CIFE CENTER FOR INTEGRATED FACILITY ENGINEERING

Design Project Optimization

By

John Chachere

CIFE Working Paper #095
DECEMBER 2004

STANFORD UNIVERSITY

If you would like to contact the authors, please write to:

c/o CIFE, Civil and Environmental Engineering Dept.,
Stanford University
Terman Engineering Center
Mail Code: 4020
Stanford, CA 94305-4020

DESIGN PROJECT OPTIMIZATION

This paper presents a quantitative algorithm that jointly optimizes diverse and interdependent decisions that shape a collaborative engineering venture of limited scope and duration. Planning decisions include product component and subsystem configuration; organizational participants and structure; processes of design, collaboration, and testing; and environmental elements including design norms, incentives, and facilities. We formulate project effectiveness and constraints in algebraic, domain-extensible terms, and integrate within the nonlinear optimization two established methods that originally addressed isolated problem aspects.

We use the Virtual Design Team (VDT) [Cite] to simulate an information processing view of organizational behavior [Galbraith 1972] with precision that exceeds most competing theories. VDT tests design organizations' information processing capacities against processes' information processing loads to forecast emergent distributions of project duration, labor costs, process quality, and organizational character.

We use Probabilistic Risk Analysis (PRA) to examine products (such as the NASA space shuttle) whose dependability is difficult to assess. PRA calculates a product design's reliability by decomposing it conceptually into functional blocks, assessing component and subsystem reliabilities, and aggregating to a total failure probability using fault trees. We base components' design error probabilities on VDT's predicted exception handling and communications behavior.

To improve legibility, variables and functions are named verbosely and modularized. Quantities collected directly from domain experts are <u>underlined</u>, values derived from explicit formulae are **bold**, and formulae that apply to many parallel instances of a variable use $_{\text{subscripts}}$. External modules such as VDT and PRA identify input with *Italics* and output with ***Bold Italics***.

We are grateful to NASA Ames Research Center's Engineering for Complex Systems Program for supporting this work under Grant Number NNA04CK20A, and for providing valuable feedback.

BEHAVIOR METRICS

We determine the value of a project alternative using a mathematical function of product structure (using PRA fault trees), project information processing behavior (using VDT simulation), and domain knowledge (ranging from a discount rate to the sensitivities and specificities of product components' test suites). We present product testing as a detailed module that may be omitted or adapted to represent development, operations, or other phases. This section breaks intermediate calculations into product performance, product cost, integrated testing, element testing, project costs, and organizational behavior. Later sections show how to introduce assessed uncertainties and investment opportunities into this system of equations.

PROJECT PROFITABILITY

Space mission designers might aim to maximize product performance times reliability (minus costs), constrained by a launch window. Other projects may optimize design time and cost tradeoffs by discounting anticipated profits according to the design project's duration. We can support pricing decisions by formulating sales quantity as an explicit function of product performance, reliability, and price over time.

$$(\mathbf{ProfitNPV} \mid \text{volume sales}) = \Sigma_{time} (\mathbf{Sales}_{time} * \mathbf{Margin}_{time}) * (1 - \underline{\text{DiscountRate}})^{time}) - \mathbf{Cost}_{project}$$

$$(\mathbf{ProfitNPV} \mid \text{one product, internal use, risk neutral}) = (\mathbf{Performance}_{product} * \mathbf{Reliability}_{product} - \mathbf{UnitCost}_{product}) * (1 - \underline{\text{DiscountRate}})^{\mathbf{Duration}_{project}} - \mathbf{Cost}_{project}$$

$$(\mathbf{Sales}_{time} \mid time <= \mathbf{Duration}_{project}) = 0$$

$$(\mathbf{Sales}_{time} \mid time > \mathbf{Duration}_{project}) = \underline{\text{Sales}} (\mathbf{Performance}_{product}, \mathbf{Reliability}_{product}, \underline{\text{Price}}_{time}, time)$$

$$\mathbf{Margin}_{time} = \underline{\text{Price}}_{time} - \mathbf{UnitCost}_{product}$$

