Towards More Intuitive Frameworks

in Project Portfolio Selection

by

Siddhartha Sampath

A Dissertation Presented in Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy

Approved February 2018 by the
Graduate Supervisory Committee:

Esma Gel, Co-Chair
John Fowler, Co-Chair
Karl G Kempf
Rong Pan
Jorge Sefair

ARIZONA STATE UNIVERSITY

May 2018

ABSTRACT

Project portfolio selection (PPS) is a significant problem faced by most organizations. How to best select the many innovative ideas that a company has developed to deploy in a proper and sustained manner with a balanced allocation of its resources over multiple time periods is one of vital importance to a company's goals. This dissertation details the steps involved in deploying a more intuitive portfolio selection framework that facilitates bringing analysts and management to a consensus on ongoing company efforts and buy into final decisions. A binary integer programming selection model that constructs an efficient frontier allows the evaluation of portfolios on many different criteria and allows decision makers (DM) to bring their experience and insight to the table when making a decision is discussed. A binary fractional integer program provides additional choices by optimizing portfolios on cost-benefit ratios over multiple time periods is also presented. By combining this framework with an 'elimination by aspects' model of decision making, DMs evaluate portfolios on various objectives and ensure the selection of a portfolio most in line with their goals. By presenting a modeling framework to easily model a large number of project inter-dependencies and an evolutionary algorithm that is intelligently guided in the search for attractive portfolios by a beam search heuristic, practitioners are given a ready recipe to solve big problem instances to generate attractive project portfolios for their organizations. Finally, this dissertation attempts to address the problem of risk and uncertainty in project portfolio selection. After exploring the selection of portfolios based on trade-offs between a primary benefit and a primary cost, the third important dimension of uncertainty of outcome and the risk a decision maker is willing to take on in their quest to select the best portfolio for their organization is examined.

DEDICATION

In the hallway of the university, Wittgenstein asked a colleague: "I've always wondered why for so long people thought that the sun revolved around the earth."

"Why?" said his surprised interlocutor, "well, I suppose it just looks that way."

"Hmm", retorted Wittgenstein, "and what would it look like if the earth revolved around the sun?"

This puzzled the interlocutor.

***

I'd like to thank my family for their patience. My grandparents Major Kumar and Lalitha Kumar, my father Sampath Kumar and mother Asha Kumar, my sister Swathi Sampath and my wife Chithu Mohanan for believing in me and goading me on over the finish line by God's grace.

ACKNOWLEDGMENTS

I'd like to thank my advisor at ASU Dr. Esma Gel for her continued support and close guidance without which this thesis would still be in progress.

To Dr. John Fowler for his practical approach to matters in this thesis.

To Dr. Karl Kempf, who gave me an opportunity to work on a problem of this magnitude and helped navigate this tumultuous area of research with patience and vision.

To Dr. Rong Pan and Dr. Jorge Sefair for steering the dissertation in the right direction when it threatened to go off the rails.

To the Decision Engineering Team at Intel, Andre Lowe, the finance lead whose insight into the modeling and mapping of the various projects drove our effort. To Anthony Vizcarra, and later Danielle Jarczyk, project leads for the portfolio implementation project, for keeping us focused on the details. To Murali Samanthapudi, Vivek Pai, Robert Hill and Michael Kilkenny for their valuable support and practical technical assistance in bringing this effort to fruition.

TABLE OF CONTENTS

CHAPTER Page

CHAPTER Page

LIST OF TABLES

Table Page

LIST OF FIGURES

Figure Page

Figure Page

Figure Page

Figure Page

Chapter 1

INTRODUCTION

Project Portfolio Selection (PPS) is a problem of great significance to the success of a company and refers to the activity of choosing a feasible roadmap while optimizing certain objectives, minimizing risk and the use of limited resources in a manner sustainable over the entire planning horizon that may include many time periods. The objectives are often intangible, strategic or qualitative in nature. As such, a robust product portfolio selection process must allow for all kinds of conflicting objectives to factored into the decision making process. Another difficulty in the process is the inter-dependencies between the various entities in the selection process. In our implementation we came across numerous such interdependencies that included projects that were mutually exclusive, depended on other projects in order to be completed and projects that were essential for the road map to work. Moreover, resource consumption of the various entities may not be well known and the available resource to optimize against may also be uncertain. For example, decision makers at Intel often have a 'signing authority' in US dollars that they may use at their own discretion. If a demonstrable case can be made for value addition, decision makers can often endorse large sums of money to expand their portfolio to far exceed the initial constraint. Headcount constraints can also be overcome as Intel's standing as the world's largest chip manufacturer translates into the fact that it can often find and hire the right skill set at short notice. While these may not always be the case (and signing bonuses and additional hiring to include expansion in a division's portfolio) are not frequent events, a useful decision support methodology will incorporate the possibility of these occurrences into the process. The difficulty in product portfolio selection is compounded by the fact that not all projects for consideration in the process are at the same stage of their life cycle. While some projects may be in the manufacturing stage, others may still be in R & D while still others are only concepts that have just entered the exploratory stage. The uneven nature of resource consumption also means that a portfolio full of R& D projects that are resource feasible for one year and promising future payouts, may be unsustainable in future years when additional R & D and manufacturing costs cause the resource consumption of a portfolio to balloon. Entities in the portfolio optimization process themselves may not be immutable. A project close to the end of its R & D stage may now be valued much more (or less) than initial estimates in previous time periods. As such a senior decision maker may want alternatives that are provided more (or less) resources to that particular entity. It is thus important to provide the ability to decision makers to model alternatives to entities and view how portfolios may change when including different versions of a particular product. Another challenge for a PPS framework that aims to be successful within an organization is its defensibility to senior decision makers. Unrealistic assumptions that present senior decision makers alternatives that differ from actual choice behavior are often exposed and rejected. Thus any framework that seeks to address the problem succesfully must provide a quick and effective way to allow interaction between the intuition of senior decision makers and the analytics of various mathematical models. In Archer and Ghasemzadeh (1996), the authors state that "One of the major reasons for the failure of traditional optimization techniques is that they prescribe solutions to portfolio selection problems without allowing for the judgment, experience and insight of the decisionmaker". The authors also present a review of the various product portfolio and optimization techniques, and note that when projects have a lot of interdependencies project selection is typically handled by optimization techniques. Optimization techniques address a lot of quantitative issues and combined with the right decision support systems, can be used to inspect and incorporate qualitative issues into the decision making process as well Liberatore and Titus (1983). In Stummer and Heidenberger (2003), the authors present an optimization technique to optimize a portfolio with a lot of project interdependencies, and use a three stage approach to minimize the number of projects that enter the optimization stage before an interactive phase with decision makers. This paper attempts to include all projects in the optimization stage and not prematurely discard opportunities that seem poor but may actually be part of an optimal contribution. Dickinson *et al.* (2001) describes the use of a dependency matrix, to model project interdependencies and couple it with a nonlinear, integer program model to optimize project selection, but admit that obtaining accurate estimates for the dependency matrix can be difficult and often inaccurate as a corporate exercise. Mathieu and Gibson (1993) deal with a methodology for large scale R& D planning based on Cluster Analysis and model a framework to manage risk in R& D project portfolio management and note that project selection literature indicated a wide variety of models, but low acceptability by R & D practitioners. We present in this paper a novel binary integer fractional linear programming model to maximize the ratio of conflicting objectives and constraints using cuts detailed in Nemhauser and Wolsey (1988). The elimination by aspects method of discarding less desirable portfolios as a decision rule that we employ is discussed in some detail in Gigerenzer and Selten (2002).

PPS is a major problem faced by most organizations around the world in every major industrial sector, each with its own challenges and complexities. We develop in this thesis a mathematical framework with an emphasis on practicality and defensibility to senior decision makers at a major semiconductor manufacturing firm that largely follows the three broad stages as detailed in Archer and Ghasemzadeh (1996) of project screening, project evaluation and portfolio selection. The mathematical models used in project evaluation and portfolio selection are discussed in detail and finally, a discussion of how to factor in risk during portfolio optimization is presented to illustrate in greater detail the workings of the framework. While the second chapter sets up the basic framework and demonstrates how to apply the framework, to smaller problem instances, the third chapter of the thesis generalizes the formulation and describes a optimization algorithm to solve very large problem instances when the number of interdependencies between projects is so large that even listing out the number of elements in each category is not feasible for practical purposes. We attempt to investigate a genetic algorithm that will fit in with the decision making framework as described before and examine the complexities involved with this approach. The fourth chapter takes the framework further to tackle incorporating risk considerations into the PPS framework.

While further explanations are provided within the chapters, a quick note to motivate this thesis and the various problems it focuses on would be a good thing to get into. One of the major difficulties with PPS is the interdependencies between the various portfolio units. To understand this concept better, consider the following analogy of a hiker who is about to embark on a long trek and is faced with many options on various items she can carry in her knapsack. Considering finite capacity, she has to choose a subset of items that maximizes the objectives she has for the trip. This is the classic knapsack problem to which extensive literature has been dedicated throughout the years. But what if the value of the items she had to choose from depended on what other items she decided to carry? In Figure 1.1, we have a collection of items or portfolio units to choose from. The portfolio unit Ice shares some relationships with the units Cooling Gel and Plastic Bags, both of which define its neighborhood. We do not consider Water in Ice's neighborhood, because it doesn't (for the sake of this example) affect the value of Ice. Consider a scenario where only Plastic Bags exist but not Cooling Gel. This scenario defines a particular decision unit of Ice and upon deriving the valuation we find that the Plastic Bags enhance the value of Ice as the hiker can now store the Ice longer and prevent the rest of the knapsack from getting wet. In terms of the MCKP, a decision unit is an alternative for a category within the knapsack. The hiker may choose the Cooling Gel or Ice (but not both). They both cost different amounts and can be used differently. Plastic Bags are required for Trail Mix. Plastic Bags may be included in the knapsack with our without Trail Mix, but Plastic Bags must be included if the hiker wishes to have Trail Mix on the hike. The hiker may eventually decide to pack her knapsack with the units Ice, Plastic Bags and Water. This would define her Portfolio.

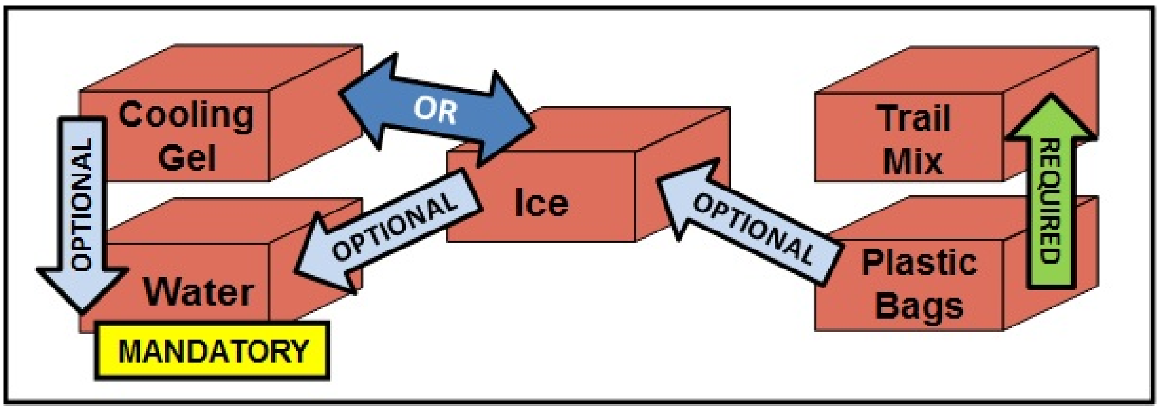


Figure 1.1: A Simple Example of a Portfolio Problem with Entities and Relationships

It becomes obvious that as the number of interdependencies increases between projects, the complexity explodes, and a framework to firstly formulate and then solve is needed. A more detailed description of the problem with further granulation of the entities in the framework is discussed in the second and third chapters. These chapters consider portfolios, that are generated using a primary cost metric and primary benefit metric, but later evaluated using many other metrics. However, the values considered for these metrics are considered deterministic. A chance-constrained based formulation that considers ranges on each of the metrics to produce portfolios that are more robust in the face of uncertainty are considered in the fourth chapter before we present our conclusions.

Chapter 2

A DECISION-MAKING FRAMEWORK FOR PROJECT PORTFOLIO PLANNING AT INTEL CORPORATION

Abstract: The work we describe addresses the problem of deciding between project-funding opportunities under budget and headcount constraints. Although the projects lead to products that yield revenue in the market, complex interactions between the projects and products make the selection of a portfolio difficult. Furthermore, the senior managers in the company have a wealth of business intuition that can inform the required decisions. We combine modeling, simulation, and optimization techniques to provide a set of the best portfolios possible from the proposed projects and resulting products. We also provide a rich set of analysis and visualization tools for the decision makers to use in exploring the suggested portfolios and applying their intuition to make the final selection. The resulting interplay between analytics and intuition produces better business solutions through a more focused and effective debate in a shorter time than achieved previously. **Key words:** analytics; binary integer linear program; decision support; elimination by aspects; intuition; portfolio management; practice of OR; simulation

Funding reviews occur regularly in companies offering a variety of products in a number of markets. At any point in time, many product-development projects are in various stages of execution, ranging from those in their initial ramp-up stages to those nearing new-product launch; all require continued funding. Other projects are new and seeking startup funding. In innovative companies, the number of requests usually exceeds the budget available to fund these requests. Management has to periodically make decisions on the effective allocation of limited budgets to achieve corporate goals, including profit maximization. Making these decisions is inherently difficult because of the combinatorial complexity resulting from the number of projects, products, and markets involved and their extensive interrelationships. For example, one project may support many products. Other projects may depend on each other for intermediate deliverables. One product may be released into multiple markets and one market may receive many different products. Product offerings can affect each other synergistically or cannibalistically in the marketplace. These interactions result in a single project or product having different costs and (or) benefits depending on the other projects or products the company decides to undertake. Funding (or not funding) any specific development project will have far-reaching consequences for products in the marketplace. We define the result of this strategic selection of funding opportunities that advance the company's stated goals as project portfolio planning. Portfolio selection is further complicated by the business reality of dealing with multiple objectives (e.g., revenue, sales volume, investment efficiency), many participants, and the long-lasting effects of decisions. Various projects compete for limited resources with their respective champions' advocating supporting strategies and agendas. Some of these competing ideas are quantifiable, while others are based on qualitative intuition. What constitutes a good portfolio depends on goals that align with an organization's overall vision and resources. A successful portfolio selection framework must therefore allow for open and transparent comparison of both tangible and intangible objectives, as well as project and product relationships, to a committee of decision makers to enable the committee members to ultimately select a good portfolio. With hundreds of innovative products in tens of competitive markets across the globe, Intel Corporation faces project portfolio planning decisions on a regular cadence. With 2014 revenues of $55.9 billion, Intel is a world leader in silicon innovation. Intel product divisions compete in a wide variety of markets, including desktop and laptop computing, servers, networking, communications, entertainment, and embedded computing. In addition, Intel Research operates on an innovation treadmill that regularly produces new product technologies, including enhanced graphics, security, novel sensors, input-output modalities, and connectivity advancements. Considering the diversity and dynamics of both markets and technologies, effectively solving project portfolio planning problems at Intel is vitally important. We have designed and implemented a decision support system that addresses the combinatorial complexities encountered in this process, while making use of the business intuition that senior managers have gained through years of experience in developing and marketing Intel products. We use modeling, simulation, and multicriteria optimization to analyze high-quality data to produce a relatively small set of project portfolio recommendations that maximize net present value (NPV), while respecting the available budget and project relationships. Conversely, we provide a rich set of analysis and visualization tools and easy-to-use what-if capabilities to enable senior decision makers to apply their intuition and unique perspectives to further evaluate our suggestions and select the best portfolio. We have found that managing the interaction between analytics and intuition is a very effective method to produce superior project portfolios from the perspective of NPV and efficient use of budgets. All members of the decision-making team are able to reach decisions with transparency in less time than they previously required. Analytics directs decision makers to the best portfolios in the combinatorial search space. Intuition helps decision makers compare the best portfolios against each other beyond the optimization metrics. The analytical tools support what-if scenarios stimulated by intuition, and drive the convergence to a single project portfolio. We developed the new framework described in this paper with the goals of producing the best analytics and enabling the refinement of the results with the decision-makers' business intuition. Analytics informs intuition and intuition informs analytics; the result is superior business results.

2.1 The Planning Framework

We begin to describe our framework by providing an initial definition of key terms as defined in Table 2.1

Portfolio Unit	Any project or product that a company decides to fund and develop moving forward.
Relationships	A relationship between two portfolio units that affects the value of one or both of them.
Neighborhood	All portfolio units that affect through a relationship the value of a specific portfolio unit.
Scenario	A unique combination of all portfolio units in a neighborhood that affect a portfolio unit.
Decision Unit	A particular version of a portfolio unit as determined by a specific scenario.
Valuation	The end result of assigning values to different metrics of a particular decision unit.
Portfolio	A collection of portfolio units.

Table 2.1: Definition Of Key Entities And Terms Used In Our Framework

We include three types of portfolio units in Table 2.2 and five types of relationships between those portfolio units Table 2.3.

Project	A modeling construct that stores the engineering cost information of a given venture. Projects have positive spending and negative NPV.
Products	A modeling construct that stores the manufacturing cost information of a certain physical product or venture. Products have zero spending and zero NPV.
Prospect	A modeling construct that stores the volume and average selling price information of a specific product in a specific market. Prospects have zero spending and positive NPV.

Table 2.2: Definition of the Three Types of Portfolio Units

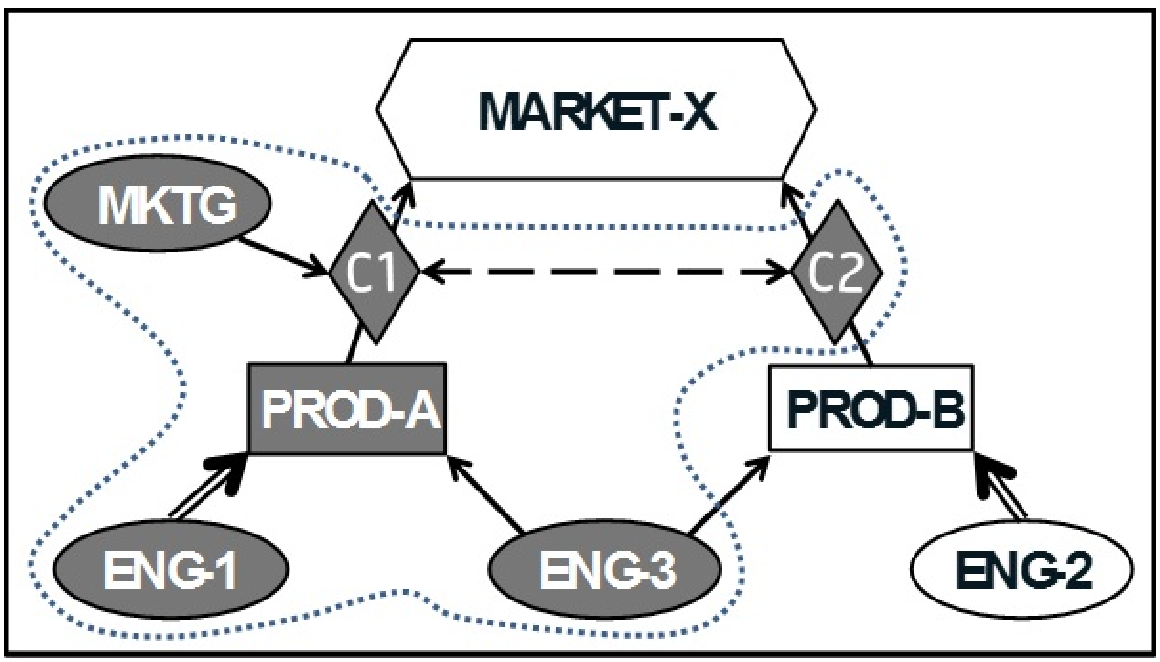


Figure 2.1: A Simple Example of a Portfolio Problem Shows Entities (Projects in Ovals, Products in Squares, Prospects in Diamonds, Markets in Hexagons), and Relationships (Required with Double Arrows, Optional with Single Arrows, Influence with Dashed Arrow)

Required	When one portfolio unit is required by another in order to be funded or developed.
Optional	When a portfolio unit affects the value of another portfolio unit.
AND	When two or more portfolio units must all be funded or worked on together or not at all.
Soft OR	When two or more portfolio units are mutually exclusive and picking one is optional.
Hard OR	When two or more portfolio units are mutually exclusive but one must be picked.
Influence	When two or more Prospects influence market impacts (synergistic, cannibalistic)

Table 2.3: Six Types of Relationships Between Portfolio Units

In the example in Figure 3.1, the ovals represent projects, including three engineering projects (ENG-1, ENG-2, ENG-3) and one marketing project (MKTG). Execution of the engineering projects results in two products represented by boxes (PROD-A and PROD-B). ENG-1 is required to realize PROD-A and ENG-2 is required to realize PROD-B, as the double arrows show. The single arrows show that ENG-3 is optional for both products, although inclusion of ENG-3 enriches both products. Prospects represented by diamond shapes (C1 and C2) connect products to a market represented by hexagons (MARKET-X). Although they are not portfolio units, markets are a construct that stores information for associated prospects concerning the total available market and share of the available market we believe we can capture. The dashed arrow between prospects indicates that there may be synergism (at least one of the respective products nets more value in that market than it would if done without the other) or cannibalism (at least one of the products nets a lesser value than it would if done without the other) between products. The dotted line indicates the neighborhood of prospect C1 including all portfolio units connected to that prospect. Because making a decision regarding the various prospects to bring to a particular market uniquely determines which products to manufacture, and because we define products to have no NPV or spending (these values are incorporated into corresponding projects or prospects), we typically do not include products in our optimization formulations or directly in the calculation of the statistics of portfolios. However, we do indirectly use the information contained in markets and products to determine various statistics associated with prospects and projects using simulation, which we discuss later. We next explain in detail the manner in which some portfolio units affect the value of other portfolio units.

	Decision Unit	Neighborhood Units excluded	**NPV**
1.	C1 : PROD-A + ENG-1 + ENG-3 + C2 + MKTG		166
2.	C1 : PROD-A + ENG-1 + ENG-3 + C2	(- MKTG)	125
3.	C1 : PROD-A + ENG-1 + ENG-3 + MKTG	(- C2)	75
4.	C1 : PROD-A + ENG-1 + C2 + MKTG	(- ENG-3)	55
5.	C1 : PROD-A + ENG-1 + ENG-3	(- C2, - MKTG)	42
6.	C1 : PROD-A + ENG-1 + C2	(- ENG-3, - MKTG)	23
7.	C1 : PROD-A + ENG-1 + MKTG	(- ENG-3, - C2)	59
8.	C1 : PROD-A + ENG-1	(- ENG-3, - C2, - MKTG)	30

Table 2.4: Eight Decision Units in the Neighborhood of Prospect C1 (Figure 1)

The third phase is optimization, which has been previously approached in different ways as reported in the literature. Archer and Ghasemzadeh (1996) and Krishnan and Ulrich (2001) present reviews of various project portfolio selection techniques and note that when projects have many interdependencies, project selection is typically best realized using integer optimization techniques. These techniques address the quantifiable issues and, when used as the foundation for a decision support system, can incorporate qualitative issues based on business-user intuition into the decision-making process Liberatore and Titus (1983). Our optimization approach relies heavily on including both quantitative and qualitative aspects necessary for adoption by the Intel business community. Stummer and Heidenberger (2003) present a technique to optimize a portfolio with many project interdependencies using a three-stage approach to minimize the number of projects that enter the optimization stage prior to an interactive phase with decision makers. Our approach includes all projects in the optimization stage, and does not prematurely discard opportunities that seem poor, but may actually be part of an optimal contribution. Dickinson *et al.* (2001) use a dependency matrix to model project interdependencies and couple it with a nonlinear integer programming model to optimize project selection, but note that obtaining accurate estimates for the required matrix can be difficult and often inaccurate as a corporate exercise. Mathieu and Gibson (1993) use a methodology for large-scale project portfolio planning based on cluster analysis to manage risk in project portfolio management, but note low acceptance by planning practitioners. Our approach models the business by first visually mapping all the projects involved, collecting from analysts estimates for the various metrics associated with them, explicitly simulating various scenarios, and finally converting this nonlinear knapsack problem (NLKP) into an integer program (IP). The decision phase is the fourth phase; in it, senior business managers, who use their intuition to select a portfolio that achieves company goals, further evaluate the results of the optimization phase. A variety of visualizations, novel and innovative incremental-value reports, and waterfall diagrams generated by additional optimization techniques are available to help them. This phase includes executing what-if scenarios, which may lead to more analytics and further pruning of solutions. The elimination-by-aspects method, which we employ as a decision rule, was proposed by Tversky (1972) and discussed in Gigerenzer (2004). In this paper, we present an extensive framework that details modeling the problem in a manner that makes it intuitive to present to both analysts and decision makers. We divide this framework into a mapping section in which we detail an algorithm that recursively solves a binary-integer linear program (BILP) to define the various decision alternatives from which to select, a simulation section in which we assign each decision alternative relative numeric attributes, an optimization section in which we detail an algorithm that recursively solves a BILP to generate an efficient frontier of feasible portfolios that maximize the NPV, while minimizing budget consumption, and a what-if section in which we detail various algorithms we employ to generate additional portfolios that decision makers may consider interesting to explore. In the optimization section, we also present a novel binary-integer fractional linear programming model to maximize the ratio of conflicting objectives and constraints using cuts, as detailed in Nemhauser and Wolsey (1988). Figure 2.2 details the entire process flow.

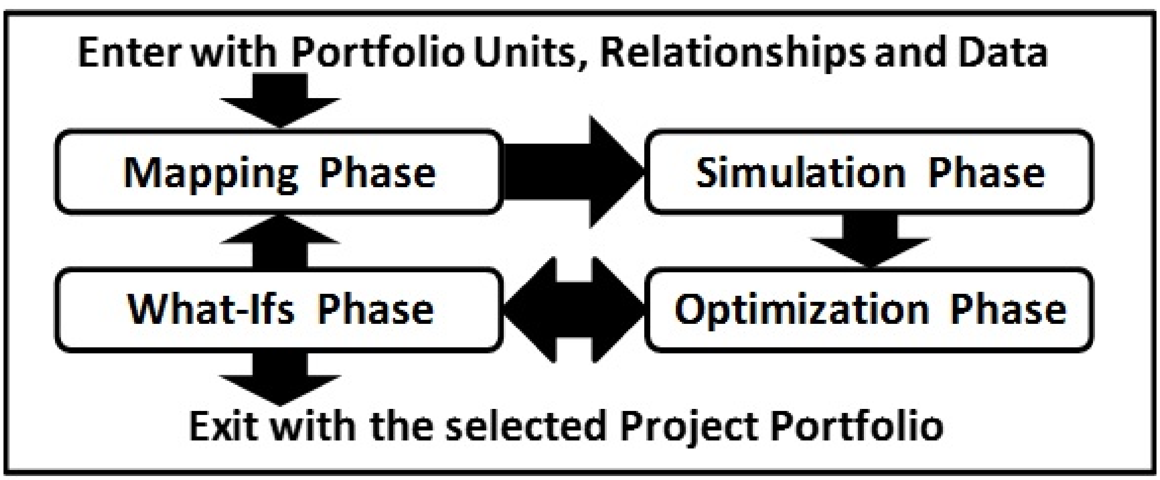


Figure 2.2: Details of the Process Flow in Our Four-phase Decision-making Framework

2.2 Mapping Phase

Analysts typically work on their individual responsibilities for a few weeks before meeting together to map out the company's existing portfolio units and identify all new portfolio units. Assumptions are stated and viewpoints are shared until analysts agree on all portfolio units, their relationships with each other, and all alternatives. Then, quantitative estimates on various portfolio unit metrics, such as spending, headcount, expected sales volume, ASP, time for demand to ramp-up, years at peak, and any other metrics that are important to the decision-making process, are calculated. Analysts enter a low, base, and high estimate for each variable, corresponding to the 10th percentile, median, and 90th percentile values. They also record variable correlations and log annotations capturing their assumptions. Identifying all possible different decision units is often difficult because a single portfolio unit may interact with many others. Identifying all decision units becomes arduous for even moderately connected portfolio units, because each optional relationship doubles the number of decision units. We define the valuations for hundreds of portfolio units by employing an optimization algorithm that generates only valid decision units and determines distinct valuations for each. For a given portfolio unit surrounded by many neighborhood items, we wish to discover all possible feasible combinations of these portfolio units, thus determining the decision units and then evaluating them. We use an IP formulation to iteratively generate all different decision units of a given portfolio unit; see Appendices A and B. We then iterate the IP until all valuations have been generated and no further valuations exist, as determined by finding no feasible solutions for the IP. Each iteration of the IP includes all cuts generated to that point. Aggregating all the solutions from all iterations for all portfolio units, we are able to define matrices that summarize the various valuations and their corresponding decision units.

We use these matrices to constrain feasible combinations of portfolio units when we optimize for various objectives in a later stage.

2.3 Simulation Phase

Now that we have identified and explicitly expressed each decision unit, we need to assign a quantitative value to objective and constraint metrics to perform the subsequent optimization. To achieve this, we record historical and expert input and used the data to calculate decision-unit metrics to arrive at a valuation. The expected NPV (eNPV) of a decision unit is derived via a financial calculation with a number of different variables. Many of these variables may have some uncertainty associated with them, as well as a time dimension. We must therefore run a Monte Carlo simulation to determine the eNPV and other metrics. Consider a simple estimate in which the profit earned equals the number of units sold, multiplied by the difference between the selling price and the manufacturing cost, minus the total engineering investment. Even for this simplified calculation, many of the variables have a time dimension. Engineering costs occur before any manufacturing costs are incurred, and manufacturing costs are incurred before and during product sales. The volume sold in different periods will ramp up quickly, peak for a few periods, and then gradually decline. The problem is thus compounded if we wish to calculate the eNPV of a decision unit, forcing the summation of the investment and (or) costs and (or) revenues in each period, including the discount rate in the multiyear life cycle of the decision unit. Because of these complications, we use a Monte Carlo simulation to estimate the NPV distribution. Depending on the phase of project execution, the engineering investment, manufacturing cost, sales volume, and ASP will not be known with certainty and may best be modeled using distributions. The distribution of overall profit is thus a convolution of distributions that might not all belong to the same family of distributions. The eNPV is equal to the sum of the profits at various periods and is a random variable itself. Some variables may be estimated from historical information. For example, engineering cost depends largely on the number of engineers working on a project over time. This can be inferred from a company's long history of working on similar projects. For typical projects that are underway, headcount numbers for the remainder of the decision unit's life cycle can thus be estimated by extrapolating the existing information until the end of the project as seen in Figure 2.3

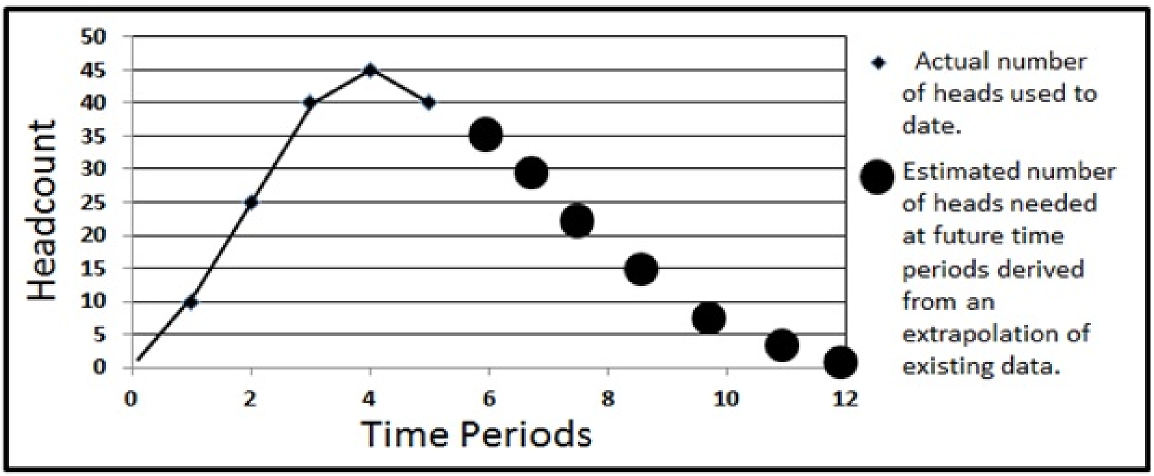


Figure 2.3: This Example Estimates Future Data from Current Data and Historical Experience, Here for Engineering Headcount Needed to Finish a Project

Once all necessary information is at hand, simulations can be run to estimate the values for various decision unit metrics at different stages of their life cycle, and thus derive the appropriate valuation. It is typical to sample the 10th percentile 30 percent of the time, the 50th percentile 40 percent of the time, and the 90th percentile 30 percent of the time. One simulation run will then sample from the distributions (taking into account any correlations that may exist between these variables and simulation runs of other decision units) of each of these for each period and derive a value for the profit at various periods. These time-dependent profits, when appropriately discounted and summed, will provide the eNPV distribution. We typically run between 2,000 and 5,000 simulations and aggregate the results. At this stage, the results are visualized, showing analysts the distribution of the NPV, profit, and other variables at various periods in case they want to adjust some of their initial estimates. Once this cycle has been repeated the required number of times, and analysts are satisfied with their inputs, a quality review team consisting of independent analysts and senior managers reviews the data and especially the assumptions. The review team may call for a revision of assumptions and advocate changing the value of other input parameters. Simulations are rerun until the quality review board is satisfied. We now have the quantitative metrics for the decision units that we can use as an objective to maximize and the spending on engineering efforts in dollars, which could represent a constraint with regard to a group's available budget.

