and 60%, Tukey's HSD cannot distinguish between the top 10 mitigation plans. Tukey's HSD Test fails to converge with a lower confidence level. Since Tukey's HSD Test cannot find the best single arc mitigation plan at 40% damage with arc 1078 protected, we conclude that the brute force procedure failed.

6.3.4 Conclusions

Comparing our results for GSP and the brute force procedure, we find that GSP can determine the best single arc mitigation plan at 40% damage with arc 1078 protected with 95% confidence while the brute force procedure is unable to determine the best single arc mitigation plan when using the same number of IPs. The brute force procedure can only do as well as to identify the top 10 mitigation plans, with a low confidence level between 75% and 60%. In this case, we conclude that GSP significantly outperformed the brute force procedure.

6.4 Overall Conclusions

In the results for the sequential mitigation plans, there are a few primary conclusions to draw. While the brute force procedure was able to determine the best single arc mitigation plan when arc 1078 is protected with 95% confidence in one case, it performed very poorly in the other two cases. In the 10% damage case, it was able to determine the best single arc mitigation plan, and confirmed the results from GSP. However, it could not distinguish between the top 13 mitigation plans for the 25% damage case until the confidence level was lowered. In the 40% damage case, it could not distinguish between the top 12 mitigation plans until the confidence level was lowered. Therefore, we conclude in general that GSP significantly outperformed the method using standard statistical testing. As with the initial results, the same single arc mitigation plan, arc 1076, is selected as the winner in all three damage cases. This result is useful from an application perspective for a manager seeking to make a sequential long-term mitigation decision. There is a strong argument to select this arc to use a mitigation plan, given that arc 1078 was selected as the first mitigation decision. With arc 1078 already protected, arc 1076 as a mitigation plan will contribute to the resilience of the network regardless of the damage level of future events. Therefore, using this method and considering multiple damage levels would be a good way for a network manager to make a sequential long-term mitigation decision if the same mitigation plan wins at all levels.

7 Conclusions and Future Work

We have developed a framework for finding the best single arc mitigation plan in a network. We have shown that R&S procedures like GSP can outperform standard statistical testing approaches in determining the best single arc mitigation plan. In all but one case, the brute force procedure was unable to determine the best single arc mitigation plan as it could not distinguish between some of the top mitigation plans. In some cases, it could not distinguish between more than 10 of the top mitigation plans, until the confidence level was lowered substantially. Therefore, we can conclude that in general, GSP significantly outperformed the brute force procedure. The table of results is given in Table 7. Note that the results for the brute force procedure indicate the smallest number of top mitigation plans between which it was unable to distinguish before being unable to converge.

Table 16 Summary Results

Case	GSP		Brute Force		
	Total IPs	Confidence	Total IPs	Confidence	No. of Top Plans
10% Damage	3,974	95%	3,996	< 60%	2
25% Damage	3,702	95%	3,848	< 60%	3
40% Damage	7,390	95%	7,400	< 60%	6
10% Damage, 1078 in	2,556	95%	2,628	95%	1
25% Damage, 1078 in	4,992	95%	5,110	< 60%	3
40% Damage, 1078 in	9,404	95%	9,490	< 60%	10

A primary observation to make from this table is that higher levels of damage generally require more IPs to be solved. The reason behind this increase in the number of IPs required is related to the large standard deviations that pose a problem to either approach. However, it appears that this increased variation in the data is handled better by the R&S procedure GSP rather than through standard statistical testing. In this general procedure we developed, we have some clear advantages over a traditional stochastic programming approach. We can consider a complicated second-stage problem without being concerned that the problem size will blow up with the number of damage scenarios we consider. We have integrated mitigation decisions with restoration to form a more complete picture of how a mitigation plan will contribute to the resilience of a network. We have established an application of R&S procedures to the problem of finding the best single arc mitigation plan with a given level of statistical certainty, and shown it to significantly outperform approaches using standard statistical tests in all but one case.

