Faster Solutions to the Interdiction Defense Problem using Suboptimal Solutions

Joshua Ong[a], Andrew Mastin[a], William Yang[a], Jean-Paul Watson[a]

[a]*Lawrence Livermore National Laboratory,*

Abstract

The interdiction defense (ID) problem solves a defender-attacker-defender model where the defender and attacker share the same set of components to harden and target. We build upon the best response intersection (BRI) algorithm by developing the BRI with suboptimal solutions (BRI-SS) algorithm to solve the ID problem. The BRI-SS algorithm utilizes off-the-shelf optimization solvers that may return suboptimal solutions at no additional computation cost. We derive novel cuts from suboptimal solutions, generally reducing the number of iterations required for the algorithm to converge while maintaining optimality guarantees. We also present a heuristic that utilizes all obtained suboptimal solutions to select the next defense to evaluate at each iteration of the algorithm. We perform computational experiments applied to power grid interdiction on standard test cases. Our results demonstrate that the BRI-SS algorithm consistently outperforms the BRI algorithm across all test cases.

Keywords: interdiction, multi-level optimization, power grid resilience

1. Introduction

The *interdiction defense* (ID) *problem* solves a defender-attacker-defender (DAD) model where the defender hardens and the attacker targets the same set of components. This three-stage model proceeds sequentially: first the defender hardens components to improve resilience, thereafter the attacker targets unprotected components in order to maximize disruption, and finally the defender attempts to stabilize the system by solving a recourse problem. Note the ID problem is formulated comparably to the interdiction problem with fortification [1], but with the added assumption that the defender and attacker share the same set of assets. Applications of interdiction and fortification formulations to critical infrastructure, such as roads, communication systems, and power grids, are discussed in [2].

Descriptions for exact algorithms to solve DAD problems such as Benders decomposition method can be found in [3]. Another common approach is the column-and-constraint generation (CCG) method, which has been widely implemented in a variety of DAD contexts [4]. The Best Response Intersection (BRI) algorithm solves the ID problem by implicitly enumerating all feasible defenses [5]. For every defense considered, a subproblem is solved to determine the *attacker's best response*, and a valid inequality is generated.

[1]This work was funded by the US Department of Energy's Office of Electricity, under the North American Energy Resilience Model (NAERM) and Advanced Grid Modeling (AGM) programs. This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under contract DE-AC52-07NA27344. Lawrence Livermore National Security, LLC.

Computational studies show that the BRI algorithm can perform an order of magnitude faster than comparable CCG algorithms [5]. The BRI algorithm demonstrates improved scalability as it only adds one constraint per iteration while CCG algorithms must duplicate the constraints of the entire recourse problem at each iteration.

In this work, we build upon the BRI algorithm by utilizing suboptimal solutions obtained from off-the-shelf mixed integer program (MIP) solvers. We modify the BRI algorithm to include novel cuts derived from suboptimal solutions while maintaining global optimality and finite convergence guarantees. We also present an additional heuristic that selects the next defense to evaluate at each iteration of the algorithm, which we find improves scalability in practice. Although previous work explores using suboptimal solutions in inexact algorithms [6], no algorithms utilize our specific approach to solve DAD problems, to the best of our knowledge. Our contributions are as follows:

1. We develop the best response intersection with suboptimal solutions (BRI-SS) algorithm, which utilizes suboptimal solutions to improve scalability compared to [5] while maintaining global optimality and finite convergence guarantees.
2. We demonstrate the computational improvements of the BRI-SS algorithm by showing that it consistently outperforms the BRI algorithm by achieving runtimes up to 4 times faster on power grid interdiction test cases.