PRODUCT PERFORMANCE

To estimate the benefits that a functioning product coveys to end users, we mediate a QFD analysis by the performance of individual components, critical interfaces, and interdependent subsystems. VDT's design process analysis can signal the compromise of elements' intended quality.

$$(\mathbf{Performance}_{product} \mid \text{FullyIndependent, SingleString}) = \Pi_{component} \mathbf{Performance}_{component} * \Pi_{interface} \mathbf{Performance}_{interface} * \Pi_{system} \mathbf{Performance}_{system}$$

$$\forall_{element \in \{component\} \cup \{interface\} \cup \{system\}}$$

$\mathbf{Performance}_{product} = \textbf{\textit{QFDPerformanceEstimate}}(\underline{\text{ProductStructure}}, \mathbf{Performance}_{element1}, \mathbf{Performance}_{element2}, \ldots)$

$\forall_{component} \forall_{interface} \forall_{system}$

$\mathbf{Performance}_{component} = \textbf{\textit{VDTCommunicationsRisk}}_{component} * \underline{\text{PlannedPerformance}}_{component}$

$\mathbf{Performance}_{interface} = \textbf{\textit{VDTCommunicationsRisk}}_{interface} * \underline{\text{PlannedPerformance}}_{interface}$

$\mathbf{Performance}_{system} = \textbf{\textit{VDTMeetingRisk}}_{system} * \underline{\text{PlannedPerformance}}_{system}$

PRODUCT COST

Each finished product costs as much as the sum of its elements (components, two-component interfaces, and multi-component systems), inflated by the fraction of products that fail integrated testing. Qualified parts in turn are more costly to produce (though generally more reliable) when element testing rejects a larger fraction. Each element's cost before testing depends upon a target cost and the design process's support of economical design.

$\mathbf{UnitCost}_{product} = \Sigma_{element} \mathbf{UnitCost}_{element} / \mathbf{TestPassRate}_{product}$

$\forall_{element \in \{component\} \cup \{interface\} \cup \{system\}}$

$\mathbf{UnitCost}_{element} = \mathbf{ProductionCost}_{element} / \mathbf{TestPassRate}_{element}$

$\forall_{component} \forall_{interface} \forall_{system}$

$\mathbf{ProductionCost}_{component} = \textbf{\textit{VDTCommunicationsRisk}}_{component} * \underline{\text{PlannedCost}}_{component}$

$\mathbf{ProductionCost}_{interface} = \textbf{\textit{VDTCommunicationsRisk}}_{interface} * \underline{\text{PlannedCost}}_{interface}$

$\mathbf{ProductionCost}_{system} = \textbf{\textit{VDTMeetingRisk}}_{system} * \underline{\text{PlannedCost}}_{system}$

INTEGRATED TESTING

We assess costs for the facilities that test finished product and constituent elements' reliability. We measure the effectiveness of testing using sensitivity (fraction of good products accepted) and specificity (fraction of bad products rejected). The finished products' reliability measure is the fraction of approved products that PRA analysis of approved constituent elements indicates will not fail. The least reliable product configurations will fail whenever any component, interface, or subsystem fails.

$\mathbf{TestCost}_{project} = \mathbf{TestCost}_{product} + \Sigma_{element}\mathbf{TestCost}$

$\mathbf{Reliability}_{product} = \mathbf{PSucceed}_{product} * \text{TestSensitivity}_{product} \mathbf{/ Yield}_{product}$

$\textbf{Yield}_{product} = \textbf{PSucceed}_{product} * \text{TestSensitivity}_{product} + (1 - \textbf{PSucceed}_{product}) * (1 - \text{TestSpecificity}_{product})$

$(\textbf{PSucceed}_{product} \mid \text{FullyIndependent, SingleString}) = \Pi_{element} \textbf{Reliability}_{element}$

$\forall_{elementi \in \{component\} \cup \{interface\} \cup \{system\}}$

$\textbf{PSucceed}_{product} = \textbf{\textit{PRA}}(\underline{\text{ProductStructure}}, \textbf{Reliability}_{element1}, \textbf{Reliability}_{element2}, \ldots)$

Element Testing

As with finished products, design elements' reliability is probability that an approved product won't fail. Failure probability is based on the selected application and on the effectiveness of the design process.

