Proceedings of the Annual General Donald R. Keith Memorial Conference
West Point, New York, USA
April 24, 2025
A Regional Conference of the Society for Industrial and Systems Engineering

Less Drag, More Drive: Automating the Special Warfare Pipeline

Lauren Arnold, Caroline Guiler, Lauren Pettit, Ben Kallemyn, and Brian Lemay

Department of Operations Research, United States Air Force Academy, USAF Academy, Colorado 80840

Corresponding author's Email: Brian.Lemay@umich.edu

Author Note: This project was conducted by a team of cadets and their faculty advisors from the United States Air Force Academy, specializing in operations research and dedicated to improving the efficiency of Special Warfare training. We extend our sincere gratitude to our project sponsors at the Special Warfare Training Wing (SWTW) and to our capstone advisors, Lt Col Brian Lemay and Dr. Kallemyn, whose guidance and expertise have been invaluable to the success of this project.
The views expressed herein are those of the authors and do not reflect the official positions of the United States Air Force Academy, the Department of the Air Force, or the Department of Defense.

Abstract: The Air Force's Special Warfare training program prepares combat-ready Airmen through a rigorous multi-course curriculum, typically spanning an average of three years. However, inefficiencies such as manual scheduling, resource constraints, and student attrition significantly increase the weekly man hours required to manage and update the training schedule. To address these challenges, we developed a scalable, Python-based tool that automates the scheduling process, reducing manual effort and freeing up scheduler time. The tool incorporates course prerequisites, capacity limits, instructor availability, and real-world disruptions such as injuries and retakes. This adaptive solution streamlines scheduling, reduces administrative workload, and provides decision-makers with a data-driven framework to manage pipeline changes efficiently. Ultimately, it enhances training throughput and accelerates the production of mission-ready Airmen.

Keywords: Special Warfare, Training Pipeline, Scheduling, Optimization

1. Introduction

In recent years, optimizing the training pipeline for the Air Force's Special Warfare Training Wing (SWTW), located in San Antonio, Texas, has become a priority due to its impact on operational readiness and mission capability. At the outset of this project, our goal was to reduce Student Not in Training (SNIT) days – a metric representing student downtime between courses due to scheduling gaps, course failures, injuries, or other disruptions. However, after numerous discussions with SWTW leadership and schedulers, we learned that their top priority is reducing the number of man hours spent on the scheduling process itself.

Currently, up to 12 personnel spend around 30 hours per week manually scheduling students across dozens of courses. This time-intensive effort is necessary to keep the pipeline running smoothly, but it diverts skilled manpower away from other mission-critical tasks. Each time a student drops from a course due to injury, failure, or voluntary withdrawal, the schedule must be reworked manually – making it difficult to adjust quickly and accurately. To address these challenges, we developed an automated scheduling tool that accomplishes two objectives: Reduces SNIT days by improving scheduling accuracy, minimizing delays caused by human error, and leveraging more organized, centralized data, and reduces man hours spent scheduling, allowing personnel to reallocate their time and effort toward higher-value operational needs.

Special Warfare candidates train for elite roles such as Pararescue (PJ), Combat Control (CCT), Special Reconnaissance (SR), and Tactical Air Control Party (TACP). Each candidate must complete a complex, multi-year sequence of over 20 courses, managed by various DoD units at multiple locations. Manual scheduling – relying on spreadsheets, static calendars, and informal communication – is inefficient and brittle in the face of dynamic changes.

Our Python-based tool automates course assignments using real-time inputs, enabling rapid adjustments when students drop, fail, or return to the pipeline. This paper presents the design, methodology, and impact of our tool, developed to streamline the Special Warfare training pipeline through a dynamic, constraint-based scheduling framework.

1.1 Project Background

The Special Warfare training pipeline consists of foundational, intermediate, and advanced phases, guiding students through a multi-year progression of over 20 specialized courses. Students often train alongside members of other military branches, attend Army- or Navy-run schools, and must meet strict prerequisites before advancing. These constraints make the pipeline long, complex, and difficult to manage, where even minor misalignments between course availability and student readiness can create significant delays.

