Research Report

KATHARINA LEY BEST, JEREMY M. ECKHAUSE, M. WADE MARKEL, NATHANIEL EDENFIELD, DUNCAN LONG, LAUREN A. MAYER, TONY NUBER, LIAM REGAN, MICHAEL J. D. VERMEER, DULANI WOODS, EMILY YODER

Optimizing Portfolio-Level Modernization Investment

An Overview of the Aim Point Investment Model (APIM)

This report introduces the Aim Point Investment Model (APIM), an optimization tool for portfolio-level resource allocation across U.S. Army programs and time. The report is intended to provide a technical overview of the model and its capabilities but also details the motivation for creating the model and recommendations from related research. The recommendations should be of interest to decisionmakers and those interested in improving decision support tools for asset allocation problems.

This section briefly describes the resource allocation problem faced by Army leaders allocating billions of dollars among hundreds of equipping programs each year and outlines the approach and tool we developed to help solve this problem. In brief, the project's objective was to build a method and tool to support quick-turn exploration of modernization investment portfolios in light of changing budget constraints and operational priorities in order to develop rough-order optimal investment strategies across a preestablished set of investment options and a set of budget and requirement assumptions. Given the enormous complexity of the decision space, some sort of automated decision support tool was required. To build that decision support tool, the team explored alternative approaches to extracting the information needed about programs' relative utility and any constraints on the Army's ability to procure the capability from existing data. This report describes one of these approaches, which uses Army prioritization guidance—synthesized from several sources—combined with plausible constraints to produce resource allocation solutions that were at least consistent with the Army's stated modernization strategy.

KEY FINDINGS

- A force package construct helps differentiate value across Army programs while addressing the issue of interdependency.
- While existing Army data can provide information on value categories and constraints, additional value categories and constraints at the program level could improve the efficacy of a decision support tool.
- Optimization analysis requires better information about objectives and constraints.

Defining the Problem

Every year, Army leaders must allocate billions of dollars among hundreds of equipping programs over many years to advance Army objectives in support of the prevailing National Defense Strategy. These decisions are made in the context of preparing the Program Objective Memorandum (POM) and the Strategic Portfolio Analysis Review (SPAR). To provide some idea of the problem's scope, the *National Defense Budget Estimates for FY 2021* lists 310 Army procurement programs—exclusive of chemical weapons demilitarization and ammunition—and 264 research, development, test, and evaluation (RDTE) programs, exclusive of chemical weapons demilitarization (Office of the Under Secretary of Defense [OUSD], 2020a; OUSD, 2020b).[1] The foregoing statistics only hint at the complexity. Many of those programs consist of multiple discrete, albeit related, efforts. The Army's M109 self-propelled howitzer modification program is funding seven different modifications on three variants, all operating on different schedules (Assistant Secretary of the Army [ASA], 2020). This myriad of different programs with all their subcomponents helps realize different kinds of Army capabilities over different time frames. All of them have been deemed necessary to the joint force's current or future effectiveness through the Joint Capabilities Integration and Development System. Moreover, achieving full return on investment in one capability often requires investments in other capabilities.

Abbreviations

APIM	Aim Point Investment Model
APIM-SPARGEL	Aim Point Investment Model for Strategic Portfolio Analysis and Review Goal and Expenditure Limits
POM	Program Objective Memorandum
RDTE	research, development, test, and evaluation
SPAR	Strategic Portfolio Analysis Review

A Need for Decision Support Tools

The enormous complexities of these decisions require decision support from robust analytic processes. The cognitive load of holding the relationship and relative importance of hundreds of programs across many years easily exceeds the cognitive load that any individual decisionmaker can carry. That kind of complexity suggests the need for decision support tools. The Army G-8 has two major decision support tools, the General Fleet Model and the Capital Planning Model. Both are essentially linear programs, the first of which is used to allocate resources within a given portfolio and the second of which looks to make cross-portfolio trades.[2] Both require an assessment of each individual program's utility relative to that of all other programs under consideration, and both assume a linear utility function. That is, the models implicitly assume that adding the last increment of capability provides just as much utility as adding the first.

The Force Package Construct

A key component of the analytic approach and model described in this report is what we refer to here as *force packages*. Force packages are balanced sets of capabilities, defined by the Army as preferred sets of forces needed to achieve a specific end. The force package construct was derived from definitions of such balanced capability sets found in Army strategy. For example, the Army's modernization strategy includes the identification of force packages or subsets of capability and quantity identified as particularly high priority or for employment in particular strategic environments. The highest-priority force package represents the most important "first to fight" set of capabilities. Although definitions of force packages continue to evolve, shift, and merge with other constructs, the general idea of identification of a balanced force with a prioritized capability set is what is important to the analysis presented here. With regard to any individual system, the force package construct implies that the Army gains more utility in equipping the highest-priority force package than it does for equipping the rest of the Army. Therefore, the force package construct can inform resource allocation through a piecewise utility function in which investment in the highest-priority force package yields more value than investment in lower priorities, even if the same platforms are part of both packages. This concept is described in greater detail in the "Defining Value in APIM" section.

Simplifying the Resource Allocation Problem

In this report, we describe an analytic approach and related tool that uses the Army's existing data and information on strategic priorities and their supporting capabilities to inform decisionmaking on resource allocations across programs and time, simplifying and providing structure to the problem described in the previous section. There are four primary strengths of our approach. First, the approach allows for automated identification of an optimal solution (in terms of the model's objective function) with a relatively small number of inputs designed to capture strategic preferences. Second, the information on preferences required to identify a suitably unique optimal solution is limited to include force package–level preferences, program-level value categories, and information on constraints. Candidate categorical variables, force package definitions, and information to inform constraints exist at varying levels of detail in Army data and could be further developed by the Army. Third, the optimization model accounts for force package–level preferences and accommodates the resulting piecewise objective function. Finally, the approach allows for iterative refinement of constraints and preferences to move from the original solution (identified as optimal according to the model's objective function) to the most desirable solution as agreed on by decisionmakers. Combining existing value categories and force package–level preferences with an automated optimization tool provides an optimal resource allocation as a starting point. A decision support tool then gives decisionmakers the power to move from this starting point closer to the best possible resource allocation.

Although the APIM approach has many desirable features for a resource allocation decision support tool, it still requires a significant amount of data collection and cleaning related to force package contents, funding goals, categorical variables for prioritization, and constraints. In addition, like other approaches, its usefulness could be limited by the availability and quality of such data, although the iterative approach mentioned above might help overcome such limitations in data.

Defining Value in APIM

In this section, we introduce the concepts necessary to understand how APIM assigns value to investment in different programs over time. This value assignment is central to APIM, because APIM optimizes resource allocation across programs and time to maximize this value, as described in more detail later. Value assignment is rooted in not only the force package construct but also program-level values based on integrated categories derived from Army strategy. We begin by introducing the small set of input variables that define program value for the purposes of prioritization and then explain how these input variables can be weighted using expert elicitation of high-level priorities and a combined score calculated from value categories. Finally, we introduce the concept of constraints.

