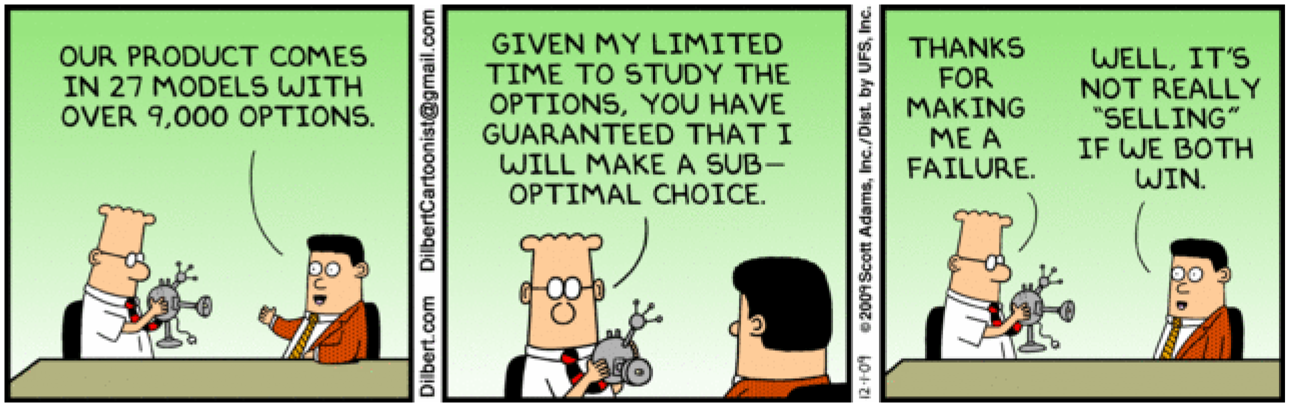
Utilizing Optimization Techniques to Enhance Cost and Schedule Risk Analysis

Colin Smith, Brandon Herzog
SCEA 2012

Booz | Allen | Hamilton

Table of Contents


OUR PRODUCT COMES IN 27 MODELS WITH OVER 9,000 OPTIONS.
GIVEN MY LIMITED TIME TO STUDY THE OPTIONS, YOU HAVE GUARANTEED THAT I WILL MAKE A SUB-OPTIMAL CHOICE.
THANKS FOR MAKING ME A FAILURE.
WELL, IT'S NOT REALLY "SELLING" IF WE BOTH WIN.
Dilbert.com DilbertCartoonist@gmail.com
12-1-09 ©2009 Scott Adams, Inc./Dist. by UFS, Inc.

What is Optimization?

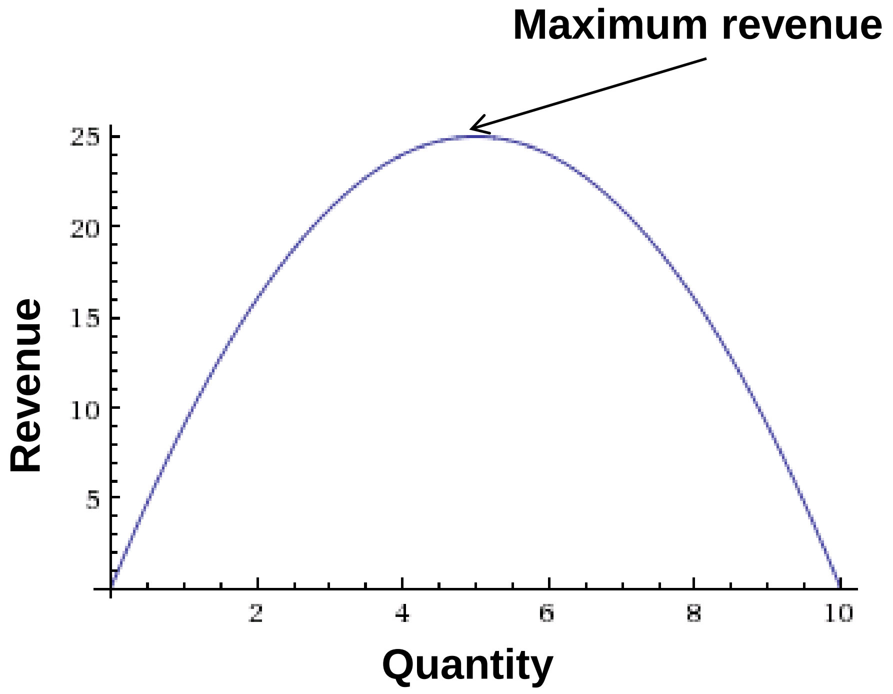
Optimization is the process of selecting the correct input values to the model to get the best output

- An optimization routine changes **variables** according to **constraints** in order to maximize an **objective function**

- **Variables:** What you have power to change

- **Constraints:** Rules on how you are able to change the variables

- **Objective Function:** How 'good' is each solution the optimization routine processes