2. Problem Formulation and Algorithm

2.1. Problem Formulation

Let T denote the set of shared components in an interconnected system that a defender can harden and an attacker can destroy. The feasible sets of defenses and attacks are denoted as $\mathcal{D}_{\mathrm{F}} \subseteq 2^{|T|}$ and $\mathcal{A}_{\mathrm{F}} \subseteq 2^{|T|}$, respectively. Let $C(A, D)$ denote the cost incurred by the system if attack A is performed after defense D is implemented. The attacker's best response (ABR) problem maximizes the cost function given a fixed defense D and is defined as

$$\mathrm{ABR}(D) \ := \underset{A \in \mathcal{A}_{\mathrm{F}}}{\operatorname{argmax}} \ C(A, D). \tag{1}$$

While the defender wishes to minimize the cost function, the attacker seeks to maximize the cost function. The interdiction defense problem (ID) is given by

$$\mathrm{ID}(\mathcal{A}_{\mathrm{F}}, \mathcal{D}_{\mathrm{F}}, C) \ := \min_{D \in \mathcal{D}_{\mathrm{F}}} \max_{A \in \mathcal{A}_{\mathrm{F}}} \ C(A, D) = \min_{D \in \mathcal{D}_{\mathrm{F}}} C(\mathrm{ABR}(D), D). \tag{2}$$

The Best Response Intersection (BRI) algorithm from [5] relies on two conditions to solve the ID problem. Condition 1 establishes that given a defense D, the original cost function can be equivalently expressed as a new cost function that is defined over the restricted feasible set of attacks.

Condition 1. *The function $C : \mathcal{A}_{\mathrm{F}} \times \mathcal{D}_{\mathrm{F}} \to \mathbb{R}$ can be represented by an equivalent function $\widetilde{C} : 2^T \to \mathbb{R}$ where for any attack $A \in \mathcal{A}_{\mathrm{F}}$ and defense $D \in \mathcal{D}_{\mathrm{F}}$, $C(A, D) = \widetilde{C}(A \setminus D)$.*

Condition 2 asserts that for any given attack that intersects with the defense, there exists an alternative attack that does not intersect with the defense and has an equal cost.

Condition 2. *For any $A \in \mathcal{A}_{\mathrm{F}}$ and $D \in \mathcal{D}_{\mathrm{F}}$ such that $|A \cap D| \geq 1$, there exists an $A' \in \mathcal{A}_{\mathrm{F}}$ such that $A' \cap D = \emptyset$ and $C(A, D) = C(A', D)$.*

The BRI algorithm improves on an exhaustive search for all feasible defenses of the ID problem by introducing two constraints on the set of feasible defenses at each iteration of the algorithm. Let u represent a lower bound to the ID problem, given some A^*. Then for any defense to improve,

$$C(A^*, D) < u. \tag{3}$$

Let $\mathcal{A}_{\mathrm{E}}$ be the set of evaluated attacker's best responses. Then to ensure a reduction in cost, every subsequent defense must intersect $A \in \mathcal{A}_{\mathrm{E}}$. This motivates the intersection constraint:

$$|D \cap A| \geq 1 \quad \forall A \in \mathcal{A}_{\mathrm{E}}. \tag{4}$$

2.2. Best Response Intersection with Suboptimal Solutions

Extending upon the BRI solution approach, we present the best response intersection with suboptimal solutions (BRI-SS) algorithm in Algorithm 1. The modification is motivated by ABR problems with a MIP structure. While the BRI algorithm utilizes the optimal solution $A_0 := \mathrm{ABR}(D)$ to generate constraint (4), off-the-shelf MIP solvers may obtain sub-optimal solutions with no additional computational cost. Let J denote the user-supplied parameter specifying the maximum number of suboptimal solutions to store. The BRI-SS algorithm modifies the BRI algorithm by solving $\mathrm{ABR}(D)$ to obtain additional suboptimal solutions, $A_1, ..., A_j$, where $j \leq J$. Without loss of generality, we assume the solutions are sorted in decreasing cost such that $C(A_0, D) \geq C(A_1, D) \geq ... \geq C(A_j, D)$. Note that the costs observed are all for the same defense D, as D does not intersect with any $A_0, ..., A_j$. Let H be the sequence of indices that sorts all observed attacks in descending order of cost. For each iteration, we store A_0 in the set of evaluated attacker's best responses $\mathcal{A}_{\mathrm{E}}$ and store all solutions $A_0, ..., A_j$ in the history of all observed attacks $\mathcal{A}_H$. For any attack A_h in the history, let D_h be the defense considered that produced A_h, and let $C(A_h, D_h)$ represent the corresponding cost observed. Given $\mathcal{A}_H$, we introduce an additional constraint that any new feasible defense must satisfy,

