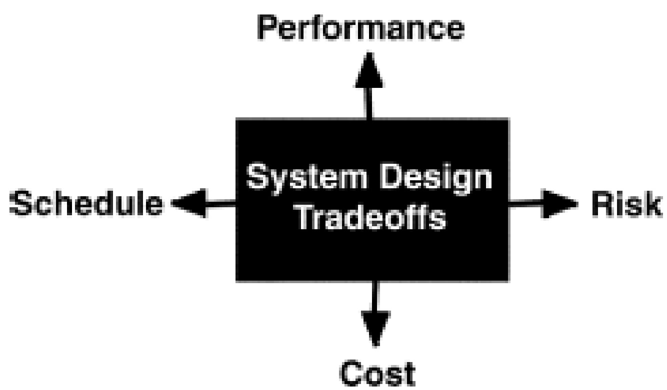
Isoperformance: Analysis and Design of Complex Systems with Desired Outcomes

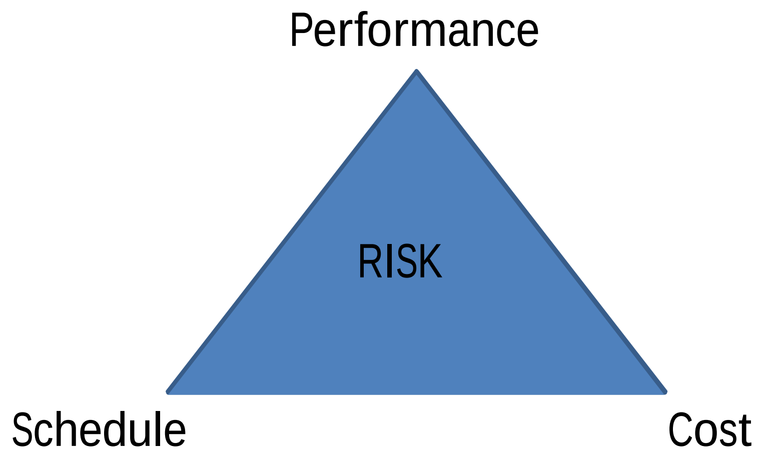
O. De Weck and M. Jones

Presented by
Student 3 and Student 4
September 25, 2009

Motivation

- Traditional design is based on maximizing performance.
- This focus on pure performance maximization can result in drawbacks in terms of cost, schedule, and risk





- Diminishing returns at the margin of the performance envelope
- Augustine Law Number XV: “The last 10 percent of performance generates 1/3 of the cost and 2/3 of the problems.”

Introducing isoperformance design

- <u>Core concept</u>: accept "good enough" performance instead of seeking "best achievable"
- Performance = NIB (vs. performance = LIB). Willingness to trade some performance for benefits in terms of cost, schedule or risk.

