SAND2023-09081J

Holistic Fleet Optimization Incorporating System Design Considerations

Stephen M. Henry, Matthew J. Hoffman
Sandia National Laboratories, PO Box 5800 MS-1188, Albuquerque, NM 87185-1188, USA
smhenry@sandia.gov, mjhoffm@sandia.gov

Lucas A. Waddell
Department of Mathematics, Bucknell University, 1 Dent Drive, Lewisburg, PA 17837, USA
lucas.waddell@bucknell.edu

Frank M. Muldoon
Applied Materials, 5225 Wiley Post Way, Salt Lake City, UT 84116, USA
frank_muldoon@amat.com

Abstract

The methodology described in this paper enables a type of holistic fleet optimization that simultaneously considers the *composition and activity of a fleet* through time as well as the *design of individual systems* within the fleet. Often, real-world system design optimization and fleet-level acquisition optimization are treated separately due to the prohibitive scale and complexity of each problem. This means that fleet-level schedules are typically limited to the inclusion of predefined system configurations and are blind to a rich spectrum of system design alternatives. Similarly, system design optimization often considers a system in isolation from the fleet and is blind to numerous, complex portfolio-level considerations. In reality, these two problems are highly interconnected. To properly address this system-fleet design interdependence, we present a general method for efficiently incorporating multi-objective system design trade-off information into a mixed-integer linear programming (MILP) fleet-level optimization. This work is motivated by the authors' experience with large-scale DOD acquisition portfolios. However, the methodology is general to any application where the fleet-level problem is a MILP and there exists at least one system having a design trade space in which two or more design objectives are parameters in the fleet-level MILP.

Key words: acquisition planning, fleet scheduling, system design trade space, multi-objective, MILP

1 Introduction

Planning the optimal acquisition strategy for a fleet of systems is a popular topic in the Operations Research literature. Such problems are often challenging due to the large number of systems being affected and the variety of complex governing behaviors being modeled. Successfully solving these problems can be hugely impactful; at large scale, even incremental improvements to extant solutions can translate into large cost savings and significant real-world performance improvements. While no fully consistent nomenclature exists throughout the literature, these problems come in a variety of flavors. For example, Fleet Assignment Problems (Abara, 1989; Subramanian, Scheff, Quillianan, Wiper, & Marsten, 1994; Hane et al., 1995; Rushmeier & Kontogiorgis, 1997) are extensively used by the airline industry to match airplanes with scheduled departure times and passenger loads. Portfolio Optimization (Bertsimas, Darnell, & Soucy, 1999; Beaujon, Marin, & McDonald, 2000; Benati & Rizzi, 2007) generally refers to the acquisition of financial instruments or other assets to maximize return on investment, minimize risk, or meet other financial goals. Fleet Optimization, Fleet Scheduling, or Vehicle Fleet Mix Problems (Magnanti, 1981; Roy, 1989; Nulty & Ratliff, 1991; Powell & Perkins, 1997; Dondo & Cerda, 2007; Raa, 2015) typically involve optimal procurement and routing of vehicles and are commonly seen in the transportation and shipping industries. Fleet/Portfolio Modernization Problems (Brown, Clemence, Teufert, & Wood, 1991; Hartman, 2000; Davis et al., 2016) often take a long-term, multi-decade planning approach to optimal fleet or portfolio composition and are less concerned with optimizing day-to-day operations.

While each of the above classes involves unique (often complex) modeling challenges, they can broadly be categorized as determining optimal decisions about *which* systems should be acquired, *how many*, and *when*. Of interest to this study are such problems posed with scalar-valued objectives (e.g., maximize total profit, minimize total expenditures, or maximize aggregate fleet value) under myriad complex constraints. Thus, due to 1) the large number of inherently discrete decisions, 2) the variety of governing constraints and 3) the single-objective function, these problems are typically linearized and solved via algorithms which perform well on large-scale, highly-constrained, mixed-integer linear programs (MILPs).

In contrast to single-objective fleet problems, system-level design trade space optimization is focused on providing a *trade space of discrete, Pareto optimal system configurations* that balance competing design criteria. These diverse problems can also be difficult to solve, but unlike fleet-level optimization problems whose difficulty is driven primarily by problem scale and quantity of constraints, system design trade space optimization difficulties usually arise from the fidelity, complexity, and nonlinearity of the physical processes and design constraints being modeled. It is not uncommon to see computational fluid dynamics (CFD), finite elements (FE), Monte Carlo, and other computationally expensive evaluation algorithms embedded within a system design trade space optimization. Equally diverse are the multi-objective optimization methods commonly employed to solve these problems, which can range from simple Latin hypercube sampling (LHS), to population-based metaheuristics such as genetic algorithms (GA) or particle swarm optimization (PSO), to hierarchical multidisciplinary design optimization frameworks such as collaborative optimization or disciplinary analysis optimization, and even unique, hybrid combinations of multiple techniques.

To illustrate the considerable heterogeneity in system trade space problems, consider a sampling of papers from the aerospace, maritime shipping, and automotive industries. Optimizing supersonic aircraft design, for example, may involve exploration of various wing shape parameters; Obayashi, Sasaki, Takeguchi, & Hirose, 2000 couples GAs and complex airflow code to optimize super/transonic drag and bending moments, while Alonso, LeGresley, & Pereyra, 2009 combines LHS, neural nets, and sonic boom propagation tools to minimize ground noise and structural weight. Sea surface ship optimization might explore multiple hull geometries and bulkhead designs; Tahara, Tohyama, & Katsui, 2006 fuses GAs, successive quadratic programming, and CFD code to maximize maneuver and propulsive performance, while Cui & Turan, 2010 uses hybrid PSO and water-on-deck code to trade cargo capacity and water-inundation survivability. Passenger vehicle safety is often concerned with the thickness of panels, fenders, rails, and other structural components in order to minimize crash deformation distance and reinforcement mass; Lio, Li, Yang, Zhang, & Li, 2008 does this using LHS, surrogate models, and FE simulations while Yildiz & Solanki, 2012 combines PSO and FE. Still other design optimization strategies treat the vehicle as a system-of-systems and employ GAs to explore various combinations of subsystem alternatives: Desai & Williamson, 2009 balancing fuel economy and emissions of hybrid electric vehicles, Henry & Waddell, 2016 trading off load capacity, cost, and autonomy of small robotic transporters. The critical takeaway here is that, unfortunately, no unifying optimization approach nor common modeling tools exist across disparate system-level domains.

Most challenging is when system design choices and acquisition plans are *both* under consideration simultaneously such that an interplay exists between the choice of solutions from the system-level design trade space and their assignment to a fleet-level acquisition strategy. Unfortunately, the many diverse constraints governing fleet operations often obfuscate the effects that system design choices will have on the optimal fleet strategy. For example, an expensive but highly capable system might add great value to a fleet, but a cheaper, less-capable system could actually allow for higher fielding quantities within budget constraints and thus provide a greater aggregate fleet value. Interactions such as this mean that system-level and fleet-level problems are inextricably linked.

Typically, however, these two problems are approached separately due to the significant challenge and complexity of each. In fact, we are not aware of any existing literature discussing their integration. Furthermore, the techniques commonly employed on fleet-level and system-level problems are not amenable to unification. MILP approaches for fleet-level optimization cannot readily incorporate the complex, nonlinear, and even black box evaluation metrics common in system-level trade space optimizations. On the other hand, metaheuristics and other multi-objective approaches cannot readily handle the scale and highly-constrained nature of fleet-level problems. To enable "holistic fleet optimization" that unifies system and fleet-level considerations into one formulation, this paper presents a general framework that efficiently incorporates system-level multi-objective trade space results into fleet-level MILP optimization formulations. The goal is to allow the fleet-level optimization to down-select to a particular Pareto optimal system design that best contributes to the overall objective of the entire fleet.

It is important to note that this approach is distinct from the extensive literature on optimization over a Pareto set of solutions, which focuses on the problem of finding the single Pareto optimal solution that optimizes some given real-valued utility function. The methods described in these works are typically tailored to the structure of the underlying multi-objective problem from which the Pareto set is generated. For instance, authors have considered optimization over the Pareto set of solutions arising from multi-objective linear programs (Yamamoto, 2002; Belkhiri, Chergui, & Ouaïl, 2021), multi-objective integer linear programs (Jorge, 2009; Boland, Charkhgard, & Savelsbergh, 2017; Cherfaoui & Moulaï, 2021), multi-objective integer linear fractional programs (Moulaï & Mekhilef, 2021; Chaiblaine & Moulaï, 2022), and multi-objective nonlinear programs (Benson, 1984; An, Tao, & Muu, 1996; Horst & Thoai, 1997; Thoai, 2000; Liu & Ehrgott, 2018). In contrast, the holistic fleet optimization problem requires the selection of a Pareto optimal system design that can best be *incorporated* into a complex fleet-level modernization plan. That is, the choice of system design is just one of many interconnected factors affecting both the objective function *and* constraints of the fleet-level problem. Furthermore, one holistic fleet optimization problem could potentially involve many different system types, each with its own corresponding multi-objective system design problem having a unique (potentially even non-closed-form) structure.

To address this challenge, the proposed methodology utilizes the *results* of the multi-objective trade space optimization, and is therefore agnostic to the system-level optimization technique itself and does not require that the machinery of fleet-level and system-level optimization be integrated. Instead, system-level trade space results are pre-computed using whichever technique is most appropriate for the given study. Once the Pareto optimal system-level solutions are obtained, the convex hull of the performance space (i.e., the objective space) of these solutions is incorporated into the fleet-level MILP via bilinear reformulations. The convex hull representation is used as an efficient approximation for the discrete system trade space because a more obvious approach of allowing the MILP to select directly from hundreds to thousands of discrete system design alternatives would be computationally prohibitive – especially if done for *multiple* system types, each with its own system-level trade space. But because the convex hull is an approximation of a discrete system-level trade space, care must be taken to ensure the fleet-level optimization realizes values within the convex hull that correspond to an actual Pareto solution of the system design problem.

The remainder of this paper is organized as follows. Section 2 provides real-world motivation for holistic fleet optimization arising from large-scale fleet modernization and combat system trade studies performed by the United States Army. Section 3 describes the methodology for capturing the convex hull of a discrete Pareto optimal frontier, reformulating the bilinear terms that arise when the convex hull is incorporated within a fleet-level MILP, and performing disjunctive iterations to ensure convex hull vertices are selected. Section 4 provides a full example that illustrates our methodology on a simplified fleet modernization problem with two system design trade spaces. Section 5 uses real-world, large-scale fleet modernization and system trade spaces from our motivating work to demonstrate the computational efficiency of our method compared with direct representation of every system design alternative within the MILP. Finally, Section 6 briefly summarizes this work and considers avenues for future study.