$$|D \cap A| \geq 1 \ \forall A_h \in \mathcal{A}_H^u. \tag{5}$$

Where $\mathcal{A}_H^u = \{A_h \mid A_h \in \mathcal{A}_H, C(A_h, D_h) \geq u\}$. This constraint represents an important set of suboptimal solutions that must be intersected by the defender.

Algorithm 1 returns a globally optimal solution if run to completion, as we prove in Theorem 1. If terminated early, the best defense D^* and best response A^* provide a lower bound u. Since the modified BRI-SS algorithm ensures that, at each iteration, the attacker's best response A_0 is distinct from $\mathcal{A}_{\mathrm{E}}$, the algorithm terminates in a finite number of iterations. The proof follows from [5].

Algorithm 1 Best Response Intersection with Suboptimal Solutions (BRI-SS)

Require: Feasible set $\mathcal{D}_{\mathrm{F}}$, cost function $C(\cdot)$, attacker's best response oracle $\mathrm{ABR}(\cdot)$, max number of suboptimal solutions collected J

Ensure: Optimal min-max defense D^*, min-max cost u

1: Initialize best defense $D^* \leftarrow \emptyset$, best response $A^* \leftarrow \emptyset$, best cost $u \leftarrow \infty$, evaluated defenses $\mathcal{D}_{\mathrm{E}} \leftarrow \emptyset$, evaluated attacker's best responses $\mathcal{A}_{\mathrm{E}} \leftarrow \emptyset$, history of all attack solutions $\mathcal{A}_H \leftarrow \emptyset$

2: **for** $D \in \mathcal{D}_{\mathrm{F}}$ such that $C(A^*, D) < u,\ |D \setminus D'| \geq 1\ \forall D' \in \mathcal{D}_{\mathrm{E}},\ |D \cap A| \geq 1\ \forall A \in \mathcal{A}_{\mathrm{E}}, |D \cap A_h| \geq 1\ \forall A_h \in \mathcal{A}_H^u$ **do**

3: Solve $\mathrm{ABR}(D)$ such that $A \cap D = \emptyset$ to generate the optimal attack A_0 and suboptimal solutions $A_1, ..., A_j$

4: **if** $C(A_0, D) < u$ **then**

5: Update best defense $D^* \leftarrow D$, best response $A^* \leftarrow A_0$, best cost $u \leftarrow C(A_0, D)$

6: **end if**

7: Update evaluated defenses $\mathcal{D}_{\mathrm{E}} \leftarrow \mathcal{D}_{\mathrm{E}} \cup \{D\}$, evaluated attacks $\mathcal{A}_{\mathrm{E}} \leftarrow \mathcal{A}_{\mathrm{E}} \cup \{A_0\}$, history of attacks $\mathcal{A}_H \leftarrow \mathcal{A}_H \cup \left(\bigcup_{i=0}^{j} \{A_i\}\right)$

8: **end for**

Theorem 1. *The BRI-SS algorithm returns an optimal solution to any interdiction defense problem* $\mathrm{ID}(\mathcal{A}_{\mathrm{F}}, \mathcal{D}_{\mathrm{F}}, C)$ *that satisfies Condition 1 and Condition 2.*

Proof: Suppose that the BRI-SS algorithm terminates with an optimal solution $c^* = C(A^*, D^*)$. For the sake of contradiction, suppose c^* is not the optimal cost. Then there must be some feasible defense $\widetilde{D}$ and attacker's best response $\widetilde{A} := ABR(\widetilde{D})$ such that $\widetilde{c} = C(\widetilde{A}, \widetilde{D}) < c^*$. The algorithm evaluates all defenses that satisfy the conditions in line 2 of Algorithm 1, so $\widetilde{D}$ must fail to satisfy constraints (3), (4), or (5). If $\widetilde{D}$ does not satisfy constraints (3) or (4), a contradiction follows the proof given in [5].

