

Multi-Objective Optimization Using LS-OPT

Tushar Goel, Nielen Stander

Livermore Software Technology Corporation, Livermore, CA, USA

Summary:

Most engineering optimization problems have more than one objective. Often these objectives conflict such that no single solution can be considered optimum with respect to all objectives. Then, the optimum to this problem is a set of solutions known as Pareto optimal set. Traditionally, multi-objective optimization problems are solved by converting the problem into a single-objective optimization problem via, a weighted-sum strategy (combine multiple objectives using designer-specified weights) or introducing constraints on all-but-one objectives (ϵ -constraint strategy). The advantages and disadvantages of such methods are well-documented. However, the most prominent drawback of these methods is that the solution to each optimization problem results into a single optimum without any information about the trade-offs among different objectives. In last few years, there have been significant efforts in developing methods to simultaneously identify many Pareto optimal solutions. Multi-objective evolutionary algorithms (MOEAs) have been among the most successful methods for identifying Pareto optimal fronts. Nevertheless, the applications of such methods for real-world problems, particularly in the area of automobile crash analysis, have been limited. We present one such effort in this area. We implement and utilize a popular MOEA, elitist non-dominated sorting genetic algorithm (NSGA-II). The implementation is validated with three benchmark analytical problems and then a simplified multi-disciplinary car crash worthiness optimization problem that aims to simultaneously minimize HIC and intrusion while constraining the torsional mode frequency was solved.

Keywords:

Multi-objective optimization, Multi-disciplinary optimization, Elitist non-dominated sorting genetic algorithm, Pareto optimal front, crash-worthiness simulation.

1 Introduction

Most engineering problems are required to simultaneously satisfy multiple criteria. Often these criteria conflict (i.e., improving on one criterion deteriorates others) such that no single design can be considered optimal with respect to all objectives. Instead the optimum is a set of solutions, known as Pareto optimal solutions, that reflect trade-offs among objectives. In conventional methods, designers use domain-specific knowledge to develop a cost function to combine different criteria (weighted sum strategy [1]) or specify constraints on all but one objective (ϵ -constraint strategy [2]), and optimize a single objective optimization problem. They repeat multiple simulations with different cost functions or constraint values to identify multiple Pareto optimal solutions. Existing framework in LS-OPT [3] caters to this scenario by allowing user to allocate weights to different objectives or assign multiple constraints. This approach is effective when chosen weights or constraint bounds truly reflect the desired compromises among designs. However, there are a few disadvantages of this approach [4]: i) it requires a lot of experience with similar problems to gain insights about the objectives so that appropriate bounds on the objectives (that are used as constraints) or weights associated with the objectives can be specified, ii) we obtain one solution for each single objective optimization problem without any information about trade-offs among objectives, iii) weighted sum strategy or ϵ -constraint strategy might result in a suboptimal solution, if the trade-offs among objectives result in non-convex and/or non-continuous behaviour in function space.

There is considerable outgoing research in the area of multi-objective optimization in an attempt to alleviate some of these issues [4]-[11]. Multi-objective evolutionary algorithms (MOEAs) have been one of the most popular methods to find multiple trade-off solutions in single simulation ([4], [6], [12]-[14]). Among all MOEAs, elitist non-dominated sorting genetic algorithm (NSGA-II, [12]) has been reported to perform well for a large variety of problems of varying complexities. We have implemented NSGA-II algorithm in LS-OPT to provide users with the capability to identify many trade-off solutions in a single run. We demonstrate the applicability of this algorithm on a full vehicle multi-disciplinary crash worthiness problem. The rest of the paper is organized as follows. We briefly discuss multi-objective optimization problems and prominent algorithms used to find potential Pareto optimal solutions in Section 2. Details of NSGA-II are furnished in Section 3. Next, we validate the implementation using a few analytical benchmark problems in Section 4 and solve a full vehicle crash problem to study trade-off among objectives in Section 5. Major findings are recapitulated in Section 6.

2 Multiobjective Optimization

While details of multiobjective optimization theory can be found in many references ([2], [4]-[6]), we briefly review the basics in this section. Unlike single objective optimization problems, multi-objective optimization problems (MOOPs) have many objectives. A standard multi-objective optimization problem with M objectives, N design variables, J inequality and K equality constraints is formulated as follows.