2 Real-World Motivation

This paper is motivated by prior work in acquisition, modernization, and design trade space decision support for Program Executive Offices (PEOs) within the United States Army. As an example, PEO Aviation is responsible for a fleet of nearly 5,000 AH-64 Apache, UH-60 Blackhawk, and CH-47 Chinook helicopters as well as a variety of fixed-wing and unmanned aircraft (GAO, 2020). Similarly, PEO Ground Combat Systems (GCS) is responsible for a fleet of more than 20,000 specialized combat vehicles including the M1 Abrams Tank, the M109A6 Self-Propelled Howitzer, and the Bradley and Stryker Families of Vehicles (Davis et al., 2016). PEO Combat Support and Combat Service Support (CS&CSS) manages a portfolio of over 200,000 vehicles including Humvees, HEMTTs, HETs, unmanned ground vehicles, bridge-building vehicles, and even watercraft transport systems (ASA-ALT, 2021). As a key acquisition authority, PEOs are tasked with both 1) developing next-generation systems and 2) implementing a long-term fleet modernization plan for all current and future systems within their portfolio. At the fleet level, challenges emerge from a complex confluence of political, budgetary, production, inventory, and industrial base considerations. At the system level, each vehicle can be enormously complex with dozens of integrated subsystems that combine to meet hundreds of (often competing) design requirements. To address these significant PEO challenges, two existing capabilities have been developed that separately address the fleet-level and system-level problems – reviewed briefly in the remainder of this section.

The Capability Portfolio Analysis Tool (CPAT) (Davis et al., 2016) is a MILP-based capability that prioritizes acquisition and modernization plans to create an optimal fleet composition schedule spanning multiple decades (see Figure 1). CPAT solution schedules must satisfy a wide variety of constraints enforcing behaviors such as maximum and minimum production rates, ramp-up and consistency of production lines, coordinated schedules across disparate production lines, initial fielding dates, retirement or age-out schedules, deployment cycles, upfront research, development, testing and evaluation costs, fixed production costs and per-unit acquisition costs, multiple allocation budgets, earmarks allocated to specific programs, and many other business rules. In total, the CPAT MILP formulation has 98 classes of constraints that enforce 45 unique scheduling behaviors (see electronic companion to Davis et al., 2016).

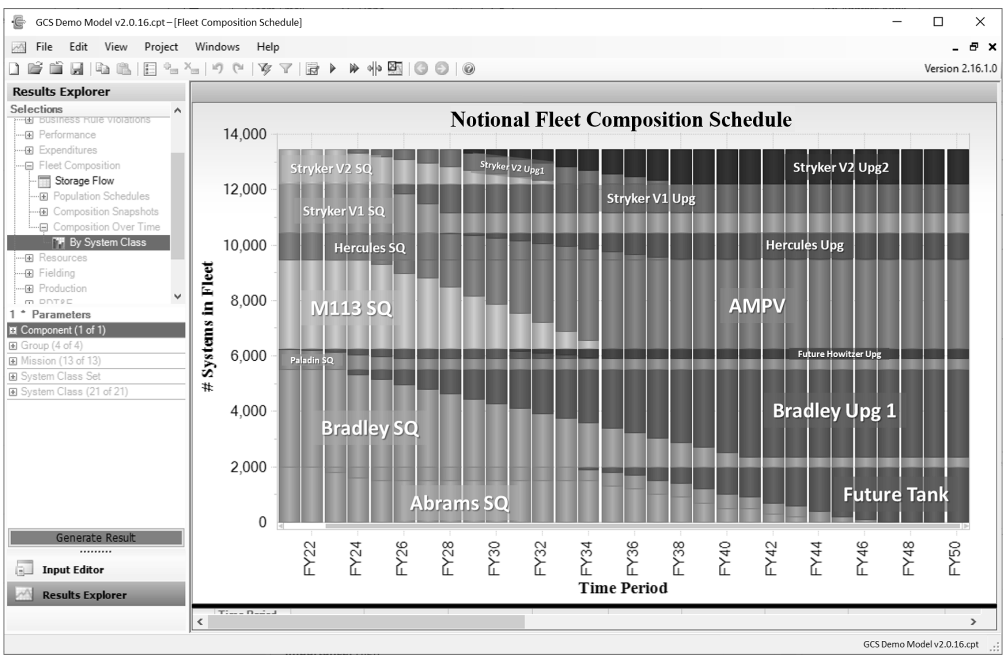


Figure 1: This CPAT screenshot depicts a notional ground combat fleet with optimized composition changes over a 30-year time horizon where current-day "Status Quo" (SQ) systems are replaced with new or Upgraded (Upg) variants that maximize cumulative fleet value. All systems in CPAT are "point designs" with fixed cost and value parameters.

Along with these constraints, CPAT also includes a value model that constructs a single, unique score (later called v_s for each system s) for all current and future systems in the fleet. These system scores define the coefficients of the formulation's cumulative value objective function. The CPAT value model draws from dozens of individual system attributes in categories such as survivability, lethality, mobility, and any other consideration relevant to the PEO. Each performance attribute is normalized using predefined walk-away, threshold, and objective values (corresponding to 0, 0.7, and 1, respectively). These normalized system attribute scores are then additively aggregated together using importance weights for each attribute – these weights being elicited from subject matter experts (SMEs) via the Swing-Weight methodology of Parnell & Trainor, 2009.

While it may seem oversimplified to represent the complexities of a system's operational value as a singular score v_s, and perhaps even more simplistic to model group-level value as linear addition of individual system v_s, PEO leadership preferred this compromise for a number of reasons. First, neither CPAT nor the PEO is responsible for tactical mission execution where the number of systems needed and most relevant characteristics can vary wildly from scenario to scenario. Rather, the responsibility of the PEO (and thus of CPAT) is to procure the most overall value for the largest number of systems. The construction of v_s, with its consideration of many different value categories weighted by SME importance, was deemed acceptable for this purpose. Second, PEO portfolios contain a large number of identical system types within a relatively static, self-supporting force structure. Hence, questions of value focus less on "What is the value of 1 tank vs. 50 tanks vs. 500 tanks?" (where linearity assumptions are much more dubious) and more so around "What is the value of upgrading 1000 existing tanks vs. purchasing 1000 new tanks?" (where linearity is more accurate due to the averaging effect of the large quantities at play). Lastly and perhaps most importantly, this approach was the considered most explainable to higher leadership when the PEO was required to defend acquisition decisions.

Another advantage of linear value assumptions is the computational tractability this affords to PEO problem scales. As one example of problem size, a typical PEO GCS model spanning several decades requires roughly 70,000 constraints and 10,000 integer variables to capture all modernization possibilities and complex business rules. Models for other PEOs, such as the aforementioned CS&CSS, can be significantly larger. Even with a well-tuned model, optimization run times typically require several hours or days to reach an acceptable gap tolerance (often required to be very stringent) using the latest CPLEX or Gurobi solvers.

Switching to the topic of system-level trade spaces, the Whole System Trades Analysis Tool (WSTAT) (Edwards et al., 2015) uses a multi-objective genetic algorithm (GA) to explore the trade space of a future system design – mapping out the best possible designs across a range of different stakeholder value dimensions. WSTAT has been applied to a very broad range of systems across DOD portfolios, including ground vehicles, robotic systems, watercraft, forward operating bases, intercontinental ballistic missiles, hypersonic boost glide vehicles, and nuclear warheads. Each type of system has a unique 1) subsystem decomposition, 2) selection of parts that can be chosen for each subsystem, and 3) set of competing requirements and evaluation metrics. The number of possible unique solutions in a system trade space typically ranges from 10^{25} to 10^{100} designs. The WSTAT GA examines millions of subsystem combinations over successive generations using mutation, crossover, and other specially-designed operators to converge on a set of hundreds or thousands of Pareto optimal designs that best balance objectives such as value, cost, growth potential, and schedule risk. (As in CPAT, value metrics can include survivability, lethality, mobility, and any other stakeholder-relevant considerations.) Evaluating system designs based on the choice of subsystem parts can involve calculations including simple equations, lookup tables, or even nonlinear iterative subroutines with no closed form. Optimization times are usually several hours to ensure the evolution of the GA population has converged to an adequate, repeatable representation of the true Pareto frontier (see Figure 2).

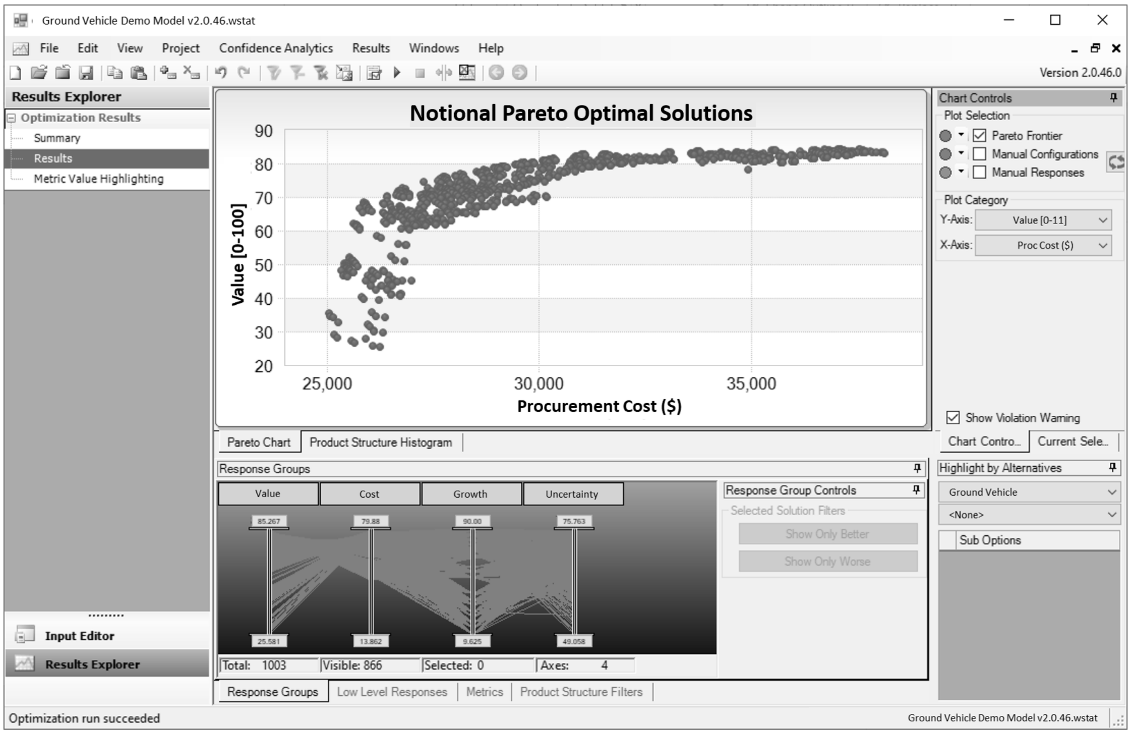


Figure 2: This WSTAT screenshot depicts a Pareto optimal trade space for a notional ground vehicle. Each point on the scatter plot represents an optimal system design created from a unique collection of subsystem parts. In this example, designs are optimized in the four-dimensional performance space of value, procurement cost, growth potential, and schedule uncertainty. Efficiently representing design performance spaces within an overarching fleet optimization is the primary goal of this research.

Even with dedicated tools such as CPAT and WSTAT, these fleet and system-level problems can be quite difficult on their own – yet ideally should be integrated when system and fleet applications overlap. However, this integration is challenging since these fleet formulations assume a *fixed* cost and value for every system type, while a system-level trade space optimization provides *thousands* of Pareto optimal alternatives for that system. Simply stated, the goal of this paper is to enable fleet-level optimization to utilize the rich set of optimal system design alternatives to make choices that simultaneously consider both the system *and* fleet perspectives. For example, if a system-level tradeoff exists between cost and value, it may be preferable from a fleet perspective to choose a lower-value but cheaper system in order to procure more of those systems within a fixed budget. This is but one example of how a more thorough unification of system and fleet-level optimization can provide better decision support to organizations responsible for both the operations of their fleet and the designs of the systems therein.

3 Methodology

This section outlines a methodology for incorporating Pareto optimal design alternatives from one or more system-level trade spaces into a fleet-level MILP formulation in order to capture the full spectrum of system design possibilities without compromising the tractability of the fleet modernization optimization. While the PEO portfolios mentioned in Section 2 are a classic application space, this methodology is general to any portfolio ecosystem having the following key properties:

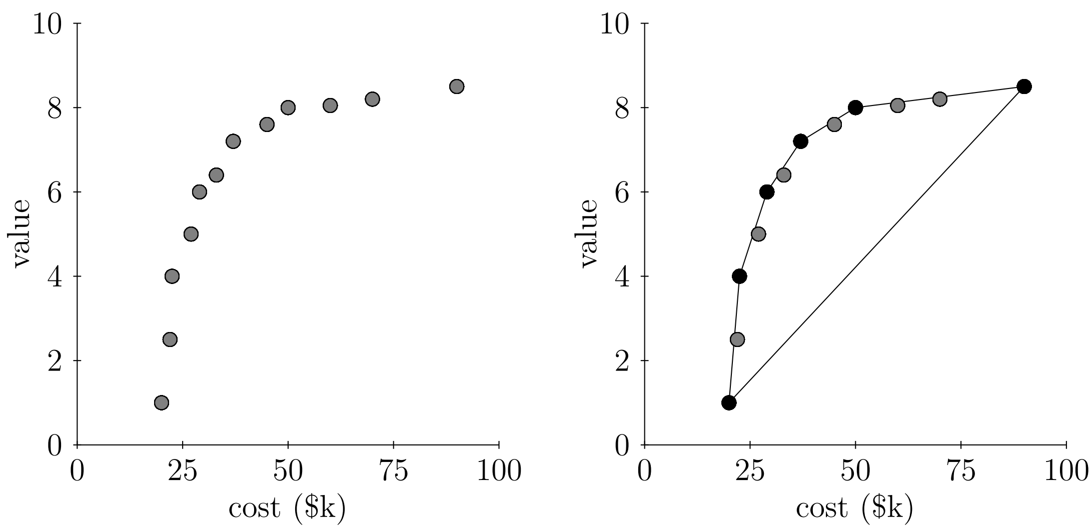
- the portfolio scheduling problem is solved via a MILP formulation,
- the design of at least one system in the portfolio is not yet finalized and multiple discrete alternatives are available, and
- at least two dimensions of the objective space of the system-level design problem are equivalent to system parameters in the fleet-level formulation.

Since 1) real-world system design trade space problems often produce thousands of Pareto optimal solutions and 2) large-scale fleet-level MILPs are often near the boundary of tractability, it is generally impossible to individually incorporate all Pareto optimal system designs as separate alternative systems within the fleet problem. Instead, we introduce the new concept of "adaptive" systems whose performance parameters (such as cost and value) vary within bounds determined by the optimized system-level trade space. The fleet-level MILP then chooses a particular instantiation of the adaptive systems' parameters to best meet the optimization goals of the entire fleet – the essence of the holistic fleet optimization problem. By design, the fleet-level choice of adaptive systems' parameters is a *post hoc* decision independent of (but obviously informed by) the system-level design problem(s); fleet-level choices do not change the optimal system-level trade space. Thus, the holistic fleet-level problem is still a MILP, rather than a more complex multiobjective bilevel optimization problem (as described, for example, in Sinha et al., 2018).

Our methodology is as follows. First, we acquire the set of Pareto optimal system designs using whichever technique is most appropriate for the given study. For the U.S. Army applications discussed in Sections 2 and 5, we use the WSTAT GA to approximate the Pareto set for the non-linear, multi-objective system design problem. Then, for each adaptive system, we acquire the convex hull representation of the performance space of the Pareto optimal solutions - providing a bounding polytope within which the design parameters of each adaptive system can vary. Next, these convex hull constraints are appended to the fleet formulation and new continuous variables are introduced to represent the adaptive systems' design parameters – allowing the fleet-level MILP to optimally choose each adaptive system's design. With the creation of these new adaptive parameter variables, nonlinearities are introduced into the fleet formulation in the form of bilinear products of existing discrete fleet decision variables and continuous adaptive parameter variables. These bilinear terms are then linearized via the introduction of new auxiliary variables and constraints to enable solution by standard MILP solvers. Finally, if the adaptive parameter variables chosen by the fleet optimization do not match one of the original configurations from the system-level trade space, we iteratively disjunct the convex hull representation and re-optimize until each adaptive system's parameters correspond to a Pareto optimal system configuration. Key steps are explained in more detail in the following subsections.

3.1 Acquiring the Convex Hull of a Pareto Set

The convex hull of a set of discrete points S in $\mathbb{R}^d$, denoted $\text{conv}(S)$, is the smallest convex set that contains all points in S. Stated another way, the convex hull can be represented as the set of linear constraints that most tightly envelops S. As an example, the left side of Figure 3 shows a set of system design configurations that optimally balance two performance objectives, cost and value; on the right, the convex hull constraints form the tightest possible linear bounds around the performance space of the design alternatives. The convex hull allows approximation of a set of discrete configurations from multi-objective optimization with a set of *simple linear constraints*.



Pareto optimal solutions that best balance cost & value for an adaptive system

The convex hull of the Pareto solutions yields a linear envelope of the discrete points

Figure 3: This figure shows an example Pareto optimal performance space of cost vs. value (left) and its convex hull (right). Pareto solutions that form vertices of the convex hull are filled in black.

By representing the performance parameters of the adaptive systems (e.g., cost and value in Figure 3) as new continuous decision variables bounded within the convex hull constraints, the fleet optimization can determine the best realization of these variables that is realistic for the system and best satisfies the optimization goals of the entire fleet. In Figure 3 for example, the 12 discrete system design alternatives on the left can be approximated on the right by 6 convex hull constraints and a continuous cost and value variable. Various techniques (Barber, David, & Huhdanpaa, 1996; Chazelle, 1993; Graham, 1972) are readily available to generate convex hull constraints given a set of discrete points. Section 5 leverages a Python implementation of the Qhull library (Barber et al., 1996) in order to acquire convex hulls for the computational study. Although this convex hull computation would become computationally intractable for large numbers of adaptive parameter variables (i.e., a high-dimensional system performance space), most real world applications limit the number of competing objectives within a system design trade space optimization to less than six.

While appending new convex hull constraints to an existing fleet-level MILP formulation is trivial, a complication arises in dealing with the new adaptive parameter variables (e.g., cost and value in Figure 3). In defining the convex hull, what would previously have been *fixed data* about a discrete set of alternatives, is now a set of *continuous variables* that will likely be multiplied by other existing discrete variables in the fleet formulation (e.g., total cost might be calculated as the number of adaptive systems procured times the adaptive system cost). This gives rise to nonlinear terms, and the next subsection outlines the use of bilinear reformulations to model such products of continuous and discrete variables.

3.2 Linearizing Bilinear Terms

Bilinear terms (e.g., a continuous variable times a binary or integer variable) are a natural result of introducing convex hulls of the performance spaces of system design alternatives into a fleet-level MILP. Expanding on the example of Figure 3, if v_s gives the value of system s and x_{st} determines how many systems s are in the fleet at time t, then $\sum_{s,t} v_s x_{st}$ gives the cumulative fleet value – a natural choice for a fleet-level objective function. When v_s is a fixed parameter, as in typical fleet MILP formulations, this cumulative fleet value expression is linear. But when v_s is a continuous variable, reformulation is needed to accommodate the bilinear term. Many techniques to linearize such bilinear terms are available in the optimization literature (Bergamini, Grossmann, Scenna, & Aguirre, 2008; Gupte, Ahmed, Cheon, & Dey, 2013; Liberty & Constantinos, 2006), and the technique of Gupte et al., 2013, detailed below, is employed for its computational efficiency and minimal creation of new auxiliary variables and constraints.

Consider the general bilinear term xy, which is the product of a bounded non-negative integer variable $x \leq U$ and a bounded non-negative continuous variable $y \leq V$. Linearization requires replacement of xy with a single continuous variable w that always realizes the desired product. This is achieved by enforcing the linear constraints found in

$$\mathcal{L}_{xy} = \left\{ (w, \mathbf{e}^x, \mathbf{e}^y) \in \mathbb{R}_+ \times \{0,1\}^\ell \times \mathbb{R}_+^\ell : \quad w = \sum_{i=1}^{\ell} 2^{i-1} e_i^y, \quad x = \sum_{i=1}^{\ell} 2^{i-1} e_i^x, \quad y - V(1 - e_i^x) \leq e_i^y \leq \begin{matrix} V e_i^x \\ y \end{matrix} \quad \forall i \in \{1, \ldots, \ell\} \right\} \tag{1}$$

where $\ell = \lceil \log_2 U \rceil$ and where the final inequality denotes that e_i^y is less than or equal to both $V e_i^x$ and y. In this formulation, the binary variables e_i^x are used to find a binary expansion of the integer variable x while the continuous variables e_i^y set the correct value of the continuous variable w. As is demonstrated in Gupte et al., 2013, the variable w as represented in $\mathcal{L}_{xy}$ will always equal the product xy for $0 \leq x \leq U$ and $0 \leq y \leq V$. It is in this manner that we linearly represent terms such as $v_s x_{st}$ within the fleet optimization problem.

