2023-06

# OPTIMIZING SERVICE LEVEL TRAINING EXERCISE (SLTE) SCHEDULES USING RANGE SCHEDULING ASSISTANCE TOOL (RSAT)

Byers, Colton L.

Monterey, CA; Naval Postgraduate School

https://hdl.handle.net/10945/72126

# NAVAL POSTGRADUATE SCHOOL

**MONTEREY, CALIFORNIA**

# THESIS

**OPTIMIZING SERVICE LEVEL TRAINING EXERCISE (SLTE) SCHEDULES USING RANGE SCHEDULING ASSISTANCE TOOL (RSAT)**

by

Colton L. Byers

June 2023

Thesis Advisor: Jefferson Huang
Second Reader: Emily M. Craparo

**Approved for public release. Distribution is unlimited.**

THIS PAGE INTENTIONALLY LEFT BLANK

| REPORT DOCUMENTATION PAGE | | | | *Form Approved OMB No. 0704-0188* |
|---|---|---|---|---|



| | | |
|---|---|---|
| **1. AGENCY USE ONLY** ***(Leave blank)*** | **2. REPORT DATE** June 2023 | **3. REPORT TYPE AND DATES COVERED** Master's thesis |

| | |
|---|---|
| **4. TITLE AND SUBTITLE** OPTIMIZING SERVICE LEVEL TRAINING EXERCISE (SLTE) SCHEDULES USING RANGE SCHEDULING ASSISTANCE TOOL (RSAT) | **5. FUNDING NUMBERS** |
| **6. AUTHOR(S)** Colton L. Byers | |
| **7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)** Naval Postgraduate School Monterey, CA 93943-5000 | **8. PERFORMING ORGANIZATION REPORT NUMBER** |
| **9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)** N/A | **10. SPONSORING / MONITORING AGENCY REPORT NUMBER** |

**11. SUPPLEMENTARY NOTES** The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

| | |
|---|---|
| **12a. DISTRIBUTION / AVAILABILITY STATEMENT** Approved for public release. Distribution is unlimited. | **12b. DISTRIBUTION CODE** A |

**13. ABSTRACT (maximum 200 words)**

The Marine Corps conducts its largest Service Level Training Exercise (SLTE) four times a year and can include more than four reinforced battalion-sized forces. The Marine Corps' Integrated Training Exercise (ITX) is resource intensive and requires an enormous amount of planning hours prior to execution. Currently, the planning and scheduling for the SLTE is conducted manually and follows a standard schedule of events. The Range Schedule Assistance Tool (RSAT), a linear optimization program, will optimize SLTE schedules subject to training unit requirements, unit priority lists, and range availability. The primary output will be a deconflicted range schedule that will include the range's assignments, training dates and times, and type of training event to serve as an initial planning document for schedule development. The RSAT will be designed to minimize penalties based on commander's guidance, schedule changes, and unscheduled events. The main inputs for the RSAT will come from supervising commands, executing units, and range guidance. Additionally, RSAT can be used to evaluate resource and range capacity for SLTEs.

| | |
|---|---|
| **14. SUBJECT TERMS** optimization, scheduling, range, schedule, linear program, Service Level Training Exercise, SLTE, Integrated Training Exercise, ITX, Range Schedule Assistance Tool, RSAT | **15. NUMBER OF PAGES** 53 |
| | **16. PRICE CODE** |

| | | | |
|---|---|---|---|
| **17. SECURITY CLASSIFICATION OF REPORT** Unclassified | **18. SECURITY CLASSIFICATION OF THIS PAGE** Unclassified | **19. SECURITY CLASSIFICATION OF ABSTRACT** Unclassified | **20. LIMITATION OF ABSTRACT** UU |

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. 239-18

THIS PAGE INTENTIONALLY LEFT BLANK

**Approved for public release. Distribution is unlimited.**

**OPTIMIZING SERVICE LEVEL TRAINING EXERCISE (SLTE) SCHEDULES USING RANGE SCHEDULING ASSISTANCE TOOL (RSAT)**

Colton L. Byers
Captain, United States Marine Corps
BS, United States Naval Academy, 2017

Submitted in partial fulfillment of the
requirements for the degree of

**MASTER OF SCIENCE IN OPERATIONS RESEARCH**

from the

**NAVAL POSTGRADUATE SCHOOL**
**June 2023**

Approved by: Jefferson Huang
Advisor

Emily M. Craparo
Second Reader

W. Matthew Carlyle
Chair, Department of Operations Research

THIS PAGE INTENTIONALLY LEFT BLANK

# ABSTRACT

The Marine Corps conducts its largest Service Level Training Exercise (SLTE) four times a year and can include more than four reinforced battalion-sized forces. The Marine Corps' Integrated Training Exercise (ITX) is resource intensive and requires an enormous amount of planning hours prior to execution. Currently, the planning and scheduling for the SLTE is conducted manually and follows a standard schedule of events. The Range Schedule Assistance Tool (RSAT), a linear optimization program, will optimize SLTE schedules subject to training unit requirements, unit priority lists, and range availability. The primary output will be a deconflicted range schedule that will include the range's assignments, training dates and times, and type of training event to serve as an initial planning document for schedule development. The RSAT will be designed to minimize penalties based on commander's guidance, schedule changes, and unscheduled events. The main inputs for the RSAT will come from supervising commands, executing units, and range guidance. Additionally, RSAT can be used to evaluate resource and range capacity for SLTEs.

THIS PAGE INTENTIONALLY LEFT BLANK

# Table of Contents

# List of Figures

THIS PAGE INTENTIONALLY LEFT BLANK

# List of Tables

THIS PAGE INTENTIONALLY LEFT BLANK

# List of Acronyms and Abbreviations

**FSCEX** Fire Support Exercise

**GRC** Ground Reconnaissance Course

**ITX** Integrated Training Exercise

**MAGTF** Marine Air Ground Task Force

**MCAGCC** Marine Corps Air Ground Combat Center

**MDMX** MAGTF Distributed Maneuver Exercise

**MIP** Mixed-Integer Linear Program

**MTC** Movement to Contact

**MWX** MAGTF Warfighting Exercise

**RSAT** Range Scheduling Assistance Tool

**RSOI** Reception, Staging, Onward Movement and Integration

**RTA** Range Training Areas

**SAT** Scheduling Assistance Tool

**SLTE** Service Level Training Exercise

**TTECG** Tactical Training and Exercise Control Group

THIS PAGE INTENTIONALLY LEFT BLANK

# Executive Summary

The Integrated Training Exercise (ITX) is the premier live-fire training exercise for the United States Marine Corps. The exercise is the branch's largest and most dynamic Service-Level Training Exercise (SLTE) and is executed four times a year in Twentynine Palms, CA at Marine Corps Air Ground Combat Center (MCAGCC). Each iteration of the exercise typically involves more than 65 individual units and is composed of more than 75 events. The ITX schedule is specifically tailored according to the ranges available, executing units, unit requirements, and staffing/resources available. The current SLTE scheduling efforts are conducted manually, are inefficient, and are prone to error. Each ITX schedule goes through multiple iterations of adjustments to meet specific unit needs and requests. Automating the early iterations of scheduling efforts would reduce the number of errors and total time required to produce the finalized schedule.

This thesis proposes a mixed-integer linear program called RSAT (Range Schedule Assistance Tool), which looks to produce an initial scheduling solution for ITX's planners. RSAT uses a minimum cost multicommodity network flow structure to optimize a deconflicted ITX event schedule. RSAT is programmed in Python using the Pyomo open-source optimization programming language with Gurobi's LP solver. The objective function heavily penalizes scheduling excess events for a given unit and the cost of scheduling events increases as the planning horizon proceeds. Simply put, RSAT looks to assign and complete all required events as soon as possible while being constrained by unit and event relationships.

After assessing multiple arc cost functions, we find that RSAT is able to produce viable scheduling solutions for ITX in less than a minute. RSAT's performance was tested on two realistic-sized instances. First during development, RSAT was used to produce scheduling solutions following the ITX 2-23 exercise schedule template. After initial development, RSAT was then applied to ITX 5-23 as a subsequent verification of scheduling solution. In both cases, RSAT was able to produce daily deconflicted training schedules for ITX planners to use as initial planning and refinement documents. The application of RSAT in these use cases allows planners to automate initial and adhoc scheduling efforts.

THIS PAGE INTENTIONALLY LEFT BLANK

# Acknowledgments

First and foremost, I would like to thank my advising team, Professor Huang and Professor Craparo. Your guidance, expertise, and patience throughout this thesis have been invaluable. I am grateful for the hours you invested in reviewing my work and providing constructive feedback. I could not have asked for better mentors.

I would also like to extend my deepest appreciation to my last Company Commander and Assistant Operations Officer at TTECG, Major Leeds. Your collaboration, insights, and feedback have significantly elevated this project. Your willingness to share your expertise and provide data and resources have been instrumental in shaping the research outcome. Working for and alongside you all these years has been a privelege.

To my amazing wife, Kelley, words cannot express my gratitude for your unwavering love, understanding, and encouragement throughout this demanding journey. Your patience, support, and belief in me cannot be overstated. Thank you for taking care of our daughters and listening to me vent about every stressful homework, project, quiz, and test I've taken for two years. I am truly blessed to have you in my life.

Last but not least, I would like to acknowledge my fellow Operations Research classmates. Your camaraderie, collaboration, and support have made this program fulfilling and fun, at times. Our data analysis discussions, stochastic study sessions, and shared pain from Pyomo have created a tight community we can all call on for support. Special thanks to each one of you for your friendship.

To everyone mentioned above, and to all those who have contributed to my journey in any way, I am profoundly grateful. Thank you all for being an important part of this process and for helping me achieve my goals.

THIS PAGE INTENTIONALLY LEFT BLANK

# CHAPTER 1: Introduction and Background

## 1.1 Integrated Training Exercise

The Integrated Training Exercise (ITX) is the premier live-fire training exercise for the United States Marine Corps. In addition to being the branch's largest and most dynamic Service-Level Training Exercise (SLTE), it also serves as a testing venue for the Marine Corps to perform real-world evaluations of emerging concepts of operations. The exercise is executed four times a year in Twentynine Palms, CA at Marine Corps Air Ground Combat Center (MCAGCC). Each iteration of the exercise typically involves more than 65 individual units and is composed of more than 75 events. The exercise is managed and facilitated by a Marine cadre assigned to the Tactical Training and Exercise Control Group (TTECG). According the TTECG handbook, the cadre is tasked with "designing and supervising the conduct of service level, live-fire and maneuver combined arms exercises in order to train battalion/squadron size MAGTF units in the tactics, techniques, and procedures required to execute their core, and selected core-plus mission essential tasks (METs)" (Tactical Training and Exercise Control Group, Marine Air Ground Task Force Training Command, Marine Corps Air Ground Combat Center 2018). While TTECG does not evaluate the performance of executing units in ITX, the SLTE serves as an important data point for the service to use during force design and capability investment decisions.

In recent years the structure of ITX has changed to the following five parts:

1. Reception, Staging, Onward Movement and Integration (RSOI)
2. 400 range series and Fire Support Exercise (FSCEX)
3. Movement to Contact (MTC) and Ground Reconnaissance Course (GRC)
4. MAGTF Distributed Maneuver Exercise (MDMX)
5. MAGTF Warfighting Exercise (MWX)

This thesis focuses on the events that comprise parts two through four. The majority of these events are completed prior to the culminating event, MWX, which all executing units are required to simultaneously complete.

## 1.2 Training Area

MCAGCC's range training areas (RTAs) are separated into bounded land and air space, shown in Figure 1.1. This allows multiple units to deconflict training schedules and gives them purpose-built spaces to accomplish METs during their predeployment training cycles. In addition to schedule deconfliction, the sectioned training areas ensure that units can safely employ weapon systems while mitigating the risk of injuring or incapacitating units in adjacent ranges. ITX events use the same deconfliction measures as regular training to mitigate risks and efficiently complete events during the SLTE.

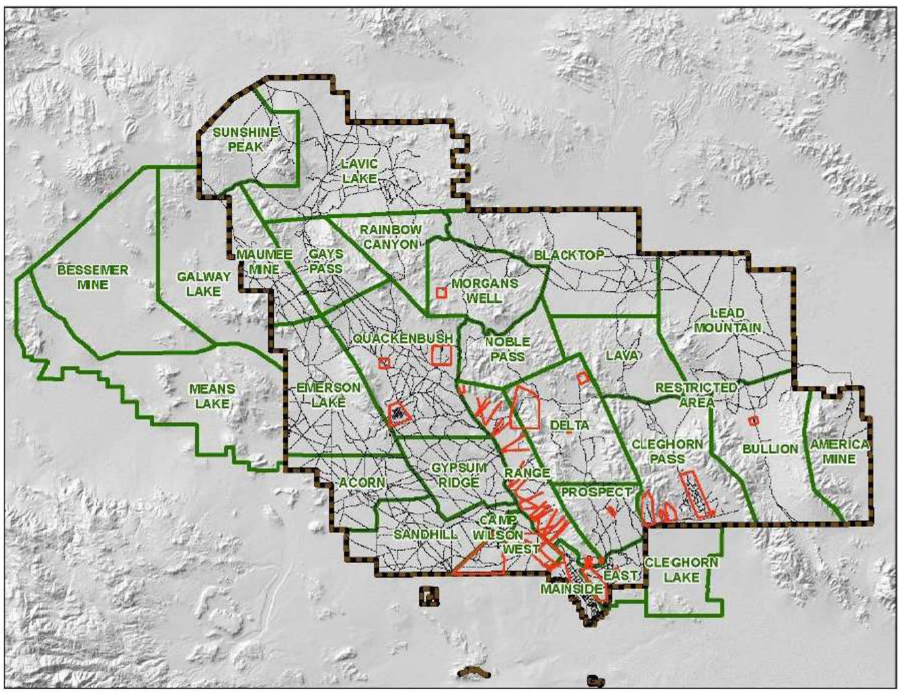


Figure 1.1. Map of MCAGCC range training areas. Source: Marine Air Ground Task Force Training Command, Marine Corps Air Ground Combat Center, Range, Training Area and Airspace (RTAA) Program (2022).

## 1.3 Training Events

As previously mentioned, there are five main parts of ITX's execution, with this thesis focusing on the events executed in parts two through four. Each of the parts are comprised of a number of main events and multiple sub-events for preparation and debriefing. Additionally, support units such as the Combat Logistic Battalion will execute auxiliary events to accomplish logistic-centric METs. In general, the training events in each part of ITX follow a progression, in terms of complexity and size, and have associated prerequisites to mitigate risk and ensure MET competency. As an example, in part four of ITX, the MDMX is composed of five sub-events within the structured series: MDMX planning, MDMX CAST, Rehearsal of Conduct (ROC)/Reconnaissance and Surveillance (R&S) Insert, MDMX, and MDMX Debrief.

Executing units may be required to execute some or all of the events in the series. The prerequisites of each event also depend on the executing unit. For example, the Motor Transport Platoon is not required to complete MDMX planning but is required to complete MDMX and MDMX Debrief. This is directly dependent on the function the unit will provide in the execution of an event. As a supporting unit, Motor Transport Platoon is not necessarily required to execute all sub-events leading to the main event in a series because they need less preparation to conduct convoy operations in support of the event when compared to a unit executing live-fire and maneuver.

During the conduct of training events, groups of units may execute the same event at the same time to satisfy the requirements for the event. For example, the company-supported live-fire range R400 requires a rifle company, elements of a weapons company, an engineer detachment, and, in some cases, an Amphibious Assault Vehicle platoon to satisfy range and mobility requirements. These groupings change throughout the execution of training events based on range and unit requirements, as well as the capacity of a given range. The dynamic nature of event series, range requirements, and group composition make each ITX schedule unique and time-intensive to maintain.

## 1.4 Current Scheduling Practice and Methodology

TTECG currently creates, manages, and revises the ITX schedule manually according to a general template. The general template is based on a standard ITX unit composition and requirements. During a series of conferences leading to RSOI of the units in MCAGCC, Twentynine Palms the schedule is revised to be consistent with the specific unit composition for the current ITX iteration, executing unit requirements, and range space availability. In the months leading to the first training day, the operations shop at TTECG is required to work through each cell pictured in Figure 1.2. The schedulers manually work through the range requirements, series of events, unique executing groups, and cross-check range availability to produce a schedule. While schedulers have cultivated a nearly optimized schedule, in terms of minimizing overall exercise length and meeting leadership priorities, they must work through competing interests and multiple unique updates manually for each iteration of ITX.

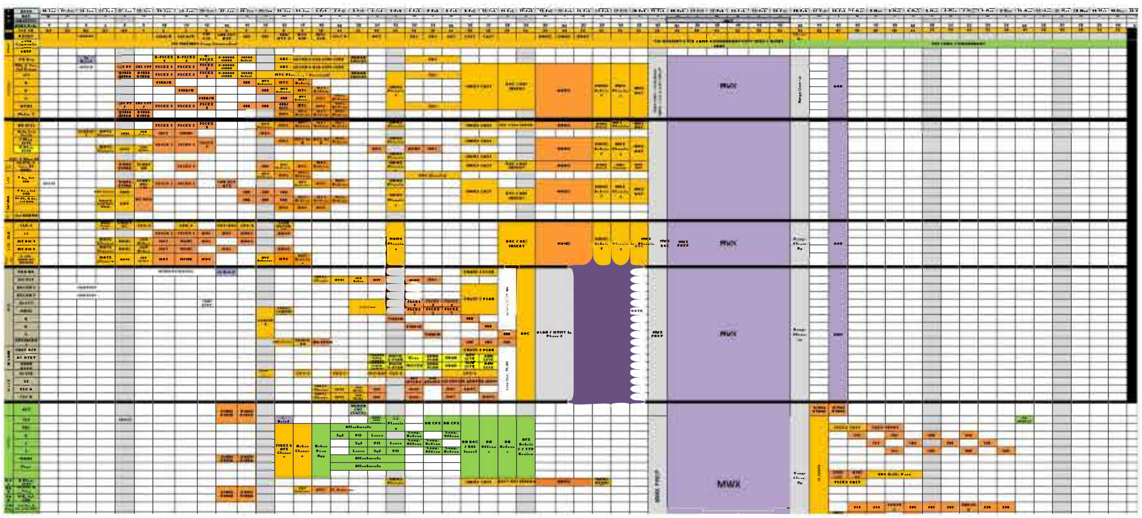

Figure 1.2. ITX schedule template. Each for listed in the first column in a unit. Each column is a training day in the planning horizon with the last day in the furthest right column. Source: Leeds (2023).

## 1.5 Course of Study

The current SLTE scheduling efforts are conducted manually, are inefficient, and are prone to error. Each ITX schedule goes through multiple iterations of adjustments to meet specific unit needs and requests. An initial schedule solution that considers the unique unit composition and training requirements would reduce the number of man-hours dedicated to schedule adjustments and free up TTECG's operations personnel for other priority tasks. Additionally, a computer-generated solution has the potential to produce a more efficient schedule.

This thesis proposes a mixed-integer linear program known as RSAT (Range Schedule Assistance Tool), which looks to produce an initial scheduling solution for ITX's planners. RSAT uses a minimum cost multicommodity network flow structure to optimize a deconflicted ITX event schedule. Chapter 2 discusses relevant work that helped in the formulation of RSAT. Chapter 3 describes the formulation and construction of the RSAT model. Chapter 4 outlines the results and performance of RSAT. Chapter 5 concludes the thesis with conclusions and recommendations for further improvements and applications of the model.

THIS PAGE INTENTIONALLY LEFT BLANK

# CHAPTER 2:
# Literature Review

The model and findings in this thesis are motivated by other mixed integer linear programs (MIPs) with military scheduling applications and the minimum cost multi-commodity flow network structure discussed by Bellmore et al. (1971). This chapter reviews the Schedule Assistance Tool (SAT) developed by Slye (2018), the training scheduling program that was improved by DeWinter (2012), and the multi-commodity flow tanker schedule discussed by Bellmore et al. (1971).

## 2.1 Optimizing Training Event Schedules at Naval Air Station Fallon

Slye (2018) develops SAT to optimize the daily flight training schedule at Naval Air Station (NAS) Fallon. SAT uses limited training resources to produce a daily flight schedule over a 31-day planning horizon that minimizes the cost of scheduling and rescheduling flight events. The model schedules flight events according to unit and training area requirements, available aircraft, unit priority, and rescheduling penalties. Due to the expensive nature of aircraft training and the limited resources at NAS Fallon, SAT penalizes schedules changes to maintain the original schedule as much as possible. The unit training requirements in SAT follow a prescribed continuum that requires events be scheduled in a particular order with the required support aircraft. The training areas in SAT are capacity constrained and deconflicted according to grouped airspace above the land training areas.

RSAT leverages a structure inspired by SAT. Over a SLTE planning horizon (~45 days), RSAT considers unit, range space, and SLTE requirements to produce a daily training schedule for executing units. While SAT schedules events according to half-hour increments, RSAT schedules ranges according to entire days. The biggest difference between RSAT and SAT, other than focusing on ground military units, is the number of intended model iterations. RSAT looks to optimally solve the SLTE schedule once for staff planning, while SAT is solved daily for a subsequent 31-day planning horizon to maintain the daily training schedule.

## 2.2 Persistent Planning Model for Explosive Ordnance Disposal Training and Evaluation Unit Two

The training schedule optimization model discussed in DeWinter (2012) optimizes the use of limited resources, including personnel and materiel, while maximizing training value. The model, named EODSKED, optimizes training schedules over a predeployment schedule horizon for a variety of different platoon types each with different training requirements. Additionally, EODSKED integrates persistence in subsequent model solutions. This allows the unit to amend training schedules and practically react to training requirements or resource availability changes with minimum schedule disruption. EODSKED's planning horizon and resource consideration differ slightly from RSAT despite their similar use cases. While both models look to optimally schedule events for multiple units, EODSKED has an expanded planning horizon (months to years), for a limited number of units. Additionally, EODSKED considers individual personnel and material resources, such as trainers and trucks, in the schedule solution. In contrast, RSAT has a limited planning horizon within an isolated service-level training exercise but schedules more events for more units. Furthermore, RSAT makes assumptions about lower-level resource availability and focuses more on unit-to-unit, event-to-event, and unit-to-event relationships. While the use cases of EODSKED and RSAT are similar the objective functions and model constructions differ. EODSKED looks to maximize the objective function in a MIP construction while RSAT minimizes its objective function within a multicommodity network flow formulation.