2.4 Optimization Phase

Once all the decision units have been simulated and their eNPV, spending, and other metrics have been determined, we run an iterative optimization algorithm to successively generate all nondominated portfolios in terms of eNPV versus next year's spending budget. We define a nondominated portfolio as a combination of decision units for which no other combination exhibits a lower spending and a higher eNPV. Appendices A and C show the formulation of the algorithm used to generate all nondominated solutions. To generate the first portfolio with the largest possible eNPV, we start by removing any spending constraint. We then iteratively generate all nondominated solutions by including the previous solution as a cut and adding a spending cut to ensure that each solution generated is nondominated. We continue this procedure until we have generated all nondominated portfolios. A subset of the nondominated portfolios generated defines an outer convex hull or the efficient frontier. We identify the portfolios on the efficient frontier algorithm via a fast algorithm by starting with the portfolio that has the lowest spending. We then calculate the slope to every other portfolio in the set and identify the portfolio that provides the highest slope in the objective function space. We select this point and repeat the procedure until we can identify all the points that generate the efficient frontier and present them for initial exploration and analysis. Note that other metrics for optimization can be implemented or incorporated as constraints if they are of interest to senior business unit managers. These may include, but are not limited to, meeting specified revenue targets, maximizing the volume of the company's portfolio, minimizing and (or) meeting headcount specifications, and maximizing the efficiency or the ratio of the eNPV divided by the total spending over all years for a portfolio. At this point in the process, most of the original analysts who mapped the portfolio units are no longer involved. Their role is to assemble a high-quality set of inputs for the optimization. The decision makers involved in the next phase are the senior managers of the business unit and their senior financial analysts who have more global business intuition to incorporate into the generated portfolios. In the next section, we explain in detail the manner in which we manage the incorporation of their intuition, including additional optimization techniques that support intuitive what-if analyses in the decision-making process.

2.5 Decision Phase

The final selection is made by comparing and contrasting various portfolios generated in the previous step, which have been identified as desirable, and discussing the implications of implementing any portfolio on the future of the company. The analytics we describe in the previous sections inform the business intuition through a wide variety of reports supplied by the system, and the intuition informs the analytics through the execution of various what-if scenarios requested by the decision makers. The efficient frontier visualization is an easy way to survey the nondominated portfolios generated in a manner that the decision makers can understand. Figure 2.4 informs decision makers about the least amount of budget they need to spend to achieve a desired portfolio eNPV and the most eNPV they can achieve at their current budget. If a currently approved but relatively inefficient portfolio or future funding strategy exists, the efficient frontier shows the increase in value the company can potentially achieve by switching to a lower-spending, higher-eNPV portfolio. The results of business-intuition-driven what-if analyses are initially displayed on the efficient-frontier visualization.

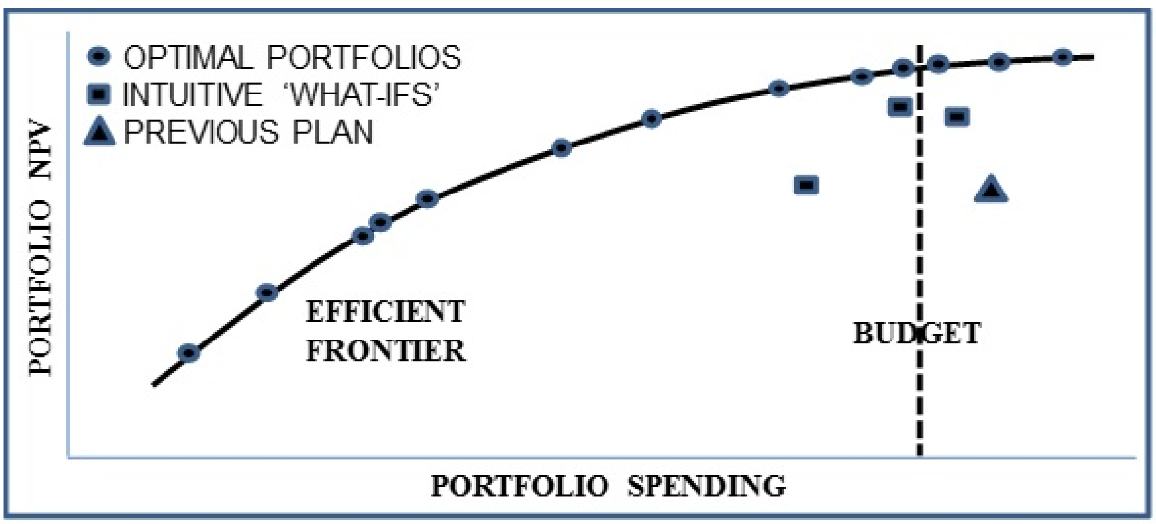


Figure 2.4: A Visualized Efficient Frontier Shows a Variety of Portfolios, Including Optimal Portfolios That Our System Generates Automatically, What-if Portfolios That Our System Generates and That Include the Intuition of Senior Managers, and the Portfolio or Plan Used Prior to the Current Planning Session

Next, a set of basic reports are generated to guide the iteration between analytics and intuition. Selecting a specific portfolio on the efficient frontier generates the listing of all projects and products in the selected portfolio. Selecting two portfolios generates a report that indicates the similarities and differences between the portfolios with respect to the projects and products included. A variant of these reports shows a product roadmap, plotting time on the X-axis and markets on the Y-axis with bars spanning introduction to end of life for each product in the model. Colored bars indicate inclusion in the portfolio selected, while grey bars indicate exclusion. If products are excluded (included) that disagree with intuition, a what-if analysis can be run by forcing those products in (out) as a constraint in our basic IP optimization model; this forces the entity in question to be included (excluded) in the portfolio that resulted from solving the IP and presenting it to the decision makers for review. Although resource-pool size and skill-set mix can be included in our formulation as constraints, a popular option for decision makers is to get a resource view through reports rather than optimization. Selection of a portfolio generates an estimate of the resources required for execution; using their historical knowledge of projects, the decision makers can judge whether the required increase (decrease) in size and change of skill sets to realize a portfolio are practical. We have generated more sophisticated reports to inform decision-makers' intuition in the later stages of portfolio selection. Below, we describe three of the most useful reports; each requires the development and use of further optimization models. The first example is the budgets-efficiency report. Optimizing overall efficiency over time is usually one of the most important objectives for senior managers. It is a difficult task, because spending budgets are often hard to estimate beyond the near future, which may comprise the next year or two; however, companies may still want to increase the amount of benefit realized for every dollar spent over the entire planning horizon. An important consideration arises when a portfolio that looks promising with respect to next year's budget contains multiyear projects that have back-loading spending causing that attractive portfolio to exceed budget in the out years. Optimizing efficiency over many years to generate portfolios within certain spending limits in the current year is mathematically challenging, because it requires maximizing a ratio objective of two linear functions. This involves converting the BILP (see Appendices A and C) into a fractional linear program. Not only is this challenging to formulate, but the difficulty is compounded by the binary nature of the variables. Transforming the BILP into a suitable fractional linear program also makes cutting off previous solutions difficult. It invalidates our previous cuts because, although the value of the new variables still toggles between two distinct values, the higher of those values is no longer 1. As Appendices A and D show, we overcome this problem by converting the original binary integers into a form suitable for the fractional program and introducing additional indicator variables that enable us to continue implementing the cuts that are critical for our basic optimization approach (Appendices A and C). Another example is the incremental-value report, which compares the difference in value of a particular portfolio when a selected portfolio unit is dropped (added) in response to an intuitive what-if analysis. Each portfolio unit has many alternate decision units, each of which is determined by a specific scenario and may cost and be worth different amounts; therefore, the value of a portfolio unit to a portfolio is not readily apparent. A portfolio unit may define scenarios for a variety of other portfolio units. Thus, dropping a portfolio unit from a portfolio based on business intuition may have a cascading effect on a variety of other portfolio units and cause them to be worth a different amount based on the relationships defined between all the units. We explain our incremental-value algorithm in Appendices A and E.

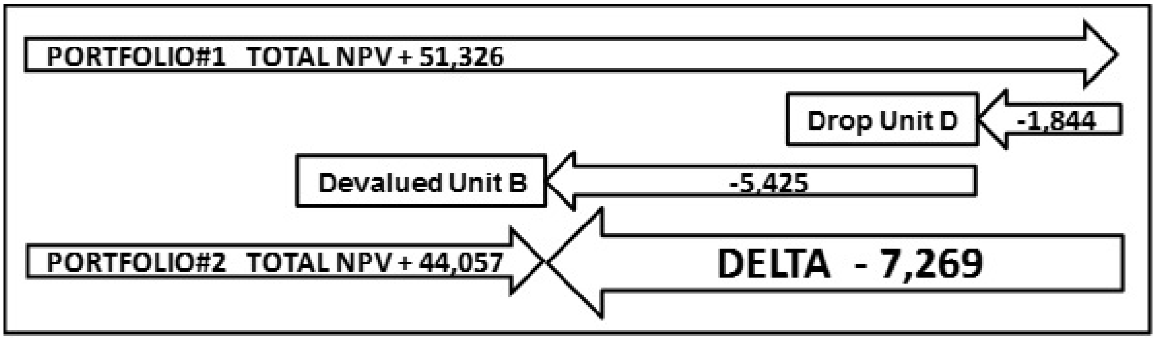


Figure 2.5: Typical Incremental-value Report Shows the Consequence of Dropping Unit D and the Unintended Devaluation of Unit B Caused by Dropping Unit D on Portfolio NPV

In PORTFOLIO#1 in Figure 2.5, when Portfolio Unit D is dropped in a requested what-if analysis, this affects the values of all other portfolio units in the portfolio. Our algorithm provides a feasible PORTFOLIO#2, which does not contain Portfolio Unit D, with minimal changes to PORTFOLIO#1. As Figure 4 illustrates, when Portfolio Unit D is dropped, PORTFOLIO#1 is worth less because of the loss of revenue from Portfolio Unit D; however, Portfolio Unit B is also now worth less because Portfolio unit D contained a technology that would complement the sales of Portfolio Unit B. Differencing the eNPV values between these portfolios gives us the incremental value for the value lost from removing that portfolio unit from the original portfolio, in addition to the changes in value wrought to all the remaining portfolios units in the portfolio. These relatively simple comparisons enable decision makers to have meaningful discussions on the value of various investment options or portfolio units in their portfolios. The waterfall report, which allows decision makers to visualize the changes they would need to make to migrate between any two portfolios, provides a more sophisticated comparison. One common example is from an existing portfolio of record that resulted from a previous planning cycle to a new and improved portfolio suggested in the current planning cycle. Utilizing the same formulation used to generate the incremental-value report, we iteratively add or drop portfolio units, recording the largest negative change at each step and designating that as the next step in the trajectory to generate the waterfall report; see Appendices A and F. Part of the challenge of building a waterfall diagram is the necessity to maintain a feasible portfolio at each step. If an intermediate step is not a feasible portfolio, with respect to satisfying all relationship and budgetary constraints, it is not truly representative of a migration plan from one portfolio to another. Figure 2.6 shows a typical report from executing the waterfall algorithm to morph PORTFOLIO #3 into PORTFOLIO #4. This report indicates that by dropping Portfolio Units M, N, O, P, Q, and R, and adding Portfolio units S, T, U, V, and W, we obtain a net gain in portfolio value. Waterfall reports generated from intuitive what-if analyses lead to important discussions among decision makers looking to choose an optimal portfolio.

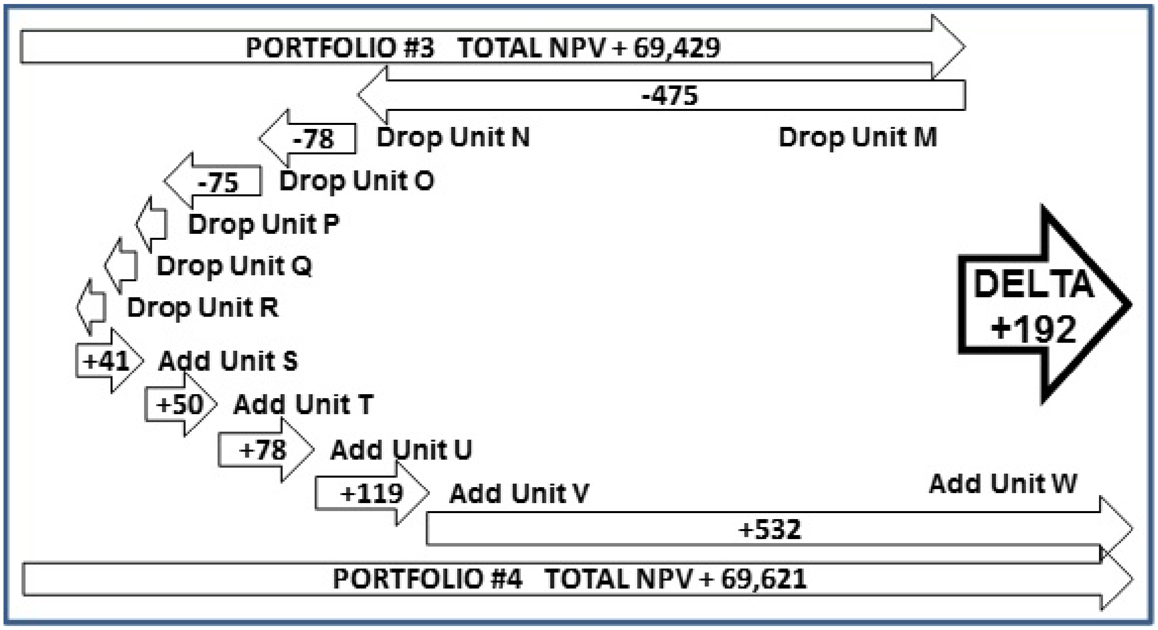


Figure 2.6: Typical Waterfall Diagram Shows the Change in Portfolio Enpv as Six Portfolio Units Are Removed and Five Portfolio Units Are Added in Morphing Portfolio #3 into Portfolio #4

At this stage in the business process, a few valuable portfolios have been identified, in some cases using multiple criteria. Based on the preference of our senior executives, we use an elimination-by-aspects method to prune the existing solutions to choose the plan of record. In using this method, as Tversky (1972) proposes, decision makers prioritize objectives and constraint metrics and then eliminate portfolios at each step as they traverse the list of objectives in a lexicographical fashion. The decision makers identify and discuss portfolio units until they select a final portfolio upon which they can all agree.

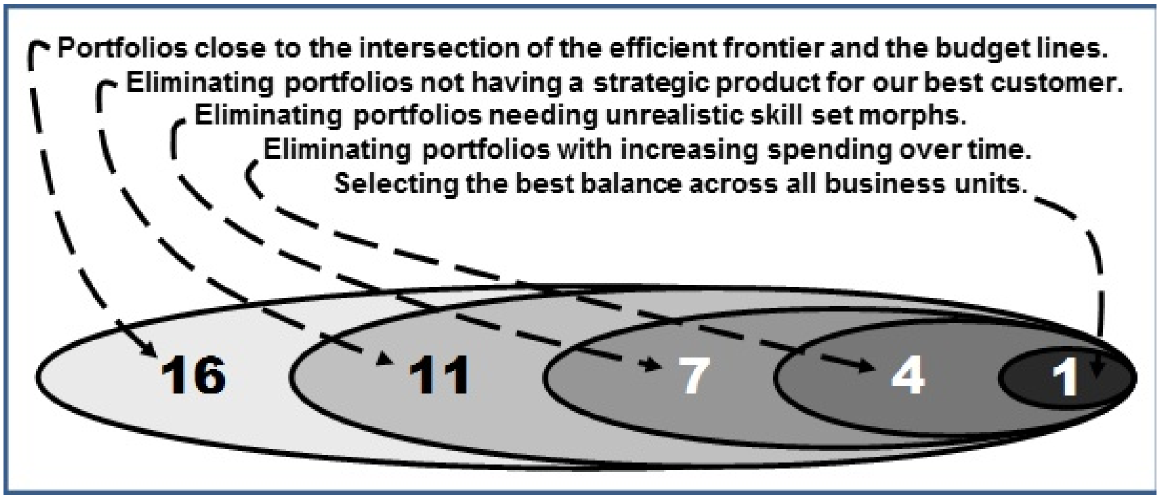


Figure 2.7: This Example Illustrates Pruning the Set of Promising Portfolios Aspect by Aspect to Get to the Final Decision, Specifically, by Budget, Then Strategic Product Inclusion, Engineering Skill-set Availability, Spending over Time, and Balance over Business Segments

Various aspects can be chosen and visited in a number of sequences depending on business goals; Figure 2.7 illustrates one plausible example of this process. After running our analytics process and using our reports and what-if analysis tools, the senior managers select the 16 most promising portfolios relative to the corporate goals they are striving to achieve. Careful consideration of strategic products for key customers, some of which the optimization might have rejected because of low eNPV, reduces the set to 11. The feasibility of finding the number of engineers required to implement the portfolio, and especially the necessary skill-set mix, trims the set to seven members. Using the multiyear-spending what-if analysis tool to detect portfolios with unrealistic spending into the future eliminates three more portfolios. Assuming the decision makers involved in the process represent different divisions of the business being planned, the final selection is made based on balance of investment and return across the divisions, strategic versus tactical, and established business versus new ventures.

2.6 Business Implementation

To implement this process, we need a suite of tools that enables analysts and decision makers to realize the four phases we list above in this paper. At Intel, we built such a suite of tools, which we called Voyager. In the mapping section of the suite, analysts can draw portfolio units and their relationships, and input all relevant details. Our simulation section automatically generates the necessary decision units and their valuations and feeds them to the optimization portion of the suite. Here analysts and decision makers can set up optimization parameters and proceed to investigate and identify possible portfolios. The decision portion of the suite allows for easy navigation of the solution space through intuition-based what-if scenarios and building reports that aid in portfolio selection. We wrote the entire suite in C# and all visualizations utilize the Windows Presentation Framework (WPF) controls. For each optimization, we used IBM's ILOG Concert/CPLEX solver, populating the necessary data structures in C# before calling the CPLEX solver; the solver returns the results in C# to be processed and used as required in interactive displays or reports. The suite is deployed on a remote server and is accessed through a client on analysts' machines, removing most client-machine dependencies. The server's specifications include quad-core Intel i7 2nd generation processors, and 8 GB RAM with 100 GB of disk space. For problem sizes of approximately 200 portfolio units (which in turn yielded approximately 500 decision units), it takes a few days of coordinated planning by the business team to enter all the requisite information into the mapping step. It takes roughly two hours for the simulator to value a problem of this size and make the necessary details available to the optimization engine. The efficient frontier is then built in approximately 30 seconds. Performing what-if scenarios, such as forcing portfolio units in or out and comparing portfolios, are fast operations that can be performed in a few seconds. Building the incremental-value report on a given portfolio and the waterfall report between two portfolios both take about 30 seconds to generate. These short turnaround times are vital to maintain the momentum of a decision meeting among senior Intel decision makers.

2.7 Business Results

The funding process that Intel had used for years prior to deploying Voyager relied on data-driven advocacy. Leaders of individual business segments assembled the case for each of their projects based on market projections for the resulting product(s) and estimates of engineering and manufacturing costs. Many of the projects in their business segments were in some stage of execution; others were new. The projects were put in a priority order based on expected cost and return or perceived strategic importance of the resulting products. The lists from each individual segment were passed up to the next level of management for discussion and merged into a master list of projects for the larger business group. Dollars were allocated to projects starting at the top of the master list until the budget was exhausted and the cut line drawn. Although projects well above (below) the cut line were considered in (out) of the plan of record, debates continued for projects just below (above) the cut line, sometimes resulting in additional budget requests up to the next level of management. In late 2011, the vice president of an individual business segment in Intel's Data Center Group (DCG) first used our decision support tool as an aid to the standard advocacy process. DCG's senior vice president was intrigued by the approach and mandated its trial use as an advocacy aid by a number of (but not all) DCG business segments for a plan cycle in mid-2012. The positive feedback from this trial lead to a defining experiment, including all DCG business segments, in late

2012. The advocacy process was executed without the aid of our tool. Using the same input data, a set of project portfolios was built using our tool without the aid of advocacy. Direct comparison was achieved by forcing our tool to reproduce the same product roadmap that resulted from the standard advocacy process. The small set of project portfolios produced by our approach exhibited an overall eNPV roughly 10 percent higher than the project portfolio produced by the standard advocacy process. Furthermore, because we could directly compare portfolios in our system, we avoided debate that might normally have included discussions of many projects. It was relatively easy to see that our approach identified three large projects that were being overfunded and six smaller projects that were being underfunded or prevented from starting in the advocacy-generated portfolio. Studies of our reports provided decision makers with a broader and deeper analysis than they had previously been able to access. The use of our model enabled them to satisfactorily address a number of their data-related questions. An evaluation of a number of intuition-based what-if analyses demonstrated the soundness of our recommendations. In only a few days, the managers of the three large projects, which we had questioned, were able to provide options for sufficiently trimming their budgets to fund the six smaller projects. Somewhat surprisingly, the managers who were requested to give back funding did so willingly once they were shown the rationale supported by the overall analysis. (They had been engaged early in the process to supply data, but only for their own projects.) Based on these positive business results and the transparency and speed with which the results were realized, for the past two years, DCG has adopted our approach as the principle project portfolio planning tool. Although we continue to improve data-collection activities to provide input to the system and incrementally refine and expand the set of output reports for the decision makers, the core mathematics have been robust, requiring only minor modifications to cope with new business scenarios.

Perhaps the most important contribution of our decision support system has been to show the highest levels Intel management that the appropriate give-and-take interaction between analytics and intuition produces higher-quality business solutions than either approach can produce individually. The resulting plan is superior in terms of NPV return for the budget dollars invested; in addition, decision makers engage in higher-quality debate (their feedback to the Voyager team) and arrive at consensus in a much shorter time (as measured). Given a receptive but tough-minded group of decision makers, analytics can inform intuition and intuition can inform analytics to the benefit of the business. Review of the DCG results at the corporate level has exposed other product groups and their senior management and finance personnel to the Voyager tools and processes. The combination of improved results and a transparent, fast process has caught the attention of many Intel decision makers. Voyager is currently under preliminary evaluation for use in Intel's largest and oldest product group, the Personal Computer and Client Group (PCCG), as well as Intel's newest product groups, the Mobile Communications Group (MCG) and the Internet of Things Group (IoTG).

Chapter 3

GENERALIZING THE FRAMEWORK AND SOLVING LARGE PROJECT PORTFOLIO SELECTION PROBLEM INSTANCES

Abstract: Project Portfolio Selection (PPS) is a complex problem faced by major companies whenever multiple funding opportunities present themselves with insufficient budget to fund them all. PPS is a difficult problem for a number of reasons including interactions between the projects being considered causing them to either reduce or enhance the benefit contribution or spending of other projects in the portfolio. In the work presented by Sampath *et al.* (2015) the authors present a quantitative approach to tackle PPS that involves generating *a priori* all the different versions of each projects based on various possible combinations with other projects to be included in a portfolio. However, as the number of projects and relationships between them increase, this approach is unable to scale efficiently. This is a common problem with optimization based approaches to PPS. Here we present an interactive modeling framework to describe the PPS problem that employs a specialized heuristic to seed a genetic algorithm and generate an efficient frontier of desirable portfolios. We call this beam search informed genetic algorithm BIG, and address the key scalability issue that arises in the course of solving the problem and generate attractive solutions within reasonable times. We also discuss how the framework can enable intuitive decision strategies to choose portfolios that best satisfy business goals.

Key words: Project Portfolio Selection, Decision Engineering, Meta-heuristics, genetic algorithm seeding, intuitive decision frameworks

3.1 Introduction

Project portfolio selection (PPS) is a problem faced by many companies that have to survive in the global marketplace by developing new technologies, products and solutions. Considering the multitude of options for projects, this periodic process of identifying the right portfolio in which to invest funds is an extremely complex undertaking. One source of complexity is the set of inter-dependencies that may exist between the projects. For example, projects may yield different benefits and consume different amounts of resources depending on the other projects that the company is undertaking at any given time period. Moreover, projects within the same portfolio are often in different stages of completion and the information for some projects may be known with higher certainty than that for others. Additional challenges often arise from the array of competing objectives considered by the different parties involved in the portfolio decision making. Beyond these challenges, the solutions recommended by a PPS framework must be defensible to senior decision makers. Unrealistic assumptions that result in undesirable alternatives are often identified and rejected by decision makers, who, over the years, may have developed deep intuition on the markets and technologies involved in developing these projects. Thus, any framework that seeks to address the problem successfully must provide a fast and effective way to allow interaction between the intuition of senior decision makers and the analytics resulting from advanced mathematical models.

Quantitative optimization models deal with many important aspects of the PPS problem that other approaches often do not, and are preferred whenever they are implementable. This is because quantitative optimization models allow for the incorporation of standard company financial measures as well as allowing decision makers to perform spending-benefit comparisons between various projects. However, several challenges have to be addressed when considering optimization models for PPS, including access to data and development of a practical solution framework that is flexible enough to incorporate the decision makers' intuition. One such framework that identifies attractive portfolios based on multiple conflicting objectives has been successfully implemented at Intel Corporation, and is discussed in detail in the work presented by Sampath *et al.* (2015). In this paper, the authors present a combination of modeling, simulation, and optimization techniques to optimize the project portfolio while providing a rich set of analysis and visualization tools for management to use in exploring the suggested portfolios and applying their intuition to make the final selection. The advantage of this decision framework is that as the underlying structure of the business itself changes, the optimization methodology can also be changed without affecting the overall framework for decision makers. The model discussed in that paper is a first step towards a more holistic approach to solving the problem; however, it is restrictive in two ways: (i) it presents ways to represent the projects and characterize the relationships between them that are specific to a particular optimization model, and (ii) it fails to scale well with the increase in the number of interactions a project has with other projects.

The contributions of this paper, then, are to present a more generalized modeling construct to incorporate modeling organization dynamics while providing a quick way to solve this problem for large problem instances to generate a Pareto-optimal front or *efficient frontier*. To ensure usability of the results, the paper also includes a discussion on the various steps that help decision makers to parse the efficient frontier and select a desirable portfolio, drawing from their real life experiences. The work is significant because it provides a successfully implemented PPS modeling framework for a large company that scales well with increasing interactions between projects.

The rest of the paper will discuss a brief literature review, explain the problem and the role of various agents involved and then present details on a versatile and intuitive way to model an optimization framework for the PPS problem. Details regarding a genetic algorithm that can be modified to handle a wide variety of PPS problems using the modeling framework we propose are then presented. Lastly, we discuss the results of numerous experiments examining the efficacy of the genetic algorithm after a brief description on how to process the results of the optimization model to help better incorporate intuition of the decision maker in choosing the most appropriate portfolio.

3.2 Literature Review

Most successful frameworks that tackle the PPS problem are divided into stages as discussed by Archer and Ghasemzadeh (1999). This involves (i) pre-screening projects for viable funding alternatives, (ii) benefit measurement of the individual projects and (iii) project selection. Chien (2002) notes that financial analysis methods that use some form of quantitative financial measure as a criterion are used by most firms for the first two stages. It would follow that an optimization model to aid in selecting a portfolio using these quantitative results would seem natural. The advantage of using optimization models for solving PPS problems, as described by Archer and Ghasemzadeh (2004), is that such models tend to structure the problem in a very intuitive manner that is easily understood by decision makers while effectively handling inter-dependencies between projects and allowing easy analysis of changing the supply of available resources on the portfolio. Despite these advantages, these methods have largely not been adopted because they do not lend themselves well to handling increasing model complexity, as well as incorporating qualitative measures, as discussed in the work of Wu *et al.* (2017). The overall decision making framework should engage decision makers with the process and present them with multiple options, allowing them to apply their intuition for a successful implementation of the solution in a corporate setting. Recent efforts at improving the visualizations for a single-criterion optimization problem to increase its chances of being adopted by an organization are discussed in da Silva *et al.* (2017).

Quantitative PPS optimization problems are usually modeled as knapsack problems (KP), where a subset of items must be selected from a set of items each with a benefit and spending profile, maximizing the overall benefit of the selection while not exceeding the total available amount of resources. When attempting to find an appropriate solution for the purpose of solving knapsack problems, one is faced with a problem of choosing from a large number of different solutions, each of which is suited to a different requirement (Martello and Toth (1990)). There are many different styles in which a KP can present itself. The two most relevant for PPS problems are the "Multi Dimensional KP"(MDK) (where one faces the problem of the data spanning multiple dimensions Bas (2012), Sakawa *et al.* (2001), Stewart (2016), Lifshits and Avdoshin (2016) and Fonseca *et al.* (2015)) and the "Multiple Choice KP"(MCK) (where it is mandatory to select one element from each of the multiple groupings of items present Tavana *et al.* (2013), Razi *et al.* (2015), Kuchta (2002), Altuntas and Dereli (2015) and Gafarov *et al.* (2016)). Some of the problems discussed in the works cited above present characteristics of both of the above types, and are referred to as "Multiple Choice Multidimensional KPs"(MMKP). In MMKPs, the condition where one needs to definitely choose an element from each grouping is not strongly enforced. This allows the possibility of omitting an item from a grouping if it does not add to the optimality of the solution.

In general, PPS problems are made harder by the fact that there may be non-linear effects on the benefit or spending of the individual elements based on the inclusion or exclusion of other items in the selection. Capturing these inter-dependencies and relationships between the projects requires a systematic modeling framework and a solution methodology that supports this complexity. This difficulty is illustrated in the work presented by Li *et al.* (2010), for selecting a green technology portfolio from an environmental strategy perspective, where the objective is to maximize the economic and environmental benefits of an enterprise. The authors use an integer-programming model to maximize the synergies between various technologies. The model, however, becomes laborious while trying to explicitly address more than six ‘synergistic components,’ and faces scalability issues in terms of data capture as well as solution times.

To avoid these issues, we present a generalized modeling framework that considers the interactions between the project alternatives being considered and that can be extended to an exponentially large number of relationships. With regard to the solution methodology, KPs with even a few hundred thousand variables often quickly become intractable to solve to optimality by traditional commercial integer-programming solvers with increasing problem instance size. A discussion can be found in Pisinger (2005). This necessitates the adoption of a heuristic to solve the problem. In the study by Gomede and de Barros (2014), the authors note that while the literature has seen a steady increase in the number of PPS papers, it has not seen a proportional rise in the use of heuristics to solve the PPS problem. This paper is a contribution towards this gap in the literature.

An added element of difficulty in choosing an appropriate solution methodology is the fact that a single solution to the portfolio optimization problem is never enough during real-life decision making for PPS. Decision makers often demand multiple solutions to choose from so that they may explore and contrast them from the viewpoint of many complex, and often conflicting objectives. One way to visualize and present multiple solutions to a decision maker is the efficient frontier. As discussed in the work of Levine (2007), the efficient frontier remains one of the best tools in the decision maker’s toolkit while selecting an optimal portfolio. It provides an easy and intuitive way to visualize the decision space with the corresponding spending of the portfolio on the horizontal axis, and the vertical axis displaying the value of the project quantified with respect to the expected revenue based on the impact and alignment with business drivers. Our experience has shown the efficacy of quickly comparing and contrasting multiple solutions and was one of the primary reasons for the ready acceptance of the framework by all parties involved in the PPS process.

Liu and Wang (2015) note that genetic algorithms lend themselves very well to producing such a set of solutions in addition to lending themselves well to parallelization. This is important with recent trends towards the availability of more parallel processing computational power, be it "scale-up" or "scale-out." In addition, genetic algorithms like the Non-dominated Sorting Genetic Algorithm (NSGA) family of algorithms developed by Deb *et al.* (2002) remain some of the best in terms of generating multiple solutions quickly to form efficient frontiers Yuan *et al.* (2014). This leads to the preference of genetic algorithms over other meta-heuristic algorithms like tabu search and simulated annealing. One problem, however, with evolutionary algorithms like NSGA is the poor algorithm efficiency measured in terms of the total number of evaluations required for convergence to the true optimal frontier due to getting stuck on local optimal solutions Sindhya *et al.* (2011). Another cited problem is the diversity of solutions with a view towards obtaining a wide variety of options for decision makers to choose from Vachhani *et al.* (2016).

In this study, we present BIG, a genetic algorithm that uses a specialized heuristic based on a beam search-like technique to improve convergence and another heuristic to replace NSGA's traditional method of maintaining diversity through its crowding distance operator while performing fewer operations, and thus, having better worst-case complexity for the problem at hand. This is due to the fact that instead of a "cold start," our genetic algorithm is seeded with a set of solutions to start with and improve upon rather than starting from scratch, and hence converges to the true efficient frontier faster Saavedra-Moreno *et al.* (2011); Grosan and Abraham (2007); Paul *et al.* (2015). Elitism is achieved by tracking how close solutions are to the first front of nondominated solutions, and how many generations a solution has been retained to try and fill in large gaps in the efficient frontier.

Generating a large number (if not all) of nondominated solutions and presenting them on a two-dimensional efficient frontier, however, is not enough. We still need to allow the decision maker to switch between the various portfolios on the basis of different qualitative and quantitative metrics and considerations. One could generate and visualize multiple efficient frontiers based on the other quantitative and qualitative metrics. However, these are difficult to communicate, and furthermore, it is often difficult to prioritize objectives especially if there are many different decision makers each with different preference functions. One way to tackle this problem is to try and measure the preference function of the decision maker and generate an appropriate solution tailored to this estimate. This approach is discussed in Fowler *et al.* (2010), where the authors describe an interactive methodology to achieve multi-objective optimization by estimating and using a linear utility function to order solutions. A feedback loop is established and the population order is corrected if it is not in accordance with the decision maker's expressed preferences. This methodology is able to handle any quasi-concave objective function and presents a genetic algorithmapproach to solve for the decision maker's subjective preferences.

A similar methodology is presented by Yu *et al.* (2012), where the authors attempt to solve a 0-1 knapsack PPS with a genetic algorithm in conjunction with linear programming methods, but collect decision maker preferences before optimization and incorporate it into their model. In the study presented by Amid *et al.* (2009), a "fuzzy objective" is discussed where a fuzzy measure of the decision maker's risk preferences is obtained by assigning a degree of satisfaction between criteria and constraints, and defining tolerances for constraints in the interest of obtaining the goal value in the objective risk function. Such *compensatory* decision strategies of choosing between portfolios, where the various attributes are amalgamated, and good values of one attribute can make up for deficiencies in other attributes are quite popular in the literature. However, one of the major problems with these kinds of approaches is that they are both time consuming and often confusing for decision makers to understand because of the large time gap between giving their preferences in the benefit measurement stage and the actual optimization or project selection stage, which typically occurs much later. The decision makers' preference information may not be static and change over time, as new market and technological information becomes available. Furthermore, optimization models designed with certain individuals' preference functions may not accurately reflect current decision makers' preferences, since decision makers may rotate in and out of departments, or leave the company during this interval. Finally, such approaches are also not very transparent for the decision maker as it may not be clear why some portfolios are being preferred over other ones.

We find that even if the above problems were tackled somehow by weighting the various decision makers' preferences correctly, and adjusting for time and new information, a decision maker's time is extremely expensive and gets progressively more expensive the higher up one goes along the chain of management. Hence, measuring the decision makers' preferences to incorporate into the optimization problem is often prohibitively expensive in the first place. Considering that the possible outcomes of this exercise are often the funding or defunding of large scale projects, it is extremely common for final calls to be made by the highest levels of management. While a project is in its inception, gaining an audience with senior decision makers may be difficult unless there is a strong champion to make the case.

For these reasons, we employ a non-compensatory method for decision making. In a non-compensatory method, instead of combining different attributes, a decision maker may drop an option if it does not meet a cut-off on one attribute, even if it has desirable values for other attributes. These kinds of strategies describe more accurately how people make decisions in real life, and are therefore, more likely to get various members in the decision making pipeline to buy in to the process, as it allows them to bring their intuition to the analytics in a holistic manner to arrive at a decision.

3.3 Modeling Framework

In this section, we illustrate various concepts we will need for the remaining sections of the paper. The key requirements of this framework are two-fold: (i) to define a framework that easily allows the representation of interaction effects between various projects under consideration,and (ii) ensuring that the problem can be modeled easily as a mathematical program. If it is possible to formulate the actual underlying business problem as a 0-1 knapsack problem, solving the problem for an arbitrary objective function yields a feasible portfolio.

We present the modeling and solution of one particular problem instance in the appendix of this paper. We call the fundamental unit in our modeling framework a *portfolio unit*, which may represent any project or product that the company is considering to fund and develop. Portfolio units may have any number of relevant attribute metrics or attributes that pertain to the business such as volume of units projected to be sold, net present value expected to be realized by executing the portfolio unit, the spending associated with the design and engineering of the portfolio unit, and so on. For the set N of portfolio units, binary decision variable $x_i, i \in N$ indicates whether portfolio unit i was selected or not. We refer to the model presented here as the Portfolio Selection Integer Programming Model (PSIPM).

Relationship constraints describe the inclusion and exclusion states of each portfolio unit with respect to each other in a feasible portfolio. Resource constraints define bounds on the amounts of various resources that can be consumed by a feasible portfolio. Qualitative constraints typically describe rules detailing the composition of the portfolio based on some qualitative portfolio unit attribute. For example, only portfolios that have at least a third of their spending allocated to early stage R&D projects may be desired by management. Alternatively, the portfolio selection exercise may call for limiting portfolios to the set that allot only a certain amount of headcount to projects nearing completion. These can be attained by modeling constraints in PSIPM using qualitative portfolio unit attributes. If the value of the portfolio depends on the interaction effects between various portfolio units and all the values a portfolio unit may realize depending on which relationships are active in the portfolio are too numerous to generate, then exact methods to solve PSIPM do not scale well with problem instance size due to an increase in the number of variables and constraints. This becomes apparent as we continue with the description of our model framework and will be discussed in detail later.

A *portfolio* is a combination of portfolio units. *Relationships* are constructs that are an attempt to model business rules that define a feasible portfolio. They contain information regarding the interaction between two or more portfolio units in a feasible portfolio, although it should be noted that feasibility is not determined only by relationships but also through the qualitative and quantitative constraints discussed earlier. Through a relationship, a portfolio unit may *impact* another portfolio unit. The impact is a modeling construct that contains information regarding the change in the value of some attribute metric of the portfolio unit being impacted when the portfolio unit that impacts it is also selected within the same portfolio. We find in our experience that when each of these constructs are represented visually such as in Figure 3.1, they provide a more intuitive understanding of the business as a whole.

When one portfolio unit impacts another, the unit being impacted may now find itself having multiple values for some attribute metric, e.g., one when the portfolio unit impacting it is also executed in the same portfolio, and one where the impacting portfolio unit is not. We call these different versions of the portfolio unit that may have different values of some attribute metric based on the presence or absence of other portfolio units in the portfolio, a *decision unit.* To better express the concept of a decision unit, we define a portfolio unit's *neighborhood.* Each portfolio unit's neighborhood includes all portfolio units that, through a relationship, affect the value (of some attribute) of that specific portfolio unit. A unique combination of the elements of a neighborhood determines a *scenario.* Each scenario corresponds to a different decision unit. To "valuate" a decision unit, we first determine its corresponding scenario. This allows us to identify all the impacts affecting the portfolio unit. Evaluating all the impacts yields the decision unit's valuation. A valuation is the end result of some function of the input attributes of a portfolio unit and its impacts. It represents a quantitative value that we can use in some tangible way for decision making. Portfolios may have the same attribute metrics as portfolio units. The value of a portfolio's benefit or spending metric is some function of the valuations of the decision units of the corresponding portfolio units that constitute the portfolio. Portfolio units (and by extension, portfolios) have a primary benefit metric (usually NPV) and a primary spending metric (usually engineering spending over the planning horizon.

Even for very complex business cases, we find that a few key relationships listed below are sufficient to model the vast majority of portfolio unit interactions in practice.

1. When one portfolio unit is *required* by another, it cannot be funded or executed without that unit. This relationship is usually used to model cases where a higher or more complex technology requires some lower or basic technology, without which it cannot be built.

2. When one portfolio unit is *optional* to another, it affects the value of the other portfolio unit through an *impact*. For example, a marketing effort is not fundamental to building a product, but it may greatly affect the number of units sold. It is thus optional to the product, but impacts it by increasing the sales.

3. *At-Most-Or* relationships between a group of portfolio units indicate that at most a subset of them can exist in any one portfolio. This is usually used to model cases where portfolio units with similar functionality exist. For example, a feasible portfolio may include at most one of the portfolio units in a At-Most-Or relationship. One business example would be three variants of a new product, each with a different feature set. If management has decided that only one of these variants can be funded at most, but that all three may be eschewed for other opportunities, then the At-Most-Or relationship is used to ensure that at most one of these portfolio units is selected.

4. A *Hard-Or* relationship between a group of portfolio units, on the other hand, describes the situation where a certain number of portfolio units must be selected from a group of portfolio units. A feasible portfolio must include the specified number of the portfolio units that share the Hard-Or relationship. In the context of semiconductor manufacturing, for example, this may involve choosing between various versions of a bread-and-butter technology offering. If there exists a version that can be brought to market in two years versus one that can be brought to market in three years, then management must decide on which one to execute. If the technology has been promised to an external partner, then not choosing one of the variants is not an option.

5. A similar concept to the Hard-Or relationship can occur when a feasible portfolio must execute at least a certain number of portfolio units from a given a set. For

example, when from a group of five technology projects, three or more must be executed, we say that that the five technology projects share an *At-Least-Or* relationship.

6. When a group of portfolio units cannot exist without each other, they are said to share an *Hard-And* relationship. This is used to model portfolio units that typically have no value without the other, like a microprocessor and its corresponding chipset. It is different from the relationship defined by the required dependency sets because portfolio units that share this relationship cannot be executed alone without the others in that dependency set.

7. An *At-Least-And* is similar. In this case, at least a certain number of portfolio units must be selected from a group if any portfolio units from the group are selected at all.

We define the concept of a *dependency set* to represent relationships. These dependency sets can be used to set up the relationship constraints in PSIPM. There are two types of dependency sets that we have encountered most frequently in our experience. The first is a *required* dependency set, denoted by Δ^R. Each element of a required dependency set consists of an ordered pair. The first element of the pair is a set of portfolio units and the second element of the pair is a single portfolio unit. To generate a feasible portfolio containing the portfolio unit in the second element of the ordered pair, at least one of the portfolio units in the first element must also be selected. This can be converted into a constraint in PSIPM and is shown in Table 3.1.

The second type of dependency set will be represented by the notation ${}_d\Delta^Z$ where the superscript Z of the set indicates what type of dependency set it is (At-Most-Or, Hard-Or and so on) and the subscript d indicates a parameter of the relationship the

Dependency Set	Set Element	Constraint
Required	$(\delta_{l1}^R, \delta_{l2}^R) : (\delta_{l1}^R, \delta_{l2}^R) \in \Delta^R$	$\sum_{i \in \delta_{l1}^R} x_i \geq x_{i \in \delta_{l2}^R}$

Table 3.1: Translating Required Dependency Sets to Constraints

dependency set represents. This is presented in Table 3.2. For the And and At-Least-And constraints, we define indicator variables w_{ld}^A, and w_{ld}^G that indicate whether the dependencies shared by the elements of sets ${}_d\delta_l^A$ and ${}_d\delta_l^G$ are satisfied by the portfolio with the corresponding portfolio units selected.

Dependency Set	Set Element	Constraint
At-Most-Or	${}_d\delta_l^S : \quad {}_d\delta_l^S \in \Delta^S$	$\sum_{\{i \in {}_d\delta_l^S\}} x_i \leq d$
Hard-Or	${}_d\delta_l^H : \quad {}_d\delta_l^H \in \Delta^H$	$\sum_{\{i \in {}_d\delta_l^H\}} x_i = d$
At-Least-Or	${}_d\delta_l^T : \quad {}_d\delta_l^T \in \Delta^T$	$\sum_{\{i \in {}_d\delta_l^T\}} x_i \geq d$
Hard-And	${}_d\delta_l^A : \quad {}_d\delta_l^A \in \Delta^A$	$\sum_{\{i \in \delta_l^A\}} x_i = dw_{ld}^A, \quad w_{ld}^A \in \{0, 1\}$
At-Least-And	${}_d\delta_l^G : \quad {}_d\delta_l^G \in \Delta^G$	$\sum_{\{i \in {}_d\delta_l^G\}} x_i \geq dw_{ld}^G, \quad w_{ld}^G \in \{0, 1\}$ $\sum_{\{i \in {}_d\delta_l^G\}} x_i \leq \lvert {}_d\delta_l^G \rvert w_{ld}^G$

Table 3.2: Translating Dependency Sets to Constraints

Figure 3.1 shows an example with some of the possible relationships that can be modeled with this framework. The neighborhood of portfolio unit A is shown by the dotted line. If we look at portfolio unit A, we see that it shares a required relationship that allows decision makers to model the business case where either one of portfolio

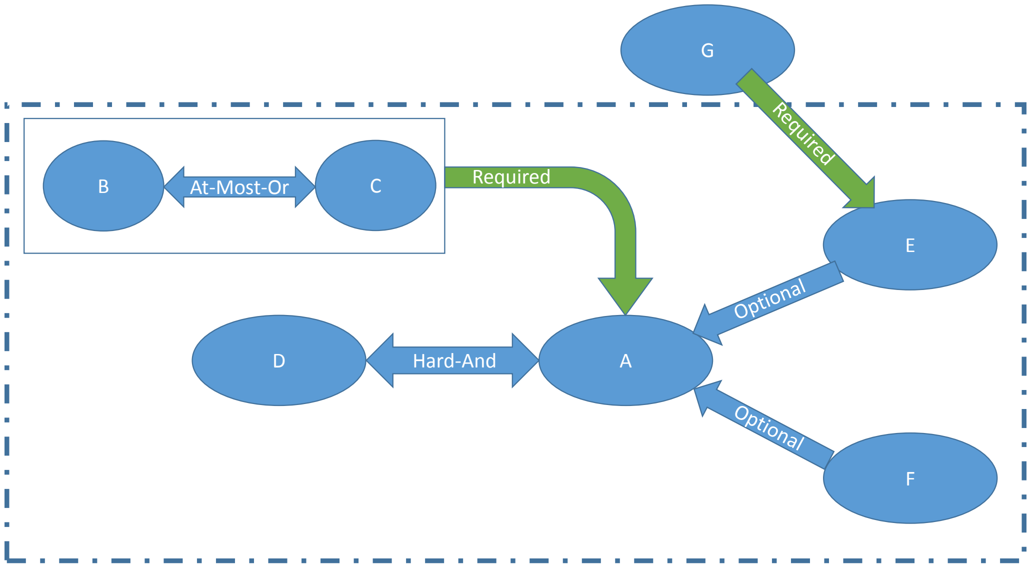


Figure 3.1: Example of a Portfolio Unit and Its Neighborhood. Here Portfolio Unit A's Neighborhood Is Denoted by the Dotted Line

units B or C, must be selected to execute A. In this case, an At-Most-Or dependency set exists between portfolio units B and C, which implies only one of these can be present in a feasible portfolio. A feasible portfolio that contains portfolio unit A must then contain either portfolio unit B or C. Portfolio unit A must be executed in all cases alongside portfolio unit D. Portfolio units E and F impact the value of A in some way. Portfolio unit A's value contribution to the overall value of a portfolio may thus change depending on the combination of E and F units that are also picked in any given portfolio. There are a total of eight decision units that arise for portfolio unit A in this example. The associated scenarios with each of the eight decision units and the value of portfolio unit A in each of those cases is detailed in Table 3.3. Portfolio unit G is not in portfolio unit A's neighborhood. It does not directly impact A's value, although it is required for executing portfolio unit E. It does not, however, need to be considered when enumerating the decision units of portfolio unit A.

Decision Unit	B (In/Out)	C (In/Out)	D (In/Out)	E (In/Out)	F (In/Out)	Valuation of arbitrary metric of A
1	1	0	1	1	1	100
2	1	0	1	1	0	200
3	1	0	1	0	1	149
4	1	0	1	0	0	322
5	0	1	1	1	1	312
6	0	1	1	1	0	112
7	0	1	1	0	1	199
8	0	1	1	0	0	121

Table 3.3: Various Decision Units That Exist for Portfolio Unit A in Figure 3.1. A Value of *1* Signifies the Presence of That Portfolio Unit in the Scenario, and a Value of *0* Signifies the Absence. The Last Column Reports the Value of Portfolio Unit A in That Scenario

Generating all of the decision units for a particular portfolio unit with t other portfolio units that are optional to it, has a worst-case complexity of $O(2^t)$. It must be noted that after generating each decision unit, each must be valuated to derive all of the corresponding quantitative metrics of interest in the decision making process. Thus, even for moderate values of t it may not be practical to generate all of the decision units, valuate them, and then pass them off to a complex integer program that must be repeatedly solved to generate an efficient frontier of portfolios. The last column in Table 3.3 denotes the valuation of some metric of portfolio unit A for that particular scenario. The concept of a decision unit, therefore, is to explain interaction effects between different portfolio units. Hence, with increasing model complexity, due to increasing number of interactions, we need a solution methodology that does not depend on a complete enumeration of all decision units as well as their valuation, prior to setting up and solving a large-scale integer program.

Evaluations depend on the "existence" of other projects, not the values derived from them, or costs incurred to perform them. This is why it is sufficient to only consider the immediate neighborhood of the project when valuating it. The information for the impact resides on the relationship itself and is separate from the portfolio unit it originates from. Thus each decision unit's (of a given portfolio unit) values are determined merely by the presence or absence of other portfolio units in its neighborhood and not by the actual underlying values of those other portfolio units.

One way to achieve that is by not generating all of the decision units beforehand but determining them on-the-fly for a given portfolio and then computing the value of the portfolio at a point where we know all of the portfolio units included in the current solution. In contrast, a traditional integer program requires all decision units to be identified and valuated *apriori*. We can calculate the benefits and spending metrics of a portfolio by determining the corresponding decision units that were realized for each portfolio unit, calculating the value for each decision unit, and then computing the value of the portfolio using some appropriate function.

In Sampath *et al.* (2015), the authors generate an efficient frontier by first enumerating all decision units, valuating them, and then passing the results of the valuation step to an algorithm that employs an exact method to generate the efficient frontier of portfolios. If the number of decision units exceeds a certain threshold, this approach of "Enumerate, Valuate and Solve" (EVS) may not be computationally tractable.

Depending on how the financial model valuates a portfolio, the value of a metric for a portfolio may be more or less than the sum of the values of the decision units that comprise it, and this could add to the difficulty of the problem. For example, many high value projects may all operate in the same market and hence may cannibalize

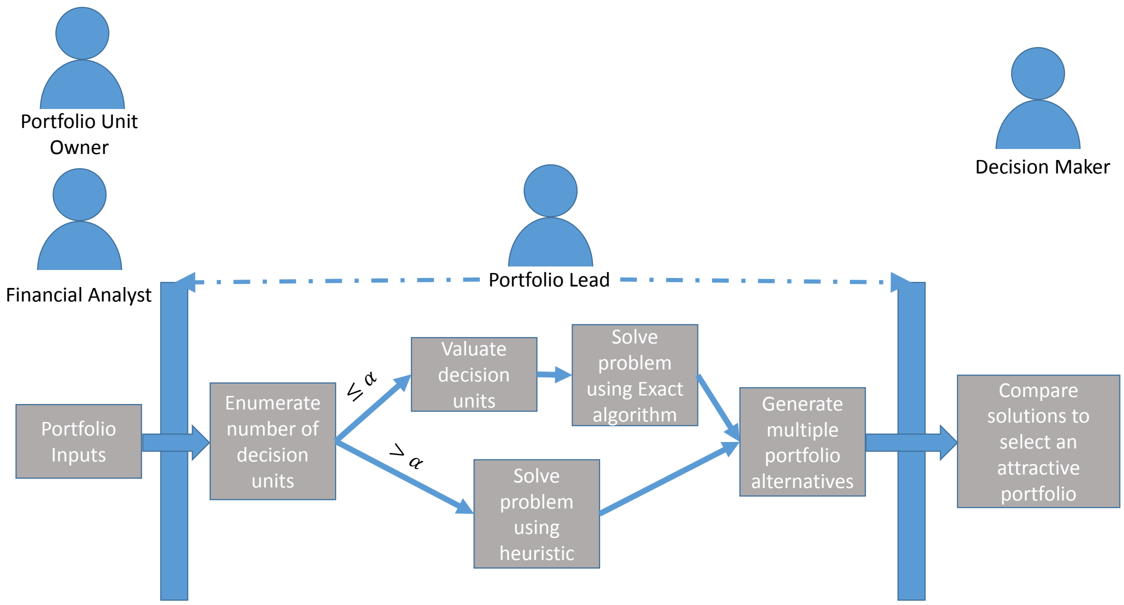


Figure 3.2: Various Parties Involved in the Portfolio Selection Process

each others benefit. These portfolio units may not share relationships with each other directly, but the financial model may cap the aggregate of their contributions. The value of the portfolio as a whole may then be less than the sum of the values of individual decision units. Such cases sometimes arise when there are a number of portfolio units that have a high probability of success that all operate in the same market segment. Another example of this would be a penalty component that the financial model ,might add to the individual values of the decision unit. An instance of such a case would arise if while valuating the NPV of a portfolio, the manufacturing costs or a tax component were modeled separately outside of the individual portfolio unit for reporting purposes as decision makers may only be interested in the pure value contribution of a portfolio unit.

More discussion into the financial model used to valuate portfolios and portfolio units is beyond the scope of this paper, but as evidenced by this discussion, it plays a huge role in the process. The framework itself, it should be noted, is agnostic to the nature of the valuation model. However the fact that the value of the portfolio may not be a linear combination of the values of the individual portfolio units that constitute it, further makes the EVS approach undesirable. Therefore, another approach must be taken to generate the efficient frontier.

Figure 3.2 shows the various parties involved in the portfolio selection process we have described above. *Portfolio Unit Owners* are typically project leads, or senior management, and are generally the people most vested in decisions regarding the funding or defunding of a particular project. *Financial Analysts*, who report to Portfolio Unit Owners, are responsible for capturing input information regarding the portfolio unit. Once these inputs are captured, a *Data Scientist*, who is often agnostic to the data itself, oversees the algorithm and the support of the decision support system (DSS) for the input as well as the output visualizations of the results. *Decision Makers* are usually senior managers, Vice Presidents or the Chief Executive Officer of the company, and finally decide on what projects to fund. They often have the most intuition to bring to the process, and through interactions with the optimization process, decide on the best portfolio. A *Portfolio Lead* oversees the entire process and the flow of data and information from each stage and is responsible for developing the list of portfolio units to be considered by the PPS exercise and what outputs to present to decision makers to aid them in the decision making process. Portfolio Leads are responsible for running the entire PPS process and are the ones who request optimization runs and collate results to present to decision makers.

From our experience in building a decision support system (DSS) at Intel Corporation to support the PPS process, we find that visualizing the business in the manner discussed in this section generates focused conversations on how the business is run and helps to quickly reach a consensus on how the business is modeled that can easily be communicated with all levels of management. Through a DSS, individual analysts can enter input information for their relevant portfolio units and all relevant neighborhood items, the impact of optional portfolio units and other information. Specifying impacts in a meaningful manner depends on the underlying financial model used to valuate the portfolio units and is beyond the scope of this paper. Typically, a Portfolio Lead is tasked with maintaining the overall visual of how portfolio units interact with each other and performing data quality checks and vetting the assumptions of the individual analysts. Decision Makers typically prefer exploring the solution space, comparing and contrasting different portfolio alternatives using an efficient frontier of portfolios that displays the best portfolio alternative at various spending levels. As stated in the study presented by Gruia (2003), efficient frontiers are especially useful for clarifying basic questions on investment efficiency, trade-offs between greater benefits and increasing spending, the cost of breaking various constraints and avoiding investments in regions of diminishing returns.

It should be noted that PPS optimization models of the kind described here typically avoid sunk cost biases. As such projects that possess momentum from having significant engineering efforts dedicated to them will invariably look more attractive than newer projects whose future seems uncertain and risky. Senior decision makers can weigh in on allotting a certain amount of budget towards new projects. By running the model with up-to-date information throughout the year, progress can be tracked when optimal portfolios from the current run are compared to optimal portfolios from the last PPS exercise. Senior decision makers can weigh in and kill projects they feel are not delivering and contributing to the optimal roadmap going forward. However, in the cases where there may exist a real sunk-cost of the kind that cannot be ignored, such as a contract with another company for example that, if reneged upon, involves a penalty that cannot be avoided, the cost of stopping or exiting a project must be factored in. By analogy, this means that not choosing an item in the knapsack problem still reduces the capacity of the knapsack by a certain amount. Not selecting a portfolio unit to be included in the portfolio thus carries a real quantifiable impact that must be factored into the optimization model. This can be achieved by modeling the case where a portfolio unit is not selected, as it's own separate portfolio unit, and then modeling a *Hard-Or* relationship between the original portfolio unit and the new one with the condition that only one of the two can be selected and every feasible portfolio must contain one or the other. For a portfolio unit A that models the cost and benefit of including a certain portfolio unit in the portfolio, portfolio unit A' would model the cost and benefit of not including the portfolio unit in any feasible portfolio. Each feasible portfolio would contain one or the other.

The rest of this paper provides further details on the solution methodology we used to generate an efficient frontier of portfolios to present to Decision Makers. Identifying this efficient frontier is difficult given the black-box nature of either the objective function, the spending function, or both. It should be noted that this does not prevent us from modeling the portfolio units and only their relationships as an integer program and solving for a zero objective to generate feasible solutions.

3.4 Solution Methodology

In this section we describe the details of BIG. While genetic algorithms lend themselves well to solving the kind of knapsack problems that PPS problems are usually modeled as Zhang *et al.* (2011); Yu *et al.* (2012); Cruz-Reyes *et al.* (2017); Abbasianjahromi *et al.* (2016). While there are some common features that most genetic algorithms share, they must be adapted to suit the specific problem structure at hand. The NSGA-II multiobjective evolutionary algorithm is one such algorithm for quickly identifying hierarchical dominance frontiers for multiple objectives. However, for our methodology we only need to generate an efficient frontier based on two metrics, a primary spending and a primary benefit metric. This is because representing information in more than two dimensions is expensive to compute, difficult to communicate to senior decision makers, and is typically not represented well on their usual means of communication such as slide presentations.

Working on a parent population, the NSGA-II algorithm generates offspring similar to conventional evolutionary algorithms, using the usual operators of tournament selection, crossover, and mutation. By employing elitism, non-dominated solutions from each generation are retained until the final stage. NSGA-II in general divides solutions into different dominance frontiers while attempting to spread them out in the objective space using a concept called the crowding distance. This is the estimate of the distance of the end points of a cuboid whose vertices are the two nearest neighbors, with extreme solutions assumed to define a cuboid of infinite size. Elitism is achieved by retaining solutions with higher dominance frontier ranks and higher crowding distances.

While the NSGA-II is one of the better evolutionary algorithms in the literature for this kind of application, convergence and diversity are still issues that arise for a large number of problem instances, especially if the objective function is not smooth or the feasible region is not part of a convex set. If the black box nature of the objective function is computationally expensive, we are further motivated to avoid a situation of starting from scratch with no knowledge of good search directions. Thus, we attempt to seed the algorithm with good starting solutions to search from rather than starting from nothing. The concept of seeding is discussed in some detail in the work of Friedrich and Wagner (2015), who propose some seeding algorithms for the NSGA-II. However, no seeding algorithm has any real advantages over another for all problem instances, and we find that many are needlessly complex with no real computational payoff. Therefore, it is often more practical to choose a simple heuristic with low worst-case complexity such as a beam search that can be controlled effectively and is known to work well for single function minimization problems given the problem structure.

3.4.1 Identifying Attractive Initial Solutions

In choosing an appropriate algorithm to generate seed solutions for the genetic algorithm, we need to take into consideration the efficacy of the algorithm at solving knapsack problems, its complexity and how easy it is to retool it to produce a rough set of non-dominated solutions. Beam search is one such algorithm that satisfies all three criteria, as discussed by Jaszkiewicz (2004). The worst case complexity of the beam search algorithm depends on the beam width β, or the number of solutions retained at each level and the depth of the tree, which is the maximum number of one-step transitions to ending a branch. Thus, the worst case complexity can be controlled very easily by limiting the beam width, and this makes it an attractive seeding algorithm to run for the NSGA-II.

The beam search algorithm is a robust algorithm that provides good solutions for knapsack type problems Baldi *et al.* (2014); Hu and López-Ibáñez (2017). It is a modification of best-first search that creates a search tree, and then uses breadth-first search. It lends itself easily to providing multiple solutions (by sampling at various stages of the beam search) that are typically diverse in the spending of the portfolios generated, and can be sorted to provide a set of non-dominated solutions, which we can later use to seed our genetic algorithm. The fact that multiple solutions can be evaluated at every stage of the beam search also makes it attractive due to the opportunity for parallelization it offers when being implemented on a computing system with access to a parallel computing cluster.

Over the spending-benefit objective space, this can be done by one of two approaches. If the spending is not a black-box function and is known beforehand, we can start with the highest spending portfolio, and at every stage of the beam search, go down to a lower spending portfolio by some predefined rule, ending either when a predetermined number of solutions have been generated or no solution of lower spending can be generated. If the spending is a black-box function, then one can start with the portfolio with the highest number of portfolio units and work backwards to the portfolio with the least number of portfolio units.

At every stage of the beam search, solutions are evaluated, and only a specified number solutions are retained to continue on to the next level of the beam search and so on, until the stopping condition is met. At the end of this exercise, sorting the remaining portfolios and eliminating the dominated solutions provides us with a preliminary Pareto-optimal front that can be used to seed the genetic algorithm, leading to both faster convergence times and greater diversity in spending values of solutions generated than a so-called cold start. An example of the results of a specific implementation of the beam search for a particular PPS problem instance discussed in the appendix are presented in Figure 3.6a. Typically, the beam search itself does not produce a diverse efficient frontier, but without the initial population it provides, the genetic algorithm's performance is severely impacted and takes a long time to approach the true efficient frontier.

3.4.2 Genetic Algorithm Details

Crossover and mutation are both relatively straightforward steps and depend on the encoding scheme. While we mainly used random mutation operators and flat crossover operators, recently improvements in convergence rates have been achieved by using specialized operators for mutation and crossover Lim *et al.* (2017). Elitism and the stopping condition used to generate an efficient frontier as part of a successful PPS framework require more careful thought, however, and are discussed in more detail below. Another important operation to consider is error correction, especially when feasible solutions are hard to generate. It should be noted that for any generated offspring x', if A_r and b_r are the relationship coefficient matrices from setting up the PPS problem as an integer program, we can easily check if a given portfolio is feasible, i.e., a candidate portfolio x' generated by the genetic algorithm is feasible if and only if $A_r x' \leq b_r$. Moreover, by setting the objective decision vector of the integer program to match an arbitrary portfolio generated by the genetic algorithm and solving the integer-program, we can achieve error correction by finding the 'nearest neighbor' in terms of being closest to the generated candidate portfolio in portfolio unit composition. For example, assuming the genetic algorithm generates a solution x', then only modeling the relationship constraints while maximizing the objective $\sum_{\{i:x'_i=1\}} x_i - \sum_{\{i:x'_i=0\}} x_i$ in model M1 and solving will yield a feasible portfolio that will attempt to incorporate all the portfolio units included in x' with the minimal number of additional portfolio units needed to make the portfolio feasible.

The encoding scheme used for the genetic algorithm will depend on the structure of the problem. An example is discussed in the appendix. Genetic algorithms often tend to throw away a large number of generated chromosomes as infeasible when applied to KPs. The authors present some repair operators in Chih (2015). However, candidate repair is still generally considered expensive. Moreover, most fail if the problem reaches the level of complexity described in this paper. Feasible portfolios may be generated by solving Model M1 for a dummy objective but error correction using this method is only recommended when the number of feasible portfolios is low to improve the performance of the genetic algorithm. In general, it is more effective to discard an infeasible portfolio and generate a new one in its place, rather than try to correct it.

3.4.3 Retention of Elite Solutions

Once seeded, the NSGA-II algorithm converges to the true Pareto-optimal front much faster than it would have done from a cold start. However, this still does not guarantee obtaining a diverse set of solutions that span the whole spectrum of spending values with corresponding revenues. Qualitatively, our practice at Intel Corporation has shown that a smooth and well-populated efficient frontier was better received and inspired more confidence in the results than a jagged and sparse efficient frontier, even when resulting from relatively quick runs of approximately 10 minutes. It is entirely possible, however, that there are large gaps in the efficient frontier due to the underlying structure of the problem, the relationships, the input values of the portfolio unit metrics, and the impacts on each other. When generating an efficient frontier of solutions, the solutions retained after each generation greatly affect the quality of solutions in the next generation. Hence, the retention policy must identify gaps in the efficient frontier and work towards filling them up to provide viable portfolios for all spending levels while striving to maximize the benefit at every spending level. The goal here is to not have gaps in the efficient frontier. This is to ensure portfolio alternatives exist at every spending level for decision makers to choose from. Since we are only building an efficient frontier in two dimensions, this means that the crowding distance operator used in the NSGA-II algorithm may not be very effective at retaining solutions that may, in the future, fill up the defined gaps in the efficient frontier. This happens because elitism in the NSGA-II is achieved by first retaining numbers with a higher non-domination rank, and then, by crowding distance. This may lead to cases where dominated solutions may be retained in a region where the efficient frontier is already well-populated in terms of portfolios for that benefit level, but not for the resource level.

We address this problem by implementing the following steps. First, all non-dominated portfolios are automatically retained. For each dominated portfolio, a weighting function is applied to order the portfolios by their potential viability to produce children with better spending-benefit ratio. The weighting function is as follows:

i. the non-dominated portfolios are ordered by resource value

ii. each non-dominated portfolio has an "age" value, which increases by one for every generation the portfolio is included on the frontier.

iii. for every consecutive pair of non-dominated portfolios, compute the ratio of:

    (a) the Cartesian distance, from the dominated portfolio closest to the mean spending of the two non-dominated portfolios, to the line segment defined by the two non-dominated portfolios, and

    (b) the Cartesian distance between the two non-dominated portfolios.

iv. to the computed ratio, add the lesser of the two non-dominated portfolios' ages, divided by an "age mitigation value" parameter, which increases as the frontier becomes more stable. In other words, as the average efficiency of solutions on the frontier (defined in Section 3.4.4) begins to change less, the age of the items on the frontier becomes less of a factor in the weighting function.

The dominated portfolios are sorted by ascending weight. If there are gaps in the frontier, this causes dominated portfolios close to the gaps in the frontier to be favored over others and elitism to be achieved in this way. The two fronts are merged with crowding distance.

A specified number of portfolios with the highest spending-benefit ratio are selected from the ordered dominated set, with each remaining offspring selected with a certain probability of stopping when a specified threshold on the number of portfolios that survive each generation is reached. The parameters, probability of survival and the offspring population threshold are dynamically increased, up to an appropriate value, as more generations pass without finding a new efficient solution.