Lastly, consider the case that $\widetilde{D}$ does not satisfy constraint (5). In this case, there exists some $A_h \in \mathcal{A}_H$ where $C(A_h, D_h) \geq u$ and $|\widetilde{D} \cap A_h| = 0$. Let D_h be the defense that generated A_h in line 3 of Algorithm 1 such that A_h was a suboptimal solution. Since c^* is the lowest cost observed during the entire BRI-SS algorithm execution,

$$c^* \leq u \leq C(A_h, D_h). \tag{6}$$

By Condition 2 and line 3 of the algorithm, $|D_h \cap A_h| = 0$. Likewise, $|\widetilde{D} \cap A_h| = 0$ by supposition. Since both defenses are disjoint with A_h, by Condition 1,

$$C(A_h, D_h) = C(A_h, \widetilde{D}). \tag{7}$$

By the definition of the ABR problem,

$$C(A_h, \widetilde{D}) \leq C(\widetilde{A}, \widetilde{D}) = \widetilde{c}. \tag{8}$$

Combining (6), (7), and (8) altogether we have,

$$c^* \leq C(A_h, D_h) = C(A_h, \widetilde{D}) \leq C(\widetilde{A}, \widetilde{D}) = \widetilde{c}. \tag{9}$$

Since we assumed $c^* > \widetilde{c}$, this leads to a contradiction. □

2.3. A Suboptimal Solution Heuristic for Selecting New Defenses

Algorithm 1 iterates until there is no improving, feasible defense. In practice however, choosing which defense to examine next can greatly affect the number of iterations the algorithm takes to converge. We introduce the suboptimal-solutions new-defense MIP (SS-NDMIP) in equations (10a) - (10i) to identify the next feasible defense, $D \in \mathcal{D}_{\mathrm{F}}$, for line 2 of Algorithm 1. The SS-NDMIP seeks to maximally intersect attacks from $\mathcal{A}_H$, prioritized by cost, while maintaining feasibility. We introduce the variable $b_h \in \{0,1\}^{|H|}$, where $b_h = 1$ if defense D intersects with attack $A_h \in \mathcal{A}_H$, and $b_h = 0$ otherwise.

$$
\begin{aligned}
\text{(SS-NDMIP)} \quad & \max_{b_h, D} \sum_{h \in H} b_h && && \text{(10a)}\\
& |D \cap A_h| \geq b_h, && \forall h \in H, && \text{(10b)}\\
& b_h \geq b_{h+1}, && \forall h \in H, && \text{(10c)}\\
& C(A^*, D) \leq u, && && \text{(10d)}\\
& |D \backslash D'| \geq 1, && \forall D' \in \mathcal{D}_E, && \text{(10e)}\\
& |D \cap A| \geq 1, && \forall A \in \mathcal{A}_E, && \text{(10f)}\\
& |D \cap A_h| \geq 1, && \forall A_h \in \mathcal{A}_H^u, && \text{(10g)}\\
& b_h \in \{0,1\}, && \forall h \in H, && \text{(10h)}\\
& D \in \mathcal{D}_f. && && \text{(10i)}
\end{aligned}
$$

Intuitively, selecting a new defense that protects against as many substantive attacks from $\mathcal{A}_H$, even if they are suboptimal, results in more information for the defender. Therefore the objective (10a) is to maximize the number of attacks defended from the history of all attacks. Constraint (10b) asserts that $b_h = 1$ only if the defense intersects attack A_h. Since the attack history is sorted by cost, constraint (10c) ensures the defense prioritizes more costly attacks before less costly ones. Constraints (10d) - (10g) are the conditions from line 2 of Algorithm 1.

3. Computational Studies on Power Grid Interdiction

3.1. Experimental Setup

We present computational results of the BRI-SS algorithm outlined in Algorithm 1 using the SS-NDMIP detailed in equations (10a) - (10i) to generate new defenses compared to the BRI algorithm with the attacker-chasing objective detailed in [5]. Note we initialize both algorithms with $D = \emptyset$ for the first iteration. We use the parameters (attacker budget) k and (defender budget) h to define the feasible budgeted sets as $\mathcal{A}_{\mathrm{F}} := \{A : A \in 2^{|T|},\ |A| \leq k\}$ and $\mathcal{D}_{\mathrm{F}} := \{D : D \in 2^{|T|},\ |D| \leq h\}$. Since computational studies show the BRI algorithm can perform an order of magnitude faster than comparable CCG algorithms, it is sufficient to compare the BRI-SS algorithm to the BRI algorithm [5]. We solved the ID problem with application to power grid interdiction. We implemented the direct-current optimal power flow (DC-OPF) formulation with load shedding for the cost function and the corresponding attacker-defender formulation with branch interdiction (where the attacker may only target edges) from [7]. For big-M values in the ABR implementation, we used $M_{\mu l} = 20$ and $M_{\eta l} = 4$. The ID problem was solved for attacker and defender budgets $k = h \in \{0, 2, 4, 6, 8, 10\}$ for three test systems: the Reliability Test System Grid Modernization Lab Consortium (RTS-GMLC) [8], and the

Table 1: Gurobi solver settings for the ABR and SS-NDMIP subroutines.

Setting	Presolve	MipFocus	Heuristics	MIPGAP	Cuts	PoolSolutions	PoolSearchMode
ABR	2	3	0.5	-	3	50	0
SS-NDMIP	-	1	0.5	0.9	-	-	-

IEEE 118-bus and IEEE 300-bus cases from the PGLib v23 repository [9]. All instances were run for 10 different random seed numbers, unless otherwise stated, to obtain the average runtime and standard deviation.

Algorithms were implemented in Python 3.9.12 with Pyomo 6.8.2 [10] and solved with Gurobi 12.0.0 [11]. Gurobi settings to solve the ABR problem and SS-NDMIP are shown in Table 1. Note the MIPGAP may be set to 1, but since SS-NDMIP is a heuristic, a feasible solution is sufficient. Computational results presented were run on a compute node with the Intel(R) Xeon(R) Gold 6140 CPU, 2.30GHz processor, 36 cores, and 187 GB of memory. For all experiments, we validated that the BRI and BRI-SS algorithms' objective values were the same within solver tolerances of 10^{-3}. All test cases were solved to optimality.

3.2. Experimental Results

The efficiency of the BRI-SS algorithm compared to the BRI algorithm is shown in Tables 2-4. For both algorithms, the runtime increases rapidly with k and h. However, the BRI-SS algorithm demonstrates improved scalability compared to the BRI algorithm as k and h increase for all test cases. Although the BRI-SS algorithm tends to require more computation time per iteration, this additional overhead is offset by the fact that it converges in fewer iterations. For smaller budgets $k = h = 2$, the BRI-SS algorithm is on average .57 to .70 times slower compared to the BRI algorithm. This is because both algorithms converge in relatively few iterations, but the SS-NDMIP incurs a longer overhead per iteration. The additional computation cost is warranted for $k = h = 4$ and larger values though, as the BRI-SS algorithm begins to outperform the BRI algorithm by 1.07 to 1.95 times on average for all test cases. For the largest budgets of $k = h = 10$, the BRI-SS algorithm reduces runtime by a factor of 3.3 to 4.31 times compared to the BRI algorithm for all test cases. We believe that the improved scalability is due to the additional information from suboptimal solutions incorporated in the SS-NDMIP formulation, resulting in fewer iteration to converge.

We observe that the BRI and BRI-SS algorithms exhibit increasing runtime variance as k and h grow. This behavior is likely due to the non-deterministic nature of the Gurobi solver when solving MIPs. We conclude that the observed standard deviations are not large enough to affect the comparison of mean runtimes.

4. Future Research

In the future, we believe a machine learning approach such as in [12] could be used to build the attacker history in the BRI-SS algorithm. This machine learning approach could quickly find disruptive attacks, without sacrificing the theoretical guarantees provided by the BRI framework. Further the efficiency of the BRI-SS algorithm could be improved with heuristics to select the initial defense evaluated or with methods to improve the SS-NDMIP average runtime per iteration.

Table 2: Mean runtime and number of iterations to converge (± standard deviation) for the RTS-GMLC test case.

k , h	BRI			BRI-SS			BRI/BRI-SS	
	Time (s)	Iterations	Time/Iter	Time (s)	Iterations	Time/Iter	Ratio Time	Ratio Iter
0,0	0.11 ± 0.09	1.00 ± 0.00	0.11	0.12 ± 0.00	1.00 ± 0.00	0.12	0.92	1.00
2,2	7.48 ± 0.21	3.00 ± 0.00	2.49	13.08 ± 0.35	4.00 ± 0.00	3.27	0.57	0.75
4,4	67.22 ± 9.33	10.50 ± 1.02	6.40	62.54 ± 12.17	8.40 ± 1.43	7.45	1.07	1.25
6,6	367.60 ± 34.54	31.70 ± 2.45	11.60	236.02 ± 29.90	19.30 ± 2.49	12.23	1.56	1.64
8,8	2148.80 ± 227.63	115.10 ± 10.24	18.67	962.05 ± 60.01	47.90 ± 2.34	20.08	2.23	2.40
10,10	10812.30 ± 704.67	413.40 ± 23.98	26.15	3271.55 ± 317.58	114.50 ± 9.40	28.57	3.30	3.61

Table 3: Mean runtime and number of iterations to converge (± standard deviation) for the IEEE 118-bus test case. Results marked with the † were only run for one seed due to time limits.

k , h	BRI			BRI-SS			BRI/BRI-SS	
	Time (s)	Iterations	Time/Iter	Time (s)	Iterations	Time/Iter	Ratio Time	Ratio Iter
0,0	0.08 ± 0.02	1.00 ± 0.00	0.08	0.11 ± 0.00	1.00 ± 0.00	0.11	0.73	1.00
2,2	11.46 ± 3.10	4.10 ± 0.83	2.79	16.36 ± 3.71	4.50 ± 0.81	3.63	0.70	0.91
4,4	214.66 ± 17.60	28.10 ± 2.17	7.64	109.80 ± 15.07	13.30 ± 1.49	8.26	1.95	2.11
6,6	2619.61 ± 254.15	95.40 ± 10.86	27.46	904.71 ± 125.44	27.70 ± 3.03	32.66	2.90	3.44
8,8	33663.93 ± 2489.86	469.90 ± 42.64	71.64	7227.19 ± 606.93	67.30 ± 5.80	107.39	4.66	6.98
10,10	337568.44 †	1361.00 †	248.03	78290.91 †	147.00 †	532.59	4.31	9.26

Table 4: Mean runtime and number of iterations to converge (± standard deviation) for the IEEE 300-bus test case. Results marked with † were only run for one seed due to time limits.

k , h	BRI			BRI-SS			BRI/BRI-SS	
	Time (s)	Iterations	Time/Iter	Time (s)	Iterations	Time/Iter	Ratio Time	Ratio Iter
0	0.15 ± 0.00	1.00 ± 0.00	0.15	0.24 ± 0.03	1.00 ± 0.00	0.24	0.62	1.00
2	28.52 ± 2.20	4.00 ± 0.00	7.13	40.96 ± 7.85	4.70 ± 0.64	8.71	0.70	0.85
4	499.50 ± 23.02	27.20 ± 1.47	18.36	301.08 ± 45.92	15.20 ± 2.18	19.81	1.66	1.79
6	3632.81 ± 226.00	86.30 ± 5.44	42.10	1918.16 ± 144.19	42.90 ± 2.91	44.71	1.89	2.01
8	24697.80 ± 1665.33	287.20 ± 19.43	86.00	9609.06 ± 954.49	98.17 ± 9.26	97.89	2.57	2.93
10,10	330664.87 †	1405.00 †	235.35	92616.19 †	288.00 †	321.58	3.57	4.88