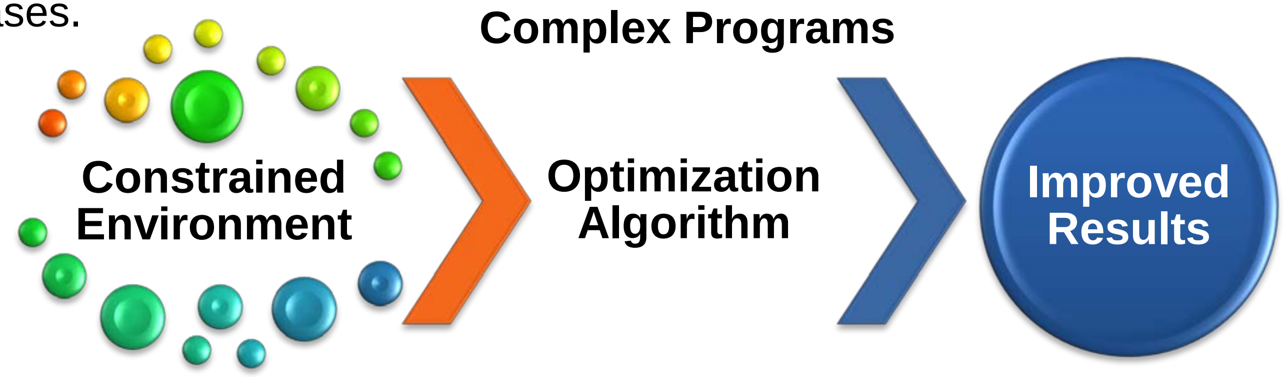


Why apply optimization to an Uncertainty Analysis?

- Program managers are already making decisions within constrained environments
  - Adjusting the schedule given an ever-changing budget profile
  - Accelerating different tasks in response to deadline changes
  - Determining which risk mitigation strategies to implement

- Manual optimization and expert judgment are exceedingly inefficient as programs grow in complexity and more accurate results are required

- **Example**: Program managers choose which task should receive additional resources in order to meet project deadlines. However, this is unsustainable as complexity increases.



Risk register example: Showcasing a need for Optimization

- Lets imagine for a moment we are the risk management team for an acquisition program required to plan against the 70th percentile confidence level schedule
  - We have constructed a Monte Carlo simulation to measure our confidence levels
- Consider a traditional qualitative risk register for this program:

Qualitative Risk Register			
Risk Name	**Probability (1-5)**	**Consequence (1-5)**	**Risk Score**
Risk A	5	5	25
Risk B	3	4	12
Risk C	4	4	16
Risk D	3	3	9

- Now lets assume we have enough money to mitigate any two risks. Which two do we pick?
  - At face value we will pick the two risks with the highest risk scores, Risk A + B

Risk register example: Showcasing a need for Optimization

- Based on information about the risks, lets transform this to a quantitative risk register we can use in our Monte Carlo simulation to measure confidence levels:

Quantitative Risk Register			
Risk Name	**Task Affected**	**Probability**	**Consequence**
Risk A	Task A	0.95	10 days
Risk B	Task B	0.6	8 days
Risk C	Task C	0.8	10 days
Risk D	Task D	0.6	6 days

- Now lets still assume we have enough money to mitigate any two risks. Which two do we pick?
  - At face value we will pick Risk A + B; these two risks have both higher probability of occurrence and higher consequence than the other two

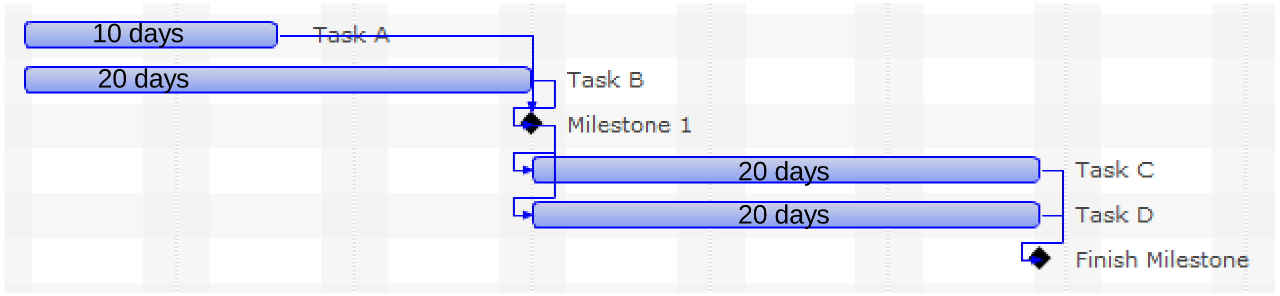
- We can mitigate these two risks in our simulation model and see the impact:

Strategy	**70th Percentile Duration**
no mitigation	58 days
mitigate A + C	54 days

- We improved our schedule by 4 days. However, this is not the optimal solution!