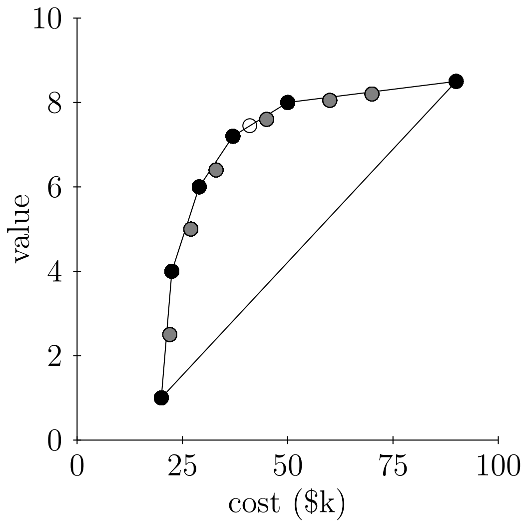
3.3 Iterative Process for Non-Vertex Cases

Although the performance space convex hull provides the tightest possible convex envelope constraining an adaptive system's parameters, it is possible that the fleet-level optimization may choose an instantiation that does not correspond to one of the discrete solutions from the original system trade space (i.e., a solution that is not a vertex of the convex hull). This possibility arises because fleet-level constraints can cut into the system-level trade space. For instance, in the illustrative example of Section 4, fleet-level budgetary restrictions cause the optimization to choose a system design that does not correspond to a vertex of the convex hull of the system's cost-value trade space. In fact, depending on the structure of the fleet-level constraints, the optimal adaptive system parameters may not even fall on the *boundary* of the convex hull. For example, in addition to budget constraints that would cut through the cost axis, some applications might involve requirements from treaties or other international agreements that put limits on the total value or capability of a fleet.

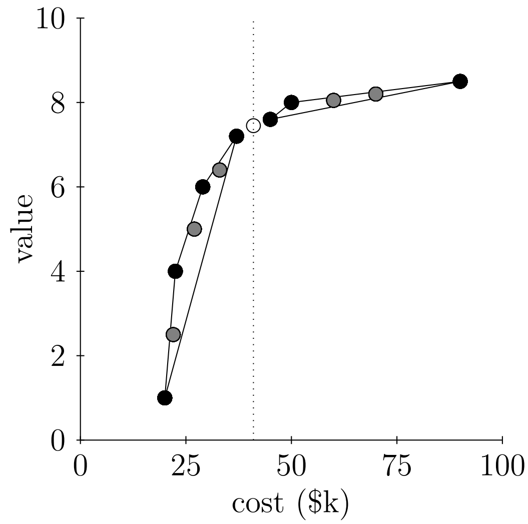
In such cases, the fleet-level plan may not be valid as the characteristics of the adaptive system do not represent one of the realizable Pareto optimal designs. Empirical evidence from our computational experiments suggests this happens relatively rarely. Nevertheless, this possibility must be addressed. In some analyses, it may be acceptable to simply choose the Pareto optimal system configuration whose parameters most closely match the adaptive parameters selected by the fleet-level optimization (especially if within some pre-defined acceptable neighborhood). Although this technique is computationally trivial, there is no guarantee that it will provide an optimal fleet-level solution.

Another possibility is to take advantage of the valuable new information that the fleet desires a system that is not currently available from the system-level problem. With this new information in hand, we could revisit the system design optimization with a focus on generating additional solutions whose parameters (closely) match the chosen fleet-level adaptive parameters – both to determine engineering feasibility of those fleet-level choices and generate attractive new system options. However, the non-vertex issue may still persist with the introduction of these new options.

For the purposes of this paper, we assume that an exact solution is ultimately required. We therefore strictly enforce that the fleet-level optimization must choose all adaptive parameters to correspond exactly with an existing system-level Pareto optimal design. To accomplish this, we introduce a simple iterative procedure to employ if any adaptive parameters do not fall on a convex hull vertex. Suppose, as shown on the left of Figure 4, that the fleet optimization chooses a non-vertex point. Note that any hyperplane passing through this point can be used to partition the Pareto solutions into two subsets with disjoint convex hulls, as shown on the right side of the figure. Using this insight, we can re-solve the fleet-level optimization, this time enforcing that the adaptive system parameters must fall in the *union* of the partition convex hulls. If the optimization still does not choose a vertex we can iteratively disjunct and re-solve until a corner point is chosen. This procedure will certainly converge – at worst, once each partition contains exactly one solution – and computational experiments suggest that multiple iterations are rarely needed for realistic problems.



The convex hull of a system trade space with a non-vertex, fleet-level optimal denoted by ∘.



A hyperplane (dotted line) passing through the non-vertex optimal defines disjoint convex hulls.

Figure 4: When a non-vertex solution is chosen from the convex hull (left), the first iteration of the procedure re-frames the trade space as the union of two convex hulls (right) of the Pareto points separated by a hyperplane through the chosen non-vertex solution.

To model the union of disjoint convex hulls within a MILP, we employ the classic disjunctive approach (Balas, 1988). In particular, let $\mathbb{S}$ be a partition of the set S of discrete points in $\mathbb{R}^d_+$ where $\mathbb{S} = \{S_1, \ldots, S_n\}$ and the convex hull of each subset S_i is defined by $\text{conv}(S_i) = \{\mathbf{x} \in \mathbb{R}^d : A_i\mathbf{x} \leq b_i\}$ as discussed in Section 3.1. Then using the Balas, 1988 method, the union of each subset's convex hull, $\cup_{i=1}^n \text{conv}(S_i)$, can be formulated linearly as

$$\text{unionconv}(\mathbb{S}) = \Bigg\{ (\boldsymbol{x}, \boldsymbol{X}, \boldsymbol{y}) \in \mathbb{R}^d_+ \times \mathbb{R}^{d\times n}_+ \times \{0,1\}^n : \\ \sum_{i=1}^n y_i = 1, \\ \sum_{i=1}^n \boldsymbol{X}_{*,i} = \boldsymbol{x}, \\ A_i \boldsymbol{X}_{*,i} \leq b_i y_i \qquad \forall i \in \{1, \ldots, n\}, \\ \mathbf{0} \leq \boldsymbol{X}_{*,i} \leq \boldsymbol{M} y_i \qquad \forall i \in \{1, \ldots, n\} \Bigg\} \tag{2}$$

where $\boldsymbol{X}_{*,i}$ is the i^{th} column of $\boldsymbol{X}$ and $\boldsymbol{M} \in \mathbb{R}^d$ is a vector of upper bounds on the values in S. In this formulation, the binary y_i indicator variables select exactly one of the subset convex hulls to be "activated." Each column of continuous variables, $\boldsymbol{X}_{*,i}$, either 1) falls within the selected convex hull constraints for that i which is activated or 2) equals $\mathbf{0}$ for any other i. Finally, the continuous $\boldsymbol{x}$ variables always match that column $\boldsymbol{X}_{*,i}$ which falls within the activated convex hull.

3.4 Holistic Fleet Optimization Algorithm

Our approach to holistic fleet optimization is summarized in the algorithm presented in Figure 5. The algorithm begins with a MILP formulation that plans the optimal composition and activity of a fleet of systems, as well as multi-objective optimization problems for the design of one or more of the systems (called adaptive systems) within the fleet. The optimization performance dimensions, such as cost and value, of the system design problems are referred to as adaptive parameters. The output of the algorithm is an optimized fleet-level plan that also prescribes which design alternative should be used for each adaptive system. Note that the algorithm is agnostic to the method by which a Pareto optimal trade space of designs (or an approximation thereof) is obtained for each adaptive system. In the next section, we provide an illustrative example that demonstrates the use of this algorithm.

Holistic Fleet Optimization Algorithm

---

begin
  Formulate the fleet-level optimization problem as a MILP
  for each adaptive system
    Generate the Pareto optimal trade space of system designs
    Acquire convex hull constraints of the Pareto optimal performance space (Section 3.1)
  end-for
  Update the MILP to represent the performance trade space of each adaptive system:
    1) Substitute a continuous variable for each adaptive system parameter
    2) Linearize any resulting bilinear terms using Equation 1 (Section 3.2)
    3) Append convex hull constraints for each adaptive system
  Optimize the updated fleet-level MILP
  while adaptive system parameter variables do not match an original discrete solution
    Disjunct the convex hull of that adaptive system with Equation 2 (Section 3.3)
    Re-optimize the updated fleet-level MILP
  end-while
end

Figure 5: This figure presents our holistic fleet optimization algorithm, which summarizes the methodology of Section 3.

4 Illustrative Example

To illustrate the ideas described in Section 3 and summarized in the algorithm of Figure 5, we present a simple, example fleet modernization problem (not approaching the true complexity of a real-world application such as Davis et al., 2016) wherein each system has two key performance parameters of cost and value. Note that in general, our approach works with any number of adaptive system parameters, but for clarity we will illustrate our approach using two-dimensional system trade spaces.

Consider the following simple fleet modernization problem over a collection of time periods T and a collection of system types S. First, we will assume that each system $s \in S$ has a *fixed* value v_s and purchase cost c_s. Fleet parameter b_t gives the budget at time t, and r_t represents the required number of systems in the fleet at time t. The integer variable x_{st} represents the number of systems of type s assigned to the fleet in time t while the integer variable p_{st} tells how many systems of type s are purchased at time t. Production limits U^p_{st} and assignment limits U^x_{st} put an upper bound on the number of systems of type s in time t that can be purchased or placed in the fleet, respectively. The goal of maximizing the cumulative fleet value is accomplished by this fleet modernization problem (FMP).

$$
\begin{aligned}
\text{FMP:} \quad & \text{Max} && \sum_{\substack{s\in S\\ t\in T}} v_s x_{st} && \\
& \text{s.t.} && \sum_{s\in S} x_{st} = r_t && \forall t \in T \\
& && \sum_{s\in S} c_s p_{st} \leq b_t && \forall t \in T \\
& && x_{st} \leq \sum_{t^* \leq t} p_{st^*} && \forall s \in S, t \in T \\
& && x_{st} \leq U^x_{st} && \forall s \in S, t \in T \\
& && p_{st} \leq U^p_{st} && \forall s \in S, t \in T \\
& && x_{st}, p_{st} \in \mathbb{Z}_+ && \forall s \in S, t \in T
\end{aligned}
$$

For a small example over five years ($T = \{1, 2, 3, 4, 5\}$) with two system types ($S = \{1, 2\}$), suppose we have the following parameters:

Year	1	2	3	4	5
r_t	5	10	10	20	20
b_t	\$120k	\$120k	\$120k	\$120k	\$120k

System	1	2
c_s	\$21.5k	\$25k
v_s	0.5	2.5

Furthermore, each system type has a production limit of four systems per year and a loose assignment limit of 20 (i.e., $U^p_{st} = 4$ and $U^x_{st} = 20$ for all $s \in S$ and $t \in T$). The optimal fleet schedule of this simple problem is outlined in Figure 6 having an optimal cumulative fleet value of 124.5.

Now suppose that instead of having a *fixed* value and cost, each of the two system types is adaptive with a simple cost-value *trade space*. The Pareto optimal configurations for these systems, along with the bounding convex hull constraints (see section 3.1), are shown in Figure 7. The cost-value Pareto optimal design parameters of the systems are listed in the tables below, where $P_{s,i}$ is the i^{th} Pareto solution for system type s.

System 1	$P_{1,1}$	$P_{1,2}$	$P_{1,3}$
c_1	\$21.4k	\$21.5k	\$24k
v_1	0.2	0.5	1.95

System 2	$P_{2,1}$	$P_{2,2}$	$P_{2,3}$	$P_{2,4}$
c_2	\$21.75k	\$22k	\$23k	\$25k
v_2	0.4	1.2	1.8	2.5

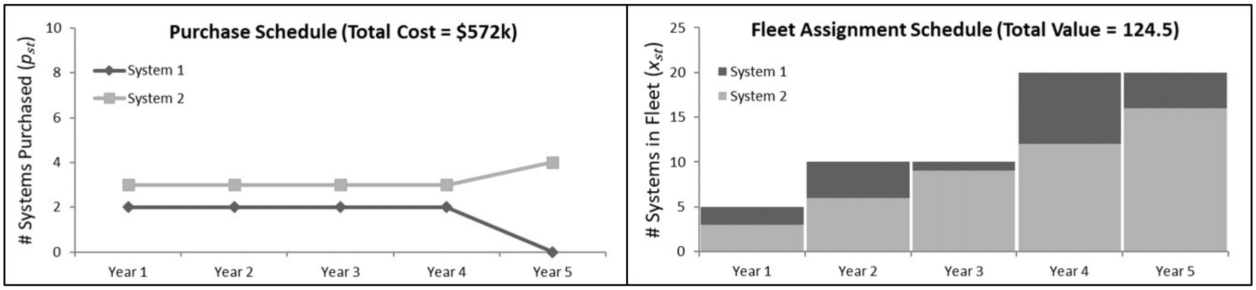


Figure 6: For FMP, this schedule is optimal given yearly budgets and production limits.

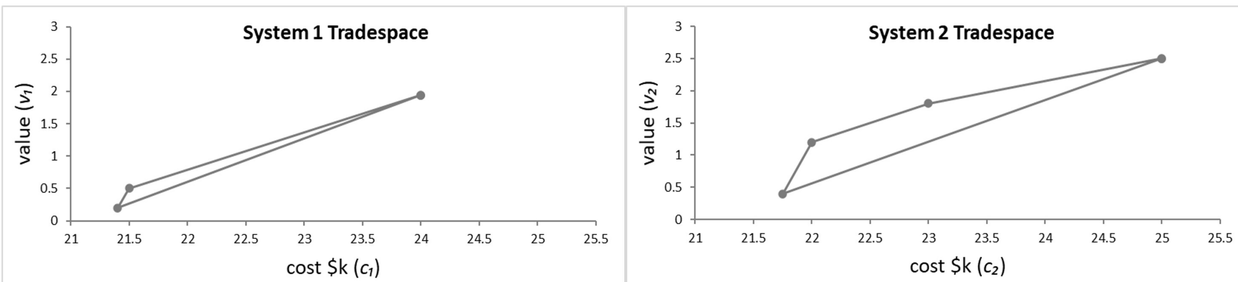


Figure 7: This figure shows the Pareto optimal cost-value trade spaces for System 1 and 2 designs, along with the bounding convex hull constraints.

Note that for FMP, the cost and value parameters were fixed at $P_{1,2}$ and $P_{2,4}$. We will show by the end of this section that such a pre-selection of system parameters does not result in the best possible fleet plan.

To capture these adaptive system trade spaces within the fleet modernization problem, we require only a minor modification to FMP – accomplished by defining c_s and v_s as new continuous variables and enforcing that they satisfy their associated convex hull constraints. Here, conv(s) represents the convex hull of the discrete set of Pareto optimal configurations for system type s. These modifications are captured in the *holistic* fleet modernization problem (HFMP).

$$
\begin{array}{llll}
\text{HFMP:} & \text{Max} & \displaystyle\sum_{\substack{s\in S\\ t\in T}} v_s x_{st} & \\
& \text{s.t.} & \displaystyle\sum_{s\in S} x_{st} = r_t & \forall t \in T \\
& & \displaystyle\sum_{s\in S} c_s p_{st} \leq b_t & \forall t \in T \\
& & x_{st} \leq \displaystyle\sum_{t^*\leq t} p_{st^*} & \forall s \in S, t \in T \\
& & x_{st} \leq U^x_{st} & \forall s \in S, t \in T \\
& & p_{st} \leq U^p_{st} & \forall s \in S, t \in T \\
& & (c_s, v_s) \in \text{conv}(s) & \forall s \in S \\
& & c_s, v_s \in \mathbb{R}_+ & \forall s \in S \\
& & x_{st}, p_{st} \in \mathbb{Z}_+ & \forall s \in S, t \in T
\end{array}
$$

HFMP is nonlinear due to the bilinear product terms $v_s x_{st}$ and $c_s p_{st}$. To linearize each of these terms, we employ the reformulation $\mathcal{L}$ provided in Equation 1 of Section 3.2. In doing so, we replace $v_s x_{st}$ and $c_s p_{st}$ with continuous variables w^v_{st} and w^c_{st}, respectively. This gives the equivalent *linear* holistic fleet modernization problem (LHFMP).

$$
\begin{array}{llll}
\text{LHFMP:} & \text{Max} & \displaystyle\sum_{\substack{s\in S\\ t\in T}} w^v_{st} & \\
& \text{s.t.} & \displaystyle\sum_{s\in S} x_{st} = r_t & \forall t \in T \\
& & \displaystyle\sum_{s\in S} w^c_{st} \leq b_t & \forall t \in T \\
& & x_{st} \leq \displaystyle\sum_{t^*\leq t} p_{st^*} & \forall s \in S, t \in T \\
& & x_{st} \leq U^x_{st} & \forall s \in S, t \in T \\
& & p_{st} \leq U^p_{st} & \forall s \in S, t \in T \\
& & (w^v_{st}, \mathbf{e}^{v_s}, \mathbf{e}^{x_{st}}) \in \mathcal{L}_{v_s, x_{st}} & \forall s \in S, t \in T \\
& & (w^c_{st}, \mathbf{e}^{c_s}, \mathbf{e}^{p_{st}}) \in \mathcal{L}_{c_s, p_{st}} & \forall s \in S, t \in T \\
& & (c_s, v_s) \in \text{conv}(s) & \forall s \in S \\
& & c_s, v_s \in \mathbb{R}_+ & \forall s \in S \\
& & x_{st}, p_{st} \in \mathbb{Z}_+ & \forall s \in S, t \in T
\end{array}
$$

Solving LHFMP using the system trade spaces of Figure 7 results in the choice of adaptive system design parameters and new optimal fleet modernization schedule shown in Figure 8. Here the optimization chooses to keep $P_{1,2}$, but chooses a non-vertex solution for System 2 (near $P_{2,4}$, but slightly cheaper in order to purchase 4 per year). This new schedule achieves a cumulative value of 140.8875, which is more than 13% higher than the 124.5 value of the original FMP solution, but since System 2 was not chosen at a vertex, this plan is not guaranteed to be realizable.

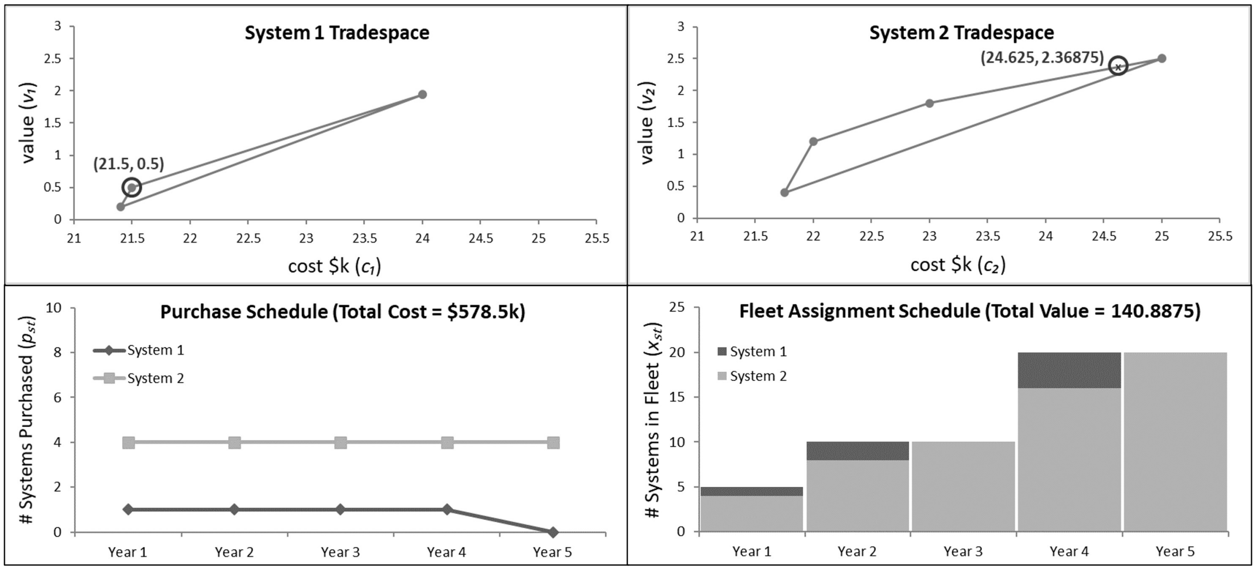


Figure 8: For LHFMP, where (c_s, v_s) are continuous variables bounded by the convex hulls, the optimization chooses a non-vertex solution for System 2.

Since we assume that rounding a non-vertex solution to its nearest Pareto neighbor (or trying to find a new system configuration near the non-vertex solution) is not sufficient for our methodology, at this stage we must employ the iterative disjunctive procedure described in Section 3.3, and partition the trade space of System 2. Note that for this small example, testing the twelve combinations of Pareto optimal System 1 and System 2 designs would be computationally trivial; however, this enumerative method would not scale well to hundreds or thousands of Pareto solutions.

Let $\mathbb{S}_{s,k}$ denote a partitioning of Pareto solutions for system type s at the k^{th} iteration of our procedure with $n_{s,k}$ denoting the number of subsets in $\mathbb{S}_{s,k}$ (so $n_{s,0} = 1$ for all s). Recall that any hyperplane passing through the non-vertex solution for System 2 will create a valid partitioning of the Pareto solutions. For simplicity, we employ a vertical plane to achieve the partitioning $\mathbb{S}_{2,1} = \{\{P_{2,1}, P_{2,2}, P_{2,3},\}, \{P_{2,4}\}\}$. The convex hull for System 1 does not need to be partitioned since the adaptive parameters were chosen at the vertex $P_{1,2}$ – hence $\mathbb{S}_{1,1} = \{\{P_{1,1}, P_{1,2}, P_{1,3},\}\}$. This means that $n_{1,1} = 1$ and $n_{2,1} = 2$. Now we introduce our final, general formulation for the *disjunctive* holistic fleet modernization problem (DHFMP$_k$), where k represents the current iteration of the disjunctive procedure so that DHFMP$_0$ is equivalent to LHFMP.

$$
\begin{aligned}
\text{DHFMP}_k: \quad & \text{Max} \quad \sum_{\substack{s\in S\\ t\in T}} w^v_{st} \\
& \text{s.t.} \quad \sum_{s\in S} x_{st} = r_t && \forall t \in T \\
& \sum_{s\in S} w^c_{st} \leq b_t && \forall t \in T \\
& x_{st} \leq \sum_{t^*\leq t} p_{st^*} && \forall s \in S, t \in T \\
& x_{st} \leq U^x_{st} && \forall s \in S, t \in T \\
& p_{st} \leq U^p_{st} && \forall s \in S, t \in T \\
& (w^v_{st}, \mathbf{e}^{v_s}, \mathbf{e}^{x_{st}}) \in P_{v_s, x_{st}} && \forall s \in S, t \in T \\
& (w^c_{st}, \mathbf{e}^{c_s}, \mathbf{e}^{p_{st}}) \in P_{c_s, p_{st}} && \forall s \in S, t \in T \\
& \left(\begin{bmatrix} c_s \\ v_s \end{bmatrix}, \begin{bmatrix} c_{s,1} & \cdots & c_{s,n_{s,k}} \\ v_{s,1} & \cdots & v_{s,n_{s,k}} \end{bmatrix}, \boldsymbol{y}_s \right) \in \text{unionconv}(\mathbb{S}_{s,k}) && \forall s \in S \\
& c_s, v_s \in \mathbb{R}_+ && \forall s \in S \\
& x_{st}, p_{st} \in \mathbb{Z}_+ && \forall s \in S, t \in T
\end{aligned}
$$

Here unionconv($\mathbb{S}_{s,k}$) is the linearized union of partition convex hulls from Equation 2 of Section 3.3.

For our example, the solution to DHFMP$_1$ is shown in Figure 9. Here the partitioning $\mathbb{S}_{2,1}$ disallows the previous non-vertex solution for System 2 and causes the fleet-level optimization to choose $P_{2,4}$. This more expensive realization of System 2 necessitates a reduction in yearly production to 3 systems - freeing up some budget and allowing for a more expensive instantiation of System 1. Unfortunately, this new System 1 design is not a realizable Pareto optimal configuration, and we must iterate again, this time partitioning the System 1 designs as $\mathbb{S}_{1,2} = \{\{P_{1,1}, P_{1,2}\}, \{P_{1,3}\}\}$ as seen in Figure 10.

Our procedure terminates after solving DHFMP$_2$, since both adaptive systems now choose a Pareto solution from their trade spaces as shown in Figure 10. Notice that the solution of DHFMP$_2$ represents a drastic departure from the original solution of FMP, both in the system designs chosen and the optimal fleet purchases and assignment schedules – improving cumulative fleet value by 0.96% (125.7 vs 124.5). Interestingly, the holistically-suboptimal solution to the original FMP is the same as if we had ignored the disjunctive iterations at DHFMP$_0$ and simply rounded the non-vertex realization of System 2 to its nearest neighbor, $P_{2,4}$. It is only by solving the holistic fleet modernization problem with the disjunctive programming iterations that we discover that it is optimal from a fleet perspective to produce a majority of System 1.

5 Computational Results

To investigate optimization run times, we tested our methodology on a realistic complex MILP model based on the real-world PEO GCS fleet (as described in Section 2), incorporat-

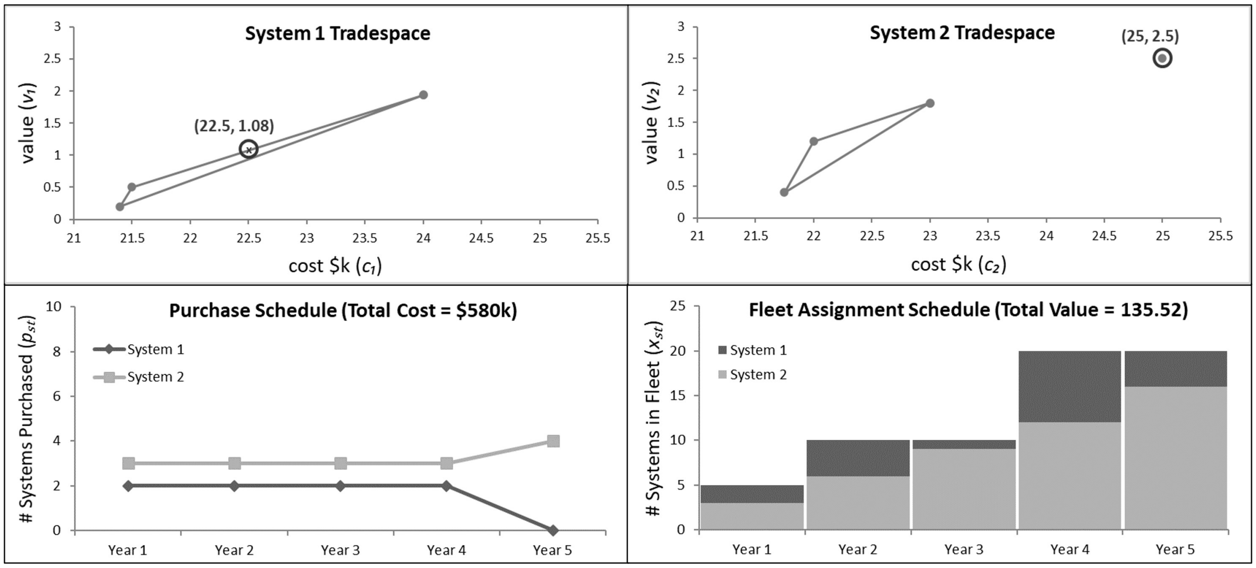


Figure 9: For DHFMP$_1$, where the trade space for System 2 is represented by the union of two convex hulls, the optimization chooses a non-vertex solution for System 1.

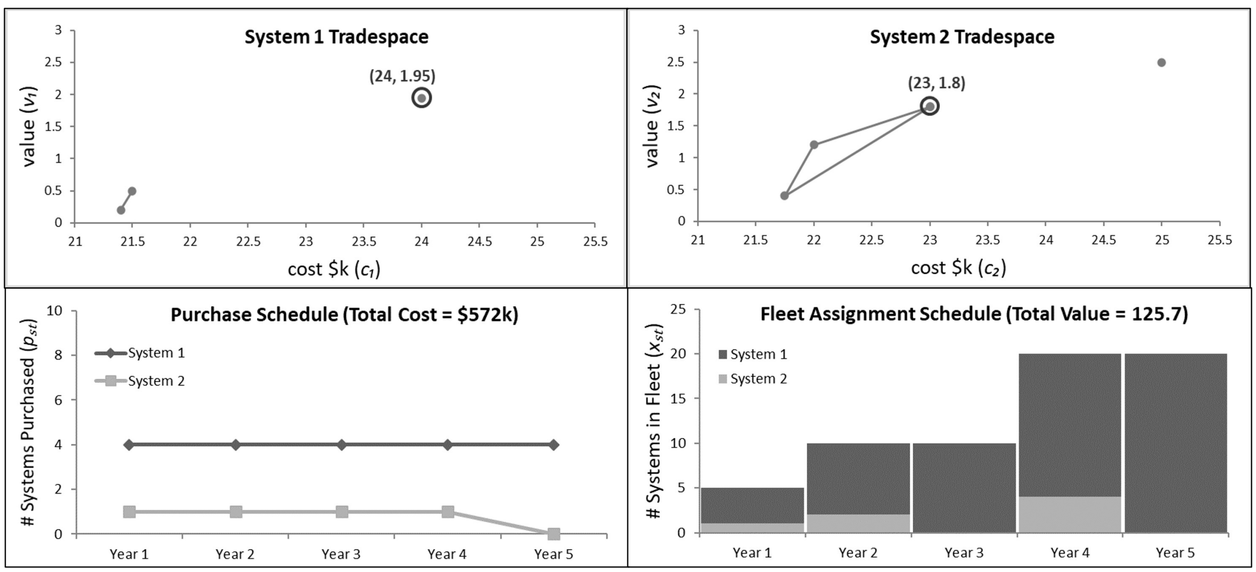


Figure 10: The procedure converges (with all system solutions at vertices) in DHFMP$_2$, where both system trade spaces are represented by a union of two convex hulls.

ing a realistic system design trade space of potential modernization options for the Bradley Infantry Fighting Vehicle. The MILP schedules the optimal modernization of a fleet of nearly 12,000 individual systems over 35 years, with over 70 possible system types (one of which is the Bradley modernization effort). The Bradley trade space contains over 400 solutions that are Pareto optimal in the four dimensions of value, purchase cost, R&D cost, and year first available.

Letting N denote the number of Pareto solutions to be included in the fleet problem, we compare the optimization run time of our adaptive methodology (using the convex hull of these N designs) against an "obvious" approach that represents each of the N designs as a separate, discrete system and includes a constraint enforcing that at most 1 of the N systems may be selected for investment and inclusion in the fleet. To compare the computational scalability of these equivalent representations, we select various values for N and compare the average run time of 10 replications each of our adaptive methodology and the obvious approach. All optimization runs were performed on an Intel Xeon E5-1603 v3 CPU (benchmark score of 4449[1]) using 32-bit CPLEX 12.6 and 16Gb of RAM.

