RT-134: Analytic Workbench for System of Systems

Robust Portfolio Optimization (RPO) Basic Overview

Center for Integrated Systems in Aerospace
http://www.purdue.edu/research/vpr idi/cisa/

This material is based upon work supported, in whole or in part, by the U.S. Department of Defense through the Systems Engineering Research Center (SERC) under Contract H98230-08-D-0171. SERC is a federally funded University Affiliated Research Center managed by Stevens Institute of Technology.
Concept of Use: SoS Analytic Workbench

Methods in Toolset:
- Robust Portfolio Optimization
- Multi-Stakeholder (based on ADP)
- System Importance Measures (SIMs)
- SODA/SDDA

Input Data (e.g. DoDAF OV, SV, SysML, PV declarations)
Graph-basis Data Model / Representation

- Translate SoS problem into network topology with hierarchy (nodes, links, inputs, outputs)
- Map data and description to equivalent network representation

Inputs (e.g. req.)

Outputs e.g. capab.

OV – Operation Flow
SV – Service Flow
PV – Project Flow
…
Simulation/Actual data

Mapping

Physical System/Functional Node
Addressing the Archetypal Questions

- **Design**
  1. What combination of systems gives the desired aggregate SoS capabilities?
  2. What changes to which systems offer the most (performance, resilience, etc.) leverage?
  3. Which systems are critical to SoS performance? SoS risks?
  4. Which parts of the SoS have excess or inadequate resilience?
  5. Which design principles can improve SoS robustness and resilience?

- **Development**
  6. How do/should partial capabilities evolve over time?
  7. How do we optimize multi-stage acquisitions in SoS development?
  8. How do we coordinate planning between local and SoS-level stakeholders?
  9. How do changes in system properties affect SoS development?

- **Failures and Delays**
  10. What is the impact of partial/total system failures during operations?
  11. What is the impact of partial/total failure of a system during development?
  12. What are the most critical systems in a given operational (or developmental) network?
  13. What is the impact of development delays in an interdependent network?
Robust Portfolio Optimization

- Treat SoS as ‘portfolio’ of systems
- Model individual systems as ‘nodes’
  - Functional & Physical representation
- Rules for node connectivity
  - Compatibility between nodes
  - Bandwidth of linkages
  - Supply (Capability)
  - Demand (Requirements)
  - Relay capability
- Represent as mathematical programming problem
Robust Portfolio Optimization

Decision support approach from financial engineering/operations research to identify ‘portfolios’ of systems by leveraging performance against risk under uncertainties

- Represent behaviors as connectivity constraints
- Employ robust optimization techniques to deal with data uncertainty
- Computationally efficient tools to solve even for very large number of nodes
Dealing with Uncertainty

• Entities
  – **System Capability**: Actual performance of system individually and as a whole SoS entity
  – **System Interdependence**: Interdependencies between systems and effects on translation of capability uncertainties

• Addressing data uncertainty in portfolio selection
  • Uncertainties in node (system) performance and connections (links) for constraints (Bertsimas-Sim): *CSER 2013 paper*
  • Uncertainties in development times (*NPS Approach – MISDP*)
  • Capture variation in performance at each node from data/simulation (e.g. ABM) – CVaR Approach (*CSER 2014 paper*)
Robust Portfolio Optimization

Decision support approach from financial engineering/operations research to identify ‘portfolios’ of systems by leveraging performance against risk under uncertainties

Performance Efficiency Frontiers
Robust Portfolio Optimization (RPO)
Robust Portfolio Optimization

- Treat SoS as ‘portfolio’ of systems
- Model individual systems as ‘nodes’
  - Functional & Physical representation
- Rules for node connectivity
  - Compatibility between nodes
  - Bandwidth of linkages
  - Supply (Capability)
  - Demand (Requirements)
  - Relay capability
- Represent as mathematical programming problem
Robust Portfolio Optimization

Decision support approach from financial engineering/operations research to identify ‘portfolios’ of systems by leveraging performance against risk under uncertainties

- Represent behaviors as connectivity constraints
- Employ robust optimization techniques to deal with data uncertainty
- Computationally efficient tools to solve even for very large number of nodes

Robustification to include data uncertainties
Dealing with Uncertainties

• Entities
  – **System Capability**: Actual performance of system individually and as a whole SoS entity
  – **System Interdependence**: Interdependencies between systems and effects on translation of capability uncertainties

• Addressing data uncertainty in portfolio selection
  • Uncertainties in node (system) performance and connections (links)
  • Capture variation in performance at each node as **uncertainty sets**.
  • Variations/uncertainty bounds from ABM simulation or design choice.
Baseline: Portfolio Optimization