Risk register example: Showcasing a need for Optimization

- Lets now take a look at our schedule for a hint as to why this might not be the optimal solution:



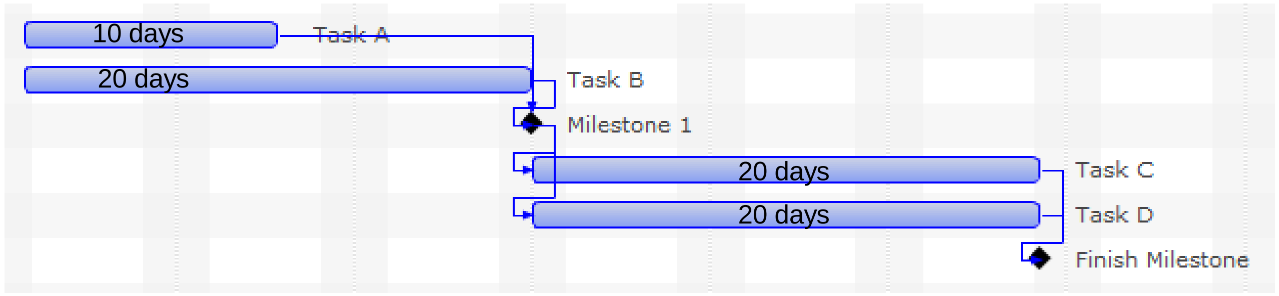
- After examining our schedule more carefully, it becomes obvious that, because of the parallel nature of tasks, mitigating Risk A + C was a poor choice

- Instead lets try mitigating Risk A + B and Risk C + D; one of these must be the optimal strategy because they affect tasks in parallel

Strategy	70th Percentile Duration
no mitigation	58 days
mitigate A + C	54 days
mitigate A + B	50 days
mitigate C + D	48 days

- We did better, saving 10 days this time, but this is still not as good as we can do!

Risk register example: Showcasing a need for Optimization



- Out of frustration, we decide to try every possible combination of mitigations and run our simulation:

Strategy	70th Percentile Duration
no mitigation	58 days
mitigate A + C	54 days
mitigate A + B	50 days
mitigate C + D	48 days
mitigate A + D	58 days
mitigate B + C	46 days
mitigate B + D	50 days

- Unintuitively, mitigating Risks B + C was actually the optimal choice, saving 12 days

- If our “expert opinion” and intuition on this simple example was so wrong, imagine trying to make sense of a large, extremely complex schedule

Booz | Allen | Hamilton

What should be the goal of the optimizer?

- Unlike traditional optimization problems there is no easy way to calculate the correctness of a solution given the stochastic nature of the model
  - Checking the correctness of a traditional problem is easily measured – comparing deterministic profit values or deterministic performance
  - Checking the correctness within an **Uncertainty Analysis** is difficult – how to measure two different risk mitigation strategies when risks occur in parallel, occur rarely, or when their probability of occurrences are correlated with each other
- We propose two different measurement strategies for the community to consider

Percentile

- Measure the fiscal year budget and schedule based on a specified percentile in order to determine the effectiveness of a solution.

Variance

- Removing variance from the project should be a goal of the optimizer so that more precise results can be found. Solutions will be sought which reduce the amount of variance.

Sample applications of Optimization for Cost and Schedule Risk Analysis

Risk Mitigation

- Given a limited amount of risk mitigation funding, which risk mitigation strategies should be implemented to either lower expected cost or shorten the expected schedule

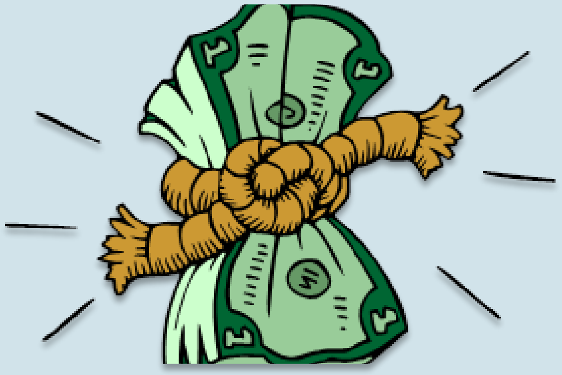
Scheduling

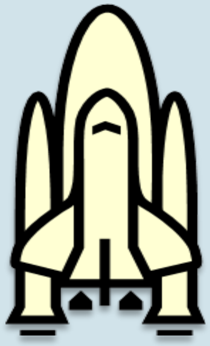
- Given a schedule and limited fiscal year budget, which tasks should be delayed, shortened, or extended in order to meet budgetary and schedule requirements

Performance

- Given capability performance requirements and fiscal year budgets, which tasks or scope should be cut in order to deliver the best expected performance while remaining under budget

The constraints for the optimization problem include fiscal, schedule and performance requirements





Fiscal Year Budget

- The optimization routine must abide by the given fiscal year budgets and will not shift the schedule, implement risk mitigation strategies or other changes that would cause the program to violate the given budget.

Project Milestone Goals

- The optimization routine cannot adjust the schedule if the shift results in one of the specified milestones being delayed

Performance Levels

- The optimization routine will not remove scope or delay tasks if it causes the performance levels to drop to unacceptable levels or if the program reaches those capability levels after the given date

The variables the optimization algorithm can manipulate include risk mitigation strategies, task scheduling, and scope

- **Risk Mitigation Strategies**
  - Which subset of risk mitigation strategies to implement in order to achieve the desired budgetary goals and deadline
- **Task dates and duration**
  - The optimization routine will shift the start dates of different tasks in order to better balance fiscal year budgets or to meet project deadlines
  - If the budget allows, the optimizer will apply additional resources to task in order to shorten its duration or extend the duration of a task to lessen the per fiscal year cost of the task
- **Changes in scope and requirements**
  - The optimizer will determine if certain scope or requirements are feasible given the budgetary requirements and schedule constraints, and if necessary it will remove that scope

By changing the above variables according to the given constraints, the optimization algorithm will determine the optimal strategy in order to achieve budget, schedule and performance goals