Initially, our team set out to reduce SNIT – downtime when students are not actively progressing. While some SNIT is unavoidable due to injury or course failure, we learned through interviews and a site visit to SWTW at Joint Base San Antonio that a major root cause lies in the manual scheduling process itself. Scheduling is performed weekly and relies heavily on spreadsheets, email updates, and disconnected systems. The responsibility currently falls on three full-time personnel working 40 hours a week, with support from a supervisor who regularly works overtime due to the high volume of scheduling needs. This creates a significant administrative burden, often pulling leadership into scheduling simply to meet weekly demands.

Training data from FY21 to FY23 reveal consistent inefficiencies. For example, PJ students averaged a pipeline duration of 1,185 days – 399 days longer than the ideal 786-day timeline. CCT and TACP pipelines showed similar delays, with students frequently waiting multiple weeks between training events despite being fully qualified to progress. These delays cascade downstream, reducing throughput and impacting force generation across the Air Force Special Warfare community.

To better understand these inefficiencies, we conducted interviews with instructors and schedulers, analyzed five years of anonymized student training records, and examined trends in course capacity, dropout rates, and scheduling gaps. One key insight was the influence of Training Request Quota Identifier (TRQI) codes, which prioritize students based on their specialty and unit needs. While these are intended to help match students with time-sensitive mission requirements, current manual scheduling practices often fail to align TRQI priorities with available course slots.

Recognizing this, our team developed an automated scheduling tool that integrates course offerings, student readiness, TRQI priority, and prerequisite structures. While the tool still reduces SNIT by improving scheduling accuracy and organization, its primary impact is reducing the weekly man hours required to schedule students, making the process faster, more efficient, and less prone to error. The remainder of this paper details our modeling approach, tool architecture, and testing results.

1.2 Related Works

Our work draws upon a range of existing literature in military manpower planning, scheduling optimization, and operations research modeling. Several key methodologies influenced our approach:

We reviewed process modeling and bottleneck analysis frameworks used in the AFIT Pilot Training Pipeline study (Colbath, 2020). That study applied Little's Law and queuing theory to analyze bottlenecks in pilot production, a concept we mirrored by identifying SNIT-causing bottlenecks in Special Warfare.

Maximal Flow Network Modeling (Troutt et al., 2001) demonstrated how to identify critical paths and capacity-constrained nodes in sequential production systems. These principles directly applied to our course flow diagrams and scheduling simulation logic, where course dependencies and limited seats form the network through which students flow.

In terms of data-driven scheduling tools, RAND's Tech School Pipeline study (Diener et al., 2004) was instrumental. That research emphasized real-time data collection, dashboard-based monitoring, and automated alerting systems—all of which shaped our vision for an interactive interface that integrates with existing training systems.

Finally, the airline industry's use of multi-input scheduling tools was highly relevant. In particular, Continental Airlines' pilot training optimization framework (Yu et al., 2004) integrated inputs from HR, training managers, and availability databases to dynamically assign training slots. Their scheduling logic inspired our template-based model and influenced our emphasis on making the tool accessible to non-programmers.

Together, these works established the foundation for our decision to develop a Python-based, Excel-integrated scheduling engine that could be used by SWTW staff to make real-time, optimized course assignments with minimal manual intervention.

2. Data and Methodology

To identify inefficiencies and build an effective scheduling model, we first collected and analyzed five years of anonymized student progression and course scheduling data provided by SWTW. The raw data consisted of multi-tab Excel files that logged student names, TRQI identifiers, course attempts (including pass/fail outcomes), class rosters, and training dates. Each training cycle was formatted differently and required significant cleaning. We filtered out outdated pipeline phases and standardized course naming conventions, failure codes, and AFSC labels. All data preprocessing was conducted in Python using Google Colab to ensure reproducibility.

Through our analysis, we found that most SNIT periods were not due to injury or course failure but rather occurred between completed courses—where students were ready to continue but were left unassigned. While some SNIT is unavoidable, the number of preventable idle days highlighted opportunities to optimize scheduling.