This methodology could also be extended to identifying non-dominated frontiers in more dimensions, since the basic principle applies in that case as well. The NSGA-II algorithm that is designed to solve for M dimensions and an offspring of N candidate solutions has a worst case time complexity of $O(2MN \log 2N)$ for crowding distance calculation and $O(2N \log 2N)$ for then sorting on the crowding distance. In our proposed methodology, calculating the crowding distance and sorting on it is replaced by calculating the weights and sorting on it. This is significantly easier, given the problem structure and our assumption that sorting is always done on the primary spending metric and hence, it has a worst-case complexity of ordinary sorting.

3.4.4 Stopping Criterion

If the spending-benefit ratio of a solution is defined by the benefit of a solution divided by the resource consumption of a solution, then we define the *average efficiency* of a set of solutions by the sum of the spending-benefit ratios of all solutions in the set divided by the cardinality of the set. We used the average change in the average spending-benefit ratio of all non-dominated solutions discovered so far between generations as a metric to determine the stopping condition of the genetic algorithm. At each generation, the average value of the spending-benefit ratio of all offspring can be calculated and the average change in this metric over the last few generations is computed.

When this change is less than a certain threshold value for a certain number of generations, the algorithm is terminated. However, the problem with this approach, when applied to time-sensitive planning exercises, is that depending on the machine it is run on and the parameter combination chosen, the time taken for the optimization to complete is not very predictable. As such, setting stopping criteria to have the optimization run for a particular amount of time is much more attractive during 'real time' PPS exercises. The problem with this approach, in turn, is that based on the machine it is run on, the algorithm could result in varying solution quality;,in spite of using the same random number seed to inject randomness into the process.

3.5 Selecting the Ideal Portfolio

Though optimization techniques represent the best quantitative approach for PPS and are the most flexible in terms of attacking all the relevant issues, they have largely been ignored due to (i) poor input modeling frameworks, (ii) burdensome computing requirements, and (iii) non-intuitive results from optimization models. So far, we have addressed the first two of these three issues. In this section, we will address the third issue.

In Hess (1993), the author contends that management science has failed to produce intuitive and attractive selection models. These models, the study states, have little practical impact despite the increased sophistication displayed by optimization models. This has largely been true in our experience, and is a sentiment echoed by the authors in Ghasemzadeh and Archer (2000), where the authors go so far as to say "One of the major reasons for the failure of traditional optimization techniques is that they prescribe solutions to portfolio selection problems without allowing for the judgment, experience and insight of the decision maker." The "increased sophistication" referred to by the PPS literature tends to focus on optimizing portfolios on multiple objectives, or on one normative utility maximizing decision vector. However, decision makers prefer minimally restrictive decision support systems that allow them to model their own intuition along with the analytics, as demonstrated in Pfeiffer *et al.* (2014). Decision strategies such as *satisficing* (SAT) where solutions are retained for every objective that meet a certain threshold until only a certain number of solutions are left, *lexicographical* (LEX) where objectives are ordered by priority and only a certain number of solutions that maximize the most important objective are retained, and *elimination by aspects* (EBA), which is a combination of LEX and SAT where objectives are prioritized and then evaluated sequentially with solutions that do not meet a certain threshold being discarded at each stage until only a certain number remain, are typically used to whittle down the number of portfolios. All three have received considerable attention in the literature as decision strategies employed to choose between various alternatives in different decision environments. These strategies can be extended to selecting the ideal portfolio, once enough alternatives have been generated. The advantage of using these kinds of strategies, and especially EBA, is that they can easily be made interactive and allow decision makers to bring their intuition to the problem, and spend more time engaging with the decision making process in a meaningful manner.

From our experience at Intel Corporation, Financial Analysts work on the financial modeling of various portfolio units along with Portfolio Leads. This portfolio team is responsible for modeling interdependencies and relationships between portfolio units in addition to performing some level of portfolio unit screening to reduce the number of portfolio units to be considered in the selection exercise. This pre-processing step may, for example, exclude portfolio units that have already been committed and thus cannot be excluded, or portfolio units that do not align with the overall business strategy. This "roadmap" is then sent off to individual portfolio unit owners to verify the values for a number of *attributes*, ranging from volume to timing to headcount, and any others used to calculate a benefit metric, such as NPV.

Once the financials have been modeled, the Portfolio Lead then “plays out” various scenarios by constraining different resources and optimizing for different objectives, queuing a separate run of the optimization process each time. This results in a set of attractive portfolios. The Portfolio Lead then interacts with the efficient frontier, exploring the objective space to select a few portfolio options usually around the current portfolio being executed. For example, in Figure 3.3, the Portfolio Lead may explore the efficient frontier in the objective space defined by the NPV and portfolio spending by picking any one of the portfolios and adding or excluding portfolio units that are currently not in that portfolio to observe the change in NPV and spending . With a DSS, this change can be immediately visualized and understood, and can help communicate the impact of such a change. Usually, the Portfolio Lead may choose portfolios close to the current execution portfolio that allows the building of a compelling story. These decisions have to be based off of the current ”plan of record” or portfolio when communicating to senior management to bring out the benefit of changing the company portfolio. For example, if I is the current portfolio being implemented at the company, a powerful case could be made for moving from portfolio I to II for a little extra spending , which would allow the company a significant rise in NPV. Alternatively, moving from portfolio I to III would allow the company increased NPV, and additionally, some savings in spending. Moving to IV would mean maintaining the same NPV but would allow the company to have a significant savings in spending. The efficient frontier thus allows for visualizing the impact of various decisions and help communicate this effectively.

The EBA methodology is then used by the Portfolio Lead to prune the set of portfolio alternatives building a detailed story for each one of the remaining options.

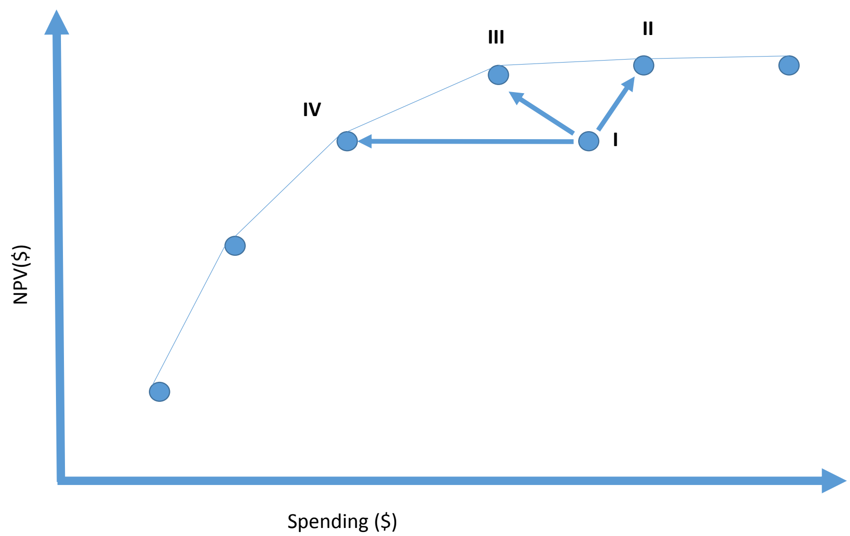


Figure 3.3: Objective Space with Efficient Frontier Relative to Portfolio I, the Current Portfolio Being Executed by the Company. Portfolio II, III and IV Are Candidate Portfolio Alternatives the Portfolio Lead May Select

The Portfolio Lead presents these to the Decision Maker and his staff, who then examine the portfolio unit composition of each portfolio and have detailed discussions regarding the composition of the final portfolio they decide to execute. This process flow is summarized in Figure 3.4.

Given a black-box valuation function for the portfolio metrics, it is important at this stage to present all relevant information to the decision makers while they make their decision. As such, we find that the features most useful to the process, apart from the efficient frontier, are:

- A comparison grid to compare portfolios on various metrics
- A comparison grid to compare and contrast the constituents of a portfolio, the contribution of each portfolio unit to the overall portfolio, and various portfolio

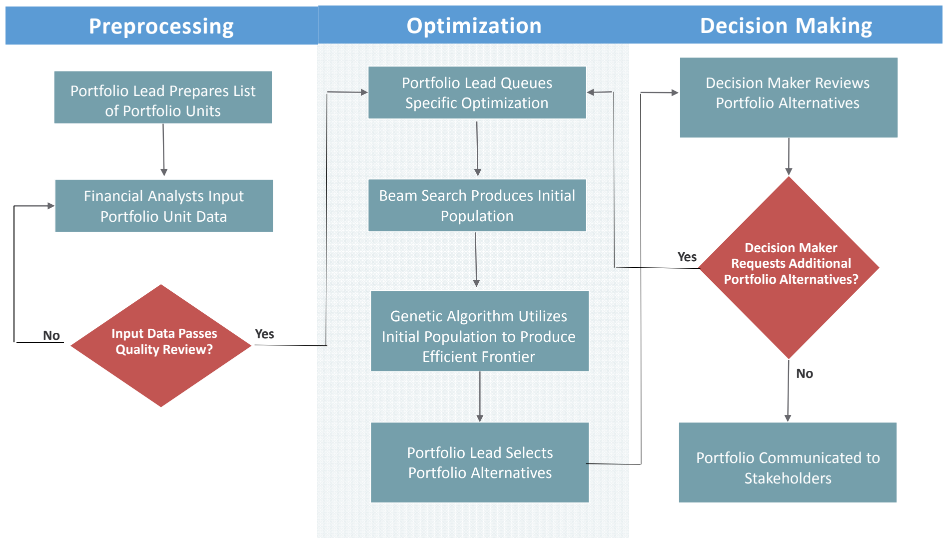


Figure 3.4: Process Flow for Portfolio Decision Making

metrics

- A "fence" report that summarizes the portfolio units that are "on the fence," i.e., not in all portfolio choices but present in at least one, allowing decision makers to focus their attention on the important portfolio units and not on those that are in all portfolios or are always left out

- The ability to navigate within the objective space and investigate the effects of forcing portfolio units in and out.

- An "incremental value" report that indicates what the change in spending, benefit or some other metric of a portfolio would be if a particular portfolio unit was dropped. Given the black-box nature of the valuation, this may not be immediately obvious.

- A "waterfall" report that details, step-by-step, how the organization can migrate from one portfolio to another.

The goal of a waterfall report is to produce an ordered series of feasible portfolios each with minimal changes from the previous portfolio in the series in terms of portfolio unit composition. Hence each step in a waterfall report should be a feasible portfolio. This provides decision makers with an idea of the portfolio units to be added or dropped from the starting portfolio to reach the ending portfolio, and the impact of each of these steps or funding decisions. A sample waterfall report is displayed in Table 3.4, ordered by the benefit impact between two portfolios comprising portfolio units A, C, D, E and G, and B, E, F and G. By recalling the structure of the problem in Figure 3.1, we notice that portfolio units A and D shared a Hard-And relationship, and hence, must be added or dropped together. This is why step two includes a change comprising two portfolio units. Each of the transition portfolios is a valid portfolio by itself. Portfolio Leads are able to generate useful conversations with the aid of waterfall reports regarding funding or de-funding of portfolio units.

Type of Portfolio	Portfolio Composition	Decision	Benefit Impact	Resource Impact
Starting Portfolio	ACDEG		\$100M	\$10M
Transition Portfolio	CEG	Drop AD	$-\$20M$	$-\$3M$
Transition Portfolio	CEFG	Add F	$-\$50M$	$+\$2M$
Transition Portfolio	EFG	Drop C	$+\$10M$	$-\$1M$
Transition Portfolio	BEFG	Add B	$+\$80M$	$+\$3M$
Ending Portfolio	BEFG		\$120M	\$11M

Table 3.4: A Waterfall Report Generated Between Two Portfolios Containing the Portfolio Units A, B, C, D, E and A, B, F, G Respectively. Values in the Last Two Columns for Rows with No Decision Indicate the Value of the Combination of Portfolio Units in the Portfolio Composition Column

Implementing these tools in a DSS is possible, thanks to the fact that the problem was modeled as an integer program. Therefore, by making simple constraint changes and solving the integer program, it is possible to make the various transitions and investigate the effects of the various changes. Each new resulting portfolio of the new 'solve' will have to be 're-valuated', as usual for the new metrics. Allowing users to inspect and contrast portfolios on various metrics in this manner also overcomes another traditional shortcoming of integer programming based optimization models, namely, the ability to incorporate risk in the decision making framework since the risk involved with any particular portfolio can be inspected just like any other portfolio metric in the DSS.

3.6 Computational Results

For initial testing, we began by comparing the performance of the solution algorithm outlined in this paper for a small set of 2,000 decision units resulting from around 150 portfolio units based on a real portfolio selection problem within the company. All decision units were exhaustively enumerated and valuated for this small example, and compared to an efficient frontier generated via the EVS methodology, which represents an exact method for obtaining the efficient frontier. We then solved the problem using BIG, which was able to identify all of the same portfolios on the efficient frontier, taking approximately the same time (around two minutes) as the EVS methodology, as shown in Figure 3.5. A video of the progression of the genetic algorithm can be viewed at `https://www.youtube.com/watch?v=hYb3-9PR7Fs`. Figure 3.6 provides the efficient frontiers produced by the genetic algorithm at intermediate stages to produce the final efficient frontier shown in Figure 3.5.

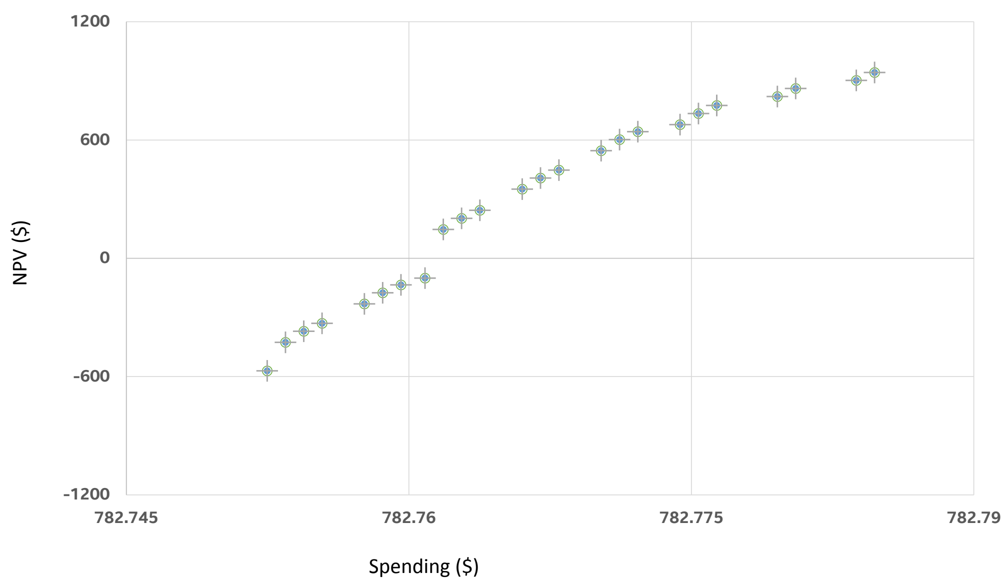
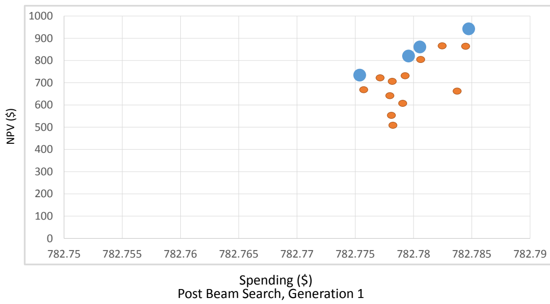
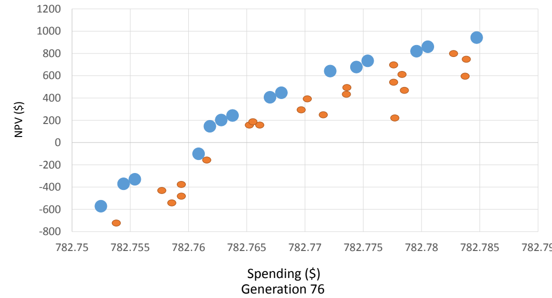


Figure 3.5: A Small Problem Instance Where the Generated Efficient Frontier Through Evs (Hollow Circles) Perfectly Corresponds with the Results from Big (Crosses) for a Problem Instance of 2000 Decision Units

For larger problem instances, the time taken for the EVS methodology increases

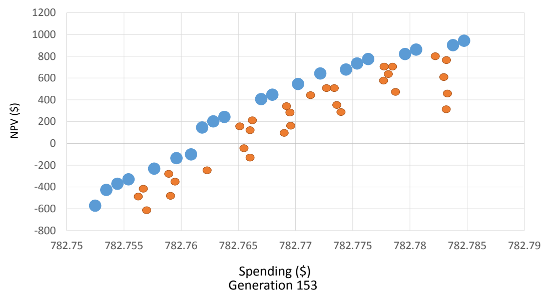
Genetic algorithm population at intermediate stages

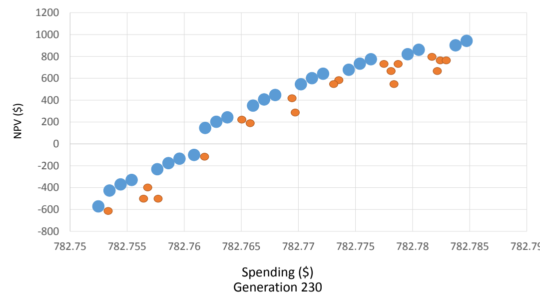




(a) Generation 0, Post Beam Search

(b) Generation 36





(c) Generation 115

(d) Generation 121

Figure 3.6: Plots at Various Stages of BIG to Arrive at the Same Solutions That Comprise the Efficient Frontier in Figure 3.5. The Algorithm Favors Retention of Solutions That Fill the ‘gaps’ in the Efficient Frontier

exponentially, which of course is one of the primary motivations for developing BIG. As discussed earlier, this stems from the increased number of interactions, especially the optional relationships, between the various portfolio units that causes a large number of decision units to exist for each portfolio unit. Generating the decision units before hand, so that the appropriate values can be looked up by the beam search and genetic algorithm as needed greatly speeds up the performance of BIG, but depending on the size of each decision unit and the fact that the number of decision units for each portfolio unit increases by a factor of two for every optional relationship in the neighborhood, there may only be a certain number of decision units that can be stored in memory. This means that if a new decision unit not stored in memory is encountered, it must be valuated on the fly, and this can considerably slow down the performance of the algorithm.

Further computational studies were then conducted on variations of the portfolio selection problems of the six main groups for which our methodology was implemented at Intel Corporation. The six groups represent actual divisions within the company and can be broadly described by the market they service. Three of the groups (PC, Server and Embedded Computing) serve external markets, while the other three work on portfolio units that were used by the first three. For example, a hardware or software module used to build desktop computers by the PC group would be classified as a support portfolio unit that belonged to one of the latter three groups.

For each of the six divisions, based on the actual data from their respective PPS exercises, 20 different problem instances were generated. These divisions largely differ from each other in terms of the parameter α, or the ratio of total decision units to portfolio units in the problem instance. efficient frontier, and recorded the average EVS solve time for all 20 problem instances. The results are shown in Table 3.5.

Department	Number of Decision Units	Number of Portfolio Units	Ratio of Decision Units to Portfolio Units	Average EVS Solve Time (mins)
Division 1	700	700	1	0.5
Division 2	400	200	2	0.5
Division 3	8,000	200	40	5
Division 4	50,000	200	250	60
Division 5	180,000	200	900	180
Division 6	36,000,000	1000	36000	1440

Table 3.5: Problem Instance Parameters and Average Solve Time for Six Different Divisions at Intel Corporation

In the Division 4 problem instances, for example, with 50,000 decision units,the genetic algorithm was able to quickly approximate the efficient frontier within about 5 minutes and reach the true efficient frontier within twenty minutes, on average. In comparison, the EVS methodology took about an hour to identify the same frontier on the same server. This is shown in Figure 3.7, where the "5 minute Genetic" shows the status of the genetic algorithm after five minutes of run time. The "20 minute Genetic" shows the status of the efficient frontier and the improvements to it after running it for an additional 15 minutes. These results are contrasted with the EVS run that took about sixty minutes to generate the same portfolios on the efficient frontier.

All groups had similar financial models with slight variations in how they were modeled in terms of the optimization problem. An example of Division 6's problem structure is detailed in the Appendix. The overall framework was well received by all the divisions we worked with and is now an integral part of their decision making.

Efficient Frontier comparisons between EVS and the genetic algorithm

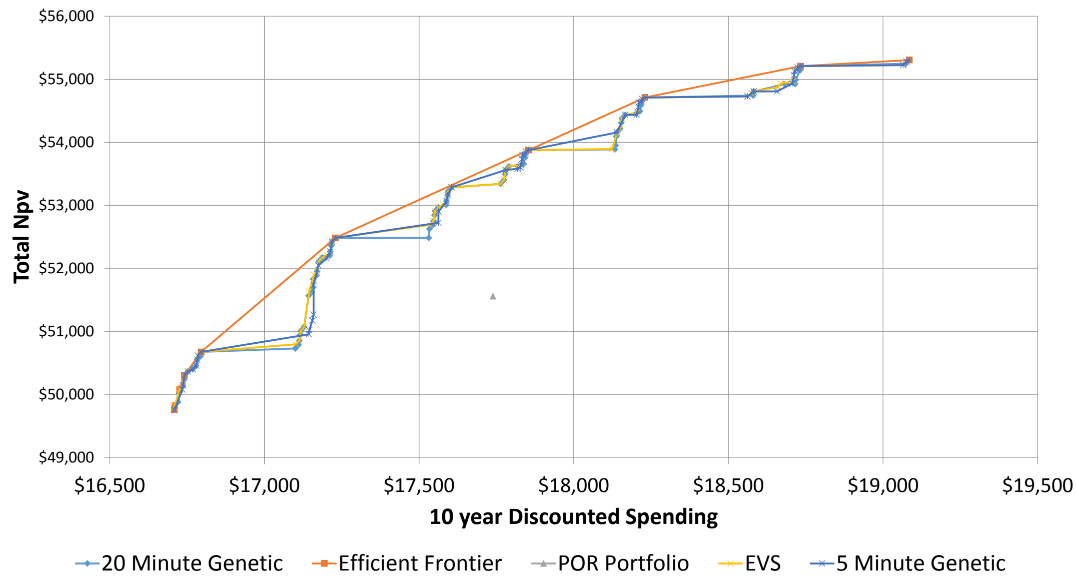


Figure 3.7: Comparison of the Genetic Algorithm Run for Different Amounts of Time Versus the Efficient Frontier Obtained via EVS

As the ratio of decision units to portfolio units rises, the solve time for the EVS begins to increase exponentially. We found from our results that when this ratio is low, approximately less than 5, the genetic algorithm takes a long time to identify all non-dominated solutions. This is because the genetic algorithm is not very effective at handling problems that have a high number of portfolio units that are required for others, as the number of feasible solutions in those cases are very low. As such, the EVS works better for problem instances with a ratio of decision units to portfolio units less than five. Otherwise, the correction operator discussed earlier in this paper would have to be employed. As evidenced by Figure 3.8, as α increases, the genetic algorithm performs significantly better. For low levels of α the EVS approach is able to identify the efficient frontier faster.

In all the instances, the weights for elitism were contrasted with the crowding

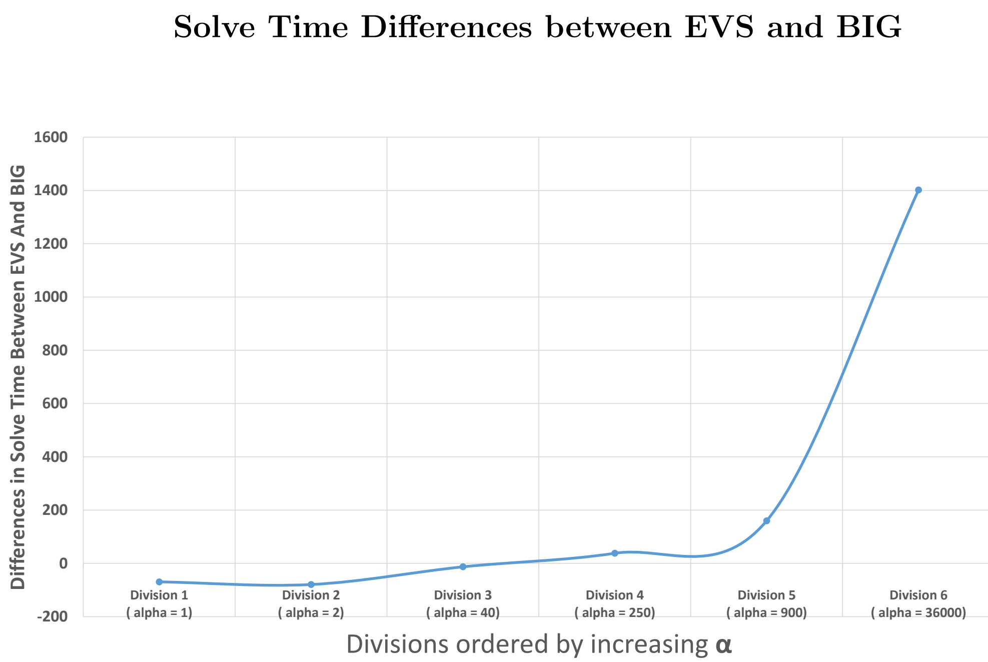


Figure 3.8: Differences in Solve Time for Various Divisions Between the EVS and BIG. As Alpha Increases, the EVS Solve Time Rises Significantly. BIG Was Run for Twenty Minutes for All Problem Instances

operator and observed to reach stopping conditions in fewer, or, in at most an equal number of generations. Diversity can also be achieved by dividing the objective space into "windows" or various resource consumption zones and conducting local searches within those windows. This has proved promising in improving upon solutions within a resource consumption zone already identified by the genetic algorithm. Candidate solutions that do not fall into the current resource consumption zone are thrown away and generating new feasible ones or correcting infeasible ones can get computationally expensive. As such, this methodology is best suited for fine tuning the results in a particular resource consumption or spending range rather than being used for developing the entire efficient frontier.

All experiments were run on an Intel Xeon E5-2680 v4 @ 2.40 Ghz with two physical processors, each with 14 cores and hyperthreading yielding 56 logical cores and 1.5 TB of RAM. Before we discuss the performance of the algorithm as a whole, it should be noted that in our experiments, 70 to 90% of the running time of the algorithm was spent evaluating the NPV of generated solutions. The complex financial model used to determine the benefit of each solution was formulated by senior financial analysts and involves a Monte Carlo simulation. To remind the reader, valuating a portfolio involves identifying the decision units comprising the portfolio and then using a financial model to valuate the decision units. Storing the results of common decision units greatly helped in speeding up the identification of the efficient frontier, however, due to the amount of information stored, including various benefit and resource metrics for each decision unit, we limited the number of decision units for which results were stored to a maximum of two million, which in our exercise occupied roughly around 4 GB of memory.

3.7 Discussion on Real-life Business Implementation

Genetic algorithms have a higher probability of producing better solutions the longer they are run. However, Portfolio Leads require attractive solutions quickly, and discovering the right parameters for the genetic algorithm is key to executing an efficient PPS framework. In practice, we abstracted the various genetic algorithm parameters into short, medium and long runs based on the amount of time Portfolio Leads were willing to spend discovering attractive solutions. We also developed a real time visual of the progress of the genetic algorithm so that Portfolio Leads could see, at any stage, the change in the efficient frontier of solutions discovered at any point in time. Portfolio Leads were, in general more favorable to having the solver run for a deterministic amount of time rather than waiting for it to terminate on its own. Given the number of changes to various inputs based on new information coming in at the last moment, having the portfolio solver run for long hours to terminate on its own, was not an attractive option for the Portfolio Leads. They typically ran the tool many times during a planning exercise, often going back and adjusting various inputs and then rerunning the solver again. The ability to enable Portfolio Leads to easily perform these operations is key to having the PPS process run smoothly.

This visualization provided two benefits. One was to manage expectations while Portfolio Leads waited for the results of the optimization which depending on the running time specified ranged from between five minutes to twenty four hours. The second was to inform the Portfolio Leads that there could possibly exist an efficient frontier with a greater average spending-benefit ratio of all solutions. This was important because using the genetic algorithm does not yield provably optimal solutions like EVS.

The difficulty of solving the problem via EVS, and the increased solution times it results in, are simply not acceptable to fast moving dynamic PPS exercises. Typically new information is continuously being made available to various analyst teams regarding market conditions that could result in them reworking their inputs to their various portfolio unit models. As such an algorithm that runs for a day that provides a provably optimal answer may be less useful than one that approximates closely the efficient frontier in a fraction of the time to better accommodate requests to rerun the optimization to incorporate last minute changes to inputs. Moreover, this also allows the process to be repeated more frequently through the financial year.

Overall, the entire approach, including the modeling framework and the optimization, was deployed through a DSS called "Voyager." Initially deployed with Intel Corporation's server division, Voyager was soon mandated by company leadership to manage the PPS decisions for the rest of Intel Corporation's product divisions totaling roughly 25 billion dollars in terms of spending and 195 billion dollars in NPV. Surveying the analysts who input information into "Voyager," it was unanimously agreed that it sped up the process from months to days and helped improve the quality of PPS decisions while making the entire process easier and more intuitive compared to previous years practice, which included ad-hoc advocacy based approaches, off-the shelf tools and hiring outside consultants. The framework was easy to learn and communicate, and extremely flexible to work with, requiring little to no training on how to use it. Transitioning portfolio analyst teams typically had less than a day's training to begin using the tool.

The ability to arrive at a good efficient frontier quickly using the genetic algorithm detailed in this paper depends on the following parameters: the number of solutions retained after every step of the beam search, the number of solutions retained after every step of the genetic algorithm, and the number of stagnant generations before evaluation is terminated.

Using the various tools mentioned earlier, the decision maker and his staff were able to examine, compare and contrast the various portfolios presented by the portfolio leads and have discussions where the portfolios were evaluated on the basis how they performed for various conflicting objectives like NPV, revenue in a particular year, headcount consumption for specific skill sets and so on. The portfolio exercise was more effective and decision making at the company dropped from a period of a few months to a few days.

3.8 Conclusions

PPS is an extremely complex task because of the many competing objectives on which portfolios have to be optimized and the large number of interdependencies these portfolio units have with each other. These interdependencies typically cause the benefit expected to be realized from a portfolio unit to be dependent on the inclusion of other portfolio units in the portfolio. This greatly increases the complexity of the portfolio optimization problem and presents a real challenge to decision makers.

An organized framework is needed to tackle the PPS problem, and one such integrated framework is presented in Sampath *et al.* (2015). However, the optimization methodology presented in that paper falls short when the number of interactions between portfolio units is too large. To address this deficiency, we present in this paper an alternative solution methodology that aims to generate an efficient frontier of portfolios for decision makers in an acceptable amount of time. We have attempted to combine heuristics in an intelligent manner to arrive at attractive solutions quickly and believe that it has great promise towards solving other problems of this nature, where the objective function is hard or expensive to evaluate. The ready acceptance of of this optimization-based PPS framework by the PPS committee at Intel Corporation due to its ability to handle problem instances involving great model complexity demonstrates that it is a step in the right direction towards increasing the acceptance of optimization-based PPS models in industry.

With regards to the framework itself, this paper has addressed (i) a scalable framework that incorporates increasing model complexity, and (ii) better ways to increase decision maker confidence in the results produced. There are a number of avenues of further research we have not addressed, however. One of them is how to normalize the inputs from various analysts who may interact with the tool in the modeling stage to input data regarding each portfolio unit. We have also not addressed the issue of how to model portfolio units and impacts from a financial perspective, and have assumed the financials and impacts have already been modeled.

Further research is also needed toward algorithm implementation to improve the speed of portfolio generation to approach the true EVS frontier and better ways to incorporate risk considerations in the framework. With regards to algorithm implementation, future research should focus on achieving attractive solutions faster, including better utilization of modern parallel processing technologies to evaluate the benefit of large number of offsprings in each generation, as this is often very expensive in terms of processor cycles. Further research should be focused towards possibly tracking the direction of improvement for portfolios towards the efficient frontier for the duration of the genetic algorithm and replacing the complex objective evaluation function for a few cycles to speed up the algorithm.

More research into the use of combining PPS optimization models with low-cognitive, non-compensatory decision strategies will ultimately lead to a higher adoption rate of versatile models like the one presented here that tackle many different aspects of the PPS process. However, until risk considerations are incorporated into the model before optimization, portfolios attractive from a risk perspective may not be present among the portfolio choices decision makers are provided.

Chapter 4

INCORPORATING PROJECT COST AND BENEFIT RISK CONSIDERATIONS INTO THE PPS FRAMEWORK

4.1 Introduction

Project portfolio selection is a very important activity for most organizations as it prioritizes and determines the projects a company will undertake by determining where spending is directed. With large organizations, project portfolio selection may occur once or twice a year and the final portfolio must be approved by senior management and then communicated to the rest of the company. It is a complex process that involve many different players that gather, process and visualize the data needed to make such decisions. Often times, decisions to fund various project alternatives within an organization cannot be made in isolation - the impact of a candidate project must be examined in context of the projects. In other words, the interrelationships of various projects that are being considered for selection determine the final contribution of a project to the portfolio. Based on the financial model used to *valuate* (i.e. to quantify and assign numerical value to), the benefit accrued from a project, and the nature of the impacts that these projects have on one another, the process of modeling and solving a problem instance can be extremely complex and it is often not possible to represent the resulting benefit of the combination of projects as a linear relationship. This may result in a project having many different versions, each with its own values for its benefit (for example, the revenue accrued upon the completion of a project), depending on other projects that are included alongside it in a portfolio. Enumerating and valuating all the different versions of a project may require off-line pre-processing that takes into account the various intricate relationships between projects and valuates them in order to feed into an optimization model. Once all the different viable versions for each project have been valuated however, they still must be combined in a feasible manner to yield a viable portfolio.