## 2.3 Multivehicle Tanker Scheduling Problem

The multivehicle tanker scheduling program developed by Bellmore et al. (1971) uses a minimum cost network flow formulation to schedule a non-homogeneous tanker fleet subject to arc constraints. The objective of this model is to produce a tanker schedule and routing that maximizes utility against certain penalties. In the shipping network, nodes represent corresponding ports of origin and delivery at allowable origination and receipt times. The network graph is acyclic, with all arcs representing a forward progression in time. The flow constraints over arcs permit an acceptable amount of non-negative flow by select types of tankers. Additionally, tankers are capacity-constrained based on the tanker type and are allowed to be partially loaded. The combination of tanker, cargo load, and chosen arc contribute to the objective value according to the tanker type, cargo size, and utility (cost) of the chosen node. In this way, the authors maximize the utility of the network by utilizing high yield arcs in combination with fully loaded tankers.

Similarly, RSAT uses a minimum cost network flow construction to schedule unit training in the confines of an SLTE. The arc constraints in RSAT control the flow of units and ensure that the appropriate events are scheduled according to unit and SLTE design requirements.

THIS PAGE INTENTIONALLY LEFT BLANK

# CHAPTER 3:
# RSAT Network Model

## 3.1 RSAT Objective

RSAT's objective is to produce a deconflicted event schedule for all executing units during the desired SLTE planning horizon. Scheduling excess events for a given unit is heavily penalized and the cost of scheduling events increases as the planning horizon proceeds. Simply put, RSAT looks to schedule all required events as soon as possible while being constrained by event and unit relationships.

## 3.2 RSAT Limitations

RSAT is limited by the following aspects, which will be considered for future work to improve the RSAT program. First, while RSAT deconflicts the schedule according to time and event capacity, it does not account for the physical range location of a specified event. Second, as a result of the first limitation there is no direct connection between the range control scheduling office and RSAT to quickly determine range space availability. Third, planners must specifically assign execution groups for each event in the exercise. To give the program more flexibility, using unit-type tags (e.g., infantry, artillery, motor transport, etc.) and allowing the program to create execution groups based on the tags has the potential to create more optimal schedules.

## 3.3 RSAT Assumptions

RSAT assumes that the event-to-event, unit-to-unit, and unit-to-event relationships captured in the derived sets are adequate to produce a valid schedule. The event schedule assumes all units must be consolidated at conclusion of planning horizon for culminating MWX event. RSAT will be used during early and intermediate planning as an initial planning document to be refined by exercise planners. RSAT assumes physical range availability and TTECG staffing will be satisfied to support scheduling solution. Lastly, the cost and penalty functions used in RSAT were primarily designed to produce a schedule according to exercise planners' priorities.

## 3.4 RSAT Network Construction

RSAT is formulated as a minimum cost flow network because of the core routing problem being assessed, ease of visualization, and the potential to apply specialized algorithms for network flow problems. The nodes of the network are the days for the entire planning horizon and all exercise events. In the physical world, units will predominantly go to Camp Wilson, a small transit unit facility in Twentynine Palms, during their day node assignments to rest and prepare for the next event. Event nodes are introduced to the network according to the day they become available. As events are introduced they are connected to every day node after their initial start date until the end of the planning horizon; because of this feature the network resembles a funnel-like shape as it expands to the end of its directed arcs. Some events must be completed more than once in order for all units to fulfill their training requirements; we refer to these events as *iteration events* and we include multiple nodes for each such event. As illustrated in Figure 3.1, these duplicate nodes become available on the same day. In this case, event 1 has two iterations (A and B). The events in each iteration list use the same range facility or resources. As such, RSAT prevents units from executing more than one event in this list at a given time to prevent over-booking resources. Arcs connect day to event nodes and day to day nodes, but do not connect event to event. Arcs between events were not constructed in the RSAT to keep the network as simple as possible. A "virtual arc" connects the last day in the planning horizon to the first, in order to allow flow conservation to be enforced at these nodes. In the generalized network graph, the day nodes can be thought of as a holding area where units stay until all of the next event's prerequisites are satisfied. It is important to note that the scheduling solution may have a unit returning to and exiting from a day node on the same day; in this case the unit would most likely finish executing the first event and move to the second event immediately after conclusion, rather than going to Camp Wilson. The arc capacities are determined by the ingoing and outgoing nodes. Arcs flowing into and out of events have a capacity that reflects the exact size of the event being executed. Arcs flowing between days have an effectively infinite arc capacity. The infinite capacity is required for two reasons. First, once a unit is complete with all events in their curriculum, RSAT models them as flowing into the day nodes until the conclusion of the planning horizon. Second, the negative cost incurred from one day node to the next incentivizes the units to complete the necessary requirements before moving to the day nodes indefinitely. In Section 4.3, we will evaluate multiple arc cost functions to ensure adequate encouragement of early event execution.

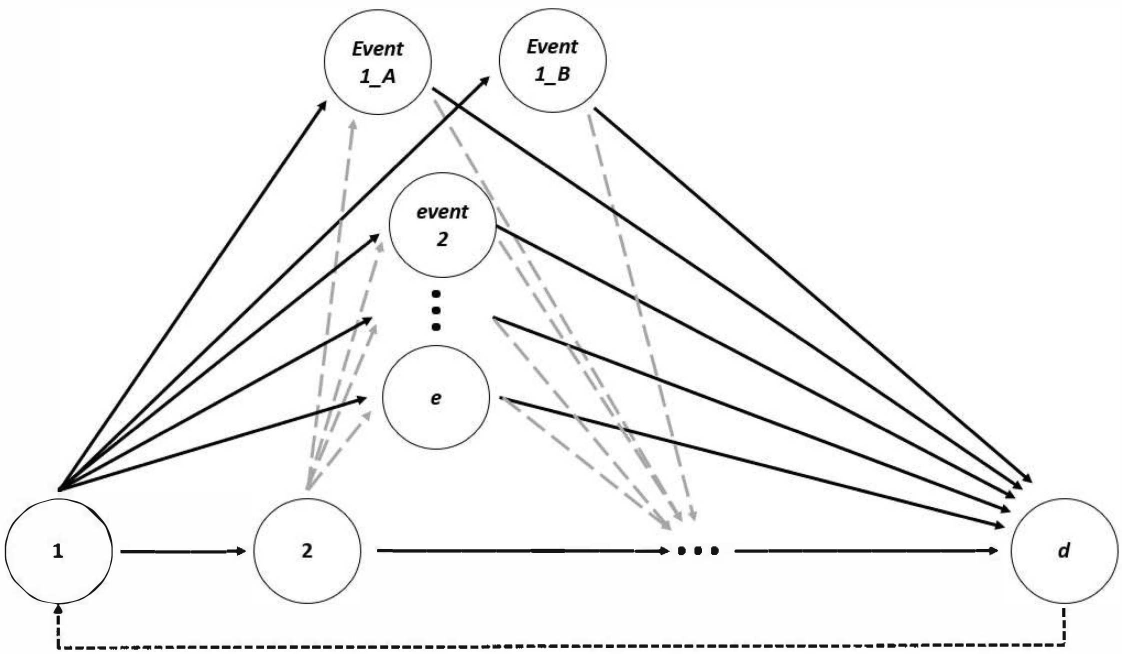

Figure 3.1. Generalized RSAT network graph.

## 3.5 RSAT Formulation

### Sets

| | | |
|---|---|---|
| $i, j \in N$ | | set of nodes |
| $d, d', d_{exit} \in D \subset N$ | | set of training days (1, 2,...) |
| $e, e' \in E = N \setminus D$ | | set of training events (R410_A, MDMX ,...) |
| $a \in A$ | | set of arcs |
| $u \in U$ | | set of units (A Company, B Company, ...) |
| $k \in K$ | | set of iteration event lists (1, 2, ...) |
| $s \in S$ | | set of event series (1, 2, ...) |
| $req\_train_u \subset E$ | $u \in U$ | required training events for unit $u$ |
| $pre\_train_e^u \subset E$ | $u \in U, e \in E$ | previous events $e'$ required for unit $u$ to complete event $e$ |
| $groups_e^u \subset U$ | $u \in U, e \in E$ | units $u'$ with which unit $u$ must be grouped in order to execute event $e$ |
| $iter\_events_k \subset E$ | $k \in K$ | events $e$ in iteration event list $k$; these are executed multiple times on same physical range and with similar range requirements |
| $series_s \subset E$ | $s \in S$ | events $e$ in series $s$; these are to be executed in an uninterrupted series |

### Parameters and Data

| | | |
|---|---|---|
| $cap_{ij}$ | $(i, j) \in A$ | capacity associated with arc $(i, j)$ |
| $cost_{ij}$ | $(i, j) \in A$ | cost associated with arc $(i, j)$ |
| $\beta$ | | positive penalty constant |
| $\alpha_{ij} = \beta cost_{ij}$ | $(i, j) \in A$ | penalty associated with arc $(i, j)$ |
| $first\_event_s$ | $s \in S$ | first event in series event list $s$ |
| $last\_event_s$ | $s \in S$ | last event in series event list $s$ |
| $series\_dur_s = \sum_{e \in series_s} dur_e$ | $s \in S$ | total duration for series event list $s$ |

#### Derived Sets

| | | |
|---|---|---|
| $iter_e \subset K$ | $e \in E$ | k index of event if in iteration events |
| $iter\_cap_k = \max_{d \in D, e \in iter\_event_k} cap_{d,e}$ | $k \in K$ | capacity of event in iteration events |

**Variables**

| | | |
|---|---|---|
| $x_{ij}^u$ | $u \in U, (i, j) \in A$ | flow of units through training schedule |
| $z_{ij}^u$ | $u \in U, (i, j) \in A$ | indicates assignment of non-required training events |

**Objective**

$$\min \sum_{u \in U} \sum_{(i,j) \in A} x_{i,j}^{u} cost_{i,j} + \alpha_{i,j} z_{i,j}^{u} \tag{3.1}$$

**Constraints**

$$\sum_{j:(i,j) \in A} x_{i,j}^{u} - \sum_{j:(j,i) \in A} x_{j,i}^{u} = 0, \forall u \in U, i \in N \tag{3.2}$$

$$\sum_{e \in E} x_{d,e}^{u} \leq 1, \forall d \in D, \forall u \in U \tag{3.3}$$

$$x_{d,e}^{u} = x_{e,d+dur_e}^{u}, \forall e \in E, d \in D, u \in U \tag{3.4}$$

$$\sum_{u \in U} x_{i,j}^{u} \leq cap_{i,j}, \forall (i,j) \in A \tag{3.5}$$

$$\sum_{d \in D} x_{d,e}^{u} = 1, \forall e \in req\_train_u, u \in U \tag{3.6}$$

$$x_{d,e}^{u} \leq \sum_{d' < d:(d',e') \in A} x_{d',e'}^{u}, \forall u \in U, d \in D, e \in E, e' \in pre\_train_e^{u} \tag{3.7}$$

$$x_{d,e}^{u} \leq x_{d,e}^{u'}, \forall u' \in group_e^{u}, u \in U, d \in D, e \in E \tag{3.8}$$

$$\sum_{u \in U} \sum_{e \in iter\_events_k} x_{de}^{u} \leq iter\_cap_k, \forall k \in K, d \in D \tag{3.9}$$

$$x^u_{d,first\_event_s} \leq x^u_{last\_event_s,d'=d+series\_dur_s}, \forall s \in S, u \in U, d \in D \quad (3.10)$$

$$x^u_{d,e} \leq z^u_{d,e}, \forall a \in A, u \in U, d \in D, e \notin req\_train_u \quad (3.11)$$

$$x^u_{i,j} \in \{0, 1\}, \forall u \in U, (i, j) \in A \quad (3.12)$$

$$z^u_{ij} \in \{0, 1\}, \forall u \in U, (i, j) \in A \quad (3.13)$$

## Objective Function

The objective function for RSAT minimizes the total cost of the flow throughout the network. The cost of each arc is dependent on three cost functions. Arcs flowing from day node to event node and event node to day nodes incur a cost of $cost_{ij}$. Arcs flowing from day to day have negative cost, i.e., $cost_{d,d'} < 0$. The objective value incurs a penalty term, $\alpha_{ij}$, when the indicator variable $z^u_{ij}$ is set to one. This occurs when any unit is assigned an non-required training event. Additionally, the penalty construction allows planners flexibility to incentivize other behaviors according to exercise priorities in other use cases.

## Constraints

- Constraint set (3.2) maintains flow conservation for the established network. Each commodity, in our case the individual units, must flow in and out of nodes at an equivalent rate.
- Constraint set (3.3) ensures that each unit is executing at most one event per day. This constraint set prevents units from being tasked to do multiple events on a given day.
- Constraint set (3.4) states that a unit must exit a given event on a specified day, $d'$, given the duration of the event.
- Constraint set (3.5) dictates that all flows across the arcs linking training events to days, days to events, and days to days must be maintained under the arc's capacity. This capacity constraint ensure that units are executing events at or below the size of

their specified execution group.

- Constraint set (3.6) ensures that each unit is completing the events in their required training list exactly once during the exercise.
- Constraint set (3.7) ensures that specified prerequisite training event is completed before the specified event.
- Constraint set (3.8) ensures units execute their required training events according to specified execution groups.
- Constraint set (3.9) ensures that events with multiple iterations (R400_A, R400_B, R400_C) are executed one at a time. The constraint set ensures that the number of executing units remain below the $iter\_cap_k = \max_{d \in D, e \in iter\_event_k} cap_{d,e}$.
- Constraint set (3.10) ensures events that are to be executed as series are completed in succession without interruption. This constraint set ensures the last event in a series must be completed by day $d' = d + series\_dur_{series_s}$.
- Constraint set (3.11) ensures that in the occurrence a non-required event is scheduled for a specified unit that the associated $z_{ij}^{u}$ variable is equivalent.
- Constraint set (3.12) ensures that the decision variable schedules at most one requirement for each unit on a given day.
- Constraint set (3.13) ensures that the decision variable indicates if a non-required training event is assigned to a given unit.

## 3.6 RSAT Summary and Small Application

The small application illustrates how RSAT is structured and performs. The network's nodes are events and days within the planning horizon. The network graph is directed and progresses through the planning horizon. The graph in Figure 3.2 only shows the arcs traveled by the identified units in the scheduling solution, while the generalized network and arc construction will be discussed in Chapter 4. In this example there are three units: two main units and one support unit. The units involved must satisfy the following requirements:

1. Both of the main units (Unit 1 and Unit 2) must complete R410 and R400.
2. R410 is a prerequisite for R400 for the main units.
3. R400 must be completed with the assistance of the support unit to satisfy the range requirements.
4. R410 is a single-day event.

5. R400 is a two-day event.

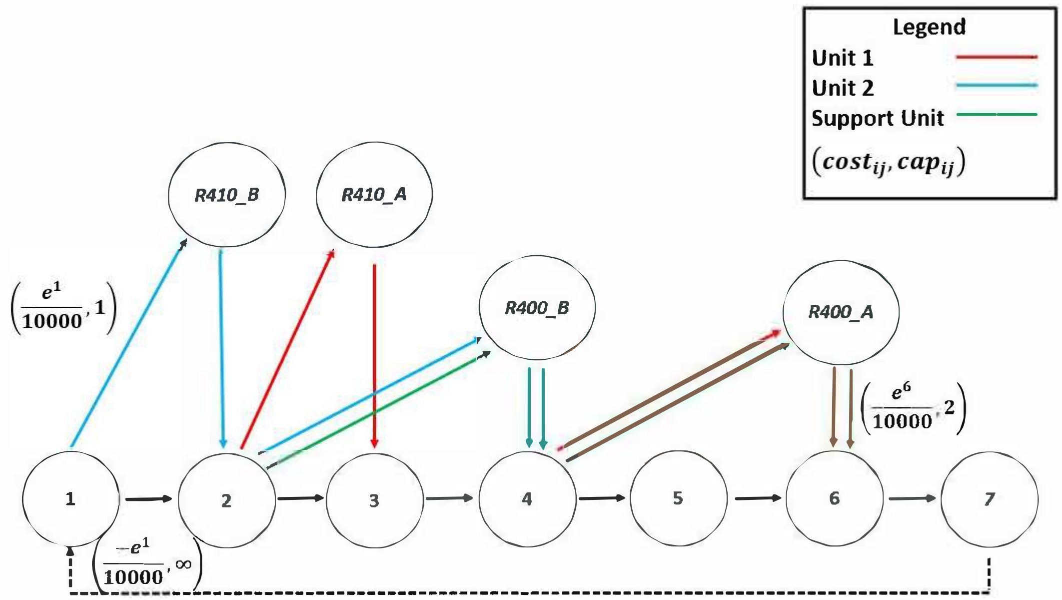


Figure 3.2. Small RSAT application network graph. Scheduling solution depicted with chosen unit flows in the following colors: Unit 1 in red, Unit 2 in blue, and Support Unit in green.

The costs on the network arcs depend on the starting and end nodes involved. The day to event and event to day arcs incur a cost of $e^d/10,000$. For the day to day nodes, a cost of $-e^d/10,000$ is incurred. The cost structure incentivizes the early completion of required events. The capacity of day to event and event to day arcs is limited to the size of the execution group. For example, in this small network the capacity going into R400 is two units. To deconflict events that occur on the same physical location, iteration events (R400_A, R400_B) are considered separate nodes. However, according to constraint set (3.9), only one of the two nodes can be occupied at one time. This allows the current network construction to prevent double booking range locations, but does require an extra pre-processing step to identify all iteration events during planning. The delineation of individual iteration events is helpful for planners and data input for reasons discussed in Chapter 4. In the network graph and scheduling solution, we observe that during the seven-day planning horizon both units are able to complete all requirements.

In the scheduling solution shown in Figure 3.3, Unit 2 (depicted in blue) completes the two events first. It is important to note that an equivalent solution would show Unit 1 finishing the events first. As the network is scaled for more units and events the same structure is retained. The expanded planning horizon increasingly incentives units to complete events early while regarding other constraints and event requirements.

| Training Day | R8 | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|---|
| RSAT Day | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| Unit 1 | | R410_A | | R400_A | | | |
| Unit 2 | R410_B | R400_B | | | | | |
| Support Unit | | R400_B | | R400_A | | | |

Figure 3.3. Sample network scheduling solution produced by RSAT.

# CHAPTER 4: Analysis and Performance

## 4.1 RSAT Implementation

The ultimate goal is to allow a single SLTE planner to have the ability to complete multiple RSAT iterations on their own individual machine. In this case, RSAT is run on a 64-bit Dell XPS 15 with an Intel Core i7-10750H CPU. The model is programmed in Python using Pyomo's open-source optimization programming language with Gurobi's LP solver. The RSAT model is composed of two Python files and two input .csv files. The first Python file creates the network structure for RSAT and establishes applicable relationships between units, events, and groups. The second Python file uses the Pyomo library to solve the MIP. The first input file contains general SLTE data including: day numbers, event names, event series, and iteration events. The second input file contains specific unit data and requirements including: unit names, event requirements, pre-training requirements for events, and execution groups for events. RSAT produces a single Excel file output. The output file contains a single deconflicted event schedule for every unit for the entire planning horizon.

## 4.2 Input Setup

The SLTE data is captured in four inputs: training days, events, series events, and iteration events. Training days are identified in two ways. First, using the planner's notation which identifies a number of "R#" days to delineate "receiving days" from training days. The second way is through the "RSAT day," which is a number from 1 to the end of the planning horizon. The events and event iterations are listed in the inputs along with their associated start date availability. The start date availability is a particularly important input because in most cases the scheduling solution will be optimized to have units start events as close as possible to this input data. Series input data informs the associated constraint set to ensure that units execute the identified series without being interrupted by other events or rest days. Lastly, iteration events are identified to ensure the same range facility or resources are not overbooked.

Data on specific units is captured in four inputs: units, unit requirements, unit required pre-training, and execution groups. The unit specific requirements serve as the guide post for the other two data entries. Unit required pre-training for an event is important as not all units have the same prerequisites to execute an event. In this formulation, the pre-training input is limited to a single event, therefore the planner must consider the unit's training curriculum to effectively stack events. Lastly, the planner must enter the execution group intended for each event. RSAT considers each execution group for events and ensure that once all units complete prerequisite events then the execution group can move to the specific training event together.

## 4.3 RSAT Performance and Validation

RSAT's performance is evaluated with quantitative and qualitative measures of effectiveness using a ITX scheduling template user input data. The network is comprised of: 51 units, 119 events (including event iterations), and 45 days. The inputs were derived from the template specific to the ITX 2-23 schedule provided by TTECG. The units and events that were not included in our analysis were primarily aviation-based units that have a higher frequency of last minute schedule changes stemming from aviation maintenance and equipment availability reasons. The network results in over 656,000 variables and 203,000 constraints that must be considered by the MIP solver. Three methods for assigning costs to arcs were compared. The solutions were evaluated quantitatively by assessing solution time and number of events assigned after the desired end date. Additionally, the schedules were assessed with TTECG planners' input to determine the overall usability of the final solutions.

### 4.3.1 Arc Cost Analysis

The arc costs were assigned using three different methods: constant, linear, and exponential. The arc cost functions followed:

| Function | Day-to-Event | Event-to-Day | Day-to-Day |
|---|---|---|---|
| Constant: | $cost_{d,e} = 1$ | $cost_{e,d} = 1$ | $cost_{d,d'} = -1$ |
| Linear: | $cost_{d,e} = d$ | $cost_{e,d} = d$ | $cost_{d,d'} = -d$ |
| Exponential: | $cost_{d,e} = \frac{e^d}{10,000}$ | $cost_{e,d} = \frac{e^d}{10,000}$ | $cost_{d,d'} = -\frac{e^d}{10,000}$ |

The functions incentivize early event completion to varying degrees. The desired scheduling solution output has no events after RSAT day 36. As determined by the application scope, all units must consolidate in the day nodes for execution of the culminating MWX event. According to the ITX template, RSAT day 36 is the start of MWX planning preceding the MWX event. The cost functions are assessed by the number of events assigned after the start of MWX, and qualitatively for overall usability. Table 4.1 summarizes the arc analysis conducted and the objective values for each scheduling solution. The solver uses an optimality-gap of 1e-4 as its termination criterion.

| Arc Cost Analysis | | | |
|---|---|---|---|
| Arc Cost Function | Time to Solve (sec) | Objective Function Value | Events Assigned after MWX |
| Constant | 551.5 | -8.22e12 | 133 |
| Linear | 22.9 | -1.89e14 | 0 |
| Exponential | 21.5 | -3.01e30 | 0 |

Table 4.1. RSAT arc cost analysis performance and solution metrics.

As a baseline, a constant cost function was used to assess RSAT's scheduling behavior. In the constant cost structure all arcs have equivalent cost magnitudes, with day-to-day arcs still having negative cost. The constant arc cost formulation motivates units to flow through day-to-day nodes but does so uniformly across the planning horizon. As a result, the scheduling solution produced using the constant arc cost structure has a significant number of events scheduled past the start of MWX, and hence is impractical. RSAT days 30–45 are shown in the scheduling solution; the 133 events shown in red are past the required MWX start date. Although all individual unit-to-unit, event-to-event, and unit-to-event relationships are established, the cost structure does not capture the ITX planners' desire to complete smaller, unit-specific events prior to the culminating MWX. Lastly, the computation time required to produce an optimal solution for the constant arc cost structure is significantly longer than the other two structures assessed.

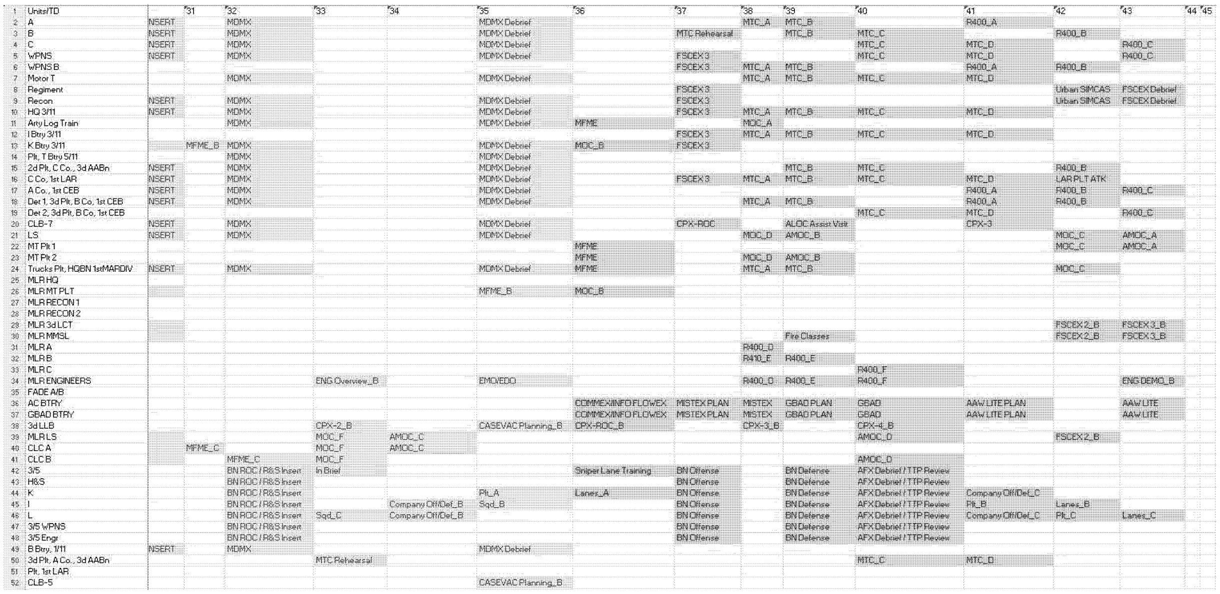

Figure 4.1. Scheduling solution, days 30–45, produced by constant arc cost structure. Events highlighted in red are scheduled past start of MWX.

Next, a linear cost function is implemented to incentivize the early execution of required events. It uses the in-coming or out-going day node associated with each arc to increase arc costs linearly as the planning horizon continues. For example, arc (R410, 10) will use the day 10 node to determine the associated cost and will be less than the cost associated with (R410, 11) arc. Additionally, the day-to-day nodes use the out-going day to determine the magnitude of the associated negative-cost arc. The biggest performance increase observed by using the linear function is that all events are assigned and executed prior to the start of MWX. Overall, we observe that RSAT brings events as close to their respective available start dates as possible. Under this cost structure, the ITX planners' intention for early completion of individual events is captured and produces a usable solution.

Figure 4.2. Scheduling solution, days 28–45, produced by linear arc cost structure. Events highlighted in red are scheduled past start of MWX.

Finally, an exponential cost function was implemented to further encourage the early execution of required events. Similar to the linear cost structure, the in-coming or out-going day node associated with a given arc determines the associated arc cost. As depicted in Figure 3.2, the cost is determined by the formula: $cost_{d,e} = e^d/10,000$, where $d$ is the day node in the arc and $e$ is the base of the natural logarithm. Similarly, the day-to-day arcs retain the $cost_{d,d'} = \frac{-e^d}{10{,}000}$. There is a negligible decrease in time to solve over the linear arc cost structure. Additionally, between the linear and exponential cost scheduling solutions, there are only two individual events that were assigned at different times. Both events were 'MTC Rehearsal', an event that requires no previous training and is intended to be completed prior to events in the MTC series. The two solutions are computationally the same because the events can be swapped and maintain the same objective value. From a solution usability perspective, this slight difference in the scheduling solutions does not have an impact, and both solutions are equally useful.

When comparing the exponential arc cost scheduling solution to the original ITX 2-23 scheduling template, there are eight events scheduled later than intended. Five of the eight events comprise the MTC event series: MTC Rehearsal, MTC_A, MTC_B, MTC_C, and MTC_D. These events, depicted in red in Figure 4.3, are scheduled two days later than intended but do not delay events after MTC_D. Similar to the differences between exponential and linear scheduling solutions, these differences do not negatively impact the usability of the RSAT product for ITX planners. Intuitively, it would seem that the exponential arc costs could motivate RSAT to more aggressively assign events earlier than the constant and linear cost structures. For all of the RSAT instances that we solved, the linear and exponential cost functions produced identical scheduling solutions.