Each row in Figure 1 illustrates the training sequence for the AFSC listed on the far left, showing required courses and their dependencies. While some pathways are linear, others allow flexibility in course order that could be leveraged to reduce idle time. For example, a student may complete Survival Training before or after a core specialty block, depending on seat availability. The top line in each course is the course name. The second line refers to the number of Training Days required for each course for completion. The third line in each course is the number of weeks for how long each course is expected to take.

Figure 1. SWTW Pipeline Overview

In collaboration with SWTW instructors and schedulers, we identified that TRQI prioritization plays a critical role in how students should be placed. Students assigned to high-priority TRQIs (such as deployment-bound units or understrength specialties) need to progress as quickly as possible. Tables 1 and 2 summarize the AFSC-specific course sequencing requirements and TRQI prioritization hierarchy. Our tool uses these structures to inform decisions. The courses presented in the tables are: Pre-Dive (PD), Special Warfare Assessment and Selection (SWAS), Dive, Airborne (AB), Survival, Evasion, Resistance and Escape (SERE), Military Free Fall (MFF), STO Apprentice (STO AO), Modernized Pararescue Provider Program (MP3), Air Traffic Control/ Combat Control Apprentice (ATC/CCA), Special Reconnaissance Apprentice Course (SRAC), Pararescue Apprentice Course (PJU), CRO Operations (CRO Ops), and CRO Course.

Table 1. SWTW Course Priorities

AFSC	Priority 1	Priority 2	Priority 3	Priority 4	Priority 5	Priority 6	Priority 7
STO	PD	Dive	AB	SERE	MFF	STO AO	STO Course
CCT	SWAS	PD	Dive	AB	SERE	MFF	ATC/CCA
SR	SWAS	PD	Dive	AB	MFF	SERE	SRAC
PJ	SWAS	PD	Dive	AB	MFF	MP3	PJU
CRO	PD	Dive	AB	SERE	MFF	CRO OPS	CRO Course

ISBN: 97819384969-9-8

Table 2. SWTW TRQI Priorities

TRQI Code	Description	Scheduling Priority
CNN0 & RR10	ANG/USAFR	1
AM11	Officer Prior Service	2
AM10	Officer Non-Prior Service	3
AJ3J	Enlisted Cross-Trainee	4
AJ1K	Enlisted Prior Service	5
AJ30	Enlisted Non-Prior Service	6

We developed a scheduling model that incorporates the following logic and rules:

- Course prerequisites: A student cannot attend a course unless all required prior courses are completed.
- Capacity constraints: Each course has a limited number of available seats per offering.
- Course dates: Some courses run year-round, others only once or twice per year.
- Instructor availability: Courses with the same instructor cannot overlap.
- TRQI priority: Ensures high-priority students are placed first in time-sensitive courses.

By structuring course and student data into a unified scheduling engine, we were able to simulate pipeline flow under different policy assumptions and scheduling priorities.

2.1 Data

The dataset, provided by SWTW, included five years of anonymized training records across all Special Warfare AFSCs. After resolving formatting inconsistencies and cleaning duplicate or conflicting entries, we created a consistent dataset with student IDs, course histories, outcomes, and training timelines. To focus on relevant insights, we excluded data from outdated pipeline configurations and courses that had been replaced or removed in recent years. The cleaned dataset included over 1,200 student-course combinations across multiple AFSCs. We used this to calculate average time-between-courses, failure rates, course capacity utilization, and the frequency and length of SNIT periods.

After preprocessing, we created structured input tables for the scheduling model, mapping each course to its prerequisite list, available dates, capacity, and AFSC eligibility. TRQI prioritization was also included to simulate the impact of demand urgency. This data-driven structure allowed us to test different policy levers – such as priority reordering, prerequisite waivers, or course frequency adjustments – and observe their downstream effects.