Optimization models for Project Portfolio Selection (PPS) problems are great for building portfolios that maximize or minimize certain portfolio metrics, but their use in practice until recently has not been comparable to other selection techniques (such as ad-hoc approaches and strategic planning tools) in part due to the amount of data they required to be collected Alpaugh (2008). This has however started to change. The literature on PPS is extensive incorporating multiple realistic features such as Net Present Value(NPV) and headcount. These models are easily incorporated into existing decision making processes and have been successfully deployed at various organizations. Sampath *et al.* (2015) present a framework that overcomes many of the challenges associated with traditional optimization-based frameworks for PPS like the ability to handle model complexity, project interactions and user acceptance, all of which are problems well documented by the work of Archer and Ghasemzadeh (1999). This paper extends the framework presented in the paper and describes a methodology of prioritizing portfolios that are not as "risky" as others.

Maximizing profit (by either maximizing a benefit or minimizing a cost, or both) and minimizing risk are two common objectives in PPS problems that decision makers in large organizations find useful Bible *et al.* (2011). An effective decision making methodology must allow for the exploration of trade-offs between the two objectives while at the same time taking into consideration the risk involved in meeting certain targets for the benefit or the risk involved in exceeding a certain pre-determined budget. This framework should thus incorporate risk as a metric to assess portfolios.

Projects may have a number of benefit and cost metrics associated with them. Some of these metrics are captured directly through reports from analysts familiar with the project's financials. These reports are then used to infer the parameters of the underlying probability distribution that represents the set of possible outcomes for that project metric. The remaining project metrics are functions of these input metrics. The expected value approach, where the expected value of distribution of the various metrics are maximized, may yield solutions that are sub-optimal when evaluated in terms of risk. Based on the expression that determines the value of these derived metrics, the expected value approach may hide the dispersion of the distribution of the project metric used to evaluate risk. As such, portfolios with significant probabilities of extreme case events may be selected and presented to decision makers.

In large organizations, there are often various stages involved in selecting the optimal portfolio with different parties involved each with different roles and responsibilities. If a senior decision maker higher up the company decision making hierarchy rejects a portfolio optimized for some central tendency measure, this may lead to another cycle of portfolio generation and selection. In the worst case, a portfolio optimized for some central measure tendency like the expected value, may lead to catastrophic realizations of outcome. This may particularly be problem when the distributions of the various project metrics are represented by asymmetric distributions. Offline processing allows the handling of complex evaluations of the distributions of each project metric because it allows us to work with the entire distribution rather than the expected value which may be derived from the input distributional information of the various projects. By incorporating the distributions into decision making, uncertainties and risks are no longer ignored and decision makers can be confident of selecting a portfolio that meets their level of risk aversion.

Senior DMs who often have the final say in PPS decision making exercises must be presented with options that are attractive given their risk preferences without taking up too much of their time eliciting said preferences. The optimization model must then be able to generate attractive portfolios that maximize utility for a given DM or group of DMs based on their varying risk preferences.

In this paper, we present a chance-constrained PPS model that generates an efficient frontier of portfolios considering the two objectives of benefit (to be maximized) and cost (to be minimized) for various risk levels that enables decision makers to choose the most preferred portfolio. This model seeks to incorporate various kinds of constraints that model business rules that are encountered in the selection process. It allows for the incorporation of a decision maker's risk tolerance by generating portfolio options for various risk levels for various cost budgets and benefit targets. It also incorporates project inter-dependencies in a novel and generalizable way and presents a new way to evaluate projects. We follow this section with a brief literature review discussing incorporating risk in the PPS process, followed by a section that discusses project evaluation and optimization. We then present a numerical study to better illustrate the models in the previous section and finally present our conclusions.

4.2 Literature Review

A number of studies including those by Heidenberger and Stummer (1999) and Archer and Ghasemzadeh (1999) note that when it comes to PPS, mathematical programming approaches are viable tools that can incorporate project interdependencies into the model and support sensitivity analysis. However these studies also discuss why these models are often overlooked for a number of reasons that include a) the extent of data that needs to be collected, b) model complexity and c) the inability in general to incorporate risk considerations in an effective manner. The need to collect large amounts of data cannot, in most cases, be skirted for better quality decision making. Assuming that the organization undertaking the exercise has the necessary resources to collect a sufficient amount of information to derive meaningful results for decision making, the works by Sampath *et al.* (2015) and Sampath *et al.* (2018) introduce a framework that is intuitive to understand and readily accepted by senior decision makers. This model is also able to handle increasingly complex problem instances that involves numerous project interactions.

However, as stated before, since these kinds of optimization models often optimize on some central tendency measures, the portfolio options generated must then be examined from a risk perspective which may lead to more portfolio alternatives being generated than necessary with no guarantee of generating a portfolio that meets the desired risk tolerance of the decision maker. These also may not produce solutions for a given risk tolerance level. Moreover, an optimization model must have specified a suitable benefit metric to optimize to produce portfolios. This is why there is a lot of literature on developing the appropriate benefit metrics like net present value (NPV) Sefair *et al.* (2017), Li *et al.* (2015). Other metrics like Value at Risk (VaR) Zhao and Xiao (2016), Molina *et al.* (2017), Conditional Value at Risk (CVaR) Paquin *et al.* (2016), Maier *et al.* (2016) are also considered. These approaches seek to incorporate the risk into the optimization metric, and may be considered as metrics to be minimized in an optimization model at a given risk level. NPV remains the most popular metric to measure the benefit of portfolios in practice Harmsen *et al.* (2010).

Recently, more complex measures have also been developed and used. Hall *et al.* (2015) specify alternative risk indices to VAR and CVAR to optimize portfolios against. These indices model correlation and interaction effects such as synergies and use a discrete non-linear optimization model to produce robust portfolios. Similarly, Costantino *et al.* (2015) develop a neural network model to develop project critical success factors (CSFs) and use it to select for portfolios that are less likely to fail. These works along with Abbassi *et al.* (2014) where the authors employ a 0-1 Mixed Integer Program to maximize the cross-entropy of a project portfolio where the projects have various risks and interdependencies all seek to find ways to characterize the risk of portfolios with a new measure while using it as the objective in their optimization models but may suffer the same problems as the expected value approach by ignoring the dispersion present in asymmetric distributions used to evaluate risk.

The use of fuzzy models for project portfolio evaluation and selection has also received some attention in the literature. Various portfolio metrics like the net present value, spending and so on are represented by 'fuzzy sets'. Fuzzy sets are characterized by membership functions that determine the degree of membership of any number to a set and have to be explicitly stated before being used in an optimization model as noted by Perez and Gomez (2016). Some efforts have been made to make the fuzzy approach to project portfolio selection more intuitive in the works of Mohagheghi *et al.* (2015). However, fuzzy logic rules are difficult to develop and analysis is difficult because fuzzy membership functions and outputs can be interpreted in a number of different ways. Other quantitative approaches to PPS along with their shortcomings are detailed in Iamratanakul *et al.* (2008).

An alternative way to deal with the risk inherent in various portfolio metrics is to incorporate chance-constraints into an optimization model. Typically, given a certain risk-level or tolerance, a chance-constrained formulation makes sure that the probability of meeting a certain constraint meets a specified risk-level. Originally developed by Charnes and Cooper (1959), much work has been done in the field of chance-constrained programming. Many studies such as Henrion (2004) over the years have sought to look into ways of increasing the efficiencies of solving such problems. Chance-constraints in financial portfolio optimization has also begun to receive some attention as evidenced by the studies of Dombrovskii and Obyedko (2015) and Noorian and Leong (2014). A more detailed review of the practical applications of chance-constraints is discussed in Geletu *et al.* (2013), Shen *et al.* (2010), Luedtke (2014), Luedtke *et al.* (2010) and Küçükyavuz (2012). The use of chance-constraints is necessitated because when the distributions being considered are not normal, it is hard to find a deterministic equivalent mathematical formulation that can be solved such as those in (Wiesemann *et al.*, 2014; Goh and Sim, 2010; Kalashnikov *et al.*, 2017).

To the best of our knowledge, there have been few approaches discussed so far that allow the incorporation of risk into the optimization model with the use of joint chance constraints to solve a PPS problem while evaluating projects by incorporating dependencies from other projects for various cost and benefit levels. In this paper, we employ chance constraints defined over various risk tolerance levels to provide decision makers portfolio choices so that they may explore the available optimal portfolios generated and select the most attractive portfolio. The optimization framework takes into account the probability of meeting certain targets, and the subsequent optimization to identify non-dominated portfolios meeting certain benefit targets at various cost levels for certain levels of risk.

4.3 Problem Description

For the purposes of this paper, we shall refer to a project, or any engineering effort that consumes company resources and yields some benefit as a *portfolio unit*. Inter-dependencies or relationships that these portfolio units share with each other may cause a given portfolio unit to yield a different benefit depending on the other portfolio units being executed. We handle this by defining a *decision unit* as a specific version of a portfolio unit that is defined explicitly by the other portfolio units it shares inter-dependencies with that have also been picked to be executed alongside the portfolio unit. A decision unit is thus a 'version' of a portfolio unit. We divide this section into two parts, the first of which describes project evaluation, and the second the optimization.

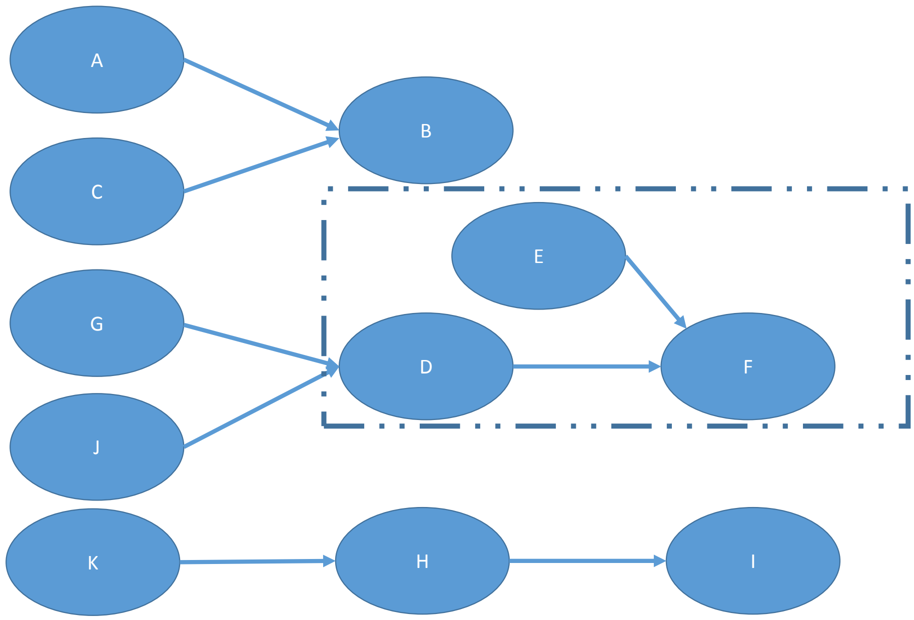


Figure 4.1: A Few Portfolio Units with Optional Relationships That Yield 22 Total Decision Units. The Dotted Line Indicates the Portfolio Units in the Neighborhood of Portfolio Unit D

4.3.1 Project Evaluation

For example, in Figure 4.1 portfolio unit B would have four different decision units each of which may yield a different benefit based on whether B was executed alongside portfolio unit A, portfolio unit C, both, or neither. These will be explained in more detail later.

To begin, we describe the input and output metrics of a portfolio unit. Input metrics are those metrics for which distributional information is directly captured, whereas output metrics are functions of various other input or output metrics. The implicit assumption that is being made in this methodology is that the portfolio units, and their "neighborhood" or other portfolio units they have relationships with, exist in silos, in the sense that capturing distributional information about a portfolio unit and the relationships that affect that particular portfolio unit is enough to derive all the distributional information of the decision units. This is illustrated in 4.1, an inter-dependency graph that shows how these portfolio units effect each other. The portfolio unit at the end of the directed arc is impacted by the portfolio at the beginning of the directed arc. The portfolio unit at the beginning is not affected by the portfolio at the end of the arc. The incoming arcs indicate a dependency between two portfolio units. Portfolio units exist in silos and their decision unit values only depend upon its immediate neighborhood and incoming arcs. To reinforce the concept of how the portfolio units and their neighborhood exist in silos, we present the example of a decision unit and look at the distribution of its various metrics. To remind the reader, the distributional information of the input metrics is captured for the underlying portfolio unit and the relationships that are active for that particular decision unit. For example, in Figure 4.1 the portfolio unit F, and in particular, the decision unit of F when D is also included in a portfolio (but not E). The distribution of the metrics of F depend only on the probabilistic reports captured for portfolio unit F and the relationship value modifiers for the relationship between D and F. The decision unit also captures that portfolio unit E was excluded. The inclusion of portfolio unit D, however, does not depend on the evaluation of portfolio units G and J, because portfolio unit F resides in a silo with respect to those portfolio units. Relationship value modifiers here refers to the change in value of the metric of a portfolio unit because of another portfolio unit its neighborhood being selected in the portfolio alongside it. The distribution of the various metrics of F are agnostic to whether, portfolio units G and J are included alongside in the portfolio or not, because it exists in a silo where it only depends on its immediate neighborhood, that includes portfolio units D and E, but excludes portfolio units G and J. The values associated with portfolio unit F depend on the relationship between portfolio units

D and F rather than the portfolio unit D, which is why the evaluation of portfolio units G and J which affect portfolio unit D, do not affect the evaluation of portfolio unit F. Based on the decision unit selected and the relationships that are “active” or present in the portfolio, the values reported for the relationships will be modifiers that appropriately affect the input metric distributions of the portfolio unit for the specific decision unit. The output metrics of the decision unit can then be calculated by using the sampled input distributions and the financial model. It should be noted that the input metrics contain stochastic factors which may lead to the output metrics being stochastic as well. Moreover, due to the complex project evaluation, even assuming the silos, the output metrics' distribution is not easy to embed into an optimization model.

The final values of the distributional information captured for the input metrics of each decision unit are not directly reported - they in turn are a function of the values reported for the underlying portfolio unit plus any relationships that affect that particular portfolio unit that are active for the portfolio and hence modify the distributions of the portfolio unit metrics. The output metric could thus be a function of other distributions that may not be known for more complicated financial models. This is illustrated in Figure 4.2.

Each input and output metric of every decision unit is treated as a random variable. In order to incorporate these random variables into an optimization model, we sample from their underlying distribution in order to model joint chance constraints. The distribution that models these metrics may not be symmetric, and therefore modeling an exact formulation is not possible. The possibility of these distributions being highly asymmetric reinforces the need for sampling from them and then creating joint chance constraints to model uncertainty and risk. A number of different metrics may be considered for the financial model utilized to valuate the various decision units.

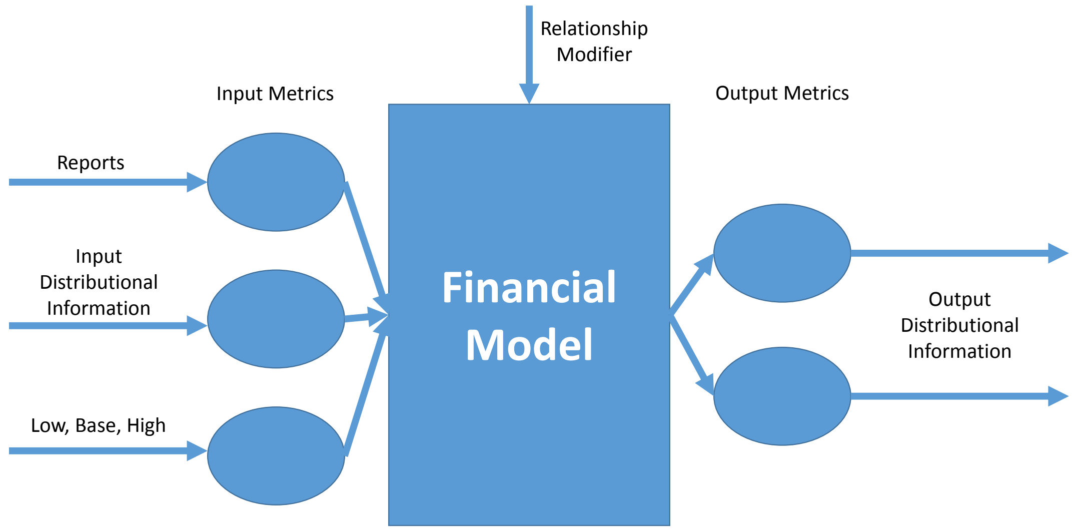


Figure 4.2: Decision Unit Valuation Depends on the Underlying Portfolio Unit Input Distributional Information and the Presence of Other Portfolio Units. The Impacts Are Coded In the Relationships Which Modify the Input Distributional Information and Yields the Output Distributional Information After Passing Through The Financial Model

We discuss an example of one such simple model whose metrics are presented in 4.1. For a set of samples, $s \in S$, for time periods $t \in T$, the set of portfolio units I and the sets of decision units for each portfolio unit $J_i, i \in I$, we can use a financial model to valuate the output and input metrics for each decision unit of ever portfolio unit for sample.

Input Metrics	
L_{jt}^{s}	Number of units sold $i \in I, t \in T$
A_{jt}^{s}	Average selling price of a single unit $i \in I, t \in T$
M_{jt}^{s}	Manufacturing cost of a single unit
G_{jt}^{s}	Total engineering costs for development including research and development $i \in I, t \in T$
D_{jt}^{s}	Discount rate $i \in I, t \in T$
$\mathcal{T}_{jt}$	Taxes and depreciation $i \in I, t \in T$
Output Metrics	
γ_{jt}^{s}	Cost of goods sold $i \in I, t \in T$
ν_{j}^{s}	Net Present Value (NPV)

Table 4.1: Parameters of a Simple Model to Calculate the Net Present Value of a Decision Unit For a Particular Sample From the Underlying Metric Distribution

Table 4.1 shows some input and output metrics for a simple financial model to calculate the NPV of a decision unit. At this point, the distributions of the input metrics are known, since offline evaluation of impacts of the input metrics due to project dependencies has been evaluated.

If the NPV is defined as by the contributing metrics as demonstrated in Equations (4.1) and (4.2), any small change to any of the input metrics through a relationship value modifier (a concept explained in more detail later) can have a cascading effect on the NPV.

$$\gamma_{jt} = L_{jt} * M_{jt} \qquad , j \in J_i, i \in I, t \in T \tag{4.1}$$

$$\nu_j = \sum_{t \in T} (L_{jt}(A_{jt} - M_{jt}) - G_{jt} - \mathcal{T}_{jt})^{D_{jt}} \qquad , j \in J_i, i \in I \tag{4.2}$$

For more complex models, the NPV distributions may not be symmetrical making the implementation of a closed-form for the chance constraints impractical and hence motivates the methodology with sampling. Each of the terms that make up the NPV in Equation (4.2) may be an output metric of its own and may be chosen to be the primary benefit or cost metric. For example, if $L^s_{jt}A_{jt}$ is the revenue, $L^s_{jt}\mu_{jt}$ the cost of goods sold, $(L^s_{jt}(A^s_{jt}-\mu^s_{jt})$ the profit margin or the gross profit, $(L^s_{jt}(A^s_{jt}-\mu^s_{jt})-E^s_{jt}$ the contributing margin or the net profit, $L^s_{jt}(A^s_{jt}-\mu^s_{jt})-E^s_{jt}-\tau^s_{jt}$ the free cash flow or the tax adjusted net profit then it becomes clear that the distributions of each of these may not be well behaved based on the input distributions the financial model used. The input metrics are sampled first and corresponding samples for the output metrics are calculated for each sampling.

Once the distributions for the input metrics for each portfolio unit have been derived and sampled and fed into the financial model to produce various samples for the output metrics, then we can formulate an optimization model that incorporates the chance constraints, the relationships and other business logic. We detail the optimization model below along with the elements required to set it up.

To illustrate the above concepts, we define G as the set of neighborhoods. G_i is the set of all portfolio units in the neighborhood of portfolio unit i. $l_{i'i}$ is then a relationship between portfolio unit i' and portfolio unit i, where $i' \in G_i$. For the sake of example, if we assume that the input metrics are PERT-Beta distributed, and we collect reports on estimates of what the low, median and high values will be for a particular metric. For a particular input metric u and an output metric v, u^L_i is the value of the metric reported in the low case, u^B_i for the median case, and u^H_i for the high case. Relationship modifiers may also exist, and they are mapped to the relationships. Thus, $u^L_{i'i}$, $u^B_{i'i}$ and $u^H_{i'i}$ are used to model the distribution that can be sampled to realize the modification to the reported values of the sampled distribution value of u_i if portfolio unit i' is also included in the portfolio.

4.3.2 Portfolio Optimization

Once the decision units are identified, the appropriate samples can be calculated for the output metrics as well, using the appropriate financial model. These distribution samples are then sampled and used to set up the optimization model. We will present an example to better illustrate these concepts in the next section. We detail the various inputs to the optimization model in 4.2. The allowable budget and target benefit define the targets for the optimization. B_{max} and R_{min} define the maximum budget and minimum revenue in case the corresponding targets are not met. c_{ij} and p_{ij} are the realizations of the random variables used to model the primary cost and benefit metric. S is the number of samples, τ is the risk tolerance level or the maximum percentage of samples that may miss either target. Variance reduction during sampling ensures that each sample p_{ij} for the benefit metric is taken from the same simulation that produces corresponding cost metric c_{ij}.

S	Set of samples for chance constraints $s \in S$
b_{max}	Allowable budget for a feasible portfolio
B_{max}	Maximum cost metric value of a portfolio allowable for a feasible portfolio
r_{min}	Target benefit metric value of a portfolio allowed for a feasible portfolio
R_{min}	Minimum benefit metric relaxation value of a portfolio allowed for a feasible portfolio
c_{ij}	Cost of decision unit $j \in J_i$ of portfolio unit i, $i \in I$
p_{ij}	Benefit of decision unit $j \in J_i$ of portfolio unit i, $i \in I$
τ	Risk tolerance level

Table 4.2: Optimization Parameters

The model can be set up in terms of decision units using coefficient matrices C and E that ensure that the right mix of portfolio units in the neighborhood of portfolio unit i are picked alongside each decision unit selected.

$$C_{jk}^{i} = \begin{cases} 1, & \text{if portfolio unit } k \text{ is included in the valuation of decision unit } j \\ & \qquad \text{of portfolio unit } i, \text{ for all } i, k \in I, \text{ and } j \in J_i \ , \\ 0, & \text{otherwise.} \end{cases} \tag{4.3}$$

$$E_{jk}^{i} = \begin{cases} 1, & \text{if portfolio unit } k \text{ is excluded in the valuation of decision unit } j \\ & \qquad \text{of portfolio unit } i, \text{ for all } i, k \in I, \text{ and } j \in J_i \ , \\ 0, & \text{otherwise.} \end{cases} \tag{4.4}$$

The inclusion matrix and the exclusion matrix defined above are used to ensure that for any particular decision unit selected, the appropriate portfolio units that make up the specific scenario for that decision unit are included or excluded accordingly. The decision variables for the model are defined as follows.

y_{ij} Binary variable that takes a value of 1 indicating if Decision Unit j of Portfolio Unit i is included in the current portfolio and 0 otherwise.

z_s Binary variable that takes a value of 1 indicating if the chance constraint associated with sample s has been violated and 0 otherwise

This model is presented here below that aims to maximize the number of samples that satisfy given budget and target benefit targets called MRL.

$$\text{MRL :Minimize} \quad \tau = \sum_{s \in S} z_s \tag{4.5}$$

subject to:

$$\sum_{i \in N} \sum_{j \in J_i} c^s_{ij} y_{ij} \leq b_{max} + (B_{max} - b_{max}) z_s \qquad \forall s \in S \tag{4.6}$$

$$\sum_{i \in N} \sum_{j \in J_i} p^s_{ij} y_{ij} \geq r_{min} - (r_{min} - R_{min}) z_s \qquad \forall s \in S \tag{4.7}$$

$$\sum_{j \in J_i} y_{ij} \leq 1 \qquad \forall i \in N \tag{4.8}$$

$$\sum_{k \in N : C^i_{jk} = 1} y_{ij} - \sum_{k \in N : C^i_{jk} = 1} \sum_{l \in J_k} y_{kl} \leq 0, \qquad \forall i \in N, j \in J_i \tag{4.9}$$

$$\sum_{k \in N : E^i_{jk} = 1} y_{ij} + \sum_{k \in N : E^i_{jk} = 1} \sum_{l \in J_k} y_{kl} \leq \sum_{k \in N : E^i_{jk} = 1} 1, \quad \forall i \in N, j \in J_i \tag{4.10}$$

$$y_{ij} \in \{0, 1\} \qquad \forall i \in N, j \in J_i$$

The objective function ensures that the least amount of chance-constraints are violated when targeting a specific revenue and budget target or minimizes the probability of a portfolio violating the chance constraints. Constraint sets 4.6 and 4.7 are the primary cost and revenue constraints for each sample of the corresponding metrics. When $z_s = 0$, the specific revenue target or budget for that particular scenario are met. If $z_s = 1$, the constraint is essentially relaxed, and one or both of the targets may be violated for a particular scenario. This means that the model seeks to minimize the probability of violating the chance constraints, $P(\sum_{i \in N} \sum_{j \in J_i} c^s_{ij} y_{ij} \leq b_{max}$ and $\sum_{i \in N} \sum_{j \in J_i} p^s_{ij} y_{ij} \geq r_{min})$. We only use one variable per scenario to guarantee that the same scenarios are selected (or discarded) for the revenue and budget chance constraints. Constraint set 4.8 ensures that at most only one decision unit per portfolio unit is selected. Constraint sets 4.9 and 4.10 ensure that the right combinations of portfolio units are selected for the decision units selected. For example, if the decision unit of portfolio unit F discussed earlier is selected, constraint set (4.9) would ensure that some decision unit of portfolio unit D is selected and constraint set (4.10) would ensure that no decision unit of portfolio unit E was selected. Solving this model leads us to map a specific target revenue and budget to a risk level, so we call this map-to-risk level model (MRL). This model's objective function estimates the probability of meeting the specified targets for benefit and cost. In the next model, this is contrasted against an objective function that, for a decision maker's risk preferences, maximizes the achievable benefit target for a specific cost level.

However, for a specific budget, decision makers may be interested in maximizing the benefit they can achieve for a certain risk level. We can achieve this with a second optimization model, the benefit-for-risk model (BFR). We define additionally a decision variable x_r that refers to the target benefit that we wish to maximize for a given budget and risk level. Here M is a sufficiently large value that relaxes the right hand side of Equation (4.13) for $s \in S$ that result in the case where the left hand side is not greater than the decision variable r. Note that we know that a solution providing at least R_{min} and satisfying the budget constraint in at least τ^* scenarios exists, due to the first phase. τ^* is a specific risk level, the number of scenarios that we know can be satisfied from solving MRL, and we can use this to maximize r in BFR for a specific cost level.

$$\text{BFR}: \text{Maximize} \quad r \tag{4.11}$$

subject to:

$$\sum_{i \in N} \sum_{j \in J_i} c^s_{ij} y_{ij} \leq b_{max} + (B_{max} - b_{max}) z_s \forall s \in S \tag{4.12}$$

$$\sum_{i \in N} \sum_{j \in J_i} p^s_{ij} y_{ij} \geq x_r - M z_s \qquad \forall s \in S \tag{4.13}$$

$$\sum_{i \in N} \sum_{j \in J_i} p^s_{ij} y_{ij} \geq R_{min} \qquad \forall s \in S \tag{4.14}$$

$$\sum_{s \in S} z_s \leq \tau^* \tag{4.15}$$

Constraint sets (4.8), (4.9), (4.10)

$$y_{ij} \in \{0, 1\} \qquad \forall i \in N, j \in J_i$$

Incorporating Additional Business Logic

This framework allows for many other constraints that reflect the business logic commonly encountered in PPS problems to be modeled. For example, so far we only showed relationships that impact other portfolio units but do not preclude specific combinations of various portfolio units. Such soft constraints may not be sufficient to describe the underlying business. Many hard constraints as precedence and logical constraints can also be modeled Sampath *et al.* (2016). For example, one portfolio may be required by another in order to be selected for the portfolio. Relationships may also exist between subsets of portfolio units that make only certain combinations of units within the subset feasible for selection in a portfolio. Apart from the primary cost metric and benefit metric quantitative constraints can also be modeled sampling from the distributions of the other metrics.

These relational and quantitative constraints certain qualitative constraints may also be necessitated in order to describe the underlying business logic faithfully. For example, portfolio units may also possess certain qualitative attributes. A feasible portfolio may necessitate a certain percentage of the portfolio to possess a certain attribute. The framework allows for the easy modeling of these kinds of constraints as well. For example, for any qualitative attribute of the portfolio unit, business needs may require all portfolios to contain, in terms of cost or some other metric, a certain percentage to be directed towards portfolio units possessing that characteristic. For example, if $h_i = 1$ indicates that portfolio unit i belongs to a certain category and $h_i = 0$ indicates that it does not, business needs may require at least $\tau_H\%$ of the portfolio's cost to be directed towards units that possess that quality. This can be modeled as such:

$$\sum_{i \in N} \sum_{j \in J_i} c^s_{ij} h_i y_{ij} >= \tau_H \sum_{i \in N} \sum_{j \in J_i} c^s_{ij} y_{ij} \tag{4.16}$$

Solving the model detailed earlier for various values of cost and benefit targets allows us to identify similar risk regions that allows decision makers to initiate conversations regarding trade-offs between the various metrics.

4.4 Numerical Study

A computational study that seeks to validate the framework will have to address a number of different areas detailed in the paper. Since problem complexity depends on the number of relationships impacting a given portfolio unit, experimental studies should characterize how many decision units and optional units the framework can handle before the problem becomes impractical to solve. The second is how many samples need to be generated to effectively characterize the distributions of the primary benefit and cost metrics. Based on the complexity of the model and the shape of the resulting distributions the number of samples and corresponding chance constraints that need to be encoded to generate reasonable answers may vary. This should also be considered while stress testing the capability of the model. Generating results and sweeping across the axis for various values of the budget and revenue target, along with varying values of risk will have to be explored in order to identify what visualizations are most intuitive to decision makers.

For the numerical study, the data we will present corresponds to the problem instance described by Figure 4.1. This problem instance has 11 portfolio units and 22 decision units and 8 relationships. This is illustrated in Table 4.3. For a decision unit, the superscript indicates which portfolio units from the neighborhood are included, and the subscript indicates which decision units from the neighborhood are excluded. The $-$ indicates no portfolio units exist that should be considered.

Portfolio Unit	Neighborhood	Decision Units
A	-	A_{-}^{-}
B	A, C	B_{-}^{AC} B_{C}^{A} B_{A}^{C} B_{AC}^{-}
C	$-$	C_{-}^{-}
D	G, J	D_{-}^{GJ} D_{J}^{G} D_{G}^{J} D_{GJ}^{-}
E	$-$	E_{-}^{-}
F	D, E	F_{-}^{DE} F_{E}^{D} F_{D}^{E} F_{DE}^{-}
G	$-$	E_{-}^{-}
H	K	H_{-}^{K} H_{K}^{-}
I	H	I_{-}^{H} I_{H}^{-}
J	$-$	J_{-}^{-}
K	$-$	K_{-}^{-}

Table 4.3: Optimization Parameters

Each portfolio unit in a portfolio unit's neighborhood impacts said portfolio unit. For each input metric, an ordered triplet (u^l, u^b, u^h) of low, base and high values is reported and used to model the various metric distributions.

Portfolio Unit	Metrics	Reports [l,b,h]
	Volume	[1, 9, 30]
	ASP	[8, 12, 13]
A	Manufacturing Cost	[2, 3, 4]
	Engineering Cost	[1, 6, 7]
	Tax & Depreciation	[3, 4, 5]
	Volume	[20, 30, 35]
	ASP	[5, 6, 8]
B	Manufacturing Cost	[2,3,4]
	Engineering Cost	[5,6,7]
	Tax & Depreciation	[3, 4, 5]
	Volume	[10, 30, 40]
	ASP	[4, 5, 9]
C	Manufacturing Cost	[2, 3, 4]
	Engineering Cost	[5, 6, 7]
	Tax & Depreciation	[3, 4, 5]
	Volume	[10, 20, 30]
	ASP	[4.5, 5, 5.5]
D	Manufacturing Cost	[2, 3, 4]
	Engineering Cost	[5, 6, 7]
	Tax & Depreciation	[3, 4, 5]
	Volume	[5, 20, 21]
	ASP	[4, 8, 10]
E	Manufacturing Cost	[2, 3, 4]
	Engineering Cost	[5, 6, 7]
	Tax & Depreciation	[3, 4, 5]
	Volume	[10, 20, 50]
	ASP	[1, 5, 9]
E	Manufacturing Cost	[2, 3, 4]
	Engineering Cost	[5, 6, 7]
	Tax & Depreciation	[3, 4, 5]

Table 4.4: Portfolio Unit Data for Units A-E

Portfolio Unit	Metrics	Reports [l,b,h]
	Volume	[10, 20, 50]
	ASP	[1, 5, 9]
F	Manufacturing Cost	[2, 3, 4]
	Engineering Cost	[5, 6, 7]
	Tax & Depreciation	[3, 4, 5]
	Volume	[1, 20, 30]
	ASP	[1, 9, 10]
G	Manufacturing Cost	[2, 3, 4]
	Engineering Cost	[5, 6, 7]
	Tax & Depreciation	[3, 4, 5]
	Volume	[13, 52, 70]
	ASP	[4, 5, 12]
H	Manufacturing Cost	[2, 3, 4]
	Engineering Cost	[5, 6, 7]
	Tax & Depreciation	[3, 4, 5]
	Volume	[10, 20, 30]
	ASP	[4, 5, 12]
I	Manufacturing Cost	[2, 3, 4]
	Engineering Cost	[5, 6, 7]
	Tax & Depreciation	[3, 4, 5]
	Volume	[10, 20, 30]
	ASP	[4, 5, 6]
J	Manufacturing Cost	[2, 3, 4]
	Engineering Cost	[5, 6, 7]
	Tax & Depreciation	[3, 4, 5]
	Volume	[10, 25, 26]
	ASP	[5, 10, 15]
K	Manufacturing Cost	[2, 3, 4]
	Engineering Cost	[5, 6, 7]
	Tax & Depreciation	[3, 4, 5]

Table 4.5: Portfolio Unit Data for Units F-K

Relationship	Modifying Metric	Reports [l,b,h]
r_{AB}	Volume Adder	[10, 20, 30]
	ASP Adder	[5, 9, 11]
	Manufacturing Cost Adder	[2, 3, 4]
	Engineering Cost Adder	[5, 6, 7]
	Tax & Depreciation Adder	[3, 4, 5]
r_{CB}	Volume Adder	[11, 12, 13]
	ASP Adder	[4, 5, 6]
	Manufacturing Cost Adder	[2, 3, 4]
	Engineering Cost Adder	[1, 2, 3]
	Tax & Depreciation Adder	[3, 4, 5]
r_{DF}	Volume Adder	[10, 20, 30]
	ASP Adder	[4, 5, 6]
	Manufacturing Cost Adder	[2, 3, 4]
	Engineering Cost Adder	[1, 1.5, 2]
	Tax & Depreciation Adder	[3, 4, 5]
r_{EF}	Volume Adder	[12, 22, 32]
	ASP Adder	[4, 5, 6]
	Manufacturing Cost Adder	[0.5, 0.75, 1]
	Engineering Cost Adder	[1, 2, 3]
	Tax & Depreciation Adder	[5, 6, 7]
r_{GD}	Volume Adder	[10, 20, 30]
	ASP Adder	[4, 5, 6]
	Manufacturing Cost Adder	[2, 3, 4]
	Engineering Cost Adder	[5, 6, 7]
	Tax & Depreciation Adder	[3, 4, 5]
r_{HI}	Volume Adder	[20, 21, 30]
	ASP Adder	[4, 5, 10]
	Manufacturing Cost Adder	[0, 0.5, 1]
	Engineering Cost Adder	[1, 4, 7]
	Tax & Depreciation Adder	[3, 4, 5]
r_{JD}	Volume Adder	[1, 2, 3]
	ASP Adder	[4, 5, 6]
	Manufacturing Cost Adder	[1, 2, 3]
	Engineering Cost Adder	[1, 2, 3]
	Tax & Depreciation Adder	[3, 4, 5]
r_{KH}	Volume Adder	[10, 20, 21]
	ASP Adder	[4, 5, 6]
	Manufacturing Cost Adder	[1, 1.5, 2]
	Engineering Cost Adder	[1, 2, 3]
	Tax & Depreciation Adder	[3, 4, 5]

Table 4.6: Relationship Modifier Data

The reports in Table 4.4, 4.5 and 4.6 are used to fit a PERT-Beta distribution. For each chance-constraint, all decision unit values are calculated by sampling from the input distribution of the portfolio unit and then adding the value to the sample from the distribution for the relevant relationship modifier distribution if one exists. These modified values are then sampled and plugged into Equations (4.1) and (4.2) to produce one sample of the output metrics.

In order to evaluate the various decision units, with the data presented so far, the reported values were assumed to correspond to the low, base and high values of the PERT-Beta Distribution whose probability density function is described by the equation

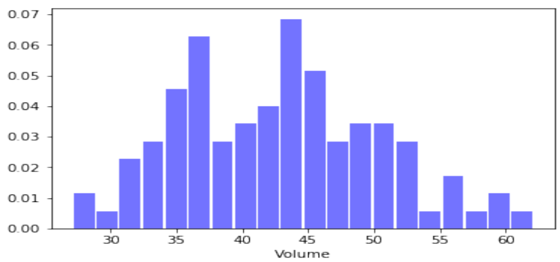
$$\boldsymbol{P(X = x)} = \frac{(x-a)^{(\alpha-1)}(b-x)^{(\beta-1)}}{B(\alpha,\beta)(b-\alpha)^{(\alpha+\beta-1)}}$$

Here, u^l, u^b and u^h correspond to the ordered triplet reported earlier for each portfolio unit metric. α and β are the shape parameters that can be calculated from the low, base and high values using the following equations.

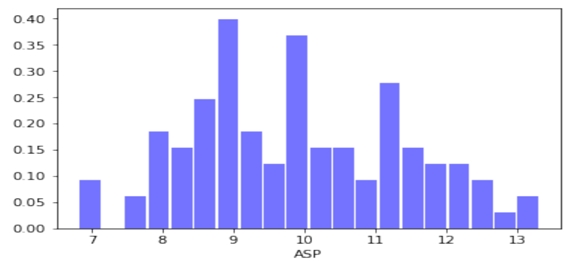
$$\boldsymbol{\alpha} = \frac{4b + h - 5l}{h - l}$$

$$\boldsymbol{\beta} = \frac{5h - l - 4n}{h - l}$$

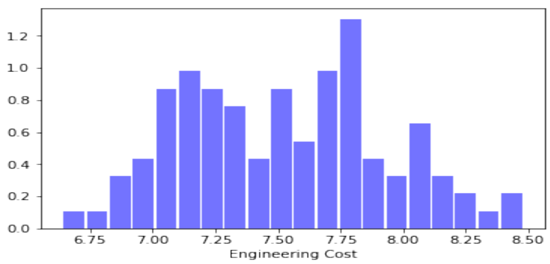
Distributions of Various Metrics for a Decision Unit



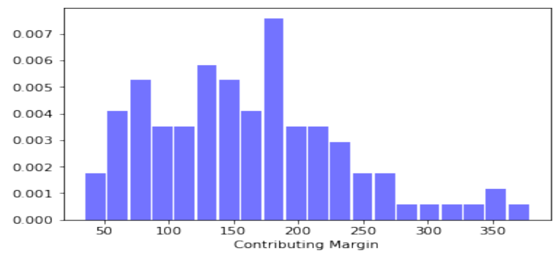
(a) Volume



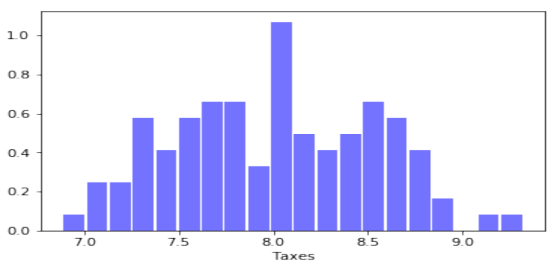
(b) ASP



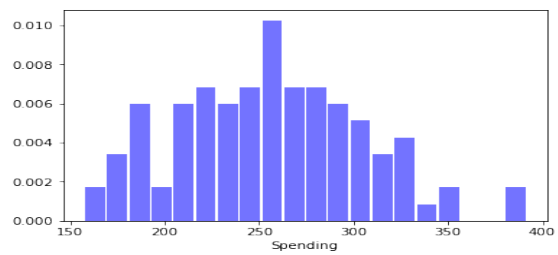
(c) Engineering Cost



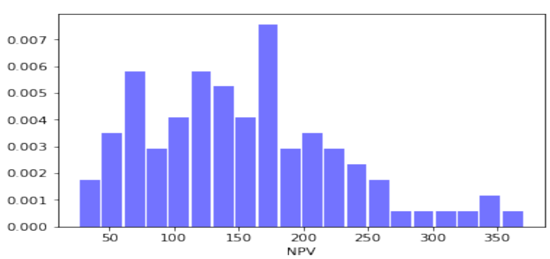
(d) Contributing Margin



(e) Taxes



(f) Cost

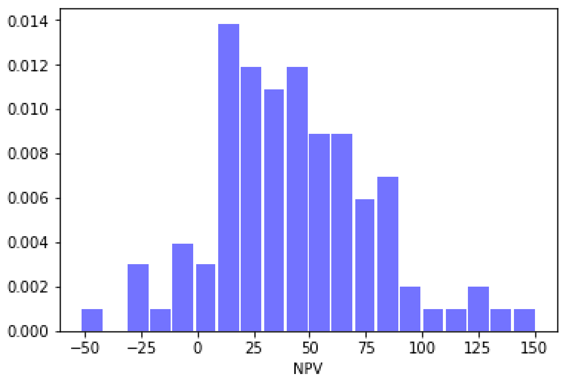


(g) NPV

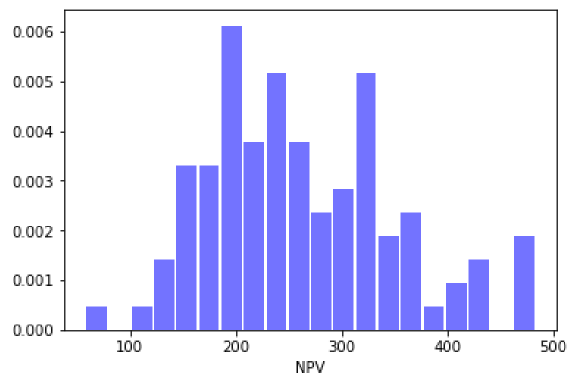
Figure 4.3: Distribution of various metrics of the decision unit of portfolio unit F in a portfolio that includes portfolio unit D, but not portfolio unit J from the problem instance shown in Figure 4.1. The distributions are a result of the probabilistic reports captured for portfolio unit F modified by the probabilistic reports captured for the modifying relationship between D and F

Figure 4.3 shows the distributions for the various metrics for a decision unit of portfolio unit F. The distributions for a particular metric of the various decision units of portfolio unit F may differ significantly as can be observed in Figure 4.4.

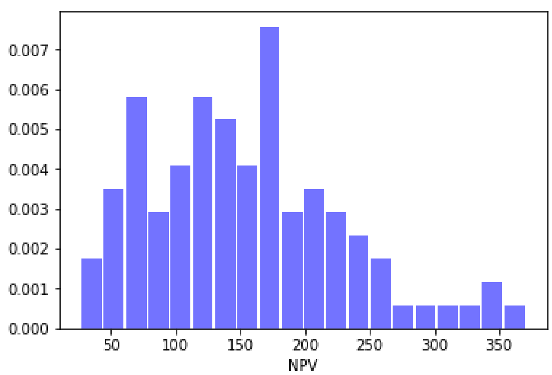
NPV distributions for various Decision Units



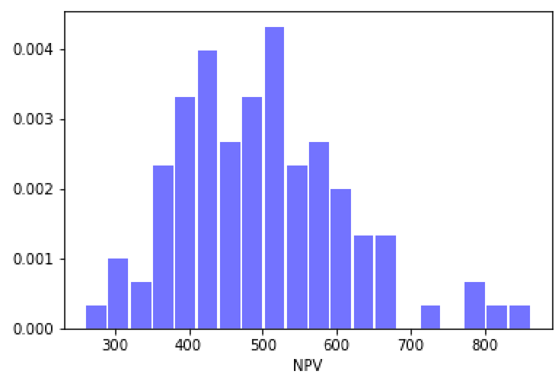
(a) F without D and J



(b) F with J without D



(c) F with D without J



(d) F with D and J

Figure 4.4: NPV Distributions for Various Decision Units of Portfolio Unit F

Solving model MBL for 7 different budget values equally spaced between 1000 and 2000 (1000, 1166, 1332, 1498, 1664, 1830, 1996) and 7 different target NPV values equally spaced between 900 and 2500(900, 1166, 1432, 1698, 1964, 2230, 2496), gives us the bubble plot in Figure 4.5. This is a satisfiability diagram. For any budget and NPV target values, it shows the percentage of chance-constraints that were violated.

This diagram helps orient the decision maker in terms of available options and focus attention on a region of interest, depending on the level of risk she is willing to take on as well as depicting the NPV target and allowable budget used in the model.

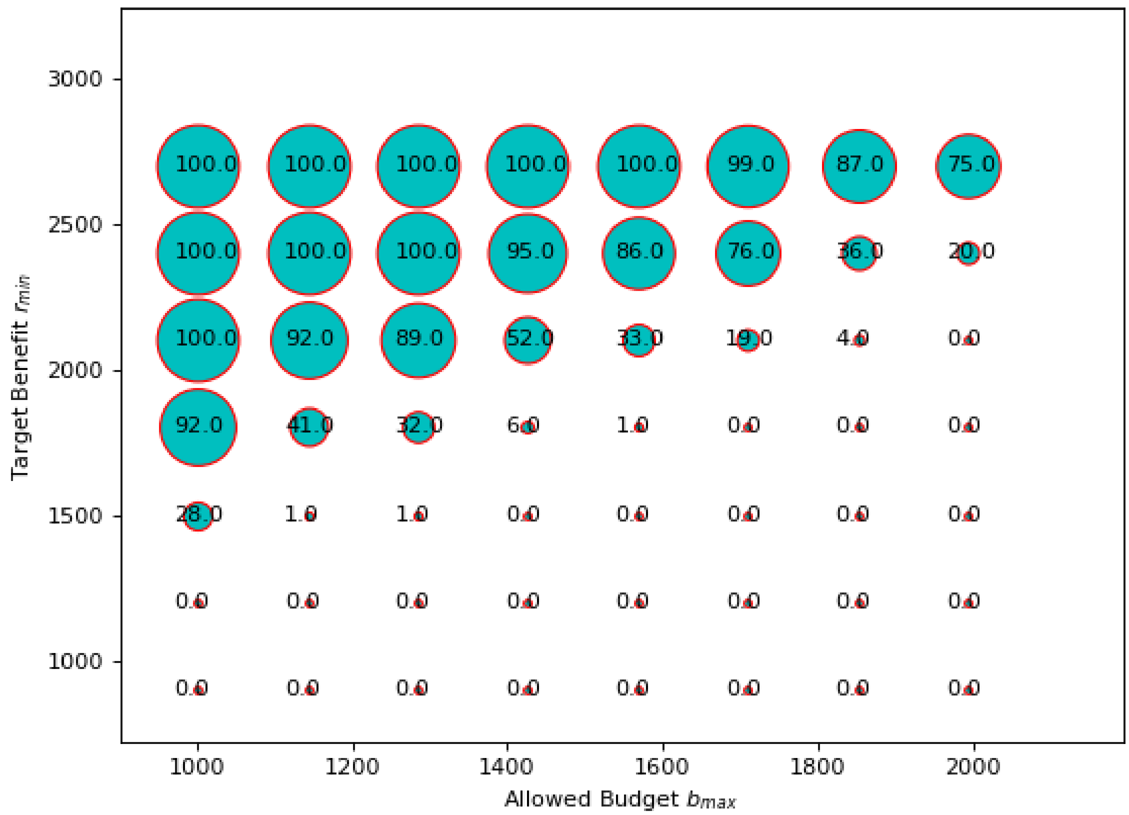


Figure 4.5: Bubble Plot Showing the Number of Chance Constraints Violated for Various Benefit Targets and Allowable Budgets

Solving the BFR for the same values allows us to generate the bubble plots to determine the maximum NPV target realizable for a given budget and risk value. This yields an efficient frontier of portfolios attractive when evaluated from a risk perspective. The risk level to solve at (τ) to choose at various levels of NPV target and budget can be informed by the previous diagram where an existence proof was established at every level for a portfolio that violated the least number of chance-constraints. In Figure 4.6, we observe the efficient frontiers solved for risk levels of $5\%, 25\%$ and 50%.

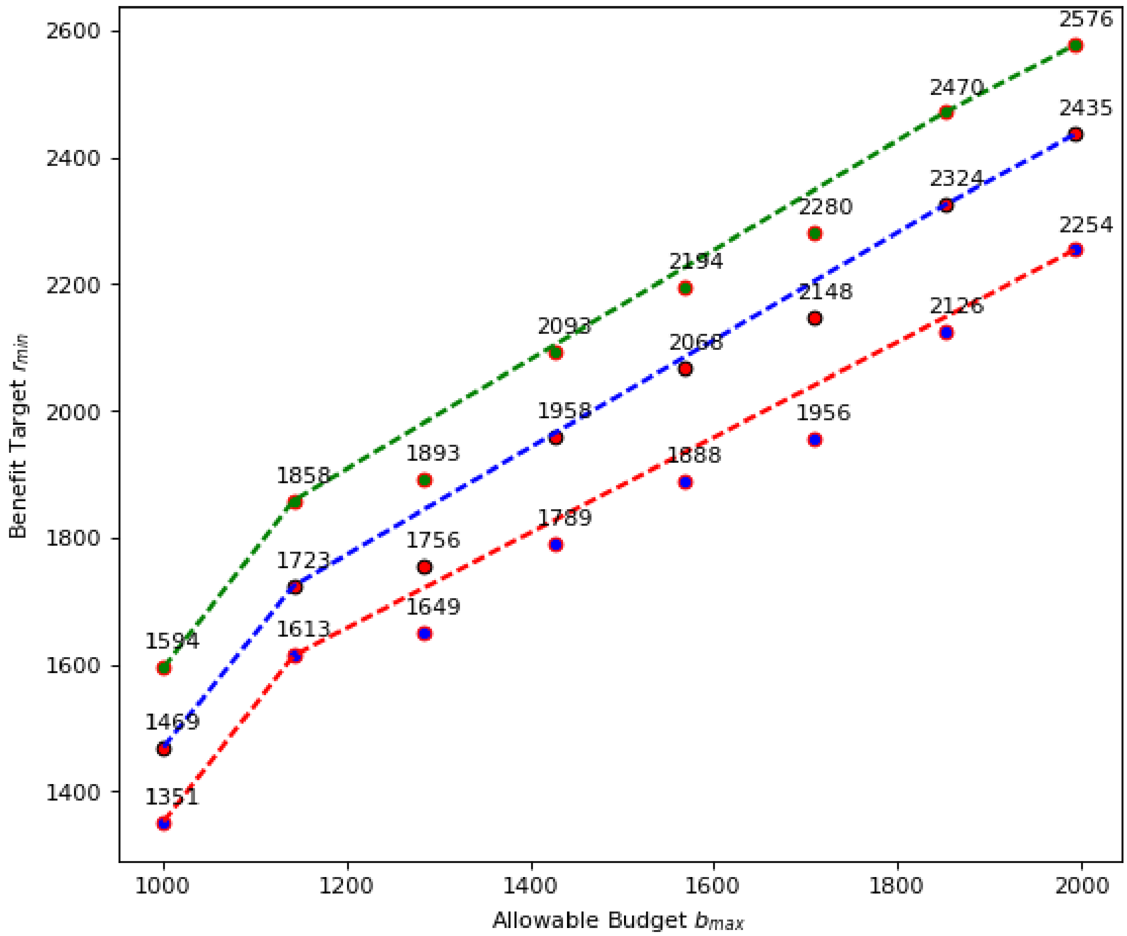


Figure 4.6: Efficient Frontier Plot Showing Portfolios Values for Solutions That Maximize the Target Benefit for a Particular Risk Level at Various Budgets. The Target Revenues Are Displayed. The Top Curve Corresponds to a Risk Level of 50%, the Middle Curve to a Risk Level of 25% and the Lower Curve to a Risk Level of 5%

Figure 4.5 shows the direction of the visualizations that this methodology ultimately aims to provide. For various budget levels and NPV targets, portfolio that minimized the number of chance constraints violated were produced. This gives decision makers some idea of the tradeoffs in the objective space of NPV versus Cost of the kind of risk they incur in increasing their budgets for higher NPV portfolios. Intuitively these provide decision makers of an understanding of the quality of portfolio solutions realizable given the problem instance. The dotted lines connect the convex hull of the best portfolios for every risk level.

The initial bubble plot was solved to determine the best risk level achievable at various target benefit and cost levels. The second figure showed what the best achievable target benefit was achievable by plugging in the best risk level from the first experiment for every cost level. This is the essential difference between the two models.

Extensive testing must be conducted that show the actual trade-offs in the objective space for three different dimensions - benefit, cost and risk.

4.5 Conclusions

Capturing the intuition of senior DMs built over years of experience and the inherent complexities of the nature of PPS itself including the interdependencies and conflicting objectives make this a hard problem. In this work, we seek to present an approach that makes it easier for DMs to interact with a PPS framework to choose project portfolios that are attractive from a risk perspective without being too burdensome on their times in terms of input.

We use joint chance-constraints to present optimization models that present intuitive visualizations that quickly orient the DM regarding the solution space and then explore the solution space to choose the best portfolio.

The framework handles many common problems seen in quantitative optimization models such as the need to gather lots of data, handling lots of inter-relationships and also handling risk considerations. By only capturing a minimal set of reports to characterize the distributions of the various portfolio units, handling inter-relationships via decision units, and using chance-constraints to generate robust solutions, the optimization addresses all the major problems that have traditionally plagued the adoption of such models in industry.

Chapter 5

CONCLUSIONS

This dissertation has primarily sought to present more intuitive frameworks that make the adoption of optimization based project portfolio selection methods more easily adopted at organizations. Traditionally, a variety of other methods such as ad hoc approaches (portfolio profiles, interactive selection) and strategic planning tools (cluster analysis, portfolio matrices) have been preferred over mathematical optimization based approaches. It has long been lamented that optimization models have not gained acceptance in industry, because i) they did not take decision makers' intuitions into consideration, ii) could not handle a large number of inter-dependencies between projects well, iii) needed large amounts of data collected to be useful, iv) did not handle risk considerations in a practical manner.

Chapter 2 presents a problem instance at Intel Inc. and how their product portfolio selection problem was tackled with the aid of a planning framework we designed and the results that came of it. It presents details of the problem instance and describes the problem in some detail. The paper fills a real need in the literature for practical product portfolio selection frameworks.

Chapter 3 of this document is an extension of the mathematical models used in Chapter 2. It generalizes the types of problem instances that the framework presented in the previous chapter can be used for and presents details of all the mathematical models necessary for solving it. These two chapters essentially try to generalize the project portfolio selection formulation to extend it to large problem instance sizes especially when there are lots of interdependencies between projects. The effort was part of a framework along with a decision support tool deployed at a large semi-conductor manufacturing company, which was vital towards buying acceptance into the larger finance community at the company. Chapters 2 and 3 address the challenges of handling large problem instances, developing a general framework for modeling inter-dependencies between projects and allowing decision makers to apply their intuition to the selection process, while the fourth chapter concentrates on handling risk and uncertainty in more detail by presenting a formulation to optimize risk optimized portfolios and incorporate risk considerations into the decision making process. The proof of the success of this framework was its ready acceptance by the large finance community at Intel Inc, and was deployed using a Decision Support System (DSS) the details of which we will discuss briefly alongside our findings.

5.1 Mapping

In the mapping stage or project evaluation stage, the current project roadmap was visualized and all new project and product opportunities were formally stated. Typically, analysts worked on their individual responsibilities for weeks before meeting together to map out the companies roadmap and represent all new portfolio units. Assumptions were stated and viewpoints were calibrated until analysts agreed on all portfolio units, their relationships with each other and all alternatives. The mapping session was usually followed by multiple quality review sessions where details for all mapped units was discussed and passed by a board of experts. Data for each project and its interdependencies was reviewed and closely scrutinized. Analysts stated all assumptions in their model for the review board who either okayed the financial model or asked for it to be reworked. Typically, at least the 10th, 50th and 90th percentile values were captured for each variable. If a certain variable was suspected of being multi-modal, additional percentiles were captured.

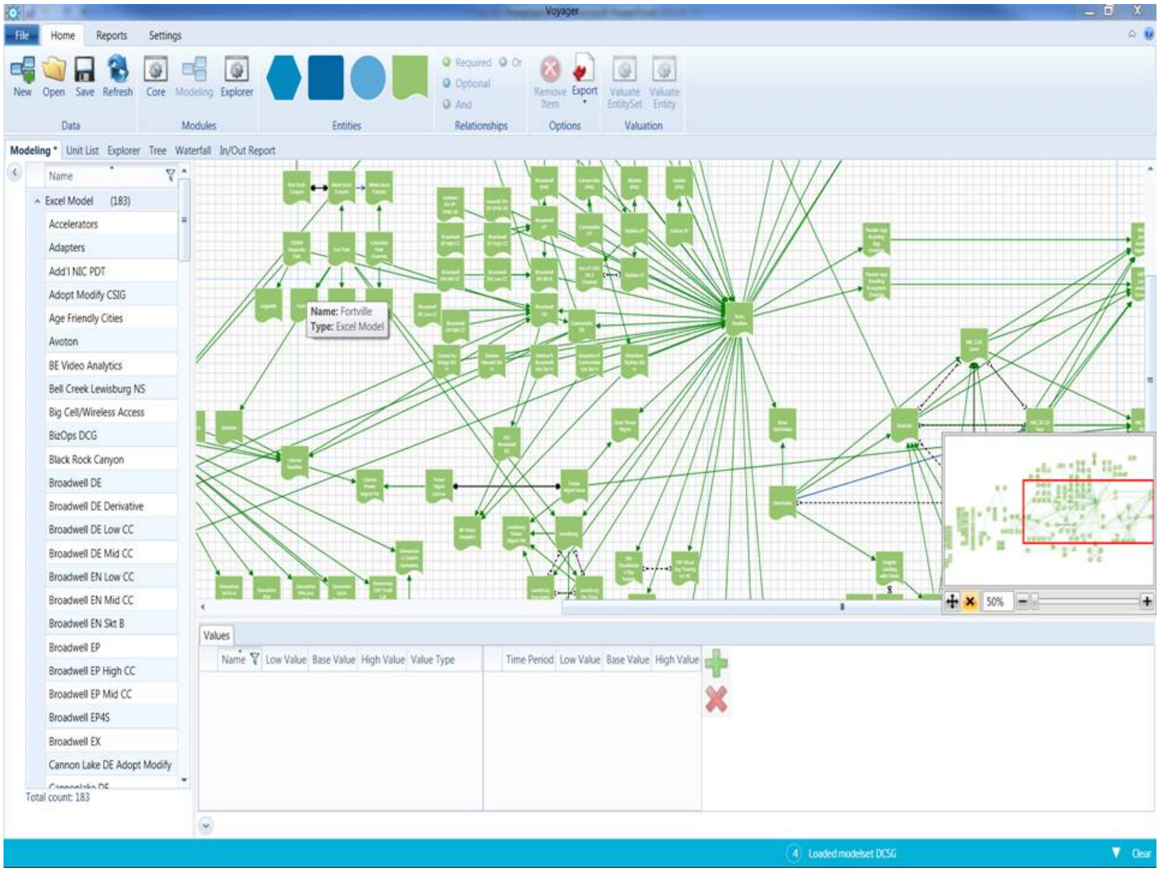


Figure 5.1: Mapped Portfolio Units of a Particular Division

Figure 5.1 shows a part of the project road map for a particular division at Intel Inc. The minimap at the bottom right gives analysts an idea of the overall project road map and the dialogs at the bottom allow the input of project and relationship data. Before the DSS that deployed the framework, the data existed in individual spreadsheet formats and combining the models to make good decisions was cumbersome and in most cases simply impossible within a practical amount of time. This framework implemented by the DSS allowed each project to be further examined along with its inter-dependencies. It allowed easy communication of the organization's options being considered and allowed analysts to better understand final decisions regarding the portfolio selection process which in turn allowed more ready buy-in by everyone.

5.2 Simulation Phase

A Monte Carlo simulation was performed for every valuation of every portfolio unit. Probability distributions and variable correlations captured during the mapping phase were used to simulate various objective and constraint values for all alternatives. This simulation was then visualized and examined to see if they provided hither-to unknown insight to the analyst who could then revise their model. In order to perform the simulation, a number of different curves were fitted to the percentiles captured in the mapping phase and the best fit curve was chosen to represent a particular variable. If the distribution was suspected of being multi-modal, a kernel distribution was also fitted and compared for best fit. The expected values of these distributions were then calculated and fed into the optimization process. Having a systematic framework with a DSS that allowed easy inspection at different steps of the process, and again, allowed easy buy-in from the financial community and allowed for the modeling of uncertainty into the project valuation.

5.3 Optimization Phase

Once key objective and constraint values(decided upon earlier) were obtained for every decision unit, an iterative optimization algorithm was used with special cuts to quickly and efficiently generate all non-dominated portfolios with regard to a primary objective and a primary constraint. Once this was done, an algorithm to detect the upper convex hull of the generated points was implemented and used to identify an efficient frontier. Now, decision makers could inspect individual portfolios and discuss budgets for the fiscal year. Individual decision units could be forced in or out to play "what-if " games and view how portfolios were affected. In order to do this another optimization algorithm was run to generate a portfolio that contained the change indicated and was also the closest match to the original portfolio. Optimizing the cost-benefit ratio of a portfolio was also an important objective that was requested. We achieved this by implementing a binary integer fractional program that maximized the ratio of two different decision unit metrics by employing various integer cuts.

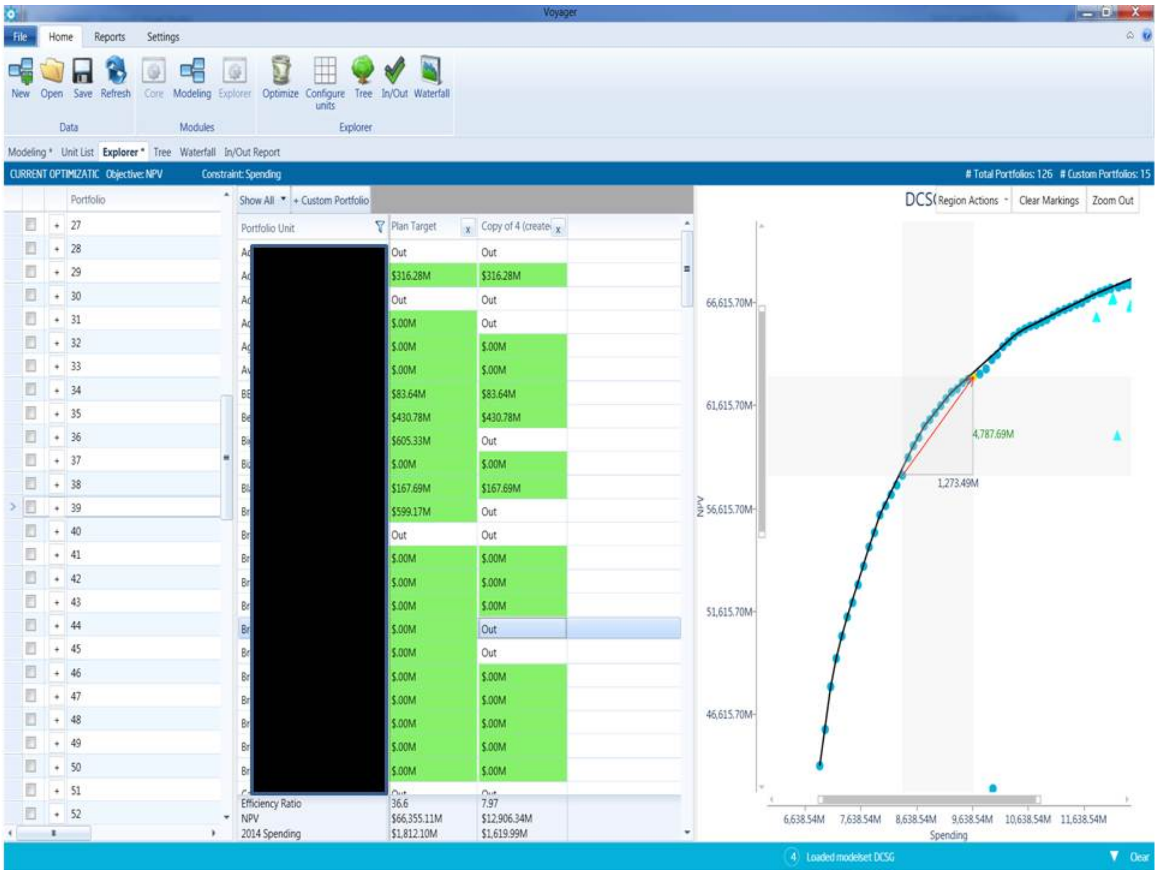


Figure 5.2: Efficient Frontier and Comparison Grid

A waterfall visualization was constructed to contrast portfolios of interest with the help of an optimization algorithm based on a specified metric that builds a path from one portfolio to another by dropping/adding portfolio units from least to highest positive impact. The utility of this exercise was that a decision maker could stop at any given point of the waterfall and still have a completely valid portfolio. This intermediary portfolio often contained characteristics of both the initial and final portfolio used to build the waterfall. It could also be used to construct a path of progress from the existing roadmap to a new one.