Cost and schedule optimization has parallels to existing problems

Problem	Solution	Complexity
Risk Mitigation	Similar to Knapsack Problem	NP-Complete
Task Dates	Stochastic Scheduling	NP-Hard
Accelerating Tasks	Min Cost/Max Profit Scheduling	NP-Hard
Performance and Scope	Similar to Knapsack Problem	NP-Complete

Computational Complexity Classes

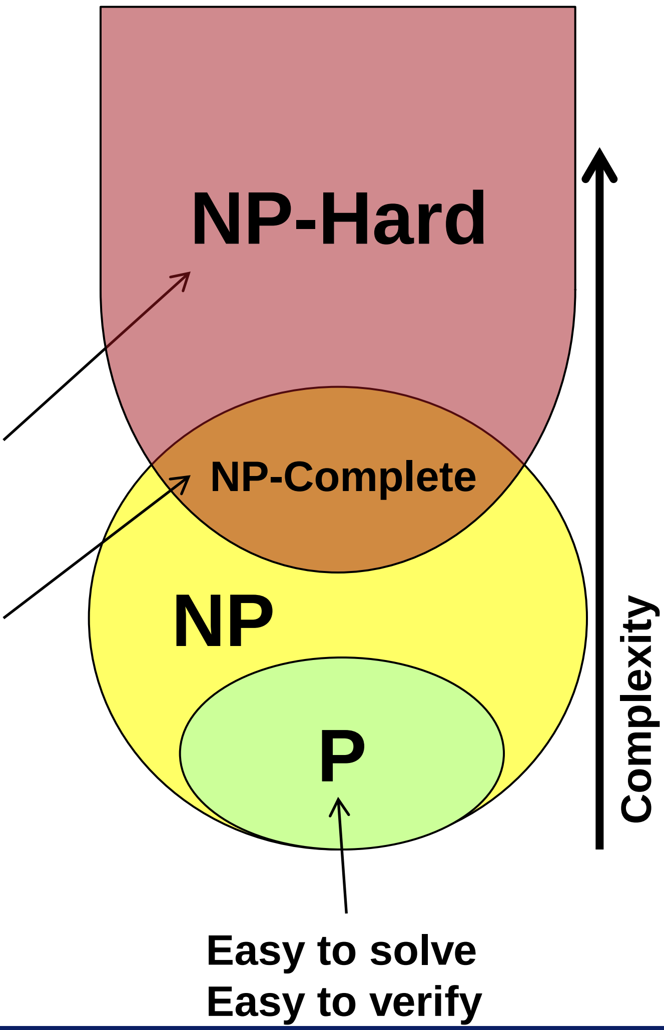
- The mathematical problem of cost and schedule risk analysis optimization boils down to some of the most complex problems known

- NP-Complete and NP-Hard problems **cannot** be practically solved

- NP-Complete and NP-Hard problems **may** be able to be approximated

Intractable to solve
Intractable to verify

Intractable to solve
Easy to verify

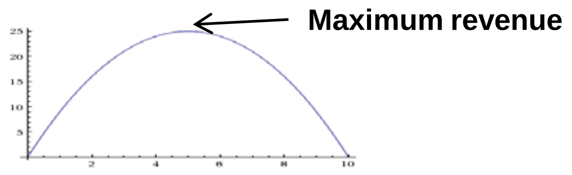


Summary

It is intractable to solve these problems exactly; however, we can create optimization routines which approximate the solution

Review of Optimization

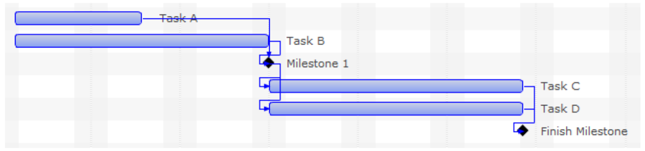
- Optimization is the process of selecting the correct input values to the model to get the best output



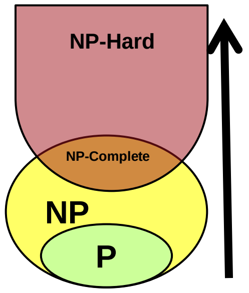
- Programs are ever growing in their complexity and the consequences for going over budget or overrunning the schedules are dire

Constrained Environment
Optimization Algorithm
Improved Results

- As we saw in Risk Mitigation example, “expert judgment” does not always result in optimal results

