Robust Optimization of Contaminant Sensor Placement for Community Water Systems

Robert D. Carr* Harvey J. Greenberg† William E. Hart‡
Goran Konjevod§ Erik Lauer¶ Henry Lin‖ Tod Morrison**
Cynthia A. Phillips††

September 5, 2004

Keywords: robust optimization, mixed-integer linear programming, combinatorial optimization, sensor placement, water security

*Algorithms and Discrete Mathematics Department, Sandia National Laboratories, rdcarr@sandia.gov.

†Mathematics Department, University of Colorado at Denver, Harvey.Greenberg@cudenver.edu

‡Algorithms and Discrete Mathematics Department, Sandia National Laboratories, wehart@sandia.gov.

§Computer Science Department, Arizona State University, goran@asu.edu

¶Algorithms and Discrete Mathematics Department, Sandia National Laboratories, elauer@sandia.gov

‖University of California at Berkeley, henrylin@eecs.berkeley.edu

**Mathematics Department, University of Colorado at Denver, tmorriso@math.cudenver.edu

††Algorithms and Discrete Mathematics Department, Sandia National Laboratories, caphill@sandia.gov.

Abstract

We present a series of related robust optimization models for placing sensors in municipal water networks to detect contaminants that are maliciously or accidentally injected. We formulate sensor placement problems as mixed-integer programs, for which the objective coefficients are not known with certainty. We consider a restricted absolute robustness criteria that is motivated by natural restrictions on the uncertain data, and we define three robust optimization models that differ in how the coefficients in the objective vary. Under one set of assumptions there exists a sensor placement that is optimal for all admissible realizations of the coefficients. Under other assumptions, we can apply sorting to solve each worst-case realization efficiently, or we can apply duality to integrate the worst-case outcome and have one integer program. The most difficult case is where the objective parameters are bilinear, and we prove its complexity is NP-hard, even under very simplifying assumptions. We consider a relaxation that provides a tight approximation, giving an overall guarantee of near optimality when used with branch-and-bound search.

1 Introduction

Combinatorial optimization techniques need to address modeling and data uncertainties in many applications. As early as the 1950s, Dantzig [5] introduced stochastic programming to deal with *aleatory uncertainty*, which describes the inherent variation associated with the system being modeled [7, 14, 15].

More recently, researchers have developed robust optimization methods [9] to deal with *epistemic uncertainty*, which describes our lack of knowledge about information in our model [7, 14, 15]. For example, a common assumption is that the coefficients in the objective function of the problem are uncertain in the sense that they each can assume any value within a finite interval. Interval data occurs often in practice [6, 9], and when quantitative parameters have a subjective nature, interval values can be used interactively to provide more intuition about the model.

Robust optimization methods generally seek a solution that minimizes some measure of worst performance with respect to the uncertainty in the data. Commonly studied criteria for robust optimization are *absolute robustness* (or minimax), and *robust deviation* (or minimax regret). Yaman, Karason and Pinar [19] survey recent research for these methods and observe that many absolute robust formulations of problems with interval uncertainties can be solved with little more difficulty than the deterministic case. Bertsimas and Sim [3, 4] adopt the interval model of uncertainty and consider a restricted version of the absolute robustness criterion. This model of robustness limits the conservativeness of the robust solution by arguing that it is quite unlikely that all data elements will assume their worst possible values simultaneously; both the absolute robustness and robust deviation criteria may find solutions that have this property. Furthermore, solving a 0-1 mixed-integer linear program (MILP) under this model is no more difficult than solving the original problem.

In this paper we consider a version of the absolute robustness criterion that is naturally restricted by properties of the uncertain data. Specifically, we consider the case where the uncertain coefficients sum to a constant value. This restricted absolute robustness criterion is motivated by our recent work with MILP formulations for sensor placement in water distribution networks [1, 2, 17]. These MILPs rely on information like attack probabilities and population consumption statistics that are difficult to assess in detail, but for which we have good aggregate estimates. For example, water utilities have little information about the water consumption within a given household at a given hour, but they have much information about total water consumption within the entire water distribution network.

Our analysis of this restricted absolute robustness criterion is divided into

three cases that are motivated by the water security application:

- **unweighted uncertainty**: the objective has the form $\sum_{ij} \alpha_i x_{ij}$, for uncertain coefficients α_i;
- **linearly weighted uncertainty**: the objective has the form $\sum_{ij} \alpha_i p_j x_{ij}$, for uncertain coefficients α_i (p_j known with certainty); and,
- **bilinearly weighted uncertainty**: the objective has the form $\sum_{ij} \alpha_i \delta_j x_{ij}$, for uncertain coefficients α_i and δ_j.

We argue that the constant-sum constraints on the uncertain parameters make the solution to these problems qualitatively more interesting than formulations with a simple absolute robustness criterion. Furthermore, these problem formulations do not require a user-defined parameter to restrict the data uncertainties, so the robustness trade-off in this problem is more intrinsic than the trade-offs consider in the restricted absolute robustness models developed by Bertsimas and Sim [3, 4].

The rest of this paper is organized as follows. Section 2 describes and motivates the three robust MILP formulations. The subsequent three sections analyze these models, followed by some discussion of avenues for further research. In particular, we note that these robust MILP formulations apply to a wide range of problems for which modeling uncertainties can be expressed as interval uncertainties in the objective with constant-sum constraints.

2 Motivation for Robust MILP Models

Recent terrorist attacks have heightened concerns about whether community water systems are sufficiently well protected to ensure a safe and reliable supply of drinking water in the United States and around the world. Consequently, there is growing interest in the use of contaminant sensors to provide ongoing monitoring of water quality. A good sensor placement maximizes the information available for contamination control and remediation across a wide range of possible contamination scenarios, so that the fewest users consume contaminated water. A variety of MILP formulations have been developed to identify good sensor placement configurations [1, 2, 11, 17], and Berry et al. [1, 2] have recently demonstrated that MILPs can be effectively applied in the manner described here to solve moderately large sensor-placement problems.

We model an attack as the release of a large volume of harmful contaminant at a single junction in the network. For any particular attack, we assume that all points downstream of the release point can be contaminated; let $\mathcal{R}$ be the

set of pairs of junctions (i, j) such that junction j is downstream of junction i.[1] The primary decision variables for optimization are where to place each of a given number of sensors. Secondary binary decision variables are x_{ij}, for $(i, j) \in \mathcal{R}$, where $x_{ij} = 1$ if a contaminant injected at junction i can reach junction j without passing a sensor. Let X denote the set of feasible 0-1 x-values.

Consider the following parameter vectors over the nodes: (1) α_i is the probability of an attack at junction i, and (2) δ_i is the number of people who consume water at junction i. Note that α is not a probability in classical sense and is sometimes called an *attack weight*. It is estimated from expert judgement about the vulnerabilities in the network. δ is estimated from a sub-network that the node represents. Accurate estimation is complicated by the fact that only water consumption is measured, and what portion is drunk, versus used for other reasons, is estimated using other (uncertain) data.

The following two problems illustrate the data uncertainties that arise in these applications.