References

[1] L. Lozano and J. C. Smith, "A Backward Sampling Framework for Interdiction Problems with Fortification," INFORMS Journal on Computing, vol. 29, no. 1, pp. 123–139, Jan. 2017, publisher: INFORMS. [Online]. Available: https://pubsonline.informs.org/doi/10.1287/ijoc.2016.0721

[2] G. G. Brown, W. M. Carlyle, J. Salmerón, and K. Wood, "Analyzing the Vulnerability of Critical Infrastructure to Attack and Planning Defenses," in Emerging Theory, Methods, and Applications, ser. INFORMS TutORials in Operations Research. INFORMS, Sep. 2005, pp. 102–123, section: 4. [Online]. Available: https://pubsonline.informs.org/doi/abs/10.1287/educ.1053.0018

[3] J. C. Smith and Y. Song, "A Survey of Network Interdiction Models and Algorithms," European Journal of Operational Research,

vol. 283, no. 3, pp. 797–811, Jun. 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0377221719305156

[4] S. Gualandi and F. Malucelli, "Constraint Programming-based Column Generation," Annals of Operations Research, vol. 204, no. 1, pp. 11–32, Apr. 2013. [Online]. Available: https://doi.org/10.1007/s10479-012-1299-7

[5] A. Mastin, A. Baxter, A. Musselman, and J.-P. Watson, "Best Response Intersection: An Optimal Algorithm for Interdiction Defense," INFORMS Journal on Optimization, vol. 5, no. 2, pp. 172–190, Apr. 2023, publisher: INFORMS. [Online]. Available: https://pubsonline.informs.org/doi/abs/10.1287/ijoo.2022.0081

[6] M. Y. Tsang, K. S. Shehadeh, and F. E. Curtis, "An Inexact Column-and-Constraint Generation Method to Solve Two-Stage Robust Optimization Problems," Operations Research Letters, vol. 51, no. 1, pp. 92–98, Jan. 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167637722001742

[7] A. Mastin, W. Hofgard, and J.-P. Watson, "A Formulation for Power Flow Interdiction with Bounded Big-M Values," Mathematical Programming, vol. Submitted, 2025.

[8] C. Barrows, A. Bloom, A. Ehlen, J. Ikäheimo, J. Jorgenson, D. Krishnamurthy, J. Lau, B. McBennett, M. O'Connell, E. Preston, A. Staid, G. Stephen, and J.-P. Watson, "The IEEE Reliability Test System: A Proposed 2019 Update," IEEE Transactions on Power Systems, vol. 35, no. 1, pp. 119–127, Jan. 2020. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/8753693

[9] S. Babaeinejadsarookolaee, A. Birchfield, R. D. Christie, C. Coffrin, C. DeMarco, M. Ferris, S. Greene, R. Huang, C. Josz, R. Korab, B. Lesieutre, J. Maeght, T. W. K. Mak, D. K. Molzahn, T. J. Overbye, P. Panciatici, B. Park, J. Snodgrass, A. Tbaileh, P. V. Hentenryck, and R. Zimmerman, "The Power Grid Library for Benchmarking AC Optimal Power Flow Algorithms," Jan. 2021, arXiv:1908.02788. [Online]. Available: http://arxiv.org/abs/1908.02788

[10] M. L. Bynum, G. A. Hackebeil, W. E. Hart, C. D. Laird, B. L. Nicholson, J. D. Siirola, J.-P. Watson, and D. L. Woodruff, Pyomo — Optimization Modeling in Python, ser. Springer Optimization and Its Applications. Springer International Publishing, 2021, vol. 67. [Online]. Available: http://link.springer.com/10.1007/978-3-030-68928-5

[11] Gurobi Optimization, LLC, "Gurobi Optimizer Reference Manual," 2025. [Online]. Available: https://www.gurobi.com

[12] N. Orkun Baycik, "Machine Learning Based Approaches to Solve the Maximum Flow Network Interdiction Problem," Computers & Industrial Engineering, vol. 167, p. 107873, May 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0360835221007774