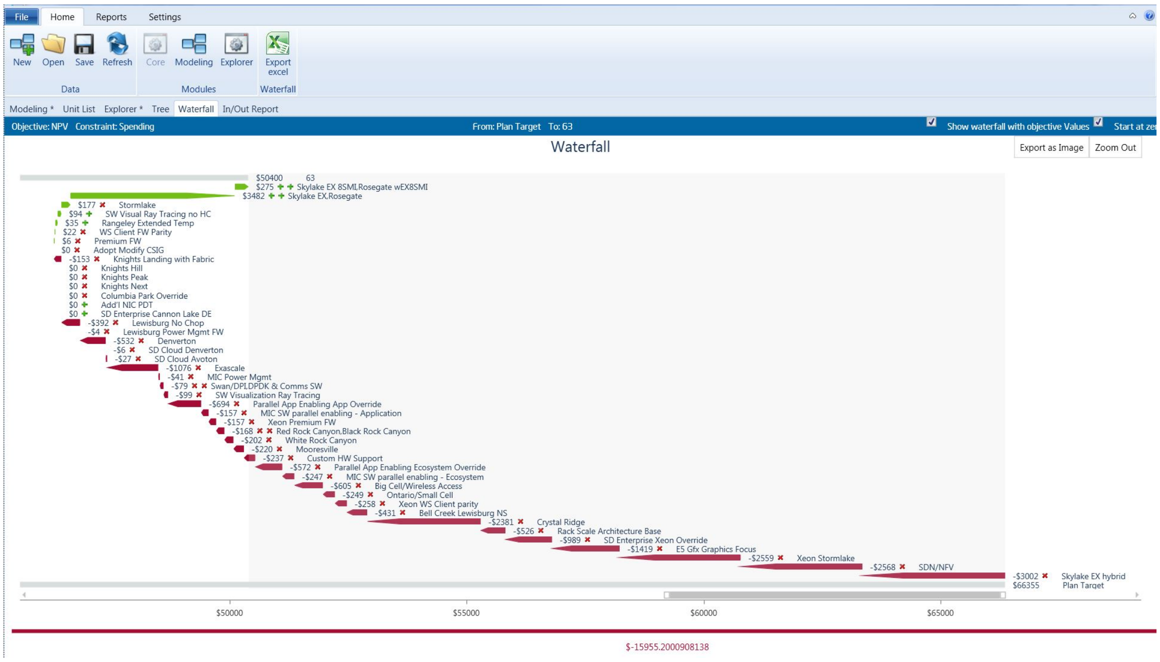


Figure 5.3: Sample Waterfall Between Two Portfolios. The Waterfall at the Top Is Worth Less than the Waterfall at the Bottom. Every Step in the Waterfall Is a Valid Portfolio. Management May Choose to Walk the Entire Path or Stop Midway Between Two Portfolios

As explained before oftentimes decision makers wanted to examine the value of various projects to a specific portfolio when that project was dropped. This "incremental value" could be easily summarized as shown in Figure 5.4.

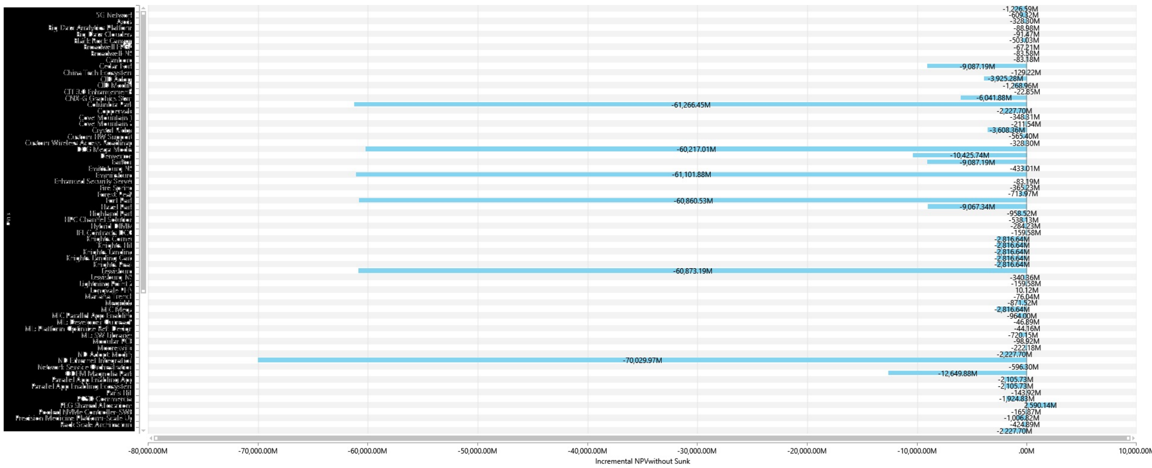


Figure 5.4: The Incremental Value of a Particular Portfolio. Portfolio Unit Names Are Blurred on the Left. On the Right, Each Bar Represents the Magnitude of NPV Lost upon Dropping the Corresponding Portfolio Unit from That Portfolio to Get a New Portfolio. Clicking Any of the Bars Reveals a Waterfall Between the Original Portfolio and the New Portfolio and the Exact Source of the Value Difference Can Be Examined

5.4 Elimination by Aspects

Once the objectives and constraints of interest and limits for these were specified and agreed on, a lexicographical traversal of each objective/constraint with non-satisficing portfolios eliminated at each stage is performed until decision makers are left with a small set of portfolios of interest. At every stage, portfolios were further examined and simulations were run to present to decision makers more information regarding the distributions of specific portfolio attributes. This was followed by much discussion on the state of the projects in the selected portfolios and whether additional alternatives were needed or whether funding should be decreased for certain portfolio units. The mapping and simulation phases were sometimes repeated. Visual aids used to aid decision making by aspects typically involved exporting tool results to spreadsheets by analysts and examining the data in more detail there before summarizing results in presentations to senior management.

5.5 Results

The most time consuming part of the framework was the Mapping phase. This was because much care must be taken in project screening and data gathering. In order to fully defend selected choices to senior decision makers during later phases, utmost care was taken to gather all necessary information and validate it. Once the data was entered and validated however, decision unit generation, simulation and optimization was fast and automated for the most part and completed in reasonable amounts of time. The entire process from project identification to final portfolio selection was shortened from many months to a couple of weeks.

Studies of our reports provided decision makers with a broader and deeper analysis than they had previously been able to access. The use of our model enabled them to satisfactorily address a number of their data-related questions. An evaluation of a number of intuition-based what-if analyses demonstrated the soundness of our recommendations. Perhaps the most important contribution of our decision support system has been to show the highest levels Intel management that the appropriate give-and-take interaction between analytics and intuition produces higher-quality business solutions than either approach can produce individually. The resulting plan is superior in terms of NPV return for the budget dollars invested; in addition, decision makers engage in higher-quality debate (their feedback to the Voyager team) and arrive at consensus in a much shorter time (as measured). Given a receptive but tough-minded group of decision makers, analytics can inform intuition and intuition can inform analytics to the benefit of the business. The transparency of the entire process also led to ready buy-in from all the decision makers involved.

5.6 Conclusions

Capturing the intuition of senior decision makers built over years of experience and the inherent complexities of the nature of PPS itself including the interdependencies and conflicting objectives make it a hard problem. We presented in this work a four phase approach that addresses this problem and was found to be intuitive and attractive to decision makers of all levels. At first, a mapping phase allowed decision makers to screen and map out projects to be included in the planning exercise and explicitly model their interdependencies. This aided in clarifying the data that needed to be gathered and promoted buy-in from project leads who had to supply this information. By clearly specifying the information needed, the data gathering effort was also sped up. In the simulation phase, project alternatives were valuated using the data gathered in the previous phase. An integer program that then generated an efficient frontier was presented. We also presented a novel transformation to generate interesting solutions when the objective function was the ratio between the cost and benefit, and ordinary MILPs could not handle it. Finally a decision making phase made use of an EBA style methodology to identify many interesting portfolios, highlight and communicate the differences between them and better capture actual choice behavior by letting decision makers interact freely with the process. This thesis also generalized the formulation to describe a wide variety of PPS problems and presented a specialized heuristic to quickly generate attractive solutions for large problem instances as well. The framework can handle a large number of projects and effectively addresses many of the concerns raised expressed by many authors in the PPS literature. Finally this thesis examined how to incorporate risk into the optimization framework and generate robust solutions that were attractive to the decision maker.

5.7 Future Work

The field of PPS has started to gain steam, and a lot of that momentum is being generated by the research into quantitative methods, especially optimization models, as organizations begin to adopt 'Big Data' solutions that allow them to collect, store and analyse more and more data. While machine learning methods have been deployed on Big Data repositories, optimization libraries that utilize the advantages of executing in parallel on Big Data do not yet exist. As such more research needs to be done towards parallelizing optimization algorithms to solve larger problem instances faster. This will increase the size of problems that can be handled by the more exact methods and reduce, to some extent, the need to turn to heuristics in identifying the ideal portfolio quantitatively.

When different analysts enter in probabilistic inputs for two different projects, how much the two projects contribute towards the risk of the entire portfolio often depends on how each of the analysts' probability models or in other words, their probability function. In other words, if one analyst has a history of under-weighting small probabilities or over-weighting large probabilities compared to the other, then the likelihood of having his project selected over the other analyst's tends to increase even though the projects may actually be quite similar otherwise. A long history of research in this area has shown however that people's inner model of probability is not linear exacerbating the differences. Moreover, depending on the analyst's risk attitude, it may skew their probability weighting function. How then can we not only elicit an analyst's true beliefs regarding the outcome of their project, but also normalize the estimates from different analysts so that comparing their projects based on their estimates is fairer? Research should be directed towards normalizing different probability functions so that projects compete evenly and the system is not gamed.

More intuitive visualizations will continue to engage senior decision makers in large organizations and allow them to bring their intuition to the decision making process. Focus should also be directed towards more risk measures to execute portfolios that lead to good outcomes more often for the organization. As always, optimization models are only as good as the data that is fed into it. Much care must be taken to ensure that the various inputs into the project portfolio selection model are vetted and trustworthy.

REFERENCES

Abbasianjahromi, H., H. Rajaie, E. Shakeri and O. Kazemi, "A new approach for subcontractor selection in the construction industry based on portfolio theory", Journal of Civil Engineering and Management **22**, 3, 346–356 (2016).

Abbassi, M., M. Ashrafi and E. S. Tashnizi, "Selecting balanced portfolios of r&d projects with interdependencies: A cross-entropy based methodology", Technovation **34**, 1, 54–63 (2014).

Alpaugh, A. D., "A systematic approach to project portfolio selection for economic development in municipalities: a case study in vienna, missouri", (2008).

Altuntas, S. and T. Dereli, "A novel approach based on dematel method and patent citation analysis for prioritizing a portfolio of investment projects", Expert Systems with Applications **42**, 3, 1003–1012 (2015).

Amid, A., S. Ghodsypour and C. O'Brien, "A weighted additive fuzzy multiobjective model for the supplier selection problem under price breaks in a supply chain", International Journal of Production Economics **121**, 2, 323–332 (2009).

Archer, N. and F. Ghasemzadeh, "Project portfolio selection and management", The Wiley guide to managing projects pp. 237–255 (2004).

Archer, N. P. and F. Ghasemzadeh, "Project portfolio selection techniques: a review and a suggested integrated approach", (1996).

Archer, N. P. and F. Ghasemzadeh, "An integrated framework for project portfolio selection", International Journal of Project Management **17**, 4, 207–216 (1999).

Baldi, M. M., T. G. Crainic, G. Perboli and R. Tadei, "Branch-and-price and beam search algorithms for the variable cost and size bin packing problem with optional items", Annals of Operations Research **222**, 1, 125–141 (2014).

Bas, E., "Surrogate relaxation of a fuzzy multidimensional 0–1 knapsack model by surrogate constraint normalization rules and a methodology for multi-attribute project portfolio selection", Engineering Applications of Artificial Intelligence **25**, 5, 958–970 (2012).

Bible, M. J., S. Bivins and S. S. Bivins, *Mastering project portfolio management: a systems approach to achieving strategic objectives* (J. Ross Publishing, 2011).

Charnes, A. and W. W. Cooper, "Chance-constrained programming", Management science **6**, 1, 73–79 (1959).

Charnes, A. and W. W. Cooper, "Programming with linear fractional functionals", Naval Research logistics quarterly **9**, 3-4, 181–186 (1962).

Chien, C.-F., "A portfolio-evaluation framework for selecting r&d projects", R&D Management **32**, 359–368 (2002).

Chih, M., "Self-adaptive check and repair operator-based particle swarm optimization for the multidimensional knapsack problem", Applied Soft Computing **26**, 378–389 (2015).

Costantino, F., G. Di Gravio and F. Nonino, "Project selection in project portfolio management: An artificial neural network model based on critical success factors", International Journal of Project Management **33**, 8, 1744–1754 (2015).

Cruz-Reyes, L., E. Fernandez, P. Sanchez, C. A. C. Coello and C. Gomez, "Incorporation of implicit decision-maker preferences in multi-objective evolutionary optimization using a multi-criteria classification method", Applied Soft Computing **50**, 48–57 (2017).

da Silva, C. G., J. Meidanis, A. V. Moura, M. A. Souza, P. Viadanna, M. R. de Oliveira, M. R. de Oliveira, L. H. Jardim, G. A. C. Lima and R. S. de Barros, "An improved visualization-based approach for project portfolio selection", Computers in Human Behavior (2017).

Deb, K., A. Pratap, S. Agarwal and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: Nsga-ii", IEEE transactions on evolutionary computation **6**, 2, 182–197 (2002).

Dickinson, M. W., A. C. Thornton and S. Graves, "Technology portfolio management: optimizing interdependent projects over multiple time periods", Engineering Management, IEEE Transactions on **48**, 4, 518–527 (2001).

Dombrovskii, V. and T. Obyedko, "Model predictive control for constrained systems with serially correlated stochastic parameters and portfolio optimization", Automatica **54**, 325–331 (2015).

Fonseca, F., M. Lisboa, D. Prata and P. Letouze, "A decision-making technique for financial grant allocation to research projects", International Proceedings of Economics Development and Research **85**, 119 (2015).

Fowler, J. W., E. S. Gel, M. M. Köksalan, P. Korhonen, J. L. Marquis and J. Wallenius, "Interactive evolutionary multi-objective optimization for quasi-concave preference functions", European Journal of Operational Research **206**, 2, 417–425 (2010).

Friedrich, T. and M. Wagner, "Seeding the initial population of multi-objective evolutionary algorithms: A computational study", Applied Soft Computing **33**, 223–230 (2015).

Gafarov, E. R., A. Dolgui, A. A. Lazarev and F. Werner, "A new effective dynamic program for an investment optimization problem", Automation and Remote Control **77**, 9, 1633–1648 (2016).

Geletu, A., M. Klöppel, H. Zhang and P. Li, "Advances and applications of chance-constrained approaches to systems optimisation under uncertainty", International Journal of Systems Science **44**, 7, 1209–1232 (2013).

Ghasemzadeh, F. and N. P. Archer, “Project portfolio selection through decision support”, Decision support systems **29**, 1, 73–88 (2000).

Gigerenzer, G., “Fast and frugal heuristics: The tools of bounded rationality”, Blackwell handbook of judgment and decision making pp. 62–88 (2004).

Gigerenzer, G. and R. Selten, *Bounded rationality: The adaptive toolbox* (Mit Press, 2002).

Goh, J. and M. Sim, “Distributionally robust optimization and its tractable approximations”, Operations research **58**, 4-part-1, 902–917 (2010).

Gomede, E. and R. M. de Barros, “A multicriteria approach to project portfolio selection: Using multiobjective optimization and analytic hierarchy process”, in “2014 9th Iberian Conference on Information Systems and Technologies (CISTI)”, pp. 1–7 (IEEE, 2014).

Grosan, C. and A. Abraham, “Hybrid evolutionary algorithms: methodologies, architectures, and reviews”, in “Hybrid evolutionary algorithms”, pp. 1–17 (Springer, 2007).

Gruia, M., “The efficient frontier technique for analyzing project portfolio management”, A UMT White Paper (November). UMT, New York, NY (2003).

Hall, N. G., D. Z. Long, J. Qi and M. Sim, “Managing underperformance risk in project portfolio selection”, Operations Research **63**, 3, 660–675 (2015).

Harmsen, F., F. Schalkwijk, J. Barjis and S. Overbeek, *Practice-Driven Research on Enterprise Transformation: Second Working Conference, PRET 2010, Delft, The Netherlands, November 11, 2010, Proceedings*, vol. 69 (Springer, 2010).

Heidenberger, K. and C. Stummer, “Research and development project selection and resource allocation: a review of quantitative modelling approaches”, International Journal of Management Reviews **1**, 2, 197–224 (1999).

Henrion, R., “Introduction to chance-constrained programming”, Tutorial paper for the Stochastic Programming Community home page (2004).

Hess, S. W., “Swinging on the branch of a tree: project selection applications”, Interfaces **23**, 6, 5–12 (1993).

Hu, B. and M. López-Ibáñez, *Evolutionary Computation in Combinatorial Optimization: 17th European Conference, EvoCOP 2017, Amsterdam, The Netherlands, April 19-21, 2017, Proceedings*, vol. 10197 (Springer, 2017).

Iamratanakul, S., P. Patanakul and D. Milosevic, “Project portfolio selection: From past to present”, Proceedings of the 2008 IEEE ICMIT pp. 287–292 (2008).

Jaszkiewicz, A., “On the computational efficiency of multiple objective metaheuristics. the knapsack problem case study”, European Journal of Operational Research **158**, 2, 418–433 (2004).

Kalashnikov, V., F. Benita, F. López-Ramos and A. Hernández-Luna, "Bi-objective project portfolio selection in lean six sigma", International Journal of Production Economics **186**, 81–88 (2017).

Krishnan, V. and K. T. Ulrich, "Product development decisions: A review of the literature", Management science **47**, 1, 1–21 (2001).

Kuchta, D., "A generalisation of an algorithm solving the fuzzy multiple choice knapsack problem", Fuzzy Sets and Systems **127**, 2, 131–140 (2002).

Küçükyavuz, S., "On mixing sets arising in chance-constrained programming", Mathematical programming **132**, 1-2, 31–56 (2012).

Levine, H. A., *Project portfolio management: a practical guide to selecting projects, managing portfolios, and maximizing benefits* (John Wiley & Sons, 2007).

Li, C., F. Liu, X. Tan and Y. Du, "A methodology for selecting a green technology portfolio based on synergy", International Journal of Production Research **48**, 24, 7289–7302 (2010).

Li, X., S.-C. Fang, Y. Tian and X. Guo, "Expanded model of the project portfolio selection problem with divisibility, time profile factors and cardinality constraints", Journal of the Operational Research Society **66**, 7, 1132–1139 (2015).

Liberatore, M. J. and G. J. Titus, "The practice of management science in r&d project management", Management Science pp. 962–974 (1983).

Lifshits, A. A. and S. M. Avdoshin, "Algorithms for project portfolio selection based on fuzzy multi-objective model", in "Emerging Trends in Information Systems", pp. 65–77 (Springer, 2016).

Lim, S. M., A. B. M. Sultan, M. N. Sulaiman, A. Mustapha and K. Leong, "Crossover and mutation operators of genetic algorithms", International Journal of Machine Learning and Computing **7**, 9–12 (2017).

Liu, Y. Y. and S. Wang, "A scalable parallel genetic algorithm for the generalized assignment problem", Parallel Computing **46**, 98–119 (2015).

Luedtke, J., "A branch-and-cut decomposition algorithm for solving chance-constrained mathematical programs with finite support", Mathematical Programming **146**, 1-2, 219–244 (2014).

Luedtke, J., S. Ahmed and G. L. Nemhauser, "An integer programming approach for linear programs with probabilistic constraints", Mathematical programming **122**, 2, 247–272 (2010).

Maier, S., A. Street and K. McKinnon, "Risk-averse portfolio selection of renewable electricity generator investments in brazil: An optimised multi-market commercialisation strategy", Energy **115**, 1331–1343 (2016).

Martello, S. and P. Toth, *Knapsack problems* (Wiley New York, 1990).

Mathieu, R. G. and J. E. Gibson, “A methodology for large-scale r&d planning based on cluster analysis”, Engineering Management, IEEE Transactions on **40**, 3, 283–292 (1993).

Mohagheghi, V., S. M. Mousavi and B. Vahdani, “A new optimization model for project portfolio selection under interval-valued fuzzy environment”, Arabian Journal for Science and Engineering **40**, 11, 3351–3361 (2015).

Molina, J. D., J. Contreras and H. Rudnick, “A risk-constrained project portfolio in centralized transmission expansion planning”, IEEE Systems Journal **11**, 3, 1653–1661 (2017).

Nemhauser, G. L. and L. A. Wolsey, *Integer and combinatorial optimization*, vol. 18 (Wiley New York, 1988).

Noorian, F. and P. H. Leong, “Dynamic hedging of foreign exchange risk using stochastic model predictive control”, in “Computational Intelligence for Financial Engineering & Economics (CIFEr), 2104 IEEE Conference on”, pp. 441–448 (IEEE, 2014).

Paquin, J.-P., C. Gauthier and P.-P. Morin, “The downside risk of project portfolios: the impact of capital investment projects and the value of project efficiency and project risk management programmes”, International Journal of Project Management **34**, 8, 1460–1470 (2016).

Paul, P. V., N. Moganarangan, S. S. Kumar, R. Raju, T. Vengattaraman and P. Dhavachelvan, “Performance analyses over population seeding techniques of the permutation-coded genetic algorithm: An empirical study based on traveling salesman problems”, Applied Soft Computing **32**, 383–402 (2015).

Perez, F. and T. Gomez, “Multiobjective project portfolio selection with fuzzy constraints”, Annals of Operations Research **245**, 1-2, 7–29 (2016).

Pfeiffer, J., I. Benbasat and F. Rothlauf, “Minimally restrictive decision support systems”, in “ICIS”, (2014).

Pisinger, D., “Where are the hard knapsack problems?”, Computers & Operations Research **32**, 9, 2271–2284 (2005).

Razi, F. F., A. T. Eshlaghy, J. Nazemi, M. Alborzi and A. Poorebrahimi, “A hybrid grey-based fuzzy c-means and multiple objective genetic algorithms for project portfolio selection”, International Journal of Industrial and Systems Engineering **21**, 2, 154–179 (2015).

Saavedra-Moreno, B., S. Salcedo-Sanz, A. Paniagua-Tineo, L. Prieto and A. Portilla-Figueras, “Seeding evolutionary algorithms with heuristics for optimal wind turbines positioning in wind farms”, Renewable Energy **36**, 11, 2838–2844 (2011).

Sakawa, M., I. Nishizaki and M. Hitaka, “Interactive fuzzy programming for multi-level 0–1 programming problems with fuzzy parameters through genetic algorithms”, Fuzzy Sets and Systems **117**, 1, 95–111 (2001).

Sampath, S., E. S. Gel, J. W. Fowler and K. G. Kempf, "A decision-making framework for project portfolio planning at intel corporation", Interfaces **45**, 5, 391–408 (2015).

Sampath, S., E. S. Gel, J. W. Fowler and K. G. Kempf, "Towards a more intuitive framework for the project portfolio selection problem", Working Paper (2016).

Sampath, S., E. S. Gel, J. W. Fowler and K. G. Kempf, "Towards a more intuitive framework for the project portfolio selection problem", Working Paper (2018).

Sefair, J. A., C. Y. Méndez, O. Babat, A. L. Medaglia and L. F. Zuluaga, "Linear solution schemes for mean-semivariance project portfolio selection problems: An application in the oil and gas industry", Omega **68**, 39–48 (2017).

Shen, S., J. C. Smith and S. Ahmed, "Expectation and chance-constrained models and algorithms for insuring critical paths", Management Science **56**, 10, 1794–1814 (2010).

Sindhya, K., K. Deb and K. Miettinen, "Improving convergence of evolutionary multi-objective optimization with local search: a concurrent-hybrid algorithm", Natural Computing **10**, 4, 1407–1430 (2011).

Stewart, T. J., "Multiple objective project portfolio selection based on reference points", Journal of Business Economics **86**, 1-2, 23–33 (2016).

Stummer, C. and K. Heidenberger, "Interactive r&d portfolio analysis with project interdependencies and time profiles of multiple objectives", Engineering Management, IEEE Transactions on **50**, 2, 175–183 (2003).

Tavana, M., K. Khalili-Damghani and A.-R. Abtahi, "A fuzzy multidimensional multiple-choice knapsack model for project portfolio selection using an evolutionary algorithm", Annals of Operations Research **206**, 1, 449–483 (2013).

Tversky, A., "Choice by elimination", Journal of mathematical psychology **9**, 4, 341–367 (1972).

Vachhani, V. L., V. K. Dabhi and H. B. Prajapati, "Improving nsga-ii for solving multi objective function optimization problems", in "2016 International Conference on Computer Communication and Informatics (ICCCI)", pp. 1–6 (IEEE, 2016).

Wiesemann, W., D. Kuhn and M. Sim, "Distributionally robust convex optimization", Operations Research **62**, 6, 1358–1376 (2014).

Wu, S., J. Toussaint and K. D. Messer, "Maximizing benefits in project selection: a hybrid approach", Applied Economics pp. 1–12 (2017).

Yu, L., S. Wang, F. Wen and K. K. Lai, "Genetic algorithm-based multi-criteria project portfolio selection", Annals of Operations Research **197**, 1, 71–86 (2012).

Yuan, Y., H. Xu and B. Wang, "An improved nsga-iii procedure for evolutionary many-objective optimization", in "Proceedings of the 2014 Annual Conference on Genetic and Evolutionary Computation", pp. 661–668 (ACM, 2014).

Zhang, W.-G., Q. Mei, Q. Lu and W.-L. Xiao, "Evaluating methods of investment project and optimizing models of portfolio selection in fuzzy uncertainty", Computers & Industrial Engineering **61**, 3, 721–728 (2011).

Zhao, P. and Q. Xiao, "Portfolio selection problem with value-at-risk constraints under non-extensive statistical mechanics", Journal of computational and applied mathematics **298**, 64–71 (2016).

APPENDIX A

VARIABLE DEFINITIONS

In this appendix, we define the mathematical models referred to in Chapter 2.

$$C^i_{j,k} = \begin{cases} 1, & \text{if Portfolio Unit } k \text{ is included in Decision Unit } j \text{ of Portfolio Unit } i \\ 0, & \text{otherwise} \end{cases} \tag{A.1}$$

$$E^i_{j,k} = \begin{cases} 1, & \text{if Portfolio Unit } k \text{ is excluded in Decision Unit } j \text{ of Portfolio Unit } i \\ 0, & \text{otherwise} \end{cases} \tag{A.2}$$

N	Total number of portfolio units
I_i	Set of portfolio units that are required for Portfolio Unit i, $i \in \{1, ..., N\}$
E_i	Set of portfolio units that must be excluded when executing Portfolio Unit i, $i \in \{1, ..., N\}$
M_i	Set of mutually exclusive portfolio units, one and only one of which is required to choose Portfolio Unit i, $i \in \{1, ..., N\}$
Q_i	Set of mutually exclusive portfolio units, one and only one of which may be chosen along with Portfolio Unit i, $i \in \{1, ..., N\}$
$m(i)$	Number of portfolio units that define the neighborhood for Portfolio Unit i.
$D(i)$	Total number of valuations for Portfolio Unit i, $i \in \{1, ..., N\}$
$p_{j(i)}$	eNPV of Valuation j of Portfolio Unit i, $i \in \{1, ..., N\}, j \in \{1, ..., D(i)\}$
$s_{j(i)}$	Cost of Valuation j of Portfolio Unit i, $i \in \{1, ..., N\}, j \in \{1, ..., D(i)\}$
B_{min}	Minimum budget that a portfolio must exceed.
B_{max}	Maximum budget that a portfolio cannot exceed.
B_y	Spending of Portfolio y.
$U_{j(i)}$	Any portfolio benefit metric of Valuation j of Portfolio Unit i, $i \in \{1, ..., N\}$
$R_{j(i)}$	Any portfolio cost metric of Valuation j of Portfolio Unit i, $i \in \{1, ..., N\}$
M	Reciprocal of $\min_{j(i)=\{j(i):R_{j(i)}>0, i\in\{1,...,N\}, j\in\{1,...,D(i)\}\}}(R_{j(i)})$
y^a	Portfolio a with respect to which we wish to determine the incremental value of various portfolio units.
$Z_{j(i)}$	Any benefit or cost metric of Valuation j of portfolio unit i.
y_i^a	Portfolio Unit i of Portfolio a.
$x_{j(i)}^a$	Valuation j of Portfolio Unit i of Portfolio a.

Table A.1: Optimization Parameters for the PPS Model

APPENDIX B

ENUMERATING ALL POSSIBLE DECISION UNITS

$x_{j(i)}$	Binary variable indicating if Portfolio Unit j is included in the current scenario of Portfolio Unit i,

Table B.1: Decision Variables to Generate All Possible Decision Units

Where:

$$x_{j(i)} = \begin{cases} 1, & \text{if Portfolio Unit } j \text{ is included in the current scenario of Portfolio Unit } i \\ 0, & \text{otherwise} \end{cases} \tag{B.1}$$

If x' is a valid combination of neighborhood items that clearly define a unique decision unit for the given portfolio unit, we add the following cut to the subsequent iteration:

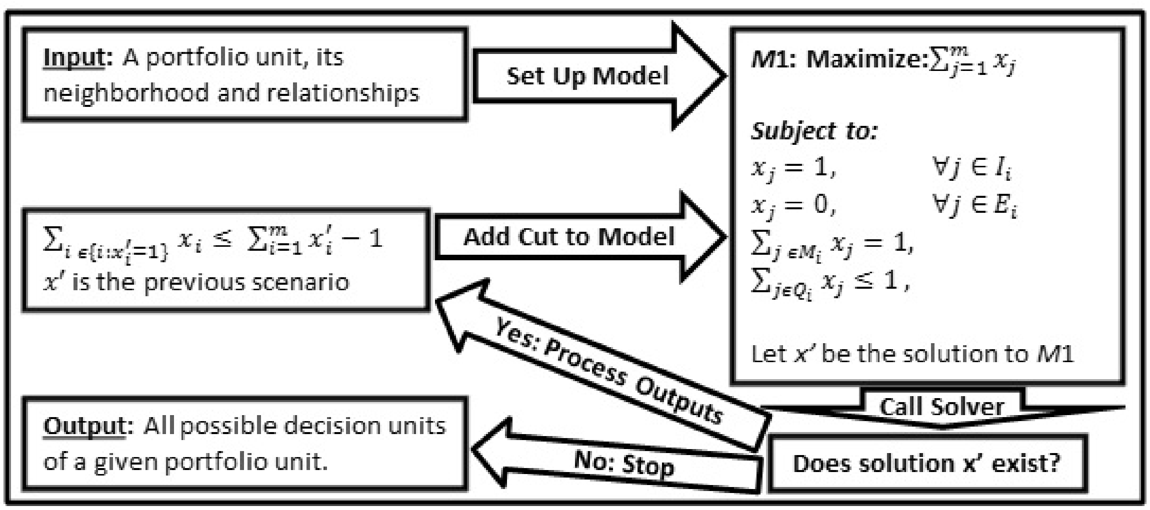


Figure B.1: Given a Portfolio Unit and Its Neighborhood, Including All Relevant Relationships, We Use This Method to Compute All Possible Decision Units of the Portfolio Unit, as We Use in the Mapping Phase of Our Framework

Both $C_{j(i),k}$ and $E_{j(i),k}$ are outputs of the mapping phase.

APPENDIX C

GENERATING NON-DOMINATED SOLUTIONS

$x_{j(i)}$	Binary variable indicating if valuation j of Portfolio Unit i is included in the current portfolio.
y_i	Binary variable indicating if Portfolio Unit i is included in the current portfolio.

Table C.1: Decision Variables for the PPS formulation

Where:

$$x_{j(i)} = \begin{cases} 1, & \text{if Valuation } j \text{ of Portfolio Unit } i \text{ is included in the Portfolio} \\ 0, & \text{otherwise} \end{cases} \tag{C.1}$$

$$y_i = \begin{cases} 1, & \text{if Portfolio Unit } i \text{ is included in the Portfolio} \\ 0, & \text{otherwise} \end{cases} \tag{C.2}$$

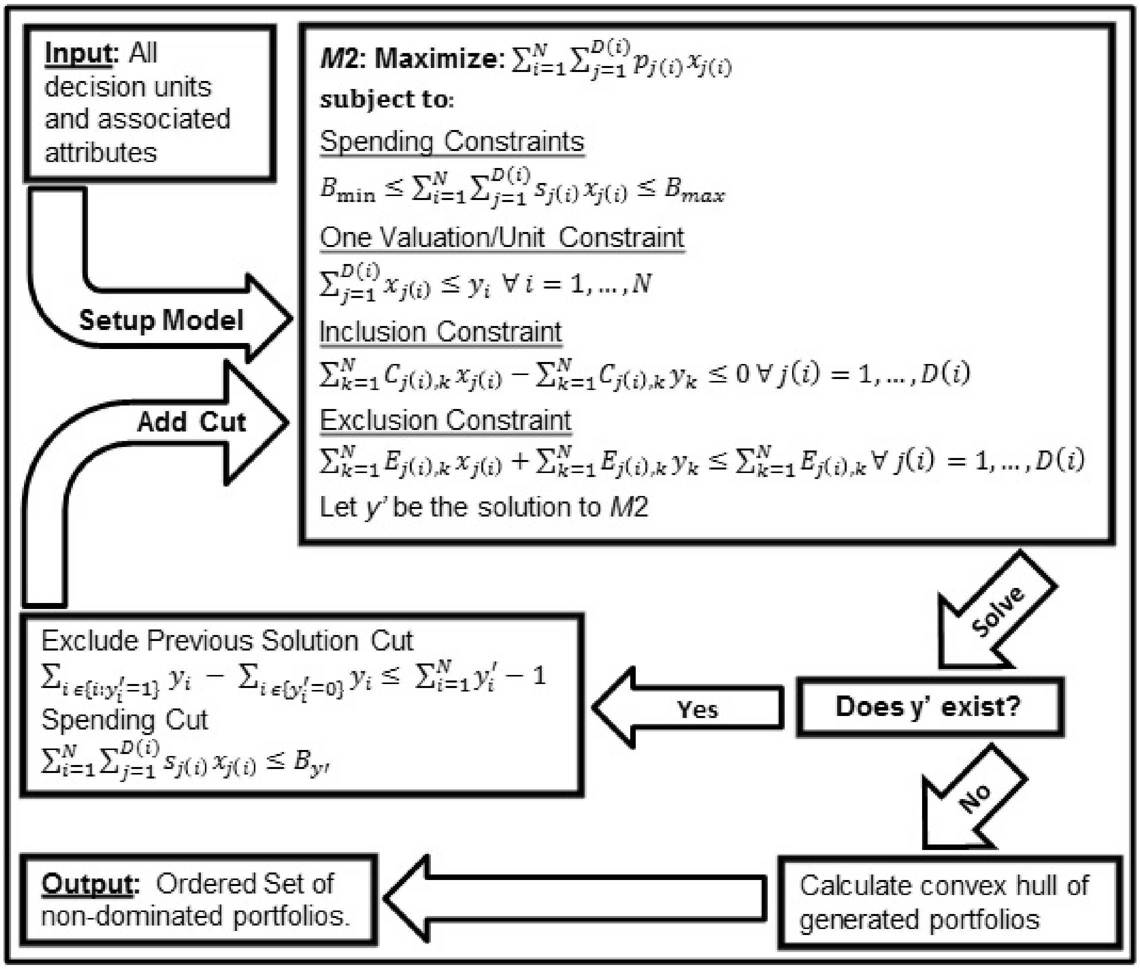


Figure C.1: Given All Decision Units Including Their Associated Attributes, We Use This Method to Compute an Ordered Set of All Nondominated Portfolios and the External Convex Hull of the Set, as We Use in the Optimization Phase of Our Framework

APPENDIX D

EXPLORING SPENDING DURATION

If the original formulation defined in Appendix C had a total of q decision variables given by $q = \sum_{i=1}^{N} D(i) + N$ and the A,$\mathbf{b}$ matrices represent all the constraints in the original optimization, with w being a $q \times 1$ column vector representing the original decision variables $[x_{1(1)}, ..., x_{N(D(N))}, y_1, ..., y_N]^T$, p representing a $q \times 1$ vector with the corresponding eNPVs for variables $x_{1(1)}, ..., x_{N(D(N))}$ and zero otherwise, then the original formulation can be represented as

$$\text{Maximize: } \mathbf{p}^T \mathbf{w} \tag{D.1}$$

$$\text{subject to: } A\mathbf{w} \leq b, \tag{D.2}$$

$$\text{where: } \mathbf{w} \in \{0, 1\}^q \tag{D.3}$$

If we wish to generate additional portfolios to improve the efficiency based on two metrics, (e.g., eNPV and total spending over 10 years), each represented by $qx1$ vectors $\mathbf{U}$ and $\mathbf{R}$ (with nonzero values only for decision units and zero for other variables), then we have the following.