2.2 Tool Development

We developed a scheduling tool in Python that automates the placement of Special Warfare students into their required courses using structured Excel templates. Designed with usability in mind, the tool is intended for SWTW staff with limited programming experience. Users input course offerings and student rosters via editable Excel files, upload them into the Google Colab environment, and generate a complete, optimized schedule with minimal manual effort.

The student and course templates used in the process are shown in Figure 2 and Figure 3, respectively. These templates serve as the primary interface for schedulers, allowing them to easily modify training data without interacting directly with the code. Once uploaded, the tool processes the inputs and produces a clean, readable output schedule that can be reviewed, adjusted, or re-run with updated parameters as needed.

The tool's logic follows a step-by-step decision process to build each student's schedule. First, it looks at the student's career path and pulls a prioritized list of required courses based on their TRQI which reflects how urgently they are needed in the field. Then, for each course, the tool checks whether the student has met all prerequisites, finds the earliest available class, and places the student into it – as long as the course isn't already full. If it is full, the tool automatically searches for the next available class that meets all requirements. This prevents students from being scheduled into overlapping courses or ones they aren't ready for. This scheduling flow is repeated for each student in the roster. The code starts by looping through each student and tracking their entry date, SNIT days, and assigned courses. It then walks through their course list and schedules each course one at a time. If the student has a gap between courses, those days are added to the SNIT calculation. If a course can't be scheduled, it's flagged in the output, so the scheduler is aware.

The output of the tool is an Excel file that provides a clear and structured overview of each student's customized schedule. As shown in Figure 4, each row represents one student and includes their name, AFSC, TRQI, entry date, full list of assigned courses, and key performance metrics such as total days in the pipeline, days spent in courses, and calculated SNIT days. For each course, the output lists the course name along with its corresponding start and end dates. This format allows schedulers to quickly view and validate each student's training path, identify scheduling gaps, and easily make comparisons across students and career fields. The structure is intuitive and designed for users familiar with Excel, providing both transparency and usability without requiring any coding knowledge.

	A	B	C	D	E
1	Student Name	Student ID	AFSC	TRQI	Entry Date
2	Doe, John	1234567890	19XX	XX##	1/1/2025
3					
4					

Figure 2. Student Excel Template

	A	B	C	D	E	F	G
1	Course Type	Total # of Offerings	Section #	Start Date	End Date	Max Students	Min Students
2	PreDive	6	1	13-Jan-25	14-Feb-25	60	4
3	PreDive	6	2	24-Mar-25	25-Apr-25	60	4
4	PreDive	6	3	19-May-25	25-Jun-25	60	4
5	PreDive	6	4	28-Jul-25	28-Aug-25	60	4
6	PreDive	6	5	23-Sep-25	25-Oct-25	60	4
7	PreDive	6	6	21-Oct-25	15-Nov-25	60	4

Figure 3. Course Excel Template

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
1	Student Name	AFSC	TRQI	Entry Date	Courses	Days in Pipeline	Days in Courses	SNIT	Course 1	Start Date 1	End Date 1	Course 2	Start Date 2	End Date 2	Course 3	Start Date 3	End Date 3
314	Student 1	CRO	AM10	01/13/2025	SWAS, Airborne, SV-97A, SV-98A, SV-84AF, SV-94C, CRO Ops	320	218	96	SWAS	03/10/2025	05/16/2025	Airborne	05/27/2025	06/13/2025	SV-97A	07/09/2025	07/18/2025
315	Student 2	CRO	AM10	01/13/2025	SWAS, Airborne, SV-97A, SV-98A, SV-84AF, SV-94C, CRO Apprentice	320	218	96	SWAS	03/10/2025	05/16/2025	Airborne	05/27/2025	06/13/2025	SV-97A	07/09/2025	07/18/2025
316	Student 3	CRO	AM11	01/13/2025	SWAS, Airborne, SV-97A, SV-98A, SV-84AF, SV-94C, CRO Apprentice	320	218	96	SWAS	03/10/2025	05/16/2025	Airborne	05/27/2025	06/13/2025	SV-97A	07/09/2025	07/18/2025
317	Student 4	CRO	AM10	01/13/2025	SWAS, Airborne, SV-97A, SV-98A, SV-84AF, SV-94C, CRO Apprentice	320	218	96	SWAS	03/10/2025	05/16/2025	Airborne	05/27/2025	06/13/2025	SV-97A	07/09/2025	07/18/2025
318	Student 5	CRO	RR10	02/14/2025	SWAS, Airborne, SV-97A, SV-98A, SV-84AF, SV-94C, CRO Apprentice	297	218	73	SWAS	03/10/2025	05/16/2025	Airborne	05/27/2025	06/13/2025	SV-97A	07/09/2025	07/18/2025