1. Minimize the expected extent of network contamination, as defined by the number of pipe junctions that become contaminated

$$\text{NC} : \min_{x \in X} \sum_{(i,j) \in \mathcal{R}} \alpha_i x_{ij},$$

2. Minimize the expected population exposed

$$\text{PE} : \min_{x \in X} \sum_{(i,j) \in \mathcal{R}} \alpha_i \delta_j x_{ij}.$$

In practice, the values of α and δ used in these sensor placement problems are not known. Although water utilities can accurately estimate the total population served by their water distribution network, most utilities do not currently maintain detailed statistics about the fraction of the population that is consuming water at each junction. Similarly, the risk assessment methodologies that are used to determine the attack weights provide a coarse assessment.

Let $\hat{\alpha}$ and $\hat{\delta}$ denote specified values of α and δ, respectively. These are sometimes called the *central values* (viz., most likely values, or best estimates). The central MILP is thus: $\min_{x \in X} \sum_{(i,j) \in \mathcal{R}} \hat{\alpha}_i \hat{\delta}_j x_{ij}$. As we let α and δ deviate from their central values, their possible values are limited by a *constant-sum constraint*, $\sum_i \alpha_i = 1$ and $\sum_j \delta_j = \sum_j \hat{\delta}_j$, respectively.

[1] To simplify our presentation, we assume a stable flow pattern for water in this paper. In particular, the models that we describe do not explicitly account for temporal effects. More detailed IP models are described by Berry et al. [1, 2].

In the next two sections, we consider robust formulations that use an absolute robustness criterion that is restricted in this sense. For objectives like NC, we show that the solution to a specific class of restricted absolute robustness problems is exactly the solution to the central MILP. More complex objectives, like PE, contain terms with products of uncertain parameters ($\alpha_i \delta_j$). The PE model is less complex if population values are presumed known, which is what we consider in Section 3.

The following notation is used in the next sections to define the domain of the robust optimization problems:

$$\mathcal{B}(\hat{c}, L, U) = \{c : L \leq c \leq U,\ \textstyle\sum_k c_k = \sum_k \hat{c}_k\},$$

where we suppose $L \leq \hat{c} \leq U$. The set $\mathcal{B}(\hat{c}, L, U)$ defines a multidimensional interval of uncertainty about a central value, $\hat{c}$, where c can be α or δ. One degree of freedom is lost by the constant-sum constraint.

3 Linearly Weighted Uncertainty

Consider the PE problem with known population values at the nodes. The following robust optimization formulation applies our restricted absolute robustness criteria:

$$\min_{x \in X} \max_{\alpha \in \mathcal{B}(\hat{\alpha}, \underline{\alpha}, \overline{\alpha})} \sum_{(i,j) \in \mathcal{R}} \alpha_i \delta_j x_{ij}, \tag{1}$$

where δ is presumed known. Since each uncertain term is weighted by a constant value, we say that this absolute robustness criteria uses *linearly weighted uncertainty*.

We can apply duality to reformulate problem (1) as follows:

$$\begin{aligned} \min \pi + \mu\overline{\alpha} - \lambda\underline{\alpha} \quad : \quad & x \in X \\ & \lambda, \mu \geq 0 \\ & \pi + \mu_i - \lambda_i = \sum_{j \in J(i)} \delta_j x_{ij} \text{ for all } i, \end{aligned}$$

where $J(i) = \{j : (i,j) \in \mathcal{R}\}$. Thus we can cast a linearly weighted robust optimization as a single MILP, having replaced the max with min. In particular, this is an augmented MILP formulation, which simply includes an extended objective and some additional side-constraints on dual variables from the maximization subproblem.

Alternatively, instead of casting problem (1) as one MILP, we can decompose it and solve the inner maximization problem to obtain $\alpha^*(x)$ for each x in the

outer minimization. The inner maximization problem can be solved simply by sorting the coefficients, which requires no more than $O(|\mathcal{R}| \ln |\mathcal{R}|)$ time. This may be computationally more efficient than the integrated formulation.

4 Unweighted Uncertainty

Consider the NC problem with interval uncertainties on the attack probabilities. The following robust optimization formulation applies our restricted absolute robustness criteria:

$$\min_{x \in X} \max_{\alpha \in \mathcal{B}(\hat{\alpha}, \underline{\alpha}, \overline{\alpha})} \sum_{(i,j) \in \mathcal{R}} \alpha_i x_{ij}. \tag{2}$$

We can clearly solve this problem using the methods described in Section 3 (letting $\delta_j = 1$ for all j), but in this section we consider the restricted case where we have intervals of uncertainty that are proportional to the central value vector. Let $\mathcal{P}(\hat{c}, \varepsilon) = \mathcal{B}(\hat{c}, (1-\varepsilon)\hat{c}, (1+\varepsilon)\hat{c})$ for $\varepsilon \in [0, 1)$. Given this restricted notion of interval uncertainty, we shall prove that the sensor placement decision for the central attack weight values $\hat{\alpha}$ remains optimal for any allowed variation.

Let $Q(\varepsilon)$ denote the robust optimization problem:

$$\min_{x \in X} \max_{c \in \mathcal{P}(\hat{c}, \varepsilon)} cx,$$

where X is any subset of binary vectors. (We use this more general notation, rather than α, because the result remains true even if the index set for x agrees completely with the index set for the uncertain coefficients, and the constant-sum is any value, not necessarily one.) The following theorem demonstrates that the solution to the central MILP (where $\hat{c}$ is the coefficient of x) is the solution to any robust formulation that allows percentage deviations within a constant proportion of its central value. Consequently, no additional computational effort is needed to generate a robust solution for these problems.

Theorem 4.1 *Let $\varepsilon \in (0, 1)$. Then, x^* is an optimal solution to $Q(0)$ if, and only if, x^* is an optimal solution to $Q(\varepsilon)$.*

Proof: We begin with some notation and general observations. Let $S = \sum_j \hat{c}_j$. Let $\sigma(x) = \{j : x_j \neq 0\}$ (called the "support set" of x). Also, let $\mathbf{1}$ denote the vector of all ones: $(1, 1, \ldots, 1)^\mathsf{T}$. The following identities follow from the definitions of S and σ: $\hat{c}x = \sum_{j \in \sigma(x)} \hat{c}_j = S - \sum_{j \notin \sigma(x)} \hat{c}_j$, and $\hat{c}(\mathbf{1} - x) = S - \hat{c}x = \sum_{j \notin \sigma(x)} \hat{c}_j$. Let $L = (1 - \varepsilon)$ and $U = (1 + \varepsilon)$.

The dual of $\max_{c\in\mathcal{P}(\hat{c},\varepsilon)} cx$ is

$$\begin{aligned}\min\ & \pi S + U\mu\hat{c} - L\lambda\hat{c} : \lambda,\mu \geq 0,\\ & \pi + \mu_j - \lambda_j = x_j \text{ for all } j = 1,\ldots,n.\end{aligned}$$

The dual variable π is associated with the constant-sum constraint, and λ, μ are associated with the lower and upper bound constraints on c, respectively.

Let x^0 be an optimal solution to $Q(0)$ and let x^ε be an optimal solution to $Q(\varepsilon)$. Our proof divides into two cases, depending on whether $\hat{c}x^0$ is greater or less than $\frac{1}{2}S$.

Case 1. $\hat{c}x^0 \geq \frac{1}{2}S$.

Consider the dual solution $\pi = 1$, $\mu = 0$, and $\lambda^\mathsf{T} = \mathbf{1} - x^0$. This is dual-feasible, where $\lambda \geq 0$ because $x^0 \leq \mathbf{1}$. The dual objective value is

$$\pi S + U\mu\hat{c} - L\lambda\hat{c} = S - L\hat{c}(\mathbf{1} - x^0) = S - L(S - \hat{c}x^0) = \varepsilon S + L\hat{c}x^0.$$

Therefore, we have

$$\max_{c\in\mathcal{P}(\hat{c},\varepsilon)} cx^0 \leq \varepsilon S + L\hat{c}x^0. \tag{3}$$

Now we define $c_j^\varepsilon = L\hat{c}_j$ for $j \notin \sigma(x^\varepsilon)$. Since we assume that $\hat{c}x^0 \geq \frac{1}{2}S$, it follows that $\hat{c}x^\epsilon \geq \frac{1}{2}S$, which implies that $\hat{c}(\mathbf{1} - x^\varepsilon) \leq \frac{1}{2}S$. Consequently, we have

$$\begin{aligned} c^\varepsilon x^\varepsilon &= S - \textstyle\sum_{j\notin\sigma(x^\varepsilon)} c_j^\varepsilon \\ &= S - L\textstyle\sum_{j\notin\sigma(x^\varepsilon)} \hat{c}_j \\ &= S - L(S - \hat{c}x^\varepsilon) = \varepsilon S + L\hat{c}x^\varepsilon, \end{aligned}$$

which gives us the bound:

$$\max_{c\in\mathcal{P}(\hat{c},\varepsilon)} cx^\varepsilon \geq \varepsilon S + L\hat{c}x^\varepsilon. \tag{4}$$

Using (3) and (4), we then obtain the following chain of inequalities:

$$\max_{c\in\mathcal{P}(\hat{c},\varepsilon)} cx^\varepsilon \geq \varepsilon S + L\hat{c}x^\varepsilon \geq \varepsilon S + L\hat{c}x^0 \geq \max_{c\in\mathcal{P}(\hat{c},\varepsilon)} cx^0 \geq \max_{c\in\mathcal{P}(\hat{c},\varepsilon)} cx^\varepsilon.$$

This establishes the following two results:

$$\begin{aligned} \max_{c\in\mathcal{P}(\hat{c},\varepsilon)} cx^0 &= \max_{c\in\mathcal{P}(\hat{c},\varepsilon)} cx^\varepsilon \text{ and} \\ \hat{c}x^0 &= \hat{c}x^\varepsilon, \end{aligned}$$

which completes this case.

Case 2. $\hat{c}x^0 \leq \frac{1}{2}S$.

The dual objective value of any dual-feasible solution is an upper bound on the primal value, cx^0. Choose $\pi = 0$, $\mu^\mathsf{T} = x^0$, and $\lambda = 0$. This is clearly dual-feasible, and its dual objective value is $U\hat{c}x^0$. Therefore,

$$\max_{c \in \mathcal{P}(\hat{c},\varepsilon)} cx^0 \leq U\hat{c}x^0. \tag{5}$$

Now consider the value of $\hat{c}x^\varepsilon$. If $\hat{c}x^\varepsilon \leq \frac{1}{2}S$, then define $c_j^\varepsilon = U\hat{c}_j$ for $j \in \sigma(x)$. This is feasible because we assume that $\hat{c}x^\varepsilon \leq \frac{1}{2}S$. It follows that $c^\varepsilon x^\varepsilon \geq U\hat{c}x^\varepsilon$, so we have $\max_{c \in \mathcal{P}(\hat{c},\varepsilon)} cx^\varepsilon \geq U\hat{c}x^\varepsilon$. Now if $\hat{c}x^\varepsilon \geq \frac{1}{2}S$, then define $c_j^\varepsilon = L\hat{c}_j$ for $j \notin \sigma(x^\varepsilon)$. It follows from our analysis in Case 1 that $\max_{c \in \mathcal{P}(\hat{c},\varepsilon)} cx^\varepsilon \geq \varepsilon S + L\hat{c}x^\varepsilon$. Taken together, this gives us the bound:

$$\max_{c \in \mathcal{P}(\hat{c},\varepsilon)} cx^\varepsilon \geq \min\left\{U\hat{c}x^\varepsilon, \varepsilon S + L\hat{c}x^\varepsilon\right\}. \tag{6}$$

Using Equations (5) and (6), we then obtain the following chain of inequalities:

$$\begin{aligned} \max_{c \in \mathcal{P}(\hat{c},\varepsilon)} cx^\varepsilon &\geq \min\left\{U\hat{c}x^\varepsilon, \varepsilon S + L\hat{c}x^\varepsilon\right\} \geq \min\left\{U\hat{c}x^0, \varepsilon S + L\hat{c}x^0\right\} \\ &= U\hat{c}x^0 \geq \max_{c \in \mathcal{P}(\hat{c},\varepsilon)} cx^0 \geq \max_{c \in \mathcal{P}(\hat{c},\varepsilon)} cx^\varepsilon. \end{aligned}$$

The equality in this chain follows from our assumption that $\hat{c}x^0 \leq \frac{1}{2}S$. From this chain of inequalities, it follows that $\max_{c \in \mathcal{P}(\hat{c},\varepsilon)} cx^0 = \max_{c \in \mathcal{P}(\hat{c},\varepsilon)} cx^\varepsilon$. Furthermore, either $U\hat{c}x^0 = U\hat{c}x^\varepsilon$ or $U\hat{c}x^0 = \varepsilon S + L\hat{c}x^\varepsilon$. But if $U\hat{c}x^0 = \varepsilon S + L\hat{c}x^\varepsilon$, then we have the following chain of inequalities:

$$U\hat{c}x^0 \leq \varepsilon S + L\hat{c}x^0 \leq \varepsilon S + L\hat{c}x^\varepsilon = U\hat{c}x^0,$$

from which it also follows that $\hat{c}x^0 = \hat{c}x^\varepsilon$. Consequently, we can conclude that $\hat{c}x^0 = \hat{c}x^\varepsilon$. ■

In terms of the sensor placement problem, this result implies that we can solve problem (2) by solving the central MILP,

$$\min_{x \in X} \sum_{(i,j) \in \mathcal{R}} \hat{\alpha}_i x_{ij},$$

because every optimal solution remains optimal for any variation allowance on the attack weights that are bounded by a common proportion of the central value, $\hat{\alpha}$. Some intuition as to why this is true is that the α-maximization increases the objective by a proportion of S, and is not affected by which x-values are one.

Following Yaman et al. [18], this result may be called a *permanent solution*. They found a spanning tree that remains optimal within interval data; we have a sensor placement that remains optimal under fixed-proportionate interval data and a constant-sum constraint. In our case, the fixed proportion is necessary — Theorem 4.1 is not true if we consider the more general set of uncertainties defined by $\mathcal{B}(\hat{c}, \underline{c}, \overline{c})$.