Objective
Maximize Performance Index

\[
\text{Maximize } \frac{\sum q \left( \frac{S_{qc} - R_c}{R_c} \cdot w \cdot X_q^B \right)}{\sum q X_q^B} \quad \text{s.t.}
\]

- **Requirement Satisfaction**
  \[
  \sum q S_{qc} X_q^B \geq \sum q S_{qR} X_q^B \quad \text{(Satisfy Requirements)}
  \]

- **Big-M Formulation**
  (number of connections)
  \[
  \sum_c X_{cij} - X_{ij} M \leq 0
  \]
  \[
  M \sum_c X_{cij} - X_{ij} \geq 0
  \]

- **Flow Balance Constraint**
  \[
  \sum_i X_{cij} - \sum_j X_{cij} - X_j^B S_{rj} = 0
  \]

- **Bandwidth Limit**
  \[
  X_{cij} \leq \text{Limit}_{cij}
  \]

- **Node Connection Compatibility**
  \[
  X_1^B + \ldots + X_n^B = D \quad \text{(System Compatibility)}
  \]

  \(\text{(e.g. } X_1^B + X_1^B + X_1^B = 1)\)

  \(X_{cij} = 0 \quad \text{if } i,j \in \{\text{incompatible set}\} \in \{0,1\} \text{(binary)}\)
Method 1: Mean-Variance Robust Portfolio

**Objective**
Maximize Performance Index

**Constraints**
- Portfolio Fraction
- Portfolio Total Budget
- Requirements Satisfaction

**Selection Rules (Compatibility)**

**Robust Formulation**
(Tutuncu & Koenig 2004)

\[
X^F_q = \frac{X^S_q C_q}{\text{Budget}} \quad \text{(Portfolio Fractions)}
\]

\[
\sum_q C_q X^S_q + \varepsilon = \text{Budget} \quad \text{(Budget Constraint)}
\]

\[
\sum_q S_{eq} X^S_q \geq \sum_q S_{eq} X^S_q \quad \text{(Satisfy All System Requirements)}
\]

\[
X_1^S + X_1^S + X_1^S = 1 \quad \text{(ASW System Compatibility)}
\]

\[
X_4^S + X_3^S = 1 \quad \text{(MCM System Compatibility)}
\]

\[
X_4^S + X_3^S = 1 \quad \text{(SUW System Compatibility)}
\]

\[
X_8^S + X_8^S + X_10^S = 1 \quad \text{(Package System Compatibility)}
\]

\[
\begin{bmatrix}
\bar{\Lambda} - \Delta & X^F_q \\
X^F_q & 1
\end{bmatrix} \succeq 0 \quad \text{(Linear Matrix Inequality)}
\]

\[
X_q^S \in \{0, 1\} \quad \text{(binary)}
\]
Method 1: Naval Warfare Scenario (NWS) Application

Table 2: System interdependency and development risk (covariance)

<table>
<thead>
<tr>
<th>Variable Depth</th>
<th>Multi Fcn Tow</th>
<th>Lightweight tow</th>
<th>RAMCS II</th>
<th>ALMDS (MH-60)</th>
<th>N-LOS Missiles</th>
<th>Griffin Missiles</th>
<th>Package System 1</th>
<th>Package System 2</th>
<th>Package System 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Variable Depth</td>
<td>0.1</td>
<td>0.6</td>
<td>0.2</td>
<td>0.3</td>
<td>0.1</td>
<td>0.2</td>
<td>0.1</td>
<td>0.2</td>
<td>0.2</td>
</tr>
<tr>
<td>Multi Fcn Tow</td>
<td>0</td>
<td>0</td>
<td>0.2</td>
<td>0</td>
<td>0.1</td>
<td>0.2</td>
<td>0.1</td>
<td>0.2</td>
<td>0.2</td>
</tr>
<tr>
<td>Lightweight tow</td>
<td>0</td>
<td>0</td>
<td>0.2</td>
<td>0</td>
<td>0.1</td>
<td>0.2</td>
<td>0.1</td>
<td>0.2</td>
<td>0.2</td>
</tr>
<tr>
<td>RAMCS II</td>
<td>0</td>
<td>0</td>
<td>0.2</td>
<td>0.3</td>
<td>0.1</td>
<td>0.2</td>
<td>0.1</td>
<td>0.2</td>
<td>0.2</td>
</tr>
<tr>
<td>ALMDS (MH-60)</td>
<td>0</td>
<td>0</td>
<td>0.2</td>
<td>0</td>
<td>0.1</td>
<td>0.2</td>
<td>0.1</td>
<td>0.2</td>
<td>0.2</td>
</tr>
<tr>
<td>N-LOS Missiles</td>
<td>0</td>
<td>0</td>
<td>0.2</td>
<td>0</td>
<td>0.1</td>
<td>0.2</td>
<td>0.1</td>
<td>0.2</td>
<td>0.2</td>
</tr>
<tr>
<td>Griffin Missiles</td>
<td>0</td>
<td>0</td>
<td>0.2</td>
<td>0</td>
<td>0.1</td>
<td>0.2</td>
<td>0.1</td>
<td>0.2</td>
<td>0.2</td>
</tr>
<tr>
<td>Package System 1</td>
<td>0</td>
<td>0.1</td>
<td>0.2</td>
<td>0.3</td>
<td>0.5</td>
<td>0.2</td>
<td>0.1</td>
<td>0.3</td>
<td>0.2</td>
</tr>
<tr>
<td>Package System 2</td>
<td>0</td>
<td>0.1</td>
<td>0.2</td>
<td>0.3</td>
<td>0.5</td>
<td>0.2</td>
<td>0.1</td>
<td>0.3</td>
<td>0.2</td>
</tr>
<tr>
<td>Package System 3</td>
<td>0</td>
<td>0.1</td>
<td>0.2</td>
<td>0.3</td>
<td>0.5</td>
<td>0.2</td>
<td>0.1</td>
<td>0.3</td>
<td>0.2</td>
</tr>
</tbody>
</table>

