Optimally Delaying Attacker Projects Under Resource Constraints

Jim Luedtke

Ashley Peper, Laura Albert

Industrial and Systems Engineering

University of Wisconsin-Madison

Mixed-Integer Programming Workshop, June 6, 2025

This work was in part funded by the National Science Foundation Award 2000986.

A Different Photo of Ashley

A Different Photo of Ashley

Outline

Bilevel/Interdiction game terminology

- Leader $\Leftrightarrow$ Defender
- Follower $\Leftrightarrow$ Attacker(s)

Plan

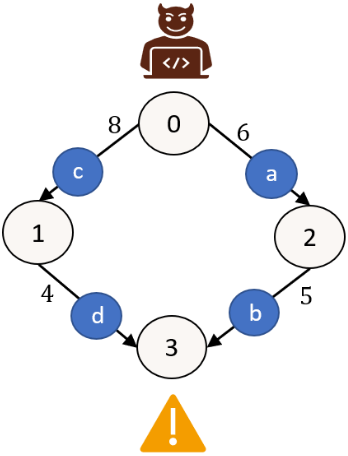
- Static model for delaying an attacker project of Brown et al. (2005)
- Extension considering more general defender actions of Zheng and Albert (2019)
- Model that considers defender resource constraints, but ignores attacker project structure Peper et al. (2024)
- New model that brings it all together
- Relaxation, reformulation, and heuristics
- Computational study: What is benefit of new model?

How to Deploy Mitigations to Delay Attacks?

Model of Brown et al. (2005)

Attacker: Minimize time to complete a project

- Working to achieve a goal (e.g., breach a cybersystem)
- Tasks required modeled in a *project network*
    - N: Set of intermediate goals
    - P: Set of tasks (i, j). Goal j achieved only when all (i, j) tasks done
    - t_{ij}: Duration of task (i, j)
    - Minimum project completion time $\Leftrightarrow$ Longest path in network

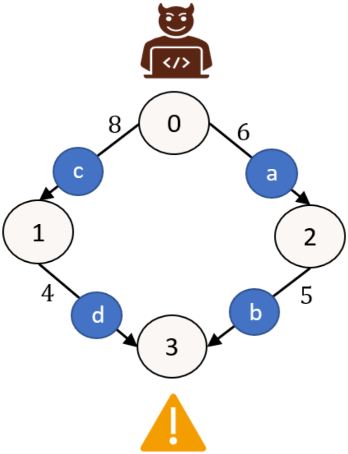


How to Deploy Mitigations to Delay Attacks?

Model of Brown et al. (2005)

Defender: Maximize attacker's project completion time

- *Before* attacker acts, can "interdict" individual task arcs $(i, j) \Rightarrow$ Delays by d_{ij}
- Interdicting arc (i, j) costs c_{ij}: Total budget B



Extension: Zheng and Albert (2019)

Defender chooses *mitigations* to implement: $m \in M$

- For each task $(i, j) \in P$, $M_{ij} \subseteq M$ is set of mitigations that "cover" task (i, j)

Defender decisions:

- x_m: Binary to indicate if select mitigation $m \in M$
- z_{ij}: Binary to indicate if task (i, j) is covered by a selected mitigation

Constraints:

$$
\begin{aligned}
&\sum_{m \in M} b_m x_m \leq B, & \\
&z_{ij} \leq \sum_{m \in M_{ij}} x_m, & \forall (i, j) \in P \\
&x_m \in \{0, 1\}, & \forall m \in M \\
&z_{ij} \in \{0, 1\}, & \forall (i, j) \in P
\end{aligned}
$$

Extension: Zheng and Albert (2019), cont'd

Multiple attackers (or attack projects): $a \in A$

- Each has its own task set $P_a \subseteq P$ and goal set $N_a \subseteq N$ and duration and delay amounts
- Weight p_a indicates importance of attacker a