Valuing Force Packages and Programs

Any approach for ranking programs and establishing an informed resource allocation across time requires some metric for assigning relative or absolute value. In an optimization approach like APIM, it is this value that is maximized during the optimization step. A primary goal of the approach developed in this project is to minimize the need for subjective evaluation of program utility and for the generation of complex new scores to capture such evaluations. For this reason, the optimization model's utility function is based on force package– and program-level values. Force package–level valuation supports the piecewise utility function, the importance of which has already been described. Force package–level valuation also helps account for issues of interdependence among capabilities in decisionmaking. Because force packages include a balanced set of capabilities, basing the value of an investment allocation on its ability to support such a package automatically places value on consistency in investment across dependent programs. Existing Army program characterizations can fill the role of categorical program-level metrics. This could include, for example, placing any programs identified as high priority by senior leaders during planning processes into their own categories; such identification of groups of modernization programs already takes place as part of Army processes. Although specific program-level metrics are used in this implementation of the approach, the approach is robust and can accommodate changes in the exact type and number of program-level metrics.

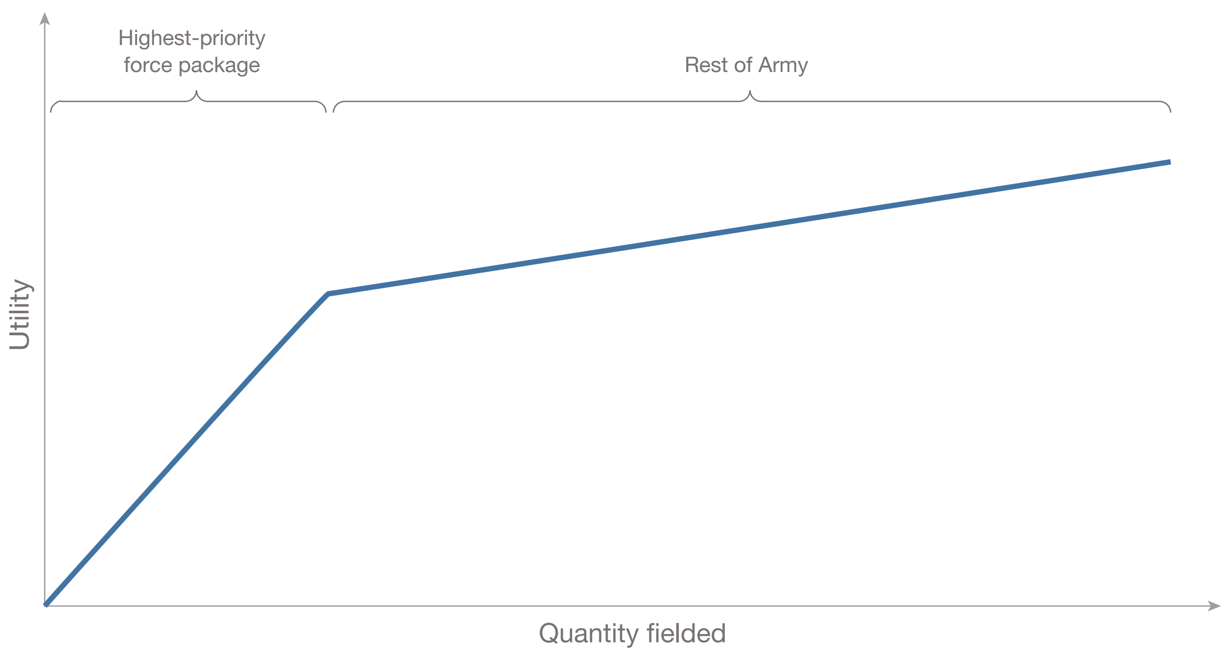
Defining Priorities Through Weighting Force Packages and Milestone Years

As introduced in the previous section, a force package construct can inform resource allocation though a piecewise utility function. Because one of these force packages will be the Army's highest priority for modernization, the Army derives greater utility from equipping it than it does from equipping the rest of the Army. Figure 1 represents this notional utility function. In reality, the dynamic is somewhat more complicated because priorities will evolve over time. Indeed, that evolution might be anticipated. The Army might plan to achieve overmatch vis-à-vis one adversary, after which prioritization might shift to another foe. In this section, we explain how eliciting a small number of force package–level weights when strategic shifts are anticipated can sufficiently capture this piecewise and time-varying dynamic for use in the APIM.

To determine the optimal allocation across programs and time, APIM must specify parameters that value each program. To achieve the kind of piecewise utility function shown in Figure 1, this value must account for the force package to which investment in a program contributes and the timing of that investment. In APIM, this value is specified exclusively by the value parameters V_{ijt}, which represent the value of program *i* within force package *j* at milestone year *t*. We define a milestone year as any year in which the cumulative spending to date is evaluated. The program-specific portion (the *i* portion of V_{ijt}) is discussed in the next section. The force package– and

FIGURE 1

Notional Utility Function for a Quantity of Fielded Modernized Army Capabilities



SOURCE: RAND analysis based on APIM model outputs and Army budget data.

NOTE: Horizontal axis represents quantity of capabilities fielded. Vertical axis represents utility from those capabilities. As quantity increases, the utility function might be nonlinear; highest-priority sets of forces have higher marginal utility.

milestone-specific portions of V_{ijt} are determined from elicited weights across force packages and time.

Using expert elicitation or other techniques outside the scope of this report, each force package *j* and its associated milestone year *t* can be assigned a value or weighting. An example of possible weights (scaled to 100) that could result from such an exercise is provided in Table 1. We denote these weights as w_{jt}.

Defining Value Through Hierarchical Categories for Programs Within a Specified Force Package

Program-level differentiation within a force package and time period is needed to define the complete value of the program in a given force package in a given milestone year, V_{ijt}. Defining differences between the value across programs by developing, for example, pairwise comparisons and stated preferences across all pairs is not realistic for the hundreds of programs in a force package. However, we find we can assign value based on a set of hierarchically ranked categories to which programs can be assigned. The value function based on hierarchical variables has a specific functional form that ensures that each unique combination of categorical scores provides a unique value and that programs in a higher value category for the first variable in the hierarchy will always have a higher value than any program in a lower value category. This functional form is described in more detail below. Army strategy and planning documents are possible sources for categorical program valuations, providing information on, for example, the extent to which particular modernization programs are prioritized by senior leaders and the extent to which capabilities contribute in particular scenarios. In this section, we introduce some illustrative example categories and explain how these combine to form a program-level valuation.

Single-Category Ranking

The simplest example of a category-based program-level value is based on a single categorical variable. For the purposes of illustration, let us assume that such a single categorical variable is a categorical *modernization priority* variable with four possible values into which a particular program may fall, denoted by $r1_i \in \{1,2,3,4\}$. Using this categorical variable, we can assign a value for spending (per dollar spent) for project *i* within each force package *j* and target year *t*, denoted by V_{ijt} as follows:

$$V_{ijt} = w_{jt} + \frac{4 - r1_i}{4}.$$

For example, when $j = A$, $t = 2028$, and $r1_i = 3$, we would have

$$V_{i,A,2028} = 24 + \frac{4-3}{4} = 24.25$$

when $w_{jt} = 24$. The normalization by a factor of 4 in the second term reflects the rank of the categorical variable (four possible levels), ensuring that this second term is always less than one and therefore never dominates the force package–level value term.

Of course, a system based on a single categorical variable might result in many programs with equal value for spending (i.e., equal V_{ijt}). Optimizing based on a value function in which many programs have the same value will result in a relatively large set of equally valued optimal solutions to the resource allocation problem. Although the set of optimal solutions might be smaller than when all programs within a given force package target-year pair are equally valued, the set of these optimal solutions could be large enough that qualitatively different funding strategies might be

TABLE 1

Value per Million Dollars for Each Possible Force Package by Milestone Year

Milestone Year	Force Package A	Force Package B	Force Package C
2028	24.0	12.0	4.0
2035	16.0	36.0	8.0

SOURCE: RAND analysis based on APIM model outputs and Army budget data.
NOTE: The table presents example weightings, scaled to 100, across three notional force packages and two milestone years. Such weightings would result from expert elicitation and be used in APIM to generate inputs to the program-level valuation.

optimal. More differentiation in program value can reduce the number of optimal solutions.