For future research, we would like to work on scaling this method for general arc mitigation plans for any network. Scaling this method will require constructing a method to eliminate some potential mitigation plans from the beginning. It will also require developing ways to reduce the high standard deviation that was already becoming a problem in single arc mitigation plans. We wish to construct a more generalized version of our method for determining the best mitigation plan that will work for any arc mitigation plans on any network.

References

1. Ahuja, R., Magnanti, T., Orlin, J.: Network flows: Theory, algorithms, and applications. Prentice-Hall, Englewood Cliffs, New Jersey (1993)
2. Alderson, D., Brown, G., Carlyle, W., Wood, R.: Solving defender-attacker-defender models for infrastructure defense. In: R. Wood, R. Dell (eds.) Operations Research, Computing, and Homeland Security, pp. 28–49 (2011)
3. Alderson, D.L., Brown, G.G., Carlyle, W.M.: Assessing and Improving Operational Resilience of Critical Infrastructures and Other Systems, chap. 9, pp. 180–215. INFORMS (2014). DOI 10.1287/educ.2014.0131. URL `http://pubsonline.informs.org/doi/abs/10.1287/educ.2014.0131`
4. Barbarosoglu, G., Arda, Y.: A two-stage stochastic programming framework for transportation planning in disaster response. Journal of the Operational Research Society **55**, 43–53 (2004)
5. Birge, J., Louveaux, F.: Introduction to Stochastic Programming. Springer-Verlag, New York, New York (1997)
6. Cavdaroglu, B.: Integrated mitigation, restoration, and scheduling problem for interdependent infrastructure networks. Ph.D. thesis, Rensselaer Polytechnic Institute (2012)
7. Cavdaroglu, B., Hammel, E., Mitchell, J., Sharkey, T., Wallace, W.: Integrating restoration and scheduling decisions for disrupted interdependent infrastructure systems. Annals of Operations Research **203**(1), 279–294 (2013)
8. Chang, M.S., Tseng, Y.L., Chen, J.W.: A scenario planning approach for the flood emergency logistics preparation problem under uncertainty. Transportation Research Part E: Logistics and Transportation Review **43**(6), 737 – 754 (2007). DOI http://dx.doi.org/10.1016/j.tre.2006.10.013. URL `http://www.sciencedirect.com/science/article/pii/S1366554507000178`. Challenges of Emergency Logistics Management
9. Church, R.L., Scaparra, M.P., Middleton, R.S.: Identifying critical infrastructure: The median covering facility interdiction problems. Annals of the Association of American Geographers **94**(3), 491–502 (2004)
10. Cimellaro, G.P., Reinhorn, A.M., Bruneau, M.: Framework for analytical quantification of disaster resilience. Engineering Structures **32**(11), 3639 – 3649 (2010). DOI http://dx.doi.org/10.1016/j.engstruct.2010.08.008. URL `http://www.sciencedirect.com/science/article/pii/S014102961000297X`
11. Cormican, K., Morton, D., Wood, R.: Stochastic network interdiction. Operations Research **46**, 184–197 (1998)
12. Current, J., Daskin, M.S., Schilling, D.: Discrete Network Location Models, chap. 3, pp. 81–118. Springer (2004)
13. Daskin, M.: Network and Discrete Location: Models, Algorithms, and Applications. John Wiley & Sons (1995)
14. U.S. Department of Energy, Office of Electricty Delivery and Energy Reliability: Hurricane sandy situation report 20 (2012). URL `http://www.oe.netl.doe.gov/docs/2012_SitRep20_Sandy_11072012_1000AM.pdf`
15. U.S. Energy Information Administration: Power outages often spur questions around burying power lines (2012). URL `http://www.eia.gov/todayinenergy/detail.cfm?id=7250#`
16. Frazier, P.I.: A fully sequential elimination procedure for indifference-zone ranking and selection with tight bounds on probability of correct selection (2013). Submitted to *Operations Research*
17. Hentenryck, P.V., Bent, R., Coffrin, C.: Strategic planning for disaster recovery with stochastic last mile distribution. In: Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems, pp. 318–333 (2010)
18. Kim, S.H., Nelson, B.: A fully sequential elimination procedure for indifference-zone ranking and selection with tight bounds on probability of correct selection. ACM Transactions on Modeling and Computer Simulation (TOMACS) **11**(3), 251–273 (2001)
19. Kim, S.H., Nelson, B.: Selecting the best system. Handbooks in Operations Research and Management Science **13**, 501–534 (2006)
20. Kleywegt, A., Shapiro, A., de mello, T.H.: The sample average approximation method for stochastic discrete optimization. SIAM Journal on Optimization **12**, 479–502 (2002)
21. Lee, E., Mitchell, J., Wallace, W.: Restoration of services in interdependent infrastructure systems: A network flows approach. IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews **37**(6), 1303–1317 (2007)
22. Losada, C., Scaparra, M.P., Church, R.L., Daskin, M.S.: The stochastic interdiction median problem with disruption intensity levels. Annals of Operations Research **201**, 345–365 (2012)
23. Matisziw, T., Murray, A., Grubesic, T.: Strategic network restoration. Networks and Spatial Economics **10**, 345–361 (2010)
24. Mete, H., Zabinsky, Z.: Stochastic optimization of medical supply location and distribution in disaster management. International Journal of Production Economics **126**, 76–84 (2010)
25. Nelson, B., Swann, J., Goldsman, D., Song, W.: Simple procedures for selecting the best simulated system when the number of alternatives is large. Operations Research **49**(6), 950–963 (2001)
26. Nurre, S., Cavdaroglu, B., Mitchell, J., Sharkey, T., Wallace, W.: Restoring infrastructure systems: An integrated network design and scheduling problem. European Journal of Operational Research **223**(3), 794–806 (2012)
27. Nurre, S., Sharkey, T.: Integrated network design and scheduling problems with parallel identical machines: Complexity analysis and dispatching rules. Networks **63**(4), 306–326 (2014)
28. Nurre, S., Sharkey, T.: Online integrated network design and scheduling problems with flexible release dates (2015). Rensselaer Polytechnic Institute

