Strategies for Design Optimization: Lessons from Automotive Systems

Erik Goodman, VP Technology
Red Cedar Technology
East Lansing, MI
USA
Outline

• Introduction to automated design exploration
• The REAL WORLD – and concessions to it:
  • Costly evaluations → Not your father’s GA
  • Typical stepwise design process
    • Design of Experiments
    • EC/hybrid search
    • SHERPA
    • CIA
  • “The Box” and a way to escape it… sometimes
    • COMPOSE
• Example of Component Crashworthiness Optimization
Typical Design Process

- Initial Design Concept
- Specific Design Candidate
- Build Analysis Model(s)
- Execute the Analyses
- Design Requirements Met?
- Yes → Final Design
- No → Modify Design (Intuition)

Automate?

Factors:
- Time
- Money
- Intellectual Capital

TECHNOLOGY
red cedar
Automated Design Process

User-Specified Design Model

Modify Design (autom.)

Initial Design Concept

Representative Design(s)

Build Analysis Model(s)

Execute the Analyses

Termination Criterion Met?

No

Yes

Optimized Design(s)
or
DOE Results
Robustness Assessment

Yes

No
Main Benefits

• **Automates** search for design alternatives with improved performance and cost
  ⭐ more efficient and thorough search

• **Reduces** design time from weeks to days
  ⭐ significant cost reduction

• **Accelerates product and process innovation**
  ⭐ increased competitive advantage

• **Integrates and leverages** existing investment in CAD/CAE tools and hardware
  ⭐ better utilization of capital

• **Improves design robustness**
  ⭐ six sigma
Main Ingredients Needed

- Process Integration
- Design Space Exploration
  - Design of Experiments
  - Design Optimization
- Reliability and Robustness
Design of Experiments

Design exploration

- Design variable screening
- Sensitivity of design performance to changes in parameters
- Suggests design improvements

Local response surface

- Surrogate analysis model
- Reduces number of full evaluations

NOT a direct optimization method
Design of Experiments

Examples of Sampling Methods:
- Full factorial
- Fractional factorial
- Central composite
- Latin hypercube
- Plackett-Burman
- Taguchi orthogonal array
- D-Optimal
- User defined

Post-Processing
- ANOVA
- Pareto charts
- Main and Interaction Effects
- Response Surfaces, …
Design Optimization Scenarios

- Seek small improvements to an already good design
  Local search, small variable range
- Seek best design or concept within a large design space
  Global search, large variable range
- Seek design that best matches a desired performance
  Minimize difference between actual and desired responses
- Modify an existing design to meet new performance targets
Parameter Optimization

Objective:

Search the performance response surface to find the highest peak or lowest valley within the feasible range

- Local searches may yield only incremental improvement
- Typically don’t know the nature of the surface before search begins
- Number of parameters may be large
- Evaluations may be expensive – CANNOT DO MANY
- Shape optimization often involves multi-modal design spaces
Design Approaches

- Perform a DOE, fit a response surface, and search the surface
  Usually valid only in local neighborhood of a given design
- Perform a DOE, select most important variables, and perform optimization over subset of variables
  Lots of preparatory work to perform optimization
- Define engineering and business goals and a meaningful design space, and then search for the design that best meets those goals
  Less overall time spent solving the problem
  Greater potential to find better or innovative designs
How Can We Make Intractable Problems (more) Tractable?

Examples of tools that can be helpful:

- Response surfaces
- Multiple, Self-Tuning Search Methods
- Parallel/cluster/grid organization
- Multiple representations/agents – problem decomposition
- “Medicine” against convergence other than large populations
- Model decomposition – “COMPOSE”
WHAT IS NEEDED

Number of design parameters (evaluations)

Traditional Methods

Time needed to analyze each proposed design
Response Surfaces

- Creates an approximate *model* of the response (a surrogate) as a function of the factors
- Low-order polynomials are often used as surrogate functions
- Typically, response surface models are “least-squares” fit to a sampling of the actual response
- This sampling may be controlled via design of experiments or using points already evaluated and saved
**Response Surfaces**

- **Strengths**
  - Eliminate many “full” performance evaluations
  - Potentially define fruitful search directions even if design criteria are not locally differentiable

- **Weaknesses**
  - Not designed to escape local optima
  - Design variables should be continuous in nature
  - Often works well only in local search
Choosing a Method or a Design Space?