A Multicategory Hierarchy

Specifying a larger set of categorical variables will allow greater differentiation among programs; programs within a given force package will have a broader set of possible unique values (i.e., V_{ijt}). In the following example, programs are ranked according to the following four categorical variables, which are used in Army planning processes:

- modernization priority (*r*1), which specifies program priority on a scale from one (highest) to four (lowest)
- key scenario contribution (*r*2), which represents an assessment of the degree to which the program supports a particular key scenario; possible values are red, amber, or green
- acquisition success (*r*3), which indicates the status of the acquisition effort; possible values are red, amber, or green
- stakeholder assessment (*r*4), representing an assessment of the program's importance across stakeholders; possible values are red, amber, or green.

The hierarchical scores for each of these variables for a particular program are combined to get a value category score for each program. As in the single-variable example, the functional form of the equation for computing value category scores is designed to ensure that each unique combination of categorical scores provides a unique value. In addition, functional form is designed to be hierarchical in nature. The value category of a particular program is based on the hierarchical order of its rank for each categorical variable; the most important (*r*1) categorical variable is the most differentiating. In other words, no program with a modernization priority of 2 (i.e., $r1_i = 2$) can have higher rank than any program with a modernization priority of 1, regardless of its values for the other lower categories (i.e., its values for $r2_i$, $r3_i$, $r4_i$).

Of course, for roughly 800 programs, the values of V_{ijt} will not be unique for every program within a given force package target year pair because there will be multiple programs with the same hierarchy. The level of uniqueness of V_{ijt} depends on both the number of categorical variables and the number of possible levels for each variable. Table 2 shows examples of two possible cases in which the number of categorical variables is either one or four, thus resulting in a large difference in the number of possible value categories. Naturally, an increase in the number of hierarchical categorical variables produces a much larger number of value categories, resulting in fewer cases of identical values of V_{ijt} for different programs.

Combining Force Package– and Program-Level Value

Combining the relative values for force package milestone year illustrated in Table 1 with the program-specific value category in Table 2 provides a value for each V_{ijt}. For example, from Table 1, we see that a program's allocation to force package B in 2028 has a value of 12.0. If force package B's multicategory value

TABLE 2
Examples of Categorical Variables and Their Respective Value Categories

	Single Category	Multicategory
Categorical variables	One tier with four ranks	First tier with four ranks Second tier with three ranks Third tier with four ranks Fourth tier with three ranks
Number of value categories	Four	144

SOURCE: RAND analysis based on APIM model outputs and Army budget data.
NOTE: The table presents two example approaches for defining program-level value based on categorical variables. The simpler single-category approach uses one categorical variable to generate four possible unique program values. The more complex multicategory approach uses four hierarchical categorical variables to generate 144 possible unique program values. Program-level values from such categorical variables are used in APIM as inputs to the program-level valuation.

is ranked $(r1_i, r2_i, r3_i, r4_i) = (4,1,2,3)$ within these four tiered categories, its overall ranking (denoted as $RANK_i$) would be

$$RANK_i = |r2||r3||r4|\left(r1_i - 1\right) + |r3||r4|\left(r2_i - 1\right) + |r4|\left(r3_i - 1\right) + r4_i = (3)(4)(3)(4-1) + (4)(3)(1-1) + 3(2-1) + 3 = 114$$

where, for example, $|r2|$ represents the rank, or number of possible categories, of $r2$. Combining this with the force package value, we can assign a value for

$$V_{ijt} = w_{jt} + \frac{(|r1||r2||r3||r4| - RANK_i)}{|r1||r2||r3||r4|} = (12.0) + \frac{(144-114)}{144} = 12.208.$$[3]

Similarly, if a program's value category is 85, then its value within force package A in 2035 would be $V_{ijt} = (16.0) + \frac{(144-85)}{144} = 16.410$. This valuation ensures that the weights for force package at each milestone year dominate differences among programs, although other hierarchies are possible.

Defining Constraints

Once V_{ijt} values have been assigned, the set of optimal solutions might still be too large to provide an informative answer to the resource allocation problem: There might be multiple asset allocations that provide the same overall value, leaving Army leaders with yet another (albeit much smaller) difficult decision problem. To further limit the number of optimal solutions, APIM applies constraints, such as budget and production constraints, either explicitly described in existing data or reasonably inferred through analysis. The specific types of constraints applied by APIM are detailed in the next section.

The Aim Point Prioritization Investment Model Formulation

In this section, we explicitly define APIM, applying the concepts from prior sections. We provide an overview of the objectives and constraints and the explicit model formulation. We also illustrate two optimal APIM variations. The first, Aim Point Investment Model for Strategic Portfolio Analysis and Review Goal and Expenditure Limits (APIM-SPARGEL), aligns model results more closely with existing programmed funding plans. One possible source for such funding plans is the SPAR, which reflects the Army's investment plans over the next 15 years. The second variation illustrates how program-specific constraints can be used to further tailor APIM. Illustrative model outputs are provided.

Model Overview

APIM, a mixed-integer programming model, provides a mechanism for generating an optimal resource allocation scheme across time and programs that uses piecewise utility functions. Developing an investment strategy over time that is feasible given real-world constraints is challenging. Developing one that is optimal in light of any set of constraints and leadership preferences is even more challenging. APIM uses the set of inputs and constraints detailed below to derive an allocation of funding across programs and modernization force packages over time that is both feasible and optimal given the constraints and objective function.[4] We note that this is not necessarily truly optimal in terms of being the absolute best resource allocation to address the complex set of threats and uncertain futures facing the United States. However, with proper specification of value and constraints, APIM identifies a solution that is a significant step toward solving the more complicated strategic problem.

Objectives

APIM's objective (i.e., utility function) is to maximize the value of *cumulative* resource allocations (i.e., spending) at specific target years subject to the budget, spending limit, and spending continuity constraints. The value of investing an amount in a particular program and force package in a specific year depends on the force package and program characteristics specified for the model run. In this implementation, value is determined by the relative force package valuation and the program's SPAR modernization priority, as described in the previous section. A value coefficient V_{ijt} is calculated for each program (i) in each force package (j) and each time period (t) (i.e., year). These coefficients allow for a given force package to be considered more or less valuable than another (i.e., index *j*) and each program within that force package to be relatively more or less important (i.e., index *i*). In addition, because these modernization force packages are defined not only by the programs that compose them but also by the year in which that portfolio of capabilities should be achieved, values might differ by target year (i.e., index *t*). APIM selects the investment allocation that maximizes overall value in light of this value function.

Constraints

Any set of feasible resource allocations must meet the model constraints. The constraints represent real-world limitations such as production constraints and budget realities. APIM includes constraints that ensure the following:

- Total spending for each program in each year is between the minimum and maximum yearly spend rates for that program.
- Aggregate spending for both procurement and RDTE programs do not exceed their respective yearly budget constraints.[5]
- Current inventory (measured in dollars reflecting the spending by program that occurred prior to the start of the APIM simulation) is allocated across force package goals in a way that maximizes the value, as perceived by APIM's objective function, of current inventory.
- Value for the cumulative spending on each program's force package is not applied to any spending that exceeds the program's modernization force package requirements (i.e., goals).
- Spending on a program does not occur prior to that program's earliest possible funding year.
- Spending on programs does not incur a temporary gap, meaning that programs that fail to meet their minimum yearly spending allocation are treated as canceled and cannot be restarted.

APIM's Formulation

This section provides the notation for the sets, parameters, and decision variables for APIM. It describes the method for optimizing program-level investments across force packages using a mixed-integer program. We first define the data inputs in terms of sets, parameters, and scalars. We then describe the decision variables and objective function. Finally, we provide the mixed-integer programming formulation, followed by a description of the constraints and how they tie into the purpose of the analysis in terms of generating candidate portfolios.

Data Inputs

Sets

We define the following set indices:

- $i \in I$—programs within any number of the force packages
- $i_1 \in I_1 \subset I$—elements of the subset of programs related to procurement
- $i_2 \in I_2 \subset I$—elements of the subset of programs related to RDTE
- j—Aim Point (i.e., force package)
- t—time periods (years) over which the model is run.

All programs are either procurement or RDTE programs (i.e., $I_1 \cup I_2 = I$).

Parameters

We define the following datasets:

- V_{ijt}—value per unit of spending in program *i* within force package *j* at time *t*
- R_{ijt}—requirement or goal (in spending units) for program *i* within force package *j* at time *t*
- MAX_{it}—maximum spending across all force packages for *i* at time *t*
- MIN_{it}—minimum spending across all force packages for *i* at time *t*, if program spending occurs
- CI_i—current inventory (spending) for program *i* at $t = 0$
- MS_i—1 if program *i* has ongoing spending at $t = 0$
- ST_i—time period in which program *i* can first have spending

- $B1_t$—budget for all procurement programs at time t
- $B2_t$—budget for all RDTE programs at time t.

In most implementations of APIM, the values for MAX_{it} and MIN_{it} do not typically vary over time t for a given program. For most values of t, $V_{ijt} = 0$. V_{ijt} is typically specified only for the milestone years of interest (e.g., 2028, 2035). Although the Army likely places value on having systems available before and between milestone years, defining the shape of the payoff curve over time would be very difficult. For this reason, giving the Army credit for acquisition of capabilities at specific milestone years in which $V_{ijt} > 0$ provides a useful simplification. The number of milestone years remains flexible in this formulation, allowing for incorporation of more-complex payoff curves over time, when known, and allowing for different measures of relative importance across milestone years (i.e., different values for V_{ijt}).

Scalars

We define the following scalar parameter: M is an arbitrarily large, positive number.

Decision Variables and Objective Function

Decision Variables

We have the following continuous and binary variables. Some of these variables are determined by the value of other decision variables.

The continuous decision variables are as follows:

- $x_{ijt} \geq 0$—spending on program i within force package j in year t
- $cx_{ijt} \geq 0$—cumulative spending on program i within force package j at goal year t
- $xCI_{ij} \geq 0$—amount of current inventory within program i allocated to force package j.

The binary decision variables are as follows:

- $z_{it} \in \{0,1\}$—1 if program i has spending for any force package in year t
- $s_{it} \in \{0,1\}$—1 if program i started spending (for any force package) in year t.

Mixed-Integer Programming Formulation

Simply put, APIM seeks to allocate budget resources in an optimal way. That is, it seeks to allocate *cumulative* spending (cx_{ijt}) of the budget resources in a way that maximizes value subject to the budget, spending limit, and spending continuity constraints.

As stated, value is assigned by the parameter V_{ijt}, which is specific to the given milestone year, force package, and program of interest. Although a given force package is considered more valuable than another, each program within that force package might be relatively more or less important. This triple indexing retains the ability to distinguish the value relative to other programs, other force packages, and other milestone years.

Below is the formulation for a general instantiation of APIM.

$$\min \sum_{ijt} V_{ijt}\left(R_{ijt} - cx_{ijt}\right) \quad (1)$$

$$\text{s.t.} \sum_{j} x_{ijt} \geq MIN_{it} z_{it} \ \forall\, i, t \quad (2a)$$

$$\sum_{j} x_{ijt} \leq MAX_{it} z_{it} \ \forall\, i, t \quad (2b)$$

$$\sum_{i,j} x_{i,jt} \leq B1_t \ \forall\, t \quad (3a)$$

$$\sum_{i,j} x_{i,jt} \leq B2_t \ \forall\, t \quad (3b)$$

$$\sum_{j} xCI_{ij} \leq CI_i \ \forall\, i \quad (3c)$$

$$cx_{ijt} \leq R_{ijt} \ \forall\, i, j, t \quad (4a)$$

$$cx_{ijt} \leq xCI_{ij} + \sum_{\tau=1}^{t} x_{ij\tau} \ \forall\, i, j, t \quad (4b)$$

$$\sum_{j} x_{ijt} \leq M z_{it} \ \forall\, i, t \quad (5a)$$

$$\sum_{j} x_{ijt} \geq z_{it} \ \forall\, i, t \quad (5b)$$

$$z_{it} = 0 \ \forall\, i, t < ST_i \quad (5c)$$

$$s_{it} = z_{it} \ \forall\, i, t = 1 \quad (6a)$$

$$s_{it} \leq z_{it} \ \forall\, i, t > 1 \quad (6b)$$

$$s_{it} \geq z_{it} - z_{it-1} \ \forall\, i, t > 1 \quad (6c)$$

$$\sum_{t} s_{it} \leq 1 \ \forall\, i \quad (6d)$$

$$s_{it} \leq (1 - MS_i) \ \forall\, i, t > 1 \quad (6e)$$

$$x_{ijt}, cx_{ijt}, xCI_{ij} \geq 0, \ s_{it}, z_{it} \in \{0,1\} \quad (7)$$

Explanation of Objective Function and Constraints in APIM

We group the constraints in APIM's mixed-integer programming formulation by their focus. In some sense, APIM is a version of a knapsack problem with additional constraints (Nemhauser and Wolsey, 1988). The following is a brief explanation of the objective function 1 and constraints 2 through 7:

- Objective 1: The objective is to minimize the value-weighted shortfall between the target for the program in that year and the cumulative spending up to that year summed over each program, force package, and target year.
- Constraints 2a–2b: If a program is funded at time *t* within any number of force packages, its total spending must be between the minimum and maximum yearly spend rates for that program.
- Constraints 3a–3c: The amount of funding for both procurement and RDTE programs across all force packages must be less than their respective procurement budgets. The allocation to the force packages of each program's current inventory cannot exceed the current inventory spending.
- Constraints 4a–4b: The tally for the cumulative spending must be the lesser of (a) the requirement or goal in that year for each program *i*'s force packages *j* or (b) the actual cumulative spending up to that year on that program's force package plus the allocation of current inventory for that program's force package.
- Constraints 5a–5c: These constraints determine whether there was positive spending on program *i* during year *t*. If so, the binary variable $z_{it} = 1$, which affects other constraints, such as 2 and 6. Equation 5c prohibits spending in years prior to when program *i* could commence spending, if applicable.
- Constraints 6a–6e: These constraints restrict the beginning of spending on a given program *i* for a given year *t* to ensure a continuity of spending over consecutive years. Constraints 6a–6d ensure that once funding has started for a given program *i* at a given year t', funding in years $t > t'$ occurs in each subsequent year, or funding must cease. Given the constraints that first year t' in which $z_{it'} = 1$ (i.e., there is spending on program *i* for any force package), we must have $s_{it'} = 1$ for only that time t'. Constraint 6c ensures that there cannot be a $t > t'$ where $z_{it} = 1$ and $z_{it-1} = 0$. Constraint 6e forbids programs with a prior funding history (i.e., there is funding prior to $t = 1$) from starting in any other year other than $t = 1$.
- Constraint 7: Nonnegativity and binary constraints are preserved.

As with many mixed-integer programs, it is important to consider that the portfolio that emerges from solving an instantiation of APIM will not necessarily be unique; it greatly depends on the rigidity of the spending constraints and the number of uniquely determined values for each V_{ijt}. Sensitivities around the values for V_{ijt} can lead to different portfolios, as does relaxing or tightening spending limits (MAX_{it} and MIN_{it}) and overall yearly budgets.

Running APIM

The APIM has been implemented in the General Algebraic Modeling System, a high-level modeling system for mathematical programming and optimization. In addition, a SAS version is available for the basic version of APIM. Run times and associated optimality gaps depend on the number of programs, time periods, and force packages but are generally solvable to within reasonable optimality bounds (e.g., 0.001 percent) within a few minutes on a standard PC for all Army procurement and RDTE programs ($\sim 10^3$).

Input files are flat text files corresponding to each of the parameters noted above (e.g., V_{ijt} being the value per unit of spending in program *i* within force package *j* at time *t* and $B1_t$ being the budget for all procurement programs at year *t*). The outputs consist primarily of the values of the decision variables. Chief among these outputs is the set of x_{ijt}, which represents the spending allocated to each program *i* within each force package *j* at each year *t*, defining the resulting funding strategy of the APIM run.

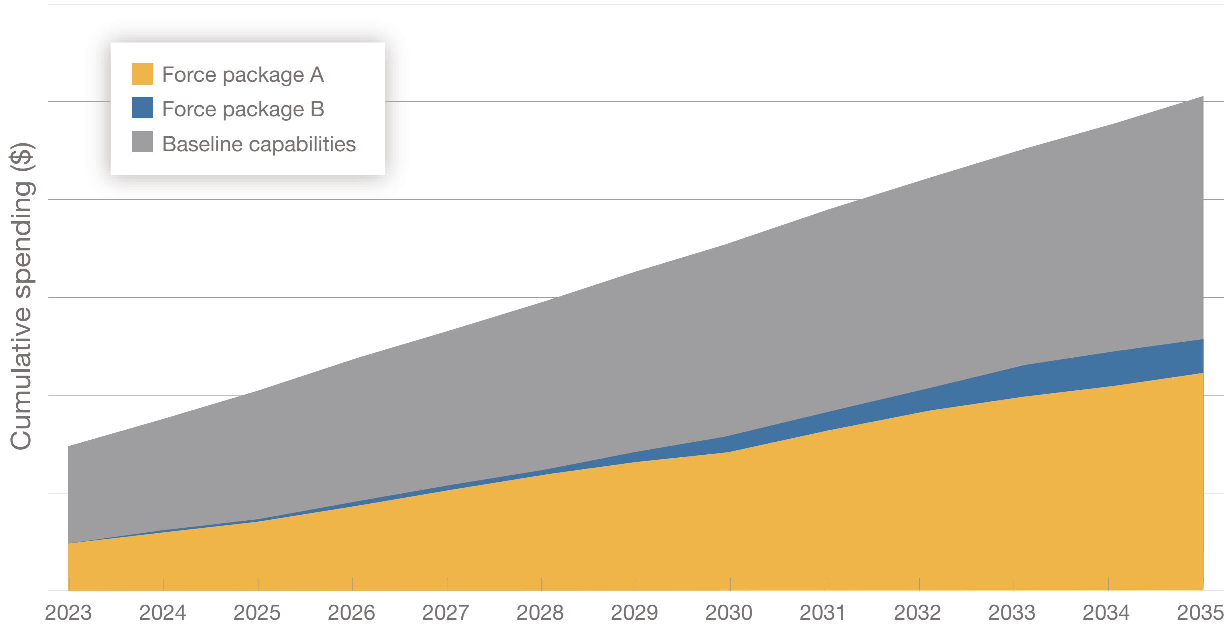
Illustrative APIM Results

In this section, we provide some sample plots of APIM outputs using illustrative data. The sample outputs cover both aggregate and program-level results. The plots shown here are generated directly from APIM output files identifying spending by program, force package, and year (x_{ijt}).

Figure 2 provides a high-level overview of resource allocation by force package. The results

FIGURE 2

Modeled Resource Allocation Across Force Packages and Time



SOURCE: RAND analysis based on APIM model outputs and Army budget data.

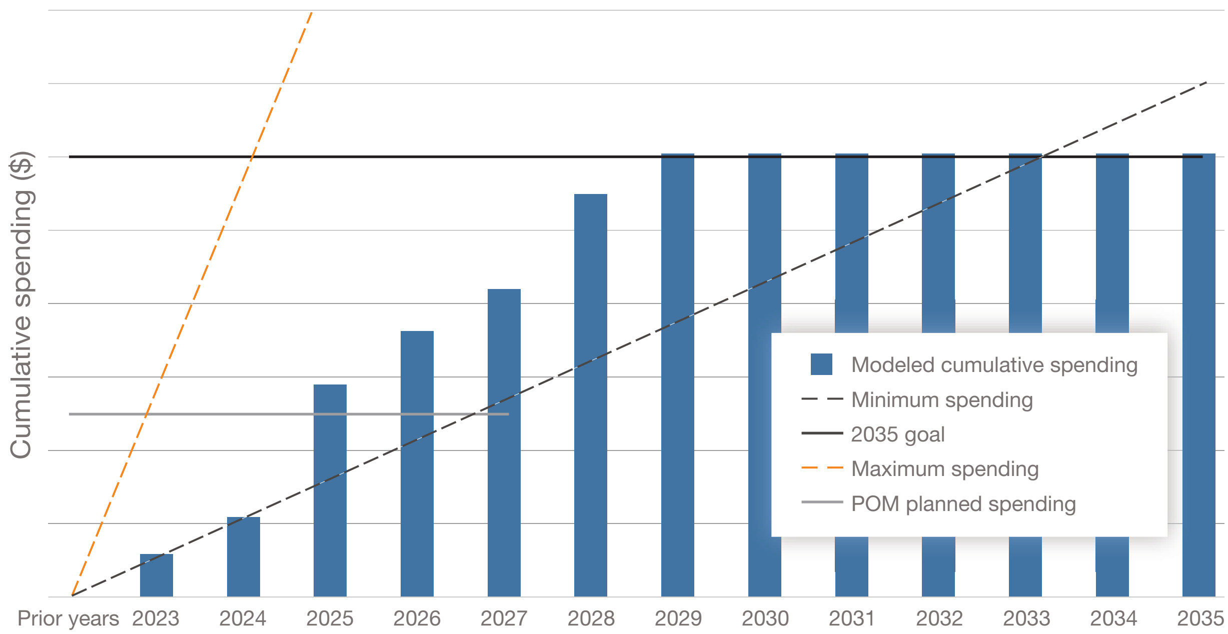
NOTE: Sample notional output based on APIM results. The horizontal axis represents time, and the vertical axis represents cumulative spending across programs. Colors split spending by force package; Force package A is the highest-priority force package from 2023 to 2028, and force package B is the highest-priority force package from 2029 onward.

in Figure 2 reflect the force package prioritization shown in Table 1. Specifically, this run of APIM highly values one force package, force package A, for the period from 2023 to 2028, and shifts the priority values to a different force package, force package B, afterward. We see investment picking up in force package B in the later period, accordingly. At the force package level, these results might not be particularly illuminating, but the lower level of detail provides additional insight.

