Proceedings of the Annual General Donald R. Keith Memorial Conference
West Point, New York, USA
May 2, 2024
*A Regional Conference of the Society for Industrial and Systems Engineering*

# Streamlined Air Force Flight Scheduling: Implementing an Automated Method for an Instructor Pilot Assignment Problem

**Elizabeth Deards[1], Ezra Paul[1], Grace Sublette[1], Noah Wentworth[1], Brian Lemay[1], John Miller[1], and Trevor Doll[2]**

[1]Operations Research Program,
United States Air Force Academy,
Colorado Springs, CO 80841

[2]47th Flying Training Wing,
Laughlin Air Force Base

Corresponding author's Email: BLemay@umich.edu

**Author Note:** The cadet authors are all first-class cadets at the U.S. Air Force Academy, partnering with the 47th Flying Training Wing (FTW) at Laughlin Air Force Base. The views expressed herein are those of the authors and do not reflect the position of the United States Air Force Academy, the Department of the Air Force, or the Department of Defense.

**Abstract:** Each day at Laughlin Air Force Base (AFB), pilots conduct approximately 140 training flights. Creating a one-day schedule of flights for both instructors and students takes 12 hours of manual work by a team of two instructor pilots who also fly regularly. To reduce this scheduling workload while efficiently satisfying training syllabus requirements, we developed an automated scheduling algorithm capable of generating a daily flying schedule for 140 flights in approximately 5 minutes while meeting training syllabus requirements. We estimate that our algorithm can save schedulers up to 40 work hours per week. Furthermore, we designed a user-friendly interface to ensure the program's continued use despite the regular turnover of personnel observed in military organizations. Our work can be leveraged at other Air Force pilot training bases, saving similar time and resources and refocusing pilots' efforts towards flying.

*Keywords:* Flight Scheduling, Heuristic, Sustainability

## 1. Introduction

The 47th Flying Training Wing is one of three organizations responsible for Undergraduate Pilot Training (UPT) in the United States Air Force. They have two squadrons of roughly 100 aircraft dedicated to their mission of training and graduating 375 total pilots each year. To meet this mission, scheduling instructor and student pilots is a crucial component of the operation. Currently, two Instructor Pilots (IPs) in the squadron are scheduling up to 140 flights daily for approximately 150 current IPs by hand, removing them from their operational tasks. This requires 12 total hours of work by the schedulers, and the complexity of this task arises from the various instructor pilot excusals, responsibilities, and constraints on their time to fly, making scheduling IPs more difficult than scheduling students. Due to this complexity and lack of automation, the current upper bound on schedulable IPs per day is estimated to be around 94, an estimate generated by averaging the number of IPs that were available and scheduled each day over the months of data provided. Like many active-duty military organizations, the scheduling office experiences frequent turnover of its personnel, so the process must be easily understandable and implementable to ensure sustained use of an algorithm or process. Our project aims to automate scheduling IPs to save schedulers time and bring the average number of flights flown per day towards the 140-flight objective.

### 1.1 Problem Statement

Creating a daily training flight schedule at Laughlin AFB is challenging and time consuming, making it difficult to generate schedules that meet the organizational goal of 140 flights per day. Based on these problems we seek to answer the following research question: What is a sustainable IP scheduling algorithm that takes less time to complete and generates 140 flights a day, while ensuring that IPs are scheduled in a consistent and balanced manner?

## 1.2 Related Work

This problem consists of developing a feasible schedule for IPs subject to the constraints of their flying availability. For simplicity and to target the requests of the training base, we consider students to be always available unless flying. Upon conducting research on this problem, we found related work that leveraged genetic algorithms. These heuristic-based algorithms leverage past solutions to improve and work towards optimality. Norman and Bean (1999) developed a genetic algorithm that could assign 270-360 jobs, a similar scope to our problem. Montana et al. (1998) and Birjandi and Mousavi (2019) extended this model by allowing their constraints to change dynamically or be ill-defined, respectively. For our algorithm, flights and their times may change due to weather, or an IPs schedule may be unpredictable. In our problem, multiple IPs may be available, but some are a better fit than others based on their availability to fly more constrained flights later in the day. These constraint formulations allow schedule resiliency, and Montana et al. (1998) and Ding et al. (2021) demonstrated the applicability by extending their solution to a military aircrew-scheduling problem.

