REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*
06/22/2021

2. REPORT TYPE
Conference Briefing

3. DATES COVERED *(From - To)*

4. TITLE AND SUBTITLE
USAF Tanker Planning Optimization

5a. CONTRACT NUMBER
FA8702-21-C-0001

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)
Douglas S Altner; Drew Armstrong; Isaac A Armstrong; Abby Ng

5d. PROJECT NUMBER
03211CY0AA

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
The MITRE Corporation
7515 Colshire Drive
McLean, VA 22102

8. PERFORMING ORGANIZATION REPORT NUMBER
PRS-21-0989

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
United States Air Force (AFLCMC)

10. SPONSOR/MONITOR'S ACRONYM(S)
AFLCMC

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
DISTRIBUTION STATEMENT A. Approved for public release: distribution unlimited.

13. SUPPLEMENTARY NOTES
preprint submission to MORS (Military Operations Research Society) Symposium, 21-25 June 2021. Virtual.

14. ABSTRACT
This talk presents an in-advance planning optimization model, developed in partnership between Kessel Run and MITRE, for assigning tankers to receivers for aerial refueling while also factoring in a large list of operational constraints and considerations. Also discussed will be the additional challenges and opportunities with developing an optimization-based decision support system for USAF tanker planning as well as “man vs. machine” results highlighting the substantial potential benefits of optimization-based planning.

15. SUBJECT TERMS
Modeling and Simulation; Air Force Operations; jigsaw; tankers; optimization capability; sliding window phase

16. SECURITY CLASSIFICATION OF: a. REPORT	b. ABSTRACT	c. THIS PAGE	17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
					Susan Carpenito
				32	**19b. TELEPHONE NUMBER** *(Include area code)* 781-271-7646

USAF Tanker Planning Optimization

Doug Altner

Drew Armstrong

Isaac Armstrong

Abby Ng

June 2021

Joint work with:

MITRE

USAF Kessel Run

Image from defense.gov

Project Background

- Until 2018, tanker plans were created on a whiteboard

- Kessel Run—the AOC-WS System Program Office—changed this by rapidly fielding a tanker planning tool called **Jigsaw**

- **Jigsaw** = Application for manually creating tanker plans
  - Saves time by automatically aggregating data
  - Provides real-time feasibility and efficiency metrics, resulting in better plans

- Now MITRE and Kessel Run are prototyping an optimization engine for a tanker planning system

Project Scope

- Focus: in-advance tanker planning
  - Optimizing refueling sequences; not tactical flight paths
  - Optimizing for a single 24-hour window
- Scale: Hundreds of requests w/ tens of tankers
- Design for Human-Machine Teaming
  - User should be able to edit and override plans
  - User may want to specify feasibility thresholds and trade-offs for optimizer

Image from af.mil

Why Build an Optimization Capability?

- **<u>Save Time</u>** – Automatically draft operator-quality plans
  - Users can do other things while plans are being drafted
  - Users can review and edit plans when they are done
  - Create time for users to solve other problems

- **<u>Boost Efficiency</u> –** Consider overwhelming numbers of possible plans and identify how to do more with limited tankers
  - Fulfill more requests within the preferred parameters while burning less fuel

Optimizer Can Double as a What-If Tool

- **Investigate how various changes could impact day-to-day operations, e.g.:**
  - What if tankers are repositioned to different bases?
  - What if the volume of requests increases?
  - What are the benefits of changing track locations?
  - What if we have more or fewer tankers? Many small unmanned tankers?
  - What if we have tankers with different performance characteristics?
  - What if we could turn tankers faster, e.g., via hot refueling?
  - What if we gain/lose certain diplomatic clearances?
  - What if we change our policy for using coalition tankers?

Optimizer Can Double as a Mechanical Turk

- Automatically generate “good enough” tanker plans for the purposes of simulation exercises and wargaming

Image from afcent.af.mil

Modeling Approach

High Level Problem Statement

- **Decide:**
  - Which tankers should service which receivers

- **Objective:**
  - Primarily maximize service—number of requests that can be fulfilled within requirements
  - Secondarily minimize a weighted combination of sorties and fuel burn

Image from amc.af.mil

Basic Problem Elements

Image from amc.af.mil

- **Tankers:**
  - Several types with different capacities
  - Have an associated base at which it must depart and return
  - Burn and deliver fuel from the same reserve
  - Can be configured to service different receiver types
- **Requests:**
  - **Basic Type:** For a specific amount of fuel at a specific track at a specific time
  - **Drag Request:** Meet at track A at time $T_{1,}$ follow receiver to track B at time $T_{2,}$ deliver X lbs of fuel over course of journey
- **Tracks:**
  - Pre-defined locations at which tankers can (wait to) meet receivers
  - At most one tanker can be in a track at any time

Model Constraints

- Limited tanker fleet
- Tanker capacities
- Tanker-receiver compatibility
- Plan for track jump time
- Maximum sortie length
- Minimum on-ground turn time
- Track capacity constraints
- Must return to home station
- Plan around sorties fixed by user
- Drag requirements
- Performance-limited requests
- Repeat customer preferences

List has been refined from many rounds of feedback with operational experts

Photo from safie.hq.af.mil

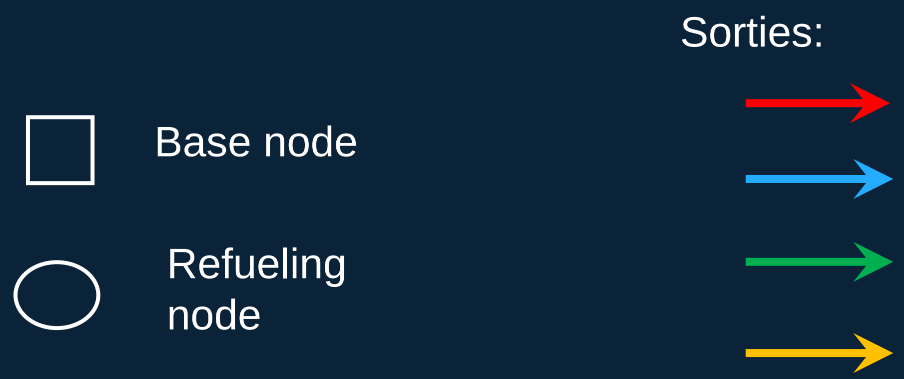
Optimization Process

- Three key steps:
  - **Step 1:** Construct a node-edge graph to define the refueling sequences the model will consider
  - **Step 2:** Use that graph to define an integer programming optimization model
  - **Step 3:** Construct a good solution according to the integer program using a two-pass decomposition heuristic

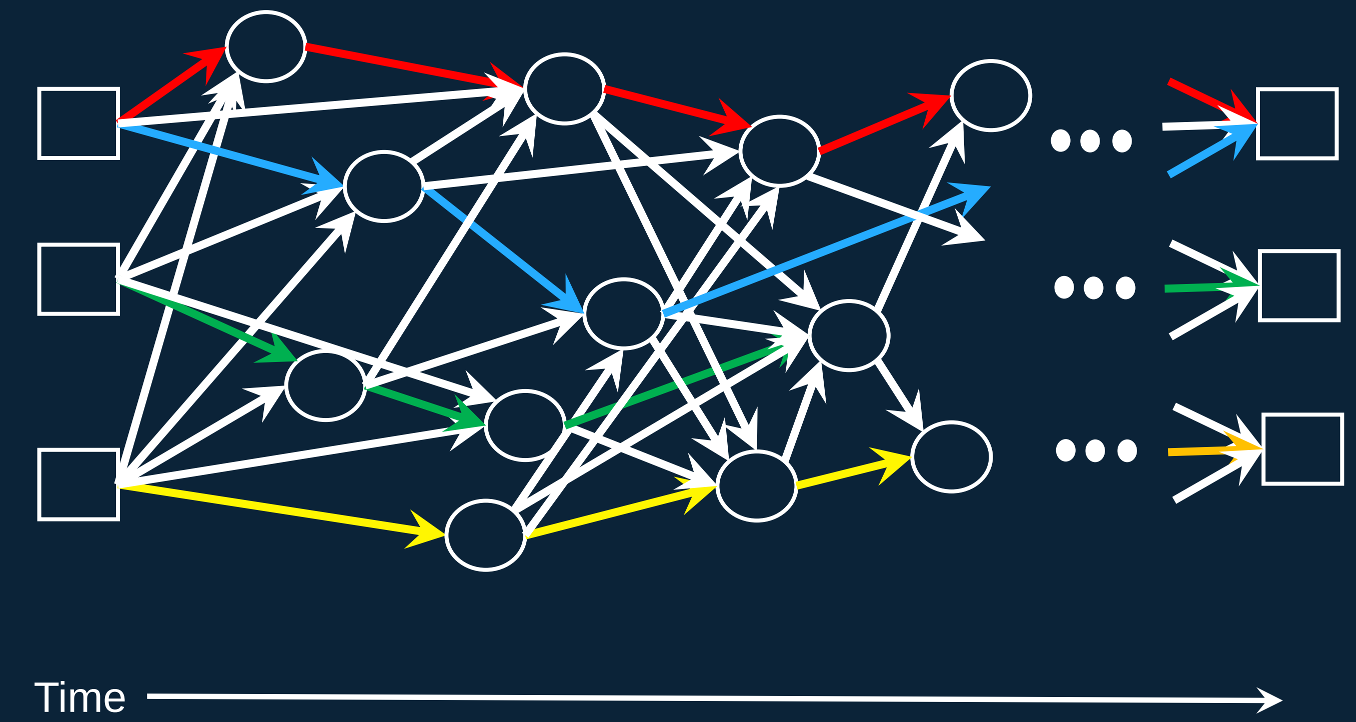
Graph Construction

- Graph defines the refueling sequences and track jumps to consider
  - **Nodes** are requests or bases
  - **Edges** specify back-to-back activities
  - **Edges** are not created if:
    - **Requests are too far apart in time**
    - **Tanker could not reposition in time for the next request**
    - **Requests would be needlessly bypassed**
  - **Paths** are potential sorties

- Graph implicitly enforces several constraints

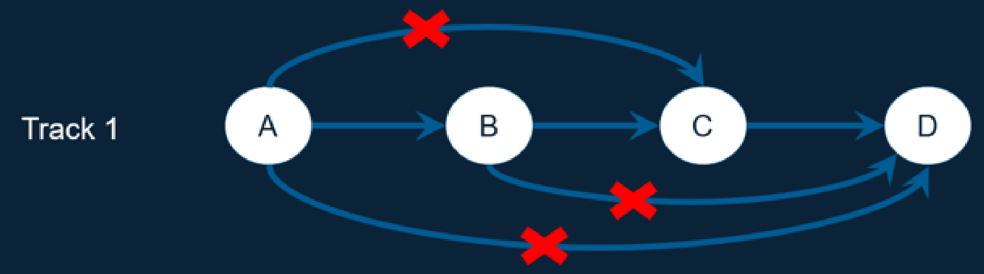


Notional time-space network with 3 bases

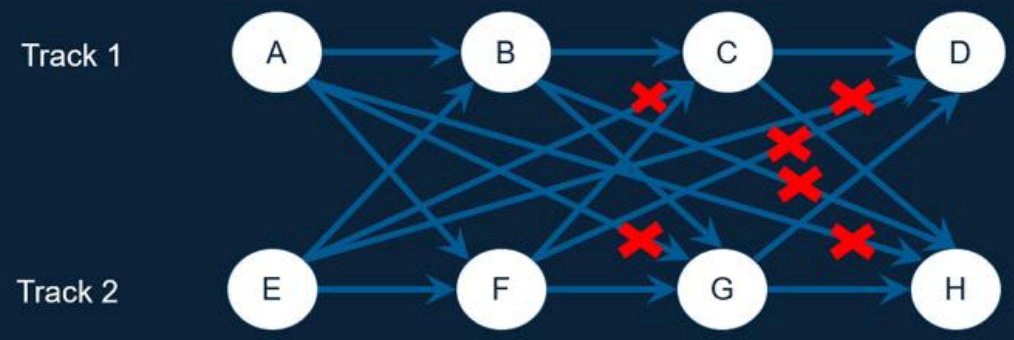


Graph explicitly enforces track capacity constraints

- Graph does not allow tankers to needlessly bypass requests
  - I.e., tanker would be in position to service a request but does not
- Prevents multiple tankers from existing in the same track at the same time
- Also a dimension reduction heuristic



Same-track notional example



Track jump notional example

Integer Programming Model

- Main Decision Variables:
  - 1 If tanker t should use edge **e** in the graph; 0 otherwise

- Minimizing weighted combination of:
  - Penalties for missing requests
  - Fuel burn estimates
  - Penalty for each additional sortie

- Basic constraints include:
  - One sortie and config per tanker
  - Tanker flow balance constraints
  - Request satisfaction constraints
  - Tanker capacity constraints
  - Maximum sortie length constraints

Minimize

$$\sum_{r \in R} pen_r \; skip_r + \sum_{\tau \in \mathcal{T}} burn_\tau \; e_\tau + \sum_{\tau \in \mathcal{T}(TO(.,.))} pen_{mx} \; e_\tau$$

$$\sum_{\tau \in \mathcal{T}(RS(r))} e_\tau + skip_r = 1 \qquad \forall\, r \in R$$

$$\sum_{\tau \in \mathcal{T}(TO(t))} e_\tau \leq 1 \qquad \forall\, t \in T$$

$$\sum_{\tau \in \mathcal{T}(RS(t,x,r))} e_\tau - \sum_{\tau \in \mathcal{T}(FS(t,x,r))} e_\tau = 0 \qquad \forall\, r \in R;\ \forall\, x \in CFG(t)$$

$$\forall\, t \in T$$

$$\sum_{\tau \in \mathcal{T}(TO(t))} (burn_\tau + deliv_\tau) e_\tau \leq cap(t) \qquad \forall\, t \in T$$

$$\sum_{\tau \in \mathcal{T}(TO(t))} time_\tau e_\tau \leq MaxTime \qquad \forall\, t \in T$$

Minimum Turn Time Constraints

- **Requirement:** Need a minimum amount of time before a tanker can fly a second sortie
- **Step 1:** Model a potential second sortie as a separate tanker and couple it with a "frontline" tanker
- **Step 2:** Add the following if-then constraints:
  - If turn tanker t_2 is going to use take-off edge e_j then corresponding frontline tanker t_1 must have used a landing edge that would afford at least X hours to turn the tanker
  - At least Y hours if the tanker is to be re-configured

$$\sum_{\tau \in \mathcal{T}(LE(\pi(t)),x,x,v) \bigcup \mathcal{T}(LE(\pi(t)),!x,x,v)} e_\tau \geq e_{t,x,h(t),v} \qquad \forall \ (t,x,h(t),v) \in \mathcal{T}(TTO(t))$$

Performance-Limited Request Constraints

- **Requirement:** Performance-limited requests require the tanker to be sufficiently empty so it can descend to a low altitude
- Add the following if-then constraints:
  - If tanker **t** is going to be assigned performance-limited-request **r** then tanker **t** must have burned and delivered a minimum amount of fuel prior to serving **r**

$$\sum_{\tau \in \mathcal{T}(RS(r,t))} burn_\tau \, e_\tau + \sum_{\tau \in \mathcal{T}(Bef(r,t))} (burn_\tau + deliv_\tau) \, e_\tau \geq MinOff(r,t) \sum_{\tau \in \mathcal{T}(RS(r,t))} e_\tau \qquad \forall \, t \in T; \forall \, r \in PLR$$

Heuristically Solving the Integer Program

- **Challenge:** Cannot feed the entire integer program to the solver at once
  - The solver may take too long to produce a decent looking solution

- **Solution:** Iteratively improve a solution by only working with a subset of the variables at any time

- Frequently called a **matheuristic approach**
  - **Matheuristic = Mathematical-programming-based heuristics**

Solving the Integer Program with Heuristics

Basic Idea:

- Define the complete integer program
- **For** each of two phases:
  - **While** not done:
    - Fix a subset of the variables
    - Solve the remaining sub-model as an integer program, starting from the current best known solution
    - Update the best known solution

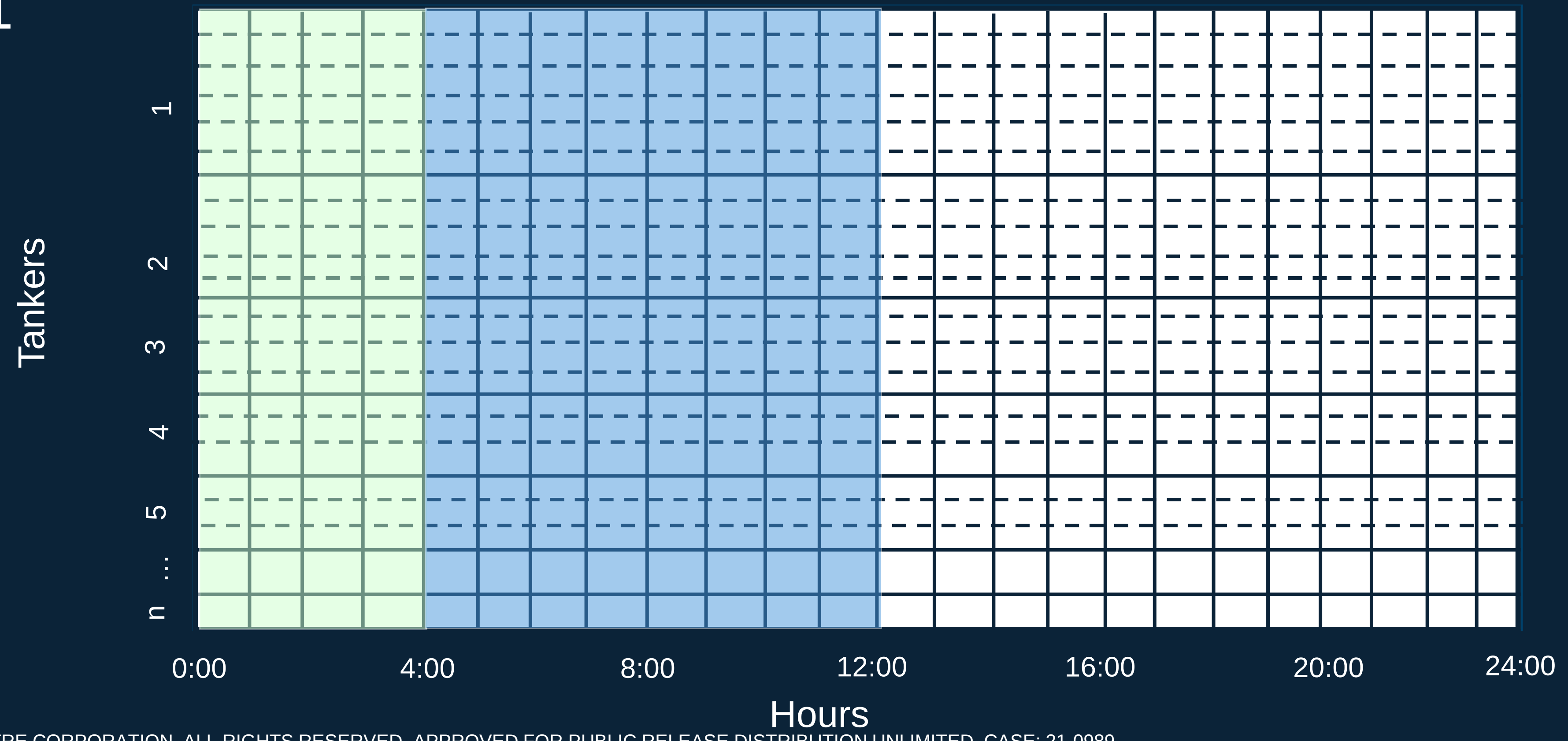
- Two phases:
  - **Sliding window phase** makes earlier day decisions before later ones
  - **Regional swap phase** looks at the entire planning window for subsets of tankers

Notional Example: The Sliding Window Phase

Key:

- (green) Decisions being optimized that will be locked by the end of this step
- (blue) Additional decisions being optimized

Sliding window phase step 1



Notional Example: The Sliding Window Phase

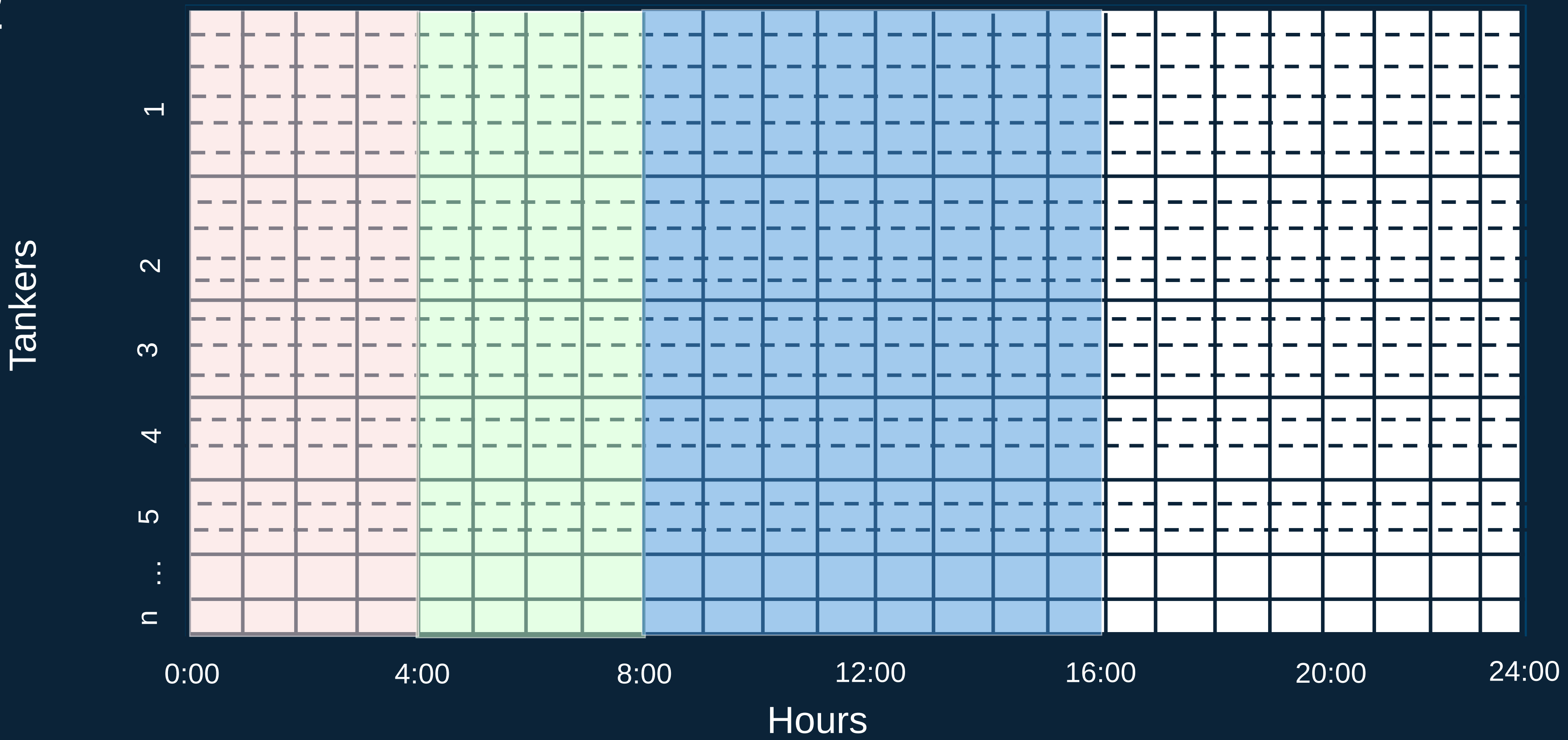
Key:

Decisions being optimized that will be locked by the end of this step

Additional decisions being optimized

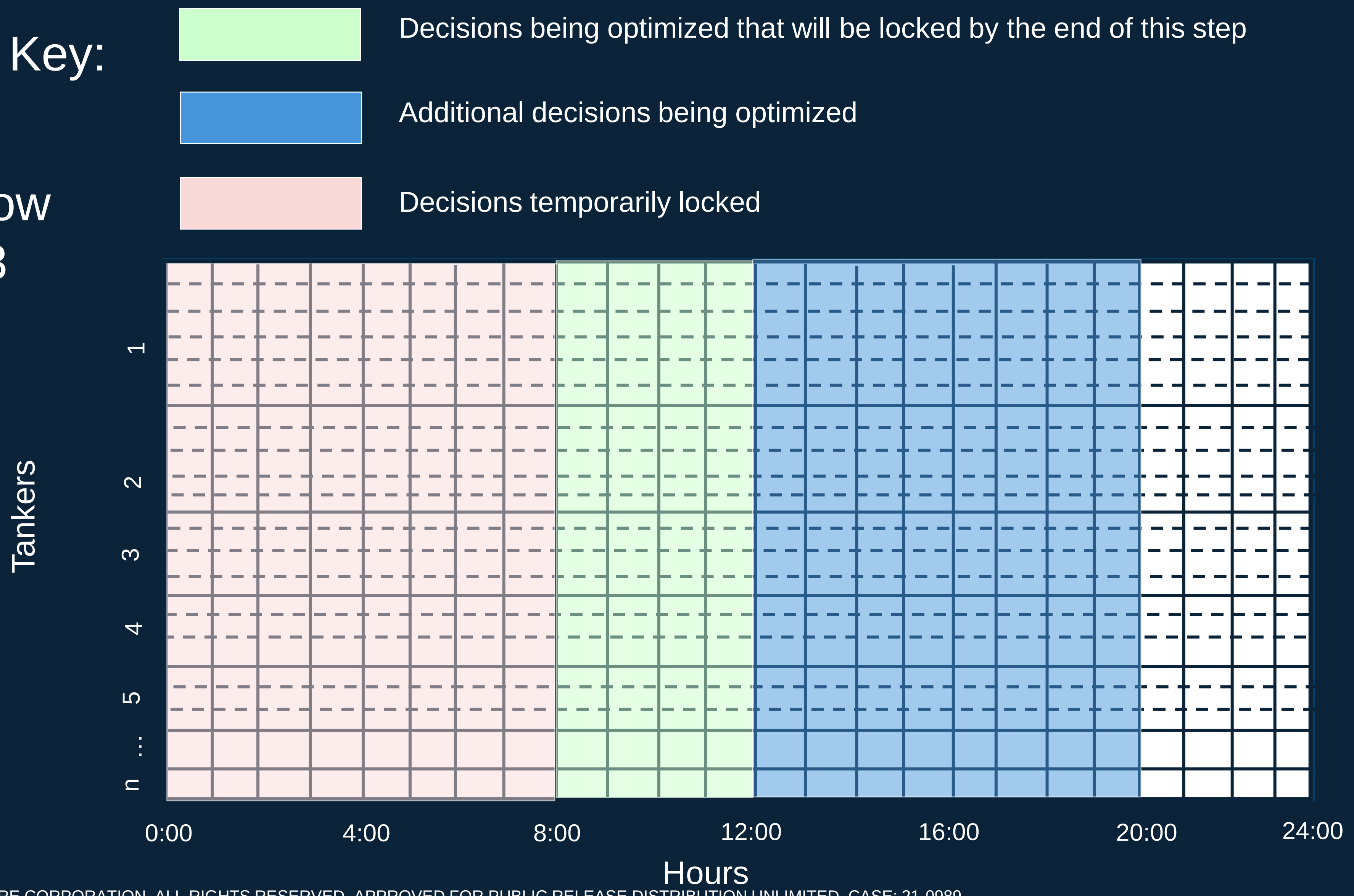
Decisions temporarily locked

Sliding window phase step 2



Notional Example: The Sliding Window Phase

Sliding window phase step 3

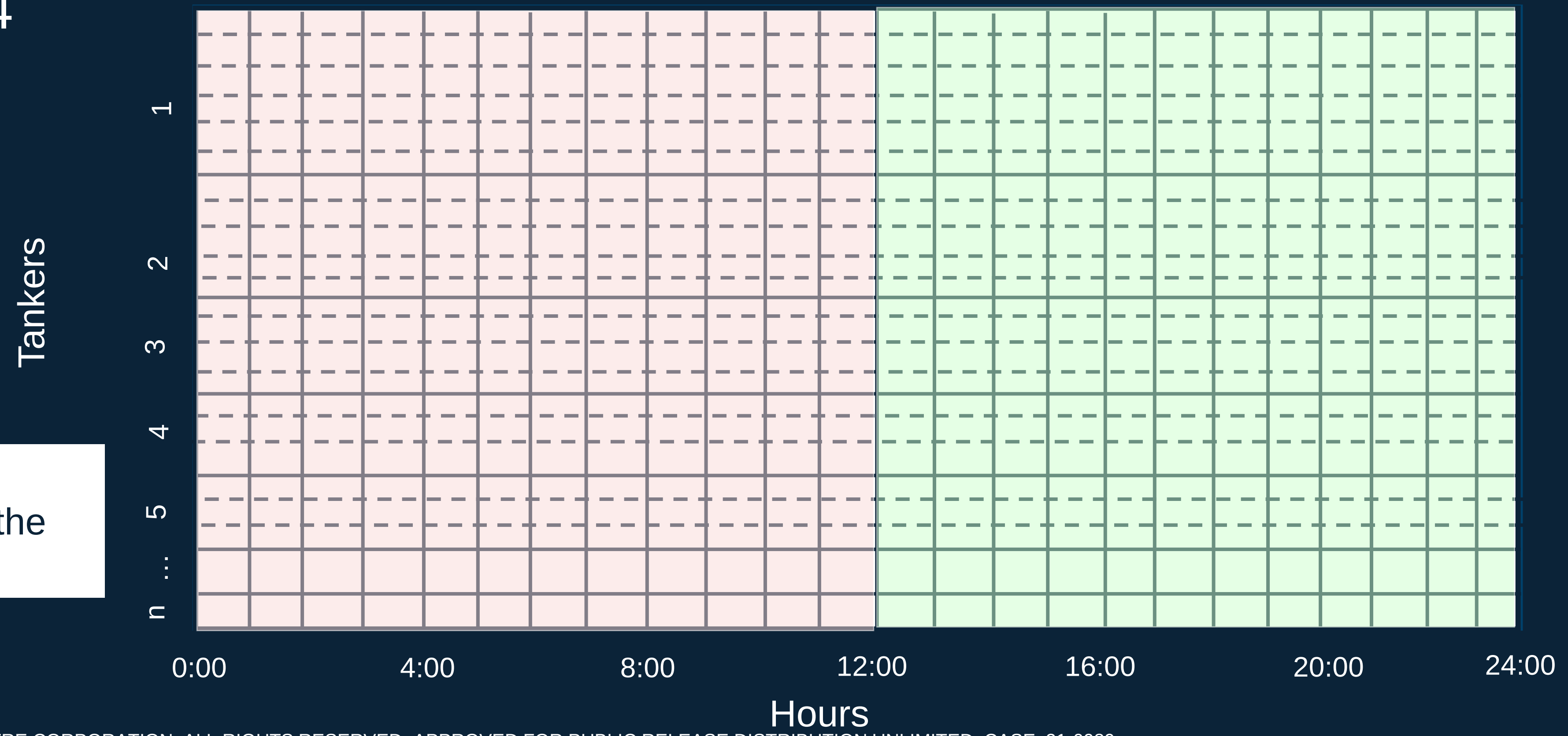


Notional Example: The Sliding Window Phase

Key:

- Decisions being optimized that will be locked by the end of this step
- Additional decisions being optimized
- Decisions temporarily locked

Sliding window phase step 4



Steps taken is configurable, as is the time spent per step

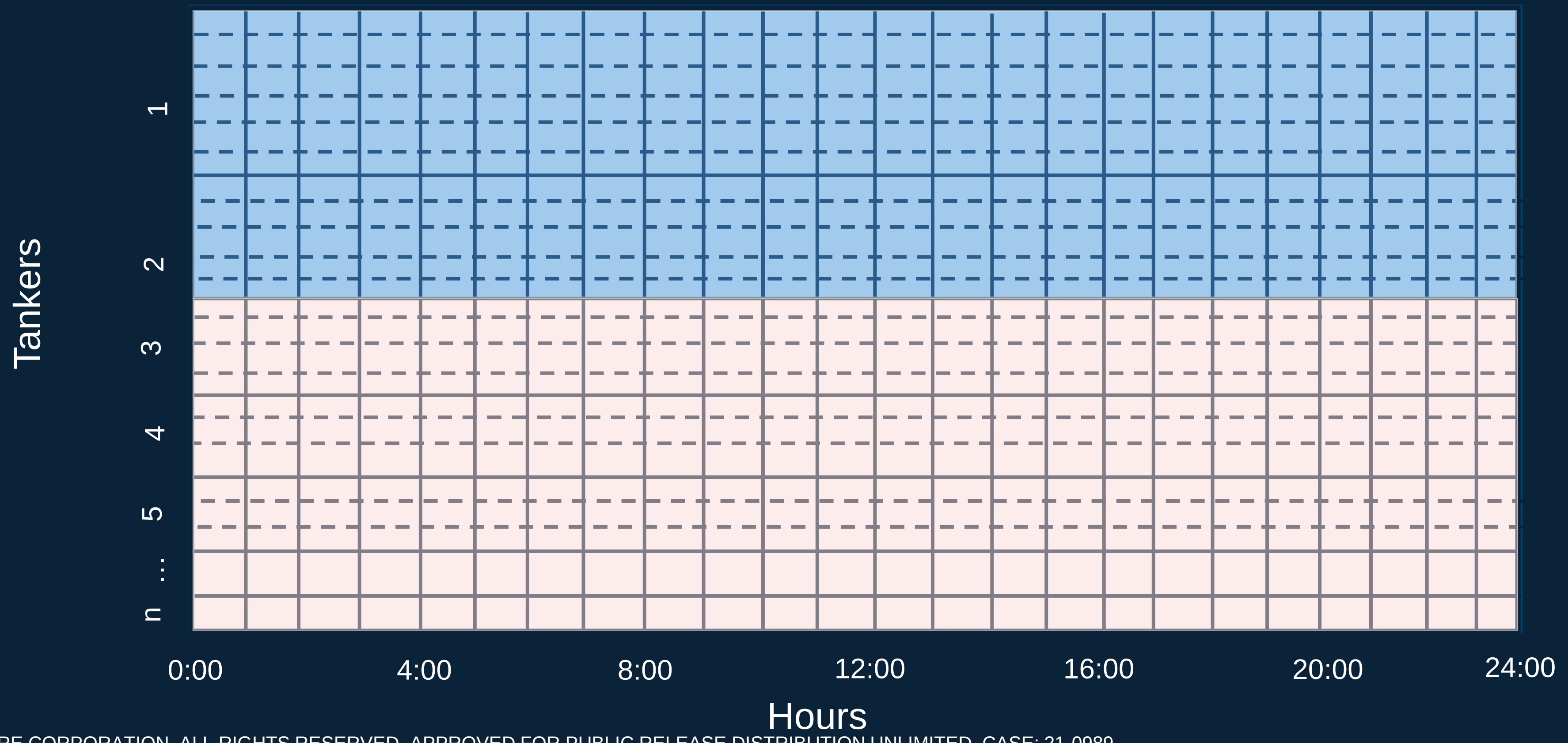
Notional Example: The Regional Swap Phase

Key:

Decisions being further optimized

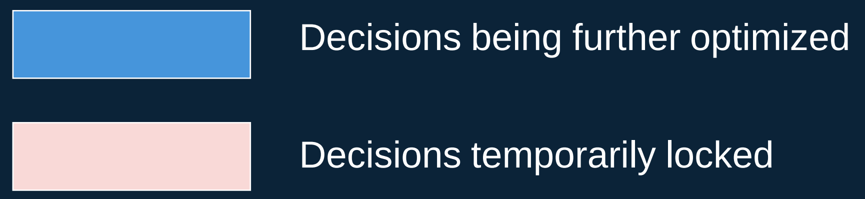
Decisions temporarily locked

Regional swap
step 1

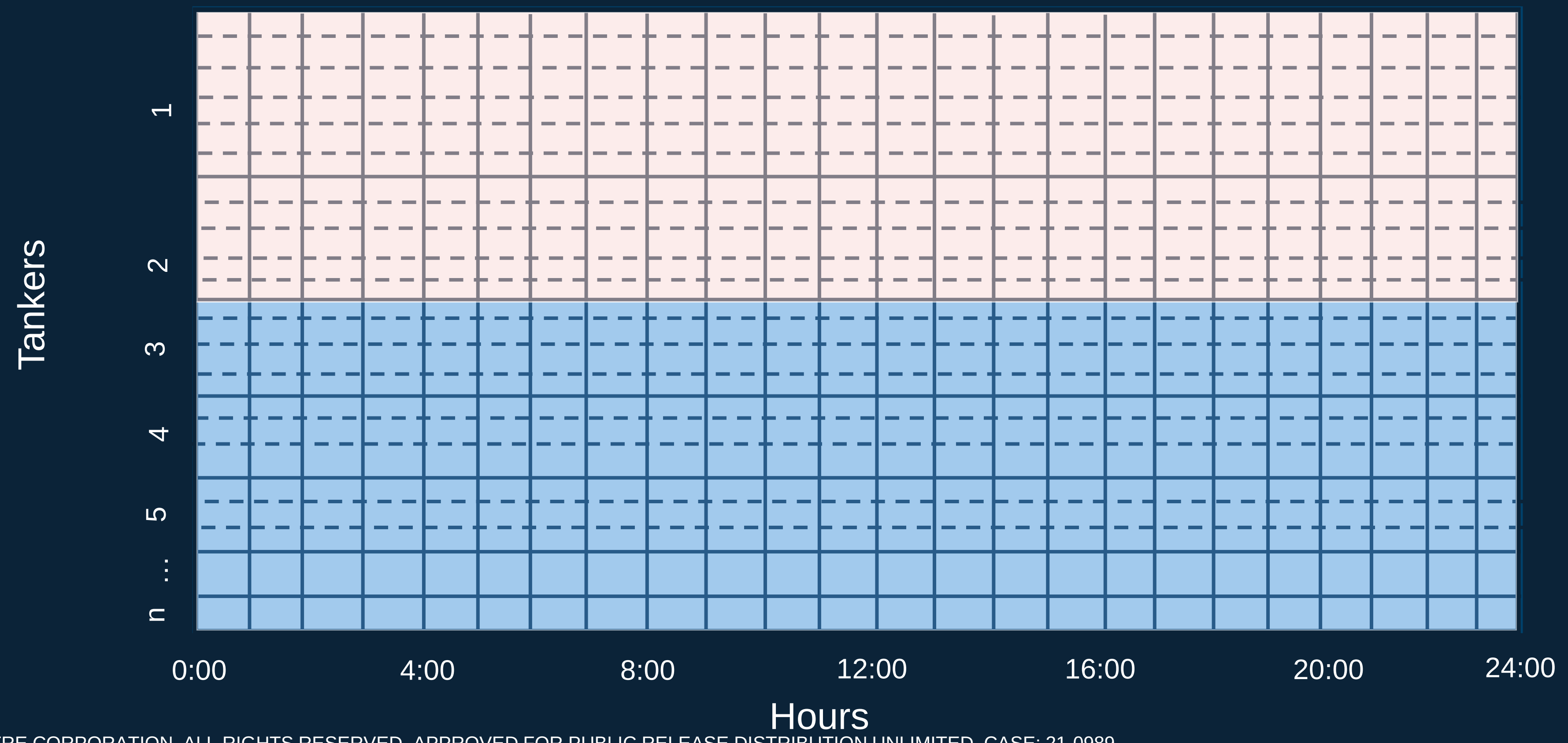


Notional Example: The Regional Swap Phase

Key:



Regional swap step 2



MITRE

Experimental Results

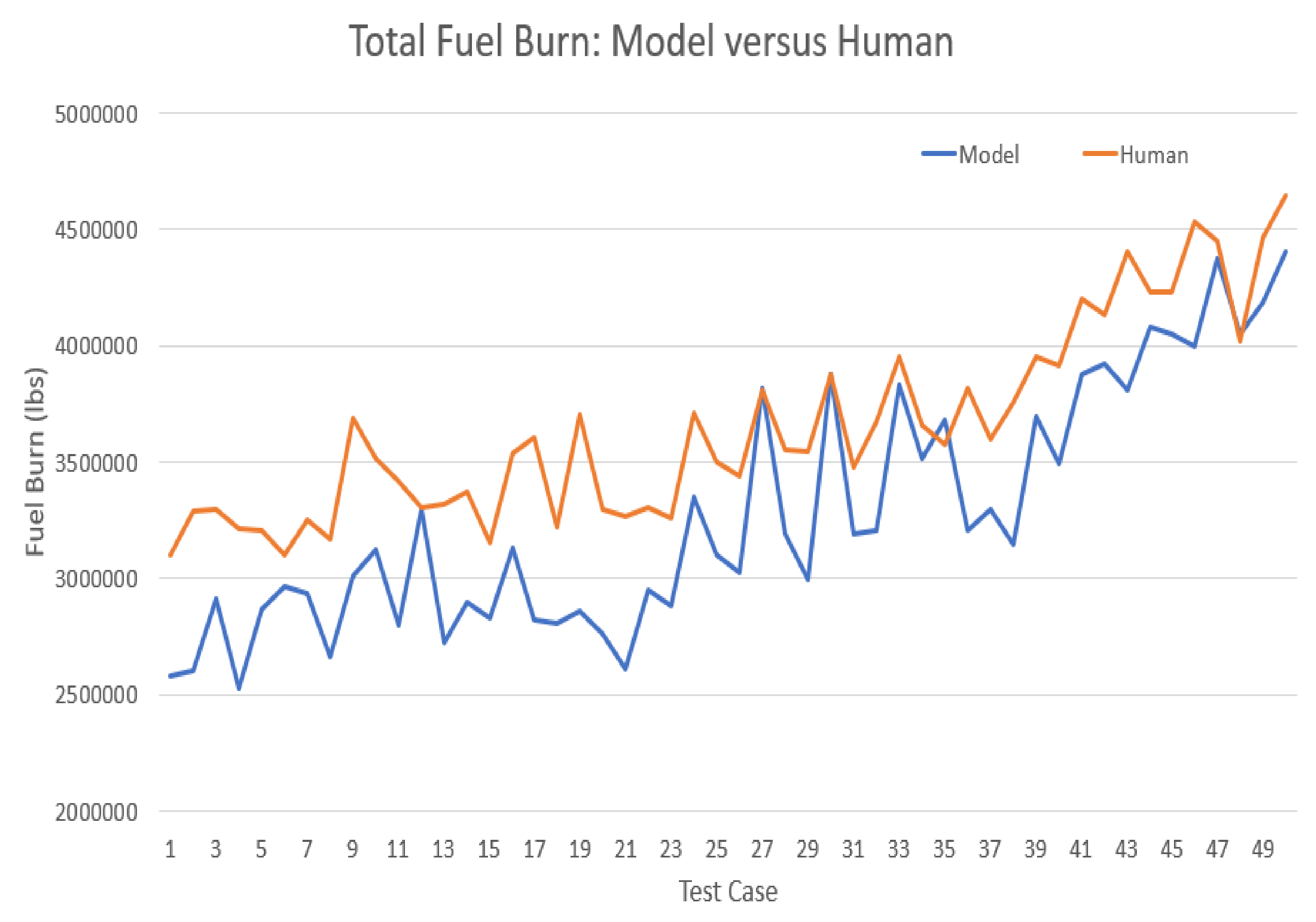
Man vs. Machine Testing

- Experiment Design:
  - Take (obfuscated) historical data on human plans
  - See how optimizer would have planned the same requests with the same resources with an expert-informed guess on the planning constraints
- Helps estimate potential efficiency gains

- Future Testing:
  - Man vs. Human-Machine Team testing (preferred to Man vs. Machine)
  - Test if optimizer allows operators to plan more requests with limited resources

Lab Results: Significant Potential Savings

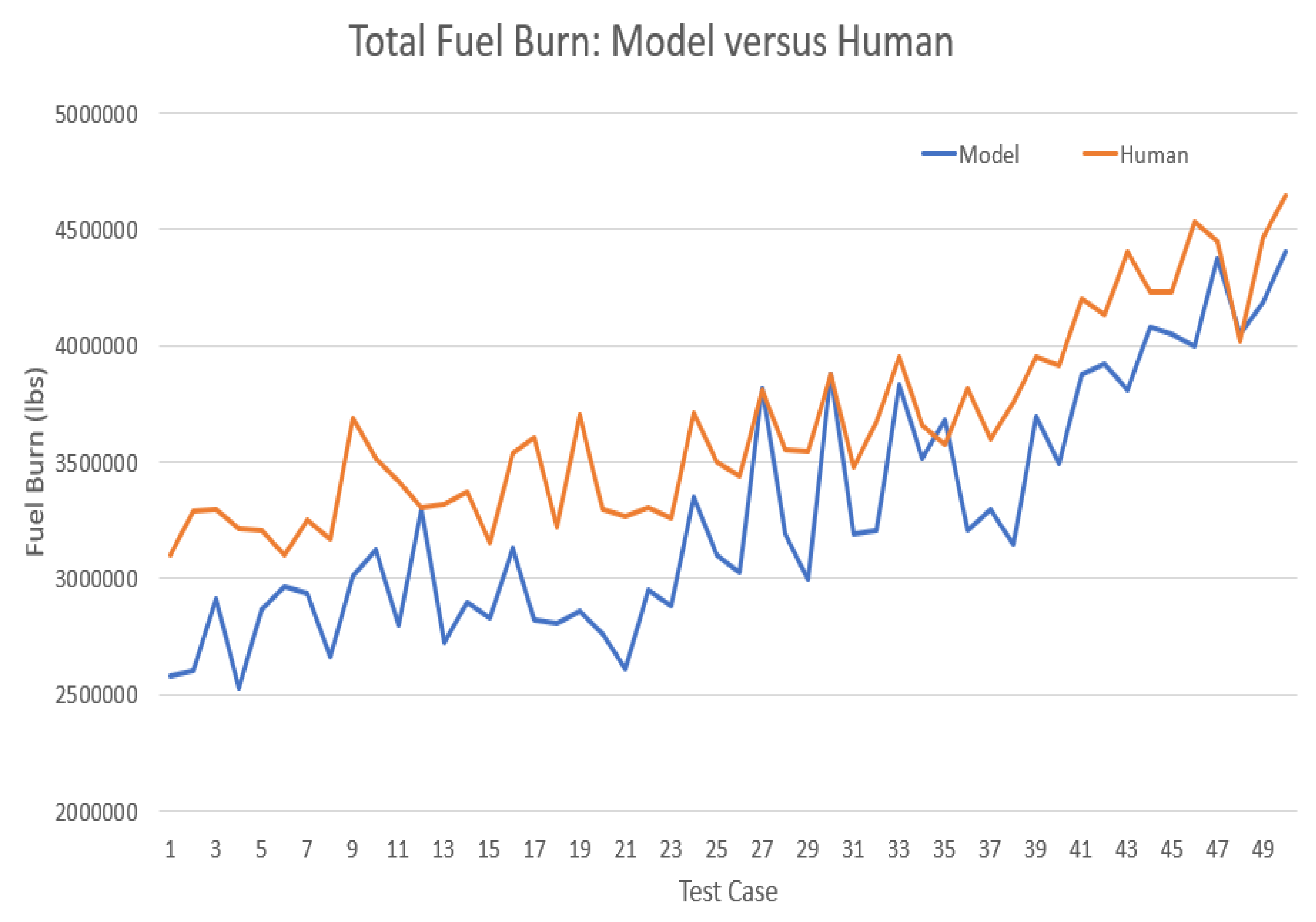
Current working results



- According to latest test results:
  - Model **burned 10% less fuel on average**
  - Model saved **over 350,000 lbs of fuel per day on average**
  - This could translate to saving an AOC **tens of millions in fuel per year** if these results are typical

Lab Results: Significant Potential Savings

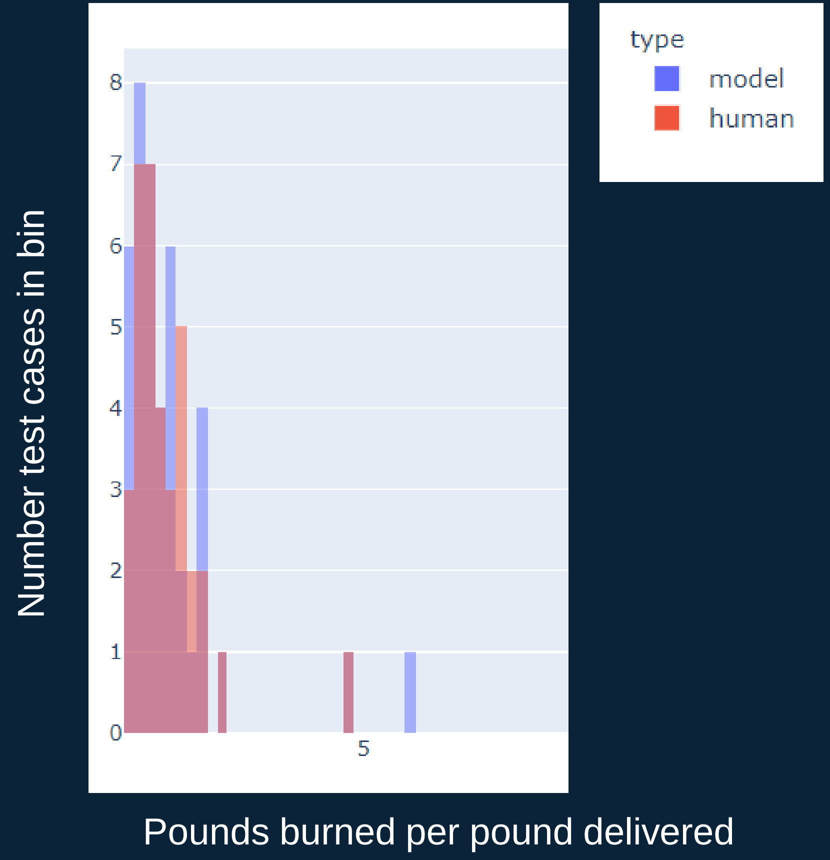
Current working results



- **Model does not always beat humans yet:**
- **Missing data**—sometimes humans made moves the model could not
- **Limited variables**—model currently cannot (even though humans can):
  - Change time of request
  - Change altitude or location of request
  - Have tankers temporarily wait in an empty track
  - Have tankers refuel other tankers

Lab Results: Man vs. Machine

Current working results



- In general, model was more fuel efficient but humans may have placed more value on having tankers available for short-notice requests
- Can be corrected by tuning the objective function (if needed)

Model Validation Strategy

- First-level validation: Valid if operational SMEs think output is useful
  - Must meet all scheduling requirements and provide a useful starting point for users
- Validating for human-machine teaming
  - Imperfections are okay if the user can easily fix them
  - Human-machine results should be measurably better than human-without-model results
- Must anticipate full-range of representative cases, including edge cases
  - Operators and researchers must collaborate on this

Opportunities for Further Improvement

- More degrees of freedom
  - Adjust request service times
  - Switch to nearby tracks or altitude blocks
  - Allow tankers to be temporarily moved to empty tracks
- Pre-caching information
  - So the model can “hot start” search once user is ready, providing responses in minutes
- Crew rest limitations
- Automated sensitivity analyses
  - Warn user about plan fragilities
- Refueling track re-positioning
  - When may it be operationally advantageous to move a track?
- Making algorithm more parallelizable
- Idle orbiting requests
  - Let users tell us when they want to pre-position a tanker in the air w/ spare fuel

Summary

- We are getting very promising lab results with our tanker planning optimizer:
  - Could potentially save **a major AOC tens of millions per year** in fuel (but it could instead be tuned to save sorties)
  - Automatically generates plans that meet complex requirements faster than humans
  - Can help operators fulfill more requests on days when tankers are limited

- An optimization capability won't just optimize; it can double as a what-if tool