APIM outputs allow us to drill down as far as the individual program level. Figure 3 shows program-level results for a notional combat platform program. Blue bars indicate APIM-allocated cumulative funding across years. The dotted lines represent the minimum and maximum cumulative spending that would be allowable in APIM based on constraints. The solid horizontal black line represents the Army Acquisition Objective, which is the overall funding goal by 2035. The solid horizontal gray line represents planned spending over the POM period (next five years), as reflected in programmed funding. The "prior years" column on the left side of the chart represents investments that have already been made in the program prior to the 2021 model start. In this example, no prior investments have been made.

The notional program shown here is among the highest-priority programs for the Army and is in value category 1. This means that APIM will value fully meeting the 2035 goal for this program very highly. This is consistent with the results, which show funding very close to the goal by 2028 and meeting it by 2029. In addition, because this program provides a great deal of funding flexibility in terms of the maximum and minimum yearly investment, we see limited investment in early years followed by a period of investment catch-up right before the 2028 milestone year. Because the valuation is only at the milestone years (i.e., 2028 and 2035), the relatively slow spending in 2023 to 2024 does not adversely affect the results given the program's position in 2028 and beyond.

FIGURE 3
Modeled Resource Allocation for Notional Combat Platform Program



SOURCE: RAND analysis based on APIM model outputs and Army budget data.

NOTE: Sample notional individual program-level output figure based on APIM results. The horizontal axis represents time, and the vertical axis represents cumulative spending for a single program. Blue bars represent cumulative spending for this program each year. The horizontal light gray line represents planned spending over the POM period, and the black horizontal line represents the 2035 total spending goal for this program. Dashed lines represent the cumulative minimum spending (black) and maximum spending (orange) for this program over time—two of the APIM constraints.

Extending the APIM to Handle Additional Army Program Data

The standard version of APIM seeks to maximize the cumulative value of spending at milestone years subject to various budget, continuity, and maximum and minimum spending constraints. The value for the spending is capped only by each program's 2035 goal, so any spending in a milestone year that occurs prior to the 2035 final milestone year (e.g., 2028) that does not exceed the final 2035 goal will count positively toward the objective function value. As a result, APIM will often prioritize investment in programs that are deemed high-value above lower-value programs in the years before the first milestone year (2028). In some cases, this may cause spending trajectories prior to 2028 to diverge from expectations, and excessive investment occurs in those highest-priority programs. Although this result makes intuitive sense given the APIM's objective function, it might not reflect the most desirable outcome for decisionmaking support. Therefore, we might want to supply APIM with information on spending plans and priorities to gain a solution more closely aligned with these plans. Such near-term plans could be reflected in things such as POM spending plans (next five years) or SPAR plans (next 15 years).

To minimize near-term deviation from priorities and to reflect the value implied by existing spending plans, we create two extensions of APIM. APIM-SPARGEL introduces constraints that affect the overall objective function in APIM to bring resulting asset allocations more in line with near-term (i.e., five-year) goals. A second, simpler extension allows for the addition of program-level spending constraints, such as near-term POM constraints, to align APIM's solution with specific plans or goals.

Adding Constraints to Create APIM-SPARGEL

Technically speaking, APIM-SPARGEL does not add any additional spending constraints over the regular APIM version. This means that any solution that is feasible within regular APIM will also be feasible in APIM-SPARGEL for the same inputs. It merely restricts the *value* of that solution to be no more than the cumulative spending plan reflected in the SPAR through the first milestone year, which in our illustrative example is 2028 and thus matches up with the end of the POM period. This intermediate goal serves to put results more in line with existing funding plans, and a goal based on a source other than SPAR and a milestone year other than 2028 could be similarly added. Let us denote the cumulative SPAR spending for program *i* at year *t* as $SPAR_{it}$. We then add constraint 4c to the cumulative value that complements 4a and 4b in the original APIM formulation:

$$cx_{ijt} \leq R_{ijt} \; \forall \; i, j, t \qquad (4a)$$

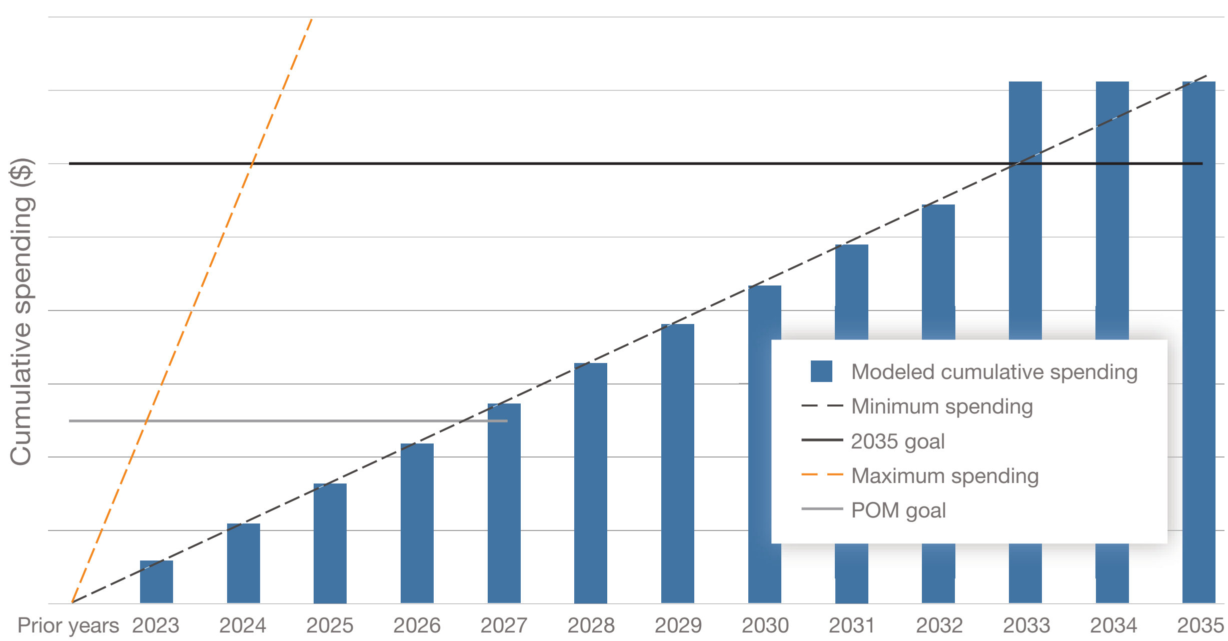
$$cx_{ijt} \leq xCI_{ij} + \sum_{\tau=1}^{t} x_{ij\tau} \; \forall \; i, j, t \qquad (4b)$$

$$\sum_{j} cx_{ijt} \leq SPAR_{it} \; \forall \; i, t. \qquad (4c)$$

So although the actual spending (reflected by the variable x_{ijt}) is not restricted by the SPAR, the cumulative value of that spending (reflected by cx_{ijt}) across all force packages can be no greater than the SPAR plan at that year for that program. Figure 4 shows a corollary to Figure 3, demonstrating how optimal investment in the same notional platform changes when the SPARGEL change to program investment valuation is introduced. The results change between the baseline APIM case and those in APIM-SPARGEL because the APIM-SPARGEL provides no additional benefit for spending beyond the POM goal. Investment in early (POM) years is reduced to the minimum amount because it is possible to meet the goal by 2028 with just this level of funding allocated to the notional combat platform program. We note that in the SPARGEL implementation of APIM, the 2028 goal is equivalent to the SPAR goal, whereas in the regular APIM version, the 2028 goal is the same as the overall 2035 goal. The spending above the 2035 goal in 2033 reflects a budget that is greater than the remaining needs among any viable programs. In reality, new programs or additional funding goals would likely be assigned these budget resources.