$\forall_{element \in \{component\} \cup \{interface\} \cup \{system\}}$

$\textbf{Reliability}_{element} = \textbf{PSucceed}_{element} * \underline{\text{TestSensitivity}}_{element} / \textbf{Yield}_{element}$

$\textbf{Yield}_{element} = \textbf{PSucceed}_{element} * \underline{\text{TestSensitivity}}_{element} + (1 - \textbf{PSucceed}_{element}) * (1 - \text{TestSpecificity}_{element})$

$\forall_{component} \forall_{interface} \forall_{system}$

$\textbf{PSucceed}_{component} = \textbf{\textit{VDTFunctionalRisk}}_{component} * \underline{\text{PlannedReliability}}_{component}$

$\textbf{PSucceed}_{interface} = \textbf{\textit{VDTProjectRisk}}_{interface} * \underline{\text{PlannedReliability}}_{interface}$

$\textbf{PSucceed}_{system} = \textbf{\textit{VDTMeetingRisk}}_{system} * \underline{\text{PlannedReliability}}_{system}$

Project Costs

VDT calculates the length of a design project, which we add to possible development and testing time. A project's cost is the sum of testing and facility expenses, organizational burden, and a product configuration's elements fixed costs (such as licensing fees).

$\textbf{Duration}_{project} = \textbf{\textit{VDTProjectDuration}} + \underline{\text{Duration}}_{product}$

$\textbf{Cost}_{project} = \textbf{TestCost}_{project} + \textbf{Cost}_{process} + \textbf{Cost}_{organization} + \textbf{DesignCost}_{product}$

$\textbf{DesignCost}_{product} = \Sigma_{element} \underline{\text{DesignCost}}_{element}$

$\textbf{Cost}_{process} = \Sigma_{task} (\textbf{\textit{VDTTaskWorkVolume}}_{task} * \underline{\text{VariableCost}}_{task} + \underline{\text{FixedCost}}_{task})$

Organizational Behavior

Labor costs equal total wages due each team for VDT predicted time spent on the project. In addition, we assign penalties for overloaded workers' burnout because it increases the likelihood of absenteeism and turnover. When VDT finds communications breakdowns between management and design teams we assign a cost to reflect the loss of goal alignment, and when VDT finds that teams communicate poorly we reflect the predicted future loss of future productivity.

$\mathbf{Cost}_{\text{organization}} = \mathbf{LaborCost}_{\text{organization}} + \mathbf{Burnout}_{\text{organization}} - \mathbf{Coherence}_{\text{organization}} - \mathbf{Leadership}_{\text{organization}}$

$\mathbf{LaborCost}_{\text{organization}} = \Sigma_{\text{team}}\ \mathbf{LaborCost}_{\text{team}}$

$\mathbf{Burnout}_{\text{organization}} = \Sigma_{\text{team}}\ \mathbf{Burnout}_{\text{team}} / |\{\text{teams}\}|$

$\mathbf{Leadership}_{\text{organization}} = \Sigma_{\text{team}}\ \mathbf{Leadership}_{\text{team}} / |\{\text{teams}\}|$

$\mathbf{Coherence}_{\text{organization}} = \Sigma_{\text{team}}\ \mathbf{Coherence}_{\text{team}} / |\{\text{teams}\}|$

$\forall_{\text{team}}$

$\mathbf{LaborCost}_{\text{team}} = \boldsymbol{VDTActorWorkVolume}_{\text{team}} * \underline{\text{Wage}}_{\text{team}}$

$\mathbf{Burnout}_{\text{team}} = \int_{\text{time}} \boldsymbol{VDTBacklog}_{\text{team, time}} * \underline{\text{StressResponse}}(\text{team}, \boldsymbol{VDTBacklog}_{\text{team, time}}) * d\text{time}$