- Design space is a function of:
  - Number, type, and range of design variables
  - Objectives and constraints
- Preferable to define design problem first, and then select search method
  - Which method to choose depends upon the design space
  - But the nature of the design space is not known, unless the problem has been solved before
- Alternative: select search method first, and then make design problem fit that method’s capabilities
  - But this limits the problem scope and number of variables, and requires high level of knowledge about the method
Example of Self-Tuning Search Method: SHERPA

• Hybrid
  – Multiple methods used simultaneously, not sequentially
  – Takes advantage of best attributes of each method
  – Both global and local search techniques are used

• Adaptive
  – Each method adapts itself to the design space
  – Master controller determines which methods get used and how much
  – Tries to learn efficiently about design space and effectively search even very complicated spaces

BUT no guarantee of suitability for any particular problem!
Multi-Agent Search: Cooperative Independent Agents (CIA)

- Each search agent
  - Controls an independent search process
  - Uses a combination of search methods
  - Uses own representation of the problem

- Multiple agents working in parallel
  - Problem decomposition
  - Diverse points of view
  - Shared discovery

Team: Intellectual Diversity
Can decompose design space according to:

- Physical/spatial domain
- Temporal extent of simulation
- Number of design variables
- Resolution of design variables
- Stochasticity of variables
- Performance measures
- Analysis models
- Search methods
- …
Putting It Together -- Example: Hydroformed Lower Rail

Crush zone

Crush zone
Shape Design Variables

67 design variables:
66 control points and one gage thickness
Optimization Statement

- Identify the rail shape and thickness
- Maximize energy absorbed in crush zone
- Subject to constraints on:
  - Peak force
  - Mass
  - Manufacturability
Three-Agent Topology

Discretize:
• crush time ( reduces CPU time )
• design variables ( reduces design space )

<table>
<thead>
<tr>
<th>Crush Time</th>
<th>Agent Topology</th>
<th>Design Variable Discretization</th>
</tr>
</thead>
<tbody>
<tr>
<td>t=6 ms</td>
<td>0</td>
<td>Coarse</td>
</tr>
<tr>
<td>t=10 ms</td>
<td>1</td>
<td>Medium</td>
</tr>
<tr>
<td>t=14 ms</td>
<td>2</td>
<td>Refined</td>
</tr>
</tbody>
</table>
Energy Absorbed
HEEDS Optimized Design
HEEDS Optimized Design
Validation
Lower Rail Benefits

- Peak force reduction by 30%
- Energy absorption increased by 100%
- Weight reduction by 20%
- Overall crash response resulted in equivalent of FIVE STAR rating
COMPOSE Breakthrough

COMPOSE –

COMPonent Optimization within a System Environment:

• Pull out a component, optimize, put back in → failure
• Iterate above → failure
• COMPOSE allows subsystems (quick to simulate) to be optimized and retain their performance when reintegrated in whole system, but requires only a FEW system-level evaluations!!!
  • Reduces design time by factor of 10 – 1,000 for some problems
  • Allows search over large number of design variables
  • Makes many intractable problems practical to optimize
Spatial Decomposition

System

System with subsystems denoted

Subsystems
COMPOSE – COMPonent Optimization within a System Environment

*Enables* high fidelity design of subsystems in *highly coupled* complex systems (10^1 – 10^3 times speedup)
COMPOSE

- Based on decomposition
- Most CPU effort to design subsystem (component)
- Small number (3-8) of system level analyses
- Full coupling maintained between system and subsystem
- Large number of variables can be studied
- CPU time reduced by factor of 10 – 1,000
Keys to the Convergence Strategy

• Make subsystem design robust against changes in interactions (subsystem boundary conditions)

• Similar modality of response in sequential designs
  – Gradients
  – Eigenvectors
  – Etc.

• Consequence: Higher level of system robustness
Vehicle Rail – Shape Optimization

Objective: Maximize Energy Absorbed
Constraint: Reaction Force
Boundary Conditions from System Model
Subsystem Design Variables

- Individually designed rails
- 7 Cross-sections on each rail
- 10 Design- master points on each cross-section
- Total of 70 shape design variables; symmetry fixes 140 values
Rail Optimization Results

Rail Energy Absorbed
(30% increase)

System Energy Absorbed
(5.5% increase)

(Optimization over 70 variables using only 6 system evaluations.)
There is no “Easy Button”

HEEDS is:
- An engineering tool
- Not “pixie dust” or an “easy button”

HEEDS requires:
- Good engineering
- Good ideas / concepts
Designing with HEEDS for Future Generation Passenger Compartment (US Auto/Steel Partnership)

- **Stage 1 – Conceptual design for side impact, rollover**
  - Topology optimization
  - Upper body, underbody load paths
  - Ignore fine details of the structural members

- **Stage 2 – Screening**
  - Design of Experiments
  - Identify members or subsystems that have greatest “effect” on performance
  - Primary factors include gage and material properties

- **Stage 3 – Detailed design of cross-sections**
  - Optimization of shape, gage, material to enhance performance while reducing mass