System Capabilities

<table>
<thead>
<tr>
<th>Package</th>
<th>System Req.</th>
<th>Develop. Time</th>
<th>Acq. Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>Package</td>
<td></td>
<td>(Years)</td>
<td>($)</td>
</tr>
<tr>
<td>ASW</td>
<td></td>
<td>250</td>
<td>3</td>
</tr>
<tr>
<td>Variable Depth</td>
<td></td>
<td>150</td>
<td>2</td>
</tr>
<tr>
<td>Multi Fcn Tow</td>
<td></td>
<td>100</td>
<td>4</td>
</tr>
<tr>
<td>Lightweight tow</td>
<td></td>
<td>200</td>
<td>1</td>
</tr>
<tr>
<td>RAMCS II</td>
<td></td>
<td>100</td>
<td>4</td>
</tr>
<tr>
<td>ALMDS (MH-60)</td>
<td></td>
<td>200</td>
<td>3</td>
</tr>
<tr>
<td>N-LOS Missiles</td>
<td></td>
<td>100</td>
<td>4</td>
</tr>
<tr>
<td>Griffin Missiles</td>
<td></td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>Seaframe package</td>
<td></td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>&amp; Combat package</td>
<td></td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>Management package</td>
<td></td>
<td>0</td>
<td>5</td>
</tr>
</tbody>
</table>
Example Tradespace Results
Method 2: CVaR Optimization Problem

**Objective**
Maximize Performance Index

\[ \max \left( \sum_q \left( \frac{S_{qc} - R_c}{R_c} \cdot w \cdot X_q^e \right) \right) \]

**Constraints**

- **Requirement Satisfaction**
  \[ \sum_q S_{qc} X_q^B \geq \sum_q S_{qR} X_q^B \quad \text{(Satisfy Requirements)} \]

- **Big-M Formulation**
  \[ \sum_c X_{cij} - X_{ij} M \leq 0 \]
  \[ M \sum_c X_{cij} - X_{ij} \geq 0 \]

- **Flow Balance Constraint**
  \[ X_{cij} - \sum_j X_{cij} - X_{ij}^B S_{rj} = 0 \]

- **Bandwidth Limit**
  \[ X_{cij} \leq \text{Limit}_{cij} \]

- **Node Connection Compatibility**
  \[ X_1^B + \ldots + X_n^B = D \quad \text{(System Compatibility)} \]
  \[ (e.g. X_1^B + X_1^B + X_1^B = 1) \]
  \[ X_{cij} = 0 \quad i,j \in \{\text{incompatible set}\} \]
  \[ \in \{0, 1\} \quad \text{(binary)} \]
Method 2: CVaR Optimization Problem

**Objective**

Minimize CVaR

\[
\min_{x,z,\gamma} \left\{ \gamma + \frac{1}{(1 - \alpha)S} \sum_{s=1}^{S} z_s \right\}
\]

**Constraints**

**Requirement Satisfaction**

\[ z_s \geq \sum_i (b_i - y_{is})'x_i^B - \gamma \]

**Expected SoS Performance (change SoS_{cap} to get Pareto)**

\[ \sum_i b x_i^B \geq SoS_{CAP} \]

**Capacity Sufficiency**

\[ \sum_j x_{cij} \leq x_i^B S_{ci} \]

**Requirements Satisfaction**

\[ \sum_i x_{cij} \geq x_j^B S_{rj} \]

**Node Connection Compatibility**

\[ x_i + L + x_n = L \]
More Constraints ....

Rest of constraints reflect combinatorial rules as usual

\[
\begin{align*}
\sum_{c} x_{cij} - x_{ij} M & \leq 0 \\
M \sum_{c} x_{cij} - x_{ij} & \geq 0 \\
x_{ij} & \leq \text{Limit}_{ij} \\
\sum_{i} x_{cij} - \sum_{j} x_{cij} - x_j^B S_{rj} & = 0 \\
x_{cij} & \leq \text{Limit}_{cij} \\
x_{cij} = 0 & \quad c \in \text{capability} \\
x_{cij} & \in \text{real, binary }, x_j^B \in \text{binary}
\end{align*}
\]
Method 2: Example NWS application

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Sys. 1</td>
<td>A</td>
<td>0</td>
<td>70</td>
<td>0</td>
<td>0</td>
<td>100</td>
<td>200</td>
</tr>
<tr>
<td></td>
<td>B</td>
<td>0</td>
<td>60</td>
<td>0</td>
<td>0</td>
<td>90</td>
<td>120</td>
</tr>
<tr>
<td></td>
<td>C</td>
<td>0</td>
<td>50</td>
<td>0</td>
<td>0</td>
<td>70</td>
<td>100</td>
</tr>
<tr>
<td>Sys. 2</td>
<td>A</td>
<td>0</td>
<td>0</td>
<td>30</td>
<td>0</td>
<td>70</td>
<td>120</td>
</tr>
<tr>
<td></td>
<td>B</td>
<td>0</td>
<td>0</td>
<td>20</td>
<td>0</td>
<td>90</td>
<td>150</td>
</tr>
<tr>
<td></td>
<td>A</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>100</td>
</tr>
<tr>
<td>Sys. 3</td>
<td>B</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>250</td>
</tr>
<tr>
<td></td>
<td>A</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>300</td>
</tr>
<tr>
<td></td>
<td>B</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>100</td>
</tr>
<tr>
<td>Sys. 4</td>
<td>A</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>300</td>
</tr>
<tr>
<td></td>
<td>B</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>150</td>
</tr>
<tr>
<td></td>
<td>C</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>200</td>
</tr>
<tr>
<td></td>
<td>D</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>500</td>
</tr>
<tr>
<td></td>
<td>E</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Sys. 5</td>
<td>F</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Candidate Systems + Simulated Data in Portfolio Optimization Problem
Method 3: Robust Constraints (Bertsimas-Sim)

- Use Bertsimas-Sim approach to uncertain (data uncertainty) constraints
- Benefits: Linear Programming approach, constraint violation control with probabilistic guarantees, extends to discrete optimization

Adjust conservatism $\Gamma_i$ term to control probability of constraint violation

Conservatism Added
(This can be converted to an LP == easy to solve even for large problems)
Method 3: Example NWS application

Candidate Systems + Uncertainty Bounds on Relevant Constraints for Portfolio Problem
Robust Portfolio Suite Overview

Method 1: Mean-Variance Robust Portfolio

- **Objective:**
  - Maximize Performance Index
  - Capability: $\sum \frac{W_i^c}{\sigma^c}$
  - Risk: $\sum \frac{W_i^r}{\sigma^r}$
  - Cost: $\sum W_i^c$

- **Constraints:**
  - Portfolio Fraction
  - Portfolio Total Budget
  - Requirements Satisfaction

- **Selection Rules (Compatibility):**
  - Robust Formulation (Tutuncu & Koenig 2004)

Method 2: CVaR Optimization Problem

- **Objective:**
  - Maximize Performance Index
  - Capability: $\sum W_i^c$
  - Risk: CVaR

- **Constraints:**
  - Requirement Satisfaction
  - Big-M Formulation (number of connections)
  - Flow Balance Constraint
  - Bandwidth Limit
  - Node Connection Compatibility

Method 3: Robust Constraints (Bertsimas-Sim)

- Use Bertsimas-Sim approach to uncertain (data uncertainty) constraints
- Benefits: Linear Programming approach, constraint violation control with probabilistic guarantees, extends to discrete optimization

Table 2: System interdependency and development risk (covariance)

- Variable Description
- Multi-Fracture Model
- High-Maintenance
- Long-Term
- RAMS
- Reliability
- Availability
- Maintainability
- Safety
- Overall System

Graph 1: Performance Efficiency Frontier

- Portfolio 1
- Portfolio 2
- Portfolio 3

Graph 2: Self Performance Index vs Variance of $X^2$

- Portfolio 1
- Portfolio 2
- Portfolio 3

Graph 3: Probability of Power Constraint Violation

- Probabilistic Power Constraint Violation
- Probabilistic Overall Constraint Violation