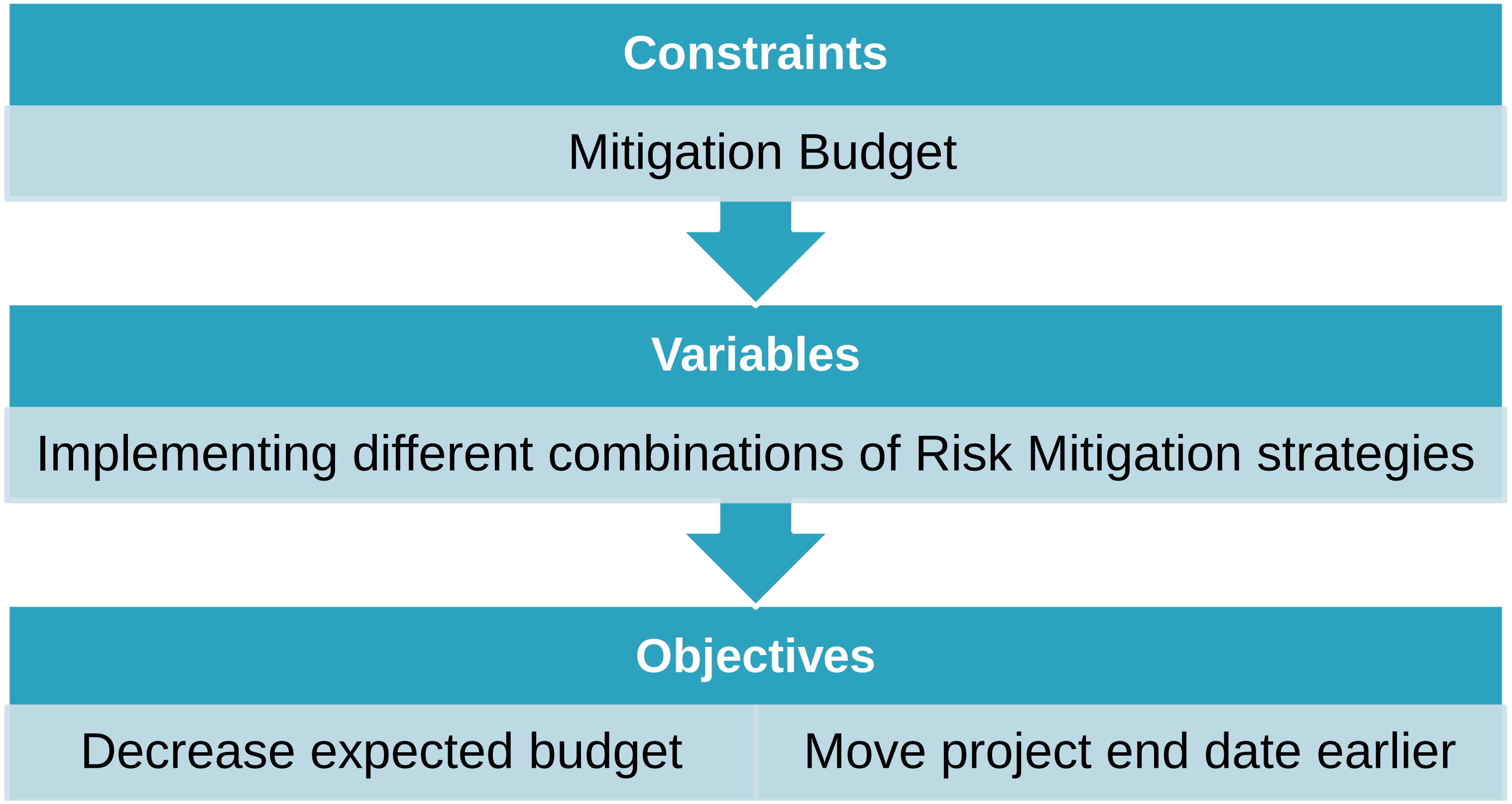


- For this reason optimization algorithms must be employed to assist the program manager in making the bests choices at any given time



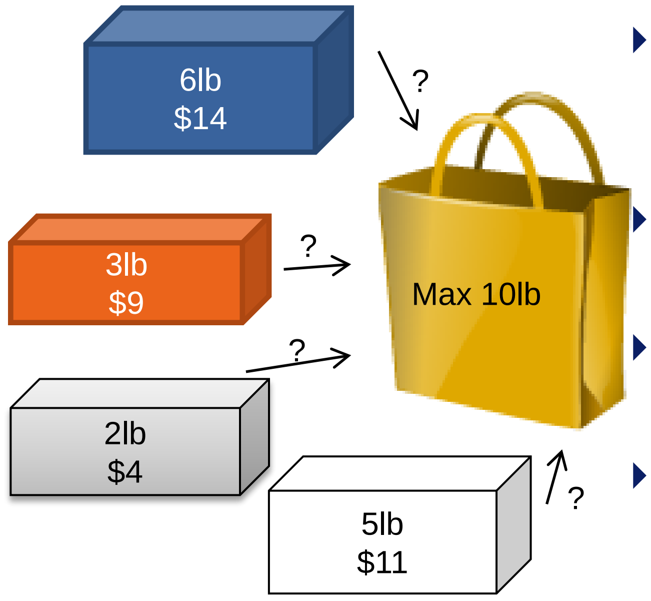
As a proof of concept we propose a solution to the Risk Mitigation Problem

- **Problem:** How can a project manager select the optimal combination of *Risk Mitigation* strategies to implement for a project while constrained by a given budget?
- We are measuring our success by gauging the program at a **specific percentile level**


Constraints
Mitigation Budget
Variables
Implementing different combinations of Risk Mitigation strategies
Objectives
Decrease expected budget
Move project end date earlier

We can draw inspiration from the Knapsack Problem

- The risk mitigation problem is similar to the classical combinatorial optimization problem known as the **Knapsack Problem**