$x_{j(i)}$	Same as Appendix C.
y_i	Same as Appendix C.
$x^z_{j(i)}$	Variable equal to the denominator value of the objective if valuation j of Portfolio Unit i is included in the portfolio and zero otherwise,
y^z_i	Variable equal to the denominator value of the objective if Portfolio Unit i is included in the portfolio and zero otherwise.
t	Variable equal to the reciprocal of the sum of $\sum_{i=1}^{N} \sum_{j=1}^{D(i)} R_{j(i)} x^z_{j(i)} + \beta$

Table D.1: Decision Variables for the Linear Fractional Program to Optimize Cost-Benefit Ratios

Where:

$$x^z_{j(i)} = \begin{cases} \frac{1}{\sum_{i=1}^{N}\sum_{j=1}^{D(i)} R_{j(i)} x^z_{j(i)} + \beta}, & \text{if } x_{j(i)} = 1 \\ 0, & \text{otherwise} \end{cases} \tag{D.4}$$

$$y_i = \begin{cases} \frac{1}{\sum_{i=1}^{N}\sum_{j=1}^{D(i)} R_{j(i)} x^z_{j(i)} + \beta}, & \text{if } y_i = 1 \\ 0, & \text{otherwise} \end{cases} \tag{D.5}$$

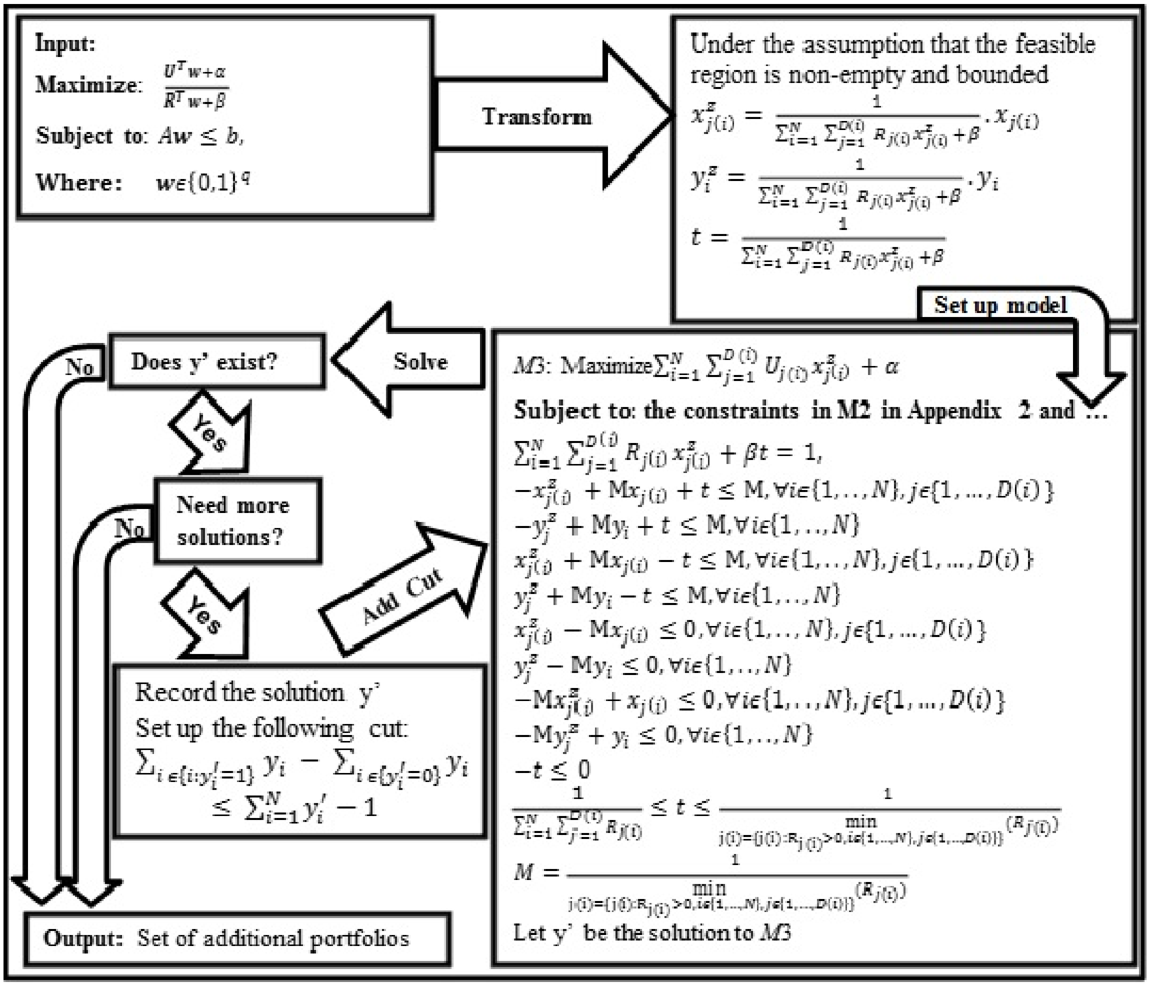


Figure D.1: The Algebraic Formulation We Use to Explore Multiple-year Funding Scenarios Helps Us to Avoid Picking a Portfolio That Looks Attractive in Year 1, but That Contains Projects That Have High Spending in Later Years, Which Would Cause the Portfolio to Exceed Budget

We achieve this mixed-integer linear formulation via the Charnes-Cooper transformation Charnes and Cooper (1962) and add the integer cuts discussed above in this paper. We can include any additional spending constraints and cuts to the $x_{j(i)}$ and the y_i variables and generate multiple portfolios, while still optimizing the ratio using the $x^z_{j(i)}$ and the y^z_i variables, which are continuous. The spending constraints from the original formulation are respected (Appendix C), but solutions are maximized for the new objective. This formulation utilizes the fact that upon solution, all nonzero values of the $x_j(i)^z$ and the y_i^z variable will be equal to t because the original formulation consisted entirely of binary variables.

APPENDIX E

IDENTIFYING THE INCREMENTAL VALUE

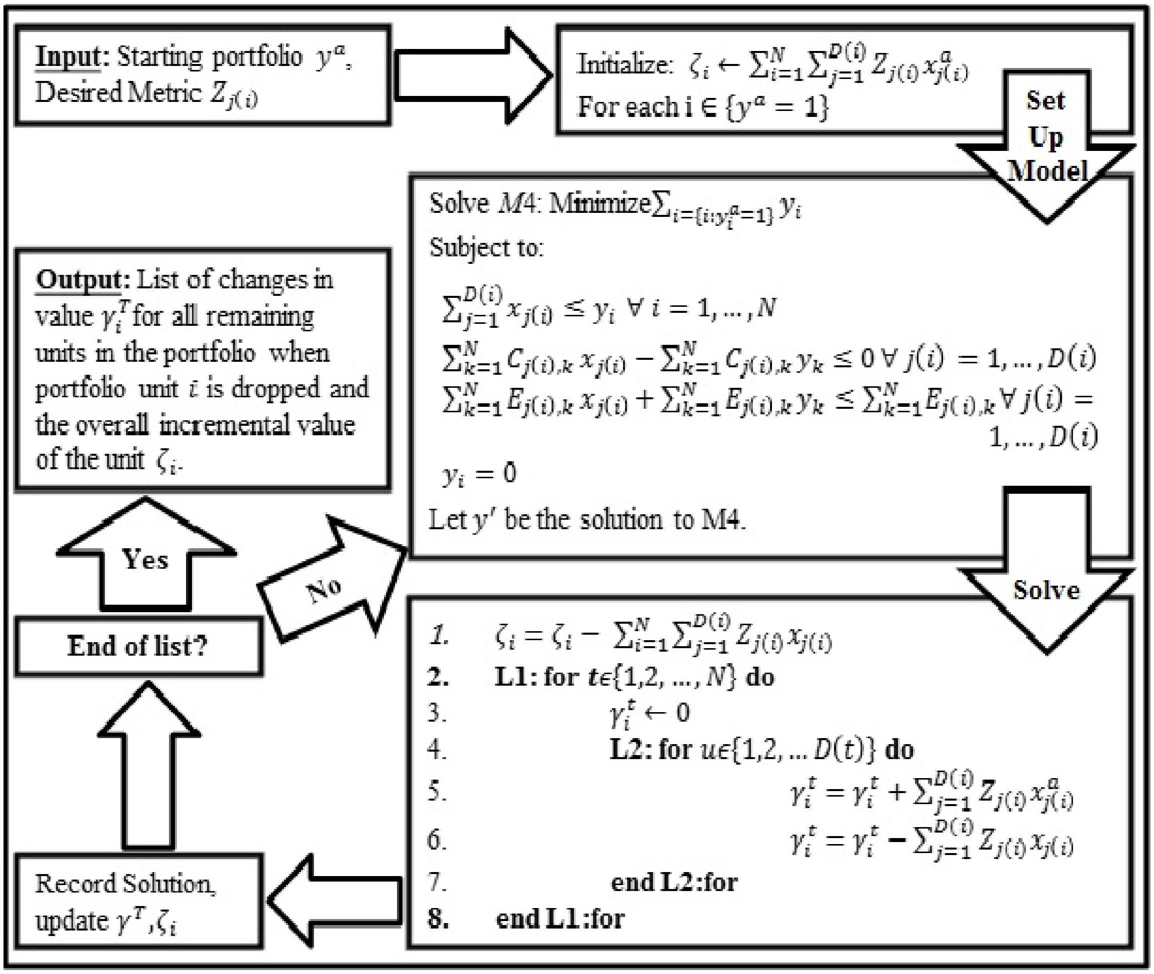


Figure E.1: The Algebraic Formulation We Use to Construct the List of All Changes Associated with the Intuition-based Elimination from a Portfolio of a Particular Unit Is Necessary, Because Eliminating Any Unit Can Have a Variety of Non-Intuitive Impacts on Other Units

We would change to $y_i' = 1$ to see the effects of adding Portfolio Unit i' to the current portfolio.

APPENDIX F

WATERFALL GENERATION

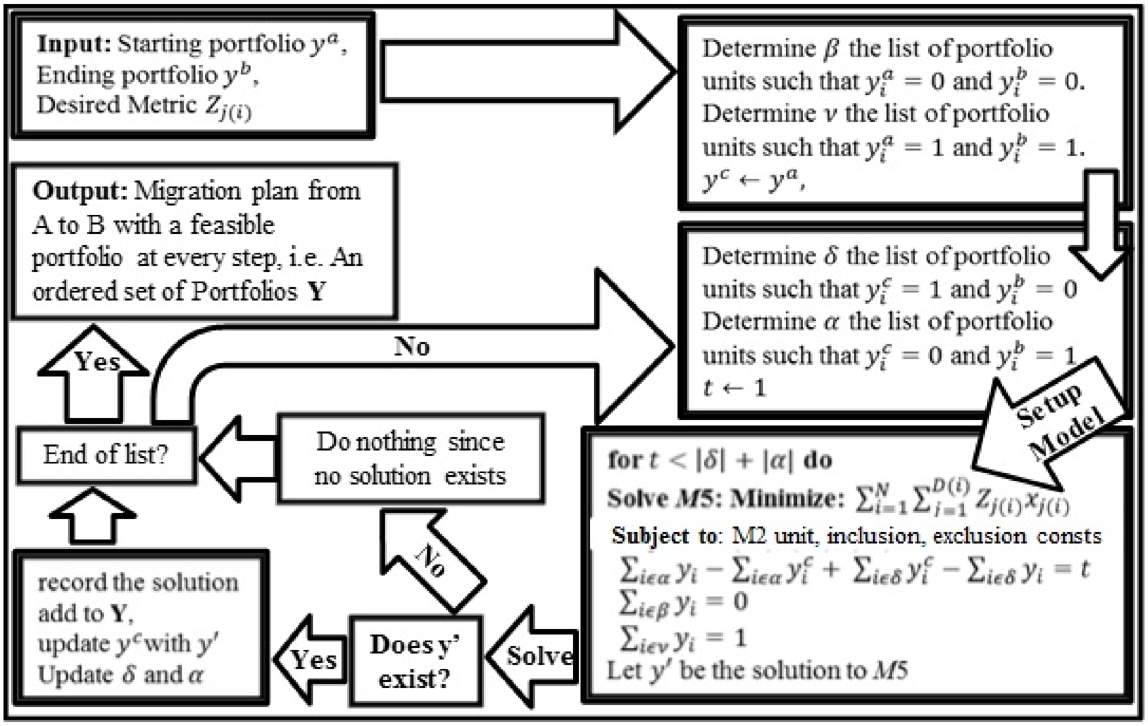


Figure F.1: The Algebraic Formulation We Use to Explore the Step-by-step Transformation of One Portfolio of Interest into Another Is Important Because, When Comparing Two Portfolios, We Must Understand the Step-by-step Addition and Subtraction of Units Required to Convert One Portfolio to the Other

The set of Portfolios Y in essence describes the set of portfolios a feasible migration from y^a to y^b.

APPENDIX G

EXAMPLE PROBLEM INSTANCE

We consider an actual project portfolio selection (PPS) decision faced by Intel Corporation for one of their major product divisions. We model the problem as a Multiple-Choice Knapsack Problem (MCKP) and solve it using our framework. This problem instance attempts to capture the real world business model employed by Division 6 described in the main paper. The names have been removed to protect Intel Corporation's data. Division 6 is one of Intel Corporation's "bread and butter" divisions, in other words, a division where the timings and volumes to be sold of various portfolio units can be predicted with a reasonably high degree of certainty.

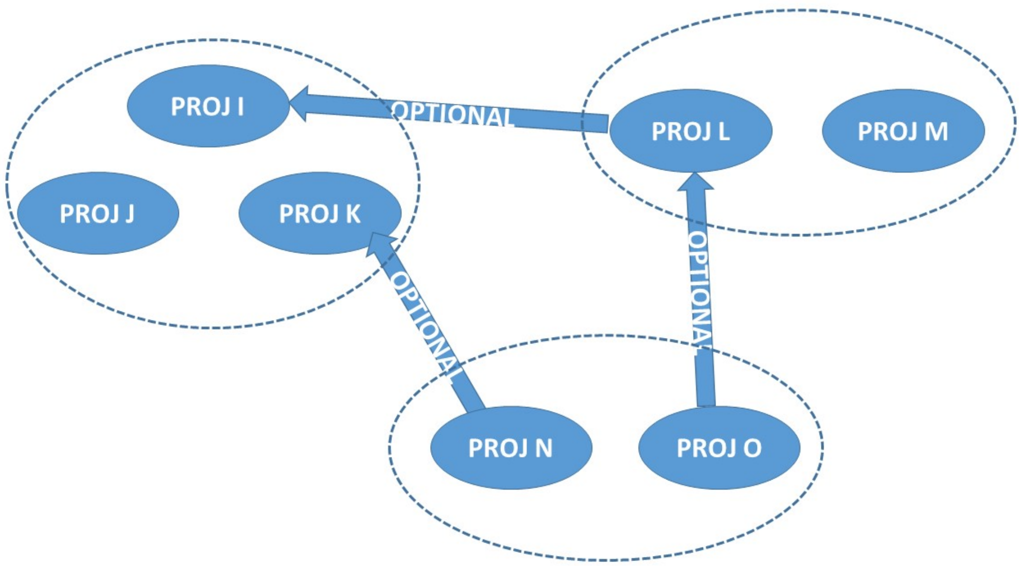


Figure G.1: A Simple Example of a Pps Problem Where Portfolio Units Are Divided into Investment Categories (as Shown by the Dotted Circles). A Valid Portfolio Must Contain One and Only One Portfolio Unit from Each Investment Category

In Figure G.1, we see portfolio units divided into "investment categories" (marked by the dotted sets) that represent a *Hard-or dependency set*. A feasible portfolio must have one and only one portfolio unit from each investment category. Portfolio units may have other relationships with other portfolio units from a different investment category. In this particular division, the *Hard-Or dependency sets* were called investment categories and used to model a particular segment of the business, with each portfolio unit in that investment category corresponding to a different strategy for that segment. For example, in the "high end gaming laptop" division, three portfolio units may represent the strategies "do-nothing", "de-fund" and "increase funding". We will use ${}_1\Delta^H$ to denote the set of all *Hard-Or dependency sets* in this appendix.

We formulate the problem as an integer-program below. The various input parameters and variables for the problem can be summarized as follows. Let N denote the set of all portfolio units, and Δ denote the set of all dependency sets to set up coefficient matrices used to generate the set of all decision units J_i for all $i \in N$. The model can be set up in terms of decision units using coefficient matrices C and E that ensure that the right mix of portfolio units in the neighborhood of portfolio unit i are picked alongside each decision unit selected.

$$C^i_{jk} = \begin{cases} 1, & \text{if Portfolio Unit } k \text{ is included in the valuation of Decision Unit } j \\ & \qquad \text{of Portfolio Unit } i \text{, for all } i, k \in N \text{, and } j \in J_i \ , \\ 0, & \text{otherwise.} \end{cases} \tag{G.1}$$

$$E^i_{jk} = \begin{cases} 1, & \text{if Portfolio Unit } k \text{ is excluded in the valuation of Decision Unit } j \\ & \qquad \text{of Portfolio Unit } i \text{, for all } i, k \in N \text{, and } j \in J_i \ , \\ 0, & \text{otherwise.} \end{cases} \tag{G.2}$$

We define s_i as the spending value of Portfolio Unit i and p_{ij} similarly as the value of the benefit metric (i.e., NPV) of Decision Unit j for Portfolio Unit i. The spending value for all portfolio units within an investment category is assumed to be the same.

The inclusion matrix and the exclusion matrix defined above are used to ensure that for any particular decision unit selected, the appropriate portfolio units that make up the specific scenario for that decision unit are included or excluded accordingly. The decision variables for the model are defined as follows.

y_{ij} Binary variable indicating if Decision Unit j of Portfolio Unit i is included in the current portfolio. That is, $y_{ij} = 1$, if decision unit j of portfolio unit i is included in the portfolio, $y_{ij} = 0$, otherwise

x_i Binary variable indicating if Portfolio Unit i is included in the current portfolio. That is, $x_i = 1$, if Portfolio Unit i is included in the portfolio, $x_i = 0$, otherwise

We can now define the optimization model for using only a primary spending metric and a benefit metric (NPV), and no other qualitative or quantitative constraints, we can define Model M1 as

$$\text{M1: Maximize} \sum_{i\in N}\sum_{j\in J_i} p_{ij}y_{ij} \tag{G.3}$$

subject to:

$$B_{min} \le \sum_{i\in N} s_i x_i \le B_{max} \tag{G.4}$$

$$\sum_{j\in J_i} y_{ij} \le x_i, \qquad \forall i \in N \tag{G.5}$$

$$\sum_{k\in N} C^i_{jk} y_{ij} - \sum_{k\in N} C^i_{jk} x_i \le 0, \qquad \forall i \in N,\ j \in J_i \tag{G.6}$$

$$\sum_{k\in N} E^i_{jk} y_{ij} + \sum_{k\in N} E^i_{jk} x_i \le \sum_{k\in N} E^i_{jk}, \qquad \forall i \in N,\ j \in J_i \tag{G.7}$$

$$\sum_{i\in {}_1\delta^H_l} x_i = 1, \qquad \forall\ {}_1\delta^H_l \in\ {}_1\Delta^H \tag{G.8}$$

$$x_i\ ,\ y_{ij} \in \{0,1\} \qquad \forall i \in N,\ j \in J_i$$

Any other relationships that exist can be added to Model M1. Given the problem structure, and the problem instance encountered at Intel with an α value of 36000, it is not possible to enumerate and valuate all of the decision units in a reasonable amount of time and solve the problem using commercial integer programming solvers on even powerful servers. Moreover the amount of memory storage required to store the decision units is also prohibitively large. However, we can model the problem using only the portfolio units as shown below by dropping equations (G.5),(G.6) and (G.7). The NPV is now returned by a function, that for every portfolio first computes which decision units constitute the portfolio, and then compute the value of the portfolio. This is shown in Model M2. We can now use BIG to solve the Integer program for various spending ranges represented in Model M2.

M2: Maximize $f(x)$

subject to:

$$B_{min} \leq \sum_{i \in N} s_i x_i \leq B_{max} \tag{G.9}$$

$$\sum_{i \in {}_1\delta_l^H} x_i = 1, \qquad \forall \; {}_1\delta_l^H \in \; {}_1\Delta^H \tag{G.10}$$

$$x_i \in \{0, 1\} \qquad \forall i \in N$$

This formulation is a simplified version of the model in Sampath *et al.* (2015) by the fact that it excludes “decision units.” Constraint set (G.9) ensures that portfolios lie within a specified spending range, while Equation (G.10) ensures that only one portfolio unit from each investment category is chosen. The primary difference between M1 and M2 is that while using the genetic algorithm to solve M2, we do not generate all the decision units beforehand. We generate them on-the-fly, and then compute the value of it at a point where we know every other portfolio unit included in the current portfolio candidate solution. This is the reason all decision units need not be enumerated and valuated *a priori*.If the spending for a portfolio unit is fixed and all decision units share the same spending value i.e. it is a function of the portfolio unit and not the decision unit, we calculate the benefit of a portfolio by determining the corresponding decision units that were realized for each portfolio unit, valuating each decision unit, and then the portfolio itself. Appendix B discusses encoding schemes, efficiently seeding the genetic algorithm and genetic algorithm operators.

APPENDIX H

BEAM SEARCH IMPLEMENTATION FOR GENERATING A PRELIMINARY EFFICIENT FRONTIER

Before we discuss the beam search algorithm, we present two different data representations we could use.

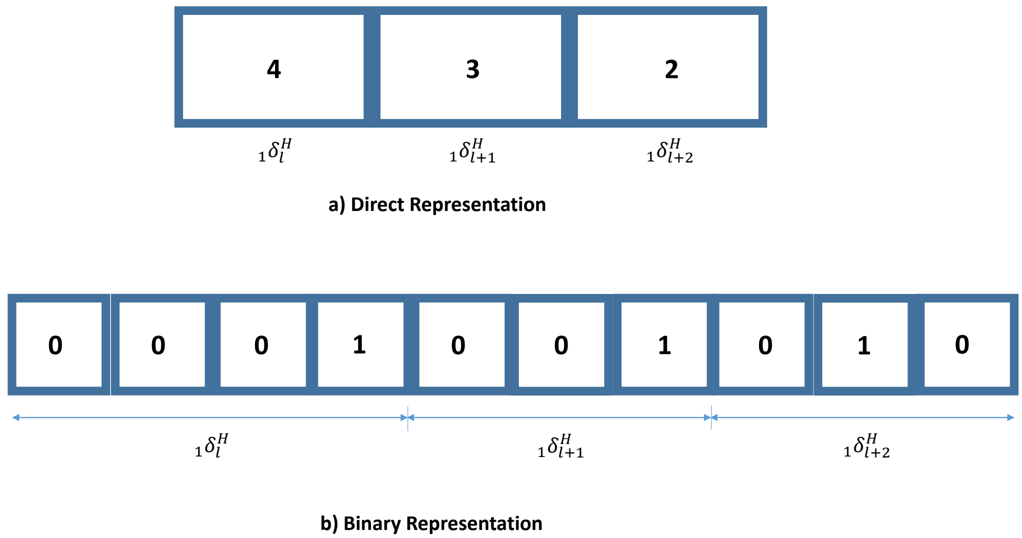


Figure H.1: Two Possible Chromosomes for the Genetic Algorithm Proposed to Solve the MCKP. (a) in the Direct Representation, Each Allele Is an Investment Category and the Max Number Possible in Each Allele Is the Cardinality of the Set of Options in That Investment Category. (b) in the Binary Representation, Each Allele Is a 0-1 Variable That Shows Whether the Corresponding Option Is Selected or Not. Each Investment Category Has a Certain Number of Alleles Assigned to It Equal in Number to the Number of Options It Contains.

In Figure H.1, we see two types of possible chromosomes that could be utilized to implement a genetic algorithm to generate feasible and interesting portfolios. Figure H.1(a) shows a "direct representation" while Figure H.1(b) shows the "binary representation." For example, with three investment categories each with four, three and three options respectively, a portfolio that chooses the fourth, third and second option from each category can be represented in two ways because the length of the solution is always fixed. A representation of this example is given in Figure H.1.

One of the advantages of the binary representation is that it makes it easy to test whether a solution is feasible or not, since if we represent the problem as an integer program and generate the relevant matrices, then it is a trivial matter to check whether a given solution satisfies all the constraints. Moreover, one way of seeding or generating an initial population is to repeatedly solve an integer program with cuts. The disadvantages of this form of encoding is the potential to generate more infeasible solutions and stricter regulation on the generation of new populations because it is much bigger than the direct representation encoding. Furthermore, it needs translation before a decision maker can make sense of the solution.

The direct representation, apart from being much more compact, also has the added advantage of being easy to manipulate and be consumed by the Decision Maker. This is important because there are often last minute changes that need to be input as more information rolls in regarding option variables. However, explicit routines need to be written to check if a solution is feasible. Moreover, generating initial solutions is also more difficult because the same feasibility routines have to be run every time during solution generation. This is especially a problem the more relationships there are between various investment categories or portfolio unit groups.

For the binary representation, the allocated units of a particular investment category are encoded by a segment of $|\delta_l^H|$ bits, or the number of options in that investment category. The chromosome's length is $\sum_{\{\delta_l^H \in \Delta^H\}} |\delta_l^H|$. In the direct representation, there are only $|\Delta^H|$ bits, so the search space is narrower. Mutation is essentially the same for both representations. A random option within each investment category is selected for mutation. Crossover, on the other hand, is much easier for the direct representation and is trivial. For the binary representation an additional vector recording the indexes of various groups must be maintained to produce a meaningful crossover. Possible repair operators must be applied to ensure that every child describes a feasible solution.

For the set of investment categories L, assuming a direct mapping from L to Δ^H, and that portfolio units are ordered by increasing resource metric $s_{l,1} < s_{l,2} < ...s_{l,|\delta_l^H|}$ within each investment category $|\delta_l^H|$ for all $l \in L$, we can now describe the implementation of the beam search algorithm. Using the direct representation in Figure H.1, we define a solution to the MCKP as a vector z defined by $|L|$ digits where the value of the digit in the l^{th} position indicates the option u within investment category l that was included in the solution to the MCKP. Thus, $z_1 = 2$ would indicate that in the solution to the MCKP defined by z, option 2 was selected from investment category 1. This is the first chromosome displayed in Figure H.1.

Using this definition, we now define the beam search algorithm inputs, z^{+} and z^{-} as follows

z^{+}	:=	portfolio with maximum spending where, $z_l^{+} = \lvert\delta_l^H\rvert$,	$\forall\, l \in L$
z^{-}	:=	portfolio with minimum weight or spending where, $z_i^{-} = 1$,	$\forall\, l \in L$
$\mathbf{Z}$	:=	the set of feasible solutions	

The details of the beam search implementation are given in Algorithm 1. The result of the beam search is a preliminary efficient frontier of solutions that will be used to seed the genetic algorithm.

The worst-case complexity of the beam search depends on the beam width, or the number of solutions retained at each level (i.e., β), and the depth of the tree or the maximum number of one step transitions from z^{+} to z^{-}. This can be calculated using the following logic. Suppose we pick a path from the root node of the beam search tree z^{+}, and at every level of the tree, adjust the level of one of the groups down in weight, moving on to the next group once we reach the option with the lowest weight in that group and repeating this process until we reach z^{-}. Then, this is the longest absolute path that we can take (because any other path would mean altering a level at a previous level of the beam search, which is disallowed). The length of this path can be easily calculated by summing up all options as $\sum_{l \in L} \sum_{k=1}^{1\delta^H} 1$. Thus, the complexity of the beam search algorithm is given as $O(\beta \sum_{l \in L} \sum_{k=1}^{1\delta^H} 1)$.

There are two other methods by which a rough initial frontier can be generated. One is to set up the model M2 described earlier, but setting the objective function to zero. Then, by varying the right hand side of Equation G.9 from B_{max} to $B_{min} + \epsilon$, we can generate feasible portfolios at different spending levels. Furthermore, storing the coefficient matrices for M2 also provides an inexpensive way to check if a given solution is feasible. This algorithm is detailed in Algorithm 2. Here we note that if x' is a feasible solution to model M2 for any iteration of the algorithm, we need to add a cut to M2 that removes the solution x' from the feasible region. One such cut is given in Equation (H.1). Another way is to update B_{max} to be less than the spending of the portfolio defined by x' by a certain amount that must be specified. The set of results produced by the beam search consistently outperforms this lazy method, but we present it here as a methodology to generate new feasible solutions.

$$\sum_{\{i: x'_i = 1\}} x_i \leq \sum_{i \in N} x'_i - 1 \tag{H.1}$$

Algorithm 1 Generating an Approximate Efficient Frontier Using a Beam Search

procedure GENERATE EFFICIENT FRONTIER
    Input: Set of Investment Categories L, and the corresponding set of Hard-Or Dependency sets that represents the investment categories ${}_1\Delta^H$
    Sets of portfolio units in each investment category ${}_1\delta_l^H, {}_1\delta_l^H \in_1 \Delta^H$
    Associated constraint metric s_{lk} ordered $s_{l1} < s_{l2} < ...s_{l|_1\delta_l^H}$ within each group $l \in L$.
    Function $f(.)$ to evaluate a solution.
    Parameter β: Number of solutions to be retained at each level of the beam search
    Output: χ A distinct set of non-dominated portfolios and
    $\mathcal{E}$, the subset of points that define the efficient frontier
    $\chi \leftarrow \{z^{+}\}$
    $\chi^1 \leftarrow \{z^{+}\}$
    $a \leftarrow 0$
    $b \leftarrow |\chi^0|$
    while $x^{-} \notin \chi$ **do**
        for $c := c \in \chi^a$ **do**
            $z^d \leftarrow c$
            for $l := l \in L$ **do**
                if $z_l^d > 1$ **then**
                    $z_l^d = z_l^d - 1$
                    $\chi^{a+1} \leftarrow \chi^{a+1} \cup z^d$
            Evaluate all solutions in χ^{l+1} using $f(.)$
            Order χ^{a+1} to be monotonically decreasing in f
            Update χ^{a+1} discarding all infeasible solutions
            Update χ^{a+1} discarding all dominated solutions
            if $|\chi^{l+1}| < \beta$ **then**
                Discard last $|\chi^{a+1}| - \beta$ solutions from χ^{a+1}
                $\chi \leftarrow \chi \cup \chi^{a+1}$
            $a \leftarrow a + 1$

    Calculate subset of points defining the convex hull in χ and assign it to $\mathcal{E}$

Algorithm 2 Generating an Approximate Efficient Frontier Using Integer-Program

procedure GENERATE EFFICIENT FRONTIER
  Input: Set up Model M2 with objective function $\mathbf{0}$, Maximum window parameter ω
  Output: χ A distinct set of non-dominated portfolios and $\mathcal{E}$, the subset of points that define an efficient frontier
  Set the RHS of Equation G.9 to B_{max}
  $\chi \leftarrow \emptyset$
  while `There exists` x' `a valid solution to M2` **do**
    $\chi \leftarrow \chi \cup x'$
    Update the RHS of Equation G.9 to $\sum_{\{i:x'_i=1\}} s_i$
  Update χ discarding all dominated solutions
  Calculate subset of points defining the convex hull in χ and assign it to $\mathcal{E}$