

Optimization Strategies to Explore Multiple Optimal Solutions and Its Application to Restraint System Design

Zhendan Xue and Sumeet Parashar
ESTECO North America

Guosong Li and Yan Fu
Ford Motor Co.

ABSTRACT

Design optimization techniques are widely used to drive designs toward a global or a near global optimal solution. However, the achieved optimal solution often appears to be the only choice that an engineer/designer can select as the final design. This is caused by either problem topology or by the nature of optimization algorithms to converge quickly in local/global optimal or both. Problem topology can be unimodal or multimodal with many local and/or global optimal solutions. For multimodal problems, most global algorithms tend to exploit the global optimal solution quickly but at the same time leaving the engineer with only one choice of design. The paper explores the application of genetic algorithms (GA), simulated annealing (SA), and mixed integer problem sequential quadratic programming (MIPSQP) to find multiple local and global solutions using single objective optimization formulation. The techniques are applied to a couple of mathematical problems as well as a restraint system design problem. Results are compared thoroughly in terms of the quality of local convergence, convergence speed, and most importantly the ability to explore multiple local and global solutions.

CITATION: Xue, Z., Parashar, S., Li, G. and Fu, Y., "Optimization Strategies to Explore Multiple Optimal Solutions and Its Application to Restraint System Design," *SAE Int. J. Passeng. Cars - Mech. Syst.* 5(1):2012, doi:10.4271/2012-01-0578.

INTRODUCTION

During the product design phase a designer is required to design a product that achieves high performance, high reliability, along with less complexity at a low cost, and deliver them within a tight time constraint. Design optimization technology, with its capability of driving multiple design variables simultaneously toward the optimal solution systematically and automatically, can greatly reduce the development time and improve the overall product competitiveness. In the automotive industry, occupant restraint system design is often relied on the design optimization approach to get the final design which will comply to the highest standards of vehicle safety requirements, such as National Highway Traffic Safety Administration (NHTSA)'s New Car Assessment Program (NCAP) 5-Star safety ratings. Without design optimization it would be very time-consuming and difficult to find designs with low complexity and robust 5-Star safety ratings.

Design optimization software tools, such as modeFRONTIER¹, are often used to model and automate the

optimization procedure to find the good design solutions. Optimization algorithms tend to converge to a global or a local optimal solution. However, this often leads the analyst with only one optimal design instead of many design alternatives which may be different in configuration yet equally good in performance. In the event where the prescribed optimal solution cannot be produced either because of supplier shortage, or production delays, or management decision to reduce complexity, etc; another optimization run may be needed to find new optimal solution. Without alternative optimal designs, the product development time could be increased at the late design phase. Using a multi-objective optimization formulation can address this issue to certain degree by providing a Pareto frontier of optimal designs. However, such a formulation would require the analyst to articulate the additional and often competitive objectives a priori which may not be obvious or easy to formulate in advance.

In this paper, single objective optimization formulation is used to explore the application of several design optimization algorithms such as: genetic algorithms (GA)², simulated



Documents

Document Sharing Hub

annealing (SA)^{3,4}, and sequential quadratic programming (SQP)^{5,6} to find multiple local and global optimal solutions. First, a mathematical problem is used to judge their ability to explore the multiple local and global optimal solutions. Later, the same algorithms are applied to a real-world occupant restraint system design problem. The results from different optimization algorithms are then compared in terms of the quality of local/global convergence, convergence speed, and particularly the ability to explore multiple local and global optimal solutions to achieve the desired objective.

DESIGN OPTIMIZATION AND OPTIMIZATION ALGORITHMS

One of the most important aspects of design optimization is the problem formulation itself. This includes selection of design variable, setting objectives and constraints. Based on the formulation, the second most critical element is the selection of optimization algorithm. This paper will focus on algorithm selection for single objective formulation in order to find multiple equally good solutions. A commercial process integration and design optimization tool called modeFRONTIER¹ is used to evaluate the capability of achieving multiple optimal solutions of the selected optimization algorithms. Optimization algorithms include various genetic algorithms^{7, 8, 9}, simulated annealing^{10,11}, mixed integer programming sequential quadratic programming¹², particle swarm optimization^{13,14}, and various gradient-based algorithms among others. Out of these algorithms, four algorithms are selected to identify the multiple optimal solutions based on single objective formulation, and they are Multi-Objective Genetic Algorithms II (MOGA-II), Non-dominated Sorting Genetic Algorithm II (NSGA-II), Multi-Objective Simulated Annealing (MOSA), and Mixed Integer Programming Sequential Quadratic Programming (MIPSQP). MOGA-II is a global search algorithm and it uses a smart multi-search elitism operator to preserve certain outstanding solutions without prematurely converging into local Pareto frontier. NSGA-II is also a global search algorithm. It has a mechanism to guarantee the diversity and spread of solutions in addition to the similar elitism operator as in MOGA-II. MOSA is a special global search algorithm because it has two phases of optimization processes: 1) the “hot” phase encourages global search behavior and can control how widely the search can explore; 2) the “cold” phase encourages local convergence around the “good” points obtained by the initial global search. MIPSQP is a gradient-based algorithm. It has fast local convergence speed while the capability of achieving global optima is highly dependent on the number and location of initial starting points. All of these algorithms can handle unordered discrete variables, which are involved in this study.

CASE STUDY: A MATHEMATICAL TEST PROBLEM

PART1: SIX HUMP CAMEL BACK FUNCTION WITH CONTINUOUS VARIABLES

A multimodal mathematical problem, called two dimensional six hump camel back function¹⁵, is selected to test different algorithms' capability to find multiple optimal solutions. The optimization problem formulation is described in Equation (1).

$$\begin{aligned} \text{Min } f(x_1, x_2) = & (4 - 2.1 \cdot x_1^2 + x_1^{4/3}) \cdot x_1^2 + \\ & x_1 \cdot x_2 + (-4 + 4 \cdot x_2^2) \cdot x_2^2 \\ \text{where } x_1 \in & [-3, 3], x_2 \in [-2, 2] \end{aligned} \quad (1)$$

There are six local minima in total existing within the design space, and two of them are global minima. Figure 1 illustrates this two dimensional test function, and the two global minima and four local minima are all marked in the 2D contour plot. The aforementioned algorithms: MOGA-II, NSGA-II, MOSA, and MIPSQP are tested using the six hump camel function.

Results of MOGA-II

Table 1 shows the parameter settings for MOGA-II and Figure 2 shows the convergence history of the optimization runs using MOGA-II. The x and y axes represent the two variables x_1 and x_2 . Designs are converged to one global minima at $x_1^* = 0.089842$, $x_2^* = -0.712656$, $f(x_1^*, x_2^*) = -1.0316$. Figure 2 also shows that MOGA-II explores near the other global optimal region, but not converge to the exact global minima point at $x_1^* = -0.089842$, $x_2^* = 0.712656$, $f(x_1^*, x_2^*) = -1.0316$.

Results of NSGA-II

Table 2 shows the parameter settings for NSGA-II and Figure 3 shows the convergence history of the optimization runs using NSGA-II. For this test function, NSGA-II converged to one of the two global minima at either $x_1^* = 0.089842$, $x_2^* = -0.712656$, $f(x_1^*, x_2^*) = -1.0316$ or $x_1^* = -0.089842$, $x_2^* = 0.712656$, $f(x_1^*, x_2^*) = -1.0316$ when different combinations of parameter settings are tested.

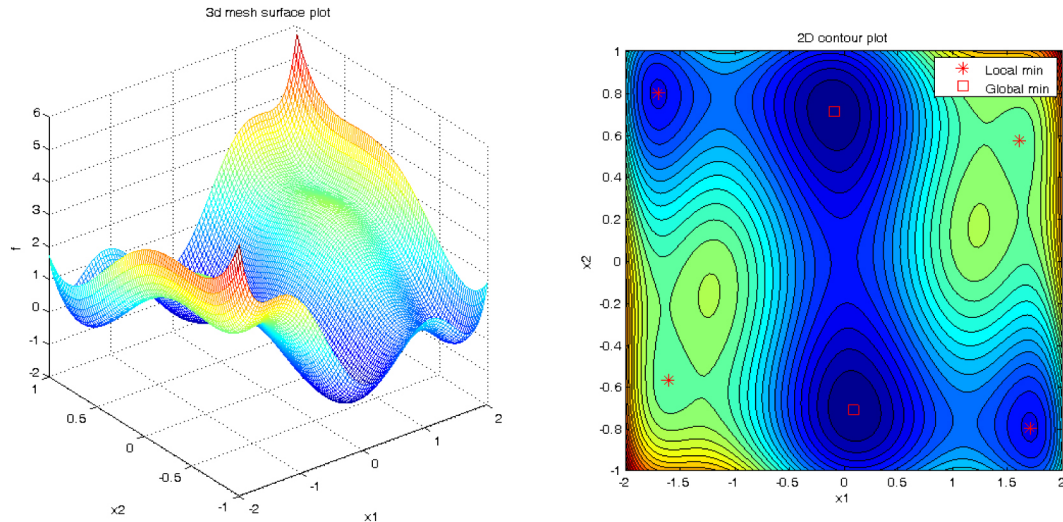


Figure 1. Six hump camel back function with continuous variables.

Table 1. Summary of parameter settings for MOGA-II - continuous six hump camel back function.

Optimization parameters	
Algorithm type	MOGA-II with Adaptive Evolution
Population Size	10 (generated from Uniform Latin Hypercube (ULH) DOE)
Number of Generations [1, 5000]	200
Probability of Directional Crossover [0.0,1.0]	0.5
Probability of Selection [0.0,1.0]	0.05
Probability of Mutation	0.1
Advanced Parameters	
DNA String Mutation Ratio [0.0,1.0]	0.05
Elitism	Enabled

Table 2. Summary of parameter settings for NSGA-II - continuous six hump camel back function.

Optimization parameters	
Algorithm type	Original NSGA-II algorithm
Population Size	10 (generated from ULH DOE)
Number of Generations [1, 5000]	200
Crossover Probability	0.9
Mutation Probability for Real-Coded Vectors	1.0
Advanced Parameter	
Automatic Scaling for Mutation Probability	Active
Distribution Index for Real-Coded Crossover[5e-4,100]	20.0
Distribution Index for Real-Coded Mutation[5e-4, 500]	20.0

Results of MOSA

Table 3 shows the parameter settings for MOSA and Figure 4 shows the convergence history of the optimization runs using MOSA. The results of MOSA are very different from the previous results of MOGA-II and NSGA-II. All of the six minima are explored, and this is largely due to the two phase optimization in MOSA. The “hot” phase enables MOSA to explore the design space to a certain extent which largely depends on the value of “Fraction of ‘Hot’ iterations” and “initial temperature”, the “cold” phase let MOSA

converge to the detected local minima, the number of the local minima that MOSA can converge to is highly related to the value of “Minimum Perturbation Length”. The smaller of this value the greater of the local convergence.

Results of MIPSQP

Table 4 shows the parameter settings for MIPSQP and Figure 5 shows the convergence history of the optimization runs using MIPSQP. The results show that two global minima and two out of four local minima are found by MIPSQP.

Table 3. Summary of parameter settings for MOSA - continuous six hump camel back function.

Optimization parameters	
Algorithm type	MOSA without steady state evolution
Population Size	10 (generated from ULH DOE)
Number of Iterations [1, 5000]	200
Initial Temperature [0.0, 1.0]	1.0
Fraction of 'Hot' Iterations	0.6
Advanced Parameter	
Minimum Perturbation Length (fract. of Diag. [1e-6, 1.0])	0.01

Table 4. Summary of parameter settings for MIPSQP - continuous six hump camel back function.

Optimization parameters	
Population Size	10 (generated from ULH DOE)
Maximum Number of Design Evaluations [1, 9999]	2000
Fraction of Evaluations Allowed For Each DOE Point [0.0, 1.0]	1
Final Termination Accuracy	1.00E-05
Derivatives Parameters	
Approximate Derivatives With	Central Differences
Finite Difference Relative Perturbation [0.0, 1.0]	1.0E-7
Finite Difference Minimum Perturbation Policy	Constant
Constant Minimum Perturbation [0.0, 1E12]	1.0E-7

Since MIPSQP is a gradient-based algorithm, the number of optimal solutions that it can find depends on where the initial starting points are located. Increasing the number of initial starting points will increase the likelihood of finding more local optimal solutions.

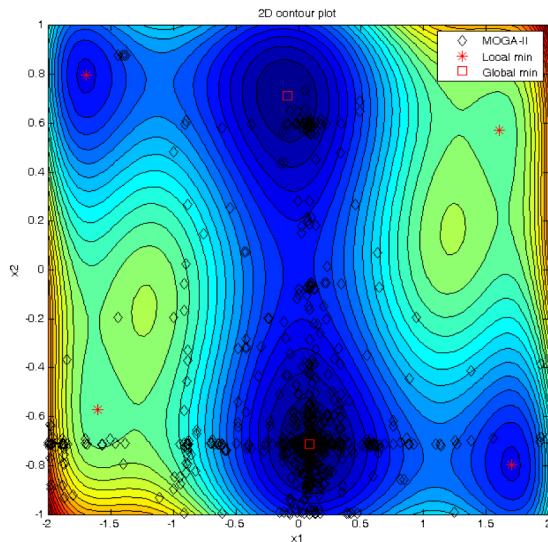


Figure 2. MOGA-II Convergence plot for six hump camel back function.

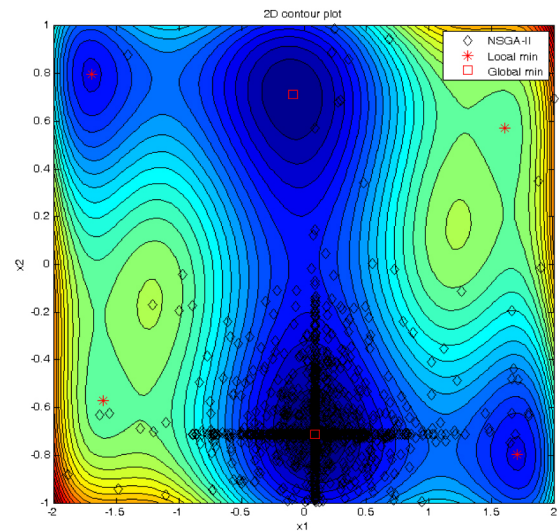


Figure 3. NSGA-II Convergence plot for six hump camel back function.