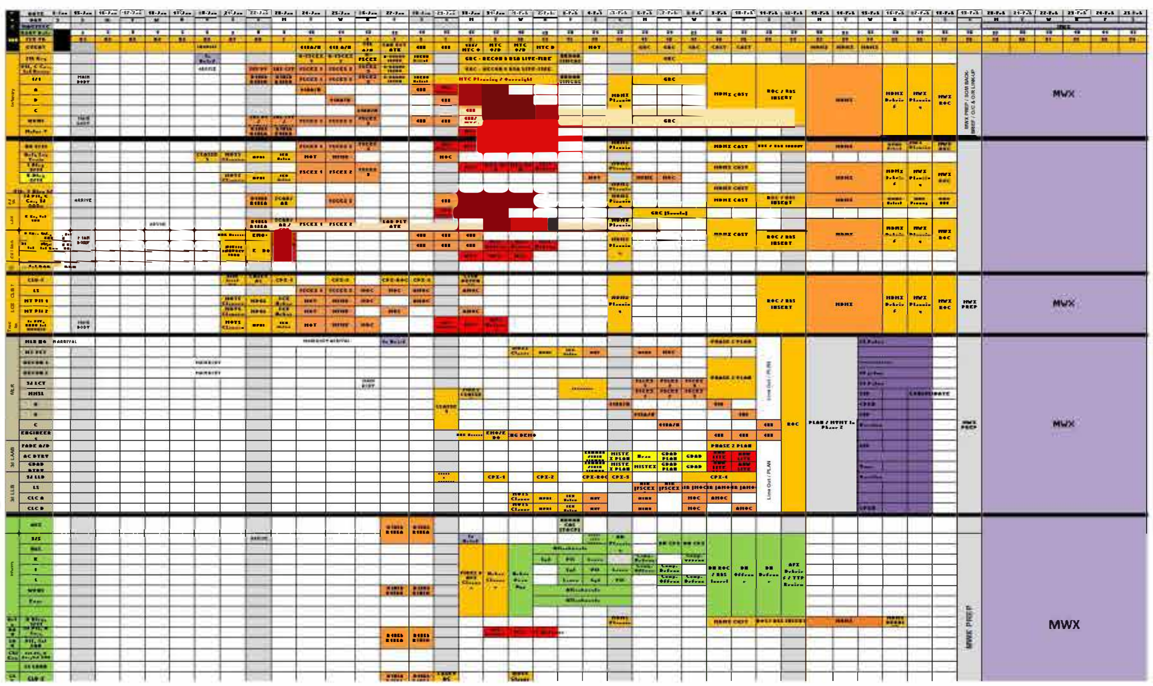


Figure 4.3. ITX Scheduling Template with incorrectly scheduled events highlighted in red. The identified events were assigned 1–2 days late by the exponential arc cost function.

## 4.4 Model Application and Solution Verification

As a full-scale model application and scheduling solution verification, a schedule for ITX 5-23 was produced and tuned using RSAT. The model application had a similar scope to the template scheduling solution used in the arc cost analysis. That is, the planning horizon for the application covered all events preceding MWX and included predominantly ground units. The scheduling solution considered 49 units and 94 events (includes event iterations) during a 45-day planning horizon. ITX 5-23 is a slightly smaller exercise than the schedule template assessed in Section 4.3. As a result, the network is slightly smaller with an associated 601,000 decision variables and 191,000 constraints. For reasons discussed in Section 4.3, the network used an exponential arc-cost structure and was solved in 19.5 seconds. Another important application consideration is that the data inputs for the 49 units took approximately five hours. Considering the time investment required to conduct data entry it is more beneficial to do early in the ITX planning cycle.

For ITX 5-23, RSAT was integrated during the second of three pre-exercise planning conferences. The timing of this integration is appropriate because most of the units are committed to the exercise, required events have been assigned, and range availability has largely been determined. During this application, it became apparent that the most efficient way to work with RSAT is to iterate over solutions and incrementally restrict the first available start date for events. By primarily focusing on the events that include the most units, in this case MDMX and MWX prep, and slowly restricting the event start dates, you quickly arrive at a workable initial solution. As the planning cycle progresses to the last planning conference, RSAT has the potential to save ITX planners a substantial amount of time if execution groups or availability for large events change. Instead of manually altering the entire schedule, small changes to the input documents and a series of RSAT iterations would allow planners to get back to a workable schedule quicker.

THIS PAGE INTENTIONALLY LEFT BLANK

# CHAPTER 5: Conclusions and Future Work

## 5.1 Conclusions

RSAT has the ability to produce scheduling solutions that minimize overall exercise length while satisfying unit-to-unit, unit-to-event, and event-to-event constraints. Additionally, RSAT gives exercise planners the ability to quickly produce initial planning schedules following changes to exercise requirements. Overall, RSAT has the potential to shorten exercise scheduling processes and limit manual planning errors.

Within the multicommodity network flow structure, RSAT must use an appropriate arc cost function to capture planners' intention for early completion of individual events. Functions that increase arc costs as the planning horizon progresses, such as a linear or exponential function, allow RSAT users to motivate event execution with varying aggressiveness. Lastly, by using RSAT's structure and objective function, planners can apply the same methodology to other scheduling problems.

## 5.2 Future Work

The RSAT model structure implemented in this thesis serves as a baseline capability and use case. An appropriate area for future work includes assessing RSAT's application in other scheduling scenarios. RSAT can be easily adapted for other military use cases such as military occupational speciality school schedules or regular military unit training schedules. These applications are inherently less insulated from schedule changes, and would benefit from researchers assessing scheduling persistence to minimize the effect changes have on scheduling solutions.

Within the use case assessed, planners would benefit from RSAT's functionality to include SLTE staffing and resource integration, range facility management integration, and less explicit data requirements. First, SLTE event capacity is limited by SLTE staff and TTECG resources. Staff members must be specifically qualified and designated to facilitate events. Integrating RSAT with the current staffing availability would allow schedulers to quickly assess staffing shortfalls and allocation. Similarly, integrating RSAT with current range availability from the planner's range facility management sources would allow for added deconfliction automation. Currently, this deconfliction must be completed by the scheduler after an RSAT solution is produced. Lastly, the current data input requirements for RSAT require many relationships to be explicitly provided within the derived data set. Another area of future work would involve mitigating the current need for explicit relationships by capturing them in more implicit guidance. For example, instead of explicitly defining each execution group, using unit-type tags (i.e., V27 - infantry, V27 Motor T - motor transport) and event-group requirements (i.e., 2x infantry, 1x motor transport) would reduce input requirements and has the potential to produce more efficient solutions. Ultimately, reducing the deconfliction tasks for the exercise planners and the amount of required RSAT inputs would greatly enhance usability and utility.

# List of References


Bellmore M, Bennington G, Lubore S (1971) A multivehicle tanker scheduling problem. *Transportation Science* 5(1), https://doi.org/10.1287/trsc.5.1.36.

DeWinter E (2012) *A Persistent Planning Model for EOD Training and Evaluation Unit Two*. Master's thesis, Department of Operations Research, Naval Postgraduate School, Monterey, CA, http://hdl.handle.net/10945/17353.

Leeds N (2023) ITX Schedule Data provided to the author via email, December 20.

Marine Air Ground Task Force Training Command, Marine Corps Air Ground Combat Center, Range, Training Area and Airspace (RTAA) Program (2022) *Standard Operation Procedures for Range, Training Areas and Airspace (SOP for RTAA)*. Author accessed via Range Facility Management Support System (RFMSS).

Slye R (2018) *Optimizing Training Event Schedules at Naval Air Station Fallon*. Masters thesis, Department of Operations Research, Naval Postgraduate School, Monterey, CA, https://hdl.handle.net/10945/58370.

Tactical Training and Exercise Control Group, Marine Air Ground Task Force Training Command, Marine Corps Air Ground Combat Center (2018) *Tactical Training and Exercise Control Group: Exercise Control Handbook*.

THIS PAGE INTENTIONALLY LEFT BLANK

# Initial Distribution List

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California