FIGURE 4

Modeled Resource Allocation for Notional Combat Platform Program, APIM-SPARGEL



SOURCE: RAND analysis based on APIM model outputs and Army budget data.

NOTE: Sample notional individual program-level output figure based on APIM-SPARGEL results. The horizontal axis represents time, and the vertical axis represents cumulative spending for a single program. Blue bars represent cumulative spending for this program each year. The horizontal light gray line represents the 2028 (POM period) spending goal, which is equivalent to the modeled 2028 spending goal in APIM-SPARGEL. The black horizontal line represents the 2035 total spending goal for this program. Dotted lines represent the cumulative minimum spending (black) and maximum spending (orange) for this program over time—two of the APIM constraints.

Other Program-Level Constraints

Another excursion that decisionmakers might want to make is the addition of specific constraints for individual programs, limiting the decision space. An example of such an excursion is accounting for spending already programmed in the POM. Decisionmakers might want to treat programmed POM funding as a preestablished minimum spending level for some or all programs. To model this in APIM, we can add constraints to either the collective spending by the end of the POM year (e.g., 2027) or the yearly spending minimums within those years, represented by $tPOM_{it}$ and $yPOM_{it}$, respectively. Constraints 4d and 4e are examples of the kinds of program-level constraints that could be added to APIM.[6]

$$\sum_j x_{ijt} \geq yPOM_{it} \; \forall \, i, t = 2023, \ldots, 2027 \quad (4d)$$

$$\sum_j \sum_{t=2023}^{2027} x_{ijt} \geq tPOM_i \; \forall \, i \quad (4e)$$

Other, potentially less restrictive constraints can be implemented, such as ensuring spending deviates no more than a certain percentage above or below the specified POM. Further refinement of the APIM formulation to account for overall or program-level constraints could increase APIM's usefulness to decisionmakers and help APIM be a practical starting point for an iterative decisionmaking approach. In addition, extensions that change the number or spacing of milestone years when investment value is evaluated could be further used to tailor APIM to specific analytic needs.

Findings and Recommendations

APIM is designed to be flexible in that it takes into account as much information on constraints and Army priorities as is readily available without reverting to complex processes of data compilation or scoring. Existing applications of the APIM use a small number of expert-elicited weightings for Army-defined force packages and categorical program-level valuations. Soliciting senior leader input to establish these weights would reflect a change from existing decision processes but could support more deliberate establishment of priorities over force packages and time. With these, we can generate a value function that differentiates between the value of investment in different programs across time sufficiently to yield a solution that is feasible and optimal in light of the value information and constraints provided to APIM. Although attempting to capture the many nuances of the value of a particular investment program with a single value is clearly an oversimplification, any optimization tool will require some such value on which to base allocation decisions. One of the key contributions of APIM is that we find it is possible to create such a value, with sufficient differentiation between programs, without having to compute a value specifically for this purpose. Existing categorizations can provide this value information. Although a single APIM run might not yield a solution that we find to be unique, the systematic identification of a feasible solution that falls somewhere in the optimal space still significantly reduces the amount of program-by-program adjustment that must be done by decisionmakers. Furthermore, the ability to adjust and add overall and program-level constraints can help make APIM part of an iterative process that helps identify the correct optimal solution. Lastly, development of refined inputs, such as clearly defined force packages, value categories, and constraints could help increase the usefulness of APIM in the future.

This section outlines some of the project's main findings and major recommendations that could help the Army use APIM or a similar tool as part of a successful decision support process for resource allocation.

Major Findings

Findings along three major themes emerge from this analysis and are discussed below.

A Force Package Construct Helps Differentiate Value While Addressing the Issue of Interdependence

Intuitively, it is important for the forces that are "first to fight" (in APIM, this is the highest-priority force package) to be equipped with the most-advanced systems before those arriving later in the force flow. Identifying such "first to fight" sets of capabilities simplifies resource allocation decisionmaking in two key ways. First, such constructs simplify the set of questions that need to be raised to senior leaders to elicit strategic preferences. Second, such constructs embody this intuition about balance and synergy. They capture at least some rough form of the aforementioned interdependencies that the Army has struggled to quantify and define. Development of force package constructs that capture the Army's shifting priorities across potential adversaries and time can help support APIM's application and its ability to perform "what if" scenarios that illuminate optimal asset allocations in light of different senior leader priorities.

Value Categories and Constraints Could Improve a Decision Support Tool

In this report, we use the term *value category* to denote the combined score assigned to a particular program based on hierarchical categorical variables that describe that program. A single value category includes multiple programs that have the same value. In APIM, value categories are built on categorical value variables that identify, for example, programs deemed by senior leaders to be high priority. Depending on the number of categorical value variables and possible values (e.g., levels) of each categorical variable, multiple programs could fall in the same value category. Because programs in the same category share the same value, value categories are insufficient to generate a unique program-level valuation for each program. However, with the inclusion of constraints that sufficiently narrow the decision space, these scores can provide enough prioritization information to support identification of a sufficiently unique optimal solution to the resource allocation problem. Examples of such constraints could include constraints around year-over-year funding fluctuation or constraints related to real-world limitations on production capacity. It is not strictly necessary to generate a unique utility score for each individual program. Identification and development of supplementary categorical identifiers that describe program value can further improve APIM's ability to provide a unique optimal resource allocation across programs and time.

Optimization Analysis Requires Better Information About Objectives and Constraints

Mathematical optimization requires constraints and an objective function. An objective function in this context requires information about equipping objectives, especially those objectives associated with priority force packages. APIM or any similar decision support tool must be able to determine the point at which additional investment in a program stops procuring systems for the priority force package and starts procuring systems for the rest of the Army. It also needs information about constraints—the smallest or the largest quantity that can be acquired over a given time frame to keep a program viable.

Existing information about objectives and constraints is incomplete. Although the Army has existing (and evolving) force package constructs, these constructs have not been fully translated into authoritative lists of equipping objectives. Similarly, constraints are explicitly defined for only a subset of acquisition programs. Although information on objectives and constraints is generated as part of planning and acquisition processes, this information is generally not aggregated and standardized in a way that would support a model like APIM. Building out constructs and data to support refined objectives and constraints would improve APIM's ability to provide a valuable and unique optimal resource allocation solution.

Recommendations

The following recommendations describe the kinds of data that the Army will need to collect or create to enable effective use of APIM. Those data fall into three categories: objectives, value, and constraints. We note that any analytic approach rooted in optimization will require similar information.

Continue to Use a Force Package Construct

As noted, the force package construct is very helpful in addressing the issue of interdependencies, something that previous decision support tools have not been able to integrate. Such a construct also helps to facilitate the economical and timely procurement of capabilities over time. Although constructs might continue to change and evolve, the general concept of a balanced force package designed to identify priority needs or meet a particular goal is critical.