2.1 Problem Definition

$$\begin{aligned} \underset{x}{\text{Minimize}} \quad & f_m(\vec{X}), & m = 1, 2, \dots, M; \\ \text{subject to:} \quad & c_j(\vec{X}) \leq 0, & j = 1, 2, \dots, J; \\ & h_k(\vec{X}) = 0, & k = 1, 2, \dots, K; \\ & x_i^L \leq x_i \leq x_i^U, & i = 1, 2, \dots, N. \end{aligned}$$

constraints. The set of all such non-dominated solutions 'a' is called as non-dominated set A. Note that all Pareto optimal solutions are non-dominated solutions but vice-versa is not necessarily true. The function space representation of Pareto optimal solutions is called Pareto optimal front. Major difficulties in solving multiobjective optimization problems arise due to the presence of local Pareto optimal fronts, non-convexity and/or discontinuities in the Pareto front, and maintaining diversity in the Pareto optimal solutions set.

2.3 Solution Methods

In this section, we discuss three prominent techniques to solve multi-objective optimization problems and pros and cons of each technique ([4], [5]).

2.3.1 Weighted Sum Strategy

This is the most popular method to convert a MOOP into a single objective optimization problem. We specify weights (w_m) associated with each objective and optimize the weighted sum of objectives.

$$\begin{aligned} \text{Minimize}_{\vec{X}} \quad & F = \sum_{m=1}^M w_m f_m(\vec{X}) \\ \text{subject to:} \quad & c_j(\vec{X}) \leq 0, \quad j = 1, 2, \dots, J; \\ & h_k(\vec{X}) = 0, \quad k = 1, 2, \dots, K; \\ & x_i^L \leq x_i \leq x_i^U, \quad i = 1, 2, \dots, N. \end{aligned} \quad (2)$$

Weights reflect the preference structure of designer. The advantage of this method is its simplicity and ease of use. This approach works the best for problems with convex and continuous Pareto fronts. However, this method results in local optimal solutions when the Pareto front is non-convex and/or discontinuous. Also, the selection of appropriate weight vector can be challenging.

2.3.2 ϵ -Constraint Strategy

This method uses a single objective as the objective function and uses bounds on other objectives to redefine the optimization problem as follows:

$$\begin{aligned} \text{Minimize}_{\vec{X}} \quad & f_i(\vec{X}), \\ \text{subject to:} \quad & c_j(\vec{X}) \leq 0, \quad j = 1, 2, \dots, J; \\ & h_k(\vec{X}) = 0, \quad k = 1, 2, \dots, K; \\ & x_i^L \leq x_i \leq x_i^U, \quad i = 1, 2, \dots, N; \\ & f_l(\vec{X}) \leq \epsilon_l, \quad l = 1, 2, \dots, M; l \neq i. \end{aligned}$$

Some of the popular MOEAs are elitist non-dominated sorting genetic algorithm (NSGA-II), and strength Pareto evolutionary algorithm (SPEA-II). Different algorithms compete for computational requirements, convergence to the global Pareto front, and diversity on the Pareto front. The major disadvantage of evolutionary algorithms is the high computational cost.

3 Elitist Non-dominated Sorting Genetic Algorithm

This algorithm was originally developed by Dr Kalyanmoy Deb and his students [12]. To eliminate the Pareto drift problem often encountered by finite population size algorithms, Goel et al. [15] proposed to maintain an external archive while using NSGA-II. We use their NSGA-II implementation in LS-OPT. A stepwise search procedure is given as follows.

- Step 1: Initialize parent population of individuals, and an empty archive
- Step 2: Evaluate objectives and constraints for all individuals in parent population, rank the population, and compute crowding distance
- Step 3: Use non-domination criterion based selection operator to identify high fitness individuals
- Step 4: Apply crossover and mutation operators to create child population
- Step 5: Evaluate all individuals in child population
- Step 6: Combine parent and child populations, rank the combined population and compute crowding distance
- Step 7: Select the best individuals to fill the new parent population
- Step 8: Copy all rank one individuals to an external archive
- Step 9: Update archive to remove dominated and duplicate solutions
- Step 10: Repeat Step 3 with new parent population till convergence criterion is met

The final set of non-dominated solutions in the archive is reported as potential Pareto optimal set. The major cost of optimization is the requirement of large number of function evaluations in Steps 2 and 5. To cater to this requirement, we facilitate two methods of using NSGA-II.