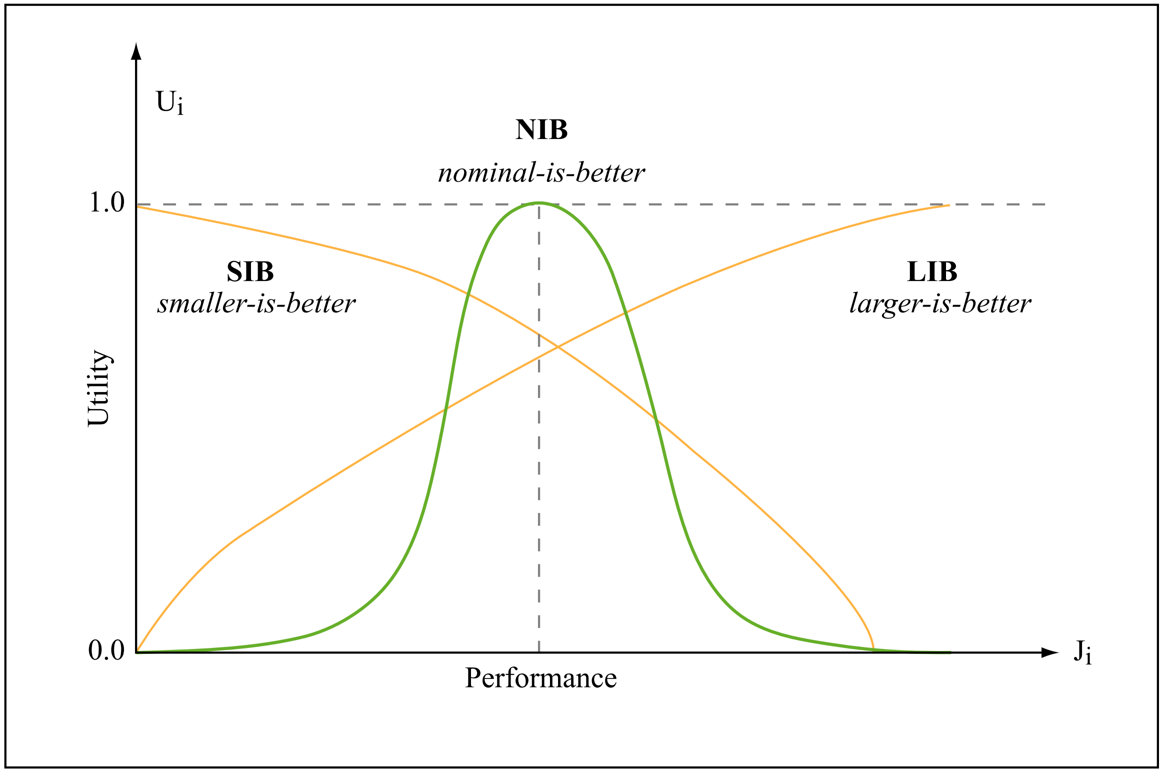


Image by MIT OpenCourseWare.

- Inverse method that finds *all the designs* that satisfy a certain level of performance: these designs exhibit ***isoperformance***
  - E.g. design of antennae, digital circuits, or airfoils.
- This subspace of designs can then be explored to find the efficient ones in terms of secondary metrics.

Methodology

1. Find performance-invariant set of designs
   - Branch and bound algorithm
   - Gradient-based contour following algorithm
   - Progressive vector spline approximation
2. Find an efficient subset of designs
   - Incorporate cost and risk considerations in this step
   - Identify non-inferior designs
   - These constitute a Pareto efficient frontier in the design space
3. Select final design
   - Stakeholder consensus (non numerical objectives)

Methodology

B: Set of all designs bounded by "side constraints"