$\mathbf{Leadership}_{\text{team}} = \boldsymbol{VDTDecisionLatency}_{\text{team}} * \underline{\text{LatencyLeadership}} + \boldsymbol{VDTDecisionDefaults}_{\text{team}} * \underline{\text{DefaultLeadership}}$

$\mathbf{Coherence}_{\text{team}} = \boldsymbol{VDTInfoExchLatency}_{\text{team}} * \underline{\text{LatencyCoherence}} + \boldsymbol{VDTInfoExchDefaults}_{\text{team}} * \underline{\text{DefaultCoherence}}$

PROJECT UNCERTAINTIES

We model uncertain quantities by introducing discrete or continuous random variables into the behavior formulae. Continuous variables defined by probability density functions offer the greatest precision for most applications. The optimization step integrates over the joint distribution of these variables to determine their range of possible impacts. When analytic complexity exceeds our resources, we sample the joint distribution and approximate in Monte Carlo fashion.

Often the simplest way to model uncertainties is with discrete probability distribution functions. Our formulation calculates the implications of each variable setting and uses a weighted average to assess the objective's outcome.

Using Probabilistic VDT

Because VDT internally models stochastic behavior using a Monte Carlo approach, to assess information processing behavior we synthesize a number of simulation output "trials". As input VDT takes task complexity and other point estimates that can be uncertain, difficult to assess, and influential. To refine the model we can define a distribution on important quantities, calculating the results of each case and then integrating into an outcome distribution.

PROJECT STRUCTURE

Structure investments include those project design parameters that influence project (information processing) behavior. Our analysis begins by enumerating structurally compatible choices of organization hierarchy, product component / subsystem decomposition, and task precedence. A detailed iteration specifies variables including teams' sizes, skills and experience levels, components' / subsystems' nominal failure probabilities, and task complexities and work volumes. VDT probabilistically simulates many of these choices' complex and subtle implications, indicating for example that a larger design team will work faster, but sometimes becomes alienated from managers who are too busy to keep the new pace.

PRODUCT CONFIGURATIONS

D = {product configuration $\mathbf{d_i}$}

Product configuration defines a project design's ***information processing deliverable***. A product configuration includes enough information to identify the set of processes that can meet the project goal. The configuration includes, for example, hardware versus software programming choices, because they determine design work volumes. The product configuration does not specify details that do not impact design behavior, such as the quality of materials or product testing.

Selecting Component Redundancies

Consider the conceptual phase design of a NASA program similar to that described in Dillon, Paté-Cornell and Guikema [2003]. At a high level of abstraction, there are four systems in the product: a launch vehicle, science instrumentation, mechanical structure, and electrical subsystem. There are two alternative product configurations: d1 is a "single-string" design that fails if any of the subsystems fail, while d2 is "redundant" in the instrumentation component, and fails either if both of two instrumentation components fails, or if any of the other subsystems fails.

PROCESS CONFIGURATIONS

$\mathbf{P}_i$ = {process configurations $\mathbf{p}_{ij}$| $\mathbf{p}_{ij}$ designs a product with configuration $\mathbf{d}_i$}

Process configuration defines a project design's ***information processing load***. A process configuration defines each design task's *complexity*, *work volumes*, *precedence relationships*, and *communication* and *rework* dependencies. A process configuration's task *complexities* and *work volumes* must generate all of a compatible product configuration's elements. Similarly, components that induce negatively interacting subgoals require corresponding *information exchange links*, *rework dependencies* connect the designs of interfacing components, and interdependent systems' component designers conduct regular *meetings*.

Assigning Responsibility for Component Interfaces

We define one feasible process $\mathbf{p}_{11}$ for $\mathbf{d}_1$ with one task of 100 full-time equivalent (FTE) –hours' design time for each subsystem. We define two alternative processes for $\mathbf{d}_2$: $\mathbf{p}_{21}$ and $\mathbf{p}_{22}$. $\mathbf{p}_{21}$ designs the interfaces among redundant instruments within the instrumentation team. In this case, the instrumentation process has *work volume* of 150 *FTE* hours, as well as increased *requirement complexity* (this increases the likelihood of exceptions pertaining to instrumentation). The second process, $\mathbf{p}_{22}$, requires interface design to be handled by the dependent mechanical and electrical subsystems. Instrumentation and launch vehicle tasks are unchanged from the $\mathbf{p}_{11}$ baseline, but the mechanical and electrical systems' *work volume* increases to 125 *FTE* hours each, and their *uncertainty* and *solution complexity* are high. This shares the direct work burden more evenly, but also increases the coordination load among subsystem designers.

ORGANIZATIONAL CONFIGURATIONS

$\mathbf{O}_{ij}$ = {organizational configurations $\mathbf{o}_{ijk}$ | $\mathbf{o}_{ijk}$ can execute processes with configuration $\mathbf{p}_{ij}$}

Organizational configuration defines a project design's ***information processing capacity***. It is feasible for a process configuration if each task is assigned one or more qualified design teams. In this step, we identify the organization's design teams, defining their *skill*, *application experience*, and task *assignments*. We also specify an *exception handling hierarchy* (including levels of management), degree of *centralization*, and other VDT *culture* measures.

Sizing Engineering and Management Resources

The example project planners may choose small or large design teams, and they may allocate a small or large amount of management Priority. Each of the four organizational configurations is compatible with all of the process configurations, so as shorthand we define $\mathbf{o}_1 = \mathbf{o}_{111} = \mathbf{o}_{211} = \mathbf{o}_{221}$ (and similarly for $\mathbf{o}_2$, $\mathbf{o}_3$, and $\mathbf{o}_4$). $\mathbf{o}_1$ contains a two-person design team (2 *FTEs*) for each design task, and all teams report to a manager that simultaneously manages many other projects (10% available). $\mathbf{o}_2$ differs from $\mathbf{o}_1$ in that each design team includes four designers (4 *FTE*s) rather than two. $\mathbf{o}_3$ is like $\mathbf{o}_1$ except that the manager is more available to handle exceptions (20% time). $\mathbf{o}_4$ has the more available manager and the larger design teams.

PROJECT BEHAVIOR

$\forall\ \mathbf{o}_{ijk} \in \mathbf{O}_{ij}, \mathbf{R}_{ijk} = \text{VDT}\ (\mathbf{o}_{ijk}, \mathbf{p}_{ij})$

VDT calculates ***information processing behavior*** by matching process and organization configurations (information processing load and capacity). A first look at the diverse VDT product, process, and organization outcome metrics may inspire $\mathbf{D}$, $\mathbf{P}_i$, or $\mathbf{O}_{ij}$ revisions. For example, predicting an intolerably long project might inspire adding a new $\mathbf{o}_{ijk}$ alternative with teams of higher *skill*, *experience* and *wage*. Once we are satisfied the VDT outcome distributions effectively forecast behavior, we proceed with an analysis that treats each simulation trial output as a constant.

Assessing Product Failure Probability

Probabilistic Risk Analysis forecasts complex products' failure probabilities by characterizing each design element and defining their interactions:

PSucceed $_{\text{product}}$ = ***PRA*** (<u>ProductStructure</u>, **Reliability** $_{\text{element1}}$, **Reliability** $_{\text{element2}}$, …)

First we translate the product structure into a functional definition of interdependencies and distill minimal cut sets. Also from the product structure, we seed each element with a failure probability that is intrinsic to its design intent and operational environment.

Using VDT's exception handling quality metrics we also estimate the probabilities of failure resulting from a design flaw. VDT creates exceptions when designers aren't sufficiently skilled or experienced to handle the complexity of their component design work. These exceptions are most likely to lead to errors when decision makers are unavailable:

$\mathbf{PSucceed}_{component} = \boldsymbol{VDTFunctionalRisk}_{component} * \underline{PlannedReliability}_{component}$

When designers are too busy to consider adjusting their designs to changes in dependent components, we assign a probability that the interface between the components will fail:

$\mathbf{PSucceed}_{interface} = \boldsymbol{VDTProjectRisk}_{interface} * \underline{PlannedReliability}_{interface}$

Finally, complex interactions leading to failure are more likely for groups of interdependent components when system integration meetings have low attendance:

$\mathbf{PSucceed}_{system} = \boldsymbol{VDTMeetingRisk}_{system} * \underline{PlannedReliability}_{system}$

Aggregating design failure probabilities for components, interfaces, and systems to a total product failure probability allows us to relate subtle design project behavior to product outcomes.

INVESTMENTS AND PRIORITIES

In this step, we identify investments that benefit project performance in ways that we understand, but that are difficult to weight against corresponding costs. We typically define them as decision variables in the optimization, with corresponding financial cost, non-negativity constraints, and (optionally) a discretionary budget limit:

$\forall_{measure}$

$\mathbf{Cost}_{project} = \mathbf{Cost}_{process} + \mathbf{Cost}_{organization} + \mathbf{DesignCost}_{product}$

$\mathbf{Cost'}_{project} = \mathbf{Cost}_{process} + \mathbf{Cost}_{organization} + \mathbf{DesignCost}_{product} + \Sigma_{measure} \mathbf{Invest}_{measure}$

$\Sigma_{measure} \mathbf{Invest}_{measure} <= \underline{DiscretionaryBudget}_{project}$

$\mathbf{Invest}_{measure} >= 0$

Focusing on Economy, Reliability, or Performance

We can also model conserved, non-monetary resources such as priorities among design economy, reliability, and performance. This may be an uncertain quality of design teams, a management choice, or a linear combination that takes VDT-predicted leadership into account. As a simple management decision, each related formula:

$\forall_{component}$

$\mathbf{PSucceed}_{component} = \boldsymbol{VDTFunctionalRisk}_{component} * \underline{PlannedReliability}_{component}$

$\mathbf{ProductionCost}_{component} = \boldsymbol{VDTCommunicationsRisk}_{component} * \underline{PlannedCost}_{component}$

$\textbf{Performance}_{component} = \textbf{\textit{VDTCommunicationsRisk}}_{component} * \underline{\text{PlannedPerformance}}_{component}$

is mediated with decision variables:

$\forall_{component} \forall_{measure}$

$\textbf{PSucceed}'_{component} = \textbf{Priority}_{Psucceed\ component} * \textbf{\textit{VDTFunctionalRisk}}_{component} * \underline{\text{PlannedReliability}}_{component}$

$\textbf{ProductionCost}'_{component} = \textbf{Priority}_{ProductionCost\ component} * \textbf{\textit{VDTCommunicationsRisk}}_{component} * \underline{\text{PlannedCost}}_{component}$

$\textbf{Performance}'_{component} = \textbf{Priority}_{Performance\ component} * \textbf{\textit{VDTCommunicationsRisk}}_{component} * \underline{\text{PlannedPerformance}}_{component}$

We also define conservation and a feasible range:

$\textbf{Priority}_{Psucceed\ component} + \textbf{Priority}_{ProductionCost\ component} + \textbf{Priority}_{Performance\ component} = 3$

$1.5 > \textbf{Priority}_{measure\ component} > 0.5$

Investments Made in Advance

In our formulation, investments that influence information processing must appear as discrete project structure choices. However, many continuously variable investments are independent of project structure and compensate for information processing weaknesses.

Improving Tests to Catch Design Errors

Although some failures may not be preventable, improving test facilities can reduce the anticipated risks of design flaws or manufacturing defects. For an electrical component, for example, we can identify the testing formulae:

$\textbf{Reliability}_{electrical} = \textbf{PSucceed}_{electrical} * \underline{\text{TestSensitivity}}_{electrical} / \textbf{Yield}_{electrical}$

$\textbf{Yield}_{electrical} = \textbf{PSucceed}_{electrical} * \underline{\text{TestSensitivity}}_{electrical} + (1 - \textbf{PSucceed}_{electrical}) * (1 - \underline{\text{TestSpecificity}}_{electrical})$

and substitute functions of a new decision variable, such as:

$\textbf{TestSensitivity}_{electrical}(x) = 1 - (1 - \underline{\text{TestSensitivity}}_{electrical})^{(1 + x/100)}$

$\textbf{TestSpecificity}_{electrical}(x) = 1 - (1 - \underline{\text{TestSpecificity}}_{electrical})^{(1 + x/200)}$

For $\textbf{Invest}_{test\ electrical}$ this yields:

$\textbf{Reliability}'_{electrical} = \textbf{PSucceed}_{electrical} * \textbf{TestSensitivity}_{electrical}(\textbf{Invest}_{test\ electrical}) / \textbf{Yield}'_{element}$

$\textbf{Yield}'_{electrical} = \textbf{PSucceed}_{electrical} * \textbf{TestSensitivity}_{electrical}(\textbf{Invest}_{test\ electrical}) + (1 - \textbf{PSucceed}_{electrical}) * (1 - \textbf{TestSpecificity}_{electrical}(\textbf{Invest}_{test\ electrical}))$

INVESTMENTS MADE WITH HINDSIGHT

Advance investments' effectiveness is limited because (as VDT predicts) each project configuration's performance is uncertain. In contrast, some decisions occur after the design project is complete and information processing results are known. We optimize investments made with perfect hindsight by linking decision variables to individual VDT trials' output. The investment's costs and benefits are weighted accordingly in the optimization. Computational complexity may limit the simulation trials of applications using this feature.

Bonding, Inspiring, and Rejuvenating a Worn-Out Team

Organizational sustainability can require investments that balance any VDT-predicted loss of coherence, leadership, and burnout. Long term investments include facility comforts, incentive compensation, and vacation, while less economical, reactive solutions include team building, topical training, and comp time (off-record vacation). We link global and (scaled) simulation - specific investments to each team's performance using (3 * |{project configurations}| * |{trials}| * |{teams}|) decision variables and constraints:

$\forall_{team} \forall_{trial}$

$\mathbf{Invest}_{\text{incentives team}} + (|\{\text{trials}\}|/2) * \mathbf{Invest}_{\text{training team trial}} \geq -100 * \mathbf{Leadership}_{\text{team trial}}$

$\mathbf{Invest}_{\text{vacation team}} + (|\{\text{trials}\}|/2) * \mathbf{Invest}_{\text{comp\_time team trial}} \geq 100 * \mathbf{Burnout}_{\text{team trial}}$

$\mathbf{Invest}_{\text{facilities team}} + (|\{\text{trials}\}|/2) * \mathbf{Invest}_{\text{team\_building team trial}} \geq -100 * \mathbf{Coherence}_{\text{team trial}}$

PROJECT DESIGN OPTIMIZATION

Finally, we combine the system of equations describing project objective, uncertainties, information processing behavior, and investments into a nonlinear optimization. We first take the Cartesian product of all sets of discrete alternatives to generate an exhaustive list of discrete strategies (simplifying with strategy tables per available computing power). For each discrete strategy, we generate the system of equations that reflects project performance and simplify wherever possible (through substitution using Mathematica™, for example). We multifurcate again by collecting the Cartesian product of discrete uncertainties (including VDT trials), then substitute each joint sample's results into a copy of the equations and simplify. Next, we integrate over the joint distribution on all continuous variables, presumably simplifying to account for widespread independence.

At this point, each strategy and unique set of discrete uncertainty values has an associated case whose objective function and constraints include **Invest** and **Priority** variables. For each strategy we average the objective function over the discrete uncertainties, weighted by joint probability, and we take the union of the associated constraints (and simplify). Solving this problem with nonlinear optimization algorithm will yield optimal investment choices for the **Invest** and **Priority** variables, and expected values on objective function outcomes, for each discrete strategy. Our algorithm recommends the discrete strategy with continuous investments having the greatest expected objective function.

Mathematically, we maximize:

$\forall_{strategy} \in$ {strategy} = {decision1_alternatives} × {decision2_alternatives} …

$\forall_{c\_uncertaintyi,\ c\_uncertaintyj\ \ldots} \in$ {continuous uncertainty}

Maximize

$\textbf{ProfitNPV}_{strategy} = \int_{c\_uncertaintyi} \int_{c\_uncertaintyj} \ldots \left[\Sigma_{trial} \left(\textbf{ProfitNPV}_{strategy\ trial} * \textbf{Probability}_{strategy\ trial}\right) * \Pi_{c\_uncertainty} \left(\textbf{Probability}_{c\_uncertainty}\ d\text{c\_uncertainty}\right)\right]$

Subject to the union of all equations developed above under the substitution:

$\forall$ **Variable**

$\textbf{Variable}_{strategy\ trial} = (\textbf{Variable}$ | evaluated at strategy and trial)

In many cases, $\textbf{Probability}_{strategy\ trial} = \textbf{Probability}_{trial}$. We recommend the strategy with highest $\textbf{ProfitNPV}_{alternative}$ and its corresponding {**Invest**} and **{Priority}** variables

VDT calculates many variables of importance, but does not define their impact on project performance, and therefore cannot recommend one project configuration over another. To determine the better of two organizational configurations after 100 VDT simulation trials, we select the highest value of:

$\textbf{ProfitNPV}_{org\_config} = \Sigma_{trial} \left(\textbf{ProfitNPV}_{org\_config\ trial} / 100\right)$

We calculate this by taking substitutions such as

$\textbf{ProductionCost}_{component} = \boldsymbol{VDTCommunicationsRisk}_{component} * \underline{\text{PlannedCost}}_{component}$

and instantiating them with individual simulation outcomes:

$\textbf{ProductionCost}_{component\ org\_config\ trial} = \boldsymbol{VDTCommunicationsRisk}_{component\ org\_config\ trial} * \underline{\text{PlannedCost}}_{component}$

If the decision-maker's preferences and constraint structure show linear independence in **ProductionCost** and its probabilistically dependent variables, we consolidate:

$$(\mathbf{ProductionCost}_{\text{component org\_config}} \mid \text{linear independence}) = \Sigma_{\text{trial}} (\boldsymbol{VDTCommunicationsRisk}_{\text{component org\_config trial}} / 100) * \underline{\text{PlannedCost}}_{\text{component}}$$

This problem does not require a nonlinear optimization solver because we exhaustively analyze all discrete cases. To instead optimize the nonlinear **Priority** allocation decision above, we would use the formulation:

$$\mathbf{ProductionCost'}_{\text{component org\_config trial}} = \mathbf{Priority}_{\text{ProductionCost component org\_config}} * \boldsymbol{VDTCommunicationsRisk}_{\text{component org\_config trial}} * \underline{\text{PlannedCost}}_{\text{component}}$$

If $\mathbf{Priority}_{\text{ProductionCost}}$ were a "hindsight" variable (subscripted by trial), its nonlinear interaction with ***VDTCommunicationsRisk*** would prohibit simplification.

REMARKS

The algorithm includes several contributions, such as focusing the interpretation of VDT predictions and quantitatively linking design collaboration effectiveness to product risk. Some important questions are addressed in preceding work for this tutorial, but others remain:

- We do not define formulations that capture certain relationships between uncertainties and decision variables. Ideally, influence diagrams should be used to clarify the general case of relationships among more complex projects' uncertainties, simulated outcomes, and decision variables.
- Analytic and computational complexity will constrain the application to an unknown degree. For example, the nonlinear optimization may require discretization (or numerical solution) when a continuous uncertainty and investment with "hindsight" share constraints or are linearly dependent in the objective function.
- Further developing and solving a consistent example would clarify the algorithm.