3.1 Implementation Details

Traditional form of genetic algorithms use binary numbers (binary encoding) that is we convert all real numbers into corresponding binary representations. We also facilitate the direct use of real numbers, also known as real encoding, which is shown to result in better convergence. There are many instances of genetic operators – selection, crossover, and mutation – that drive the search. We provide a few well-known operators in our implementation. For selection, there are three options, stochastic universal sampling (SUS), roulette wheel selection (RLT), and tournament selection (TRN) with an arbitrary tournament size. For single objective optimization, the user can use elitism and preserve a few elites to improve the search. Our implementation offers the user a choice of single point crossover or uniform crossover operator for binary representation of the variables and simulated binary crossover (SBX) or blend crossover (BLX) for real variables. Simple mutation is used for both binary and real encoding of variables. We specify number of generations as the stopping criterion.

3.2 Metamodel Based Implementation

In this approach, we develop good quality metamodels and use these metamodels to evaluate designs during the search. While this method significantly reduces the computational cost of identifying Pareto optimal set, the accuracy of the resulting solution set depends on the quality of the metamodels. If we have high quality metamodels, the resulting Pareto front would be accurate otherwise the results might not be accurate.

In the LS-OPT framework, this implementation allows the user to adopt the successive response surface methodology (SRSM). As has been done so far, the user specifies weights associated with different objectives but selects the multi-objective GA (as shown in Figure 1). This results in a set of Pareto optimal solutions. Next, we (internally) compute a weighted sum of objectives for each Pareto optimal solution and select the design with the most optimal value. This design is used as the optimal location to refine the design space in the next generation following the LS-OPT SRSM methodology.

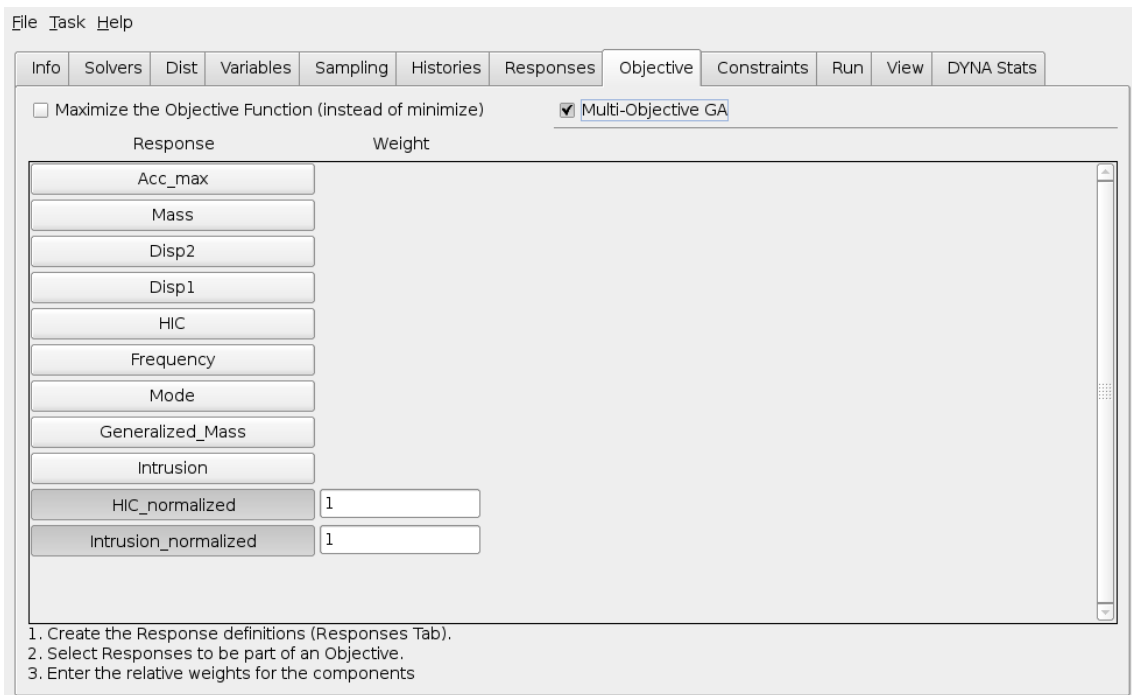


Fig. 1: Selecting the multi-objective GA option from the Objectives panel to use MOGA with metamodels.

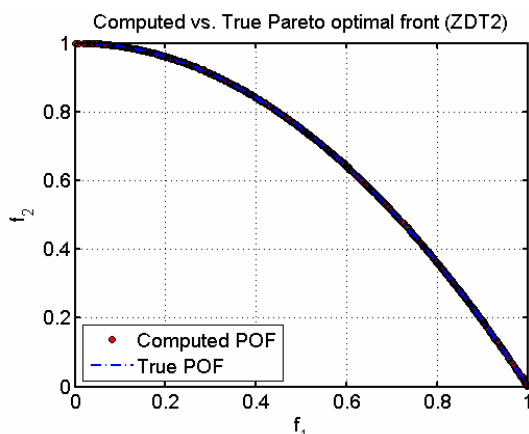
3.3 Direct Simulation Based Implementation

We also have enabled the capability to directly use simulations as function evaluators. This feature is very attractive when the function evaluation is relatively cheap or the user has enough resources to analyze a large number of designs. The major advantage is that the resulting non-dominated solution set is accurate.

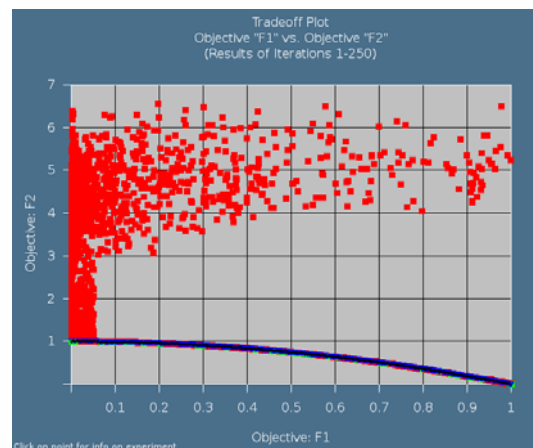
4 Analytical Benchmark Examples

We validate our implementation using a few benchmark analytical examples [4] and compare the non-dominated solution set obtained from NSGA-II with the known true Pareto optimal front. For all examples in this paper, we have used direct simulation based implementation. The details of the problem specific GA parameters are furnished in Appendix (Table 1).

4.1 ZDT2: Non-convex Pareto front



(A) Compare true and computed POF
Fig. 2: Pareto optimal front for ZDT2



B) All solutions and Pareto optimal solutions

$$\begin{aligned}
 & \underset{\vec{X}}{\text{Minimize}} \quad f_1(\vec{X}), f_2(\vec{X}) \\
 & f_1(\vec{X}) = x_1; \quad f_2(\vec{X}) = g(\vec{X})h(g(\vec{X}), f_1(\vec{X})), \\
 & g(\vec{X}) = 1 + 9(N-1)^{-1} \sum_{i=2}^N x_i; \quad h(\vec{X}) = 1 - (f_1/g)^2; x_i \in [0,1].
 \end{aligned} \tag{7}$$

This is a $N = 30$ variable unconstrained optimization problem. The true Pareto optimal front¹ is found at $x_1^* = [0, 1]$, $x_i^* = 0$ for $i=2,3,\dots,N$. Correspondingly, all solutions in Pareto optimal set have $g^* = 1$, and the function $f_1 = x_1^*$, $f_2 = 1 - x_1^{*2}$. This is an example of non-convex Pareto front that would pose difficulties in finding the solutions using the weighted sum strategy.

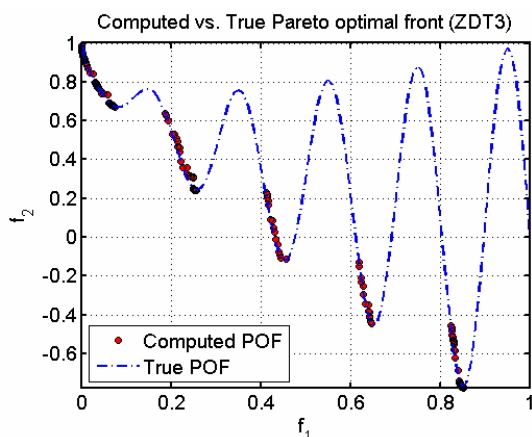
We compare the Pareto optimal front obtained from NSGA-II and the true Pareto front in Figure 2(A). The success of the NSGA-II algorithm is obvious. Figure 2(B) shows all the points evaluated during the course of optimization. Results show that NSGA-II algorithm tunes the search to find the non-convex Pareto optimal front.

4.2 ZDT3: Non-continuous Pareto front

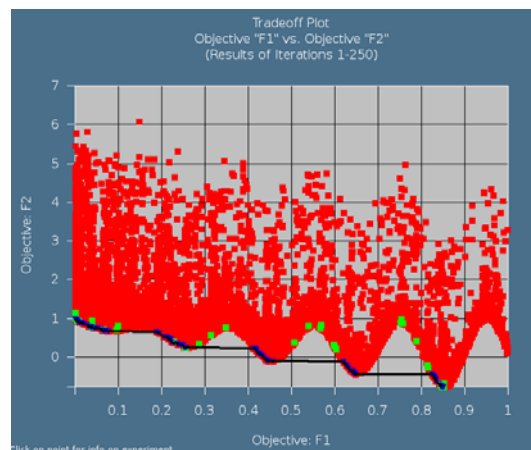
$$\begin{aligned}
 & \underset{\vec{X}}{\text{Minimize}} \quad f_1(\vec{X}), f_2(\vec{X}) \\
 & f_1(\vec{X}) = x_1; \quad f_2(\vec{X}) = g(\vec{X})h(g(\vec{X}), f_1(\vec{X})), \\
 & g(\vec{X}) = 1 + 9(N-1)^{-1} \sum_{i=2}^N x_i; \quad h(\vec{X}) = 1 - \sqrt{f_1/g} - (f_1/g) \sin(10\pi f_1).
 \end{aligned} \tag{8}$$

This is also a $N = 30$ variable unconstrained optimization problem. As before, the true Pareto optimal front is found in $x_1^* = [0, 1]$, $x_i^* = 0$ for $i=2,3,\dots,N$. Correspondingly, all solutions in Pareto optimal set have $g^* = 1$, and the function $f_1 = x_1^*$, $f_2 = 1 - \sqrt{x_1^*} - (x_1^*) \sin(10\pi x_1^*)$. This is an example of discontinuous Pareto optimal front with five distinct Pareto optimal regions. This problem would pose difficulties for ϵ -constraint strategy.

Comparison of the computed and true Pareto optimal front shown in Figure 3(A) clearly shows the ability of NSGA-II to find non-continuous Pareto front. Location of all evaluated points shown in Figure 3(B) depicts the exploration ability of the NSGA-II algorithm. We note that some points in the final non-domination set were not exactly on the true Pareto front. This result is obtained due to the use of a very greedy selection operator (roulette wheel selection) that reduces diversity among the individuals and compromises the exploration capabilities to some extent.



A) Comparison of true and computed POF
 Fig. 3: Pareto optimal front for ZDT3 problem.



B) All solutions and Pareto optimal set

¹ '*' represents global optimal solution.

4.3 BNH: Constrained Multi-objective Optimization Problem

$$\text{Minimize}_{\vec{X}} \quad f_1(\vec{X}), f_2(\vec{X})$$

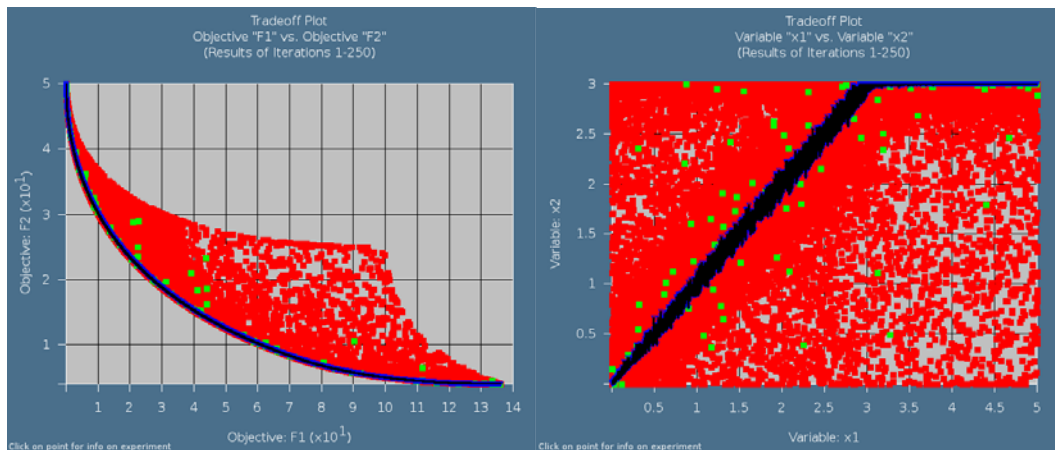
$$f_1(\vec{X}) = 4x_1^2 + 4x_2^2; \quad f_2(\vec{X}) = (x_1 - 5)^2 + (x_2 - 5)^2,$$

$$\text{subject to: } C_1(\vec{X}) = (x_1 - 5)^2 + (x_2 - 5)^2 - 25 \leq 0, \quad (9)$$

$$C_2(\vec{X}) = 7.7 - (x_1 - 8)^2 - (x_2 + 3)^2 \leq 0,$$

$$0 \leq x_1 \leq 5, \quad 0 \leq x_2 \leq 3.$$

This two variables constrained multi-objective optimization problem has two connected regions in Pareto optimal front. First region constitutes $x_1^* = x_2^* \in [0,3]$ and second region lies on $x_1^* \in [3,5], x_2^* = 3$.



A) POF and all points in function space B) POF and all points in variable space

Fig. 4: Results obtained for BNH problem. Red dots show all points evaluated during the course of optimization and blue points show the final non-dominated set.

As was observed for the unconstrained examples, NSGA-II easily finds the Pareto-optimal front for this constrained optimization problem (Figure 4(A)). The designs in variable space (Figure 4(B)) clearly show that the obtained results are indeed true Pareto optimal solutions.

The results shown here are similar to those reported in the literature and validate our implementation of NSGA-II algorithm. Next, we solve a large scale engineering optimization problem using NSGA-II algorithm with direct simulation.

5 Small Car Multidisciplinary Crash Optimization

Finally, the crash worthiness test of a simplified car finite element model is considered. A simplified car moving at a constant velocity of 35 mph (15.6 m/s) impacts a rigid pole as shown in Figure 5. The objective of the design is to simultaneously minimize the head injury criterion (HIC) over a 15ms interval of a selected point and intrusion of the pole into the vehicle. The car is also subjected to a constraint on the torsional mode vibration frequency. LS-DYNA [16] is used to perform crash (explicit solver) and modal analysis simulations (implicit solver). Four design variables that represent thicknesses of the parts influenced by the crash worthiness and noise, vibration, harshness (NVH) tests are bumper (t_{bumper}), roof (t_{roof}), front rail (t_{railf}), and back rail (t_{railb}).

This constrained multi-objective optimization problem can be summarized as follows.

Minimize $HIC, Intrusion$

subject to ;

$$1.7\text{Hz} \leq \text{Torsional mode frequency} \leq 1.9, \tag{10}$$

$$1.0 \leq t_{railb}, t_{railf}, t_{bumper}, t_{roof} \leq 5.0.$$

The HIC and intrusion values are normalized by respective target values (900 units for HIC, and 500 mm for intrusion) used by the automotive industry as a brain injury criterion.

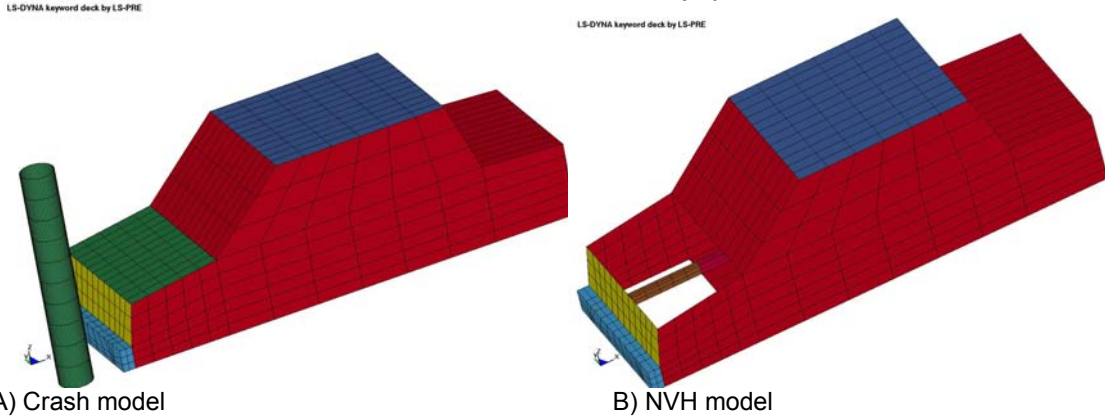


Fig. 5: Finite element models of a simplified car crash example.

5.1 Results and Discussion

We use multi-objective GA to solve this multi-disciplinary optimization problem. The GA parameter settings for this example are given in Table 1. The complete simulation took nearly 23 hours on a two-processor machine.

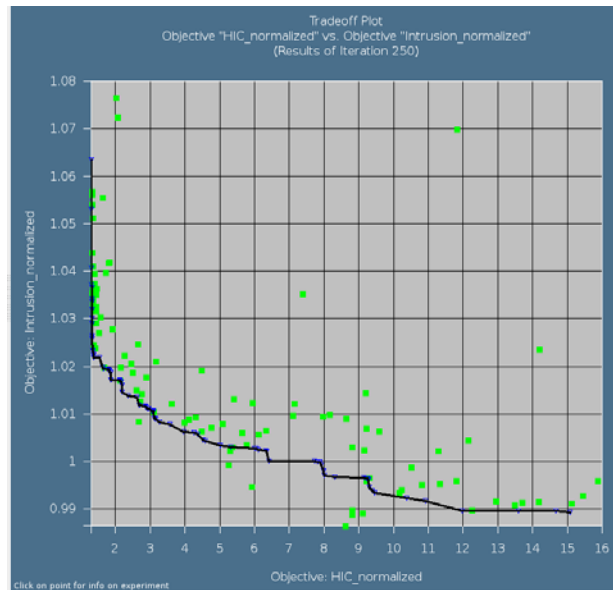


Fig. 6: Trade-off between normalized HIC and normalized intrusion.

At the end of simulation, there were 183 non-dominated solutions. The non-dominated front (candidate Pareto optimal front) from these different trade-off designs is shown in Figure 6. A small increase in intrusion values (0.99 – 1.01) may lead to a significant reduction in HIC values (15 – 3). Similarly, the vertical portion of the Pareto curve indicates a reduction in the intrusion for a small increase in the HIC value. The non-smooth nature of the resulting candidate curve reflects noise in the Pareto front, that may be either due to the nature of the problem, or insufficient convergence to the Pareto front.

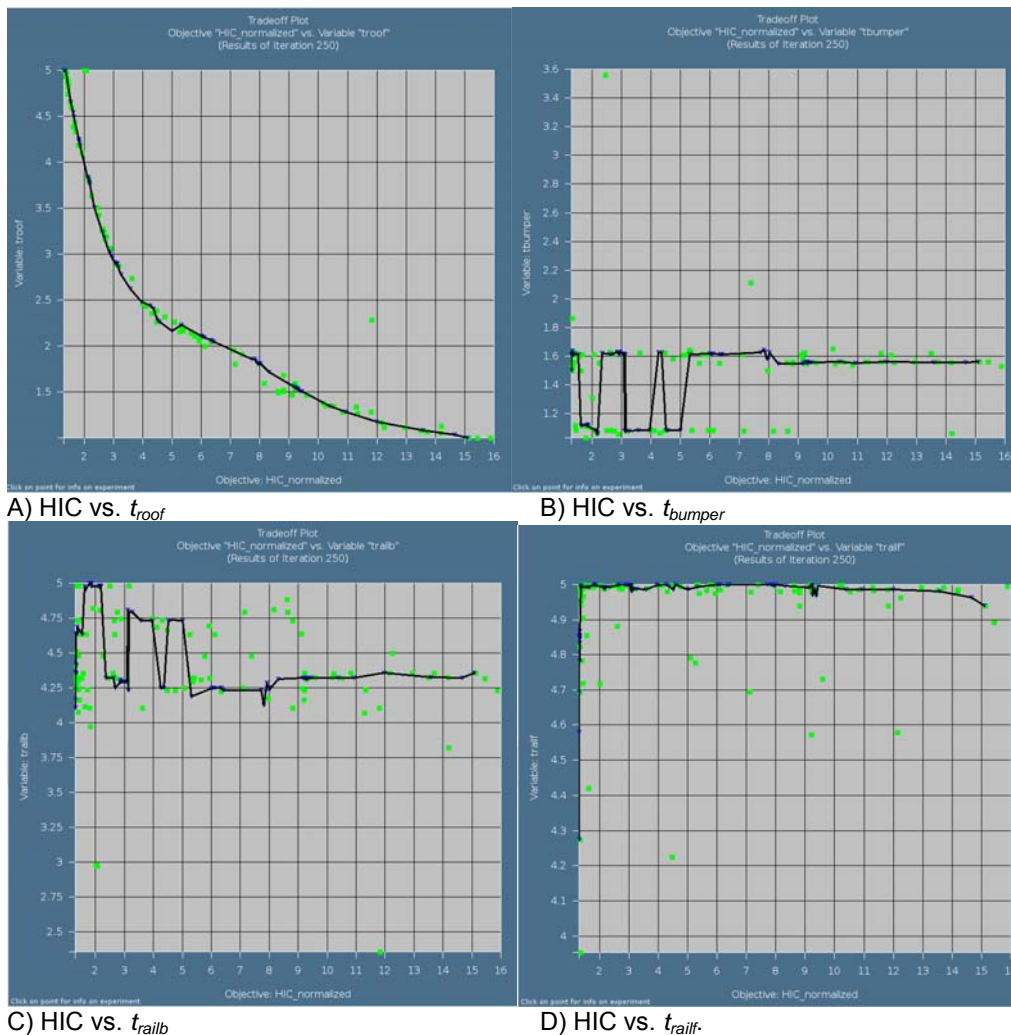


Fig. 7: Variable values on the trade-off curve.

To get more insight in the results, design variables corresponding to the candidate Pareto optimal designs is shown in Figure 7. It was observed that a reduction in the roof thickness increases the HIC value (Figure 7(A)). The thicknesses of bumper and back rails have to be largely fixed though some variations were observed (Figure 7(A-B)). Notably, the bumper thickness has to be low and the back rail thickness has to be high. The thickness of the front rail has to be maximized and it has no influence on the HIC criterion (Figure 7(D)). Such information about design variables offers insight in the results and the nature of the problem.

6 Summary

To cater to multi-objective optimization problems, an elitist non-dominated sorting genetic algorithm is implemented in LS-OPT. The code is validated by testing a suite of analytical benchmark problems. For each analytical example, the complete true Pareto optimal front was identified very well. Next, a simplified car crash problem with two objectives – HIC and intrusion – and constraint on the torsional mode vibration frequency is optimized. The resulting candidate Pareto optimal front revealed the importance of multi-objective optimization as a small increase in one objective allowed significant reduction in the other objective. Important information about the variables on the Pareto optimal front was also revealed.

7 References

- [1] Rao, SS, "Engineering Optimization", 1996, John Wiley.
- [2] Chankong, V, Haimes, Y, "Multiobjective Decision Making: Theory and Methodology", 1983, North Holland.
- [3] