Object-based Modeling and Optimization with Genetic Algorithms

Nicola Senin, David Wallace, Nick Borland
MIT CADlab

DOME Project (distributed object-based modeling environment)

Graduate Students:
Juan Deniz, Julie Eisenard, Jaehyun Kim, Jun-Beom Kim, Ben Linder, Bill Liteplo, Jeff Lyons, Shaun Meredith, Ines Sousa, Chris Tan, Priscilla Wang

Sponsors:
Ford Motor Company, NSF Center for Innovation in Product Development, Alliance for Global Sustainability
Object-based Modeling and Optimization

Presentation structure

Design need
Design modeling vision and example
Search and tradeoff vision and example
Object modeling formalism
Optimization formalism
Prototype implementation
Test results
Real world applications and summary
Design Need

Integrated system modeling

Most design problems involve complex systems, so they are difficult to understand intuitively.

Capabilities needed for system modeling or simulation require many participants from different disciplines.

Each participant uses their own software, computing tools, and proprietary models as is appropriate for their discipline.

Building in-depth system models for technology design is very time consuming and the systems are not static.

Development time and cost
Design Need
Tradeoff and search support

Product designers are not decision analysts or mathematical programmers
System models may be computationally demanding
Product designers are interested in multiple alternatives
Design tradeoffs need to be visualized

Product performance
Design Modeling Vision

DOME project

Participants empowered to publish services on WWW as part of their workflow

Distributed participants relate or modify services and the resulting network operates like an integrated product model

Designers and experts have control, flexibility
Design Modeling Vision Example
Ford movable glass system
Design Modeling Vision Example

Ford Engineering Model
Design Modeling Vision Example

Engineer defines service interface
Design Modeling Vision Example
Engineer adds Excel object in DOME environment
Design Modeling Vision Example
Engineer’s model operable over internet
Design Modeling Vision Example
CAD Designer publishes services on the internet
Design Modeling Vision Example

Systems engineer relates CAD and engineering services
Design Modeling Vision Example

Engineers relate models to services of seal suppliers
Search and Tradeoff Vision

DOME project

Participants can add the services of software tools to their design models as needed to make tradeoffs and search for multiple solutions
Search and Tradeoff Vision

Designer adds services of tradeoff support software
Search and Tradeoff Vision

Designer adds services of genetic search object
Object Modeling Formalism

Modules and services

Module (containing embedded model)

Services provided

Services required

Strength Analysis (Excel)

Hoop stress

Thickness, diameter, height
Object Modeling Formalism
Engineering Variable Modules

A. getExpectedValue()

Beta

B. setLeftBound()

Interval

C. setValue()

Delta

B. getMean()
Object Modeling Formalism

Container Modules with Solvers

Relation R1 {
    Cost = CostPerUnitLength * PillarLength;
}

CostModule.getCost()
Object Modeling Formalism

Catalog Modules

Seal Supplier Catalog

Current Selection

Seal Supplier

Door Glass Module
Mixed Optimization Formalism

Genetic search object

Problem Model

Changes to the design variables

Optimization Module

Evaluation services (from Decision Modules)

Design Search Variables

Search Engine

Objective Function

Objective Function
Mixed Optimization Formalism

Genome representation

Direct Encoding

| 2.123 | 3 | $3.2 \cdot 10^{-2}$ | $3.8 \cdot 10^{-2}$ |
Optimization Formalism

Real number similarity

Distance\((a, b)\) = \(|a - b|\)

Similarity\((a, b)\) = 1 - \(\frac{\text{Distance}(a, b)}{\text{maxDistance}}\)

\(\text{A}=0.3\) \(\text{B}=0.7\) \(\text{d}=0.4\)
Optimization Formalism

Catalog selection similarity

\[
\text{Distance}(a, b) = \begin{cases} 
|\text{pos}(a) - \text{pos}(b)|, & \text{ordered} \\
1, & \text{unordered}
\end{cases}
\]  

[catalog Units]

\[
\text{Similarity}(a, b) = \begin{cases} 
1 - \frac{\text{Distance}(a, b)}{(n\text{Modules} - 1)^\beta}, & n\text{Modules} > 1 \\
0, & n\text{Modules} = 1
\end{cases}
\]  

[catalog units]  

[catalog units]
Optimization Formalism

Similarity in catalog hierarchies

Electric Motors

Vendor A

Vendor B

Motors of vendor B

Model 1

Model 2

Max. Power

Max. Power
Optimization Formalism

Overall similarity measure

\[
Similarity(g_1, g_2) = \sum_{i=1}^{n} \frac{Similarity(\text{design variable}_i) \ast w_i}{n}
\]

\[
\sum_{i}^{n} w_i = 1
\]
Optimization Formalism

Crossover operators

Real number BLX crossover

Catalog BLX crossover
Prototype Implementation
DOME Concept

Optimizer Module
MS Excel Module
Ideas Module
Relation Solver
Java Server
Java Client
DOME Kernel
DOME Kernel
DOME Kernel
C++ API
C++ API
C++ API
C++ API
C++ API
C++ API
Java
Server
RMI
JNI API
CORBA
COM
COM
CORBA
SDRC Ideas
MS Excel
Ideas
Relation Solver
Optimizer Module
MS Excel Module
Ideas Module
DOME Kernel C++
Java Client
DOME Kernel
DOME Kernel
DOME Kernel
C++
Test Results

Genetic search object

Global Algorithms
Simple GA
Steady state GA

Niching Algorithms
Deterministic crowding
Restricted tournament selection
Struggle
Test Results

Struggle GA

Randomly seed population of genomes
Repeat
  Select parents P1 and P2
  Cross P1 and P2 yielding an offspring C
  Apply mutation with a probability \( p_{\text{mut}} \) on C, yielding C’
  Find individual R most similar to C’ in the entire population
  If fitnessScore(C’)>fitnessScore(R) replace R with C’
Until Stop Criterion
Test Results

Global optimization

Convergence to 99.99% of global optimum (probability)

Reliability finding Scheckels foxholes global optimum. Probability of converging within 99.99% of the known global optimum in 300 generations (averaged over 10 runs, popsize=20)
Reliability finding Scheckels foxholes global optimum using direct or binary genome representations. Probability of converging within 99.99% of the known global optimum in 300 generations (averaged over 10 runs, popsize=20)
Test Results
Identification of local optima

Average number of Scheckels local optima found (within 99.99% of local peak). Averaged over 10 runs (popsise=20, number of local optima = 25)
Test Results

Struggle GA identification of local optima

Number of individuals on each local optima (within 99.99% of local peak) after 500 generations.

(popsize=50, number of local optima = 25)
Real World Applications

On-going DOME projects

Tokyo Half Project *(University of Tokyo)*

Automotive Movable Glass System *(Ford Motor Company)*

US Navy
Object-based Modeling and Optimization

Vision summary

Participants use their own specialized modeling tools

Participants publish modeling services on Internet

Services are mathematically linked to form a decentralized system model

Services of tools for decision support and optimization can be incorporated into the service exchange network
Object-based Modeling and Optimization

Presentation summary

Results from example application
Object modeling formalism
Optimization formalism
Test results