5 Bilinear Weighted Uncertainty

Consider the PE problem with interval uncertainties on both the attack weights and population. Although the total population remains fixed, in practice we may not have complete knowledge of the population's geographic distribution. Consequently, there may be uncertainties in the values of the δ_i. The following robust optimization formulation applies our restricted absolute robustness criteria considering uncertainties in both α and δ:

$$\min_{x \in X} \max_{\substack{\alpha \in \mathcal{B}(\hat{\alpha}, \underline{\alpha}, \overline{\alpha}) \\ \delta \in \mathcal{B}(\hat{\delta}, \underline{\delta}, \overline{\delta})}} \sum_{(i,j) \in \mathcal{R}} \alpha_i \delta_j x_{ij}. \tag{7}$$

We say that this absolute robustness criteria uses *bilinearly weighted uncertainty* because we have a bilinear maximization problem for the inner maximization.

This is a special case of the bilinear fractional program considered by Malivert [12]. The general problem is NP-hard, but this bilinear program has several simplifications. The main simplification is that the polyhedron separates for the two sets of variables, and each polyhedron (the ball) is much simpler than the general case — just one equation with bounds on the variables.

In Section 5.1, we show that in inner bilinear optimization problem remains NP-hard with this special structure and even with further special structure related to sensor placement in water networks. Section 5.2 gives a straightforward algorithm that reaches a local maximum, which need not be global. Section 5.3 gives a $1 + (U - L)^2/4$ approximation algorithm.

5.1 Complexity

In this section we prove that the inner bilinear optimization problem is NP-hard, where we consider the restricted subproblem:

$$\max_{\substack{\alpha \in \mathcal{P}(\hat{\alpha}, \varepsilon) \\ \delta \in \mathcal{P}(\hat{\delta}, \varepsilon)}} \sum_{(i,j) \in \mathcal{R}} \alpha_i \delta_j x_{ij}. \tag{8}$$

Both intervals have the same ε, and x satisfies the following two properties of our system:

1. Water networks are dags — we cannot have $(i, j) \in \mathcal{R}$ and $(j, i) \in \mathcal{R}$.
2. x satisfies transitive closure — if there is a path from i to j with no sensors ($x_{ij} = 1$) and there is a path from j to k with no sensors ($x_{jk} = 1$), then there is a path from i to k with no sensors ($x_{ik} = 1$).

We prove hardness by reduction from the clique problem. In particular, given a graph $G = (V, E)$ with $|V| = n$ and n even, we can determine whether this graph has an $n/2$ clique by solving a bilinear program with our special structure on a dag with transitive closure.

We construct a new, bipartite graph $G' = (A \bigcup P, E')$ as follows. For each vertex $v_i \in V$, create two sets of nodes:

Attack nodes. $A = \{A_i\}_{i=1}^n$, where $A_i = \{a_{i1}, \ldots, a_{in}\}$ with $\alpha_{a_{ij}} = 1$ and $\delta_{a_{ij}} = 0$ for $a_{ij} \in A_i$, $i = 1, \ldots, n$.

Population nodes. $P = \{P_i\}_{i=1}^n$, where $P_i = \{p_{i1}, \ldots, p_{in}\}$ with $\alpha_{p_{ij}} = 0$ and $\delta_{p_{ij}} = 1$ for $p_{ij} \in P_i$, $i = 1, \ldots, n$.

(We allow the constant-sum constraint on α to be n^2, rather than one, to keep notation simple. A simple scaling argument can be applied to allow this.) Further, for each $v_i \in V$, we add arcs to form a complete, directed bipartite graph between the associated attack and population nodes: put $(a_{ij}, p_{ik}) \in E'$ for $j = 1, \ldots, n$ and $k = 1, \ldots, n$. These are *structural edges* that relate vertices associated with the same node in G. For each edge $(v_i, v_j) \in E$, add arcs (a_{ij}, p_{ji}) and (a_{ji}, p_{ij}) in E'. These are *graph edges* that reflect the structure of the given graph (G) in which we are searching for an $n/2$ clique. There is at most one graph edge adjacent to any node in G'.

We thus have constructed G' as a dag with transitive closure on the arcs, and its incidence matrix will be x in our bilinear program. Figure 1 shows a 6-node graph and the constructed graph with 72 nodes (6 attack and 6 population per v_i for $i = 1, \ldots, 6$).

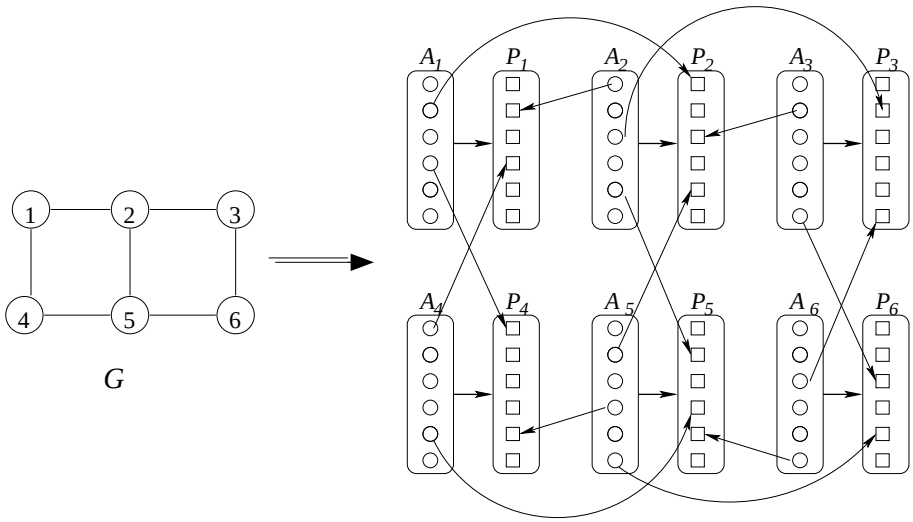

Figure 1: Example graph with induced constructed graph.

Summarizing the construction, we have two sets of nodes, each having n^2 members, and two sets of arcs, all directed from $a \in A$ to $p \in P$. One set of arcs (structural edges) have the form (a_{ij}, p_{ik}), so $x_{a_{ij}p_{ik}} = 1$ for all $v_i \in V$, $j, k = 1, \ldots, n$. The other set of arcs (graph edges) have the form (a_{ij}, p_{ji}) and (a_{ji}, p_{ij}) for $(v_i, v_j) \in E$, so $x_{a_{ij}p_{ji}} = x_{a_{ji}p_{ij}} = 1$ for all $(v_i, v_j) \in E$. All other $x_{fg} = 0$ for $f, g \in V'$. The bilinear program has the following objective value:

$$\begin{aligned}
\textstyle\sum_{(f,g)\in\mathcal{R}} \alpha_f \delta_g x_{fg} &= \textstyle\sum_{(a,p)\in E'} \alpha_a \delta_p x_{ap} \quad \text{(because } E' = \mathcal{R} \subset A \times P\text{)} \\
&= \textstyle\sum_{(a,p)\in E'} \alpha_a \delta_p \qquad \text{(because } x_{ap} = 1 \text{ for } (a,p) \in E'\text{)} \\
&= \textstyle\sum_{v_i \in V} \sum_{j=1}^{n} \sum_{k=1}^{n} \alpha_{a_{ij}} \delta_{p_{ik}} + \sum_{(v_i,v_j)\in E} \left(\alpha_{a_{ij}} \delta_{p_{ji}} + \alpha_{a_{ji}} \delta_{p_{ij}}\right)
\end{aligned}$$

The missing α and δ values (viz., α_p for $p \in P$ and δ_a for $a \in A$) are required to be zero anyway. This is because we set $\hat{\alpha}_P = 0$ and $\hat{\delta}_A = 0$. We shall prove that a solution to our bilinear program over this constructed graph with $\varepsilon = 1 - 1/n^3$ answers the question of whether G has an $n/2$ clique.

Let M be the bipartite incidence matrix for our constructed graph, so its rows correspond to A and its columns to P. (M is contained in x, which we no longer need.) Because only attack nodes allow nonzero α and only population nodes allow nonzero δ, we can redistribute α values among only the attack nodes (rows of M), and we can redistribute δ values among only population nodes (columns of M). Thus, α is constrained by $\mathcal{P}(\hat{\alpha}, \varepsilon)$ to satisfy $1 - \varepsilon \leq \alpha_{a_{ij}} \leq 1 + \varepsilon$ for row

a_{ij} of M. Similarly, δ is constrained by $\mathcal{P}(\hat{\delta}, \varepsilon)$ to satisfy $1 - \varepsilon \leq \delta_{p_{kl}} \leq 1 + \varepsilon$ for column p_{kl} of M.

We say a row or column of the matrix M is *selected* if respectively the α or δ value is $1 + \varepsilon$. An entry of the matrix is selected if both its row and column are selected. We shall always set a variable to one of its bound values, so the number of rows (columns) selected must be $\frac{n^2}{2}$ to satisfy the constant-sum constraints. This is illustrated in figure 2.

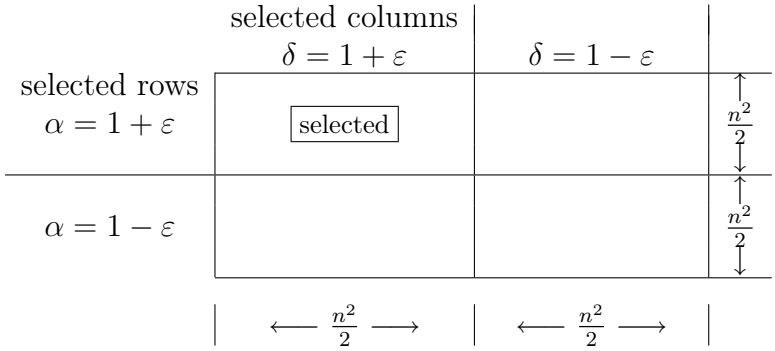

Figure 2: Partition of M upon selecting rows, columns, and elements.

A vertex $v_i \in G$ is selected if all rows and all columns associated with it (all $a_{ij} \in A_i$ and $p_{ik} \in P_i$) are selected. v_i is *partially-selected* if at least one row or column associated with it is selected and at least one such row or column is not selected.

We shall partition M in such a way that we can prove that this assignment of values to α and δ is optimal in the bilinear program such that the objective value achieves a certain value if, and only if, there is an $n/2$ clique in G. We require the following preliminary lemmas to prove this claim. In what follows, let $K = \frac{1}{2}n^3 + \frac{n}{2}(\frac{n}{2} - 1)$.

Lemma 5.1 *If there is an $n/2$ clique in G, the optimal value of the bilinear program is at least $K(1 + \varepsilon)^2$.*

Proof: Assume there is an $n/2$ clique in G. Select all the vertices in the clique, so that $\alpha_{a_{ij}} = \delta_{p_{ij}} = 1 + \varepsilon$ for all v_i in the clique and all $j = 1, \ldots, n$, and $\alpha_{a_{kj}} = \delta_{p_{kj}} = 1 - \varepsilon$ for all v_k not in the clique and all $j = 1, \ldots, n$. Now count the ones in M among the selected elements. There are $\frac{1}{2}n^3$ ones from the structural edges, $A_i \rightarrow P_i$ for $i = 1, \ldots, n/2$. The clique has $\frac{1}{2}\frac{n}{2}(\frac{n}{2} - 1)$ edges and each edge corresponds to two arcs (graph edges), so the selected elements of M have another $\frac{n}{2}(\frac{n}{2} - 1)$ ones. Adding these, we obtain K ones in the selected portion

of M, so the bilinear objective must be at least $K(1+\varepsilon)^2$. (The other terms in the objective have $(1-\varepsilon)^2$ and $(1-\varepsilon)(1+\varepsilon)$, corresponding to ones in the other three portions of M. That sum is positive, so we obtain the bound, as claimed.) ■

Lemma 5.2 *There is an extreme point optimal solution to our bilinear problem, in which every α and δ is at $1+\varepsilon$ or $1-\varepsilon$, with an equal number of each in both cases.*

Proof: It is already known [12] that the bilinear program has a solution at an extreme point of its domain. In our case, the extreme points are the cartesian product of the extreme points of $\mathcal{P}(\hat{\alpha},\varepsilon)$ and those of $\mathcal{P}(\hat{\delta},\varepsilon)$. An extreme point of $\mathcal{P}(\hat{\alpha},\varepsilon)$ has every variable, except at most one, at a bound value. Because n is even the number of nodes in G' with nonzero bounds in $\mathcal{P}(\hat{\alpha},\varepsilon)$ is even; the same applies to $\mathcal{P}(\hat{\delta},\varepsilon)$. The constant sum constraint then requires an even split in variables at $1+\varepsilon$ and those at $1-\varepsilon$. ■

Lemma 5.3 *If an extreme point optimal solution attains a value of at least $K(1+\varepsilon)^2$, at least K matrix elements with value 1 (edges of G') have been selected.*

Proof: Assume an extreme point attains a value of at least $K(1+\varepsilon)^2$. The entire matrix has at most $n^3+n(n-1)$ ones. So, selecting $K-1$ ones leaves at most $\frac{1}{2}n^3+\frac{n}{2}(\frac{n}{2}-1)+1$ unselected ones. For any $n\geq 2$, $\frac{1}{2}n^3+\frac{n}{2}(\frac{n}{2}-1)+1<n^3$. Therefore, fewer than n^3 ones would be unselected, and each contributes a value of at most $(1+\varepsilon)(1-\varepsilon)$ to the objective. But, $n^3(1-\varepsilon)=1$ by definition of ε, so the total value of the unselected ones is at most $(1+\varepsilon)<(1+\varepsilon)^2$. Finally, if there are fewer than K selected ones in the extreme point, the bilinear value is at most

$$(K-1)(1+\varepsilon)^2+(1+\varepsilon)<(K-1)(1+\varepsilon)^2+(1+\varepsilon)^2=K(1+\varepsilon)^2.$$

■

Theorem 5.1 *The bilinear problem has a maximum value of at least $K(1+\varepsilon)^2$ if, and only if, there is an $n/2$ clique in G.*

Proof: Lemma 5.1 establishes the "if" direction. We now prove that if there is an optimal extreme point with K selected ones, there is an optimal extreme point with K selected ones and $n/2$ selected vertices. The "only if" direction then follows from Lemmas 5.2 and 5.3.

Consider a minimal counterexample. This is a counterexample with the fewest possible partially-selected vertices. If there are multiple such counterexamples, this is the one that has a partially-selected vertex with the smallest number of selected rows plus selected columns.

Consider a partially-selected vertex v_i with the fewest selected rows and columns. Suppose vertex v_i has no rows selected. Then find a partially-selected vertex v_j with at least one selected row and fewer than n selected columns. Such a vertex v_j must exist by Lemma 5.2. More specifically, there are exactly $n^2/2$ selected rows and exactly $n^2/2$ selected columns. Each vertex has at most n selected rows. Therefore, since v_i has no selected rows, there must be at least $n/2$ vertices with at least one selected row. Because v_i has at least 1 selected column, there must be at most $n/2-1$ vertices for which all columns are selected. We now unselect a column in vertex v_i and select a column in vertex v_j. This unselects at most one 1 (corresponding to the edge for the column originally selected), and selects at least one 1 (in the submatrix for vertex v_j), so we still have at least K selected ones. In a single row or column swap, all the newly-selected ones now contribute the greatest value of $(1+\varepsilon)^2$ to the objective (before they contributed the middle value of $(1-\varepsilon^2)$. All unselected ones previously contributed the greatest value and now contribute the middle value. Therefore, as long as the number of selected ones after the swap does not decrease, the objective function also does not decrease. Thus, we obtain a smaller counterexample. Similarly, if no columns are selected in vertex v_i, swap a row associated with a partially-selected vertex with at least one column selected and room to select another row. Therefore, the minimal vertex has at least one row and one column selected

Suppose the minimum vertex v_i has $r_i \geq 1$ rows selected and $c_i \geq 1$ columns selected. Suppose there is a partially-selected vertex v_j with $r_j > r_i$ selected rows. If v_j has fewer than n selected columns, we can swap a column from v_i to v_j and get a smaller counterexample. If v_j has n selected columns, then swapping a row from v_i to v_j again gives a smaller counterexample provided $c_i < n$. If $c_i = n$, then r_i is the smallest among all partially-selected vertices (that is, all other partially-selected vertices v_j have $r_j \geq r_i$). There must be at least $n/2+1$ partially-selected vertices (because we are not done). These cannot all have n selected columns. Therefore, there exists a vertex v_j with $c_j < n$. We must also have $r_j > r_i$ if $r_j = r_i$ and $c_j < n$, then v_j would be the minimal vertex (recall $c_i = n$). A similar argument holds if there is a v_j with more than c_j columns selected.

Therefore, in any counterexample all partially-selected vertices have the same number of rows and columns selected as our minimal vertex v_i, say k rows and l columns. Pick one such vertex and swap a row and a column. By unselecting a row, we unselect one 1s in our vertex submatrix, and at most one edge. Then,

unselecting the column unselects at most $k-1$ 1s in our vertex submatrix, and at most one edge. When we select a row in our new vertex, we select one 1s in that vertex submatrix. When we select a column in our new vertex, we select $k+1$ 1s (including the 1 for the new row and column) in the vertex submatrix. So, we have unselected at most $k+l+1$ 1s, and selected at least $k+l+1$ 1s, thus yielding a smaller counterexample.

In all cases, it is possible to reduce our counterexample to a smaller case with at least as many 1s. This implies that there is no counterexample, and if there is an extreme point with K selected 1s, there is an extreme point with K selected 1s and $n/2$ selected vertices.

If there are K selected 1s and $n/2$ selected vertices, there must be a clique: the $n/2$ selected vertices give us $n^3/2$ selected 1s. That leaves no other selected rows or columns, and $\frac{n}{2}(\frac{n}{2}-1)$ selected 1s from the edges, which is all the possible arcs, hence a clique. Thus, both directions in our theorem have been established. ■

We have thus established that our bilinear program is NP-hard despite the simple domain. Our proof used the allowance of ε arbitrarily close to 1. In practice, we generally want $\varepsilon \leq \kappa$ for some $\kappa < 1$, such as $\kappa = \frac{1}{4}$. In that case our proof does not apply, and the hardness remains an open question.

5.2 Alternating Ascent

Although we have shown that the bilinear program (7) is NP-hard, we may still need to solve this in a practical manner. We consider a heuristic search strategy for solving the bilinear program for a given set of x values, which could be used as an inner loop for a general-purpose search strategy (e.g. meta-heuristic methods).

Figure 3 describes a simple local search strategy for solving the bilinear program for given values of x. Given x, one way to seek a solution is to alternate between α and δ, solving an LP in each iteration. Each maximization is a linear program, the same as the linearly weighted case, but with new sorting (maybe) needed each time. After a finite number of iterations, we terminate with extreme point optimal solutions to each polytope.

It is easy to show that optimal values can be found among the extreme points of $\mathcal{B}(\hat{\alpha}, \underline{\alpha}, \overline{\alpha})$ and $\mathcal{B}(\hat{\delta}, \underline{\delta}, \overline{\delta})$ by applying elementary theory of LP and noting a necessary condition for (α^*, δ^*) to be optimal [12]:

$$\alpha^* \in \operatorname{argmax} \left\{ \textstyle\sum_{(i,j)} \alpha_i \delta_j^* x_{ij} \ : \alpha \in \mathcal{B}(\hat{\alpha}, \underline{\alpha}, \overline{\alpha}) \right\}$$

$$\delta^* \in \operatorname{argmax} \left\{ \textstyle\sum_{(i,j)} \alpha_i^* \delta_j x_{ij} \ : \delta \in \mathcal{B}(\hat{\delta}, \underline{\delta}, \overline{\delta}) \right\}$$

Alternating Ascent Algorithm

0. Initialize. Choose $\alpha^0 \in \mathcal{B}(\hat{\alpha}, \underline{\alpha}, \overline{\alpha})$, and set $k = 0$.

1. Solve for $\boldsymbol{\delta}$. Given α^k, compute

$$\delta^k \in \operatorname{argmax} \left\{ \textstyle\sum_{(i,j)\in\mathcal{R}} \alpha_i^k \delta_j x_{ij} \ : \delta \in \mathcal{B}(\hat{\delta}, \underline{\delta}, \overline{\delta}) \right\}$$

2. Solve for $\boldsymbol{\alpha}$. Given δ^k, compute

$$\alpha^{k+1} \in \operatorname{argmax} \left\{ \textstyle\sum_{(i,j)\in\mathcal{R}} \alpha_i \delta_j^k x_{ij} \ : \alpha \in \mathcal{B}(\hat{\alpha}, \underline{\alpha}, \overline{\alpha})) \right\}$$

3. Increment $k \leftarrow k + 1$ and repeat steps 1–3 until $\delta^k = \delta^{k-1}$ ($k > 0$, step 1) or $\alpha^{k+1} = \alpha^k$ (step 2).

Figure 3: An alternating ascent algorithm for the bilinear program (7), for a given value of x.

Unfortunately, this is not sufficient. If we terminate the algorithm with (α^*, δ^*), all we can say is that this satisfies the above necessary condition. We cannot rule out the possibility that a simultaneous change in α and δ would increase the objective value. We shall call a point that satisfies these necessary conditions an *alternating ascent solution.*

We have found examples where the alternating ascent solution is not a global maximum. Although the objective cannot not be improved by changing α or δ, keeping the other fixed, a simultaneous change in α and δ (just to neighboring extreme points) was able to increase the objective value. We ran some experiments to see how often this occurs, and the particular runs indicate that alternating ascent gets trapped at a local solution from relatively few starting points. In all cases, an improvement could be found by combining neighbors of α in $\mathcal{B}(\hat{\alpha}, \underline{\alpha}, \overline{\alpha})$ with neighbors of δ in $\mathcal{B}(\hat{\delta}, \underline{\delta}, \overline{\delta})$. While inconclusive, this indicates that it may be possible to identify conditions under which such a search procedure would reach the global maximum. The extreme points of the bilinear program are easy to characterize, so it may be possible to exploit this information to guide the local search and jump to a non-adjacent extreme point to avoid terminating at a solution that is not a global maximum.

5.3 Linear Programming Relaxation

In this section we describe a linear relaxation of the bilinear case. We demonstrate that our LP formulation produces solutions that are provably near-optimal for any given values of x. Thus, this LP can be used to provide a performance-guaranteed approximation algorithm for the bilinear program (7).

We use I and J to denote the range of i and j, respectively. Instead of the bilinear term $\alpha_i\delta_j$ we use the variable γ_{ij} to represent its value. Since we also use variables α_i and δ_j that represent the factors in these bilinear terms, the main issue is that of relating these two sets of variables. With this notation, the objective function for a fixed $x = (x_{ij})$ becomes

$$\max \sum_{(i,j)\in\mathcal{R}} \gamma_{ij} x_{ij}. \tag{9}$$

We now proceed to give the LP constraints. Denote the bound values:

$$\begin{aligned} \alpha_i^l &= L\hat{\alpha}_i,\ i \in I \\ \alpha_i^u &= U\hat{\alpha}_i,\ i \in I \\ \delta_j^l &= L\hat{\delta}_j,\ j \in J \\ \delta_j^u &= U\hat{\delta}_j,\ j \in J. \end{aligned} \tag{10}$$

We ensure that the coefficient values are feasible by writing them as convex combinations of the lower and upper bounds:

$$\begin{aligned} \alpha_i &= \lambda_i\alpha_i^l + (1-\lambda_i)\alpha_i^u,\ i \in I \\ \delta_j &= \mu_j\delta_j^l + (1-\mu_j)\delta_j^u,\ j \in J \\ 0 &\le \lambda, \mu \le 1. \end{aligned} \tag{11}$$

The next two sets of constraints relate the singly-indexed variables α and δ to the doubly-indexed γ.

$$\begin{gathered} \gamma_{ij} = \nu_{ij}^{00}\alpha_i^l\delta_j^l + \nu_{ij}^{01}\alpha_i^l\delta_j^u + \nu_{ij}^{10}\alpha_i^u\delta_j^l + \nu_{ij}^{11}\alpha_i^u\delta_j^u,\ (i,j) \in \mathcal{R} \\ \nu^{00}, \nu^{01}, \nu^{10}, \nu^{11} \ge 0. \end{gathered} \tag{12}$$

$$\begin{aligned} \nu_{ij}^{00} + \nu_{ij}^{01} &= \lambda_i,\ i \in I \\ \nu_{ij}^{10} + \nu_{ij}^{11} &= 1-\lambda_i,\ i \in I \\ \nu_{ij}^{00} + \nu_{ij}^{10} &= \mu_j,\ j \in J \\ \nu_{ij}^{01} + \nu_{ij}^{11} &= 1-\mu_j,\ j \in J. \end{aligned} \tag{13}$$

The sum constraints are naturally linear:

$$\begin{aligned} \sum_{i\in I} \alpha_i &= \sum_{i\in I} \hat{\alpha}_i,\ j \in J \\ \sum_{j\in J} \delta_j &= \sum_{j\in J} \hat{\delta}_j,\ i \in I, \end{aligned} \tag{14}$$

and there is another set of sum constraints for the doubly-indexed γ:

$$\begin{aligned} \sum_{i\in I} \gamma_{ij} &= \sum_{i\in I} \hat{\alpha}_i \delta_j,\ j \in J \\ \sum_{j\in J} \gamma_{ij} &= \sum_{j\in J} \hat{\delta}_j \alpha_i,\ i \in I. \end{aligned} \tag{15}$$

Taken together, the objective (9) and the constraints (11)–(15) provide a relaxation of the bilinear problem (7) for a fixed x. In order to see this, pick any feasible solution (α, δ) of the bilinear program and let $\gamma_{ij} = \alpha_i \delta_j$ for $(i,j) \in \mathcal{R}$. Setting $\nu_{ij}^{00} = \lambda_i \mu_j$, $\nu_{ij}^{10} = (1-\lambda_i)\mu_j$, $\nu_{ij}^{01} = \lambda_i(1-\mu_j)$ and $\nu_{ij}^{11} = (1-\lambda_i)(1-\mu_j)$ directly satisfies the constraints (12). Multiplying out the i- and j-constraints in (13) then shows that the constraint (11) is satisfied as well.

Since the LP is a relaxation of the bilinear program (7), its optimal objective value is an upper bound on the optimum of the bilinear program. The following theorem shows a bound on the optimality gap of the linear formulation.

Theorem 5.2 *Let z^{LP} be the cost of an optimal solution to the LP (9)–(15) and z^* the cost of an optimal solution to the bilinear program (7). Then $z^{LP}/z^* \leq 1 + (U-L)^2/4$.*

Proof: From the constraints (11)–(13), it follows that

$$\gamma_{ij} - \alpha_i \delta_j = (\nu_{ij}^{00}\nu_{ij}^{11} - \nu_{ij}^{01}\nu_{ij}^{10})(\alpha_i^u - \alpha_i^l)(\delta_j^u - \delta_j^l). \tag{16}$$

The value of the first factor in the right-hand side expression is upper-bounded by $\frac{1}{4}$. (This is true because $\frac{1}{4}$ is the maximum value of $ab - cd$, given $a+b+c+d = 1$ and $a, b, c, d \geq 0$.) The product of the two remaining factors can be rewritten using (10):

$$(U\hat{\alpha}_i - L\hat{\alpha}_i)(U\hat{\delta}_j - L\hat{\delta}_j) = (U-L)^2 \hat{\alpha}_i \hat{\delta}_j.$$

Taken together, these two claims give

$$\sum_{ij} \gamma_{ij} x_{ij} \leq \sum_{ij} \alpha_i \delta_j x_{ij} + \frac{(U-L)^2}{4} \sum_{ij} \hat{\alpha}_i \hat{\delta}_j x_{ij} \leq \left(1 + \frac{(U-L)^2}{4}\right) z^*,$$

where the last inequality follows because both (α, δ) and $(\hat{\alpha}, \hat{\delta})$ are feasible for the bilinear program. ■

The above proof does not use the constant-sum constraints (14) and (15). While we have not been able to find a general result that would improve on Theorem 5.2, these inequalities do strengthen the relaxation and in some (highly restrictive) cases make it exact.

The values (α^*, δ^*) that form a part of the optimal solution to the LP naturally provide a feasible solution to the bilinear program and thus a lower bound $\sum_{(i,j)\in\mathcal{R}} \alpha_i^* \delta_j^* x_{ij}$ on its optimal value. This observation can be used to develop a performance-guaranteed approximation algorithm for the bilinear program (7); the procedure is standard so we just sketch it very briefly. Consider the dual of our linear program. This is a minimization problem, so we can leave the x variable and solve for a minimum over all x using a branch-and-bound search process. Although our LP may provide a worse relaxation when the x are relaxed to non-integral values, the bound remains valid for specific integral values. Hence, we are assured that this search process will find an integral solution whose value is within $1 + (U - L)^2/4$ of optimal.

6 Discussion

There are many possible objectives for sensor placement that reflect various costs and risks of an attack on a network [17]. Previous work has considered problem formulations that minimize the volume of water consumed before detection [8], minimize the time to detection [10], and minimize the population exposed to contaminants before detection [1, 11]. This paper presents a foundation upon which these objectives, taken separately or multiply, can be considered in a manner that addresses data uncertainties.

Robust optimization addresses a need to hedge against uncertainty, and these uncertainties are a fundamental property of sensor placement problems. Data like attack weights and population distribution are based on expert judgement and incomplete source data. Although a number of criteria for robust optimization have been studied, the interval data model, with a constant-sum constraint, fits these sensor placement problems well.

The simplest case that we have considered is the unweighted uncertainty model (§4), where only the attack weights are uncertain and the objective NC is the expected number of nodes that are contaminated without detection. If the interval is restricted to a fixed-proportion of the central vector, we have shown this problem has a permanent solution. In a real sense, this suggests that this is not the right

model for the sensor placement decision problem; our intuition suggests that data uncertainties ***should*** affect our decision.

Although robustness issues can be addressed in the linearly weighted case, the bilinear model remains the most interesting because the PE formulation is well-motivated in practice. However, our results here do not provide a complete solution for this case. We proved that the maximization subproblem is NP-hard even with the simplifications of being in a ball with fixed-proportionate bounds and one constant-sum constraint. However, it is unclear whether this maximization subproblem remains NP-hard we require that $\varepsilon \leq \kappa$ for some $\kappa < 1$. Our analysis allows for ε arbitrarily close to 1.

Our LP relaxation of the bilinear case gives a guaranteed bound on our approximate solution that has practical value. However, this should be compared with the LP relaxation one can obtain by using McCormick's bounds [13, 16] to assess the quality of solutions and speed of obtaining them. We have observed that our LP relaxation can "often" obtain the correct objective value, and further analysis could reveal conditions under which this is true.

Acknowledgements

Sandia National Laboratories is a multipurpose laboratory operated by Sandia Corporation, a Lockheed-Martin Company, for the United States Department of Energy under contract DE-AC04-94AL85000.

References

[1] J. Berry, L. Fleischer, W. E. Hart, and C. A. Phillips. Sensor placement in municipal water networks. In P. Bizier and P. DeBarry, editors, *Proceedings of the World Water and Environmental Resources Congress*. American Society of Civil Engineers, 2003.

[2] J. Berry, W. E. Hart, C. A. Phillips, and J. Uber. A general integer-programming-based framework for sensor placement in municipal water networks. In *Proceedings of the World Water and Environment Resources Conference*, 2004.

[3] D. Bertsimas and M. Sim. Robust discrete optimization and network flows. *Mathematical Programming Series B*, 98:49–71, 2003.

[4] D. Bertsimas and M. Sim. The price of robustness. *Operations Research*, 52:35–53, 2004.

[5] G. B. Dantzig. Linear programming under uncertainty. *Management Science*, 1(3, 4):197–206, 1955.

[6] S. Ferson, C. A. Joslyn, J. C. Helton, W. L. Oberkampf, and K. Sentz. Challenge problems: Uncertainty in system response given uncertain parameters. *Reliability Engineering and System Safety*, 85:11–19, 2004.

[7] F. O. Hoffman and J. S. Hammonds. Propagation of uncertainty in risk assessments: The need to distinguish between uncertainty due to lack of knowledge and uncertainty due to variability. *Risk Analysis*, 14(5):707–712, 1994.

[8] A. Kessler, A. Ostfeld, and G. Sinai. Detecting accidental contaminations in municipal water networks. *Journal of Water Resources Planning and Management*, 124(4):192–198, 1998. .

[9] P. Kouvelis and G. Yu. *Robust Discrete Optimization and Its Applications*. Kluwer Academic Publishers, Norwell, MA, 1996.

[10] A. Kumar, M. L. Kansal, and G. Arora. Discussion of 'detecting accidental contaminations in municipal water networks'. *Journal of Water Resources Planning and Management*, 125(4):308–310, 1999. .

[11] B. Lee and R. Deininger. Optimal locations of monitoring stations in water distribution system. *Journal of Environmental Engineering*, 118(1):4–16, 1992.

[12] C. Malivert. An algorithm for bilinear fractional problems. Technical Report 1998-03, Université de Limoges, Cedex, France, 1998.

[13] G. McCormick. *Nonlinear Programming: Theory, Algorithms, and Applications*. John Wiley & Sons, New York, NY, 1983.

[14] W. L. Oberkampf, S. M. DeLand, B. M. Rutherford, K. V. Diegert, and K. F. Alvin. Error and uncertainty in modeling and simulation. *Reliability Engineering and System Safety*, 75:333–357, 2002.

[15] W. D. Rowe. Understanding uncertainty. *Risk Analysis*, 14(5):743–750, 1994.

[16] H. S. Ryoo and N. V. Sahinidis. Global optimization of nonconvex NLPs and MINLPs with applications in process design. *Computers Chemical Engineering*, 19(5), 1995.

[17] J.-P. Watson, H. J. Greenberg, and W. E. Hart. A multiple-objective analysis of sensor placement optimization in water networks. In *Proceedings of the World Water and Environment Resources Congress*. American Society of Civil Engineers, 2004.

[18] H. Yaman, O. E. Karasan, and M. C. Pinar. The robust minimum spanning tree problem with interval data. *Operations Research Letters*, 29:31–40, 2001.

[19] H. Yaman, O. E. Karasan, and M. C. Pinar. Restricted robust optimization for maximization over uniform matroid with interval data uncertainty. Technical report, Bilkent University Technical Report, 2004.