F: Set of feasible designs

I: Set of isoperforming designs

P: Pareto efficient set of non-inferior designs

E: Set of Pareto efficient designs that satisfy isoperformance requirement

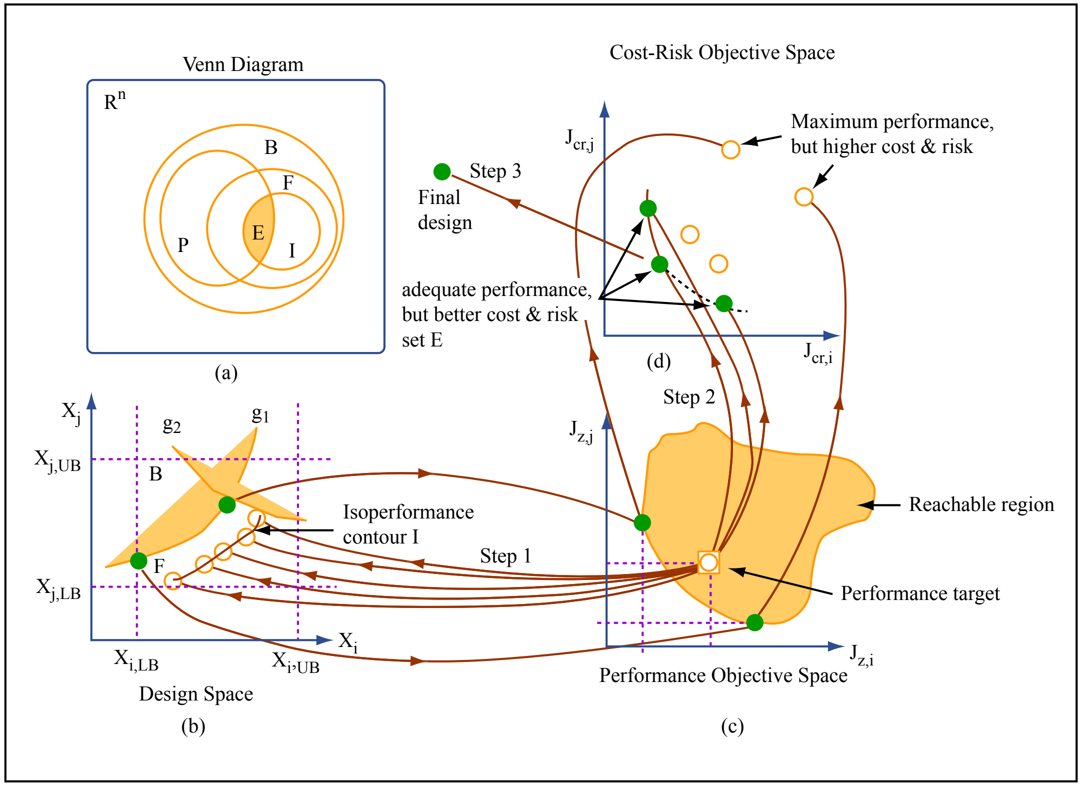
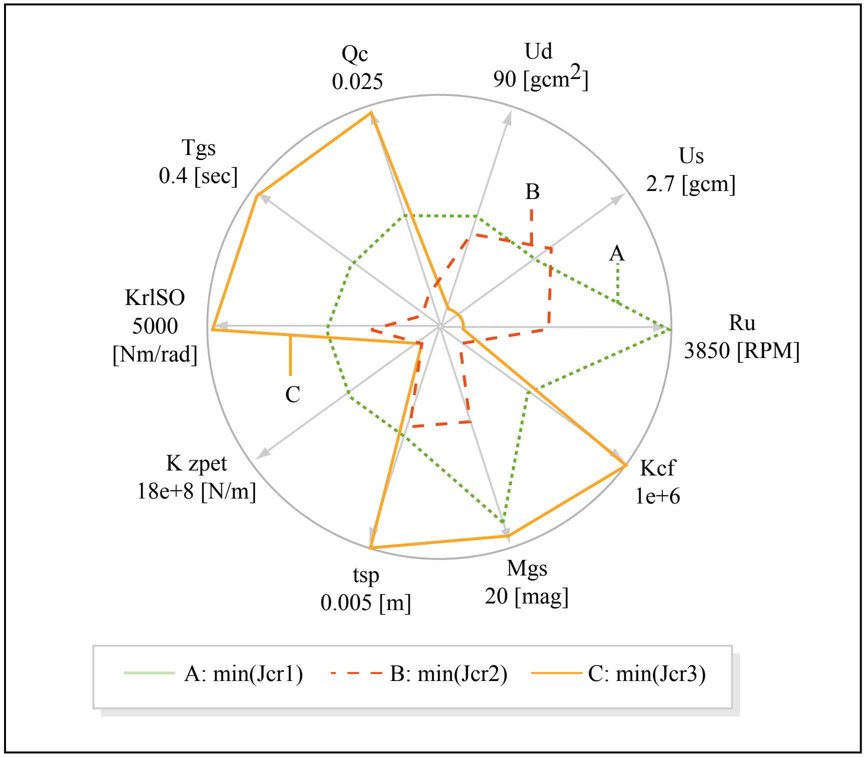


Image by MIT OpenCourseWare.

Example 1: Space Telescope

- Step 1: Several hundred potential designs identified via gradient gradient-based contour following
- Step 2: Three cost-risk variables
  - Closeness to mid-range of design variables (A) "Best balance"
  - Magnitude of control gain (B) "Minimum energy"
  - Sensitivity of performance to perturbations (C) "Robust"



	Performance Objectives		Cost-Risk Objectives		
Design	$J_{z,1}$ WFE	$J_{z,2}$ LOS	$J_{cr,1}$	$J_{cr,2}$	$J_{cr,3}$
A: x^1_{iso}	20.000	5.2013	<u>0.6324</u>	0.4668	14.32%
B: x^2_{iso}	20.0012	5.0253	0.8960	<u>0.0017</u>	8.7883%
C: x^3_{iso}	20.0001	4.8559	1.5627	1.000	<u>5.3067%</u>

Isoperformance tolerance τ=0.05.

Topics for discussion (1/3)

- The design selection step of the methodology avoids any direct comparison across metrics such as multi-attribute utility theory and states that the final decision is made on the basis of non quantified objectives and stakeholder consensus. Can this process be captured in a model?

- In general, there is an infinite set of designs that are capable of providing the desired level of performance. How do we discretize the problem so that the algorithm can explore the design tradespace? How do we choose the design variables, their range of values, and their discretization? How do we choose our figures or merit?

Topics for discussion (2/3)

- Can we explore in a systematic way the space of possible system models (i.e. one step back from exploring the design tradespace)?

- Is the method presented in the isoperformance paper a particular example of multi-attribute tradespace exploration where there is a constraint for all the architectures to have the same performance? What are the similarities/differences?

Topics for discussion (3/3)

- To what extent do requirements influence the applicability of isoperformance design? Are there certain ways in which systems engineers should formulate requirements in order to best use the isoperformance design process, or is it applicable regardless of how requirements are defined?

16.842 Fundamentals of Systems Engineering
Fall 2009