29. Owen, S., Daskin, M.: Strategic facility location: A review. European Journal of Operational Research **111**, 423–447 (1998)
30. Pichitlamken, J., Nelson, B., Hong, L.: A sequential procedure for neighborhood selection-of-the-best in optimization via simulation. European Journal of Operational Research **173**(1), 283–298 (2006)
31. Rawls, C., Turnquist, M.: Pre-positioning of emergency supplies for disaster response. Transportation Research Part B: Methodological **44**, 521–534 (2010)
32. Salmeron, J., Apte, A.: Stochastic optimization for natural disaster asset prepositioning. Production and Operations Management **19**, 43–53 (2010)
33. Scaparra, M.P., Church, R.L.: A bilevel mixed-integer program for critical infrastructure protection planning. Computers & Operations Research **35**(6), 1905–1923 (2008)
34. Shen, S.: Two-stage models and algorithms for optimizing infrastructure design and recovery operations under stochastic disruptions. Computers & Operations Research **40**(11), 2677–2688 (2013)
35. Smith, J.C., Lim, C., Sudargho, F.: Survivable network design under optimal and heuristic interdiction scenarios. Journal of Global Optimization **38**(2), 181–1999 (2006)
36. Snyder, L., Scaparra, M., Daskin, M., Church, R.: Planning for disruptions in supply chain networks. In: M.P. Johnson, B. Norman, N. Secomandi (eds.) TutORials 2006, INFORMS Tutorials in O.R. Series, chap. 9. INFORMS (2006). URL `http://www.lehigh.edu/~lvs2/Papers/SSDC_TutORial_final.pdforhttp://www.lehigh.edu/~lvs2/`
37. The White House, Office of the Press Secretary: Presidential policy directive–critical infrastructure security and resilience (2013). URL `http://www.whitehouse.gov/the-press-office/2013/02/12/presidential-policy-directive-critical-infrastructure-security-and-resil`
38. Wood, R.: Deterministic network interdiction. Mathematical and Computer Modelling **17**, 1–18 (1993)