Extension: Zheng and Albert (2019), cont'd

Multiple attackers (or attack projects): $a \in A$

- Each has its own task set $P_a \subseteq P$ and goal set $N_a \subseteq N$ and duration and delay amounts
- Weight p_a indicates importance of attacker a

Minimum project completion time of attacker a, given defender actions $\mathbf{z}$:

$$\begin{aligned} s_a(\mathbf{z}) = \min\ & h_{\text{end}} \\ \text{s.t.}\ & h_j - h_i \geq t_{ija} + d_{ija} z_{ij}, \quad \forall (i,j) \in P_a \\ & h_{\text{start}} = 0, \\ & h_i \geq 0, \quad \forall i \in N_a \end{aligned}$$

Defender objective:

$$\max \sum_{a \in A} p_a s_a(\mathbf{z})$$

MILP formulation obtained by taking dual of attacker problem and linearizing objective

Timing is Everything

This model assumes

- Defender implements all selected mitigations
- Then attacker(s) carry out their project(s)

But all these activities take time

- Attacker carrying out steps of their project
- Defender implementing mitigations

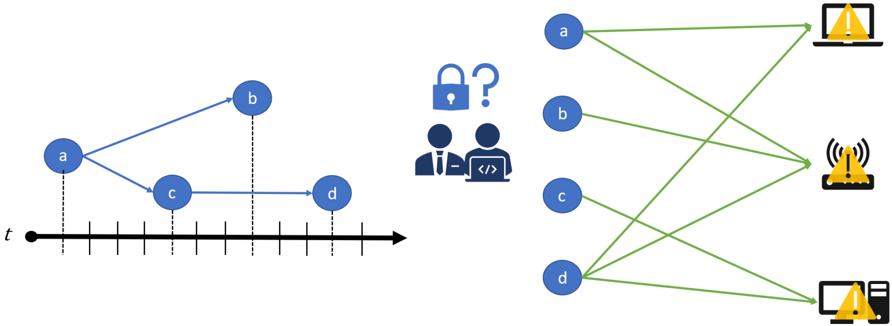
If a mitigation that covers an attacker task isn't completed before an attacker starts it, it's too late!

- How to model the timing/scheduling?

Scheduling of Mitigation Deployment

Scheduling mitigation deployment to cover vulnerabilities: Peper et al. (2024)

- Defender schedules mitigations over T time periods
- Mitigations take time and resources to implement
- Each mitigation can cover multiple vulnerability nodes
- Each node can be covered multiple times, with diminishing returns
- Defender maximizes time-weighted coverage of nodes



Scheduling Mitigations (Peper et al., 2024)

Model extends a Resource Constrained Project Scheduling Problem (RCPSP)

- Well studied problem: Pritsker et al. (1969), Yang et al. (1993), Vanhoucke et al. (2001)
- Binary variables $x_{mt} = 1$ if job m is completed in period t
- Constraints for resources and precedences

Extension

- Adds variables and constraints to capture coverage of nodes with an objective that accounts for diminishing returns for mulitple coverage.

We use a similar model for the defender

- Vulnerability nodes $\rightarrow$ attacker actions
- Maximize coverage $\rightarrow$ maximize attacker project completion times

Modeling Considerations

Bilevel Problem

- Defender's problem:
  - Defender schedules mitigations using an RCPSP-based model
  - Objective to maximize weighted average of attacker project completion times
- Attacker's problem:
  - Complete all activities as fast as possible
  - This is limited by the longest path in the graph

Modeling Considerations

Bilevel Problem

- Defender's problem:
  - Defender schedules mitigations using an RCPSP-based model
  - Objective to maximize weighted average of attacker project completion times
- Attacker's problem:
  - Complete all activities as fast as possible
  - This is limited by the longest path in the graph

Modeling Challenges

- Attacker's graph potentially changes each time period based on defender decisions
- Mitigations delaying arcs that have already been completed by the attacker have no effect

Multi-period Sequential Game?