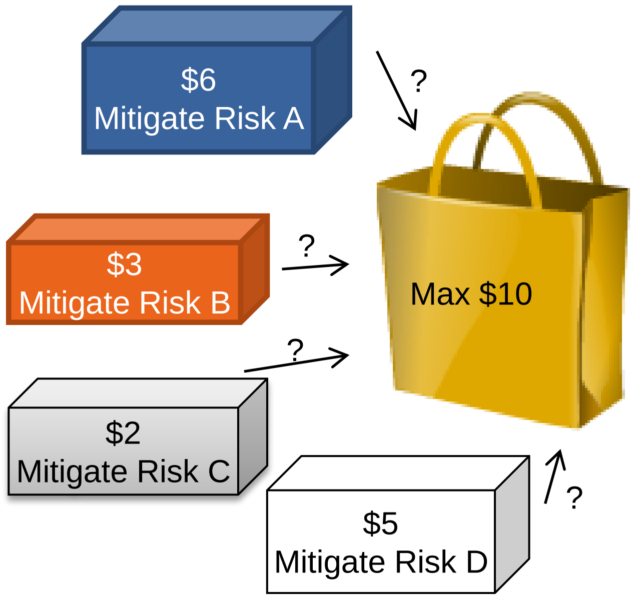


- Imagine that you have multiple packages with different weights and dollar values and a paper bag that will break if you place more than 10lb inside
- How do you maximize the value of the packages placed in the bag without breaking the bag?
- Despite the seemingly simple problem, no known method can quickly calculate the optimal solution
- For example, simply picking the packages in order of highest value per pound results in the wrong answer (Red + Blue = $23 is less than Red + Grey + White = $24)

- While certain heuristics can limit the search space, the **Knapsack Problem** can essentially only be solved by trying all 2^n possible combinations
- This sort of brute force calculation can take years for large problems
  - For 20 items there are over 1 million combinations, for 40 over 1 trillion

Risk Mitigation strategy as a Knapsack Problem

- Finding an optimal Risk Mitigation strategy given budgetary constraints can be thought of as a **Multi-Objective Knapsack Problem**



- You can mitigate different risks for different costs; however, you have at most $10 to spend.
- How do you maximize the value of the risk mitigation actions while staying under your budget?
- Value is measured in terms of multiple objectives—schedule and cost reduction
- Monte Carlo simulation can be used to test the impact of different risk mitigation strategies on confidence levels in cost and schedule
  – This is slow for each test

- Using brute force to solve this problem is impossible because use of Monte Carlo simulation for each value test is extremely slow

Our solution to the Risk Mitigation Knapsack Problem is approximation through the use of Evolutionary Algorithms

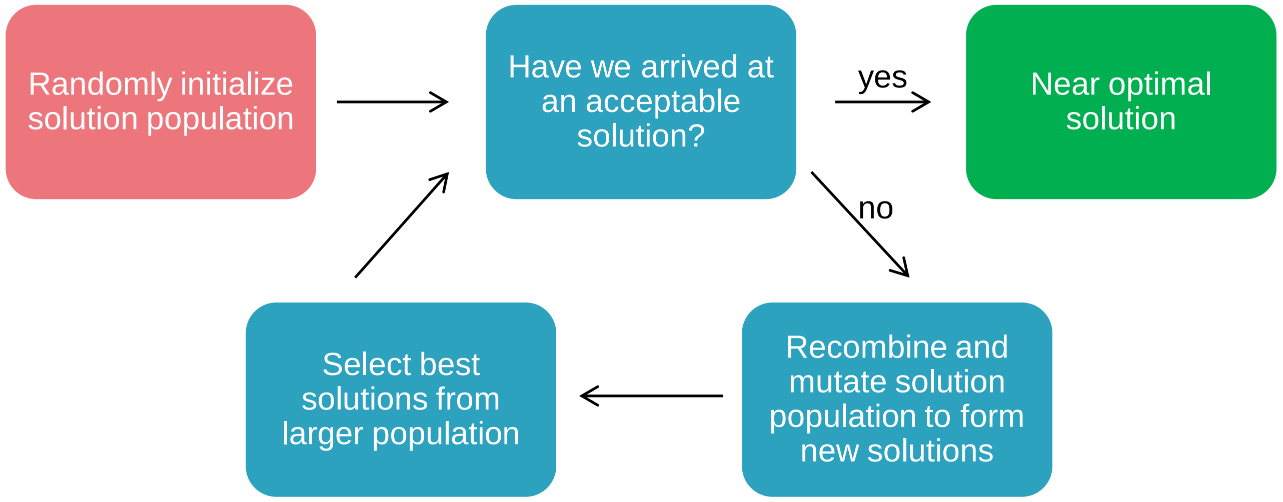
- Fast solutions or approximations to various versions of the Knapsack Problem have been found for specific domains; however, they generally make use of several assumptions that do not apply to the cost-schedule Risk Mitigation problem

Traditional Knapsack Problem approximation algorithms assume:	Risk Mitigation problem violates these assumptions:
Positive values for items	Mitigation strategies may have negative value (cost more than they save)
Linear relationships	Risk mitigation value is highly interrelated with other mitigations selected
Continuous or smooth functions	Schedule logic and risk occurrence results in discontinuous, non-differentiable functions

- Instead of these inappropriate mathematical optimization techniques, we propose the use of Evolutionary Algorithms for optimization of the Risk Mitigation Problem