Define Value Categories

As mentioned, mathematical optimization requires some value against which to measure. Traditionally, this would imply a requirement to establish a unique value for every one of the almost 900 programs we analyzed, perhaps even in relation to a specific contingency or theater. We found that it was sufficient to align programs with value categories. Although many programs share the same value, combining those values with well-defined constraints allows APIM to find a solution that is a useful starting point for iterative refinement, greatly narrowing the decision space and simplifying the resource allocation problem. APIM's solution in light of value categories is both optimal and sufficiently unique to be useful for decision support. However, for this approach to work, Army analysts must derive value categories from the prevailing modernization strategy and code programs accordingly.

Define Constraints

A comprehensive set of constraints is as important to mathematical optimization as the objective function. However, as per our findings, information on those constraints is available for only a subset of procurement programs. Constraints considered include budgetary and production constraints. No information is available on constraints on RDTE programs. Although it is possible that there are no constraints on such programs, this seems unlikely. Staff investment of time and effort to define constraints with a reasonable degree of precision and accuracy would greatly improve the ability of APIM and similar optimization tools to provide useful solutions.

Develop an Iterative Approach

APIM's results serve as the start of an exploration, not its end. Decisionmakers will want to explore alternatives, either broad priorities or specific programs. Data visualization and support for continuous adjustment of constraints can support iterative exploration of the impact of potential resource allocation decisions. We recommend that the Army further develop this capability.

Conclusion

APIM provides the starting point for a plausible decision support approach that can enable Army leaders to determine how broad prioritization decisions and specific program-level decisions affect the allocation of resources across the portfolio of RDTE and procurement investments over time. In applications of APIM performed as part of the larger study described in this report, we find that APIM can be configured to instantiate the Army's existing modernization strategy, reflecting force package constructs using a piecewise utility function. Further development of the force package construct, information to inform constraints, and program-level value categories, as outlined in the recommendations, could make APIM increasingly useful to decisionmakers in the future.

We close by noting that we developed APIM to inform decisions, not make them. APIM informs decisions by representing how the Army's stated priorities—coupled with constraints on its ability to implement those priorities—would affect resource allocation. APIM provides a point of departure for decisionmaking, not an automated prescriptive plan. It does not—and cannot—represent the full variety of contingent factors with which senior leaders might be concerned, such as their perception of a threat at any given time, how well a particular system is actually going to work in a particular operational environment, or what the ultimate outcome of a research and development effort is likely to be. It also does not serve to inform decisions for parts of the Army budget outside RDTE and Procurement where high-impact future decisions—such as force sizing, mix decisions, and related investments—are likely to be made in the middle term. Future research to expand capabilities like APIM to inform larger pieces of the Army budget could be considered. However, even if such tools are developed further and integrated, Army senior leaders will still have to make final, highly nuanced decisions by themselves. APIM provides them with a starting point for such decisions in the RDTE and Procurement space and helps them understand those decisions' impact on the rest of the program.

Notes

1 We intentionally use the term *program* loosely to connote any centrally managed group of activities directed toward a similar purpose. In the Army context, the term could denote an Army Program Element or some other level of aggregation.

2 A portfolio is a group of programs that perform a similar battlefield function (e.g., aviation platforms or fire support).

3 Note that under a hierarchical categorization, the value category maps to a unique value for the set of categorical variables (e.g., a value category of 85 must map to first tier = 3, second tier = 2, third tier = 1, fourth tier = 2); any integer value can be uniquely expressed as a sum of binary variables with appropriate coefficients of type 2^N (Williams, 2009). This type of hierarchical mapping was chosen because of its availability of existing categorical variables in Army planning and because the hierarchical approach leads to many possible value scores, supporting program differentiation.

4 In optimization, any optimal solution is by definition feasible. We point out both feasibility and optimality here for clarity.

5 Yearly spending constraints for procurement and RDTE are based on Army plans as reflected in the SPAR.

6 These constraints differ from the constraints associated with MIN_{it} in APIM because the MIN_{it} constraints require *either* a minimum spending level *or* zero spending. The constraints in this section are more restrictive because they require a minimum spending yearly and cumulatively in constraints 4d and 4e, respectively.

References

ASA—*See* Assistant Secretary of the Army.

Assistant Secretary of the Army (Army Financial Management and Comptroller), *Department of Defense Fiscal Year (FY) 2021 Budget Estimates: Justification Book of Missile Procurement, Army*, Washington, D.C.: Department of the Army, February 2020.

Nemhauser, George L., and Laurence A. Wolsey, *Integer and Combinatorial Optimization*, New York: Wiley, 1988.

Office of the Under Secretary of Defense, Comptroller, *National Defense Budget Estimates for FY 2021: Procurement Programs,* Budget Appendix P-1_Display, spreadsheet, Washington, D.C., April 2020a.

Office of the Under Secretary of Defense, Comptroller, *National Defense Budget Estimates for FY 2021: Procurement Programs,* Budget Appendix R-1_Display, spreadsheet, Washington, D.C., April 2020b.

OUSD—*See* Office of the Under Secretary of Defense.

Williams, H. Paul, *Logic and Integer Programming, International Series in Operations Research & Management Science*, New York: Springer, 2009.

About This Report

The research reported here was completed in November 2021, followed by security review by the sponsor and the U.S. Army Office of the Chief of Public Affairs, with final sign-off in February 2023.

This report documents research and analysis conducted as part of a project entitled *Financial Decision Support for Portfolio-Level Modernization Trades*, sponsored by the Deputy Chief of Staff, G-8, U.S. Army. The purpose of the project was to develop a methodology and tool to support quick-turn exploration of modernization investment portfolios in light of changing budget constraints and operational priorities and development of rough-order optimal investment strategies across a preestablished set of investment options, given a set of budget and requirement assumptions.

This research was conducted within RAND Arroyo Center's Strategy, Doctrine, and Resources Program. RAND Arroyo Center, part of the RAND Corporation, is a federally funded research and development center (FFRDC) sponsored by the United States Army.

RAND operates under a "Federal-Wide Assurance" (FWA00003425) and complies with the *Code of Federal Regulations for the Protection of Human Subjects Under United States Law* (45 CFR 46), also known as "the Common Rule," as well as with the implementation guidance set forth in DoD Instruction 3216.02. As applicable, this compliance includes reviews and approvals by RAND's Institutional Review Board (the Human Subjects Protection Committee) and by the U.S. Army. The views of sources utilized in this report are solely their own and do not represent the official policy or position of DoD or the U.S. government.

Acknowledgments

We would like to thank LTG James F. Pasquarette, the Deputy Chief of Staff, G-8 (DCS, G-8), for sponsoring this project and for his input on decision support requirements. We would also like to thank COL Jay Persons and MAJ Robert Kennedy for their day-to-day oversight of this project. Many other officials in the Office of the DCS, G-8 assisted us with access to data and their perspectives on decision criteria for resource allocation. We would like to thank our peer reviewers, Todd Harrison of the Center for Strategic and International Studies and Adam Resnick of RAND, for their thoughtful suggestions and greatly improving this report. We would like to thank our RAND colleagues, Gian Gentile and John Jackson, who helped us test elicitation instruments. Finally, we would like to thank Emily-Kate Chiusano for her assistance in preparing this document, and Quiana Fulton and Andrea Wiernik for their assistance with project administration.

For more information on this publication, visit www.rand.org/t/RRA686-2.

www.rand.org

RR-A686-2