While the genetic algorithm is a well-researched heuristic for this problem, other authors offered alternative heuristic approaches. Geiger (2017) focuses on the efficiency of the heuristic model approach in solution generation, especially if there is a tight deadline, such as scheduling flights for the next day. Rittri, Allerbo, and Carmenta (2000) additionally leverage the solution time to generate schedules for the Swedish Air Force that take less time than the manual scheduling process and contain similar constraints to our problem. Lamas and Demeulemeester (2014) focus on generating a robust schedule with the branch and bound method that can quickly re-optimize should there be any unexpected changes, such as an instructor's last-minute personal schedule adjustments. Alternatively, Gershkoff (1989) extended the heuristic linear programming model to the flight-scheduling problem but was less successful than the other authors in finding a solution in his search space. Erdimer (2014) focuses on aiding schedulers in a fighter squadron by leveraging Microsoft Excel and a greedy randomized search procedure to generate a feasible solution in much less time than their manual process that is easily implementable in the squadron. Erdimer (2014) was one of the most applicable papers methodologically towards our development as these constraints similarly drove our algorithm.

Roslof et al (2001) also targeted a near optimal solution to the scheduling problem using a mixed integer linear program. The framework that they implemented allows a scheduler to fix in place jobs that must occur at certain times and flexibly allocate the remaining jobs around these selections. Aringhieri et al (2014) used scheduling operating rooms in a hospital to demonstrate that seeking an optimal solution using a linear program for a problem of this size is infeasible. They circumnavigated this problem using a two-level strategy that heuristically searched for a near-optimal solution. Graves et al. (1993) extend the linear programming relaxation to find suboptimal solutions for flight schedules. While this model was less successful for larger problems, the motivation in this paper, combined with the more advanced techniques in the recent papers of Roslof et al. (2001) and Aringhieri et al. (2014), demonstrate the applicability and feasibility of this approach.

Our solution differs from the above papers in that the solution space is large, and the constraints of the flying organization discussed later drive us to avoid optimality in favor of feasibility. Our processing capabilities are also constrained as the users are limited to a personal laptop computer. In turn, generating a solution that can overcome these challenges will contribute to the existing literature and differ from what is outlined above.

## 2. Methodology

To create the schedule, we acquired data from the flying squadron regarding the flights for the day and their instructor pilots' availability. Data collection primarily came from two sources: the IP's schedules, and the maintenance report for the day's flights. The IP schedules can be downloaded from the platform Graduate Training Integration Management System (GTIMS), which is the platform that the squadrons use to input any absences, the schedule itself, and any upcoming time-off requests. We downloaded the personal schedule of the IPs and any upcoming time-off that they have requested from GTIMS to discern which IPs are available to fly which flights and on what days. The second source, the maintenance report, can also be downloaded from GTIMS. This allowed us to understand what types of flights are to be flown that day and how many of each type will occur. It also details when each flight will occur and the length of time that flights will take. This report guides which IPs will be available for which flights.

The IP time-off requests show each requested absence and the time stamp associated with that request. This is the same for their personal schedules and allows us to see when an IP was and was not available. There are about 100 absence requests per day that range from requesting one hour of absence to requesting an excusal for the entire day. The personal schedule outlines every other reason that a pilot may not be able to fly, including doctor's appointments, to ensure they are capable of flying and other duties besides flying, such as working in the tower as a safety officer. When our algorithm checks both the absence request file and the personal schedule file, it is able to determine a pilot's availability.

The maintenance data details when the 140 flights will take place that day and what type of flight is to take place at each time. There are two different types of flights that are scheduled: out-and-backs and locals. An out-and-back flight is where an instructor pilot and student will fly to another airport, land and stop for gas, and then return to Laughlin Air Force Base. A local flight is much shorter as it does not include this landing and refueling. The distinction between these two types affects the length of time that a pilot will need to be scheduled for and determines if that flight is feasible based on their availability. Additionally, some pilots are scheduled for two flights a day if their schedule permits in what is called a double turn. Out of the 140 flights per day, there are an average of 22 out-and-backs and the remaining flights are local flights. In regards to the double turn assignments, the number of pilots with this assignment changes significantly from day to day depending on the time-off requests and personal schedules.

Some initial exploratory analysis of the 47th FTW current flying operations provided us with some preliminary statistics and figures that detail the current state of operations. Looking at the graph below, we can see that there are considerably less flights on average flown on the weekends than on the weekdays. During the week the wing averaged around 120 flights, and is where the bulk of operations takes place.

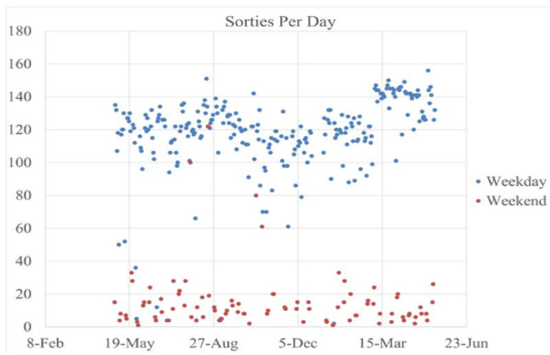


Figure 1. Flights Scheduled per Day

The weekday average of approximately 120 flights is 14.29% short of the stated goal of 140 flights. Our process aimed to increase this number to improve the overall efficiency of operations. Figure 1 also shows that flights scheduled on weekends are consistently lower than weekdays. Upon further investigation, we discovered that the weekend flights were not syllabus sorties but are outlier flights returning or leaving the base and should not be accounted for in the developed scheduling process. In turn, we are not concerned about this discrepancy. Table 1 highlights some important statistics regarding the current operational status of pilot training.

Table 1. Summary Statistics for GTIMS Data

| | |
|---|---|
| Average Number of Flights Scheduled on Weekdays | 117.451 flights |
| Maximum Flights Scheduled in a Day | 156 flights |
| Minimum Flights Scheduled in a Day | 1 flight |
| Average Duration of a Flights Scheduled | 1.59 hours |
| Standard Deviation of Scheduled Flights | 31.48 flights |

These statistics give us an understanding of the current state of operations and can be used to measure any improvements in the automated scheduling process compared to their manual process. The different phases of flight helped us to determine if one phase of flight is more constrained when scheduling than others and will allow for further analysis after our algorithm is implemented to see how the state of operations changes. This was our baseline to decide if our algorithm improved the efficiency of their flying operations.

Based on Geiger (2017) and Rittri, Allerbo, and Carmenta (2000), along with our initial assessment, we created a schedule through a heuristic-based programming algorithm that leverages a rules-based logical flow to schedule pilots. Compared to searching for an optimal solution, we used a heuristic approach to ensure access to the tool, ease of use, and transparency of the scheduling process. If our algorithm is unable to find a solution to the entire schedule, it will still produce a partial schedule, allowing schedulers to manually address any problem that may arise. Additionally, the research indicates that solving to optimality will require hours to output a solution, which would constrain schedulers that wish to make quick schedule changes. Using mathematical optimization for schedule development is an opportunity for further research. Our heuristic mimics the logic that manual schedulers currently work through while trying to maximize the number of flights flown, producing a near-optimal, sustainable solution. The current scheduling process takes 12 work hours to complete, time that these officers would prefer to spend on other operational tasks. In creating a heuristic, we can arrive at a feasible solution in a shorter amount of time that would provide an operable schedule to the squadron and limit the time the schedulers must spend each day tackling this problem. The constraints we define will ensure that IPs are being scheduled in an unbiased manner and can meet their personal scheduling requirements. By ensuring our schedule is feasible and follows the syllabus guidelines, we may find schedules that a manual scheduler would miss and, in turn, generate more flights per day. This would satisfy the goal of increasing the efficiency of operations and ensuring the 47th FTW hits their target of producing 375 pilots a year. In turn, this modeling approach will solve the current problem and improve the state of operations.

To develop the IP schedule, the team developed a user-friendly Python script. The schedulers at the squadron can leverage this script to input their files from GTIMS to generate a feasible daily schedule for the IPs. Using the data provided by GTIMS, our algorithm must first adjust the IP time-off requests. While the IPs submit a time-off request for the immediate times they cannot fly, this does not include the time they must arrive before the flight to brief, and the time after the flight to debrief. This is also true for what their personal scheduling conflicts for the day are listed as. Therefore, we ask the user to input the brief, debrief, and flight times for the flights that day. The combination of this time is the actual total flight time and is not accounted for should one simply look at the time a flight is listed in the maintenance document. The scheduler is then given the opportunity to manually schedule a specific pilot into a certain flight for the day. In coordination with the base, the schedulers identified that there are instances when a pilot has not inputted an excusal but is busy during certain flights for the day. Additionally, there may be subjective reasons for an IP to be placed on a certain flight during the day that none of these documents capture. In an effort to provide the scheduler with the subjectivity to meet operational objectives that may not be explicitly outlined in the data, we included this manual step before the schedule fills in the remaining empty flights for the day.

With this information inputted, our algorithm can now create a schedule. Here, mathematical optimization could be used, but there are numerous reasons the team decided to leverage a heuristic. The schedulers at the base requested a solution that could be solved in a matter of minutes so that they could use it as a draft that could be changed or altered quickly as new information arose in the squadron. The time required to solve problem instances to optimality is unsatisfactory for the users' requirements due to the complexity of the problem. Additionally, if there is an infeasible solution for the day, given the time-off requests or personal schedule limitations, the heuristic will still provide a partial solution and leave the infeasible flights empty. For such instances, using optimization would not output any schedule, and schedulers would not know which flights are limiting the feasible schedule development. A heuristic enables the schedulers' this flexibility and the freedom to troubleshoot scheduling issues at their level when they arise.

For the heuristic, the first step is to iterate through all of the time-off requests and the pilot's personal schedules and determine which pilots are available to fly that day. This limits the searching that our algorithm must do for when a pilot is available, as it eliminates all pilots that cannot fly any flights that day. With the new list of available pilots, our algorithm determines when a pilot is available to fly for the given day. The time-off requests and personal schedules describe the exact times a pilot will be out of the squadron or on another duty for the day. In order to determine what flights they could fly, we needed to ensure that they were not only available for the time that the flight was listed but also for the pre-brief, the duration of the flight, and the debrief as detailed above. Therefore, we took the time-off request of the pilot and determined the most constraining start and the end times of a potential flight around their unavailability that the pilot can fly, or in other words updated their time-off to account for their true flying availability considering these time commitments that come with each listed flight time. The equations to do so are depicted below.

$$Start\ Unavailable\ Time = Stated\ Start\ Unavailable\ Time - (Flight\ Time + Debrief\ Time) \quad (1)$$

$$End\ Unavailable\ Time = Stated\ End\ Unavailable\ Time + Brief\ Time \quad (2)$$

With the new detailed availability of the pilots, we can then compare their availability to the flights for the day to determine how many flights each pilot is available to fly for the day. Later, when our algorithm is assigning pilots to each flight, it begins by scheduling the pilot with the least amount of availability to ensure that the most time-constrained pilots are scheduled first. This not only maximizes the number of flights flown for the day, but also allows for flexibility in the schedule as the least constrained pilots will be able to fill in for others should a problem arise. The next step in our algorithm is to iterate through the list of flight times and, for each time, generate a list of IPs that are available to fly that specific flight that day.

At this point, our algorithm has a list of the pilots that can fly each flight and the number of flights that each pilot can fly. This list is additionally ordered by the pilot with the least number of flights that they can fly for the day to the pilot with the most. With this list, our algorithm begins to determine which pilots should be scheduled to fly during that day. Recall that an IP may get three different types of "assignments" for that day: an out-and-back, a local, or a double turn. Our algorithm begins by scheduling the out-and-backs because they require the most time and are the most constraining type of flight. On the schedule that the maintainers provide, out-and-back flights appear as two different time slots that are three hours and fifteen minutes apart. In turn, to indicate that a pilot has been scheduled for an out-and-back, our algorithm will schedule the same IP for both the exiting and the returning flight that day. Our algorithm begins by checking if the first most constrained pilot is available for the entire three-hour and fifteen-minute time range that this flight will take, including both the exit and the return flight. If the pilot is available for the entirety of the out-and-back, they are scheduled for both listed flights on the schedule, and if they are not, our algorithm moves on to the next person in the list until it finds a feasible match. As our algorithm moves to the next scheduled out-and-back, it will only consider the pilots that have not flown yet for the day. Our algorithm repeats this process until it has scheduled all the listed out-and-back flights.

Next, our algorithm schedules the double-turn assignments. A double-turn assignment is when a pilot is scheduled for two different local flights for the day. A constraint of the syllabus is that IPs are limited to a maximum of two flights per day, and they are mandated three hours and thirty minutes of rest time between flights. A scheduled duty such as working in the tower also breaks this required rest time. Our algorithm at this point only has local times left to schedule. With the remaining local times our algorithm looks at the list of available pilots for the listed flight and selects the most constrained pilot as previously. Our algorithm then checks this pilot's availability to fly the next flight at least three hours and thirty minutes later which would fulfill their mandated rest time. If the pilot is available to fly both the initial flight and the next flight that is at least three hours and thirty minutes past the initial flight, our algorithm then moves to check that pilot's availability in their personal schedule. If this pilot is not listed for a duty in that period and their rest can be met, they are scheduled for both flights and are now assigned a double turn for that day. Our algorithm will continue that process until it cannot feasibly schedule any more double turns while ensuring that pilots are meeting the syllabus required rest time or there are no available pilots.

Our algorithm is now left with the local flights that will not be assigned to pilots that are double-turning. Following the logic previously discussed, our algorithm selects the least constrained pilot for each flight. This will fill any empty flights in the schedule and round out the modeling process. The pilots left at the end will be the pilots with the most availability and will most likely be available to fly any flights for the specified day. This allows any gaps in the schedule to be filled when our algorithm reaches the end of the process and allows for flexible changes in the day. If the most available pilots are not already scheduled to meet their two-flight limit, they can fill in for any scheduling conflicts or problems that may arise. This is an additional way that our scheduling algorithm will support the base reaching the target of 140 flights each day. At the conclusion of the algorithm's run, it outputs on the user's computer in a readable and understandable form. This allows the schedulers in the squadron to have the IP schedule for the next day in the same output that their current manual process results in.

## 3. Results and Analysis

To assess the performance of our scheduling algorithm, we analyzed how the IP schedule that our algorithm created compared to the current manual scheduling process. While our algorithm does not schedule students, the algorithm is able to reduce the current 12-hour intensive schedule development process by automating the scheduling of the IPs. Our algorithm produces a feasible IP schedule in around five minutes, saving eight hours previously dedicated to this part of the process. This improvement freed up schedulers to perform other duties around the squadron and ultimately fly more frequently as they were not as busy completing the schedule for the next day. We also consistently scheduled all 140 flights for each day, which is an improvement above the 117 flight average and the 31.48 flight standard deviation that the squadron is currently achieving. This verifies that our algorithm can find solutions that a manual scheduler may miss, which leads to a more efficient schedule that meets the larger mission goal at a higher rate. Our algorithm also outputs a graph of the number of IPs available throughout the day. This graphic enables the scheduler to see what time during the day they are most constrained, and what times of the day they have many IPs to choose. In the future, this may enable the squadron to streamline their time-off requests to try and even out how many IPs are available versus unavailable if they can identify trends over time. This may mitigate losing flights due to numerous pilots being absent at the same time. An example of this visualization is depicted below for 19 Sept 2023.

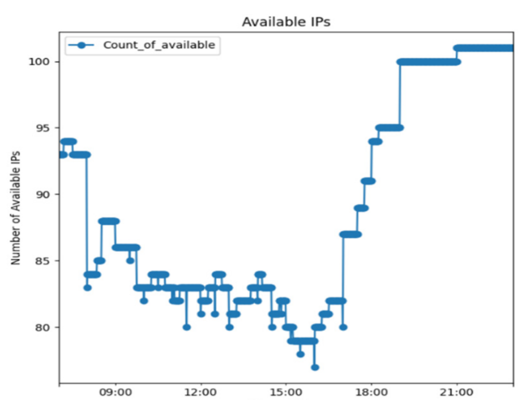


Figure 2. IPs Availability throughout the Day on 19 Sept 2023

Additionally, our algorithm will output the number of IPs available for each flight and the list of their names. This again allows them to track trends over time to see where they are constrained throughout the day, and potentially attempt to decrease the number of pilots gone simultaneously. As the schedule changes throughout the day, the pilots that are the least constrained should be available to fill in and complete a double turn. The manner in which the heuristic assigns pilots will also work towards maximizing the number of flights flown and will make last-minute scheduling changes easier. Overall, our algorithm cuts down the amount of time dedicated to schedule development, which allows pilots to focus on their flying duties versus scheduling, and works to maximize the operational effectiveness of the organization.

## 4. Conclusions, Recommendations, and Future Research

Our automated scheduling algorithm addresses the time-consuming manual scheduling process at the Laughlin Air Force base, which diverts resources away from the core flying mission. By employing heuristic-based programming, we drastically reduce the manual scheduling workload, generating daily flying schedules for instructor pilots at Laughlin Air Force Base in just 5 minutes—a substantial improvement towards reducing the previous 12-hour process. This automation saves 40 work hours weekly which translates to about $108,000 saved yearly for the squadron, a statistic found by leveraging the salary of the scheduling officers, and ensures instructors are efficiently assigned to meet the daily target of 140 flights. The user-friendly interface works to ensure continued implementation despite personnel turnover and overall enhances operational efficiency, supporting the mission of graduating 375 pilots annually.

The solution offers a replicable framework for other Air Force pilot training bases, promising significant time and resource savings while enhancing operational effectiveness. Future research could work to redefine our algorithm to work towards an optimal solution rather than a feasible one and should focus on automating the student scheduling portion of the process. These efforts would further reduce the time for schedule development and enhance mission success at the base.

## 5. References

Aringhieri, R., Landa, P., Soriano, P., Tànfani, E., & Testi, A. (2014). A two level metaheuristic for the operating room scheduling and assignment problem. *Computers and Operations Research, 54, 21-34*. https://doi.org/10.1016/j.cor.2014.08.014

Birjandi, A., & Mousavi, S. M. (2019). Fuzzy resource-constrained project scheduling with multiple routes: A heuristic solution. *Automation in Construction*, *100*, 84–102. https://doi.org/10.1016/j.autcon.2018.11.029

Ding, Hu, Su, & Xu. (2021). Multi-Objective Flight Schedule Optimization Based on Improved Evolutionary Algorithm.

Erdemir, U. (2014, March 14). *6th International Conference on Information Science, Computer Technology and Transportation, 10.1007*

Geiger, M. J. (2017). A multi-threaded local search algorithm and computer implementation for the multi-mode, resource-constrained multi-project scheduling problem. *European Journal of Operational Research*, *256*(3), 729–741. https://doi.org/10.1016/j.ejor.2016.07.024

Gershkoff, I. (1989). Optimizing Flight Crew Schedules. *Interfaces*, *19*(4), 29–43. https://doi.org/10.1287/inte.19.4.29

Graves, G. W., McBride, R. D., Gershkoff, I., Anderson, D., & Mahidhara, D. (1993). Flight crew scheduling. *Management Science*, *39*(6), 736–745. https://doi.org/10.1287/mnsc.39.6.736

Lamas, P., & Demeulemeester, E. (2014). A purely proactive scheduling procedure for the resource-constrained project scheduling problem with stochastic activity durations. *SSRN Electronic Journal*. https://doi.org/10.2139/ssrn.2464056

Montana, D., Brinn, M., Moore, S., & Bidwell, G. (1998). Genetic Algorithms for Complex, Real-Time Scheduling . *Network Operations and Management Symposium*, *1*, 245–248. https://doi.org/10.1109/NOMS.1998.654894.

Norman, B. A., & Bean, J. C. (1999). A Genetic Algorithm Methodology for Complex Scheduling Problems. *Naval Research Logistics*, *46*(2), 199–211. https://doi.org/https://doi.org/10.1002/(SICI)1520-6750(199903)46:2%3C199::AID-NAV5%3E3.0.CO;2-L

Rittri, M., & Allerbo, G. (2000). Scheduling Pilot Training. *Semantic Scholar*. https://www.semanticscholar.org/paper/SCHEDULING-PILOT-TRAINING-Rittri-Allerbo/b0203a7fae871482b1100f20400894c52ceb0222

Roslof, J., Harjunkoski, I., Bjorkqvist, J., Karlsson, S., & Westerlund, T. (2001). *Computers and Chemical Engineering*, *25*(4–6), 821–828. https://doi.org/https://doi.org/10.1016/S0098-1354(01)00674-3