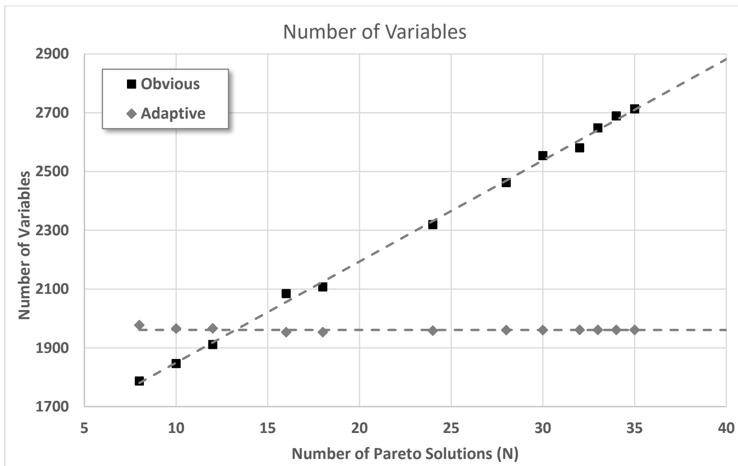


Figure 11: Variable count (after CPLEX preprocessing) increases roughly linearly with number of design options for the obvious method, but is constant for the our adaptive method.

Figure 11 shows the number of variables in our adaptive formulation versus the obvious formulation *after* CPLEX preprocessing has produced an equivalent, smaller MILP. Not surprisingly, for small N, the new variables introduced in the bilinear reformulations (1) of our approach outnumber the variables in the obvious approach. However, as N increases, the variable count for the obvious method grows linearly since each solution is represented as a separate system with its own variables (about 35 per each new system). For our adaptive method the number of variables remains constant, as the convex hull requires the same number of variables regardless of how many solutions are being enveloped.

Figure 12 similarly shows that the obvious approach has fewer constraints for small values of N (again, counted *after* CPLEX preprocessing) but gives up this advantage as N grows. For the obvious method, each new Pareto solution adds roughly 100 new constraints to the preprocessed formulation. Our method, on the other hand, requires roughly 1 new constraint for every 3 new Pareto solutions, since some Pareto solutions lie within the interior of the convex hull and do not induce new facets. Thus, in both variable and constraint count, for $N \geq 16$ our adaptive method has a clear advantage in terms of formulation compactness.

The question remains whether the smaller formulation size translates into shorter optimization times. Figure 13 shows the mean optimization run time (averaged over 10 replications) of our adaptive method versus the obvious approach. In this figure, we plot the average time taken for CPLEX to achieve a strict 0% gap using full parallelism (error bars denote

[1] https://www.cpubenchmark.net/

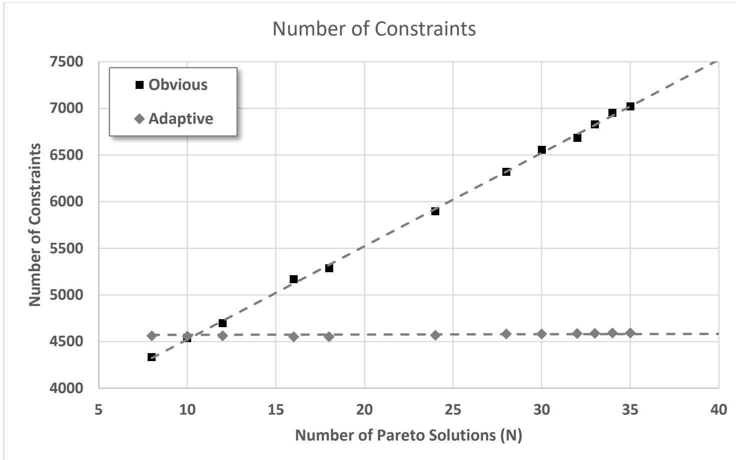


Figure 12: Constraint count (after CPLEX preprocessing) increases roughly linearly with number of design options for both methods, but much more slowly for our adaptive approach.

the standard deviation). Note that the obvious and adaptive approaches incur very similar run times (consistently around 300 seconds) up to about $N = 25$. Beyond this, however, the mean run time and run-to-run variation of the obvious method drastically increases, while the time for our adaptive approach remains relatively consistent. For $N = 35$, the obvious method breaks down and we experience numerous CPLEX crashes as the branch-and-bound tree becomes too large for memory (the best-case run reached 0.0005% gap after over 8 hours - labeled in Figure 13 as DNF for "did not finish"). Meanwhile, our adaptive approach remains consistently tractable at $N = 35$. While direct comparison of the obvious and adaptive formulations is not possible for $N > 35$, testing has shown that our adaptive method still remains tractable (times less than 10 minutes) even for $N > 400$. As expected, since formulations for both methods are equivalent and were run to 0% gap[2], both methods achieve the same objective function value for each $N \leq 35$.

Computation time for our adaptive method includes 1) the time taken to calculate the Pareto convex hulls (which is nearly instantaneous even for $N > 400$) and 2) the time spent performing disjunctive iterations. However, for *every* adaptive run in Figure 13, the optimization chooses a vertex of the convex hull on the initial iteration and *never* requires additional disjunctive iterations. Encouragingly, this suggests that it is naturally desirable (at least for our particular fleet formulation and system trade space) for the MILP to choose adaptive system parameters corresponding with a realizable Pareto optimal system design. If disjunction were required, each iteration would add 1 variable and at least 2 constraints.

These results appear to indicate that our adaptive method is a much more scalable formulation than the more obvious approach. The adaptive method solves efficiently even when including many Pareto designs, allowing larger, richer system design trade spaces to be represented – which ultimately enables better overall fleet modernization schedules. The

[2] A 0% gap may seem excessively tight, however, the primary CPAT use case is to compare multiple acquisition strategies under different input conditions (such as different budgets, production capacities, etc.) against a baseline plan and then *justify* the best strategy under those conditions. Unfortunately, these problems often have nearly alternate optimal solution strategies, hence the need for very small solution gaps.

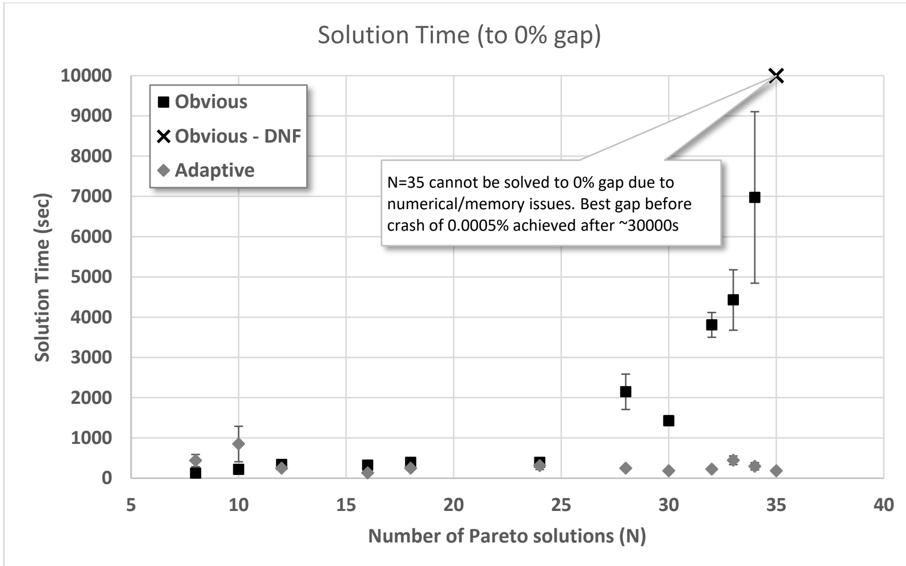


Figure 13: Our adaptive method remains tractable out to (and well beyond) $N = 35$, at which point the obvious formulation becomes computationally unstable.

method provides for the first time a scalable, holistic fleet optimization capability that unifies large-scale, real-world fleet and system trade space applications.

6 Summary

This paper illustrates a new framework for performing holistic portfolio optimization where the composition of the portfolio *and the properties of the items therein* are simultaneously optimized to achieve the overall best performance. This methodology was designed to perform at the large problem scales necessary to incorporate real-world system design trade spaces (with hundreds of individual Pareto optimal configurations) into real-world MILP fleet optimizations (with hundreds of different system types, some of them with trade spaces, respecting dozens of business rules), and has been successfully shown to efficiently unify full-scale trade space and fleet optimization models. Our approach hinges on three key ideas: 1) the use of convex hulls to efficiently capture linear approximations of discrete trade spaces, 2) the use of efficient reformulations to linearize the resulting bilinear terms, and 3) an iterative application of disjunctive programming in the cases where adaptive system parameters do not correspond to an actual Pareto solution. A simple fleet optimization example demonstrated the technique and showed how small changes to the properties of individual systems can substantially alter the overall optimal fleet plan and cumulative value. This important insight highlights the significant potential of holistic portfolio optimization.

This work suggests several avenues for further research. While the disjunctive procedure we discussed works with any hyperplane cutting through the selected non-vertex system design, it is possible that some choices of cutting plane may be more efficient or effective than others. Similarly, although the linearization method employed generates the most compact linear representation of the bilinear problem, other linearizations may produce problem structures more amenable to efficient solution by large-scale MILP solvers.

Given the somewhat unexpected result in Section 5 that disjunctive steps are *never* needed in our large-scale computational examples, it is natural to ask under what general conditions is LHFMP=DHFMP_0 guaranteed to select convex hull vertices. While this would certainly be the case when no constraints from the fleet problem cut into the convex hull of the adaptive system(s), it is neither a necessary nor sufficient condition. In general, hyperplanes defining fleet constraints may intersect with system convex hulls in any manner, yet as long as at least one original vertex remains in the intersection, it still might or might not be chosen. A generalization of these conditions may be very challenging but also very useful, as it would guarantee the holistic fleet problem could be solved via a single MILP run.

Another possible avenue for further study centers around system-level optimization to find solutions within a neighborhood of a non-vertex system solution chosen by the MILP. On one hand, if additional system designs could be discovered that better match the desired requirements of the non-vertex MILP solution, this is enormously beneficial and diminishes the challenges of the disjunctive procedure. On the other hand, if *no* system designs can be recovered near the non-vertex solution, this tells decision makers that the fleet-level desires are at odds with the physical constraints of the system design. In either case, further developing this idea could lead to interesting insights.

Finally, although we have demonstrated computational efficiency on a realistic, large-scale problem as the number of Pareto solutions is increased, the case study presented in Section 5 involved a single adaptive system with four Pareto dimensions. It would be interesting to see how this method scales in a number of other respects. For instance, further investigation of cases with 2 or more adaptive systems each having 5 or more dimensions within their trade spaces may uncover additional insights and possibilities for computational improvement. Additionally, it would be interesting to develop at-scale examples that require multiple iterations of the disjunctive process to test how that technique performs computationally.

7 Acknowledgments

The authors would like to send warmest thanks to PEO GCS and the U.S. Army Tank-automotive & Armaments Command (TACOM) Cost and Systems Analysis group. The authors also thank the reviewers of this paper whose deeply insightful comments and questions greatly improved the overall manuscript. The work relates to Department of Navy award N00014-20-1-2072 issued by the Office of Naval Research. The United States Government has a royalty-free license throughout the world in all copyrightable material contained herein.

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology and Engineering Solutions of Sandia, LLC., a wholly owned subsidiary of Honeywell International, Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

References

Abara, J. (1989). Applying Integer Linear Programming to the Fleet Assignment Problem. *Interfaces*, 19(4), 20-28.

Alonso, J.J., LeGresley, P., Pereyra, V. (2009). Aircraft Design Optimization. *Mathematics and Computers in Simulation*, 79(6), 1948-1958.

An, L.T.H., Tao, P.D., Muu, L.D. (1996). Numerical Solution for Optimization over the Efficient Set by D.C. Optimization Algorithms. *Operations Research Letters*, 19(3), 117-128.

Office of the Assistant Secretary Of the Army (Acquisition, Logistics, and Technology) (2021). U.S. Army Weapon Systems Handbook 2020-2021.

Balas, E. (1988). On the Convex Hull of the Union of Certain Polyhedra. *Operations Research Letters*, 7(6), 279-283.

Barber, C.B., David, P.D., Huhdanpaa, H. (1996). The Quickhull Algorithm for Convex Hulls. *ACM Transactions on Mathematical Software (TOMS)*, 22(4), 469-483.

Beaujon, G.J., Marin, S.P., McDonald, G.C. (2000). Balancing and Optimizaing a Portfolio of R&D Projects. *Naval Research Logistics*, 48, 18-40.

Belkhiri, H., Chergui, M.EA., Ouaïl, F.Z. (2021). Optimizing a Linear Function over an Efficient Set. *Operational Research – An International Journal*.

Benati, S., Rizzi, R. (2007). A Mixed Integer Linear Programming Formulation of the Optimal Mean/Value-at-Risk Portfolio Problem. *European Journal of Operational Research*, 176(1), 423-434.

Benson, H.P. (1984). Optimization over the Efficient Set. *Journal of Mathematical Analysis and Applications*, 98(2), 562-580.

Bergamini, M.L., Grossmann, I., Scenna, N., Aguirre, P. (2008). An Improved Piecewise Outer-Approximation Algorithm for the Global Optimization of MINLP Models Involving Concave and Bilinear Terms. *Computers & Chemical Engineering*, 32(3), 477-493.

Bertsimas, D., Darnell, C., Soucy, R. (1999). Portfolio Construction Through Mixed-Integer Programming at Grantham, Mayo, Van Otterloo and Company. *Interfaces*, 29(1), 49-66.

Boland, N., Charkhghard, H., Savelsbergh, M. (2017). A New Method for Optimizing a Linear Function over the Efficient Set of a Multiobjective Integer Program. *European Journal of Operational Research*, 260(3), 904-919.

Brown, G.G., Clemence, R.D., Teufert, W.R., Wood, R.K. (1991). An Optimization Model for Modernizing the Army's Helicopter Fleet. *Interfaces*, 21(4), 39-52.

Chaiblaine, Y., Moulaï, M. (2022). An Exact Method for Optimizing a Quadratic Function over the Efficient Set of Multiobjective Integer Linear Fractional Program. *Optimization Letters*, 16, 1035-1049.

Chazelle, B. (1993). An Optimal Convex Hull Algorithm in any Fixed Dimension. *Discrete & Computational Geometry*, 10(1), 377-409.

Cherfaoui, Y., Moulaï, M. (2021). Biobjective Optimization over the Efficient Set of Multiobjective Integer Programming Problem. *Journal of Industrial and Management Optimization*, 17(1), 117-131.

Cui, H., Turan, O. (2010). Application of a New Multi-Agent Hybrid Co-evolution Based Particle Swarm Optimisation Methodology in Ship Design. *Computer-Aided Design*, 42(11), 1013-1027.

Davis, S.J., Edwards, S.B., Teper, G.E., Bassett, D.G., McCarthy, M.J., Johnson, S.C., Lawton, C.R., Hoffman, M.J., Shelton, L., Henry, S.M., Melander, D.J., Muldoon, F.M., Alford, B.D., Rice, R.E. (2016). Maximizing the U.S. Army's Future Contribution to Global Security Using the Capability Portfolio Analysis Tool (CPAT). *Interfaces*, 46(1), 91-108.

Desai, C., Williamson, S.S. (2009). Optimal Design of a Parallel Hybrid Electric Vehicle Using Multi-objective Genetic Algorithms. *Vehicle Power and Propulsion Conference*.

Dondo, R., Cerda, J. (2007). A Cluster-Based Optimization Approach for the Multi-Depot Heterogeneous Fleet Vehicle Routing Problem with Time Windows. *European Journal of Operational Research*, 176(3), 1478-1507.

Edwards, S.B., Cilli, M., Peterson, T., Zabat, M., Lawton, C.R., Shelton, L. (2015). Whole Systems Trade Analysis. *INCOSE International Symposium*, 25(1), 1133-1146.

Fagerholt, K. (1999). Optimal Fleet Design in a Ship Routing Problem. *International Transactions in Operational Research*, 6, 453-464.

Government Accountability Office (2020). Weapon System Sustainment: Aircraft Mission Capable Rates Generally Did Not Meet Goals and Cost of Sustaining Selected Weapons Systems Varied Widely. *GAO Publication No. 21-101SP*.

Graham, R.L. (1972). An Efficient Algorithm for Determining the Convex Hull of a Finite Planar Set. *Information Processing Letters*, 1(4), 132-133.

Gupte, A., Ahmed, S., Cheon, M.S., Dey, S.S. (2013). Solving Mixed Integer Bilinear Problems Using MILP Formulations. *SIAM Journal on Optimization*, 23(2), 721-744.

Hane, C.A., Barnhart, C., Johnson, E.L., Marsten, R.E., Nemhauser, G.L., Sigismondi, G. (1995). The Fleet Assignment Problem: Solving a Large-Scale Integer Program. *Mathematical Programming*, 70(1-3), 211-232.

Hartman, J. (2000). The Parallel Replacement Problem with Demand and Capital Budgeting Constraints. *Naval Research Logistics*, 47, 40-56.

Henry, S.M., Waddell, L.A. (2016). The Whole System Trades Analysis Tool For Autonomous Ground Systems. *8th Annual Ground Vehicle Systems Engineering and Technology Symposium*.

Horst, R., Thoai, N.V. (1997). Utility Function Programs and Optimization over the Efficient Set in Multiple-Objective Decision Making. *Journal of Optimization Theory and Applications*, 92(3), 605-631.

Jorge, J.M. (2009). An Algorithm for Optimizing a Linear Function over an Integer Efficient Set. *European Journal of Operational Research*, 195(1), 98-103.

Liberti, L., Constantinos, P. (2006). An Exact Reformulation Algorithm for Large Nonconvex NLPs Involving Bilinear Terms. *Journal of Global Optimization*, 36(2), 161-189.

Lio, X., Li, Q., Yang, X., Zhang, W., Li, W. (2008). Multiobjective Optimization for Crash Safety Design of Vehicles Using Stepwise Regression Model. *Structural and Multidisciplinary Optimization*, 35(6), 561-569.

Liu, Z., Ehrgott, M. (2018). Primal and Dual Algorithms for Optimization over the Efficient Set. *Optimization*, 67(10), 1661-1686.

Magnanti, T.L. (1981). Combinatorial Optimization and Vehicle Fleet Planning: Perspectives and Prospects. *Networks*, 11(2), 179-213.

Moulaï, M., Mekhilef, A. (2021). Quadratic Optimization over a Discrete Pareto Set of a Multi-objective Linear Fractional Program. *Optimization*, 70(7), 1425-1442.

Nulty, W.G., Ratliff, H.D. (1991). Interactive Optimization Methodology for Fleet Scheduling. *Naval Research Logistics*, 38, 669-677.

Obayashi, S., Sasaki, D., Takeguchi, Y., Hirose, N. (2000). Multiobjective Evolutionary Computation for Supersonic Wing-Shape Optimization. *IEEE Transactions on Evolutionary Computation*, 4(2), 182-187.

Parnell, G.S., Trainor, T. (2009). Using the Swing Weight Matrix to Weight Multiple Objectives. *INCOSE International Symposium*, 19(1), 283-298.

Powell, B.J., Perkins, A.N. (1997). Fleet Deployment Optimization for Liner Shipping: An Integer Programming Model. *Maritime Policy & Management*, 24(2), 183-192.

Raa, B. (2015). Fleet Optimization for Cyclic Inventory Routing Problems. *International Journal of Production Economics*, 160, 172-181.

Roy, T.J. (1989). Multi-Level Production and Distribution Planning with Transportation Fleet Optimization. *Management Science*, 35(12), 1443-1453.

Rushmeier, R.A., Kontogiorgis, S.A. (1997). Advances in the Optimization of Airline Fleet Assignment. *Transportation Science*, 31(2), 159-169.

Sinha, A., Malo, P., Deb, K. (2018). A Review on Bilevel Optimization: From Classical to Evolutionary Approaches an Applications. *IEEE Transactions on Evolutionary Computation*, 22(2), 276-295.

Subramanian, R., Scheff, R.P., Quillianan, J.D., Wiper, D.S., Marsten, R.E. (1994). Coldstart: Fleet Assignment at Delta Air Lines. *Interfaces*, 24(1), 104-120.

Tahara, Y., Tohyama, S., Katsui, T. (2006). CFD-Based Multi-Objective Optimization Method for Ship Design. *International Journal for Numerical Methods in Fluids*, 52(5), 499-527.

Thoai, N.V. (2000). Conical Algorithm in Global Optimization for Optimizing over Efficient Sets. *Journal of Global Optimization*, 18, 321-336.

Yamamoto, Y. (2002). Optimization over the Efficient Set: Overview. *Journal of Global Optimization*, 22, 285-317.

Yildiz, A.R., Solanki, K.N. (2012). Multi-objective optimization of Vehicle Crashworthiness Using a New Particle Swarm Based Approach. *The International Journal of Advanced Manufacturing Technology*, 59(1-4), 367-376.