A NEW APPROACH TO CAREER FIELD MATCHING FOR COMMISSIONING AIR FORCE CADETS

THESIS

Daniel Griffen Laird, Second Lieutenant, USAF

AFIT-ENS-MS-22-M-143

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

AFIT-ENS-MS-22-M-143

A NEW APPROACH TO CAREER FIELD MATCHING FOR COMMISSIONING AIR FORCE CADETS

THESIS

Presented to the Faculty

Department of Operational Sciences

Graduate School of Engineering and Management

Air Force Institute of Technology

Air University

Air Education and Training Command

in Partial Fulfillment of the Requirements for the

Degree of Master of Science in Operations Research

Daniel Griffen Laird, B.S.O.R.

Second Lieutenant, USAF

March 24, 2022

AFIT-ENS-MS-22-M-143

A NEW APPROACH TO CAREER FIELD MATCHING FOR COMMISSIONING AIR FORCE CADETS

THESIS

Daniel Griffen Laird, B.S.O.R.
Second Lieutenant, USAF

Committee Membership:

Lieutenant Colonel Jesse G. Wales, Ph.D
Chair

Bruce A. Cox, Ph.D
Member

Major Michael J. Garee, Ph.D
Member

AFIT-ENS-MS-22-M-143

Abstract

The current method of assigning graduating cadets from the United States Air Force Academy and Reserve Officers' Training Corps (ROTC) detachments to their career fields uses an integer programming model to maximize "global" Air Force utility, subject to several Air Force-defined constraints. This utility evaluates the positive benefit of assigning a certain cadet to a certain career field. This paper discusses the issues with such a model, as well as presents a new, more refined approach to the problem. Rather than provide a one-size-fits-all formulation of this particular assignment problem, a Value-Focused Thinking (VFT) framework is applied, in conjunction with an optimization model using the framework, to measure overall solution quality for an alternative assignment of cadets to career fields, given numerous weight and value parameters. The power of optimization under a VFT framework is in the ability to capture what decision-makers want, as well as how much they want it. This research shows that the new VFT model outperforms the original model on solution quality by almost 7% when measured using many different VFT weight and value parameters on real class year instances.

AFIT-ENS-MS-22-M-143

To my loving Wife, whose patience through this extensive process was most greatly appreciated, and whose support carried me through to the very end.

Table of Contents

Page

List of Figures

Figure Page

List of Tables

A NEW APPROACH TO CAREER FIELD MATCHING FOR COMMISSIONING AIR FORCE CADETS

I. Introduction

"What distinguishes our military, what makes us the premier fighting force in the world and guarantees it will prevail in any conflict, is the quality of our service members" – Secretary of Defense Mark T. Esper

1.1 Background

Assignment problems are found in all facets of society, from assigning children to their kindergarten classes, to matching medically trained residents to their designated hospitals [1]. The assignment problem requires an appreciation of the difficulty and scale of the particular instance to be solved. Where one problem might be complex but insignificant, another may be simple yet appreciable. For most smaller problems, simple linear programming techniques utilizing the simplex method are effective for solving the problems quickly. If the problem is very large, however, a heuristic approach may be necessary to quickly limit the amount of solutions possible [2]. Additionally, the objectives of the matching process are another key factor in determining how to solve the problem. Sometimes all that is desired are the preferences of the entities being matched. Other times the assignments are generated based on entity measures of merit, entity eligibility for a particular assignment, the utility realized by a particular match, and so on. Therefore, a detailed analysis of the particular assignment problem in question is needed to determine which method of operation is appropriate.

The Air Force Personnel Center (AFPC), located in San Antonio, Texas, is tasked with a very important assignment problem every year: the assignment of Air Force cadets to their Air Force Specialty Codes (AFSCs), which are alphanumeric designations for particular career fields. For the remainder of this paper, the terms "career fields" and "AFSCs" are used interchangeably. Additionally, to protect sensitive information in this paper, real class years and AFSCs are not used (with the exception of Table 1 in Section 3.1). Instead, class years are referred to by letters (A, B, C, etc.) and AFSCs are sorted within a year group by their quotas (the number of cadets they target) and indexed in that order. The AFSC with the largest quota in class year B, for example, would be denoted as "B1". The remaining $M-1$ AFSCs would therefore be (B2, B3, ..., BM). Each year, AFPC analysts acquire AFSC preference lists for commissioning lieutenants from both the United States Air Force Academy (USAFA) and the various Reserve Officers' Training Corps (ROTC) detachments. After receiving all cadets' AFSC preferences, the analysts attempt to find the best possible match between the cadets and that year's specific set of non-rated AFSCs.

Rated AFSCs refer to career fields with an aeronautical focus, and they are separated from the rest of the AFSCs and matched through a different process. Hence, the focus for this research paper is on non-rated AFSCs. There are various objectives that the Air Force wants to meet with the solution they implement (see Section 3.1). Thus, capturing the correct trade-offs between different objective criteria is crucial. For the vast majority of officers, an AFSC assignment locks a particular graduating cadet into that profession for the duration of their career in the Air Force. Because of this, it is vital to the mission of the Department of Defense (DOD) that the AFSC matching process be done with great care. If we create a model that does not incorporate all available information in its optimization for decision making, we sacrifice better alternatives that would otherwise remain unknown.

Challenges to the effective assignment of AFSCs to cadets arise throughout the whole matching process. The current model is an integer program that maximizes the total satisfaction of cadets and AFSCs subject to several constraints mandated by Air Force leadership. These constraints almost always result in an infeasible solution due to improper delineation between mandatory criteria and desired objectives. This could, perhaps, be an unfortunate product of the current Air Force terminology within the Air Force Officer Classification Directory (AFOCD). The AFOCD dictates requirements for degree demographics of the cadets assigned to the career fields in a given year. Cadets' academic degrees determine their eligibility for the AFSCs. Since some years result in an influx of cadets with one degree at the expense of another, the AFSC matching model often becomes over-constrained. This causes the analysts to have to check their solutions with the various career field managers due to the high probability that certain AFOCD criteria are not met.

Another concern is that the implemented solution may not be the optimal solution, since it is often altered as a result of new information that was not revealed prior to the model run. Occasionally a cadet's AFSC is swapped for another, and that is all that is changed. In other instances, the cadet's new AFSC is fixed as a constraint in the model, and the model is run again. Depending on the impact that this new addition has on the overall optimality of the original solution, there could be a substantial change to the model's new solution. If this is the case, then a whole new set of matches must be checked with each of the career fields again, further resulting in a deviance from the original optimal solution. The magnitude of this deviance should be taken into consideration prior to the development of a final solution. If the model is robust, small tweaks to the solution have little effect on the entire system. The search for an adequate, robust model is the purpose of this paper.

1.2 Problem Statement

Rather than a one-size-fits all approach to the assignment of AFSCs to cadets, careful consideration of all factors in the matching process should lead to the implementation of a particular model's solution among several different alternatives. This paper uses a Value-Focused Thinking (VFT)-based optimization model to maximize solution quality for the AFSC/Cadet assignment problem. The implementation of this methodology would result in the analysts' use of VFT to compare solution results for the non-rated line AFSC match to cadets.

1.3 Summary of Contributions

We develop an optimization model, and associated VFT framework, to generate and compare alternative solution assignments for the AFSC/Cadet problem effectively. The framework provides the decision-makers with the capability of intuitively prescribing value to the various solution objectives, and the optimization model generates solutions based on that framework. We validate this optimization model against the real solutions, and find that it consistently outperforms the original model.

1.4 Thesis Structure

The remainder of this thesis is structured as follows. Chapter II presents an in-depth review of the literature behind the assignment problem as well as several methods of solving it. Chapter III presents the formulation of the AFSC/Cadet assignment problem in a VFT framework and introduces the method to solve this problem. Chapter IV discusses the results and analysis of the proposed methodology. Chapter V presents the conclusions as well as future work potential of this research.

II. Literature Review

This chapter discusses the relevance of prior research to this paper, and provides the theory and application of different methods to formulate and solve assignment problems.

2.1 The Assignment Problem

2.1.1 Overview

The generalized assignment problem seeks to find the minimum cost for the assignment of a set of N individuals to M jobs. Using these entities in an example model, an assignment cost might mean the actual monetary value associated with a particular individual working a given job. In this problem, there are two main assumptions. Each individual, i, can be assigned to at most one job, j, (1b) and each job has some maximum capacity b_j of individuals that can be assigned to it (1c). We define x_{ij} as a binary variable indicating if individual i is assigned to job j (1d). The cost of such an assignment is denoted by c_{ij}. The set of all individuals is $\mathcal{I}$, and the set of all jobs is $\mathcal{J}$. The mathematical formulation of this assignment problem is:

$$\text{minimize} \quad \sum_{i \in \mathcal{I}} \sum_{j \in J} c_{ij} x_{ij} \tag{1a}$$

$$\text{subject to} \quad \sum_{j \in \mathcal{J}} x_{ij} = 1 \quad \forall i \in \mathcal{I} \tag{1b}$$

$$\sum_{i \in \mathcal{I}} x_{ij} \leq b_j \quad \forall j \in \mathcal{J} \tag{1c}$$

$$x_{ij} \in \{0,\ 1\} \quad \forall i \in \mathcal{I},\ j \in \mathcal{J} \tag{1d}$$

Techniques to solve the assignment problem are applied in many different ways to find the best solution to the problem with the shortest amount of time and effort in doing so. Relaxation techniques, whether that be through linear programming relaxation, Lagrangean relaxation, or relaxations through deleting constraints [3], are useful methods of achieving initial solutions through tweaking the constraints, which can be used in conjunction with the branch and bound technique to find the optimal solution. While there are several other techniques in the literature that attempt to solve this problem, including a variation of the Hungarian Method (described in the next section) known as the matrix one's assignment method [3], this section presents a few of the more popular methods.

2.1.2 The Hungarian Method

One method of solving the assignment problem is the Hungarian Method. Originally developed in 1955 by H. W. Kuhn [4], this method assumes a balanced assignment problem, in which n jobs are matched to n individuals. For a balanced assignment, Kuhn distinguishes the "simple" assignment problem from the "general" assignment problem as follows. In Kuhn's simple assignment problem, the algorithm seeks to find an assignment of jobs to minimize the total number of unqualified pairs. The distinction between unqualified pairs and qualified pairs is made since individuals may be qualified for different jobs. The Hungarian Method first seeks to find a "complete" assignment, in which there are no unassigned individuals qualified for any unassigned jobs. From there, transfers between jobs and individuals are made until there are no additional transfers for which the total number of qualified pairs can be increased. The solution is optimal. For Kuhn's general assignment problem, a similar approach is taken. In this problem, the solution is found through transfers based upon the minimum costs associated with a pair. Kuhn's general assignment problem is the baseline for the literature's definition of the balanced assignment problem.

In an unbalanced assignment problem, in which N jobs are matched to M individuals ($N \neq M$), the Hungarian Method's solution includes dummy variables to force

$N = M$. While this may work for instances where N is very close to M, it is not adequate for the AFSC/Cadet problem since the number of cadets is far greater than the number of AFSCs. One method of solving the unbalanced assignment problem is a modification of the Hungarian Method proposed by Avanish Kumar [5]. Kumar's modification to the Hungarian Method partitions the unbalanced assignment problem into two balanced sub-problems based on the minimum cost columns and rows of the assignment cost matrix. Once partitioned, the two balanced assignment sub-problems can be solved using the traditional Hungarian Method. The sum of the optimal assignment solutions to each sub-problem is proposed to be the optimal solution to the unbalanced assignment problem. Kumar demonstrates his algorithm on a small unbalanced example instance, and finds a solution with an objective value of 1550. Kumar states this solution to be the optimal solution.

V. Yadaiah and V. Haragopal [6] propose a new solution to Kumar's modification in which they apply a Lexi-search method to both of the partitioned balanced assignment problems to find the optimal value. Their Lexi-search method is applied to Kumar's example in which they achieve the same value as Kumar (1550) but at a quicker rate. The issue with these two approaches, however, is that the optimal solutions to the sub-problems do not always translate to the optimal solution to the original unbalanced problem. In fact, Kumar, Yadaiah and Haragopal's optimal solution of 1550 to their numerical example is proven wrong by Nathan Betts and Francis Vasko [7] in the same year that Yadaiah and Haragopal's paper was published. Betts and Vasko use a simplified version of the Hungarian Method by which they copy the five jobs and then create two additional individual dummy variables, resulting in a 10×10 assignment cost matrix. This new problem is solved using the original Hungarian Method and a new solution of 1520 is reached. However, simpler is not always better [7], since this solution is proven to be sub-optimal again in 2019.

Rabbani et al. [8] find a solution that eliminates the assumption made by the previous authors in the literature that no more than two individuals may be assigned to a job. This assumption is a product of the fact that the unbalanced assignment problem is broken up into two sub-problems under the proposed modifications to the Hungarian Method. The new variant of the Hungarian Method results in a minimum cost of 1470. At this point, however, the accuracy of the Hungarian Method applied to the unbalanced assignment problem is called into question. We turn to the Simplex Method to solve Kumar's orginal unbalanced assignment problem to find the true optimal solution. Given the same assignment cost matrix, and the assumption that one and only one job may be assigned to an individual but there can be as many individuals assigned to a single job as needed, the minimum cost is 1450 (see Appendix A for the solution). While the Hungarian Method works for balanced assignment problems, there is no viable modification of the Hungarian Method that always produces the optimal solution for the unbalanced assignment problem. Therefore, we consider other approaches to solve the unbalanced (or generalized) assignment problem.

2.1.3 Genetic Algorithms

The genetic algorithm (GA) at its core refers to the method of utilizing the concepts of natural selection and evolution in biology to evaluate solutions and converge on optimality. Developed by John Holland in the 1970s, GAs can be used for a wide range of problems, from discovering the best strategies in the game of chess to developing complex aeronautical systems such as jet turbines [9]. As a generalized explanation of the algorithm, we begin with an initial random population of solutions to a given problem. The algorithm chooses the best solutions, also known as the most fit (in mathematical programming problems, this refers to those with the highest feasible objective value), and "mates" those solutions with each other to produce "offspring" that replace the weaker solutions in the next generation. The offspring are simply combinations of parts of each parent solution. Mutations can occur as well with some probability. These mutations switch individual "genes" (decision variables) in an offspring, allowing for completely new solutions to be generated. The algorithm iterates through this process until an optimal, or near-optimal, solution is found.

GAs can be used to solve the assignment problem as well. Chu and Beasley, in their research from 1996, create a genetic algorithm for the generalized assignment problem utilizing two solution evaluation functions, fitness as well as unfitness [10]. Fitness is the solution's objective value function as described previously. Unfitness refers to the extent to which the solution is infeasible. Chu and Beasley incorporate these two principles into their algorithm during the selection phase of the evolution iteration, by which the parents of new offspring are chosen to maximize the objective value of the solutions, while also minimizing the degree that the solutions are infeasible. Three years later, a genetic algorithm is applied to the channel-assignment problem in cellular radio networks, in which n cells are assigned to z frequencies [11].

Beckmann and Killat demonstrate that their use of a combined genetic algorithm (CGA) on this problem greatly outperforms other heuristics for this type of assignment problem, both in computational time and convergence on optimality. A CGA is another term for a hybrid genetic algorithm. Hybrid genetic algorithms are heuristics that incorporate genetic algorithms along with other conventional methods, and they can often outperform their completely GA or completely conventional counterparts. Ahmed [12] uses three different hybrid GAs to compare against two state-of-the art algorithms, DE-SK [13] and the bees algorithm BEE [14], which itself is discussed in the following subsection, and finds that one of the hybrid genetic algorithms consistently outperforms both of them. Genetic algorithms are therefore a viable candidate for implementation in this research paper.

2.1.4 Other Methods

Given the extent that the generalized assignment problem pervades various situations across the globe, there are plenty of other methods that researchers have developed to solve this problem efficiently in addition to those described previously. Relatively newer techniques to solve this problem tend to be modeled after real-world processes. The auction algorithm, for example, is used on the assignment problem to resemble a modern auction [15]. This method utilizes the idea of the cost variable c_{ij} to instead describe the bids that person i would place for object j, or rather the negative value of that bid in the case of a minimization problem. Each unassigned person simultaneously bids for objects which raises their prices. The objects are then awarded to the highest bidder. Bertsekas shows that his method can also be applied to the transportation problem, which the generalized assignment problem is a subset of [16].

Other real-world phenomena that tend to be capitalized on by mathematicians are found in nature. Along the lines of the genetic algorithm, the bees algorithm is a method that models the behavior of a swarm of bees that communicate with each other in pursuit of the best nectar sources to solve the generalized assignment problem [14]. The bees work together to inform the swarm of the magnitude and direction of nearby food sources relative to the Sun. In the form of the "waggle dance," the bees also include information on the quantity of nectar available. In a similar fashion to the genetic algorithm, the algorithm uses this behavior in bees to iterate through various neighborhoods of assignment solutions, pursuing improvements on the status quo. Neural networks, which model the functions of the brain through the firings of neurons within a network, can also be used to solve the generalized assignment problem, as is the case with the static weapon-target assignment problem in ballistic missile defense [17].

2.2 The Stable Marriage Problem

2.2.1 Overview

A very close relative of the assignment problem is the stable marriage problem. This section first describes the history and formulation of the problem, then presents several variations to the problem that introduce some complications to the purposes of finding a solution. Many of these variants are extremely necessary to the potential formulation of the AFSC/Cadet problem as a stable marriage problem.

2.2.2 One-to-One Matching

The assignment problem can be formulated and solved through various methods. When solving the problem, it is important to consider the available information behind the scenario itself. In the case of the AFSC/Cadet matching assignment problem, the cadets submit preference lists for the AFSCs they'd prefer to be assigned to. This list, historically a cadet's top six choices for AFSCs, is accompanied by a list of utility values that distinguish the cadet's measure of happiness that they would receive should they be assigned to a given AFSC on their preference list. A cadet's first choice AFSC would have a utility of one, and any AFSC not on the cadet's preference list would correspond to a utility of zero. All other AFSCs on the cadet's list may have utilities ranging from one to zero. The AFSCs themselves could be modeled to have a preference for each of the cadets as well. Rather than measuring a particular match of cadet to AFSC by its associated utility, as seen previously, matches are determined by each entity's preference for the other. We therefore could model this assignment problem as a stable marriage problem.

The stable marriage concept, initially presented by D. Gale and L. Shapley [18], describes the assignment of marriage between n men and n women. Each individual ranks the members of the opposite sex according to their own preferences for a spouse. An assignment of marriages is stable when there are no two individuals, a man and a woman, who prefer each other over their assigned spouses. Such a marriage would be unstable, since there is nothing stopping them from breaking off their current marriages for each other. A stable marriage, therefore, can include all possible combinations of preferences within the pair excluding the unstable scenario. The Gale-Shapley algorithm always ensures a stable assignment solution. The stable marriage algorithm begins as the men each propose to the top woman on their preference lists. The women then accept their best offer (i.e., the highest man on their preference lists who proposes to them) and reject the others. The men propose again to their top women who have not yet rejected them. If a woman only receives one proposal, she automatically accepts it, but has the option to reject the man if a better suitor becomes available. This process of propose-accept-reject repeats until all individuals are matched. The resulting solution is a stable marriage assignment.

While this algorithm works for what it is intended, there are several issues with the baseline interpretation of the algorithm's resulting assignments. For one, the algorithm only yields what is known as the "male optimal" solution [19]. Because the men have the power to propose to their top choices, they will always be able to optimize their marriages within the entire candidate pool. The women, on the other hand, can only choose between the men who propose to them. Additionally, the Gale-Shapley algorithm does not guarantee an optimal solution, since the objective is only to find a stable marriage solution. McVitie and Wilson [20] introduce an algorithm that finds all possible stable marriage solutions and offers a metric of egalitarian optimality, which measures the equal preference for men and women. This egalitarian measure of optimality is expanded by Irving et al. [19], in which total satisfaction is optimized through the use of male and female "shortlists" to quickly find all possible stable solutions in polynomial time $O(n^4)$ including the optimal solution.

2.2.3 Variants of the Problem

While there are methods of addressing the inequalities of the matchings, one important variant of the stable marriage algorithm is the many-to-one extension. In the literature, this is sometimes referred to as the "college admissions problem" or the "hospital and residents" problem [21]. For the purposes of this paper, we will use the latter. The hospitals/residents problem is a variant of the stable marriage problem in which n residents are matched to m hospitals, and generally n is much larger than m [22]. Quotas are introduced as the maximum number of residents that the hospitals are willing to admit. The hospitals could therefore conceivably not meet their quotas if there are not enough residents to fill all of them exactly.

Furthermore, in a many-to-one "market," another dimension of stability is developed known as "group stability" [21]. Group stability refers to the hospitals' preferences for groups of residents, rather than the individual residents themselves. Preferences over individuals are known. A resident submits a preference list for the hospitals, and the hospitals do the same for the residents (whether by direct submission or indirectly through a function that weighs all their options and candidate qualifications). In a many-to-one market, a hospital's preference over different groups of residents can sometimes be determined through inspection. For example, if the only difference between two assignments for a hospital is a particular resident, then the hospital prefers the assignment which has the more preferred resident. This second dimension of preferences complicates the problem substantially, and researchers have tried to determine the varying degrees of the assignments through the notation of weak and strong stability [23].

Indeed, there are many challenges to the practical implementation of the stable marriage algorithm on these more difficult variants of the problem. One important variant to the basic problem is the concept of incomplete preference lists [24]. Sometimes certain residents are unacceptable to the hospitals, and this is indicated by the lack of appearance of unacceptable resident r on hospital h's preference list. Such an occurrence would mean that h would rather not fill its quota than admit r. Additionally, there are instances in the literature of indifference in preferences between the sets, which occur when members of the set do not rank members of the other set in strict order [24]. Ties and incomplete preference lists can also occur simultaneously, resulting in a much harder problem to solve. An added complication to the many-to-one scenario is the concept that each hospital may have certain minimum quotas in addition to the maximum limit. Bir et al. prove that there may not actually be a stable matching in this case due to the occurrence of one or more blocking pairs [25]. In the context of the AFSC matching problem, by which there are desired minimum quotas for AFSCs, this is an issue.

The AFSC/Cadet problem's stable marriage formulation incorporates many of these variants of the classical stable marriage problem, making it nearly impossible to solve. Due to the numerous constraints (or, perhaps more accurately, objectives) that are placed on the assignment, AFSC preferences for cadets are more complicated than the typical ordinal preferences found in the literature. AFSC preferences are dynamic by nature, meaning their preferences for cadets pertain to the overall quality of the set of cadets assigned to them. They can only distinguish preferences for cadets based upon the value added by one cadet over another. Currently, there is no work in the stable matching problem community by which dynamic preference lists are incorporated. While this is a very interesting topic to explore, it is outside the scope of this paper's objective. AFSC preferences are for objective measures of the solution, and should be modeled as such. An understanding of these AFSC preferences is the inspiration for a different way of thinking about this problem presented in the next section.

2.3 Decision Analysis in Optimization

2.3.1 Overview

Decision problems arise in all areas of life. They range from uncomplicated and insignificant problems, such as what brand of cereal one should eat in the morning, to much more complex and substantial problems, such as the selection of the best military training aircraft for the Spanish Air Force [26]. Decision analysis describes the process by which a decision problem is framed, alternatives are distinguished, and a choice is made between the alternatives based upon certain decision criteria. This section outlines the concepts of Value-Focused Thinking, a methodology of decision analysis, and multi-criteria optimization, a structure of optimization problems by which the objective function is composed of several different objectives. Techniques to conduct sensitivity analysis on the alternatives generated through decision analysis are also presented.

2.3.2 Alternative-Focused Thinking vs Value-Focused Thinking

Ralph Keeney, in his book on the formation of Value-Focused Thinking (VFT) as a framework for decision analysis [27], hypothesizes that most people have the decision process backwards. People tend to come up with alternatives to a decision problem first, then identify some sort of evaluation framework by which to measure those alternatives. Keeney refers to this process as "Alternative-Focused Thinking" (AFT) [27]. AFT restricts the decision maker's potential of achieving the best value out of a particular decision context because it limits the scope of alternatives that are available. For example, if someone's decision problem is to determine where to live using an AFT mindset, their first step would be to propose some set of houses as alternatives. To do this, they narrow their search down to a few houses in a particular city and evaluate them according to certain objective measures. Since the alternatives are already set, the decision maker is limited in the value they receive by those predetermined alternatives. If the values by which to measure the quality of the housing options were defined at the beginning, perhaps better houses that reflect the true desires of the decision maker could have been found.

Value-Focused Thinking (VFT) refers to a broad component of decision analysis whereby the values and objectives of a decision problem are determined first, which constructs the problem's "decision context." Only after the objectives have been specified for a decision context are alternatives generated and compared against each other [28]. Instead of having to develop a fixed set of alternatives to a decision problem initially, and then compare those alternatives using some sort of value hierarchy afterwards, VFT allows the decision maker to distinguish the objectives that really matter. Once the objectives are understood, alternatives may be compared using the VFT framework. In a study conducted in 1999, relatively soon after Keeney's book on VFT, two groups are given the same decision problem involving the selection of advanced academic courses [29]. One group analyzes the problem through AFT, and the other uses VFT. Researchers find that the group utilizing VFT developed a hierarchical objective structure of the problem that is equal to or superior to that of the AFT group in all qualities judged by the actual analysts solving the problem. VFT is not only objectively superior to AFT, but it is also intuitively the better choice since the range of alternatives available is much wider.

VFT is widely accepted and used in the decision analysis community. Sheng et al. [30] use a VFT approach for the purposes of examining the "strategic implications" of information technology, with an emphasis on mobile technology, in a leading publishing company. The researchers discover through their VFT "objective hierarchy" that there are three main ways that mobile technology can impact the strategic organization of a company: improve work processes, increase internal communication, and enhance sales and marketing effectiveness [30]. VFT can also be used for risk management decisions, and it consistently outperforms approaches utilizing AFT. A 2002 study testing this hypothesis found that individuals in the VFT "workshops" outperformed those in the AFT workshops when presented with a decision problem concerning the management of risks to salmon habitats from hydroelectricity generation [31].

The utilization of a VFT framework involves several components. First, the analyst specifies one or more fundamental objectives for the decision context. The fundamental objectives are the "ends" objectives that reflect the main overall goal the alternatives attempt to meet. For an optimization problem, and specifically the new optimization formulation presented in this paper, the fundamental objective is most likely reflected in the objective function. For problems with multiple fundamental objectives, a multi-criteria optimization formulation is often used. Second, the "means" objectives are specified to provide the alternatives with a method of achieving the ends, or fundamental, objectives. These are the smaller objectives by which the fundamental objective(s) can be broken down. Once the objectives are determined, values are allocated to measure how close or how far the alternatives are from meeting each of those objectives. For an additive value model, these values must be in the range of zero to one. This structure is known as the objective hierarchy of a VFT model, and it is the framework with which to compare alternatives against each other.

2.3.3 Multi-criteria Optimization

Multi-criteria optimization problems (MOPs) refer to a set of optimization problems in which there are multiple, often conflicting, objectives. These objectives are listed as several different components of the objective function, meaning the overall objective function itself is a combination of various objective functions [32]. For a maximization MOP with p objective functions, the objective function can be defined as:

$$\begin{aligned} \text{maximize } & (f_1(x), ..., f_p(x)) \\ \text{subject to } & x \in \mathcal{X} \end{aligned} \tag{2}$$

where x is an n-dimensional solution vector of decision variables and $\mathcal{X}$ is the set of all feasible solutions. If the priorities of the objective functions for the MOP are unknown, trade-offs between the objectives can be calculated to estimate their underlying importance through the use of pairwise trade-offs [33]. Determining pairwise trade-offs can be conducted through comparing two objectives together and fixing all other objectives to some value. Here, the benefit of adjusting one objective at the expense of another can be evaluated to determine which objective functions are most preferred. In the case that $f_1(x)$ is preferred to $f_2(x)$, this can be denoted by $f_1(x) \succ f_2(x)$.

Along these lines, the pairwise trade-offs between the alternatives themselves can also be determined, which allows for more graphical representations of the quality of those alternatives. One mechanism to do this is through the use of convex preference cones, which can quickly eliminate alternatives that are not as desired as others [34]. This creates the Pareto-frontier for the MOP, which designates the set of solutions that are Pareto-optimal. A Pareto-optimal, or efficient, point is a solution for which there is no other solution that is better off in at least one objective while still at least as good in all others [34]. The Pareto-frontier is a very powerful curve to identify for an MOP, and it is a concept that is utilized in Chapter IV to provide sensitivity analysis on alternative assignment solutions for this paper's new optimization model.

2.3.4 Sensitivity Analysis

Additive value models are a very common approach to handling decision problems with multiple objectives. VFT incorporates an additive value model within its objective hierarchy to discern objective trade-offs. The objective value, therefore, is a weighted sum of all objective attributes. An attribute is a component of an objective on the lowest level of the value hierarchy (the objectives that do not branch into more objectives). In the decision example involving the purchase of a new home presented earlier, one objective could be to maximize resale value of the house. An attribute of that objective could be the square footage, which we also want to maximize. The overall value function of an additive value model can be simplified to

$$V(x_i) = \sum_{j=1}^{n} w_j v_j(x_{ij}) \tag{3}$$

where x_i is a vector of attribute measures for a particular solution, with $x_i \in \mathcal{X}$, v_j is the value function for attribute j, and w is a vector of attribute weights satisfying the constraint

$$\sum_{j=1}^{n} w_j = 1. \tag{4}$$

This function defines the overall objective value, $V(x_i)$, within the bounds [0, 1]. This objective value can be interpreted as the percent of a perfect solution obtained, since an objective value of 1 means that all objectives have been met (their attributes have values of 1, or 100%). The sensitive nature of $V(x_i)$ to small changes in w for these kinds of additive models emphasises just how important proper analysis of alternative solutions really is. For optimization models, we can find the optimal, or at least near-optimal, solution vector x_i that maximizes $V(x_i)$ under the subjective weight and value parameters w and v, respectively. The result, x_i, is known as the "best non-dominated multi-attribute alternative," or simply "best act" for that particular instance of the problem [35]. For example, consider two acts, x_b and x_i, such that $b \neq i$. The act x_b is the best act for weight parameters b, therefore

$$\sum_{j=1}^{n} b_j v_j(x_{bj}) \geq \sum_{j=1}^{n} b_j v_j(x_{ij}) \quad \forall i. \tag{5}$$

Barron et al. present two methods of providing sensitivity analysis on the weights of x_i: a nearly-equal weights procedure and a least squares procedure [35]. While the nearly-equal weights procedure may be useful for problems where the objectives' degrees of importance are similar in nature, the AFSC/Cadet assignment problem includes objectives that are very unequally weighted. For this reason, we will only consider the least squares procedure. In the least squares procedure, the goal is to calculate the weights for two acts, x_i and x_b, that make the value of act x_i exceed that of x_b by some amount Δ, and to calculate these new weights, w, such that they are as "close" as possible to the original arbitrary weights, b. Barron defines "close" using the minimum squared deviation principle, so that the new weights are the solution to

$$\text{minimize} \sum_{j=1}^{n} (w_j - b_j)^2. \tag{6}$$

The decision maker can determine what the initial weights are, and then alternatives may be generated and compared against each other. Where x_b may be the best act given weight parameters b, another act, x_i could be found to exceed x_b by Δ under weight parameters w calculated from the least squares procedure. This could be especially useful when comparing the alternative solutions generated by the new AFSC model with those of the original model.

2.4 Career Field Matching

2.4.1 Overview

Existing literature that involves the assignment of careers to individuals is understandably quite small. The military, for the most part, is the only real organization that determines its members' career paths for them. Additionally, within this subset of assignment problems the stable marriage concept appears often. This section discusses how current methods of assignment are implemented as well as the prevalence of the stable marriage algorithm within these methods.

2.4.2 Career Assignments

In most cases, military career assignments are allocated directly, rather than through a free market system like that of the civilian sector. It is important to note here that in the military, the word "assignment" refers to the job and location a military member occupies, rather than the process itself described through the assignment problem. Because career assignments are prescribed to military members in this way, innovation in the methods of determining those assignments is often a necessity. One example of such innovation is the Army officer assignment problem. Ferguson uses the stable marriage algorithm to develop a new model that matches Army officers to their next assignments [36]. Prior to Ferguson's thesis, the Army would match officers to their assignments by hand, which leads to higher likelihood of human error and sub-optimal solutions. If implemented, significant man-hours spent laboring over these assignments could be reduced, and better solution sets of assignments may be awarded to Army officers (those that better reflect the preferences of all entities involved).

One other variant of the stable matching problem pertains to the concept of matching with contracts, a problem that troubled researchers until the advancement by Hatfield and Milgrom [37]. The United States Military Academy (USMA) uses an incentive program that offers cadets contracts that provide guaranteed first-choice preferences in exchange for an extension of their service commitment [38]. Sonmez and Switzer utilize the matching with contracts variant of the stable marriage problem to produce an agent-based model that results in a stable optimal solution. This is not the first paper to address the issues of the assignment of AFSCs to cadets. Prior to 2003, the Air Force used a greedy matching algorithm by which higher ranking cadets would receive their top choice AFSC, and lower ranking cadets would receive their highest ranked AFSC that was still available. Armacost and Lowe, in their 2003 paper, propose an integer program optimization model that maximizes cadet utility, subject to the AFSC quota constraints [39]. The optimization model they employ greatly outperforms the previous greedy algorithm by finding a cadet preference optimal solution. When the only constraint is the number of new lieutenants that can be accessed into the AFSCs, an integer program such as the one proposed by Armacost and Lowe is sufficient.

In the context of the operational Air Force assignments, rather than the matching of AFSCs, Lepird proposes a revolution in the way these operational assignments are matched [40]. As this paper intends to prove for the assignment of AFSCs, Lepird shows that techniques using mathematical programming and/or the stable marriage algorithm for developing personnel assignments in the Air Force are the most appropriate. In fact, Lepird's research seems to have been adopted, since the current method to match Air Force officers across different career fields to their operational assignments is through a combination of the stable marriage algorithm and a greedy algorithm. This is done several times a year for many of the career fields in the Air Force. Air Force career assignments, whether that means assigning the career fields themselves (the AFSC assignment problem) or matching personnel to operational assignments (what job the airmen will be doing within their career fields), are not the only occurrences of assignment problems in the Air Force. There are many other instances of Air Force assignment problems, from the assignment of Air Force Academy cadets to their final exam schedules [41] to the more conventional military weapon-target assignment problem [42], proving that proper implementation of modeling formulations, with an emphasis on the assignment problem itself, is critical to the success of the Air Force.

III. Methodology

"Values are fundamental to all that we do. Values are what we care about; and thus, values should be the driving force for our decision making." – Ralph L. Keeney

The model formulation methodology presented in this chapter is inspired through the understanding of both Value-Focused Thinking (VFT) and Mixed Integer Programming (MIP). We first discuss the methodology of the current optimization approach to motivate the need for change. We then propose a new VFT model of the Air Force Speciality Code (AFSC) assignment problem. A new objective hierarchy is created to distinguish decision maker objectives for the alternative solution assignments. With this VFT framework, we create two different optimization models that vary in complexity. An "exact-measured" model and an "approximate-measured" model are proposed to capture the full VFT framework and provide the best solutions for a particular class year instance, given certain weight and value parameters. In addition to these two optimization models, several solution techniques are presented. These different methods of generating solutions are further contrasted in Chapter IV.

3.1 Problem Description

The current process to assign cadets their Air Force Speciality Codes (AFSCs) uses an integer program (IP). AFPC implements the constraints and the objective function for this model with the intent of conforming the IP to the Air Force-defined "priorities." The first priority designated by AFPC is to meet the mandatory education requirements specified in the Air Force Officer Classification Directory (AFOCD), as well as the target quotas for the AFSCs. This "priority" is actually two separate objectives since the mandatory education requirements in the AFOCD do not ensure the correct number of cadets are assigned. Though a bit confusing, this is how the Air Force currently defines the problem. The AFOCD states which college degrees are eligible for each of the AFSCs, as well as what tier of eligibility a particular degree falls under for an AFSC. Table 1 displays an example specification within the AFOCD for 35P, the AFSC that designates the Public Affairs career field.

Each AFSC can have up to three degree qualification tiers. For a given tier, the target proportion of cadets with degrees qualified under that tier is designated as the target accession rate. The Classification of Instructional Programs (CIP) codes, handled by the National Center for Education Statistics, classify all kinds of college degrees throughout the nation. Not only can a very specific individual degree be designated by one code, the CIP code, but so too can degrees be generalized together by the first two or four numbers in the CIP code. This is indicated in Table 1 using the generic "X"s to specify a broader range of college degrees allowed under a given tier for 35P. The three distinct tier requirement levels are "Mandatory", "Desired",

Table 1: CIP code education matrix for AFSC 35P, "Public Affairs." The target refers to the expected proportion breakdown of cadets with degrees matching the CIP code that should be assigned to the AFSC. The CIP codes in this example represent more generalized fields of study but may also be explicitly defined for a specific degree. The requirement level represents the importance of meeting the target proportions for the tier.

Tier	Target	CIP	Program Description	Requirement
1	$\geq 70\%$	09.XXXX	Communication, Journalism and Related Programs	Mandatory
2	$\geq 20\%$	23.13XX	Rhetoric and Composition/Writing Studies	Desired
		42.XXXX	Psychology	
		45.09XX	International Relations and National Security Studies	
		45.10XX	Political Science and Government	
		52.14XX	Marketing	
3	$\leq 10\%$	XX.XXXX	Any Degree	Permitted

and “Permitted”. There can be anywhere from one to three degree tiers listed for an AFSC, however. In some situations, a career field only lists two tiers, with each tier having a mandatory requirement specification. Some career fields only have mandatory and desired tiers. This is another confusing Air Force priority, but it is still one that must be modeled. AFPC states that their first priority is to meet the mandatory tier accession rates for all of the AFSCs that have specified mandatory degree tier. For 35P, this means that at least 70% of the cadets that are accessed into Public Affairs need to have Tier 1 degrees.

The first priority for AFPC also includes meeting the target quotas for each AFSC. There are three types of quotas asserted: the USAFA quota, the ROTC quota, and the combined quota. The USAFA and ROTC quotas state the target number of cadets that are to be assessed into a given AFSC from the two commissioning sources. While reasonably redundant, the combined quota is simply the target total number of cadets that should be assessed, and it is the sum of the two sources of commissioning quotas. The quotas described here are simply the lower bound on the number of cadets allowed into the career field. There is an upper bound on the number of cadets that can be assessed, and it is determined by multiplying the target quota by the “over-classification” factor. This number is usually between 1 and 2, inclusive.

The second and third priorities are set only for “large” AFSCs. An AFSC is large if its combined target quota is greater than or equal to 40. The second priority is to balance the distribution of cadet merit across the AFSCs. In other words, ideally the average percentile (in terms of graduating order of merit) of cadets assessed into an AFSC should be around 0.50. No AFSC should be composed entirely of either high performers or low performers. Similar to the second, the third priority is to balance the sources of commissioning across the AFSCs. Each AFSC should, preferably, be composed of the same distribution of commissioning sources as the class itself. For example, if a graduating class year is composed of 75% ROTC cadets and 25% USAFA cadets, then the ideal commissioning source composition for each AFSC is also 75% ROTC cadets and 25% USAFA cadets. This is to prevent certain AFSCs from being composed entirely of new lieutenants from a single commissioning source.

There is discussion within the leadership of the Air Force for using this same kind of balancing strategy on the distributions of race and gender. Just as a class is composed of a certain percentage of ROTC and USAFA cadets, the class is also composed of a certain percentage of male and female cadets, as well as other racial or ethnic demographics. Ideally, the AFSCs individually should have these same demographic distributions. At the time of this paper's publication, AFPC does not account for these factors in their model. However, since this is a possible future consideration for the AFSC assignment problem, they are incorporated in this paper's new optimization model, but "weighted" at zero. These weights are discussed in Section 3.3. These additional objectives are mainly included as an example of how objectives can be easily incorporated into the new optimization model without changing the fundamental structure of that model.

For the fourth priority, the Air Force intends to meet the desired and permitted tier requirement accession rates for the AFSCs. The desired and permitted tier requirements are additional degree distribution goals. These tier requirements specify what the career field should be composed of (in terms of cadets with degrees in the certain tiers) after the mandatory degree requirement has been met. 35P, for example, would perfectly meet the AFOCD tier requirements if 70% of the cadets assessed have tier 1 degrees (established in the first priority), 20% have tier 2 degrees, and 10% have tier 3 degrees. Because of the way the target accession rates are defined, with $\geq$ and $\leq$ specifications, 35P would also meet its AFOCD objectives with a 80%/20%/0% split for mandatory, desired, and permitted tiers, respectively.

Lastly, the fifth priority is to maximize cadet preference. Cadets submit lists for the AFSCs they would like to be assigned to, in order of preference. They also submit utility measures associated with the AFSCs on their lists. These utility measures distinguish the cadets' levels of "happiness" they would receive should they be assigned to one of the AFSCs on their lists. Currently, every cadet has the option to submit up to six preferences for AFSCs, where their first choice always has a utility of 1, and every choice utility afterwards must be less than or equal to the previous one: $utility_1 \geq utility_2 \geq utility_3 \geq utility_4 \geq utility_5 \geq utility_6$. Assigning utilities to preferences, rather than simply using an ordinal preference list, allows for cadets to incorporate trade-offs between their AFSC preferences, informing the model of the cadets' true opinions on the importance of receiving one AFSC versus another.

3.2 Original Approach

This section describes the original methodology (AFPC's current formulation) to address the problem of assigning cadets to AFSCs. The mathematical formulation is discussed, along with the many issues present within the model's representation of the problem. The model struggles to pass both verification and validation, since it has many modeling issues that do not work as intended and its implemented constraints and objective function do not adequately portray the real problem.

3.2.1 Model Formulation

To capture the priorities specified in the previous section, AFPC currently utilizes an integer program that maximizes cadet/AFSC utility (calculated through a separate utility function) subject to several constraints. The mathematical formulation of AFPC's original integer program is presented on the following pages.

Sets and Indices

$i \in \mathcal{I}$	the set of all cadets
$j \in \mathcal{J}$	the set of all AFSCs
$\mathcal{L} \subseteq \mathcal{J}$	the set of all large AFSCs

Parameters		**Units**
$merit_i$	percentile of cadet i	[fraction]
$usafa_i$	1 if cadet i is a USAFA graduate; 0 otherwise	[-]
$utility_{ij}$	utility that cadet i assigned for AFSC j	[fraction]
$mandatory_{ij}$	1 if cadet i has a mandatory degree for AFSC j; 0 otherwise	[-]
c_{ij}	calculated cadet/AFSC utility of assigning cadet i to AFSC j	[fraction]
$target^{\mathrm{q}}_j$	the target quota for AFSC j	[number]
$over_j$	factor by which AFSC j can be over-classified	[fraction]
$target^{\mathrm{m}}_j$	target accession rate for mandatory degrees for AFSC j	[fraction]

Decision Variables		**Units**
x_{ij}	1 if cadet i is assigned to AFSC j; 0 otherwise	[-]

Original Formulation

$$\text{maximize} \quad \sum_{i \in \mathcal{I}} \sum_{j \in \mathcal{J}} c_{ij} x_{ij} \tag{7a}$$

$$\text{subject to} \quad \sum_{j \in \mathcal{J}} x_{ij} = 1 \quad \forall i \in \mathcal{I} \tag{7b}$$

$$\sum_{i \in \mathcal{I}} x_{ij} \geq target_j^q \quad \forall j \in \mathcal{J} \tag{7c}$$

$$\sum_{i \in \mathcal{I}} x_{ij} \leq over_j \cdot target_j^q \quad \forall j \in \mathcal{J} \tag{7d}$$

$$\sum_{i \in \mathcal{I}} mandatory_{ij} \cdot x_{ij} \geq target_j^m \cdot target_j^q \quad \forall j \in \mathcal{J} \tag{7e}$$

$$\sum_{i \in \mathcal{I}} usafa_i \cdot x_{ij} \geq 0.2 \cdot target_j^q \quad \forall j \in \mathcal{L} \tag{7f}$$

$$\sum_{i \in \mathcal{I}} usafa_i \cdot x_{ij} \leq 0.4 \cdot target_j^q \quad \forall j \in \mathcal{L} \tag{7g}$$

$$\sum_{i \in \mathcal{I}} merit_i \cdot x_{ij} \geq 0.35 \cdot target_j^q \quad \forall j \in \mathcal{L} \tag{7h}$$

$$\sum_{i \in \mathcal{I}} merit_i \cdot x_{ij} \leq 0.65 \cdot target_j^q \quad \forall j \in \mathcal{L} \tag{7i}$$

$$x_{ij} \in \{0, 1\} \quad \forall i \in \mathcal{I},\ j \in \mathcal{J} \tag{7j}$$

The utility measure, c_{ij}, is calculated as follows:

$$c_{ij} = \begin{cases} (10 \cdot percentile_i \cdot utility_{ij}) + 250, & \text{if mandatory degree} \\ (10 \cdot percentile_i \cdot utility_{ij}) + 150, & \text{if desired degree} \\ 10 \cdot percentile_i \cdot utility_{ij}, & \text{if permitted degree} \\ -50000, & \text{otherwise} \end{cases} \tag{8}$$

when AFSC j is a preference for cadet i, and as

$$c_{ij} = \begin{cases} 100 \cdot percentile_i, & \text{if mandatory degree} \\ 50 \cdot percentile_i, & \text{if desired degree} \\ 0, & \text{if permitted degree} \\ -50000, & \text{otherwise} \end{cases} \tag{9}$$

when AFSC j is not a preference for cadet i.

The objective function (7a) seeks to maximize the total sum of the AFPC defined AFSC/Cadet utilities presented in equations (8) and (9). The utility function's purpose is to meet as many of the Air Force priorities as possible, while valuing some over others. The utility function prescribes ineligible cadets for each AFSC a very large negative utility (-50,000), thus ensuring they are not assigned to that AFSC. Constraint (7b) ensures every cadet is assigned exactly one AFSC. Constraint (7c) forces the number of cadets assigned to each AFSC to be greater than or equal to the minimum target quota. In a similar manner as the previous constraint, constraint (7d) enforces the upper bound on the number of cadets assigned to each AFSC. Constraint (7e) forces each AFSC to meet the mandatory AFOCD degree tier minimum number of cadets. This constraint, along with the subsequent four constraints, pertain to the percent of the target number of cadets for the AFSCs, not the actual proportion of cadets assigned. This is discussed further in the following section. Constraints (7f) and (7g) attempt to balance the proportion of USAFA cadets for large AFSCs, while constraints (7h) and (7i) force the average percentiles of the cadets assigned to large AFSCs (relative to the AFSC quota) to be around 0.50.

3.2.2 Issues with the Formulation

Although the priorities are specified to an extent within the formulation, this model does not adequately represent the problem defined by the Air Force. It is, however, an improvement from the greedy method that was used in the past. The greedy method almost never maximized global cadet utility, or even a weighted sum of cadet utility (using the cadets' percentiles as the weights), and therefore is inferior to this integer program. Despite being unequivocally better than the greedy method, there is a lot of room for improvement. The one priority that is effectively represented within this model is the objective of meeting the AFSC quotas, represented by constraints (7c) and (7d), which capture the range on the number of cadets that can be assessed into each of the AFSCs. Constraint (7e), the mandatory degree requirement constraint, however, is difficult to satisfy. Because every class year is composed of different cadets with various AFSC qualifications, it is very common for this constraint alone to result in an infeasible model since there may not be enough qualified cadets for one or more of the AFSCs. This constraint almost always prevents the model from finding a feasible solution when combined with the other constraints as well. Because it is rarely possible to find a feasible solution when all constraints are present, this model needs to be formulated differently.

Another issue with this model can be found within constraints (7f) and (7g), the commissioning source constraints. The constraints on the number of USAFA cadets that can be assessed into a particular AFSC attempt to tackle two things at once: ensure a target minimum number of USAFA cadets and balance the proportion of those cadets within the AFSCs. The objective of balancing the proportion of USAFA cadets and the objective of meeting the USAFA and ROTC individual quotas are similar, but they are still separate objectives. For example, the Air Force can specify the need for two USAFA cadets and six ROTC cadets in one career field for a "combined" quota of eight. Suppose the breakdown of the entire class is 20% USAFA and 80% ROTC. To meet all three quotas (USAFA, ROTC, and combined) as well as balance the proportion according to the demographics of that class year, one solution could be to assign two USAFA cadets and eight ROTC cadets to that career field. This would meet the minimum required number of USAFA cadets (two) as well as the minimum number of ROTC cadets (six). Two additional ROTC cadets are assigned to bring the overall USAFA proportion back down to 20% for the given career field.

Another substantial issue with the model pertains to all four of the proportion balancing constraints, constraints (7f) – (7i). The commissioning source constraint needs to be separated into different constraints as discussed previously because it is incapable of balancing the true proportion of USAFA cadets assigned to an AFSC. To illustrate this, constraints (7f) and (7g) can be rewritten as:

$$0.2 \leq \frac{\sum_{i \in \mathcal{I}} usafa_i \cdot x_{ij}}{target_j^q} \leq 0.4 \quad \forall j \in \mathcal{L}. \tag{10}$$

Constraint (10) is identical to constraints (7f) and (7g). Rewriting the constraint this way is only meant to demonstrate that it does not actually depict the proportion of USAFA cadets. It is merely the proportion of USAFA cadets relative to the total quota for a given AFSC. This number is meaningless since there are many instances where the total number of cadets assigned to an AFSC is not equal to the target number. There could be more or, in some cases where it is infeasible to meet all quotas, less than the desired number of cadets assigned to an AFSC. If the target number of cadets for an AFSC is 20, and 30 cadets are assigned to it composed of 4 USAFA cadets and 26 ROTC cadets, this becomes a problem. The proportion of USAFA cadets in this instance is merely 0.13, but the proportion constraint, as it is defined now, is satisfied since the number of USAFA cadets divided by the target quota is $\frac{4}{20}$, or 0.20.

In a similar form, constraints (7h) and (7i) can be rewritten as:

$$0.35 \leq \frac{\sum_{i \in \mathcal{I}} percentile_i \cdot x_{ij}}{target_j^q} \leq 0.65 \quad \forall j \in \mathcal{L}. \tag{11}$$

This constraint has the potential of being particularly harmful to the quality of the solution since the model could find solutions where an AFSC is over-classified with lower performing cadets since quantity is equally as important as quality here. The sum of percentiles is measured relative to some fixed amount rather than the actual average, so the lines between solution qualities are blurred. Constraint (7e), the mandatory proportion constraint, also falls into these same issues since it is measured against the target quota rather than the number of cadets assigned to the AFSC. If the Air Force purely values meeting the AFOCD mandatory requirement in terms of ensuring that a certain number of cadets entering the career fields have mandatory tiered degrees, then this constraint works. However, if the real emphasis is on the proportion, then this constraint is not sufficient. Adding more under-qualified cadets to an AFSC that has already met its quota and would have met its mandatory requirement otherwise only dilutes the solution quality. The issues with these constraints alone necessitate deliberate reform of this model, but perhaps the most significant problem with this formulation is the utility function.

The utility function that generates the c matrix used in the objective function, as it is defined in the current formulation, only provides a guess at the benefit obtained in solution quality from the assignment of a given cadet to an AFSC. The structure of the function is arbitrary and does not reflect any actual measurable trade-off. The inclusion of degree qualifications (mandatory, desired, or permitted) in the utility function, while enforcing only the mandatory requirement in the constraints, is confusing. This seems to be a result of the difficulty in actually meeting those AFOCD requirements for all the AFSCs, so if only one of the requirements can be constrained then it should be the mandatory requirement. The establishment of arbitrary benefits to separate cadets with the three degree tiers creates trade-offs that do not accurately translate to the trade-offs between the proportions of cadets with those degrees assigned to each of the AFSCs. The utility value between a cadet and an AFSC does not reflect the true nature of the problem because it is based on whether or not the cadet has placed preference for that AFSC, which is an unwarranted distinction. In fact, the biggest issue with this model is not the formulation – it is the problem itself.

3.3 New Approach

This section discusses the need for a new problem description, as well as the mathematical formulation of the problem under that new description. We turn to decision analysis to help guide this formulation as a more defined decision problem. A value hierarchy is implemented to parameterize the values of the objectives themselves with associated weights. This results in an optimization model that captures the true trade-offs between solutions, similar to the way that cadets can specify trade-offs between the AFSCs they could be assigned to through their own utility measures.

3.3.1 Reforming the Problem

The main problem with the current approach to the assignment of cadets to career fields is that there is not enough information captured within the model formulation on the value of one solution versus another. The objective function in the original formulation (7a) maximizes a summation of arbitrary utility measures that have no real indication of model efficacy. Where one solution may meet one objective entirely, it may fail in another. A different solution may be the opposite. These solutions cannot be efficiently compared with objective (7a). This is a direct result of the confusing priorities that are to be balanced by the model. The Air Force defines priorities, objectives, and requirements almost interchangeably. There simply is not enough information to accurately distinguish the "hard" constraints from the softer ones. Going further, there is no accurate measure for evaluating the degree of importance between those priorities. Perhaps priority one is much more important than priority two, but just how sizeable is that difference? Additionally, for priority one, two separate objectives are listed – filling the AFSC quotas and meeting the mandatory degree tier requirements – with no distinction on which one is more important than the other. The individual USAFA and ROTC quotas are described in neither the AFPC priorities nor the model's formulation, despite being listed as separate quotas within the class year datasets. Are these quotas important, or should the focus be solely on balancing the distribution of cadets from separate commissioning sources? The current priorities do not effectively answer these questions, resulting in both an improper mathematical formulation and ill-defined problem. While good alternative solutions are generated from this model, far better solutions remain unknown due to a lack of proper value specification.

The current approach to the problem is an example of Alternative-Focused Thinking (AFT). Although priorities are somewhat defined, they do not accurately prescribe value to the potential alternatives that are to be generated. The model therefore can only "guess" at the true value of the solutions it finds, and does so incorrectly as a result of inadequate constraints. The utility matrix, c, used in the original objective function, is a great example of this. Although cadet utility is fixed (a cadet measures their own quality in a solution based on the AFSC they receive in the solution), an AFSC's "utility" is much more complicated, since it is based on the set of cadets that have been assigned to it. It cannot be accurately determined through a fixed utility matrix; it is dynamic by nature. This idea is also the correction to the issues presented in Section 3.2.2 regarding the constraints that divide by $target_j^q$. For example, constraint (10) should instead be formulated as:

$$0.2 \leq \frac{\sum_{i \in \mathcal{I}} usafa_i \cdot x_{ij}}{\sum_{i \in \mathcal{I}} x_{ij}} \leq 0.4 \quad \forall j \in \mathcal{L} \tag{12}$$

which can be rewritten in linear form as

$$\sum_{i\in\mathcal{I}} usafa_i \cdot x_{ij} \geq 0.2 \cdot \sum_{i\in\mathcal{I}} x_{ij} \quad \forall j \in \mathcal{L} \tag{13a}$$

$$\sum_{i\in\mathcal{I}} usafa_i \cdot x_{ij} \leq 0.4 \cdot \sum_{i\in\mathcal{I}} x_{ij} \quad \forall j \in \mathcal{L}. \tag{13b}$$

As discussed in Section 2.3.2, the solution to AFT is Value-Focused Thinking (VFT). Rather than utilize a combination of constraints and a static utility matrix for AFSCs with dynamic preferences, the mathematical formulation should instead appropriately measure solutions based on the inherent value of those solutions. First, we must define what is important in an assignment of cadets to AFSCs, through the use of objectives. Once objectives are determined, we must define metrics, or objective measures, with which to measure how well those objectives are met. Each objective measure must then be reduced to a value between zero and one so that different objectives may be compared using the same scale. Lastly, weights are placed on all objectives to distinguish importance between them. This is the essence of a VFT framework, and it is the baseline for this new assignment model.

3.3.2 Objective Hierarchy

Understanding the problem is the most important part of operations research. This concept is especially important in decision analysis. Without a proper framework with which to evaluate alternatives within some decision context, the analyst can never know for certain whether selecting a particular alternative is truly the best course of action. A careful and intentional consideration of the objectives to be met is paramount to the success of any decision support model. For this paper's problem, the fundamental objective is to maximize the overall solution quality of an assignment of cadets to their career fields. The fundamental objective can be

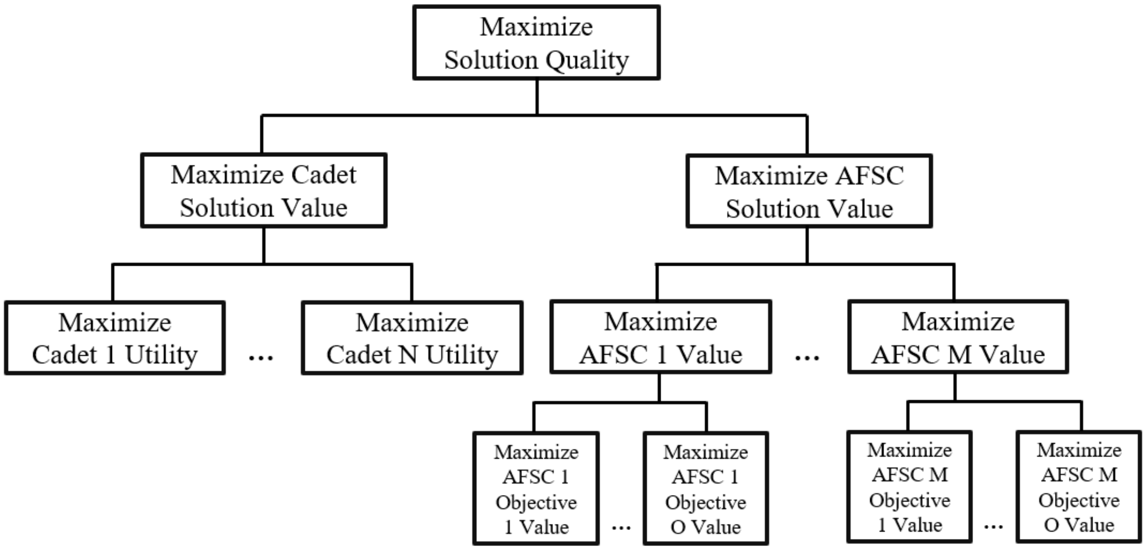


Figure 1: Model objective hierarchy. The quality of a solution assignment is a weighted sum of all of the different objectives placed on the solution, with weights placed on intermediary levels as well to distinguish importance between them. The AFSC objective values at the bottom of the tree structure are calculated from value functions using the measures of the different AFSC objectives (average merit is the measure for the "Balance Merit" objective, for example).

broken down into a composition of many other objectives for both cadets and career field managers (represented as AFSCs). This constitutes the objective hierarchy, illustrated in Figure 1. The two primary objectives that compose the fundamental objective are the maximization of cadet solution quality and the maximization of AFSC solution quality. Cadet solution quality is calculated as the weighted sum of all cadet utilities. Each cadet's utility is determined through the AFSC they are assigned to in a given solution. AFSC solution quality, on the other hand, is more complicated. Similar to the cadets, the overall AFSC solution quality is the weighted sum of all individual AFSC "utilities," or values. Each AFSC value is then the weighted sum of values for all of its individual objectives. These career field objectives are derived from the priorities listed previously.

We define 11 potential objectives to which AFSCs may ascribe importance (weights). For each AFSC, in no particular order, the first five objectives are to (i) balance the average merit (graduating rank percentile) of cadets, (ii) balance the proportion of USAFA cadets (and therefore ROTC cadets), (iii) meet the overall target quota, (iv) meet the USAFA quota, and (v) meet the ROTC quota. Differentiating the commissioning source objectives allows AFSCs to choose to either balance the proportion of USAFA cadets, or simply meet a minimum number of both USAFA and ROTC cadets. These objectives are mutually exclusive for each AFSC (they may either balance the USAFA proportion or specify quotas). Objectives six through eight are to meet the three tiered AFOCD requirements: the (vi) mandatory, (vii) desired, and (viii) permitted tiers. The next objective is to (ix) maximize the utility of the cadets assigned to each AFSC. The last two objectives are to balance the (x) male and (xi) minority proportions of cadets assigned to each AFSC. Although not currently used, they are included in the model to demonstrate the ease of creating new objectives without restructuring the entire problem formulation. Because all objective measures do not depend on each other (average merit does not depend on USAFA proportion, for example), we can assume the objectives themselves are independent. Objective independence is a necessary assumption for a VFT value model.

3.3.3 New Model Formulation

Due to the inherent nature of the problem, the 11 AFSC objectives can be measured directly by the characteristics of the solution. For example, the objective to "maximize cadet utility" can easily be measured through the average utility of the cadets assigned to an AFSC. These objective measures act as "auxiliary variables" to help make the formulation of this problem more legible, since they are derivatives of the decision variable, x_{ij}. Therefore, the mathematical formulation is defined on the following pages, where variables are emphasised in bold notation.

Sets and Indices

$i \in \mathcal{I}$	the set of all cadets
$j \in \mathcal{J}$	the set of all AFSCs
$k \in \mathcal{K}$	the set of all AFSC objectives
$\mathcal{K}^D \subseteq \mathcal{K}$	the set of AFSC objectives that seek some proportion of cadets with certain demographics. This set includes balancing "USAFA Proportion," "Male Proportion," "Minority Proportion," "Mandatory-Tier Proportion," "Desired-Tier Proportion," and "Permitted-Tier Proportion"
$\mathcal{I}^{\mathrm{C}} \subseteq \mathcal{I}$	the set of cadets with constrained minimum values
$\mathcal{I}^E_j \subseteq \mathcal{I}$	the set of cadets that are eligible for AFSC j
$\mathcal{I}^D_{jk} \subseteq \mathcal{I}^E_j$	the set of cadets with the demographic that corresponds to objective $k \in \mathcal{K}^D$ and are eligible for AFSC j. Depending on the value of k, this could be the set of USAFA, male, or minority cadets that are eligible for AFSC j or the set of cadets with mandatory, desired, or permitted tier degrees for AFSC j
$\mathcal{J}^{\mathrm{C}} \subseteq \mathcal{J}$	the set of AFSCs with constrained minimum values
$\mathcal{J}^E_i \subseteq \mathcal{J}$	the set of AFSCs for which cadet i is eligible
$\mathcal{K}^A_j \subseteq \mathcal{K}$	the set of objectives that apply to AFSC j
$\mathcal{K}^{\mathrm{C}}_j \subseteq \mathcal{K}^A_j$	the set of objectives with constrained minimum values for AFSC j

Fixed Parameters

$merit_i$	percentile of cadet i
$utility_{ij}$	utility that cadet i has assigned for AFSC j
$quota_j$	the target number of cadets for AFSC j

Value Parameters

$f_{jk}(\cdot)$	value function chosen by AFSC j for objective k
$target_{jk}$	target measure for objective k for AFSC j
ρ_{jk}	additional function parameters for objective k for AFSC j

Weight Parameters

$objective\_weight_{jk}$	the weight on objective k for AFSC j
$afsc\_weight_j$	the weight on AFSC j relative to the other AFSCs
$afscs\_overall\_weight$	the weight on the AFSCs as a whole relative to the cadets
$cadet\_weight_i$	the weight on cadet i relative to the other cadets
$cadets\_overall\_weight$	the weight on the cadets as a whole relative to the AFSCs

Constraint Parameters

$objective\_min_{jk}$	the minimum measure of objective k for AFSC j
$objective\_max_{jk}$	the maximum measure of objective k for AFSC j
$afsc\_value\_min_j$	the minimum value of AFSC j
$afscs\_overall\_value\_min$	the minimum overall AFSC value
$cadet\_value\_min_i$	the minimum value of cadet i
$cadets\_overall\_value\_min$	the minimum overall cadet value

Decision Variables

$\boldsymbol{x}_{ij}$	1 if cadet i is assigned to AFSC j, and 0 otherwise

Auxiliary Variables

$\boldsymbol{count}_j$ — number of cadets assigned to AFSC j,

$$\sum_{i \in \mathcal{I}_j^E} \boldsymbol{x}_{ij}$$

$\boldsymbol{usafa\_count}_j$ — number of USAFA cadets assigned to AFSC j,

$$\sum_{i \in \mathcal{I}_{jk}^D} \boldsymbol{x}_{ij} \text{ where } k = \text{“USAFA Proportion”}$$

$\boldsymbol{rotc\_count}_j$ — number of ROTC cadets assigned to AFSC j,

$$\boldsymbol{count}_j - \boldsymbol{usafa\_count}_j$$

$\boldsymbol{measure}_{jk}$ — real measure of objective k for AFSC j,

$$\begin{cases} \dfrac{\sum_{i \in \mathcal{I}_{jk}^D} \boldsymbol{x}_{ij}}{\boldsymbol{count}_j}, & \text{if } k \in \mathcal{K}^D \\ \dfrac{\sum_{i \in \mathcal{I}_j^E} merit_i \cdot \boldsymbol{x}_{ij}}{\boldsymbol{count}_j}, & \text{if } k = \text{“Balance Merit”} \\ \dfrac{\sum_{i \in \mathcal{I}_j^E} utility_{ij} \cdot \boldsymbol{x}_{ij}}{\boldsymbol{count}_j}, & \text{if } k = \text{“Maximize Utility”} \\ \boldsymbol{count}_j, & \text{if } k = \text{“Meet Combined Quota”} \\ \boldsymbol{usafa\_count}_j, & \text{if } k = \text{“Meet USAFA Quota”} \\ \boldsymbol{rotc\_count}_j, & \text{if } k = \text{“Meet ROTC Quota”} \end{cases}$$

$\boldsymbol{value}_{jk}$ — value of objective k for AFSC j obtained through value function $f_{jk}(\cdot)$,

$$f_{jk}(\boldsymbol{measure}_{jk}, target_{jk}, \rho_{jk})$$

Auxiliary Variables (continued)

$\boldsymbol{afsc\_value}_j$	the weighted sum of all objective values for AFSC j, $\sum_{k \in \mathcal{K}_j^A} objective\_weight_{jk} \cdot \boldsymbol{value}_{jk}$
$\boldsymbol{afscs\_overall\_value}$	the weighted sum of all AFSC values, $\sum_{j \in \mathcal{J}} afsc\_weight_j \cdot \boldsymbol{afsc\_value}_j$
$\boldsymbol{cadet\_value}_i$	the utility obtained for cadet i, $\sum_{j \in \mathcal{J}_i^E} utility_{ij} \cdot \boldsymbol{x}_{ij}$
$\boldsymbol{cadets\_overall\_value}$	the weighted sum of all cadet utilities, $\sum_{i \in \mathcal{I}} cadet\_weight_i \cdot \boldsymbol{cadet\_value}_i$
$\boldsymbol{Z}$	objective value, $afscs\_overall\_weight \cdot \boldsymbol{afscs\_overall\_value} +$ $cadets\_overall\_weight \cdot \boldsymbol{cadets\_overall\_value}$

VFT Formulation

$$\text{maximize} \quad \boldsymbol{Z} \tag{14a}$$

$$\text{subject to} \quad \sum_{j \in \mathcal{J}_i^E} \boldsymbol{x}_{ij} = 1 \quad \forall i \in \mathcal{I} \tag{14b}$$

$$\boldsymbol{measure}_{jk} \geq objective\_min_{jk} \quad \forall j \in \mathcal{J},\ k \in \mathcal{K}_j^{\text{C}} \tag{14c}$$

$$\boldsymbol{measure}_{jk} \leq objective\_max_{jk} \quad \forall j \in \mathcal{J},\ k \in \mathcal{K}_j^{\text{C}} \tag{14d}$$

$$\boldsymbol{afsc\_value}_j \geq afsc\_value\_min_j \quad \forall j \in \mathcal{J}^C \tag{14e}$$

$$\boldsymbol{cadet\_value}_i \geq cadet\_value\_min_i \quad \forall i \in \mathcal{I}^C \tag{14f}$$

$$\boldsymbol{afscs\_overall\_value} \geq afscs\_overall\_value\_min \tag{14g}$$

$$\boldsymbol{cadets\_overall\_value} \geq cadets\_overall\_value\_min \tag{14h}$$

$$\boldsymbol{x}_{ij} \in \{0, 1\} \quad \forall i \in \mathcal{I},\ j \in \mathcal{J}_i^E \tag{14i}$$

The objective function (14a) maximizes the weighted sum of all objective criteria for both AFSCs and cadets. This value can be interpreted as the percent of a perfect solution obtained, since an objective value of 1 would mean that all objectives are met. Constraints (14b) and (14i) ensure every cadet receives exactly one AFSC for which they are eligible. One key assumption is that every cadet is eligible for at least one AFSC. Despite being in the objective function, some AFSC objectives may be much harder to meet than others and therefore must be constrained. Constraints (14c) and (14d) allow the decision-maker (DM) to force certain AFSC objective criteria to fall within some range. Constraints (14e) and (14f) force certain AFSCs and/or cadets to have minimum objective values. Similarly, constraints (14g) and (14h) restrict the overall value of all the AFSCs and all the cadets, respectively. Adjusting the constraints for an objective rather than the weights does not change the objective hierarchy in any way, and therefore we are able to compare a constrained solution using the same VFT "lens" as an unconstrained solution. Section 4.3 discusses how we can conduct sensitivity analysis on both the weights and the constraints.

3.4 Optimization Model Additions

The VFT mathematical formulation presented in the previous section is in its most generalized form. The value functions are not yet defined to provide a generalized approach to the problem. The only decision variable listed is $\boldsymbol{x}$, although depending on the available value functions used in the VFT framework for the AFSCs' objectives, there could be a need for more decision variables. These decision variables would therefore be based solely on the degree of complexity of those value functions. This section discusses the main issue with this new model and the resulting need for two distinct optimization models, as well as the implementation of different value function strategies to model the DM's affinity for various AFSC objective measures.

3.4.1 Model Complexity

The current formulation described in Section 3.3.3 is non-convex. The expression

$$\frac{\sum_{i \in \mathcal{I}_j^E} merit_i \cdot \boldsymbol{x}_{ij}}{\boldsymbol{count}_j} \tag{15}$$

in the objective function alone induces non-convexity due to the presence of variables in the numerator and the denominator that cannot be factored out. For the purposes of finding a solution quicker, $\boldsymbol{count}_j$, the sum of all cadets assigned to AFSC j, can be approximated with $quota_j$, the target quota to fill for each AFSC. Doing so results in a linearization of the problem. This is not exactly the same idea as is criticized with the original model in Section 3.2.2. The original model applies this approximation in the constraints, where it is not necessary. Here, we are approximating the number of cadets assigned to each AFSC in the objective function with the sole purpose of making that function linear and convex, thereby significantly reducing the complexity of the problem.

No matter what optimization model is used to generate a solution, we will always measure that solution using the main VFT framework. Thus, the only objective value considered is the one which uses the exact non-convex objective function. For the remainder of the paper, we will refer to the full non-convex model as the "Exact" model and the model that uses the convex approximation of $\boldsymbol{count}_j$ with $quota_j$ as the "Approximate" model. Approximation, in this case, refers to the optimization model's calculations of the objective measures, and should not be confused with the linear approximation technique used on exponential value functions described in Section 3.4.3. Additionally, the value function methodology presented in the following two sections apply to both models, since the only difference between the two is the method of calculating $\boldsymbol{measure}_{jk}$.

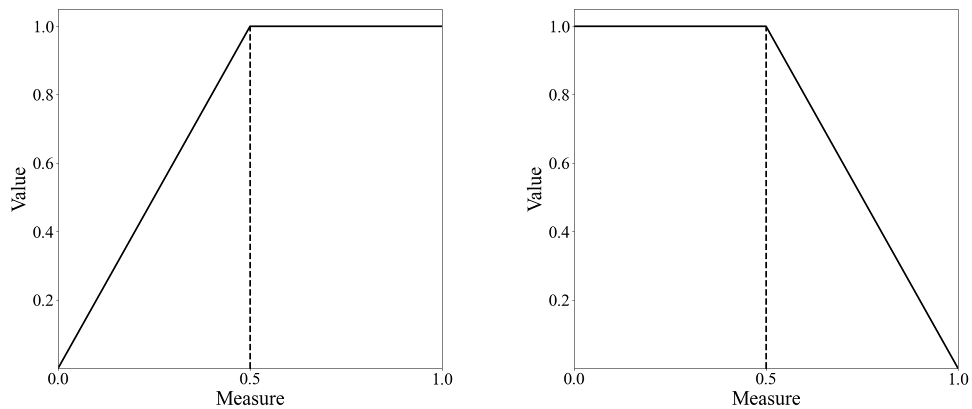
3.4.2 Simple Linear Value Functions

The goal of the value functions for the AFSC objectives is to model the DM's trade-offs between different measures within a particular AFSC objective as closely as possible. There are several strategies on how to do this for value functions within an additive value model discussed in [27]. For this research we consider piece-wise linear functions, since all objectives can be measured, or at least approximately represented, using a continuous scale. If we restrict the set of available value functions to simple minimization functions, we can formulate the VFT model in a simplified manner. One such function for the AFSC assignment problem is

$$\text{minimum}\left(\frac{\boldsymbol{measure}_{jk}}{target_{jk}},\quad 1\right). \tag{16}$$

For the objectives which have measures that exist on a continuous 0-1 scale, such as the cadet proportion or utility/merit average objectives, this function allows the DM to specify a target measure with indifference. If the actual objective measure exceeds this target, the value received is still 1 because the target has been met. This is the "increasing min" value function and is useful for quota objectives, mandatory AFOCD tier objectives, and utility objectives. The DM could just as easily specify a "decreasing min" value function that describes the measures that they are indifferent towards below some target. These kinds of functions are particularly useful for the permitted tier AFOCD requirement objectives. Both example functions are depicted in Figure 2 using target measures of 0.50.

To represent the *min* functions in the mathematical formulation, the variable $\boldsymbol{over}_{jk}$ is introduced. For the increasing min function, this variable is equal to the amount, in the "measure"-space, by which we exceed the objective target. For example, consider an increasing *min* function for some objective k for AFSC j with a



(a) Increasing (b) Decreasing

Figure 2: Minimization value functions. These functions both use a target measure of 0.5. The function on the left indicates an upward linear trend in value for measures below 0.5. The function is indifferent towards measures at or above 0.5, and they all receive a value of 1. The function on the right is the opposite, where measures below 0.5 are preferred to measures above that target.

target measure, $target_{jk}$, of 0.5. If the actual measure is 0.4 ($\boldsymbol{measure}_{jk}$), then the objective target has not been exceeded ($\boldsymbol{over}_{jk} = 0$). If the measure is 0.6, the objective measure exceeds the target by 0.1 ($\boldsymbol{over}_{jk} = 0.1$). The value obtained through the increasing *min* function for some objective measure can now be expressed as

$$\frac{\boldsymbol{measure}_{jk} - \boldsymbol{over}_{jk}}{target_{jk}}. \tag{17}$$

The decreasing *min* function is very similar, but the indifference is flipped. This variable, $\boldsymbol{over}_{jk}$, now represents the amount by which the target exceeds the measure. Using a target of 0.5, if the actual measure is 0.4 ($\boldsymbol{measure}_{jk}$), then the objective target exceeds the measure by 0.1 ($\boldsymbol{over}_{jk} = 0.1$). If the measure is 0.6, the objective target does not exceed the measure ($\boldsymbol{over}_{jk} = 0$). For this function, the value obtained from some AFSC objective measure would instead be expressed as

$$\frac{1 - \boldsymbol{measure}_{jk} - \boldsymbol{over}_{jk}}{1 - target_{jk}}. \tag{18}$$

To ensure the variable $\boldsymbol{over}_{jk}$ is the correct amount, two new constraints are introduced, constraints (19a) and (19b), where (19a) depends on the value function chosen and the two options are therefore mutually exclusive. These constraints are

$$\left\{ \begin{array}{ll} target_{jk} \geq \boldsymbol{measure}_{jk} - \boldsymbol{over}_{jk} & \forall j \in \mathcal{J},\ k \in \mathcal{K}_j^A, \quad \text{if } f_{jk}(\cdot)\ is\ f^1 \\ target_{jk} \leq \boldsymbol{measure}_{jk} + \boldsymbol{over}_{jk} & \forall j \in \mathcal{J},\ k \in \mathcal{K}_j^A, \quad \text{if } f_{jk}(\cdot)\ is\ f^2 \end{array} \right\} \quad (19a)$$

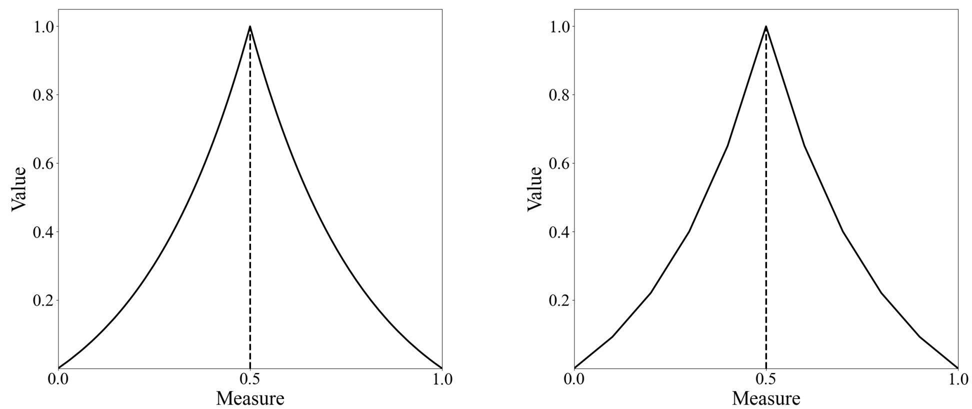
$$\boldsymbol{over}_{jk} \geq 0 \quad \forall j \in \mathcal{J},\ k \in \mathcal{K}_j^A \quad (19b)$$

where f^1 refers to the increasing *min* function and f^2 denotes the decreasing *min* function.

3.4.3 Complicated Piece-Wise Value Functions

While the value function methodology presented in the previous section may be adequate in some situations, it is very limiting in its ability to capture more complicated value trade-offs. Perhaps the rate at which value increases changes based on the ranges of the measures. Certain intervals on the x-axis ($\boldsymbol{measure}_{jk}$) are associated with steeper slopes than others. Maybe the function plateaus along some desired range, but has a sharp descent on either side. There is potential, too, for the underlying value relationship within some objective to be inherently non-linear. However, because these value functions are merely meant to provide estimates of the true values received through different objective measures, any non-linear value function can be approximated with piece-wise linear value functions for faster solve times.

Consider the exponential value functions shown in Figure 3. The left graph (a) depicts the complete non-linear function, while the right (b) shows the linear approximation of that function. Although difficult to see, the right graph contains 10 line segments with edges at each increment of 0.10 in the x-axis. Utilizing piece-wise value



(a) Non-Linear

(b) Linear

Figure 3: Exponential function with piece-wise approximation. The left function (a) depicts an actual exponential function, while the right function (b) depicts a linear piece-wise approximation of that exponential function.

functions within this optimization model allows for complete specification of any kind of value trade-off for the AFSC objectives. The DM is not locked into predefined sets of functions; they are free to design any function they wish. Additionally, the structure of the model formulation does not change depending on the kind of function used. Whereas the use of the ***over*** variable in the previous section can only be used for very specific kinds of functions, the model formulation additions discussed in this section can be applied to any function with linear segments.

The formulation additions we make to the optimization model are derived from "Applied Integer Programming" [43]. To implement this kind of function, we must break it into linear segments using "breakpoints." The new parameters for the optimization model are therefore the coordinates for each breakpoint. In the context of this problem, the DM provides the values achieved for some objective at various objective measures, thus deriving the breakpoints of the function. To implement this piece-wise methodology, we introduce formulation additions to the VFT optimization model. An example is also presented in Section 3.5.1 to further illustrate this concept.

Additional Set

$l \in \mathcal{L}_{jk}$	set of all breakpoints for objective k's function for AFSC j

Additional Parameters

r_{jk}	number of breakpoints for objective k's function for AFSC j
a_{jkl}	measure at breakpoint l for objective k's function for AFSC j
$\hat{f}_{jkl}$	value at breakpoint l for objective k's function for AFSC j

Additional Variables

$\boldsymbol{\lambda}_{jkl}$	percentage of the line segment between breakpoints l and $l+1$ that $\boldsymbol{measure}_{jk}$ has yet to travel along the piece-wise function
$\boldsymbol{y}_{jkl}$	1 if $\boldsymbol{measure}_{jk}$ is on the line segment between breakpoints l and $l + 1$; 0 otherwise

Additional Constraints

$$\boldsymbol{measure}_{jk} = \sum_{l \in \mathcal{L}_{jk}} a_{jkl} \cdot \boldsymbol{\lambda}_{jkl} \quad \forall j \in \mathcal{J},\ k \in \mathcal{K}^A_j \tag{20a}$$

$$\boldsymbol{value}_{jk} = \sum_{l \in \mathcal{L}_{jk}} \hat{f}_{jkl} \cdot \boldsymbol{\lambda}_{jkl} \quad \forall j \in \mathcal{J},\ k \in \mathcal{K}^A_j \tag{20b}$$

$$\boldsymbol{\lambda}_{jk1} \leq \boldsymbol{y}_{jk1} \quad \forall j \in \mathcal{J},\ k \in \mathcal{K}^A_j \tag{20c}$$

$$\boldsymbol{\lambda}_{jkl} \leq \boldsymbol{y}_{jk(l-1)} + \boldsymbol{y}_{jkl} \quad \forall j \in \mathcal{J},\ k \in \mathcal{K}^A_j,\ l \in \{2, ..., r_{jk} - 1\} \tag{20d}$$

$$\boldsymbol{\lambda}_{jkr_{jk}} \leq \boldsymbol{y}_{jk(r_{jk}-1)} \quad \forall j \in \mathcal{J},\ k \in \mathcal{K}^A_j \tag{20e}$$

$$\sum_{l=1}^{r_{jk}-1} \boldsymbol{y}_{jkl} = 1 \quad \forall j \in \mathcal{J},\ k \in \mathcal{K}^A_j \tag{20f}$$

$$\sum_{l \in \mathcal{L}_{jk}} \boldsymbol{\lambda}_{jkl} = 1 \quad \forall j \in \mathcal{J},\ k \in \mathcal{K}^A_j \tag{20g}$$

$$0 \leq \boldsymbol{\lambda}_{jkl} \leq 1 \quad \forall j \in \mathcal{J},\ k \in \mathcal{K}^A_j,\ l \in \mathcal{L}_{jk} \tag{20h}$$

$$\boldsymbol{y}_{jkl} \in \{0, 1\} \quad \forall j \in \mathcal{J},\ k \in \mathcal{K}^A_j,\ l \in \{1, ..., r_{jk} - 1\} \tag{20i}$$

Constraints (20a) and (20b) force the objective measures and values to fall on some point along one of the line segments of the piece-wise value functions. Constraints (20c) – (20i) enforce the value of a particular objective measure to take on the value of the point between two breakpoints by using its slope. The $\boldsymbol{\lambda}$ variable determines the percentage in the x-space (the $\boldsymbol{measure}$ axis) that $\boldsymbol{measure}_{jk}$ has left to travel between two breakpoints. For example, if $\boldsymbol{measure}_{jk}$ is 30% of the way between breakpoints 2 and 3, then $\boldsymbol{\lambda}_{jk2}$ would be 0.7, and $\boldsymbol{y}_{jk2}$ would be 1. Since $\sum_{l \in \mathcal{L}_{jk}} \boldsymbol{\lambda}_{jkl} = 1$, $\boldsymbol{\lambda}_{jk3}$ would be 0.3, but have no effect on $\boldsymbol{value}_{jk}$ since $\boldsymbol{y}_{jk3}$ would be 0. Therefore this variable can take on continuous values between 0 and 1, inclusive.

The $\boldsymbol{\lambda}$ variable is also used to determine the percentage between two breakpoints in the y-space (the $\boldsymbol{value}$ axis) that this measure corresponds to. Because the segment is linear, $\boldsymbol{\lambda}$ can be calculated using $\boldsymbol{measure}$ as defined in the VFT model formulation in Section 3.3.3, as well as with the parameter a, alongside the other constraints. Once the distance along the x-space (the $\boldsymbol{measure}$ axis) between two breakpoints is used to locate a particular objective measure with respect to its value function, this translates to the distance in the y-space, and ultimately the value of that objective. To enforce the condition that "at most two adjacent $\boldsymbol{\lambda}_{jkl}$ can be positive," as described in [43], constraints (20c) – (20e) are defined. These constraints limit the variable $\boldsymbol{\lambda}$ to only be positive when the line segment on either side of its corresponding breakpoint is activated. Constraint (20f) forces only one line segment to be activated, so the sum of all $\boldsymbol{y}_{jkl}$ variables (for each AFSC objective) must be 1. The sum of all the $\boldsymbol{\lambda}_{jkl}$ variables for a particular AFSC objective value function must also be 1, since the largest percentage along one line segment that $\boldsymbol{measure}_{jk}$ could be is 1. This is specified in constraint (20g). Constraints (20h) and (20i) define the variable domains.

3.5 Solution Methodology

This section first presents an example value function calculation using the "complicated" function additions discussed in the previous section. This value function methodology is ultimately used in the two full optimization models (Approximate and Exact). We then discuss different methods to solve the two optimization models using a combination of conventional optimization solvers and heuristics.

3.5.1 Value Function Example

To illustrate the formulation modifications presented in Section 3.4.3, Figure 4 depicts an example value function that contains 14 breakpoints. Breakpoint 2 is shown in blue. The objective measure and value corresponding to breakpoint 2 (a_{jk2} and $\hat{f}_{jk2}$, respectively) are listed in the axes. Line segment 2 is also highlighted in blue.

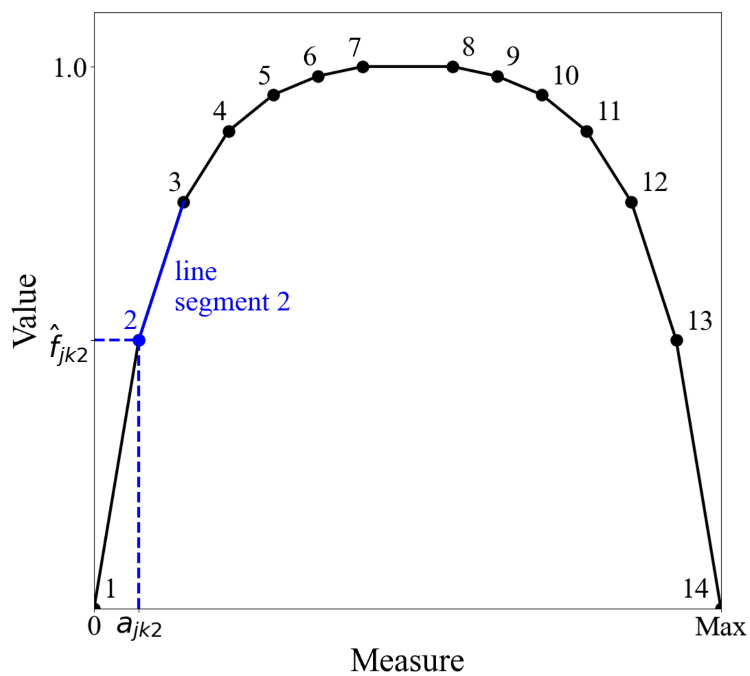


Figure 4: Example value function using 14 breakpoints. The line segment between breakpoints 2 and 3 is shown in blue, and breakpoint 2 is also highlighted in blue. The function parameters corresponding to this breakpoint are labeled in the axes.

All other breakpoints are shown in black. Suppose one particular AFSC objective value function has a domain of $\boldsymbol{measure}_{jk} \in \{0, 1, 2, ..., 140\}$. This is likely a quota objective, since they are measured by the number of cadets assigned to the AFSC, and therefore take on integer values. All objective value functions contain the range of $0 \leq \boldsymbol{value}_{jk} \leq 1$. Using the function depicted in Figure 4, the maximum value of 1 is achieved for measures between 60 and 80. Suppose also for this objective that a measure of 14 is achieved. For the combined quota objective, this would mean 14 cadets are assigned to an AFSC, but the target is 60. This is shown in Figure 5, however a real quota value function would be far less forgiving since failing to meet the target quota is very unacceptable (especially when failed by 46 cadets).

An objective measure of 14, in this case, falls on line segment 2, which is the line segment connecting breakpoints 2 and 3. Breakpoint 2 has a measure of 10, and

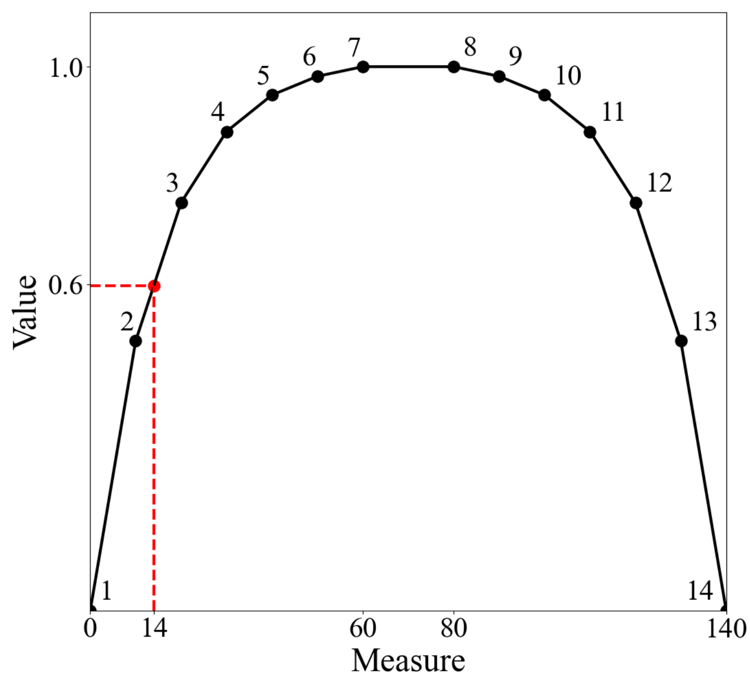


Figure 5: Example value function measured using 14 breakpoints. An objective measure falls between breakpoints 2 and 3. This particular objective measure of 14 has an associated value of 0.6.

breakpoint 3 has a measure of 20. Since $\frac{14-10}{20-10} = 0.4$, this point is 40% of the way along line segment 2. We can then use this information to find the value at this particular measure. The value at breakpoint 2 is 0.5, and the value at breakpoint 3 is 0.75. The objective value can be calculated directly as

$$\boldsymbol{value}_{jk} = 0.4(0.75 - 0.5) + 0.5 = 0.6. \tag{21}$$

This is how $\boldsymbol{\lambda}$ and $\boldsymbol{y}$ are used. The variable, $\boldsymbol{\lambda}_{jk2}$, is the percentage along line segment 2 that $\boldsymbol{measure}_{jk}$ has yet to travel. In other words, if q is the percent distance between breakpoints 2 and 3 where $\boldsymbol{measure}_{jk}$ is located, then $\boldsymbol{\lambda}_{jk2}$ is $1-q$. In this example, q is 0.4 so $\boldsymbol{\lambda}_{jk2}$ is therefore 0.6. Since this point is on line segment 2, $\boldsymbol{y}_{jk2}$ is 1. The summation over all $\boldsymbol{\lambda}$ values for this value function must be 1, and therefore $\boldsymbol{\lambda}_{jk3}$ is 1 - $\boldsymbol{\lambda}_{jk2}$, or 0.4. We can rewrite Equation 21 as

$$\boldsymbol{value}_{jk} = 0.5(0.6) + 0.75(0.4) = 0.6 \tag{22}$$

which is the same way that $\boldsymbol{value}_{jk}$ is calculated in constraint (20b).

3.5.2 Solving the Model

Given N cadets and M AFSCs, and assuming all cadets are eligible for all AFSCs, a solution assignment can be represented in an $N \times M$ binary matrix. There are a total of $(2^M)^N$ possible alternative solutions, almost all of which are infeasible. For solutions satisfying constraint (14b), the "one-job only" constraint in which all rows sum to 1, there are M^N possible solutions. Let M_i represent the number of AFSCs for which cadet i is eligible. This is equivalent to the number of AFSCs within $\mathcal{J}_i^E$. There are then $\prod_{i=1}^{N} M_i$ possible solutions when we factor in cadet eligibility. A real instance of this problem could contain 1600 cadets and 32 AFSCs. If we assume every cadet is eligible for 20 of those 32 AFSCs, there are then 20^{1600} possible solutions. Therefore, this optimization model requires an efficient solution technique.

To solve the two optimization models, we use the Python library "Pyomo" with applicable solvers [44]. For the Approximate model, the mixed integer linear programming (MILP) solvers considered are "COIN-OR Branch and Cut" (CBC) and Gurobi. For the mixed integer non-linear programming (MINLP) solvers, we consider the "interior point optimizer" (IPOPT) and Baron. CBC and IPOPT are both open-source "COIN-OR" solver packages, and are therefore very useful for implementing this model in any environment since commercial licenses are not necessary [45]. Baron and Gurobi are commercial solvers and should be expected to outperform their open-source counterparts based upon convergence on optimality. They are both global solvers, which means they will solve the models to global optimality.

While certainly not the focus of this thesis, choosing the right solution technique to solve a particular optimization problem is critical to the model's implementation. The optimization model presented in Section 3.3.3 is first and foremost a decision support tool, and should be used as such. The model is very sensitive to its parameters, and changing the importance of certain objectives could produce an entirely different solution. Therefore, this model needs to produce solutions quickly, so weight and value parameters can be altered allowing for new solutions to be generated and subsequently compared against each other. For this reason, we propose a genetic algorithm (GA) that can be implemented to take a solution generated by the Approximate model and "evolve" it using the Exact model's objective function as the fitness function. The details of the structure of the GA are presented in Appendix D. In addition to initializing the GA with the optimization models' solutions, we also incorporate two other algorithms for the purposes of generating initial solutions. These algorithms include a greedy method and a stable marriage algorithm adaptation for this problem.

These two algorithms use the VFT framework to create an AFSC utility matrix. This utility matrix incorporates the AFSC objective weights to create utility measures for each cadet based on the AFSC objectives that the cadet contributes to. For example, a cadet with a mandatory-tiered degree would add to the proportion of cadets with mandatory degrees, and therefore affect that objective. For the greedy method, we create an overall joint cadet/AFSC utility matrix by factoring in the cadet/AFSC individual weights and overall weights, along with the cadet utility matrix (the *utility* parameter as presented in Section 3.3.3). We then assign each cadet to the AFSC with the highest overall utility value. Appendix C presents the method of these utility matrix calculations. The only similarity this method shares to the one discussed in Section 3.2.1 is that a joint utility matrix is used to approximate the overall value of assigning a cadet to an AFSC.

Both algorithms are purely meant to provide quick initial solutions that are better than random ones for the GA to use in addition to the optimization models' solutions. For the stable marriage algorithm, we use the cadet and AFSC utility matrices discussed above to create cadet and AFSC preference lists. We then apply the hospital and residents variant of the stable marriage algorithm to assign cadets their AFSCs [18]. Both algorithms are very quick and provide good initial solutions to use alongside the optimization models' solutions. For the Exact model, we can use its solution, the Approximate model's solution (since it is generated in less than 10 seconds), the greedy method solution, and the stable marriage solution all together in the GA's initial population. The results of these techniques are presented in Section 4.1.2.

IV. Results and Analysis

This chapter presents the results of the new Value-Focused Thinking (VFT) assignment model. We first determine which solvers to use for both optimization models. Once the solvers are determined, we apply different techniques using heuristics on synthetic datasets generated using a conditional tabular generative adversarial network (CTGAN) [46]. The details on how we implement the CTGAN model to acquire these synthetic datasets are presented in Appendix B. We then solve the problem instances generated by CTGAN using both the "Approximate" model and the "Exact" model. A genetic algorithm (GA) is implemented using the solutions from the two models as initial solutions. We then present the results from the new VFT method for a real class year instance. A methodology for conducting sensitivity analysis is formulated and applied on the example instance. Finally, additional real class year data instances are evaluated and the original solutions are compared against new solutions generated using many different sets of value parameters.

4.1 Optimization Model Performance

To select a solution strategy, we test the two optimization models on performance (time and solution value) using different solvers on different problem sizes. We then choose a solver for each model to generate initial solutions to the models. A GA is applied to further evolve those solutions using the Exact model VFT framework and the same performance metrics are compared once more. The solution methodologies are compared to determine the best strategy for finding an adequate solution in a small amount of time. The value hierarchy must be tuned to reflect the true values of the decision maker, and so speed is critical. Finding a good solution to the correct model is much better than finding the best solution to the incorrect model.

4.1.1 Solver Results

An optimization model's solution methodology, in addition to its mathematical formulation, must be adequately specified. In many cases, selecting a conventional solver within some optimization software is all that is required. Due to the two different optimization models presented in this paper (which are almost identical, with the only difference being the way that some AFSC objective measures are calculated), further analysis is needed to determine the best way of producing solutions under a particular VFT framework. The first step is to pick a conventional optimization solver for each model. To do this, we use Python's optimization library "Pyomo" [44]. For the Approximate model, we consider the mixed integer linear programming (MILP) solvers "COIN-OR Branch and Cut" (CBC) and Gurobi. For the Exact model we use the mixed integer non-linear programming (MINLP) solvers "interior point optimizer" (IPOPT) and Baron.

The solvers for both models are tested on different problem sizes. The smallest instance tested includes 100 cadets and 2 AFSCs. For these smaller instances, we randomly generate the preferences and utilities but use historical trends to determine the cadets' demographics and qualifications. We consider this data the "fixed" model parameters, since this data includes the cadet-specific information that cannot be changed. Once we have the fixed parameters, we are then able to ascribe value parameters. The value parameters contain the information specific to the VFT framework. These parameters include the weights on all of the objectives (at each level of the objective hierarchy) as well as the value functions used for the AFSC objectives. To further separate the problem instances, different sets of value parameters are used on each of the cadet fixed datasets. These sets are tailored to the instance-specific parameters, and are generated randomly using estimates for the Air Force's real value framework as discussed in Appendix E.

For the remainder of this section, we denote the problem sizes as $N \times M$, where N is the number of cadets in the instance and M is the number of AFSCs. The set of instance sizes used for testing is $\{(100 \times 2),\ (200 \times 2),\ (400 \times 2),\ (800 \times 2),\ (1200 \times 2),\ (1600 \times 2),\ (1600 \times 4),\ (1600 \times 8),\ (1600 \times 12),\ (1600 \times 24),\ (1600 \times 32)\}$. Since there are currently 32 non-rated AFSCs, a realistic instance of this problem would have size $N \times 32$, where N is usually between 1200 and 1800. For the solver tests, five instances are generated for each problem size and a random set of value parameters is assigned to the instance. We then solve each instance using the approximate model with CBC and Gurobi, and then using the Exact model with IPOPT and Baron. To prevent solving the instances for an extensive amount of time, we iterate through solving the instances with different maximum time parameters. This is used to determine the time of convergence, since the solver would often continue solving an instance for an extended amount of time but report the same solution that it found earlier.

On the first instance tested using a problem size of 100×2, all solvers reported solutions in a few seconds except for one. The Baron solver continued solving its instance for over ten minutes, which is surprising for such a small problem instance. Because Baron did not report a solution to the 100×2 instances in a reasonable amount of time, it is removed from this testing and not considered as a viable solver for the Exact model. Additionally, some solvers fail to consistently report integer solutions. Both the $\boldsymbol{x}$ and $\boldsymbol{y}$ variables must be binary, but due to the solvers' integrality restrictions and the maximum time parameters this is not always the case. Since IPOPT uses an interior-point method to find a local solution to a non-linear programming (NLP) problem, it does not guarantee integer feasibility [47]. For this problem, IPOPT sometimes generates solutions using integer $\boldsymbol{x}$ variables, but never generates solutions with integer $\boldsymbol{y}$ variables. Even after solving without a time limit, IPOPT does not produce entirely integer solutions. For smaller problem sizes, CBC does produce entirely integer solutions but fails to do so on larger instances. Similar to IPOPT, when allowed to run uninterrupted CBC still fails to produce integer solutions. IPOPT should not be expected to produce integer solutions, but it does generate integer $\boldsymbol{x}$ variables on smaller problem instances similar to CBC. Investigating the reason that CBC, a branch and bound solver, does not always produce integer solutions to this problem with no time limit is not a priority of this thesis, so to correct the integer infeasibility issue we use simple rounding techniques.

The results from these tests are depicted in Table 2 where the time to convergence and the objective values are averaged across the instances used. The objective values shown are the rounded integer solutions measured using the Exact model, and therefore may be compared against each other. These results were obtained through the Python programming language on a Windows 10 version 1909 computer with an Intel R core processor running at 2.59 GHz using 15.4 GB of RAM. Unfortunately, some

Table 2: Average solver results on different sized problem instances. The CBC and Gurobi solvers are applied to the approximate model while the IPOPT solver evaluates the Exact model. The table indicates the average objective value ($\boldsymbol{z}$) of the solutions that the solvers find for each problem size with the average convergence time for those solutions. Each problem size initially tests 5 instances, but outliers are removed which is why some problem sizes show fewer than 5 instances. Asterisks (*) indicate problem sizes where the Gurobi solver failed to produce a solution in under 20 minutes.

Size	Instances	CBC Time (s)	Gurobi Time (s)	IPOPT Time (s)	CBC Z	Gurobi Z	IPOPT Z
100 × 2	3	3.36	0.35	**0.30**	**0.775**	**0.775**	0.762
200 × 2	3	3.45	**0.54**	0.64	0.793	0.793	**0.796**
400 × 2	3	4.79	**0.91**	2.11	**0.813**	0.805	0.792
800 × 2	5	6.40	**2.14**	10.3	0.778	**0.804**	0.758
1200 × 2	4	4.87	**2.20**	20.8	0.783	**0.811**	0.773
1600 × 2	4	6.44	**3.28**	45.4	0.814	**0.821**	0.767
1600 × 4	5	**5.35**	10.7	61.7	0.782	**0.838**	0.822
1600 × 8	5	**5.44**	445	208	0.746	**0.831**	0.807
1600 × 12	5	**5.89**	338	309	0.745	**0.811**	0.792
*1600 × 24	3	**6.02**	*	753	0.723	*	**0.741**
*1600 × 32	4	**8.16**	*	563	0.790	*	**0.806**

instances caused some of the solvers to behave very abnormally, leading to extreme solve times and/or objective values. These outliers are removed from the results, which is why some problem sizes use less than five instances. To keep the results accurate, we average across the same instances used in each problem size for each solver. Therefore the objective values and convergence times can be compared against each other along the rows of Table 2. Gurobi is unable to consistently generate solutions on larger problem sizes due to the forced integrality conditions.

While Gurobi does achieve the best solution on average out of the three solvers, the solution quality attained does not justify the time it takes to produce the solutions. Therefore, CBC is the solver used for the approximate model and IPOPT is the solver used for the Exact model. The purpose of this section is not to merely compare solvers on performance. Rather, the purpose is to provide justification for the use of open-source solvers instead of the high-performing commercial ones. Understanding the trade-off between time and solution quality is paramount to the implementation of this model as a decision support tool. Almost immediately, CBC can produce a quality solution to a realistically-sized problem instance, after rounding, that is comparable to the solutions obtained by the other solvers. On these larger instances, IPOPT only barely outperforms CBC. The next section presents the performance of the GA when using the solutions from both models as initial solutions.

4.1.2 Solution Technique Results

The approximate model with the solver CBC is able to find a satisfactory solution in a negligible amount of time. We can take advantage of this by using the model's solution as an initial solution to another algorithm, one that uses the Exact model's objective function. For the GA performance tests, we use the 15 "fixed parameter" generated datasets discussed in Appendix B. For each of these datasets, we create four different problem instances by assigning them random sets of value parameters, further discussed in Appendix E. Therefore, we test 60 unique problem instances. Because the only purpose of this test is to determine the main solution method for generating VFT solutions, no value constraints are introduced. For each of these instances, we solve them several times using different methods.

First, we solve the approximate model with a maximum time parameter of 10 seconds. This is mainly to get a quality initial solution quickly, while also limiting CBC from solving for an extensive amount of time without producing better solutions. We then run the GA for 10 minutes using an initial population containing the Approximate model's solution, the greedy method solution and the stable marriage solution discussed in Section 3.5.2. More information on the architecture of this particular GA can be found in Appendix D. The third solution obtained is the solution generated by the Exact model, with no maximum time parameter. The final solution considered is found through the same GA and initial solutions as before (including the Approximate model's solution), with one population addition: the Exact model's solution. This GA is again run for 10 minutes to find a new solution.

Because each of the four solutions are applied to the same 60 instances, these results can be averaged and compared holistically. We can find a solution to the Approximate model in 0.18 minutes (about 10 seconds) with an average objective value of about 0.75. After running the GA for 10 more minutes, we improve that initial solution by about 7% on average. It takes around 20 minutes to converge on a locally optimal solution to the Exact model. Interestingly, this solution is worse than the "GA w/Approximate" solution, on average. The "GA w/Exact" solution is obtained from the GA with an initial population containing the Approximate model solution and the Exact model solution. On average, it takes about 30 minutes to find a solution that is 9.47% better than the initial Approximate model solution.

Since a solution to this problem will never have an objective value of 0, we consider randomly generated solutions to provide a baseline average objective value with which to compare the others against. Specifically, for each of the 60 problem instances, we generate five random solution assignments of cadets to AFSCs for which they are eligible. No AFSC objectives are considered in the generation of these solutions; they are completely random draws from the set of eligible AFSCs for each cadet. The average objective value across all 300 solutions ($60 \times 5 = 300$) is 0.263. In about 10 minutes on average using the GA w/Approximate technique, we can improve this solution by 204%, or by a difference of 0.536 in the objective value. Spending 30 minutes to obtain the GA w/Exact solution is not worth the small positive benefit, especially when applied to the model comparison test in Section 4.4 (we can find improvement to many more instances). Therefore, the VFT solutions used in the following section (and remainder of this paper unless otherwise specified) are obtained using the Approximate GA methodology. Figure 6 illustrates these results.

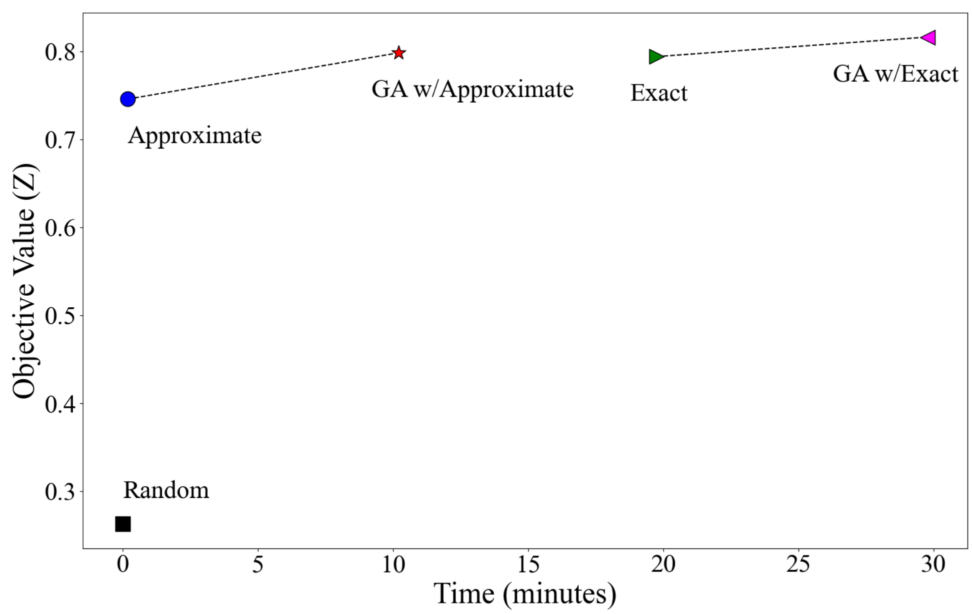


Figure 6: Average solution performance results across all 60 generated instances. The dotted lines indicate the GA's dependence on the initial solutions generated by the optimization models.

4.2 Problem Instance Analysis

We present the new approach to career field matching for cadets using the solution generation methodology discussed in the previous section. The six class years analyzed in this research are denoted by letters A through F. As an example of the analytical process that should be conducted during the assignment of cadets to their AFSCs, we consider class year C. The first step of the process is to analyze the structure of the data to identify discrepancies between what the AFSCs want and what is really possible. For the Balance Merit objective, we plot the average merit of the cadets eligible for each AFSC to observe deviations from the expected mean of 0.5. AFSCs with an average merit further away from the target of 0.5 may experience difficulty in reaching that target, depending on the quota and number of eligible cadets. In a similar manner, the USAFA proportion of eligible cadets for each AFSC is plotted to observe deviations from the expected mean (the USAFA proportion of the entire class). These charts are meant to inform the value functions used on the objectives, and are therefore presented with the discussions on value parameters in Appendix E.

Furthermore, the number of eligible cadets for each AFSC is considered when developing the target quotas for each of them. There are years when there are not enough cadets to fill one or more of the quotas. For class year C, this occurs for the AFSC "C12" and is shown in Figure 27 in Appendix E. There are only 51 cadets eligible for the AFSC, so the quota of 72 is impossible to meet. The "original" solution (AFPC's implemented solution) assigns 47 cadets to this AFSC, and this is therefore designated as the new quota for C12. To observe the AFOCD degree tier proportions for the AFSCs, Figure 7 depicts the number of cadets with different AFOCD tiered degrees that compose each AFSC. The target mandatory tier proportion for C20 designated by the AFOCD is 0.7. In class year C, this constraint is infeasible since

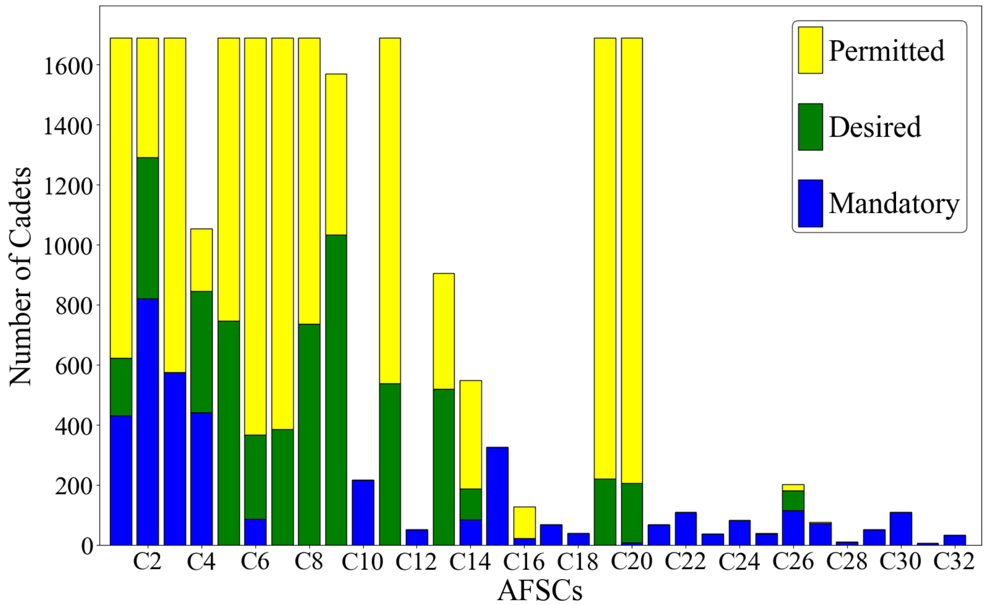


Figure 7: Number of cadets in class year C with degrees in the three AFOCD tiers for each AFSC. AFSC C20 is a concern due to the small number of cadets with mandatory tiered degrees.

there are not enough cadets with mandatory tier degrees for C20. Many of the other constraints are also not included in the original solution for this problem instance because they result in model infeasibility as well.

For the primary set of value parameters used in the model comparison, we follow the same methodology presented in Appendix E, with one difference: we generate the value parameters in a deterministic manner. The objective hierarchies for the synthetically generated data tested in the previous section are generated using normal distribution samples for many of the parameters. Here, we implement the averages used in the normal distributions for all the parameters. As a brief overview, we use a linear function of the cadets' percentiles to determine their individual weights. For the AFSCs, we use a piece-wise function of their quotas (a proxy for size) to determine their weights. Large AFSCs are distinguished from smaller ones in the constraints, effectively ascribing importance to some AFSCs over others. For this reason, the size of the AFSCs are used to determine their weights. The piece-wise function is meant

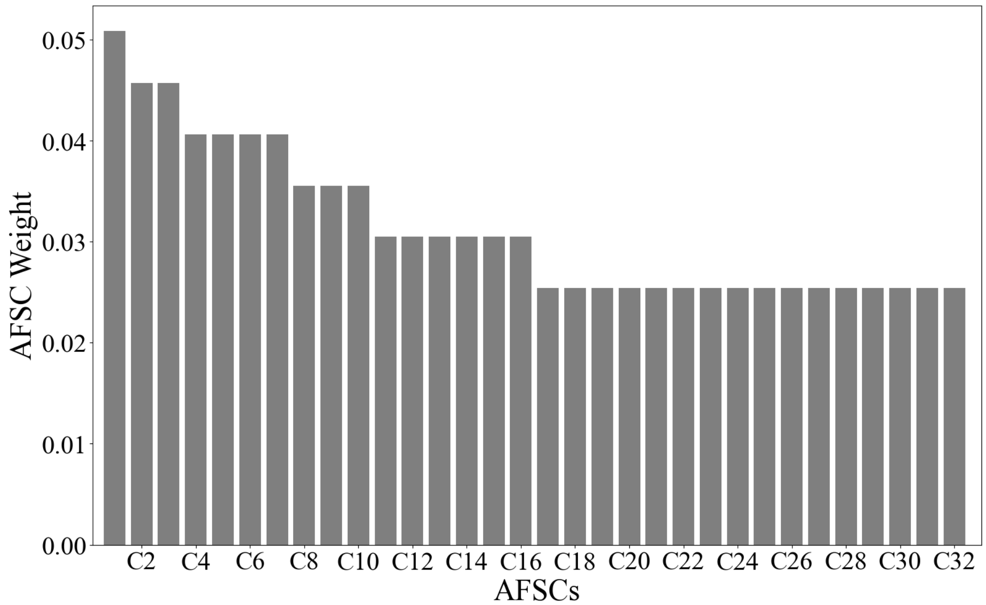


Figure 8: AFSC individual weight chart. This chart presents the weights placed on the AFSCs relative to each other. All weights shown sum to 1. These weights are determined through a piece-wise function of the AFSCs' quotas (used as a proxy for the AFSCs' size).

to bridge the gap of importance between very large AFSCs and smaller ones. The AFSC weights used are shown in Figure 8. An overall weight of 0.3 is used for cadets (and therefore 0.7 for AFSCs). This set of value parameters is an estimate of the Air Force's real value framework for the AFSC assignment problem.

As a result of the rounding technique implemented with the solution methodology, some constraints may not be entirely met. These constraints, if failed, are only missed by one cadet. Therefore, to generate feasible solutions using the GA we allow some constraints to be violated by the same amount as the CBC solver's solution. This is a very small correction, but it allows the GA to generate feasible solutions instead of failing to generate any from the Approximate model (constraint violations result in a fitness value of 0). There are only ever a few constraints violated at most, and the GA almost always finds better solutions that meet more of the constraints anyway. This is acceptable in the context of this problem, since one of the main ideas of this paper is that there should not be many constraints placed on the model. Instead, the decision-maker (DM) should specify proper weights on all the objectives to accurately distinguish trade-offs between them. Using the value framework discussed previously, the original solution for class year C has a measured objective value of 0.797. The VFT generated solution, with the same constraints as the original model, has an objective value of 0.840. This is a 5.40% improvement from the original solution.

We now compare the two solutions on each AFSC objective across all the AFSCs. The Balance Merit objective is constrained in the model to ensure all AFSCs have a minimum average merit of 0.35. The results for this objective comparing the two solutions (the original solution and the VFT solution) are presented in Figure 9. The goal is to balance the average merit, not just maintain a minimum measure of 0.35, and so the desired range is to be between 0.35 and 0.65. Because having an average merit above 0.65 is not entirely inadequate for an AFSC (they actually prefer it), the upper bound is not constrained. The lower bound, however, is constrained since no

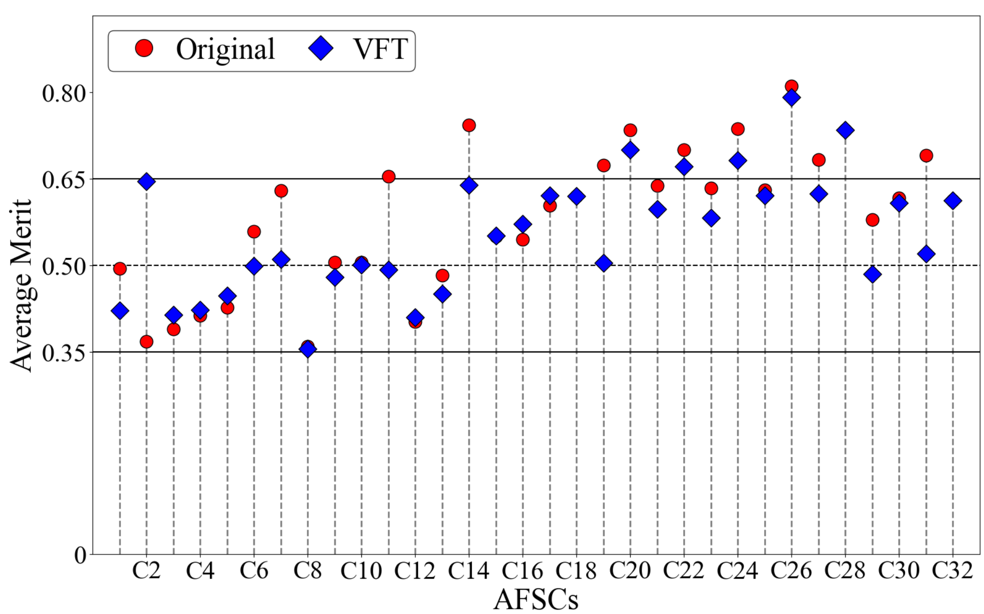


Figure 9: Average merit of assigned cadets for each AFSC in class year C. The target is 0.5, and the desired range is between 0.35 and 0.65. Measures above 0.65, though not ideal, are still acceptable. Measures below 0.35 are unacceptable.

AFSC wants a set of lower performing cadets. The target for all AFSCs, however, is still 0.5. In the two solutions, all AFSCs do have a minimum average merit of 0.35, but some are closer to 0.5 than others.

From the graph, it appears that the VFT solution more accurately balances merit around the target of 0.5. This is primarily a result of the value functions imposed on this objective, since they discourage large deviations from the desired measure of 0.5. An example value function used for one AFSC, C2, is shown in Figure 10, however more details on the value functions used can be found in Appendix E. The differences in the solutions for the Balance Merit objective can be quantitatively measured through different methods, such as the sum of squared distances between each AFSC's average merit and the 0.5 target, which is 0.692 for the original solution and 0.445 for the VFT solution. The VFT solution does outperform the original solution under this metric. It is not a completely representative metric, however, since

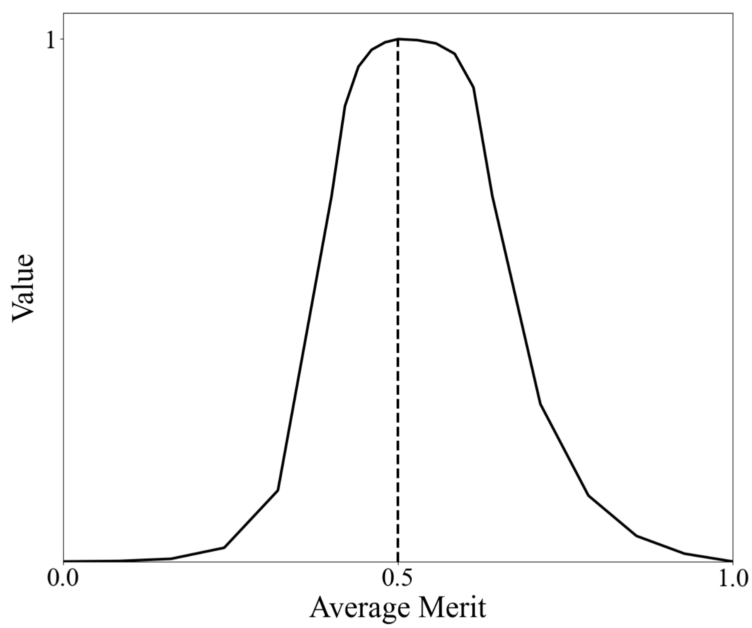


Figure 10: Average merit value function for AFSC C2. This function is a piece-wise linear approximation of an exponential function, and it is used to estimate the value received for different average merit measures for C2. The target measure is 0.5, and measures on either side are penalized, though not as drastically above 0.5.

it does not consider the trade-offs between importance of the AFSCs (their weights) or the trade-offs between importance of the measures for average merit (captured in the value functions). The sum of squared differences assumes all AFSCs are equal in importance and the penalties for deviations from 0.5 are linear in nature. Neither of these assumptions is likely accurate. A better metric for this objective can be constructed using the VFT framework. Multiplying each AFSC's individual weight by their Balance Merit objective weight effectively determines the importance of meeting this objective for each of the AFSCs, relative to each other. After scaling these weights so that they sum to 1, we can take the weighted sum of all of the average merit values for each of the AFSCs. We refer to this as the "objective score" for Balance Merit. The Balance Merit objective score for the original solution is 0.770, while it is 0.860 for the VFT solution (larger objective scores are better).

The objective to balance the USAFA proportion for each AFSC is not constrained in the VFT model, with the exception of C6, C9, C13, and C20 where the objective is given an upper bound of 0.15. The results for both solutions, including the desired ranges for each AFSC, are shown in Figure 11. One AFSC, C32, only wants USAFA cadets to be assigned to it, and is shown with a range of 0.8 to 1. All other AFSCs desire a mix of USAFA and ROTC cadets. Since the real proportion of USAFA cadets for the whole class is 0.24, we specify a range of 0.15 above and below this value to be desired. While most AFSCs fall within their target ranges for both solutions, there are some that do not. Because the original solution and the VFT solution both have measures outside the ranges and there are not any egregious failures with the VFT solution, we do not implement any more constraints on the USAFA proportion objective. Additionally, where an AFSC occurs outside its USAFA proportion range in the VFT solution, it almost always also occurs outside the range in the original solution. The only exceptions are for C14, C24, and C30. The original solution,

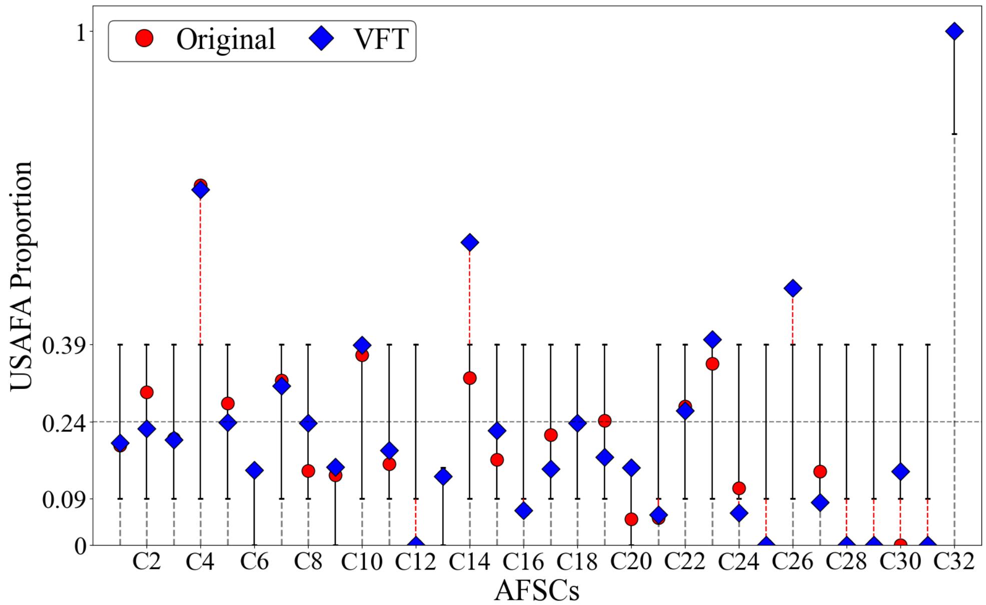


Figure 11: Proportions of assigned USAFA cadets for each AFSC in class year C. The proportion of USAFA cadets of the whole class is 0.24, and so the target proportion for each AFSC is also 0.24. The preferred ranges for the proportion are shown for each AFSC by the black range bars, where red lines indicate objective measures falling outside those preferred ranges. USAFA proportion is not constrained, except for the four AFSCs that want no more than a certain amount of USAFA cadets.

but not the VFT solution, is in the range for C14 and C24, but the opposite is true for C30. As before, the measures for USAFA proportion do tend to be closer to the target for the VFT solution, on average. The objective score for the USAFA proportion objective, accounting for the AFSCs with the 0.24 target, is 0.715 for the original solution but 0.722 for the VFT solution. Though this difference is not as pronounced as the difference in objective scores for Balance Merit, it still shows the superiority of the VFT solution.

The remaining AFSC objective results are as follows. For the quota objective, the measures are all constrained to be within the specified range and are therefore always met. The same can be said for the mandatory tier AFOCD proportion objective. Larger differences occur with the permitted and desired AFOCD objectives, since these are included in the utility function for the original model, but not the

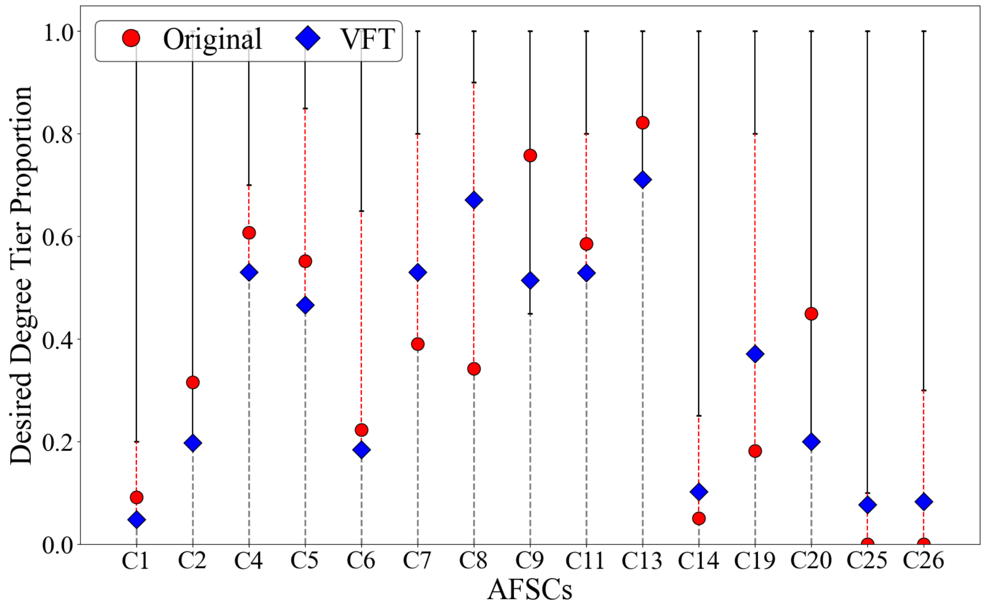


Figure 12: Proportions of assigned cadets with degrees in the desired AFOCD tier for each AFSC in class year C. The black range bars are the allowable desired tier proportion ranges for each AFSC, and the red lines indicate the solution deviations from those target ranges. This objective is not constrained, therefore many AFSCs fall outside the range for both the original and VFT solutions.

constraints. Because the AFOCD desired tier proportion objective is not constrained in either the original model or the new VFT model, several AFSCs fail to meet their objectives in both solutions. Figure 12 presents the results for the AFOCD desired degree tier across all the AFSCs for each solution. The objective score for the desired degree tier proportion objective is 0.477 for the original solution and 0.498 for the VFT solution. The proportion of cadets with permitted tier degrees is not shown, but its objective score is 0.410 for the original solution and 0.341 for the VFT solution. This is the only objective score that is weaker in the VFT solution, but it is also one of the least important objectives. The last AFSC objective is to maximize the average utility of the cadets assigned to each AFSC. There are no constraints for this objective; the only goal is to give as many cadets the AFSCs that they want. Separate from the overall cadet utility maximization objective, this is meant to serve as an AFSC's opportunity to specify importance for the satisfaction of its assigned cadets, while balancing its other objectives. The objective score for the cadet utility objective is 0.642 for the original solution but 0.702 for the VFT solution.

Due to the several different AFSC objectives across the many AFSCs, it is difficult to compare solutions based on these various charts to understand the true quality of the AFSCs. For the cadets on the other hand, their solution quality can be understood using the average utility of all the assigned cadets, as well as the overall cadet value (the weighted sum of those utilities). Figure 13 depicts a histogram of the utility received for each cadet by the AFSC they are assigned to for both solutions. There are 20 bins dividing cadet utility into 5% ranges (0 to 0.05, 0.05 to 0.1, etc.). The histograms for both solutions are superimposed so they may be compared. The average utility received by cadets in the original solution is 0.787, and it is 0.807 in the VFT solution. The overall cadet value in the original solution is 0.823, but it is

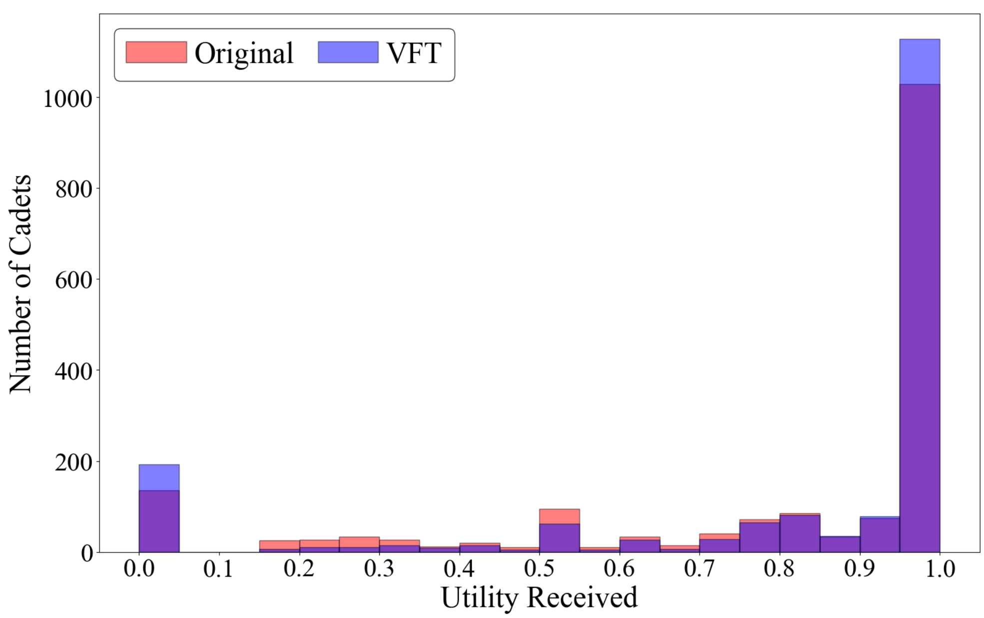


Figure 13: Histogram of cadet utility received by the two solutions for class year C. Although there are more cadets assigned to AFSCs resulting in utilities of 0 in the VFT solution, there are many more cadets receiving their top choices. The overall average utility of the VFT solution is higher than that of the original solution.

0.847 in the VFT solution. Cadets, on average, are better off in the VFT solution than they are in the original solution because they are receiving the AFSCs that they want. Additionally, higher ranking cadets in both solutions receive higher utilities than lower ranking cadets, on average. For the AFSC solution quality, there are too many different criteria considered. This is why VFT is so important to this decision problem. We must measure all the different components to this problem for an objective comparison of solutions based on the overall quality they possess. The original solution contains an overall value on AFSCs of 0.786 while the VFT solution possesses an overall AFSC value of 0.837. The VFT solution outperforms the original solution for both cadets and AFSCs.

Upon specifying overall weights on cadets and AFSCs, individual weights on cadets and AFSCs, and finally weights on AFSC objectives, we can compare aspects of the solution on a larger scale. For AFSCs, rather than compare all of their objectives

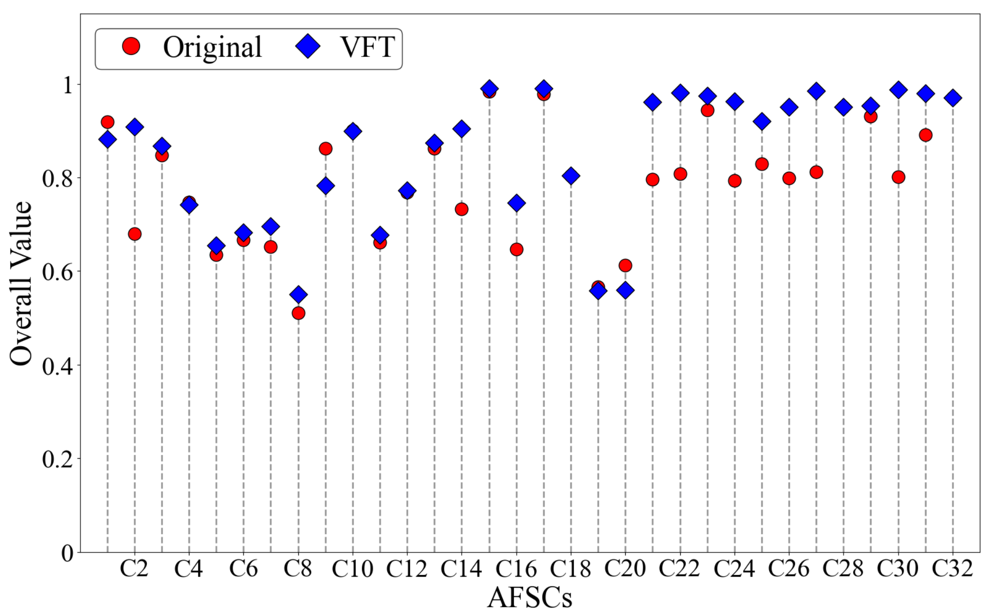


Figure 14: Individual values on AFSCs in both solutions for class year C. Each AFSC individual value is the weighted sum of all of the AFSC's objective values. Overall, the VFT solution outperforms the original solution when measured by the implemented value hierarchy.

individually, we can visualize the AFSC values, as shown in Figure 14. Since these are weighted sums of the AFSC objective values, we can compare how well we meet each of the AFSC's goals. As a whole, the VFT solution outperforms the original solution, even with the same AFSC objective constraints.

By using the VFT structure, however, a new kind of constraint is possible. Rather than specifying many different objective constraints, we could instead provide more flexibility for each AFSC by adding value constraints. If the objective weights and value functions for each AFSC are accurate, constraining the individual values on the AFSCs would allow the model to consider the trade-offs for each AFSC's own objectives in order to find a solution satisfying all AFSCs. Another approach could be to meticulously determine the weights of the model to portray the real objective hierarchy of the DM as accurately as possible, and then evaluate the VFT model with no constraints at all. This provides the upper bound on the quality of the solution possible. From here, constraints can be added on a case-by-case basis until an acceptable solution is obtained.

The VFT solution outperforms the original solution when measured according to the VFT model. This is most certainly unfair, since the solution is generated using the same model that evaluates it, indicating significant bias in the evaluation. While this is a substantial concern in the validity of the model comparisons, it is a byproduct of the inability to compare solutions otherwise. Prior to the implementation of a VFT model for this problem, there was no objective function capable of incorporating all available knowledge for solution differentiation. This section has demonstrated how difficult it is to compare just two solutions using a few of the objectives implemented in the model. What if there were more than two solutions under comparison? What if we instead presented all of the objective results? The benefit of an optimization model with a VFT framework is that all solutions can now be compared to one another using that framework. Objective calculations can be conducted to analyze the differences between the solutions. Furthermore, once the solutions are generated, the value framework itself can be adjusted to analyze the pros and cons of various components (weights) to that framework. The following section discusses how we can adjust the various weights of the model to gain insight on the solutions.

4.3 Sensitivity Analysis

Trade-offs are incredibly important for any decision problem. We must be able to consider the ramifications of imposing certain constraints on different objectives. Furthermore, we must also consider the change in quality of imposing different weights on those objectives. One substantial component to the VFT framework is the determination of the overall weights on cadets and AFSCs. Calculating the individual weights on cadets as a function of the cadets' merit is straightforward and intuitive. Calculating the individual weights on AFSCs is a bit more complicated (see Appendix E). The weights, however, that relate the overall achievement of cadet objectives (the weighted sum of their assigned utilities) and the achievement of AFSC objectives have a considerable impact on the quality of the generated solutions. These weights determine on a macro level the entities (cadets or AFSCs) that should be prioritized. An overall weight on cadets of 1 effectively removes the AFSCs from consideration, while a weight of 0 on cadets does the opposite. To determine what the overall weights should be, we can solve the model multiple times with different overall weights.

The DM must understand the trade-offs in solution quality between weighting cadets and AFSCs differently. Solving the model multiple times, and scaling the overall weight on cadets for each iteration, allows the DM to visualize the frontier for Pareto-Optimal solutions to determine which ones have the desired value for cadets and AFSCs. In the context of this problem, this "Pareto Frontier" determines the set

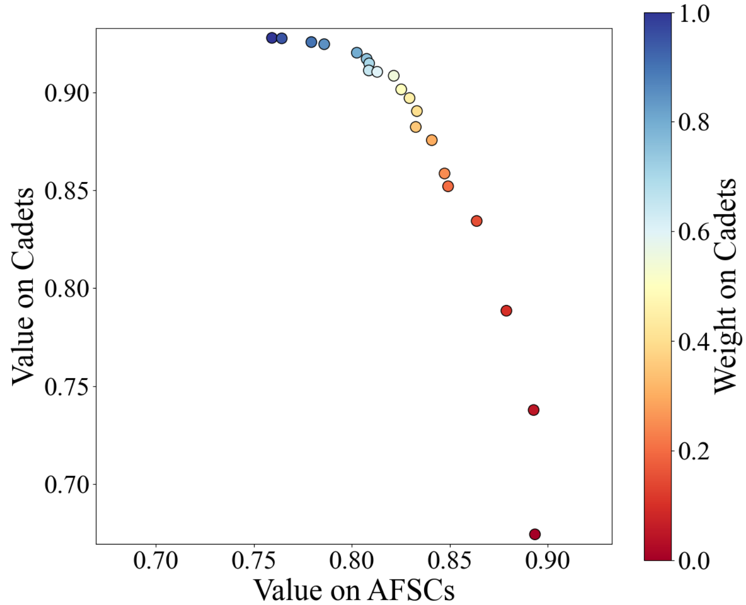


Figure 15: Estimated Pareto Frontier on solution quality for cadets and AFSCs when solving the VFT model multiple times using different overall weights on cadets. The color bar indicates the cadet weight used to generate each of the solutions (initial overall cadet weight of 1, with a gradual descent to 0). A sharp decrease in cadet value is found when using weights on cadets below 0.2. Similarly, a significant decrease in value occurs for AFSCs when using weights on cadets above 0.7.

of points that represent the presumed optimal (or near-optimal) solutions to the VFT framework using different overall weights on cadets and AFSCs. Furthermore, the Pareto Frontier is the curve for which there are no other solutions that benefit both entities mutually. Figure 15 depicts the VFT solution values for AFSCs and cadets generated by scaling the overall weight on cadets from 1 to 0. This chart illustrates the need for proper overall weight determination. The minor benefit in value on AFSCs for lower weights on cadets does not justify the immense drop in value for cadets. Likewise, the same could be said for larger weights on cadets not justifying the decrease in value for AFSCs. If the goal is to minimize the distance between the frontier shown on this graph and the point corresponding to the maximum possible value for both AFSCs and cadets (1, 1), then placing an overall weight on cadets of 0.45 (and therefore 0.55 for AFSCs) would achieve the greatest possible benefit for both entities. This example is one form of sensitivity analysis we can conduct on the weights used in the model, and it should be used to provide support for the implementation of one solution over others when the overall weights are altered.

Another possible method of conducting sensitivity analysis pertains to the AFSC objective weights. Consider two different solutions to this problem, solution b and solution t, each generated through different means. Solution b is the best solution given one value framework, while solution t is found through some other method. Such is often the case in this problem. The original solution is generated via the original methodology, and the VFT solution is created through the VFT optimization model. The value hierarchy for the VFT solution is known, but not so for the original solution. Given these two solutions, and the value framework that makes one outperform the other, we are able to conduct a certain method of sensitivity analysis on the weights used in the model. Discussed in Chapter II, the "Least Squares Procedure" for finding the weights of the model that make the value of one solution exceed the other by some threshold is a powerful tool that can be applied to this problem [35]. For this particular problem context, we consider the AFSC objective weights that make the two solutions identical in value, while minimizing the differences in these weights from the known value framework.

To conduct the Least Squares Procedure (LSP) for generating the weights that make two solutions identical in value, we create another optimization model to solve a different problem. The objective function is to minimize the sum of squared differences between the known objective weights (Weights b) and the new ones. The parameters to this model are the original framework weights and the solution values for cadets and AFSC objectives. An additional parameter is Δ, the amount by which one solution should exceed the other. Most of the time (including in this example), this parameter will be 0 to ensure the two solutions are equal in value. The full formulation for this new problem is presented in Appendix F. Applying this methodology to the class year C example from the previous section allows for the comparison of the original solution (denoted in the LSP formulation as solution t) with the VFT solution (solution b). Under the original value hierarchy using Weights b, the original solution has an objective value of 0.797 while the VFT solution has an objective value of 0.840. Solving the LSP model results in the new AFSC objective weights that minimize the squared difference from Weights b. Under these new weights, both solutions obtain a value of 0.815. For an illustration of the differences in AFSC objective weights, Figure 16 displays the objective weights on AFSC C2 within Weights b and the generated weights using LSP.

This chart's main purpose is to show the validity of the original set of weight parameters used to generate the VFT solution. The LSP weights show just how

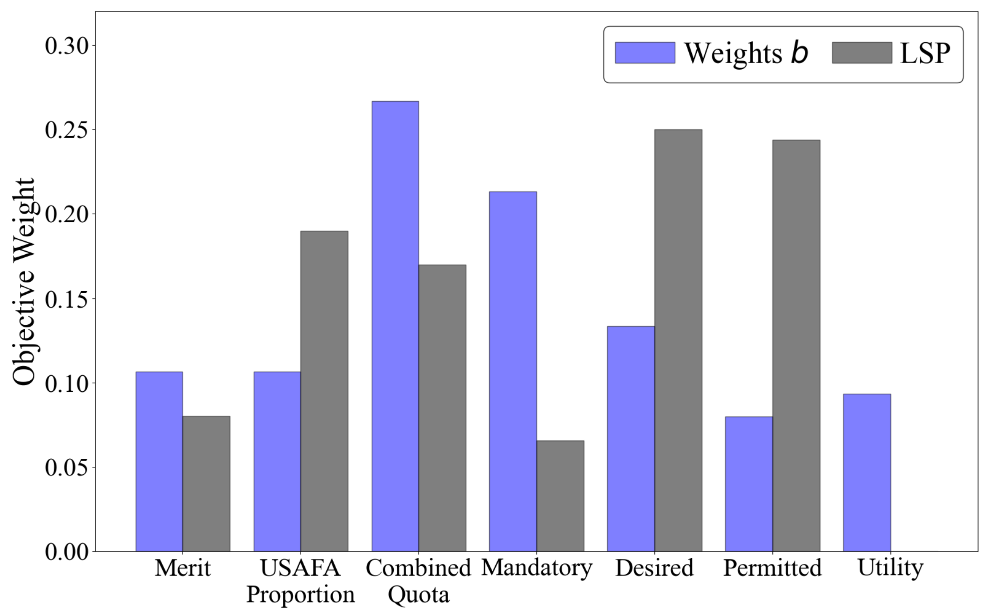


Figure 16: Objective Weights for AFSC C2 in the original value parameters (Weights b) compared with those calculated through LSP. The quota and mandatory degree tier objectives are weighted higher under Weights b, but less so in the generated weights using LSP. This shows the accuracy of the original set of value parameters over the new set.

much the DM's framework has to change in order to accept the original model's solution as equal to the VFT solution. For C2, the desired and permitted AFOCD tier proportions must be more important, and the quota and mandatory tier proportion objectives must be less so. Additionally, cadet utility should have a weight of 0. This is not an accurate depiction of priorities for C2's AFSC objective, and therefore we reject these new weights and maintain the original ones. Even if the new weights are acceptable, we may solve the model again with the new set of value parameters to get a better solution than the VFT and original solutions. The VFT model is a robust approach to career field matching for cadets, and provides many tools for conducting analysis on a particular class year instance. To further prove the validity of this model, and its dominance over the original model, the following section compares the original solutions against new VFT solutions generated using many different sets of value parameters across multiple class years.

4.4 Comparison Against Real Solutions

To show the superiority of the new VFT model over the original model, we consider the real cadet class year datasets and original model solutions. For these tests, we consider six class years, denoted A through F, for which we have AFSC quota data. For each of the six class years, we create 30 different problem instances by generating 30 different sets of weight an value parameters (different value hierarchies) following the methodology presented in Appendix E. There are therefore 180 unique problem instances tested in this section. These sets of value parameters are meant to estimate the true values of the DM. To make the solution comparisons as fair as possible to the original criteria, we implement constraints on average merit, combined quotas, and mandatory degree tier proportions. The USAFA proportion of assigned cadets for each of the AFSCs vary substantially in the original solution, due in large part to the fact that the USAFA proportion of eligible cadets for each AFSC also vary considerably. Therefore, this objective is not constrained, with the exception of the four AFSCs that specify maximum proportions of cadets: C6, C9, C13, and C20.

Using these constraints, we can generate the lower bound on the VFT solution quality for all the problem instances. If the VFT model is granted as much flexibility as the original model (the analysts are forced to pick and choose which constraints to include since the inclusion of all constraints result in infeasibility), better solutions could still be found. Nevertheless, we generate a new solution for each of the 180 instances using the same constraints as the original model. This allows us to compare the original solution with each of the VFT solutions using different VFT objective hierarchies. The results for these tests are presented in Figure 17. The "percent improvement" is the percent increase of the new VFT solution from the original

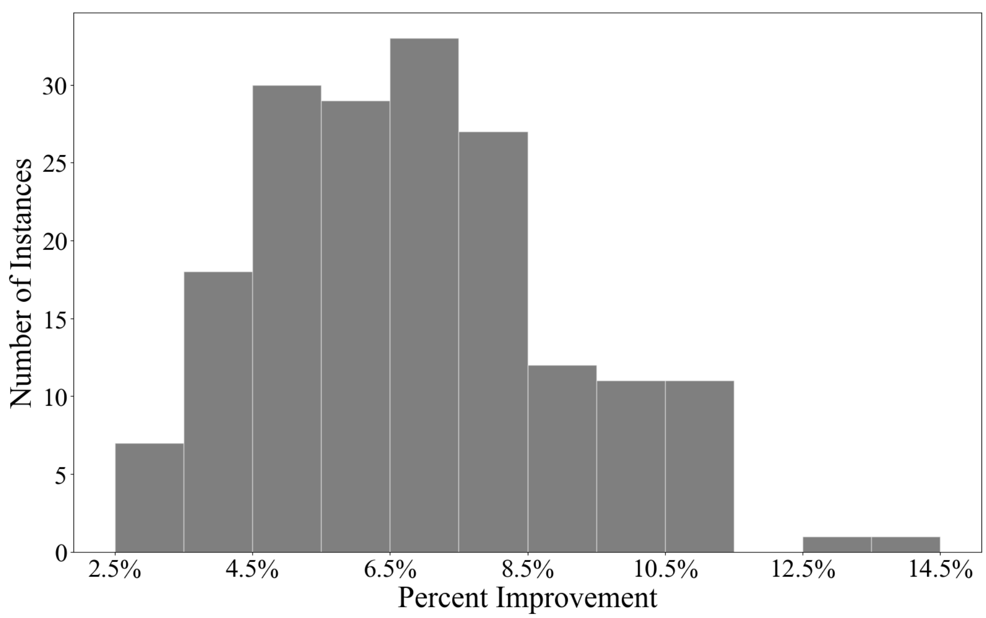


Figure 17: Histogram of VFT percent improvement over original solutions for all instances. Each instance uses the set of fixed parameters from the class years and a generated set of value parameters. We measure the original solutions for each class year against the generated VFT solutions using the sets of value parameters for that year.

solution when measured by the specified set of value parameters. This is our primary performance metric, since it is an objective measurement of two solutions that is not biased to larger objective values. The sample mean percent improvement for the 180 instances is 6.82%. The sample standard deviation is 2.17%, and (assuming a normally distributed population) a 95% confidence interval on the population mean is calculated as (6.51%, 7.14%).

No matter which objective hierarchy is used, or how important different objectives are from each other, we can still find solutions that are almost 7% better on average than the original model. One important consideration to note is the bias that these solutions have to the new VFT model. As discussed in Section 4.2, if we generate solution Y based on certain objective criteria, and then compare that solution to a different solution (created through some other means) using the objective criteria of solution Y, solution Y will most likely be better. These results share the same bias as the problem instance example presented earlier. The main idea is that the VFT framework provides the criteria with which solutions can be differentiated. Until now, there was no method to equitably compare the achievement of Air Force objectives for two or more unique solutions. This is the focus of decision analysis: comparing and choosing alternatives using some objective framework.

While we can certainly compare the VFT solutions on their superiority to the original solutions using the value framework, we can also compare them based on how similar they are. We define solution similarity between two different solutions as the percentage of cadets that are assigned to the same AFSC in both cases. Using multi-dimensional scaling (MDS), we can plot out the similarity of different solutions to see where clusters form and, more importantly, where certain solutions appear to be outliers. If we observe many solutions near each other, even when generated using different sets of value parameters, then these solutions are more robust than

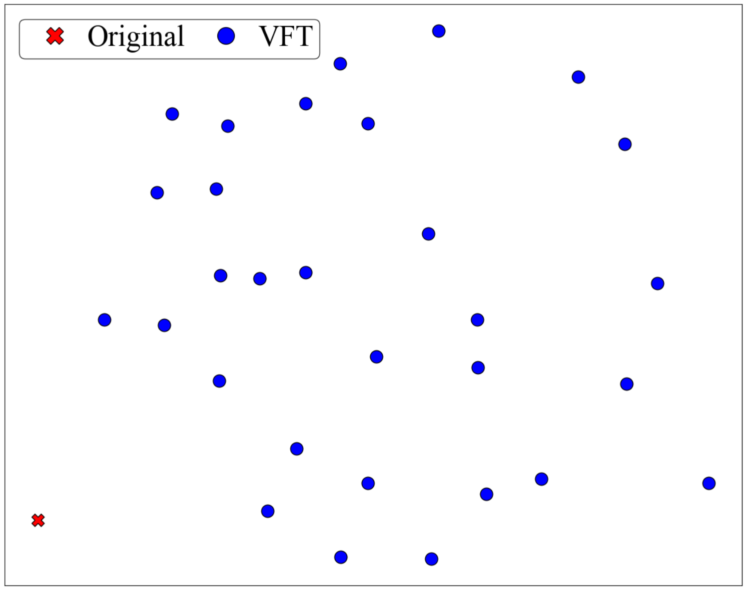


Figure 18: Class year A solution similarity plot. All 31 solutions (30 VFT solutions and the original solution) are compared to each other in a similarity matrix which is then converted to 2-dimensional coordinates using MDS. More similar solutions appear closer to each other. No axes are labeled because the axes are normalized distance values with no real meaning.

others. There is support for the cadets who are consistently assigned to the same AFSCs across different solutions to show that they should in fact be assigned to those AFSCs. However, solutions that are vastly different from the population are likely not the best alternatives. Figure 18 presents a graph of the similarity of the solutions generated for class year A. This graph indicates that the original solution is an outlier, and is in fact the furthest solution from the population. Although this test is more anecdotal than some of the others, this does provide additional support in favor of the VFT model.

4.5 Summary

We find that the best method of producing solutions to the VFT model in a timely manner is through a combination of the Approximate optimization model and a GA.

Specifically, we use the Approximate model to generate a strong initial solution in about 10 seconds and then evolve that solution using the objective function of the Exact model with a GA for an additional 10 minutes. This method allows the DM to adjust their weight and value parameters to find a new solution with an adjusted value hierarchy in a small amount of time. If the DM is confident in their framework, and desires to find the best solution possible for the purposes of implementation where computation time is not a primary concern, then the Exact model should be utilized in conjunction with a much longer GA solve time.

We present the new analysis process on an example class year and show that the new VFT solution outperforms the original solution in almost every way. Additionally, we present two methods to provide sensitivity analysis on different solutions: one on the overall weights of cadets and AFSCs and another on the AFSC objective weights. Finally, we conduct a comparison of the real solutions against VFT generated solutions on six class years using many different sets of value parameters. We find that the VFT solutions provide a 6.82% solution quality improvement, on average, over the original solutions as measured using the generated VFT value parameters. We conclude that the new VFT model is objectively superior to the original model.

V. Conclusions

This chapter presents a summary of the major contributions of this research. We first discuss those contributions and how they positively impact the Air Force as a whole. Next, we discuss the main limitations of this paper as they affect the results presented. Finally, we conclude with a discussion of several topics that could be expanded on in a future iteration of career field matching for cadets.

5.1 Key Findings and Contributions

This paper presents a new Value-Focused Thinking (VFT) model that evaluates the objectives for assigning cadets to their Air Force Specialty Codes (AFSCs). This model measures those objectives through different means, calculates the values of those objectives as defined by the decision maker (DM), and computes the overall quality of the solution through a weighted sum of all objective values. The weights and values to the model are the key parameters determined by the DM. To generate alternative solutions to compare with using the VFT framework, we present an optimization model that uses the VFT framework as its objective function. We present a solution methodology to solve the model and compare the solutions generated by this model against the real solutions used by the Air Force Personnel Center (AFPC). The new solutions are found to be 6.82% better, on average, than the real solutions when measured with many different value parameters using the same constraints as the original solutions. No VFT solutions were worse than the original.

The new model is able to discern accurate trade-offs in value between various aspects of the solutions, allowing it to effectively generate quality alternatives. Such a model is quite powerful, because all components of the quality of the solution are parameters to the model and can therefore be easily, and intuitively, altered. To allow for these very intuitive parameter adjustments, and to provide a means of conducting the problem instance analysis process presented in Section 4.2, we present a dashboard built in the Python programming language as part of the completion of this research. This dashboard performs all elements of the analytical process discussed in this paper, from observing discrepancies in the data given the AFSC objectives to determining all value parameters and ultimately providing sensitivity analysis on the solutions generated by the model. AFPC is well equipped to handle future AFSC assignment problems with new objective criteria using this VFT modeling framework.

5.2 Limitations

One important limitation to this research is the lack of full knowledge on the true values of the DM. Because the "DM" is actually many different decision makers, it is very difficult to aggregate all opinions to evaluate the true nature of solution quality. This is why there are many layers to the VFT model, which allow the decision makers to collectively place weights on the areas they are interested in. The career field managers could be responsible for different weights on AFSC objectives, while USAFA and ROTC leadership could decide how to weight cadets differently, for example. This research did not incorporate specific inputs from these many decision makers. Therefore, to address this concern we generate realistic value parameters sampled from different normal distributions using estimates provided by AFPC as the distributions' expected means. This is done to show the new model's robust nature when applied on many different value hierarchies.

An additional limitation to the research conducted in this paper pertains to the solution methodology. Unfortunately, the exact VFT model is a complicated model to solve. It cannot be solved using linear programming due to the presence of non-convex terms, resulting in a complex non-linear optimization model. Solving this model directly, even with a powerful non-linear solver, is not a viable option for implementation by AFPC due to the computational time it requires. Therefore, we propose a different means of finding solutions using a combination of conventional optimization techniques and a meta-heuristic.

5.3 Future Work

There are many potential future continuations or considerations of this research. A future adaptation of this research topic could conduct thorough discussions with decision makers to determine more accurate value parameters. There are also many opportunities to expend upon the solution methodology presented in this paper. A different meta-heuristic could be more appropriate for implementation on this optimization model, for example. Additionally, the approximate model used in this paper could be formulated in a more efficient manner. Rather than approximate the number of cadets assigned to an AFSC with the AFSC quota, separate objectives could be created specifically for a linear programming model. The mandatory tier proportion objective, for example, could instead be changed to a mandatory tier quota.

This paper focuses solely on the non-rated career fields of the Air Force, but future research could apply this model to rated career fields as well. Additionally, rated and non-rated career fields could be combined in a single VFT optimization model. AFPC would therefore be able to conduct one assignment of all cadets and all AFSCs, rather than two separate assignments. The VFT modeling formulation presented in this paper could also be applied to other areas of career matching, such as officer personnel assignments. Cadets, in this model, only have one objective: maximize their own utility. There may be other considerations such as the bases that the cadets are assigned to or their extended active duty orders date. These could each be separate cadet objectives included in the model. In the field of cadet considerations for this VFT model, there is potential for retention analysis on the long-term effect of assigning cadets to AFSCs based on preference. How much of a positive effect, if any, does giving cadets the AFSCs they want have on the overall health of the Air Force? How does officer satisfaction affect work life? As Secretary of Defense Mark T. Esper stated, “the quality of our service members” is what makes the United States military the leading fighting force in the world.

Appendix A. Solution to Kumar's Problem

As presented in the most recent paper by Rabbani et al. [8], the formulation for Kumar's unbalanced assignment problem is as follows. Let $i \in \mathcal{I}$ be the set of all machines, and $j \in \mathcal{J}$ be the set of all jobs. The cost parameter, c_{ij}, is the cost of assigning job j to machine i. The decision variable, x_{ij}, is a binary variable indicating if job j is assigned to machine i (1) or not (0). The formulation is therefore:

$$\text{minimize} \quad \sum_{i \in \mathcal{I}} \sum_{j \in \mathcal{J}} c_{ij} x_{ij} \tag{23a}$$

$$\text{subject to} \quad \sum_{j \in \mathcal{J}} x_{ij} \geq 1 \quad \forall i \in \mathcal{I} \tag{23b}$$

$$\sum_{i \in \mathcal{I}} x_{ij} = 1 \quad \forall j \in \mathcal{J} \tag{23c}$$

$$x_{ij} \in \{0,\ 1\} \quad \forall i \in \mathcal{I},\ j \in \mathcal{J} \tag{23d}$$

The assignment cost matrix for Kumar's problem instance is shown below, where the rows correspond to machines, the columns correspond to jobs, and the solution assignment is indicated in blue text:

$$\begin{bmatrix}
300 & 250 & \color{blue}{180} & 320 & 270 & 190 & 220 & 260 \\
290 & 310 & 190 & 180 & 210 & 200 & 300 & \color{blue}{190} \\
280 & 290 & 300 & \color{blue}{190} & 190 & 220 & 230 & 260 \\
290 & 300 & 190 & 240 & 250 & 190 & \color{blue}{180} & 210 \\
\color{blue}{210} & \color{blue}{200} & 180 & 170 & \color{blue}{160} & \color{blue}{140} & 160 & 180
\end{bmatrix}$$

The optimal solution is to assign job 3 to machine 1, job 8 to machine 2, job 4 to machine 3, job 7 to machine 4, and jobs 1, 2, 5, and 6 to machine 5. This results in a minimum cost of 1450.

Appendix B. Data Generation

This paper seeks to find the best method of assigning commissioning cadets to their Air Force Specialty Codes (AFSCs). To do this, the solution methodology must be tested on a sufficient number of problem instances. The problem instances are distinct cadet class years with unique cadets and, occasionally, unique AFSCs. Since this research only has access to a small set of class year instances, we artificially generate sets of cadets and AFSCs to test the optimization models. This appendix details how the use of neural networks can accomplish this desired data generation.

B.1 Literature Review

Data generation is a common method of creating more accurate models. Specifically, the usual method of data augmentation is for training more accurate machine learning models. Oftentimes datasets are imbalanced, resulting in certain classes of observations occurring much more frequently or infrequently than others. Augmenting data can help balance the classes and prevent over-classification, leading to more robust models. Image generation is also becoming more common. Given the general success achieved through the generation of synthetic images using generative adversarial networks (GANs), first presented in 2014 by Ian Goodfellow et al. [48], there has been a substantial push in the neural network community to apply similar methods to generate tabular data. One of the key factors that allows GANs to perform so well on images has to do with the data itself. An image can be represented as a matrix of pixels. In the case of grayscale images, each pixel takes on a value from 0 (black) to 255 (white). One observation in a dataset of grayscale images is therefore a flattened array of continuous values between 0 and 255. A convolutional neural network (CNN) is well-equipped at handling this kind of data, and so the classical GAN architecture allows for a straightforward method of generating synthetic grayscale images.

Tabular data, on the other hand, is arguably more difficult to adequately replicate. One reason for this is that tabular data frequently contains columns with discrete values as well as continuous values. Categorical data (the discrete columns) can further complicate the data structure when the categories themselves are highly imbalanced. Continuous tabular data can also be challenging since the data often does not follow known distributions, whereas pixel values in images usually fit a Gaussian-like distribution [46]. Other factors that cause tabular data generation to be difficult include the presence of multi-modal distributions and sparse one-hot-encoded vectors [46]. Several tabular GAN structures have been introduced in recent years to tackle some of these issues, such as the medical GAN (medGAN) [49] to generate realistic medical patient records, the Variational Encoder Enhancement to Generative Adversarial Networks (VEEGAN) [50] to overcome the issue of generating only a few modes of the data distribution, and table-GAN (TGAN) [51] for the intent of preventing real personal data leakage as a result of synthetic data generation.

In 2020, a new tabular GAN structure is introduced from the Massachusetts Institute of Technology (MIT): the Conditional Tabular GAN (CTGAN) [46]. To tackle the non-Gaussian and multi-modal distribution complications discussed previously, CTGAN invents the “mode-specific normalization” technique by which it estimates the number of modes for a given continuous column using a variational Gaussian mixture model (VGM) to then model the distribution of that column. The probability of each value in the column coming from each mode is computed and a mode is sampled from that probability density. From that sampled mode, the value is normalized. As the name implies, CTGAN is a conditional generator that is able to overcome the “class imbalance” problem by sampling and re-sampling categories efficiently, such that all discrete values are represented evenly and no infrequent categories are left out of the model as a result of generator learning deficiencies. CTGAN also utilizes several other features including "training-by-sampling," which uses a new conditional vector to implement the conditional properties of the generator and a new generator loss function that incorporates a penalty to enforce those conditional properties.

For this research, the Synthetic Data Vault (SDV) library [52] is used. Developed and maintained by MIT, the SDV library allows for easy implementation of CTGAN. The structure of CTGAN is detailed within [52], but as a brief overview, CTGAN utilizes two hidden layers for both the generator and the discriminator each with 256 hidden neurons. For the generator, rectified linear unit (ReLu) activation functions are implemented along with the paper's new mode-specific batch normalization [52]. Training-by-sampling is incorporated in the second hidden layer. This paper utilizes CTGAN for the purpose of generating realistic synthetic cadet and AFSC data that can be used as parameters to an optimization model. This is done to compare various optimization models against each other on synthetic cadet class year instances.

To ensure the generated data is realistic enough and comprises a set of cadets that the Air Force could theoretically have to assign to AFSCs, we calculate two statistical distribution measures: the Kullback-Leibler (KL) divergence and Kolmogorov-Smirnov (KS) distance. These measures are chosen for their prevalence in the statistical analysis community. Both statistics measure the distances between the theoretical distribution and the empirical distribution. For this application of CTGAN, the theoretical distribution is the combined distribution of all of the different features of the real data. The empirical distribution, therefore, is the distribution of the different features of the generated data. SDV converts these two metrics on a 0 to 1 scale where larger values indicate that the two distributions are closer together. No hypothesis test is considered; therefore only the test statistics are used in the new score. This new score is created using a weighted average of the two scaled measures. It is referred to in this appendix as the "KS/KL Aggregate Score."

B.2 Methodology

The datasets used for this research contain the parameters to the optimization models. Table 3 presents an example of how the cadet data is stored for a problem instance with three AFSCs and four cadets, each with two AFSC preferences. Since the class years in the real CTGAN data all use the same career fields (the AFSCs that correspond to those career fields change by name only), we refer to them as the AFSCs in class year A. The $USAFA$ column holds the information to construct the set of USAFA cadets that are eligible for each AFSC used in the model. The *percentile* feature contains the data necessary for the *merit* parameter. Let p represent the index for the cadet's ordinal preferences (1 to 2, in this example). The $Util_p$ column indicates the utilities that the cadets have placed on their p^{th} choice AFSCs. There could potentially be as many of these utility columns as there are AFSCs to specify preferences for, though historically cadets have only been able to indicate up to six AFSC choices. The $Pref_p$ column contains the AFSCs that the cadets have indicated as their p^{th} choices. The $Util$ and $Pref$ columns are used to determine the *utility* parameter matrix. The *qual* columns indicate the eligibility level that each of the cadets have for each AFSC. These columns are used to determine the sets of cadets with mandatory, desired, permitted, or ineligible degrees for each AFSC. These tiers are denoted by "M", "D", "P", or "I", respectively.

Table 3: Example data features and observations. Each observation corresponds to a cadet, and the features represent attributes of each cadet that are used in the optimization models. This particular small example contains three AFSCs, four cadets, and two AFSC preferences for each cadet.

$USAFA$	*percentile*	$Util_1$	$Util_2$	$Pref_1$	$Pref_2$	$qual_{A1}$	$qual_{A2}$	$qual_{A3}$
1	0.76	1	0.91	A1	A2	M	P	P
0	0.53	1	0.84	A3	A1	D	M	P
0	0.34	1	0.75	A2	A3	I	P	D
1	0.24	1	0.68	A3	A2	D	M	D

To test the optimization models' robustness and validity on many different problem instances, a realistic data generator is used. This generator must use as much of the historical data available as possible, capturing the underlying distributions of cadet preferences, their sources of commissioning, percentiles, eligibility, and so on. To create this realistic generator, we utilize the conditional tabular-generative adversarial network (CTGAN) architecture [46]. To train CTGAN, data from selected real class years of graduating Air Force cadets is used. After considerable data cleaning, such as standardizing the AFSCs to account for recent name changes, the data is structured in the same format as presented in Table 3, with two key differences. The first is the omission of the $Util_1$ feature since it represents the utilities of the cadets' first choice AFSCs, and is always a column of 1s. Additionally, rather than include the entire qualification matrix for CTGAN training, we utilize the cadets' Classification of Instructional Program (CIP) codes (their degrees). Cadets usually only have one degree (CIP_1), but occasionally declare a second (CIP_2). The same example observations presented before are shown in Table 4, using CIP codes instead of qualifications. After the data is generated, the CIP codes may then be translated into the qualification matrix using the most recent variant of the AFOCD. Now, CTGAN only has to learn the relationships between preferences, sources of commissioning, percentiles, and the cadets' degrees.

Table 4: Example data features and observations used by CTGAN. The $Util_1$ feature is removed because it is fixed for all cadets at 1. Instead of including the cadet qualifications explicitly in the data as before, we use the cadets' CIP codes (their degrees) which can be converted back into qualifications after the data has been generated. Note that cadets may have two degrees (double majors).

$USAFA$	$percentile$	$Util_2$	$Pref_1$	$Pref_2$	CIP_1	CIP_2
1	0.76	0.91	A1	A2	140201	
0	0.53	0.84	A3	A1	110701	
0	0.34	0.75	A2	A3	450901	450201
1	0.24	0.68	A3	A2	290207	

The synthetic data vault (SDV) [52], a library in the Python programming language, allows for the ease of implementation of CTGAN. Once data is generated from CTGAN, it must be evaluated using some sort of evaluation framework. SDV has a function that evaluates the Kullback-Leibler (KL) divergence and Kolmogorov-Smirnov (KS) distance measures and computes an aggregate score utilizing a weighed average of the two, as discussed in the previous section. This aggregate score is the primary value used in our model comparisons. Feature averages for the different datasets are also used to provide more context for the application of CTGAN on this specific problem. These include the proportion of USAFA cadets in the sample, the average percentile of the cadets (which intuitively should be about 0.50), and the average utilities for the cadets' second choice, third choice, fourth choice AFSCs and so on. These evaluation metrics, along with the statistical tests' aggregate scores, are depicted in the following section.

One of the biggest issues with this problem is the dependencies that certain features have on other features. For example, generating cadet preferences involves the process of sampling from the set of all AFSCs without replacement, since the preferences must be unique. This means that any AFSC can be chosen as the cadet's first choice ($Pref_1$), but the second choice cannot be the same AFSC as the first one ($Pref_1 \neq Pref_2$), and the third choice cannot be the same AFSC as the first choice or the second choice ($Pref_1 \neq Pref_3$ and $Pref_2 \neq Pref_3$), and so on. Similarly, cadet utilities for each of the cadets' choices are also dependent on each other. The utility on the first preference for every cadet is always 1, so it is removed from this model. The utility on the second choice ($Util_2$) must always be greater than or equal to the utility on the third choice and this pattern repeats for all the utility features ($Util_2 \geq Util_3 \geq Util_4 \geq Util_5 \geq Util_6$). Lastly, any instance of a blank preference for a cadet ($Pref_p = None$) must be associated with a utility of zero in the corresponding $Util_p$ position. Cadets have up to six choices, but they can make fewer choices. If there is no AFSC listed as the cadet's p^{th} preference, there should be no utility assigned to that slot. These three constraints can conveniently be defined as parameters to CTGAN in the SDV library. These constraints only affect the sampling of the model during the data generation phase and do not interfere with the training in any capacity.

B.3 Analysis and Results

After training a CTGAN model using 1,000 epochs, we are able to generate new data from it and evaluate that data according to the framework described previously. With this model as a baseline, CTGAN's hyper-parameters can be tuned to see if a better model is possible than the original structure presented in the MIT paper [46]. The process is simple: a tunable hyper-parameter of CTGAN is selected, and then new models are trained using several different optional values for that particular hyper-parameter, holding all other hyper-parameters constant. For example, the algorithm initially selects batch size as the tunable hyper-parameter. Then, the algorithm iterates over different batch size options such as 200, 300, 400, and 500, holding everything else constant for 100 epochs. For each trained model, 50 synthetic datasets are generated and evaluated under the evaluation framework described in the previous section

The average KS/KL aggregate score across these 50 samples is used to compare against the other models with different values for a specific hyper-parameter: batch size in this example. The model with the best average aggregate score is selected, and its value for batch size is used within the set of hyper-parameters for the next set of iterations on a new hyper-parameter. After performing this process for many different hyper-parameter combinations, the top performing models are selected to be trained further. These models are trained on the full 1,000 epochs, to be consistent with the baseline CTGAN model. No model was found to be substantially better than the classical CTGAN. For this reason, along with the desire to implement exactly what is presented in [46], no parameter changes are made to the structure of CTGAN. Although tuning the parameters of the CTGAN model did not result in any models that were substantially better than the original, this does provide more validation to the work done at MIT. Their development of this neural network architecture has allowed for the structure to be easily generalized to fit any sort of dataset composed of continuous and categorical columns.

For this research, 15 datasets are generated from CTGAN. These datasets are the problem instances used to test the optimization models in Section 4.1.2. The validity of these datasets can be shown in Table 5, where the CTGAN metrics are averaged across all generated instances. The average percentile of the datasets is not included in this table because it can be easily normalized to 0.50 after the data has been generated. Since this data is synthetic and used purely for optimization model performance comparisons, applying small changes to all of the cadets' percentiles to force the average to be 0.50 is an acceptable data cleaning procedure. The $Util_p$ average metrics are the average utilities that cadets place for their p^{th} choice. We can use this information to also determine the average utilities that are placed on each

Table 5: Average evaluation metrics for synthetic data and real data. The CTGAN evaluation metrics are averaged across all 15 generated instances. Average percentile is not included here since it is fixed to 0.50 after the data generation process. The KS/KL aggregate score is acceptable, and suggests that CTGAN does moderately follow the real distributions.

Data	KS/KL Aggregate Score	USAFA Proportion	$Util_2$ Avg	$Util_3$ Avg	$Util_4$ Avg	$Util_5$ Avg	$Util_6$ Avg
CTGAN	0.652	0.258	0.742	0.520	0.365	0.248	0.187
Base	N/A	0.287	0.757	0.568	0.418	0.299	0.216

of the AFSCs. This essentially depicts the popularity of the AFSCs with the cadets. Similar to Table 5, we average this popularity across all 15 instances and compare it to the baseline real data, shown in Figure 19. CTGAN accurately reflects cadet sentiment towards the AFSCs.

One other key underlying distribution that we want the generated data to capture is the distribution of degrees that cadets have within the whole class. The distribution of degrees, which ultimately determines AFSC qualifications under the AFOCD, for the generated data should be similar to that of the real data. To visualize this distribution, we create CIP histogram bins as follows. First, we create a list of unique CIP codes that cadets have in numeric order (10000, 10010, 10020, etc.). There are 245 unique CIP codes found for cadets in the real data. We then split the CIP codes into 49 groups of 5, putting CIP codes 1 through 5 in the first bin, 6 through 10 in the second bin, and so on. For each bin, we calculate the proportion of cadets in the real data that have a CIP code that falls in that bin. We then sort the bins by

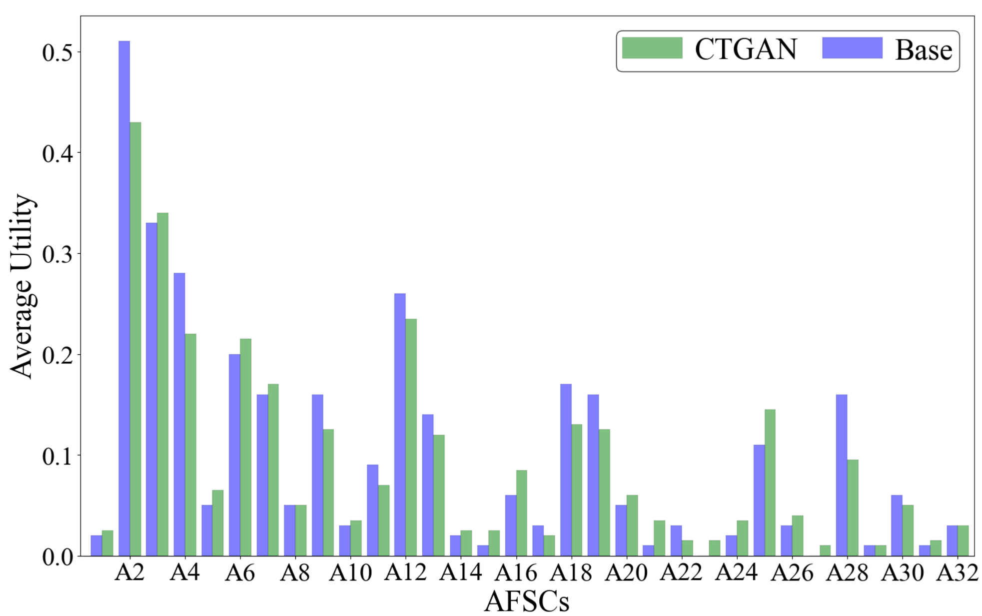


Figure 19: Average utility placed on each AFSC for both data types. These measures are averaged across all CTGAN generated datasets and compared to the baseline real data.

their proportions and re-label them 1 to 49. The averaged generated data can now be compared to the real data using this histogram, as shown in Figure 20.

There is no simple explanation for why the distribution of degrees for the CTGAN data appears dissimilar to that of the real data. Perhaps the method of binning the CIP codes and subsequently sorting them is not the best way of visualizing this data, which could mean that the distribution really is captured by CTGAN. Perhaps the distribution is not accurately represented within CTGAN, and some degrees are generated more often in CTGAN than are found in the real data. Either way, differences in the frequency of cadet degrees is no cause for rejection of the CTGAN model since it is only used to test the solution methodologies presented in Section 4.1.2. Every class year is different, and the model should be capable of assigning all kinds of unique sets of cadets, even when they differ from what is expected. Therefore, CTGAN is used to generate synthetic problem instances for model performance analysis.

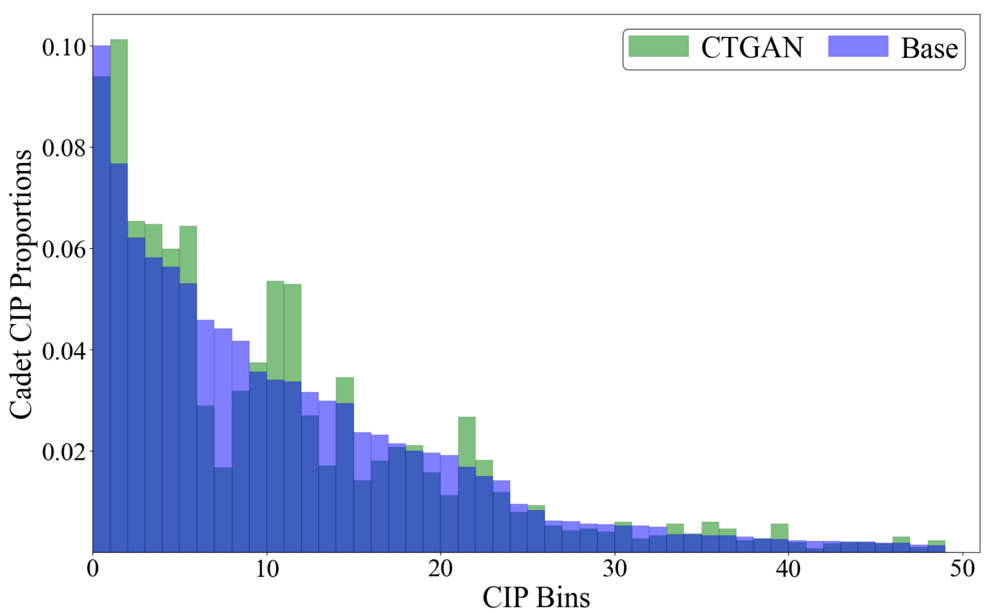


Figure 20: Average CIP sorted bins histogram across all synthetic data compared to the real data. There are five CIP codes in each bin, determined through numeric order. The bins are then sorted based on the number of occurrences in the real data of cadets with the bins' specific CIP codes.

Appendix C. Heuristic Utility Matrices

A greedy algorithm, and a stable marriage algorithm adaptation, are used as constructive heuristics to provide initial solutions to the genetic algorithm (GA) for a given problem instance. For these two heuristics, two different utility matrices are incorporated to approximate the Value-Focused Thinking (VFT) objective hierarchy of the problem instance. The cadet utility matrix used is the same utility matrix as discussed in the paper (the *utility* parameter). For the AFSC utility matrix, however, we approximate it by using several of the AFSC objectives. The five objectives incorporated in these calculations are Balance Merit, Maximize Utility, and the three AFOCD tier proportion objectives. Just like the cadet utility matrix, this can also be represented as an $N \times M$ matrix, where N is the number of cadets and M is the number of AFSCs. We define $merit^m$ as an $N \times M$ matrix where the elements correspond to the values each AFSC j would receive for the Balance Merit objective if the AFSCs were composed entirely of each cadet i. For instance, if one cadet is in the 45^{th} percentile of their graduating class, the value of that cadet for a given AFSC would be determined by the value obtained from an average merit of 0.45 through the merit value function.

For the degree tier objectives, we define the matrices $mandatory^m$, $desired^m$, $permitted^m$, and $ineligible^m$ to denote the eligibility of a particular cadet for a particular AFSC as described in the AFOCD. These are binary matrices, except $ineligible^m$ which contains a very large number for cadets that are ineligible for certain AFSCs. The last objective – maximize the utility of the cadets assigned to each AFSC – uses the same cadet utility matrix as discussed before (*utility*). All other parameters used are defined in the VFT formulation in Section 3.3.3. We abbreviate the set of five objectives {Balance Merit, Mandatory Tier Proportion, Desired Tier Proportion, Permitted Tier Proportion, Maximize Utility} as $k \in \{1, 2, 3, 4, 5\}$, for ease of notation. The $N \times M$ matrices corresponding to these objectives ($merit^m$, $mandatory^m$, etc.) can also be defined using their indices. Lastly, we define $value_{ijk}$ as the value of cadet i for AFSC j in objective matrix k. The values composing the AFSC utility matrix ($afsc\_utility_{ij}$) are calculated as shown in equation (24).

$$afsc\_utility_{ij} = \sum_{k=1}^{5} objective\_weight_{jk} \cdot value_{ijk} - ineligible^m_{ij} \quad \forall i \in \mathcal{I},\ j \in \mathcal{J} \tag{24}$$

In the stable marriage algorithm, this utility matrix is used to generate the AFSC preference lists for cadets. Although criticized in Section 3.2.2, these utility matrices are meant to provide a loose estimate of the real values of the AFSCs. The real values are still calculated by the GA using the VFT framework. For each AFSC (the columns of the matrix), the ordinal lists of cadets are calculated by sorting the values of the column in descending order. The cadet preference lists for AFSCs are done in a similar manner using the cadet utility matrix by sorting each *row* in descending order. Once the preference lists are determined, we then apply the "many-to-one" hospital and residents variant of the stable marriage algorithm on this problem. The cadets that are left unmatched by this algorithm are assigned afterwards using the greedy method. The greedy algorithm assigns every cadet to the AFSC that provides the greatest "overall value" for that cadet. The overall value matrix is determined by aggregating the previously discussed utility matrices ($afsc\_utility$ and $utility$) using the weights on cadets and AFSCs. The elements of this new matrix ($overall\_value_{ij}$) can be calculated as

$$\begin{aligned} overall\_value_{ij} = \; & cadets\_overall\_weight \cdot cadet\_weight_i \cdot utility_{ij} + \\ & afscs\_overall\_weight \cdot afsc\_weight_j \cdot afsc\_utility_{ij} \end{aligned} \tag{25}$$

for all cadets (i) and AFSCs (j). The greedy method iterates over all cadets and assigns each to the AFSC that results in the largest overall value.

Appendix D. Genetic Algorithm Structure

The genetic algorithm (GA) implemented in this paper follows a very similar architecture to others in the literature. The population used in the algorithm is a list of "chromosomes" representing the solutions generated by the GA. Typically chromosomes are binary vectors that correspond to decision variables, or "genes," but for this problem we define a chromosome as a vector of length N (number of cadets) containing the indices of the AFSCs that the cadets are assigned to. A gene in the chromosome is therefore an AFSC index for a particular cadet. For example, consider a solution containing $N = 4$ cadets, $i \in \{1, 2, 3, 4\}$, and $M = 3$ AFSCs, $j \in \{1, 2, 3\}$. In this example, let cadet 1 be assigned to AFSC 3, cadet 2 be assigned to AFSC 2, cadet 3 be assigned to AFSC 3, and cadet 4 be assigned to AFSC 1. If we assume all cadets are eligible for all AFSCs, we can represent the solution as an $N \times M$ matrix. Using the variable, $\boldsymbol{x}$, in matrix form, the solution is

$$\boldsymbol{x} = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix} \tag{26}$$

and the chromosome used by the GA in the population is

$$chromosome = \begin{bmatrix} 3 & 2 & 3 & 1 \end{bmatrix}. \tag{27}$$

Chromosomes are evaluated according to their fitness, which is calculated using the objective function with one slight modification. To incorporate the value constraints used in the model, we multiply the objective value, $\boldsymbol{Z}$, by a binary variable indicating if the solution meets all of the constraints (1) or not (0). This is the new fitness score. Additionally, because the approximate model sometimes fails certain constraints due to rounding issues (by one cadet at most), we allow the GA to fail the same constraints in the same manner. This is to allow feasible solutions to be evaluated from the beginning, so the approximate model initial solution's fitness is not 0. Since certain objectives are very strict on constraint failures, the GA almost always finds solutions that completely meet more of those constraints anyway. In fact, the final solution produced by the GA usually meets all constraints, or all but one, exactly. Many parameters are necessary for this algorithm, including those described in the optimization model's formulation as well as the various GA specific hyper-parameters. The hyper-parameters to this GA greatly affect the performance of the algorithm, and should be specified intently.

The size of the population determines how many chromosomes are generated and subsequently measured at each iteration. The fitness function for this GA is very costly, since 90% of the time in each iteration is spent evaluating chromosome fitness. To avoid extensive computation times, we implement a population size of six. The stopping criteria used by this GA can be either a specified number of iterations, as shown in Algorithm 1, or a specified amount of time, as described in Section 4.1.2. The number of crossover points determines how many segments are split for two chromosomes in the multi-point crossover function to generate new solutions. We use three crossover points for this GA. The mutation rate is another standard hyper-parameter found across the literature. A gene mutation on cadet i exchanges that cadet's assigned AFSC for a random AFSC drawn from the set of AFSCs that the cadet is eligible for, $\mathcal{J}_i^E$. The mutation rate used in this GA is 0.1. A separate hyper-parameter controls the number of genes that can be selected for the chance of mutation. We force this hyper-parameter to be $\frac{N}{75}$, rounded to the nearest integer. We present the generalized form of this GA in Algorithm 1.

Algorithm 1 Genetic Algorithm

```
input parameters
population = Initialize_Population(initial_solutions)
for each iteration do
    fitness, population = Calculate_Sort_Fitness(population)
    next_generation = [population_1, population_2]
    for each pair do
        parents = Select_Parents(population, weights)
        offspring = Multi_Point_Crossover(parents)
        offspring = Mutation(offspring)
        Append(next_generation, offspring)
    end for
    population = next_generation
end for
return population_1
```

After the parameters to the GA have been specified, the population is initialized with the solutions described in Section 4.1.2. For the main solution methodology, the initial population includes the greedy solution, stable marriage solution, and the approximate model solution. The other solutions are generated by assigning all of the cadets to random AFSCs for which they are eligible. At each iteration, the algorithm evaluates the fitness of all chromosomes in the population, and sorts the population by their fitness scores. The top two chromosomes are retained for the next generation. Since the population size used in this research is six, there are therefore four more chromosomes that must be generated and added to the next generation. This equates to two pairs of "offspring." The offspring, in this context, refers to the children that are "born" from their two parents. Because the top two chromosomes always remain in the population for the next generation, one of the parents of the offspring could be in the same generation. Therefore, offspring do not necessarily replace their parents in this context.

To select the parents for each pair of offspring, we incorporate a selection function using fitness scores. Probability weights of selecting the parents are created using the ordinal fitness rankings of the chromosomes. For each pair of offspring, parents are selected, without replacement, from the population using those probabilities. Ordinal rankings are incorporated rather than the fitness values themselves to better distinguish the superiority of more fit solutions. In later generations, the fitness values may all be close to one another in magnitude, which dilutes the contrast between the better chromosomes in the population from the worse ones. Additionally, because constraint violations result in fitness values of 0, many chromosomes would have no chance of being selected if fitness values were used directly as weights. Therefore, the GA incorporates ordinal rankings to select the parents. Once the parents are selected, a multi-point crossover function is applied on the chromosomes to generate the offspring. The offspring then have the chance to mutate according to the mutation parameters specified, and are added to the next generation. Once the next generation is completely determined, it becomes the population of the succeeding iteration.

Appendix E. Value Parameter Generation

To test the validity of the new VFT model against the original model on real class year results, an authentic value hierarchy must be implemented. Discussions with AFPC on the importance of certain objectives relative to each other help inform the weights on all of the attributes of the model. These discussions provide adequate understanding of how important all of the objectives are relative to each other. The trade-offs between the objective measures within each objective must also be discussed. These relationships are captured by the value functions. Additionally, it is often the case that the decision maker (DM) must constrain certain objective measures. Solutions are considered infeasible by the Air Force if some objectives are not met, even if the distribution of quality within the solution is quite strong otherwise. Therefore, objective measure constraints may be imposed without altering the weights and value functions of the model.

These three components make up the value parameters of the VFT model, since they can be altered without changing the "fixed" parameters of the model (cadet data). Although we have a fair understanding of what these value parameters should be, without meeting with all decision makers involved in this process (USAFA/ROTC leadership, career field managers for all AFSCs, and the leadership overseeing this task), it is impossible to know what the real value hierarchy is exactly. We must estimate it using the meetings with the DM as a guide. Going further, the model's robust nature can be shown by testing it on real instances using many different sets of value parameters. Given an estimate of what the DM believes the real objective hierarchy to be, a degree of randomness can be added to sample all kinds of unique value hierarchies. This appendix discusses the methods of generating these sets of value parameters.

E.1 Objective Hierarchy Methodology

The first component of the objective hierarchy to estimate is the overall weight on cadets and AFSCs. This is the most important "fundamental" objective weight to estimate, since this has the largest effect on the importance of the "means" objectives. The overall weight on cadets is assumed to be about 0.35. Because this weight is an estimate, we sample the overall weight on cadets from a normal distribution with a mean of 0.35 and a standard deviation of 0.2 for each set of value parameters. Not only does this random sampling technique estimate the true weight on cadets relative to the AFSCs, but it also illustrates the model's robustness across all sorts of opinions on the importance of cadet happiness versus AFSC criteria. To eliminate extreme values, we force these weights to fall in the range between 0.1 and 0.6. For the remainder of this appendix, we refer to a generated value parameter by either its name or by "v," and the distribution it follows as $v \sim \mathcal{N}(\mu,\ \sigma) \in [l,\ u]$ where μ is the estimated population mean, σ is the estimated population standard deviation, and $[l,\ u]$ are the lower and upper bounds on the range of values that the parameter can take. Thus, the distribution we use for the overall weight on cadets is $v \sim \mathcal{N}(0.35,\ 0.25) \in [0.1,\ 0.6]$. As a reminder, the overall weight on AFSCs is determined by the overall weight on cadets as $1\ -\ cadets\_overall\_weight$.

For the individual weights on cadets, as well as the remaining weight definitions in the objective hierarchy, we first distinguish "swing weights," "local weights," and "global weights." A swing weight is one method of specifying objective weights without the constraint that the weights must sum to 1. For instance, the DM could allocate swing weights of 100, 40, and 20 to three different objectives. When these three weights are each individually divided by the total sum, the "local weights" are determined. Since $100 + 40 + 20 = 160$, the local weights are $\frac{100}{160} = 0.625$, $\frac{40}{160} = 0.25$, and $\frac{20}{160} = 0.125$, respectively. The local weights for all objectives are the weights that sum to 1 relative to the other local objectives. The global weights are the weights on all of the model objectives relative to all of the objectives as a whole. For example, if 0.625, 0.25, and 0.125 are the local weights for three cadets, and the overall weight on cadets is 0.35, then the global weights would be $0.625 \times 0.35 = 0.21875$, $0.25 \times 0.35 = 0.0875$, and $0.125 \times 0.35 = 0.04375$.

We present three different methods of generating cadet weights. The first method ascribes equal weight to all cadets; higher performing cadets are not rewarded. The second and third methods determine cadet weight through a function of the cadets' percentiles (their graduating order of merit). Specifically, the methods utilize a linear function of merit as well as an exponential function of merit. These functions are depicted in Figure 21. To assign a cadet weight function to a set of value parameters, probabilities are ascribed to each function. Since it is likely that higher performing cadets should be rewarded with a higher objective importance in the hierarchy, we

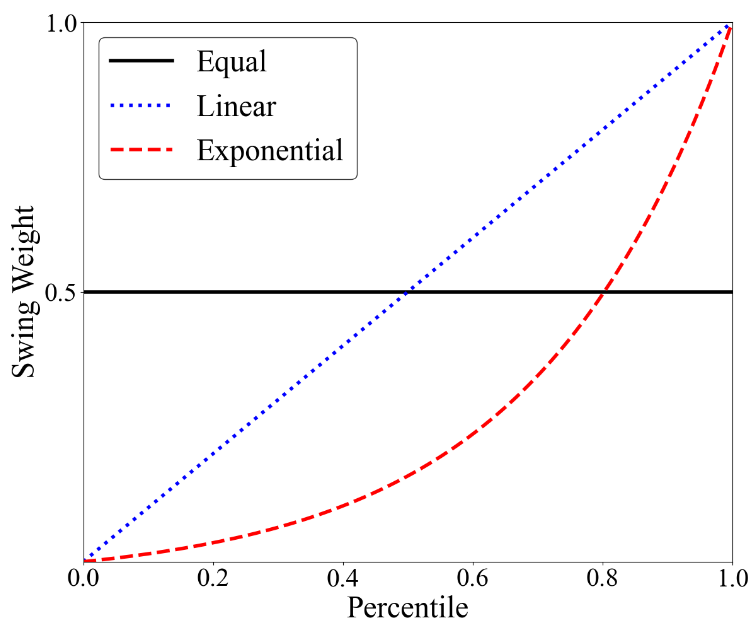


Figure 21: Different cadet weight functions. Cadet weights may either be distributed equally or determined through a function of their graduating order of merit. Specifically, a linear function of merit and an exponential function of merit are provided as options. The weights generated may then be scaled afterwards to sum to 1.

assign a probability of 0.05 for selecting the “cadet equal” function. To be more equitable to cadets across the range of percentiles, the most likely function to be used is the linear function. This function is given a probability of 0.7. The exponential function is therefore assigned a probability of 0.25. To add another random component to the importance of cadets, we alter the “steepness” of the exponential curve. The function is given by

$$y \;=\; \frac{1-e^{-\frac{x}{\rho}}}{1-e^{-\frac{1}{\rho}}} \tag{28}$$

where ρ determines that steepness. For the exponential functions generated for the sets of value parameters, ρ is sampled from the distribution $\rho \sim \mathcal{N}(0.3,\ 0.1) \in [0.1,\ 0.5]$. For AFSC weights, the model provides four possible methods of local weight specification. Similar to cadets, the first method is to treat all AFSCs equally. In the second method, an AFSC's weight is calculated as a linear function of its quota (a proxy for size). The third method also uses a function of the AFSC's size, but one that is more “forgiving” to smaller AFSCs. We implement a piece-wise function to narrow the gaps between the different sized AFSCs. This function is presented in equation (29), where $afsc\_swing\_weight_j$ is the swing weight on AFSC j.

$$afsc\_swing\_weight_j = \begin{cases} 1 & \text{if } quota_j \geq 200 \\ 0.9 & \text{if } 150 \leq quota_j < 200 \\ 0.8 & \text{if } 100 \leq quota_j < 150 \\ 0.7 & \text{if } 50 \leq quota_j < 100 \\ 0.6 & \text{if } 25 \leq quota_j < 50 \\ 0.5 & \text{otherwise} \end{cases} \tag{29}$$

The fourth method samples a quota multiplier from the distribution given by $v \sim \mathcal{N}(1,\ 0.15) \in [0.5,\ 1.5]$ for each AFSC. The swing weight for the AFSC is calculated by multiplying the AFSC's quota by that sampled scalar value. Again, probabilities of selection are assigned to each of these four swing weight functions. The "AFSC equal" method is given a probability of 0.1, and the remaining three methods are each assigned probabilities of 0.3.

Various methods of generating weights are applied for the different AFSC objectives. These methods are based not only on the type of objective itself, but also the characteristics of the AFSC. For the Balance Merit and Balance USAFA Proportion objectives, three methods of determining which distributions to sample from are included. Since the Air Force specifies a delineation between small and large AFSCs, two of these methods will also. These methods are presented in Table 6. Method 1 assumes the weights on large and small AFSCs should be calculated the same way, and so they are both calculated from the same distributions. In Method 2, the weights on

Table 6: Weight generation methods for the AFSC objectives that balance certain cadet criteria. For each objective, three different methods of determining the AFSC objective weights are presented based on the size of the AFSCs (large or small). Method 1 does not distinguish between large and small AFSCs while the other two methods do.

Objectives	Method 1	Method 2	Method 3
Balance Merit (Large AFSCs)	$v \sim \mathcal{N}(30,\ 5) \in [10,\ 50]$	$v \sim \mathcal{N}(40,\ 10) \in [20,\ 60]$	$v \sim \mathcal{N}(40,\ 10) \in [20,\ 60]$
Balance Merit (Small AFSCs)	$v \sim \mathcal{N}(30,\ 5) \in [10,\ 50]$	$v = 0$	$v \sim \mathcal{N}(10,\ 2) \in [5,\ 15]$
Balance USAFA (Large AFSCs)	$v \sim \mathcal{N}(20,\ 5) \in [5,\ 35]$	$v \sim \mathcal{N}(30,\ 10) \in [10,\ 50]$	$v \sim \mathcal{N}(40,\ 10) \in [20,\ 60]$
Balance USAFA (Small AFSCs)	$v \sim \mathcal{N}(20,\ 5) \in [5,\ 35]$	$v = 0$	$v \sim \mathcal{N}(5,\ 1) \in [0,\ 10]$

the objectives for small AFSCs are 0 while the weights on large AFSCs have higher importance. This is similar to how the constraints are specified in the original model. Method 3, on the other hand, is a hybrid of the first two. Here, the objective weights for large and small AFSCs are both nonzero, but the importance of balancing the larger AFSCs rather than the smaller ones is highlighted. For both objectives, the probabilities of selecting Method 1, Method 2, and Method 3 are 0.3, 0.3, and 0.4, respectively.

In the combined quota objective, the weight chosen is sampled from the distribution $v \sim \mathcal{N}(100,\ 5) \in [90,\ 110]$. Similarly, the "Maximize Cadet Utility" objective weight is sampled from the distribution $v \sim \mathcal{N}(35,\ 20) \in [10,\ 60]$ for all AFSCs. To determine the weight on each of the AFOCD degree tiers, we consider the degree tiers that are included in each AFSC. Let the letters "M", "D", and "P" denote the mandatory, desired, and permitted degree tiers that are available for each AFSC, respectively. For example, if an AFSC has a mandatory degree tier and a permitted degree tier, but not a desired degree tier, its tier structure is represented as "MP." If an AFSC has all three, the tiers are denoted as MDP. This allows for further differentiation between AFSCs based on AFOCD importance. For each AFSC j, the distribution on the swing weight for the Mandatory Degree Tier Proportion objective ($mandatory\_swing\_weight_j$) is given by equation (30).

$$mandatory\_swing\_weight_j \sim \begin{cases} \mathcal{N}(90,\ 5) \in [80,\ 100] & \text{if } M \\ \mathcal{N}(80,\ 5) \in [70,\ 90] & \text{if } MD \\ \mathcal{N}(80,\ 5) \in [75,\ 85] & \text{if } MP \\ \mathcal{N}(80,\ 10) \in [60,\ 100] & \text{if } MDP \\ 0 & \text{otherwise} \end{cases} \tag{30}$$

Similarly, the distribution on the Desired Degree Tier Proportion objective swing weight ($desired\_swing\_weight_j$) is calculated through equation (31).

$$desired\_swing\_weight_j \sim \begin{cases} \mathcal{N}(40,\ 5) \in [30,\ 50] & \text{if } MD \\ \mathcal{N}(50,\ 5) \in [40,\ 60] & \text{if } MDP \\ \mathcal{N}(60,\ 5) \in [50,\ 70] & \text{if } DP \\ 0 & \text{otherwise} \end{cases} \tag{31}$$

Lastly, the distribution on the Permitted Degree Tier Proportion objective swing weight ($permitted\_swing\_weight_j$) is shown in equation (32).

$$permitted\_swing\_weight_j \sim \begin{cases} \mathcal{N}(20,\ 5) \in [10,\ 30] & \text{if } MP \\ \mathcal{N}(30,\ 5) \in [20,\ 40] & \text{if } MDP \\ \mathcal{N}(30,\ 5) \in [20,\ 40] & \text{if } DP \\ 0 & \text{otherwise} \end{cases} \tag{32}$$

The remaining objectives – both the USAFA/ROTC quota objectives and the Male/Minority balancing objectives – have objective weights of 0 and are not used in the VFT model comparison tests. The local weight on objective k for each AFSC j, $objective\_weight_{jk}$, for all AFSC objectives may now be generated using the objective swing weights. Once all weights are determined, the value functions for the AFSC objectives must be constructed.

E.2 Value Function Methodology

The value functions implemented in this section are the practical applications of the modeling formulation additions presented in Section 3.4.3. These piece-wise linear functions are created from the linearization of various piece-wise exponential functions. To construct the piece-wise exponential functions, we connect 1 to 4 separate exponential functions as presented in equation (28) together. We can specify the target measure of the value function, the ρ parameters of the different exponential pieces, and many other parameters to construct any kind of value function the DM desires. Once the exponential function is determined, we may then specify the number of breakpoints used in the optimization models for the linearization of the function. The more breakpoints used, the closer the function appears to be non-linear, as well as the more complex the function becomes for the solver. An example using four different exponential functions pieced together is presented in Figure 22. The different exponential segments are divided by dashed lines and labeled. The three points on the graph break up the exponential segments of the function. This function has a target of 5, and ascribes value equally to measures on either side of that target.

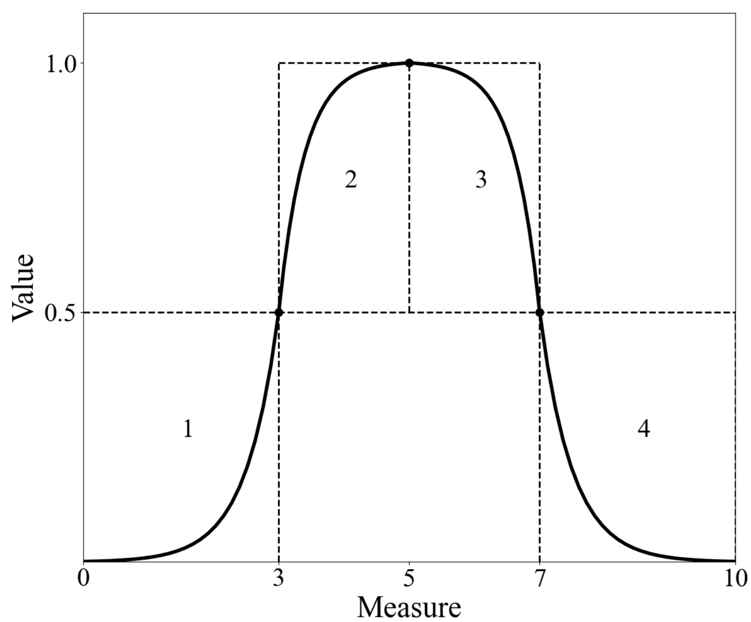


Figure 22: Piece-wise exponential value function example with four exponential "segments." Breakpoints are specified to connect the four segments together. Once the exponential value function is determined, the function may be linearized through the allocation of breakpoints that divide the non-linear segments into many linear ones.

Unsurprisingly, different AFSC objectives use different value functions. For the "Balance Merit" objective, we consider four exponential segments in the piece-wise function. The first step in creating the complete piece-wise function is to determine the target measure for the given AFSC objective. For average merit, that target measure is the average percentiles of the entire class of cadets, which should be about 0.5. However, since this objective is specific to each AFSC, the value function should consider the AFSC's own breakdown of percentiles for its set of eligible cadets. Since many AFSCs have eligibility requirements that reduce the set of candidate cadets from all cadets to some smaller subset of them, the percentiles of the cadets that are eligible for each AFSC may be higher or lower than 0.5 on average. For class year C, this concept is illustrated in Figure 23.

The Balance Merit value functions should be more fair to AFSCs that have sets of lower performing eligible cadets on average, because it may be harder to meet the 0.5 target for them. Additionally, since the goal is to "balance" the average merit of the cadets assigned to each AFSC, the value functions should also consider the AFSCs with higher performing cadets. Going further, although the objective is

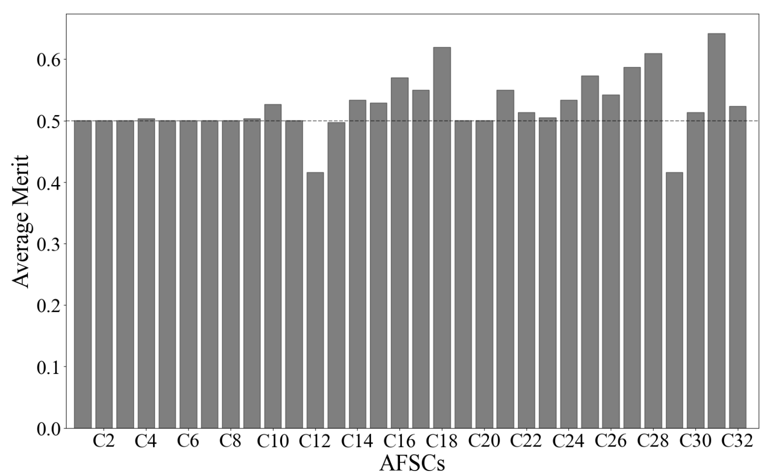


Figure 23: Average merit of the eligible cadets for each AFSC in class year C. The expected average merit is 0.5, but due to the variance in the sets of eligible cadets for each AFSC, some AFSCs may contain higher or lower performing cadets, on average.

to meet an average merit of 0.5 for all AFSCs, measures above this target should not be penalized as much as measures below. For example, an AFSC individually would be satisfied with an assignment of higher performing cadets, but less so with an assignment of lower performing cadets. Therefore, the curve for measures above 0.5 is not as steep as the curve for measures below 0.5. As a whole, however, AFSCs that receive higher performing cadets skew the balance for other AFSCs since they will have to be assigned lower performing cadets as a result. Additionally, meeting the target of 0.5 for all AFSCs is theoretically achievable, though not plausible. This is why the target for all AFSCs is set to 0.5 and not some higher measure.

In a similar manner used to generate the weights in the previous section, a degree of randomness is added to the method of creating the value functions for all AFSCs across all class year instances. For the Balance Merit objective, the controlled parameters of the function are the main breakpoints that split apart its exponential segments. The target is the average merit of the entire class (at or around 0.5) and has a value of 1. We define the left and right "margins" of the function as the distance in the x-axis between the target and each of the left and right breakpoints, respectively, that split the function's exponential segments. Let the average merit of the eligible cadets for AFSC j be denoted as $merit_j^E$. If $merit_j^E = 0.5$, then the left and right margins are 0.1 and 0.14, respectively. If $merit_j^E < 0.5$, then the left and right margins are $\frac{target - merit_j^E}{4} + 0.1$ and 0.14, respectively. If $merit_j^E > 0.5$, the left margin is 0.1, and the right margin is $\frac{merit_j^E - target}{4} + 0.14$. The y values associated with these two breakpoints follow the distribution $v \sim \mathcal{N}(0.75,\ 0.5) \in [0.65,\ 0.85]$. The ρ parameter for the first exponential segment is assigned the distribution $v \sim \mathcal{N}(0.08,\ 0.01) \in [0.05,\ 0.11]$. The middle two ρ parameters use the same normal distribution as the first but fall in the bounds $[0.06,\ 0.10]$. The last ρ parameter is sampled from the distribution $v \sim \mathcal{N}(0.125,\ 0.01) \in [0.1,\ 0.15]$. Three different value functions for Balance Merit

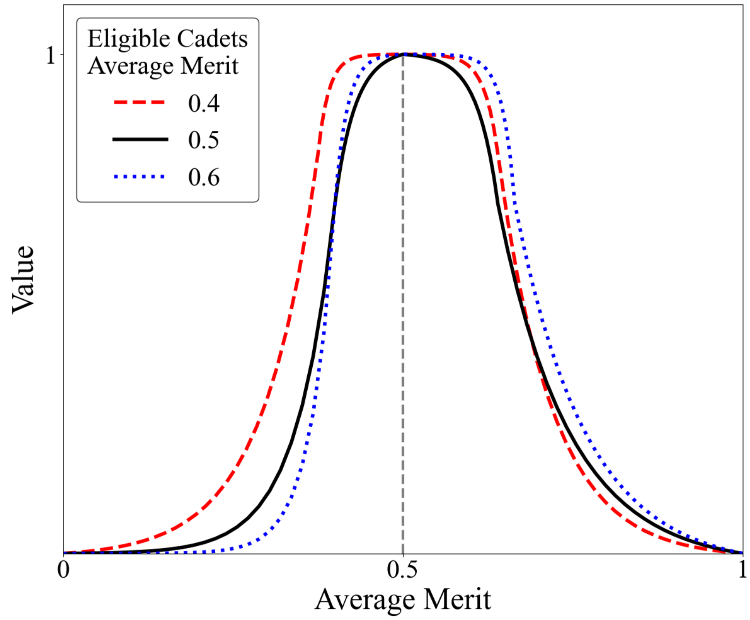


Figure 24: Example average merit value functions using different function parameters for AFSCs with different sets of eligible cadets. The three functions shown are applied on different sets of cadets. The dashed line applies to a theoretical AFSC that has lower performing eligible cadets and the dotted line represents the function for an AFSC with higher performing eligible cadets. The target, however, for all AFSCs is still 0.5.

are shown in Figure 24 using different levels for $merit_j^E$, and various function parameters.

For the USAFA proportion-balancing objective, we use a very similar approach as discussed with the Balance Merit objective with a few key changes. Like average merit, the actual proportion of eligible USAFA cadets for each AFSC helps determine the value function used. The overall USAFA proportion of class year C is about 0.24. The USAFA proportion of eligible cadets for each AFSC in class year C is shown in Figure 25. Although this is a minor deviation, the AFSCs for which all degrees are eligible (C1, C2, C3, for example) have USAFA cadet proportions that are slightly above 0.24. This is due to some AFSCs having scholarships that force many of the cadets into one of those AFSCs. The analysts at AFPC address this situation by modifying the eligibility of those cadets. These cadets are made ineligible for all

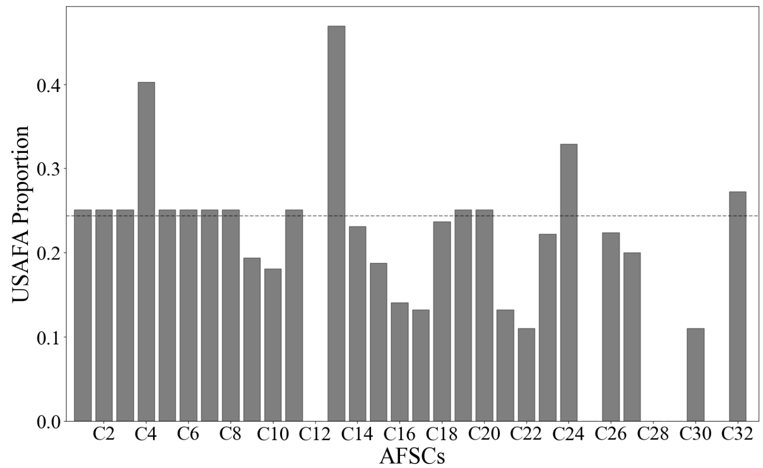


Figure 25: USAFA proportion of the eligible cadets for each AFSC in class year C. The overall proportion of USAFA cadets is about 0.24, but just as before, the variability in the sets of eligible cadets for each AFSC results in different proportions of eligible USAFA cadets. Note that the variability here is much more pronounced than with average merit.

other AFSCs, thus the USAFA proportion of the other AFSCs that should have all cadets eligible is skewed.

Additionally, some AFSCs have no eligible USAFA cadets while others have many. The variance on the USAFA proportion of eligible cadets is much higher than that on average merit. Leniency to AFSCs with drastically different USAFA proportions than that of the overall class is therefore critical. Three value functions using different USAFA proportions for the eligible cadets of a given AFSC, as well as different ρ and margin parameters are presented in Figure 26. The selection distribution parameters are similar to those of the Balance Merit function, but are not listed here to avoid redundancy. There are occasional constraints preventing USAFA cadets from being assessed into certain AFSCs. This is incorporated using a simple downward convex exponential function (value of 1 for a measure of 0, and a value of 0 for a measure of 1). Additionally, if an AFSC does not specify a constraint, yet has no eligible USAFA cadets, the USAFA proportion objective is not included in its objective hierarchy (the objective weight is set to 0).

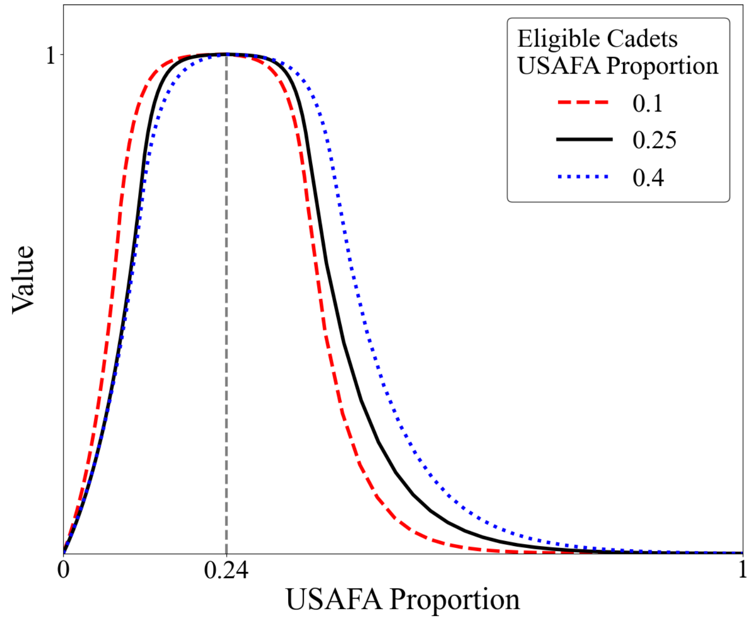


Figure 26: Example USAFA proportion value functions using different function parameters for AFSCs with different sets of eligible cadets. Just as with the Balance Merit objective, the three functions shown are applied on different sets of cadets. The dashed line applies to a theoretical AFSC whose set of eligible cadets is composed of 10% USAFA cadets, while the dotted line represents the function for an AFSC with an eligible set of cadets consisting of 40% USAFA cadets. The target, however, for AFSCs that want to balance this proportion is 0.24.

The combined quota objective value function is created in a more unique manner. Since AFSCs specify a range where they want the number of cadets assessed to fall within, we treat all such measures as equal in value. If the quota is met, and under the maximum number of cadets allowed, a value of 1 is achieved. There is a substantial decrease in value for measures outside of this range. The issue with this objective, however, is that the desired numbers of cadets for each AFSC are not known for sure. The differences between the constraints for this objective and the “indifference range” (the range that the number of cadets for the AFSC can fall within – generally the specified upper and lower bounds on the quota) must be reconciled. For example, the DM may specify the quota for an AFSC to be 100 cadets with an upper limit of 120. Perhaps, for one reason or another, this is infeasible. The analysts could then add a constraint to the optimization model and widen the bounds on their original targets. Maybe there are not enough cadets, so the constraint falls in the range [80, 120]. Perhaps we need to assign more cadets, and the range becomes [100, 140]. The intended target is still between 100 and 120, but the constraints have changed.

This is a real issue because the ranges are all specified within the problem instance formulation for class year C (this is used to estimate the upper bounds for all the other class years, though the lower bounds are known), yet some are not included in the constraints. They are simply "commented out" of the model's code. This is presumably a result of certain constraints causing model infeasibility, but it is difficult to know for sure. Therefore, there is a possibility that the range of indifference (used in the value functions) is tighter than the range on the constraints. Additionally, if the quota objective is infeasible purely because there are not enough eligible cadets for a certain AFSC to fill the quota, the targets must change. Most of the time, constraints are infeasible because of the interactions between all of them, but in this case the quota constraint is infeasible, independent of all the others. For the AFSC C12, this is a real complication in class year C as shown in Figure 27. C12 has a quota of 72 cadets, but only has 51 eligible cadets. These kinds of discrepancies necessitate the need for adjustment of the AFSC quotas themselves.

Because certain constraints are occasionally deactivated in the class years, we develop two different combined quota value function structures based on two different assumptions. For example, suppose that the maximum number of cadets that can be assigned to AFSC j in a certain class year is $1.3 \cdot quota_j$. If the constraint limits the number of cadets for AFSC j in a given class year to fall in the range $[quota_j,\ 1.3 \cdot quota_j]$, and the actual number of assigned cadets for the class year also falls in that range, the value function indifference bounds and the constraint bounds are assumed to be the same. We refer to this method as Method 1. If instead, the actual number

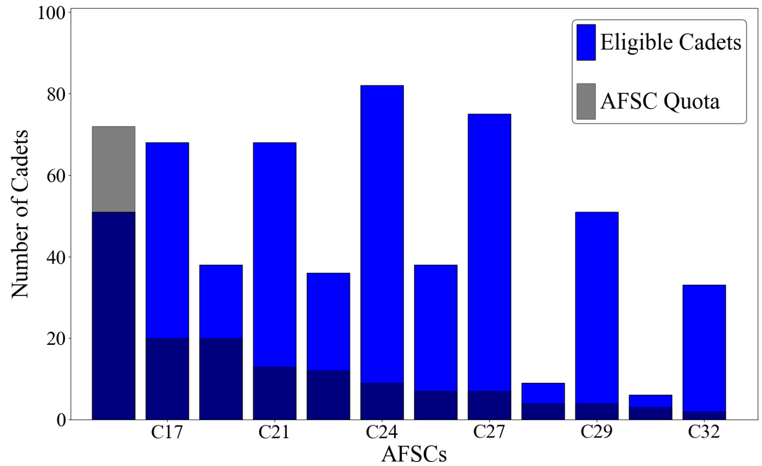


Figure 27: Target quotas and numbers of eligible cadets for AFSCs with less than 100 eligible cadets in class year C. The target for C12 is initially 72, but because the number of eligible cadets is 51, the quota is reduced.

of assigned cadets ($\boldsymbol{count}_j$) for AFSC j in the class year exceeds the upper bound on this range, we must do one of two things to estimate the true value function. Either the upper bound is increased for both the value function indifference range and the constraint (the range would be $[quota_j,\ \boldsymbol{count}_j]$), or the indifference range is maintained $[quota_j,\ 1.3 \cdot quota_j]$ for the value function but the range $[quota_j,\ \boldsymbol{count}_j]$ is used as the constraint. The former would take on a value function equivalent in structure to Method 1, with a wider indifference range. The latter, however, is handled by Method 2. Method 1 uses two convex exponential segments on either side of the indifference range. Method 2 contains three exponential segments to provide less of a penalty on measures that exceed the indifference bounds yet fall within the bounds of the constraint. The right margin (the breakpoint splitting exponential segments 2 and 3) of this piece-wise function is determined by the upper bound on the constraint. The margin's corresponding y-value and ρ parameters are sampled from normal distributions.

Suppose, for class year C, that an AFSC has a quota of 50 cadets with an upper bound of 65 $(1.3 \cdot 50)$. If the number of assigned cadets for that AFSC falls within that range (i.e. the analysts included that constraint), then the combined quota value function we choose for our model comparison tests utilizes Method 1. If 75 cadets are assigned to that AFSC instead, the indifference range as well as the constraint could be adjusted to fit Method 1, or only the constraint is adjusted and the indifference range is maintained. Measures above 65, but below 75, would not be penalized as severely as in the original Method 1 scenario. Figure 28 presents this example using all three scenarios. In the chart, Method 1 indicates the scenario where the number of assigned cadets in the real solution for class year C falls in the original range, but the other two functions indicate the scenario where that number exceeds the range. The Method 2 scenario assigns a value of 0.6 to the margin (the measure that splits exponential segments 2 and 3, 75 cadets in this case). For parameter generation, this value is sampled from a normal distribution.

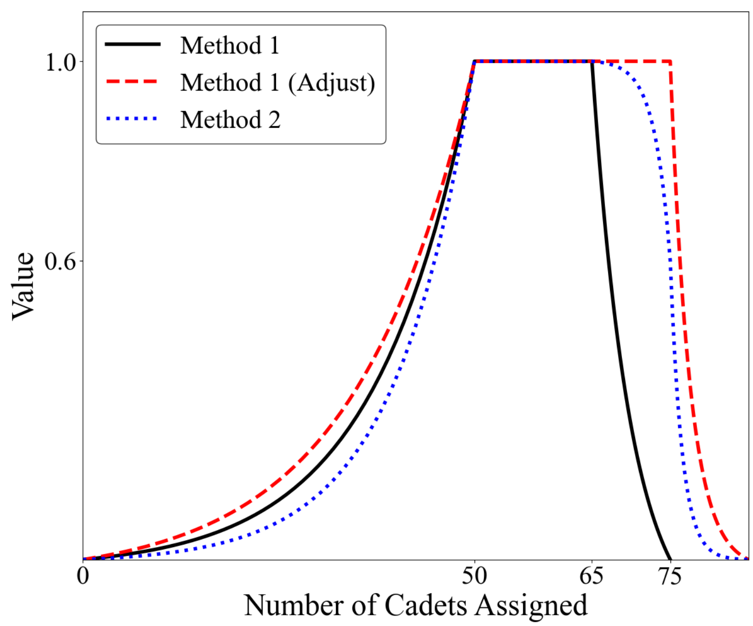


Figure 28: Example combined quota value functions. For this example, if the actual number of cadets assigned to a given AFSC in some class year is within the specified range [50, 65], we use the Method 1 function. If the actual number of cadets assigned to the AFSC is 75, however, we may adjust the upper bound on this range to 75 and utilize Method 1 or apply Method 2.

The AFOCD proportion objectives are simpler than the others. Only one exponential segment is used to link the indifference value (y = 1) with the y = 0 line. For the AFOCD tier objectives that seek the proportion of cadets with the tiers' degrees to be greater than or equal to some amount (the target AFOCD tier "accession rate"), we utilize an upward trending exponential curve with domain between 0 and the target amount. The AFSC is indifferent towards any measure above this target amount. For "less than or equal to" tier specifications, the reverse is true. This is a key assumption for this framework, since perhaps higher proportions are more preferred to the target accession rate. Nevertheless, since the only current available information is what is written in the AFOCD, we assume indifference. In a similar manner as the previous objectives, the ρ parameter is randomly sampled from a distribution. This is the only randomized parameter for these value functions. Three example value functions that can be used for the AFOCD objective value functions are presented in Figure 29. The "Maximize Cadet Utility" AFSC objective value function is the simplest of them all. It is an exponential function with a target of 1. The ρ parameter is the only aspect of the function that could change between the AFSCs. The male and minority balancing objectives, though not used in this paper's analysis, could follow an almost identical process to the USAFA proportion objective value functions. Additionally, the USAFA/ROTC quota objectives are similar to the combined quota objective and could have value functions constructed by the same methods.

Lastly, the model only imposes constraints on the combined quota, the mandatory AFOCD tier proportion, and average merit objectives. Because the original model only constrains the lower bound on the number of cadets with mandatory tiered degrees for each AFSC, we also constrain this fixed number, as opposed to the proportion of cadets assigned to the AFSC. The other constraint imposed on the model

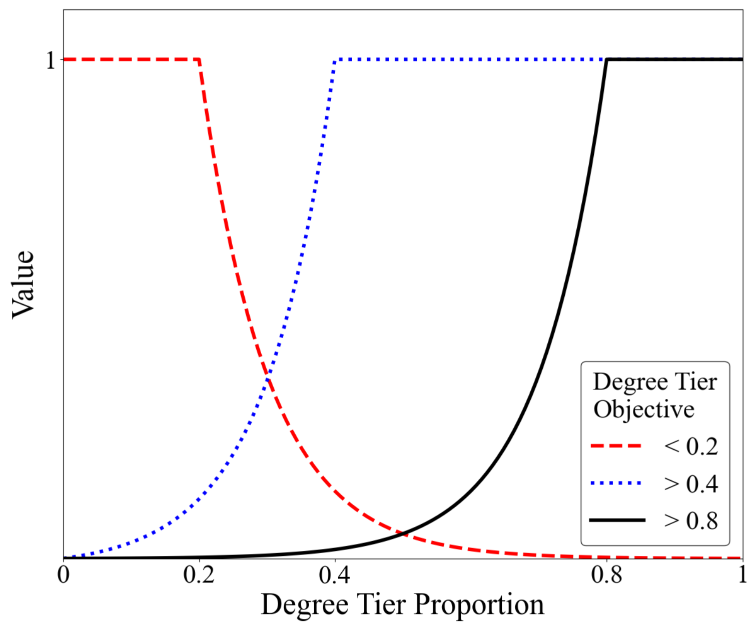


Figure 29: Example AFOCD degree tier proportion value functions. Each value function indicates a different target with either a greater than or less than inequality specification.

forces the number of assigned USAFA cadets to be under some fixed upper limit for certain AFSCs in the class years where this constraint is in effect. There are several reasons why the USAFA proportion objective is not constrained for the other AFSCs. The main reason is that this constraint is only specified for a few of the AFSCs in practice due to infeasibility issues. Another reason is that this constraint does not constrain the real proportions/averages as discussed in Section 3.2.2. The desired and permitted degree tier objectives are also not constrained, though they are not constrained in the original model either. The sets of value parameters generated through the methodologies presented in this Appendix are used to compare the new VFT model against the original model (see Section 4.4). These sets are meant to provide a diverse population of potential VFT objective hierarchies to show the robustness of the new model. No matter what the DM values in their solutions, the new model is an excellent method of generating alternatives.

Appendix F. Least Squares Sensitivity Analysis Formulation

Parameters

$value^t_{jk}$	solution t's value of objective k for AFSC j
$cadet\_value^t_i$	solution t's value for cadet i
$value^b_{jk}$	solution b's value of objective k for AFSC j
$cadet\_value^b_i$	solution b's value for cadet i
$objective\_weight^b_{jk}$	solution b's weight on objective k for AFSC j
$afsc\_weight_j$	weight on AFSC j used for both solutions
$afscs\_overall\_weight$	weight on all of the AFSCs used for both solutions
$cadet\_weight_i$	weight on cadet i used for both solutions
$cadets\_overall\_weight$	weight on all of the cadets used in both solutions
Δ	the amount by which the objective value for solution t should exceed that of solution b

Decision Variables

$\boldsymbol{objective\_weight}_{jk}$	new weight on objective k for afsc j
$\boldsymbol{Z}^t$	the new objective value for solution t $afscs\_overall\_weight \cdot \Big[\sum_{j\in\mathcal{J}} afsc\_weight_j \cdot \Big(\sum_{k\in\mathcal{K}_j} \boldsymbol{objective\_weight}_{jk} \cdot value^t_{jk}\Big)\Big] + cadets\_overall\_weight \cdot \big[\sum_{i\in\mathcal{I}} cadet\_weight_i \cdot cadet\_value^t_i\big]$
$\boldsymbol{Z}^b$	the new objective value for solution b $afscs\_overall\_weight \cdot \Big[\sum_{j\in\mathcal{J}} afsc\_weight_j \cdot \Big(\sum_{k\in\mathcal{K}_j} \boldsymbol{objective\_weight}_{jk} \cdot value^b_{jk}\Big)\Big] + cadets\_overall\_weight \cdot \big[\sum_{i\in\mathcal{I}} cadet\_weight_i \cdot cadet\_value^b_i\big]$

Formulation

$$\text{minimize} \quad \sum_{j \in \mathcal{J}} \sum_{k \in \mathcal{K}_j} (\boldsymbol{objective\_weight}_{jk} - objective\_weight^b_{jk})^2 \tag{33a}$$

$$\text{subject to} \quad \sum_{k \in \mathcal{K}_j} \boldsymbol{objective\_weight}_{jk} = 1 \quad \forall j \in \mathcal{J} \tag{33b}$$

$$\boldsymbol{Z}^t - \boldsymbol{Z}^b = \Delta \tag{33c}$$

$$\boldsymbol{objective\_weight}_{jk} \geq 0 \quad \forall j \in \mathcal{J},\ k \in \mathcal{K}_j \tag{33d}$$

The objective function (33a) minimizes the sum of squared differences between the original AFSC objective weights (those used for solution b) and the new ones. The objective weights for each AFSC are local weights, and must therefore sum to 1 as indicated in constraint (33b). The objective value of solution t must exceed that of solution b by amount Δ, as defined in constraint (33c). This amount is usually set to 0 to indicate equality. Lastly, constraint (33d) forces all objective weights to be non-negative.

Bibliography

1. David Gale and Marilda Sotomayor. Some remarks on the stable matching problem. *Discrete Applied Mathematics*, 11(3):223–232, Jul 1985.

2. Trisna Darmawansyah, Shweta Singh, G C Dubey, and Rajesh Shrivastava. A comparative analysis of assignment problem related papers. *IOSR Journal of Engineering (IOSRJEN)*, 2(8):1–15, Aug 2012.

3. Dirk G. Cattrysse and Luk N. Van Wassenhove. A survey of algorithms for the generalized assignment problem. *European Journal of Operational Research*, 60(3):260–272, Aug 1992.

4. H. W. Kuhn. The Hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 2(1-2):83–97, Mar 1955.

5. Avanish Kumar. A modified method for solving the unbalanced assignment problems. *Applied Mathematics and Computation*, 176(1):76–82, May 2006.

6. Ventepaka Yadaiah and V. V. Haragopal. A new approach of solving single objective unbalanced assignment problem. *American Journal of Operations Research*, 06(01):81–89, Jan 2016.

7. Nathan Betts and Francis J. Vasko. Solving the unbalanced assignment problem: simpler is better. *American Journal of Operations Research*, 06(04):296–299, Jun 2016.

8. Quazzafi Rabbani, Aamir Khan, and Abdul Quddoos. Modified Hungarian method for unbalanced assignment problem with multiple jobs. *Applied Mathematics and Computation*, 361:493–498, Nov 2019.

9. JH Holland. Genetic Algorithms. *Scientific American*, 267(1):66–73, Jul 1992.

10. P. C. Chu and J. E. Beasley. A genetic algorithm for the generalised assignment problem. *Computers and Operations Research*, 24(1):17–23, Jan 1997.

11. Dirk Beckmann and Ulrich Killat. A new strategy for the application of genetic algorithms to the channel-assignment problem. *IEEE Transactions on Vehicular Technology*, 48(4):1261–1269, Jul 1999.

12. Zakir Hussain Ahmed. Performance analysis of hybrid genetic algorithms for the generalized assignment problem. *IJCSNS International Journal of Computer Science and Network Security*, 19(9):216, Sep 2019.

13. Kanchana Sethanan and Rapeepan Pitakaso. Improved differential evolution algorithms for solving generalized assignment problem. *Expert Systems with Applications*, 45:450–459, Mar 2016.

14. Lale Özbakir, Adil Baykasoğlu, and Pinar Tapkan. Bees algorithm for generalized assignment problem. *Applied Mathematics and Computation*, 215(11):3782–3795, Feb 2010.

15. D. P. Bertsekas. The auction algorithm: a distributed relaxation method for the assignment problem. *Annals of Operations Research*, 14(1):105–123, Dec 1988.

16. Dimitri P. Bertsekas. The auction algorithm for assignment and other network flow problems: a tutorial. *Interfaces*, 20(4):133–149, Aug 1990.

17. Eitan Wacholder. A neural network-based optimization algorithm for the static weapon-target assignment problem. *ORSA Journal on Computing*, 1(4):232–246, Nov 1989.

18. D. Gale and L. S. Shapley. College admissions and the stability of marriage. *The American Mathematical Monthly*, 69(1):9–15, Jan 1962.

19. Robert W. Irving, Paul Leather, and Dan Gusfield. An efficient algorithm for the “optimal” stable marriage. *Journal of the ACM (JACM)*, 34(3):532–543, Jul 1987.

20. D. G. McVitie and L. B. Wilson. The stable marriage problem. *Communications of the ACM*, 14(7):486–490, Jul 1971.

21. Alvin E. Roth and Marilda Sotomayor. The college admissions problem revisited. *Econometrica*, 57(3):559, May 1989.

22. Kazuo Iwama and Shuichi Miyazaki. A survey of the stable marriage problem and its variants. *Proceedings - International Conference on Informatics Education and Research for Knowledge-Circulating Society, ICKS 2008*, pages 131–136, Mar 2008.

23. Ruth Martínez, Jordi Massó, Alejandro Neme, and Jorge Oviedo. Single agents and the set of many-to-one stable matchings. *Journal of Economic Theory*, 91(1):91–105, Mar 2000.

24. David F. Manlove, Robert W. Irving, Kazuo Iwama, Shuichi Miyazaki, and Yasufumi Morita. Hard variants of stable marriage. *Theoretical Computer Science*, 276(1-2):261–279, Apr 2002.

25. Pter Bir, Tams Fleiner, Robert W. Irving, and David F. Manlove. The college admissions problem with lower and common quotas. *Theoretical Computer Science*, 411(34-36):3136–3153, Jul 2010.

26. J. M. Sánchez-Lozano, J. Serna, and A. Dolón-Payán. Evaluating military training aircrafts through the combination of multi-criteria decision making processes

with fuzzy logic. A case study in the Spanish Air Force Academy. *Aerospace Science and Technology*, 42:58–65, Apr 2015.

27. Ralph L. Keeney. *Value-focused thinking: a path to creative decisionmaking.* Harvard University Press, 1997.

28. Ralph L. Keeney. Making better decision makers. *Decision Analysis*, 1(4):193–204, Dec 2004.

29. Orfelio G. Leon. Value-focused thinking versus alternative-focused thinking: effects on generation of objectives. *Organizational Behavior and Human Decision Processes*, 80(3):213–227, Dec 1999.

30. Hong Sheng, Fiona Fui Hoon Nah, and Keng Siau. Strategic implications of mobile technology: a case study using value-focused thinking. *The Journal of Strategic Information Systems*, 14(3):269–290, Sep 2005.

31. Joseph L. Arvai, Robin Gregory, and Timothy L. McDaniels. Testing a structured decision approach: value-focused thinking for deliberative risk communication. *Risk Analysis*, 21(6):1065–1076, Dec 2001.

32. Wolfram Stadler. *Multicriteria optimization in engineering and in the sciences.* Springer, 1988.

33. Yacov Y. Haimes and Vira Chankong. Kuhn-Tucker multipliers as trade-offs in multiobjective decision-making analysis. *Automatica*, 15(1):59–72, Jan 1979.

34. Nasim Nasrabadi, Akram Dehnokhalaji, Pekka Korhonen, and Jyrki Wallenius. Using convex preference cones in multiple criteria decision making and related fields. *Journal of Business Economics 2019 89:6*, 89(6):699–717, Jun 2019.

35. Hutton Barron and Charles P Schmidt. Sensitivity analysis of additive multiattribute value models. *Operations Research*, 36(1):122–127, Feb 1988.

36. Matthew D Ferguson. A scenario-based parametric analysis of stable marriage approaches to the Army officer assignment problem. *Master's thesis, Air Force Institute of Technology*, Mar 2017.

37. John William Hatfield and Paul R. Milgrom. Matching with contracts. *American Economic Review*, Sep 2005.

38. Tayfun Sönmez and Tobias B. Switzer. Matching with (branch-of-choice) contracts at the United States Military Academy. *Econometrica*, 81(2):451–488, Mar 2013.

39. Andrew P. Armacost and James K. Lowe. Decision support for the career field selection process at the US Air Force Academy. *European Journal of Operational Research*, 160(3):839–850, Feb 2005.

40. J. Lepird. Position paper on a next generation air force personnel assignment system. *Technical Report, United States Air Force*, 2016.

41. Lindsey Blanks, Elizabeth Frakes, Koby Hinnant, Swathi Samuel, and John Dulin. An optimization approach to the make-up final exam scheduling problem with unique constraints. *Industrial and Systems Engineering Review*, 8(1):50–58, Mar 2021.

42. Alexander Kline, Darryl Ahner, and Raymond Hill. The weapon-target assignment problem. *Computers & Operations Research*, 105:226–236, May 2019.

43. Der San Chen, Robert G. Batson, and Yu Dang. *Applied integer programming: modeling and solution*. John Wiley and Sons, Sep 2011.

44. Michael L. Bynum, Gabriel A. Hackebeil, William E. Hart, Carl D. Laird, Bethany L. Nicholson, John D. Siirola, Jean-Paul Watson, and David L. Woodruff. *Pyomo–optimization modeling in python*, volume 67. Springer Science & Business Media, third edition, 2021.

45. L. T. Biegler and V. M. Zavala. Large-scale nonlinear programming using IPOPT: an integrating framework for enterprise-wide dynamic optimization. *Computers Chemical Engineering*, 33(3):575–582, Mar 2009.

46. Lei Xu, Maria Skoularidou, Alfredo Cuesta-Infante, and Kalyan Veeramachaneni. Modeling tabular data using conditional gan. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

47. Andreas Wächter and Lorenz T. Biegler. On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Mathematical Programming*, Apr 2005.

48. Ian J Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems*, volume 3, pages 2672–2680, 2014.

49. Edward Choi, Siddharth Biswal, Bradley Malin, and Duke. Generating multi-label discrete patient records using generative adversarial networks. In *Proceedings of Machine Learning Research (PMLR)*, Nov 2017.

50. Akash Srivastava, Lazar Valkov, Chris Russell, Michael U. Gutmann, and Charles Sutton. Veegan: Reducing mode collapse in gans using implicit variational learning. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vish-

wanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

51. Noseong Park, Mahmoud Mohammadi, Kshitij Gorde, Sushil Jajodia, Hongkyu Park, and Youngmin Kim. Data synthesis based on generative adversarial networks. *Proceedings of the VLDB Endowment*, 11(10):1071–1083, Jun 2018.

52. Neha Patki, Roy Wedge, and Kalyan Veeramachaneni. The synthetic data vault. *Proceedings - 3rd IEEE International Conference on Data Science and Advanced Analytics, DSAA 2016*, pages 399–410, Dec 2016.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704–0188

1. REPORT DATE *(DD–MM–YYYY)*	**2. REPORT TYPE**	**3. DATES COVERED** *(From — To)*
24–03–2022	Master's Thesis	Sept 2020 — March 2022

4. TITLE AND SUBTITLE

A New Approach to Career Field Matching for Commissioning Air Force Cadets

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

Laird, D. Griffen, 2d Lt, USAF

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Air Force Institute of Technology
Graduate School of Engineering and Management (AFIT/EN)
2950 Hobson Way
WPAFB OH 45433-7765

8. PERFORMING ORGANIZATION REPORT NUMBER

AFIT-ENS-MS-22-M-143

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)

AFPC/DSYA
Mitchell N. Gillespie, Lt Col
550 C Street West
Randolph AFB TX 78150-4721
DSN 665-2293, COMM 210-565-2293
Email: mitchell.gillespie@us.af.mil

10. SPONSOR/MONITOR'S ACRONYM(S)

AFPC

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

12. DISTRIBUTION / AVAILABILITY STATEMENT

DISTRIBUTION STATEMENT A:
APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

13. SUPPLEMENTARY NOTES

14. ABSTRACT

The current method of assigning graduating cadets from the United States Air Force Academy and Reserve Officers' Training Corps (ROTC) detachments to their career fields uses an integer programming model to maximize "global" Air Force utility, subject to several Air Force-defined constraints. This utility evaluates the positive benefit of assigning a certain cadet to a certain career field. This paper discusses the issues with such a model, as well as presents a new, more refined approach to the problem. Rather than provide a one-size-fits-all formulation of this particular assignment problem, a Value-Focused Thinking (VFT) framework is applied, in conjunction with an optimization model using the framework, to measure overall solution quality for an alternative assignment of cadets to career fields, given numerous weight and value parameters. The power of optimization under a VFT framework is in the ability to capture what decision-makers want, as well as how much they want it. This research shows that the new VFT model outperforms the original model on solution quality by almost 7% when measured using different VFT weight and value parameters.

15. SUBJECT TERMS

optimization, linear programming, non-linear programming, value-focused thinking, machine learning, generative adversarial neural networks, cadets, career fields, AFSCs, simulation, genetic algorithms, piece-wise functions

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON**
a. REPORT	**b. ABSTRACT**	**c. THIS PAGE**			Jesse G. Wales, Lt Col, USAF, AFIT/ENS
U	U	U	UU	143	**19b. TELEPHONE NUMBER** *(include area code)* (937) 255-3636 x4740; jesse.wales@afit.edu