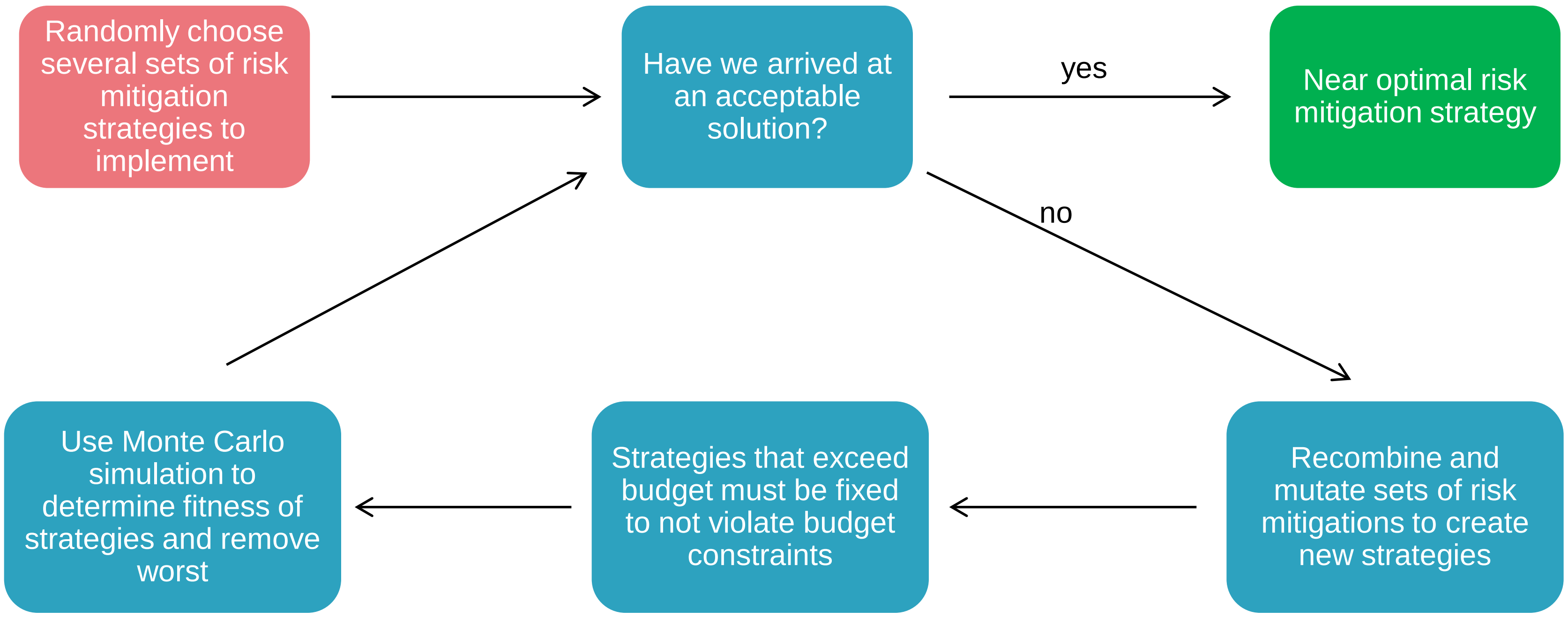
Quick overview of Evolutionary Algorithms

- Evolutionary Algorithms (EAs) are a robust type of metaheuristic that make use of stochastic optimization methods to find globally optimal solutions
  - Metaheuristics generally refer to any combinatorial optimization method that iteratively improves a solution until an approximately optimal solution is found rather than relying on domain specific heuristics
  - Inspired by natural evolution, EAs maintain a fixed population of individual possible solutions which are randomly recombined and mutated (while the worst solutions are continually removed from the population pool) until a nearly optimal solution is "evolved"


Randomly initialize solution population
Have we arrived at an acceptable solution?
yes
Near optimal solution
no
Recombine and mutate solution population to form new solutions
Select best solutions from larger population

Risk Mitigation Problem as an Evolutionary Algorithm

- The Risk Mitigation Problem can be implemented as an Evolutionary Algorithm by considering a set of selected risk mitigations ("strategies") as one individual solution in a population of such strategies


Randomly choose several sets of risk mitigation strategies to implement
Have we arrived at an acceptable solution?
yes
Near optimal risk mitigation strategy
no
Recombine and mutate sets of risk mitigations to create new strategies
Strategies that exceed budget must be fixed to not violate budget constraints
Use Monte Carlo simulation to determine fitness of strategies and remove worst

Why apply Evolutionary Algorithms to Risk Mitigation when you could just find the optimal solution manually?