Figure 4. Sample Tool Output in Excel

3. Results

Our tool demonstrates a significant impact on both pipeline efficiency and administrative workload. Using a test case of 397 students scheduled across 275 courses, we observed that the automated scheduling process reduces unnecessary SNIT by an average of 40 days per student. This translates to nearly 15,880 training days recovered annually across all AFSCs within the Special Warfare Training Wing. Among the different career fields, TACPO had the lowest SNIT, averaging only 19 days of downtime across the full pipeline, indicating how optimized scheduling can vary by specialty.

In addition to improving student flow, the tool nearly halves the man hours required for weekly scheduling. Previously, three full-time personnel spent upwards of 30 hours each per week managing schedules. With the implementation of the tool, that number dropped to approximately 20 hours per person, representing a more efficient use of skilled manpower.

ISBN: 97819384969-9-8

Furthermore, our analysis revealed that restructuring the order of key courses – specifically moving Pre-Dive, Dive, and MFF to the end of the pipeline – can reduce SNIT by at least 131 days per student, regardless of career field. These findings highlight the tool's value not only in automating routine scheduling tasks but also in informing broader strategic decisions to enhance pipeline flow and readiness.

4. Conclusion

This project demonstrates that automating the Special Warfare training pipeline's scheduling process can significantly improve efficiency, reduce student downtime, and ease the administrative burden on SWTW personnel. By replacing a manual system with a Python-based scheduling tool, we recovered an average of 40 SNIT days per student and nearly halved the number of man hours spent on scheduling each week. The tool not only enables faster, more accurate student placement but also provides data-driven insights for improving pipeline design – such as the substantial SNIT reduction achieved by repositioning Pre-Dive, Dive, and MFF to later in the training sequence. One of the most impactful features currently in development is the tool's ability to dynamically respond to student removal and return due to medical or other pipeline disruptions. Users will be able to upload an updated Excel file with a student's status code, removal date, and expected return date. The tool will automatically drop all courses after the removal date and reschedule those courses after the student's return, while simultaneously filling the gaps left behind with other students, ensuring no time is wasted in the pipeline. This function is expected to be fully implemented within the next two weeks. Additionally, the tool serves as a powerful "what-if" engine for decision-makers. By adjusting variables such as the number of course offerings, seat capacities, or minimum student counts, users can instantly see how those changes ripple through the schedule—affecting specific courses, career fields, and overall throughput. This capability empowers SWTW staff to make informed, strategic decisions grounded in real-time feedback from the system. Looking ahead, future developments may include a user-friendly interface, real-time data integration, and predictive modeling to estimate injury or failure risks. We encourage future users and developers to continue building on this foundation to adapt the tool for broader training environments and continuously improve pipeline efficiency.

5. References

Colbath, D. J. (2020). A model to analyze the capacity of pilot training production. Air Force Institute of Technology.

Diener, D., Peltz, E., Lackey, A., Blake, D. J., & Vaidyanathan, K. (2004). Value recovery from the reverse logistics pipeline. RAND Corporation.

Troutt, M. D., White, G. P., & Tadisina, S. K. (2001). Maximal flow network modelling of production bottleneck problems. *Journal of the Operational Research Society*, 52(2), 182-187.

Yu, G., Pachon, J., Thengvall, B., Chandler, D., & Wilson, A. (2004). Optimizing pilot planning and training for Continental Airlines. *Interfaces*, 34(4), 253-264.