Do we need to consider sequence of Defender-Attacker-Defender... moves?

Multi-period Sequential Game?

Do we need to consider sequence of Defender-Attacker-Defender... moves?

Fortunately not!

- Attacker model is just completing a project
- Always optimal to begin tasks as soon as possible
- Defender decisions just influence how long the tasks take

⇒ Can still model as single Defender-Attacker sequence

Multi-period Sequential Game?

Do we need to consider sequence of Defender-Attacker-Defender... moves?

Fortunately not!

- Attacker model is just completing a project
- Always optimal to begin tasks as soon as possible
- Defender decisions just influence how long the tasks take

⇒ Can still model as single Defender-Attacker sequence

Limitation

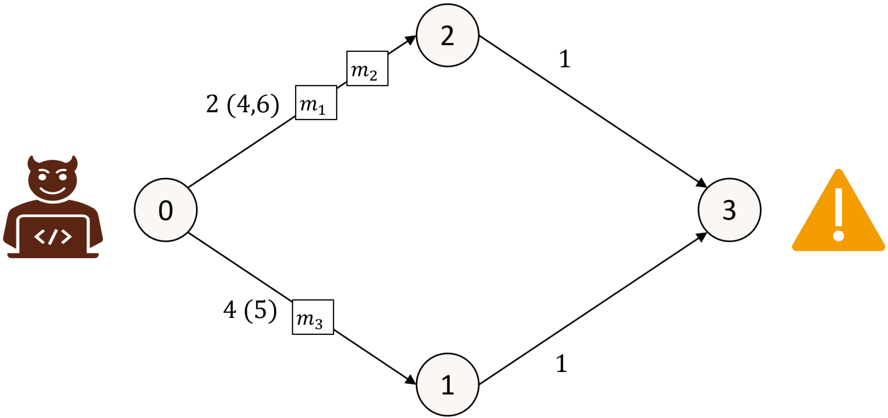
- Would not be true if attacker had nontrivial decisions, e.g., due to limited resources or ability to expedite a task

A Time-indexed Formulation

To address the time variable nature of the attacker network, we use a time-expanded network with arcs defined for all possible task durations

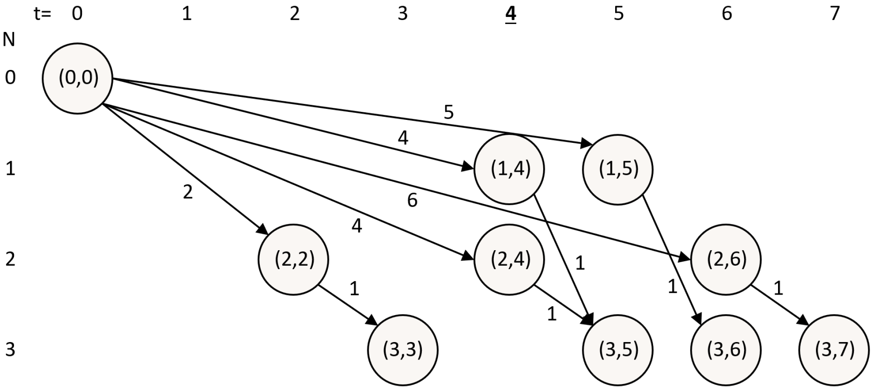
A Time-indexed Formulation

To address the time variable nature of the attacker network, we use a time-expanded network with arcs defined for all possible task durations



A Time-indexed Formulation

To address the time variable nature of the attacker network, we use a time-expanded network with arcs defined for all possible task durations



Includes nodes of the form (i, t), where arc $((i, t), (j, s))$ has length $s - t$.

Defender's Problem

- RCPSP job scheduling: $x_{mt} = 1$ if job m completed in period t

Defender's Problem

- RCPSP job scheduling: $x_{mt} = 1$ if job m completed in period t
- Variables z_{ijt} give duration of arc (i, j) as of period t:

$$z_{ijt} \leq \sum_{m \in M} \delta_{ijm} x_{mt} + z_{ij,t-1} \qquad \text{(job completion adds delay)}$$

$$z_{ijt} \leq d_{ij} + \bar{\delta}_{ij} \qquad \text{(max arc duration)}$$

$$z_{ij0} = d_{ij} \qquad \text{(initial arc duration)}$$

Defender's Problem

- RCPSP job scheduling: $x_{mt} = 1$ if job m completed in period t
- Variables z_{ijt} give duration of arc (i, j) as of period t:

$$z_{ijt} \leq \sum_{m \in M} \delta_{ijm} x_{mt} + z_{ij,t-1} \qquad \text{(job completion adds delay)}$$

$$z_{ijt} \leq d_{ij} + \bar{\delta}_{ij} \qquad \text{(max arc duration)}$$

$$z_{ij0} = d_{ij} \qquad \text{(initial arc duration)}$$

- Binary variables ρ_{ijts} indicate if time indexed arc $((i,t),(j,s))$ *is possible for the attacker* given arc duration z_{ijt}
  - $\sum_{s \geq t + d_{ij}} (s - t) \rho_{ijts} \leq z_{ijt}, \quad \sum_s \rho_{ijts} = 1$

Defender's Problem

- RCPSP job scheduling: $x_{mt} = 1$ if job m completed in period t
- Variables z_{ijt} give duration of arc (i, j) as of period t:

$$z_{ijt} \leq \sum_{m \in M} \delta_{ijm} x_{mt} + z_{ij,t-1} \qquad \text{(job completion adds delay)}$$

$$z_{ijt} \leq d_{ij} + \bar{\delta}_{ij} \qquad \text{(max arc duration)}$$

$$z_{ij0} = d_{ij} \qquad \text{(initial arc duration)}$$

- Binary variables ρ_{ijts} indicate if time indexed arc $((i, t), (j, s))$ *is possible for the attacker* given arc duration z_{ijt}
  - $\sum_{s \geq t + d_{ij}} (s - t) \rho_{ijts} \leq z_{ijt}, \quad \sum_s \rho_{ijts} = 1$
- Maximize $\sum_{a \in A} p_a Y^a(\rho)$,

  where $Y^a(\rho)$ is optimal value of attacker a problem

Attacker's Problem

Dual of attacker a problem is a longest path problem:

- Flow variables: $y^a_{ijts} = 1$ if attacker uses time-indexed arc $((i,t),(j,s))$.
- Flow balance constraints

Attacker's Problem

Dual of attacker a problem is a longest path problem:

- Flow variables: $y^a_{ijts} = 1$ if attacker uses time-indexed arc $((i, t), (j, s))$.
- Flow balance constraints
- Project network is a directed acyclic graph $\implies$
  - Can model as an LP

Attacker's Problem

Dual of attacker a problem is a longest path problem:

- Flow variables: $y^a_{ijts} = 1$ if attacker uses time-indexed arc $((i,t),(j,s))$.
- Flow balance constraints
- Project network is a directed acyclic graph $\Longrightarrow$
  - Can model as an LP
- $y^a_{ijts} \leq \rho_{ijts}$ (only use edge with duration determined by defender)

Attacker's Problem

Dual of attacker a problem is a longest path problem:

- Flow variables: $y^a_{ijts} = 1$ if attacker uses time-indexed arc $((i,t),(j,s))$.
- Flow balance constraints
- Project network is a directed acyclic graph $\implies$
  - Can model as an LP
- $y^a_{ijts} \leq \rho_{ijts}$ (only use edge with duration determined by defender)
- Maximize length: $\displaystyle\sum_{((i,t),(j,s))\in\mathcal{E}} (s-t)y^a_{ijts}$

Combined Model

- Since both attacker and defender problems are maximizations, we can combine them into one model

Combined Model

- Since both attacker and defender problems are maximizations, we can combine them into one model

$$\text{Maximize} \quad \sum_{a \in A} p_a \sum_{((i,t),(j,s)) \in \mathcal{E}} (s - t) y^a_{ijts}$$

Subject to:

RCPSP constraints on x	(Defender decisions)
Constraints to set z & ρ	(Calculate connecting variables)
$y^a \leq \rho$ constraints	(Use connecting variables)
Flow balance constraints	(Attacker decisions)
Binary x, ρ; $y, z \geq 0$	

Baseline Heuristic 1: Ignore Attacker's Problem

RCPSP: Solve defender's problem as an RCPSP with simplified objective.

- Can pass solution to attackers' problems to evaluate true objective.

Baseline Heuristic 1: Ignore Attacker's Problem

RCPSP: Solve defender's problem as an RCPSP with simplified objective.

- Can pass solution to attackers' problems to evaluate true objective.

Option 1: Simple time-weighted objective based on job completion ($\alpha \in (0, 1]$)

$$\text{Maximize} \quad \sum_{t=1}^{T} \alpha^t \sum_{m \in M} \sum_{a \in A} \sum_{(i,j) \in \mathcal{A}_a} p_a \delta_{ijm} x_{mt}$$

$$\text{Subject to:} \quad \text{RCPSP Constraints on } x$$

Baseline Heuristic 1: Ignore Attacker's Problem

RCPSP: Solve defender's problem as an RCPSP with simplified objective.

- Can pass solution to attackers' problems to evaluate true objective.

Option 2: Edges provide time-weighting based on possible completion times

- For each attacker a, each node i has an earliest and latest reachable time, $\underline{t}_i^a$ and $\bar{t}_i^a$
- Found by solving a longest path problem to i given no mitigations or all mitigations implemented

Baseline Heuristic 1: Ignore Attacker's Problem

RCPSP: Solve defender's problem as an RCPSP with simplified objective.

- Can pass solution to attackers' problems to evaluate true objective.

Option 2: Edges provide time-weighting based on possible completion times

- For each attacker a, each node i has an earliest and latest reachable time, $\underline{t}_i^a$ and $\bar{t}_i^a$
- Found by solving a longest path problem to i given no mitigations or all mitigations implemented

$$\text{Maximize} \sum_{t=1}^{T} \sum_{m \in M} \sum_{a \in A} \sum_{(i,j) \in \mathcal{A}_a} p_a w_{ijmt}^a x_{mt}$$

where

$$w_{ijmt}^a = \begin{cases} \delta_{ijm} & \text{if } t < \underline{t}_i^a \\ \alpha^{t-\underline{t}_i} \delta_{ijm} & \text{if } \underline{t}_i^a \le t \le \bar{t}_i^a \\ 0 & \text{if } t > \bar{t}_i^a \end{cases}$$

Relaxation: Ignore Simultaneous Scheduling

- Non-scheduling models implicitly assume the defender completes all interdictions before attacker starts
- We can make this assumption to obtain a **relaxation**
- Can also evaluate the resulting defender solution in attacker problems to get true objective $\Rightarrow$ Baseline heuristic 2

Relaxation: Ignore Simultaneous Scheduling

- Non-scheduling models implicitly assume the defender completes all interdictions before attacker starts
- We can make this assumption to obtain a **relaxation**
- Can also evaluate the resulting defender solution in attacker problems to get true objective $\Rightarrow$ Baseline heuristic 2

Modeling Notes

- Arc lengths don't depend on time started
  $\implies$ Time-indexed attacker network isn't needed
- Arc lengths still depend on defender decisions
  $\implies$ Index each arc variable by set of possible arc lengths $\ell \in \mathcal{L}^{ij}$:

$$\rho_{ij\ell}, \quad y^a_{ij\ell}$$

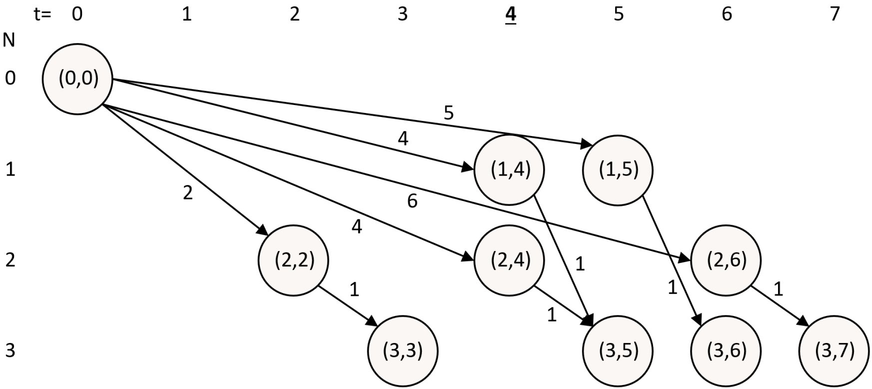
- This is comparable to existing formulations, with the extension of more than one possible delayed arc value.
- Model defender decisions with RCPSP, but only use z_{ijT} to determine arc lengths.

Reformulating the Original Model

- Decrease the size of the model by only time-indexing when needed
- Motivation: Defender planning horizon may be shorter than attacker's
- Once the defender's horizon ends, no need for time-indexing of attacker model

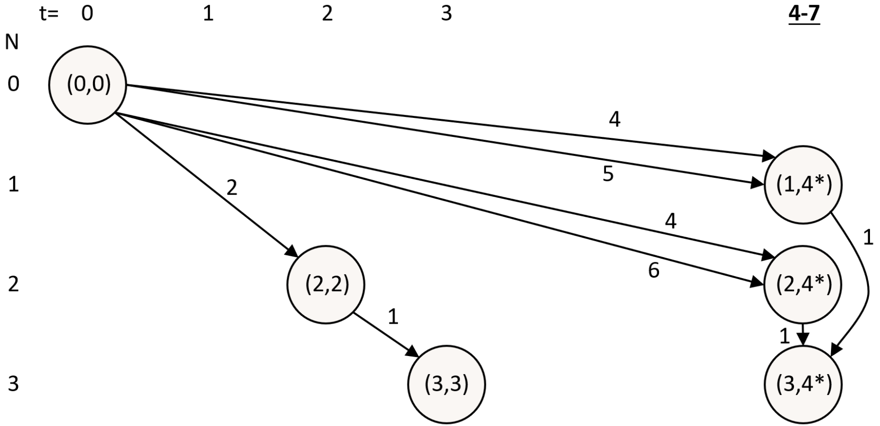
Reformulating the Original Model

- Decrease the size of the model by only time-indexing when needed
- Motivation: Defender planning horizon may be shorter than attacker's
- Once the defender's horizon ends, no need for time-indexing of attacker model



Reformulating the Original Model

- Decrease the size of the model by only time-indexing when needed
- Motivation: Defender planning horizon may be shorter than attacker's
- Once the defender's horizon ends, no need for time-indexing of attacker model



Adding in Sequential Model

- Empirical observation: Sequential LP relaxation provides better bounds than original LP relaxation
- Idea: Create a model that merges the two

Adding in Sequential Model

- Empirical observation: Sequential LP relaxation provides better bounds than original LP relaxation
- Idea: Create a model that merges the two
- Add variables/constraints for each model
  - Enforce $\tilde{y}^a_{ij\ell} = 1$ iff time-indexed $y^a_{ijts\ell} = 1$ for some time indexed arc.
  - Enforce $\tilde{\rho}^a_{ij\ell} = 1$ only if $\rho_{ijts\ell} = 1$ for some time-indexed arc.

Adding in Sequential Model

- Empirical observation: Sequential LP relaxation provides better bounds than original LP relaxation
- Idea: Create a model that merges the two
- Add variables/constraints for each model
  - Enforce $\tilde{y}^a_{ij\ell} = 1$ iff time-indexed $y^a_{ijts\ell} = 1$ for some time indexed arc.
  - Enforce $\tilde{\rho}^a_{ij\ell} = 1$ only if $\rho_{ijts\ell} = 1$ for some time-indexed arc.

Maximize	$\sum_{a \in A} p_a \sum_{((i,t),(j,s)) \in \mathcal{E}} \sum_{\ell \in \mathcal{L}^{ijst}} \ell y^a_{ijts\ell}$
Subject to:	RCPSP constraints on x
	Constraints to set z & ρ
	Constraints to set $\tilde{\rho}$ using z_{ijT}
	$y^a \leq \rho$ **and** $\tilde{y}^a \leq \tilde{\rho}^a$
	Flow balance constraints for y **and** $\tilde{y}$
	Constraints to connect $y, \tilde{y}$ **and** $\rho, \tilde{\rho}$
	Binary $x, \rho, \tilde{\rho}$; $y, \tilde{y}, z \geq 0$

Decomposition Methods?

Formulations are large!

- Benders decomposition?
- Column generation?

We (Ashley) tried a few

Decomposition Methods?

Formulations are large!

- Benders decomposition?
- Column generation?

We (Ashley) tried a few

- Conclusion: Gurobi is too good!
- There may be a scale at which decomposition pays off, but we did not find it

Computational Results

50 randomly generated test instances

- Defender RCPSP data generated following approach in Kolisch and Sprecher (1997)
- Defneder has approx 150 possible mitigations (jobs), of which $\approx$ 30 can be done due to resource consraints
- Defender time horizon: 30-50 periods
- Attackers: 4-20 goals, 10-30 tasks
- Attacker time horizon: 60-200 periods

30 minute time limit

Computational Results

Method	Avg LB Gap	Avg Final UB Gap	Avg LP UB Gap	Avg Run Time	TiLim
Opt-Orig					
Opt-Reform					
Opt+SeqRelax					

Computational Results

Method	Avg LB Gap	Avg Final UB Gap	Avg LP UB Gap	Avg Run Time	TiLim
Opt-Orig	0.1%	1.3%	18.0%	437.0	10
Opt-Reform	0.0%	0.1%	13.4%	126.1	1
Opt+SeqRelax	0.0%	0.0%	7.5%	98.0	1

The reformulations decrease run-time, likely due to the tighter LP bounds.

Computational Results

Method	Avg LB Gap	Avg Final UB Gap	Avg LP UB Gap	Avg Run Time	TiLim
Opt-Orig	0.1%	1.3%	18.0%	437.0	10
Opt-Reform	0.0%	0.1%	13.4%	126.1	1
Opt+SeqRelax	0.0%	0.0%	7.5%	98.0	1
Seq	10.4%	3.3%	8.9%	130.8	2

The sequential relaxation model provides good upper bound, but poor quality solutions, and is surprisingly not faster than the reformulated model.

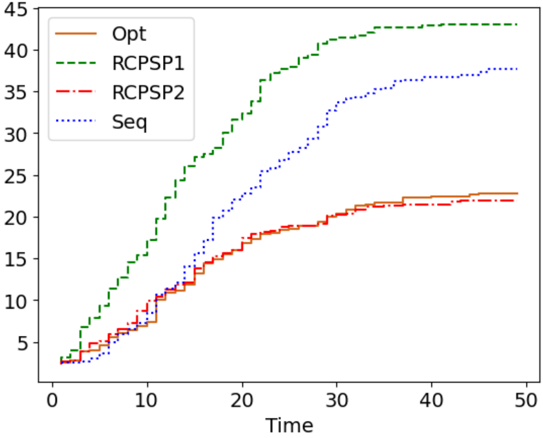
Computational Results

Method	Avg LB Gap	Avg Final UB Gap	Avg LP UB Gap	Avg Run Time	TiLim
Opt-Orig	0.1%	1.3%	18.0%	437.0	10
Opt-Reform	0.0%	0.1%	13.4%	126.1	1
Opt+SeqRelax	0.0%	0.0%	7.5%	98.0	1
Seq	10.4%	3.3%	8.9%	130.8	2
RCPSP-1	10.8%			12.8	0
RCPSP-2	6.1%			13.2	0

RCPSP approaches that ignore attacker model yield poor solutions, but solve quickly.

Where Do Heuristics Go Wrong?

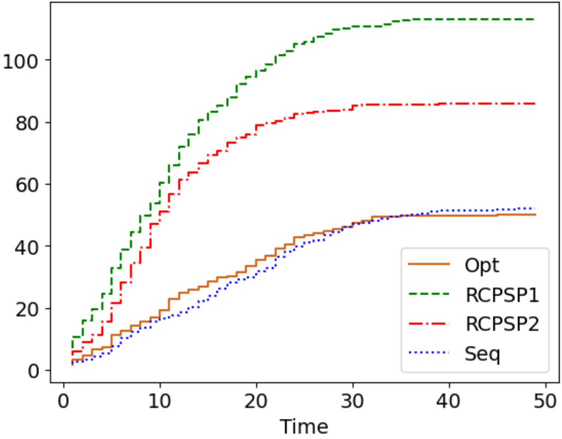
Cumulative average number of attacker arcs covered too late



RCPSP1 and Seq often cover arcs *after* the attacker has already started it

Where Do Heuristics Go Wrong?

Cumulative average number of non-critical attacker arcs covered



RCPSP1 and RCPSP2 often cover arcs that are not on the attacker critical path

Summary and Future Work

- There is benefit to considering timing of attacker and defender actions
- Formulation can be derived using time-indexed attacker network
- Reformulation reduces size $\Rightarrow$ Can solve "reasonable" size

Summary and Future Work

- There is benefit to considering timing of attacker and defender actions
- Formulation can be derived using time-indexed attacker network
- Reformulation reduces size $\Rightarrow$ Can solve "reasonable" size

Future work

- Find a decomposition method that works better?
- Attacker has nontrivial decisions (dynamic game?)
- Different attacker model (e.g., shortest path)

jim.luedtke@wisc.edu

Brown, G. G., Carlyle, W. M., Royset, J. O., and Wood, R. K. (2005). On the complexity of delaying an adversary's project. In *The Next Wave in Computing, Optimization, and Decision Technologies*, pages 3–17. Springer.

Kolisch, R. and Sprecher, A. (1997). PSPLIB - a project scheduling problem library: Or software - orsep operations research software exchange program. *European Journal of Operational Research*, 96(1):205–216.

Peper, A., Albert, L. A., and Luedtke, J. R. (2024). Selecting and scheduling cybersecurity mitigations with resource constraints.

Pritsker, A. A. B., Waiters, L. J., and Wolfe, P. M. (1969). Multiproject scheduling with limited resources: A zero-one programming approach. *Management science*, 16(1):93–108.

Vanhoucke, M., Demeulemeester, E., and Herroelen, W. (2001). On maximizing the net present value of a project under renewable resource constraints. *Management Science*, pages 1113–1121.

Yang, K. K., Talbot, F., and Patterson, J. H. (1993). Scheduling a project to maximize its net present value: An integer programming approach. *European Journal of Operational Research*, 64(2):188–198. Project Management and Scheduling.

Zheng, K. and Albert, L. A. (2019). Interdiction models for delaying adversarial attacks against critical information technology infrastructure. *Naval Research Logistics (NRL)*, 66(5):411–429.