- Attempting to view all the different Risk Mitigation combinations can quickly get out of hand

Run Times

Evolutionary Algorithms

- Each Simulation: 0.2 seconds
- Population: 300
- Generations: 500
- Total Run Time:
  - 0.2s * 300 * 500
  - 8 hours, 20 minutes

Brute Force

- Each Simulation: 0.2 seconds
- Risk Mitigations: 30
- Permutations: 2^30= 1,073,741,824
- Total Run Time:
  - 0.2s * 1,073,741,824
  - 6 Years, 292 Days

Quantum-Inspired Evolutionary Algorithms (QEAs)

- QEAs are inspired by the concepts of quantum computing and quantum mechanics where data is stored in an uncertain fashion and only has a state when it is ‘observed’
- We are evolving the different probabilities that a certain risk mitigation will be applied

Generation ‘t’ of our evolution

A solution

The jth solution we are evolving

$$\mathbf{q}_j^t = \left[\begin{array}{c|c|c|c} \alpha_{j1}^t & \alpha_{j2}^t & \cdots & \alpha_{jm}^t \\ \beta_{j1}^t & \beta_{j2}^t & \cdots & \beta_{jm}^t \end{array}\right]$$

There is a Q-bit for each Risk Mitigation

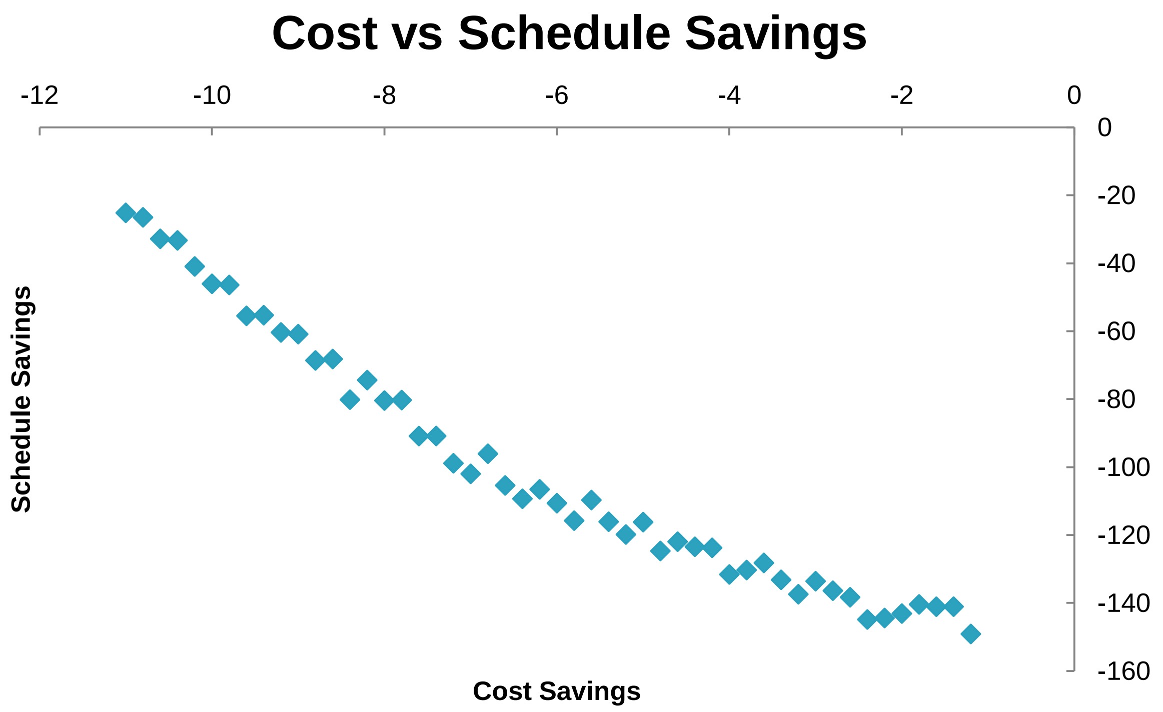
α^2 is the probability of a Risk Mitigation being applied
β^2 is the probability of a Risk Mitigation not being applied

- By evolving hundreds of different solutions (Q-bit individuals) for hundreds of generations we will be able to find the near-optimal solution
- Through migration the best decisions are slowly conveyed to the rest of the population
- Each generation every solution is compared to the best solution found so far and adjusted to be more likely to generate similar solutions in the future
- Due to the random quantum inspired nature of the algorithm local minima are avoided

Booz | Allen | Hamilton

Notional reports for the optimization routine

- Due to the multiple objective functions (lowering cost and schedule) there is not one correct strategy
- A curve of all ‘dominant’ solutions is displayed where each point on the curve represents a single solution
- The analyst will be able to select the solution which best represents the needs of the project being analyzed
- Once a solution is selected all the different Risk Mitigation strategies which were implemented are reported



Summary

- Optimization is needed to ensure that program managers are making the best decisions on increasingly complex programs
  - Remember how hard it was to reason about risk mitigations for even a small schedule!
- Monte Carlo-based Optimization can be reduced to known problems within the scientific community
- The underlying algorithms for conducting the type of optimization needed are extremely complex and is not possible to ‘solve’ them :: NP-Hard, NP-Complete
- We propose using Evolutionary Algorithms to approximate the optimal solution
- The Risk Mitigation problem provided a perfect test bed for testing our approach
- Quantum-inspired Evolutionary Algorithms (QEA) were selected to solve the Risk Mitigation problem
- QEAs provide an array of possible solutions along the cost and schedule tradeoffs, allowing program managers to select the solution best suited for their programs

Points of Contact

Colin Smith
Associate

Booz | Allen | Hamilton

Booz Allen Hamilton Inc.
Atlanta, GA
Tel (404) 658-8011
Smith_Colin@bah.com

Brandon Herzog
Senior Consultant

Booz | Allen | Hamilton

Booz Allen Hamilton Inc.
Arlington, VA
Tel (703) 526-6040
Herzog_Brandon@bah.com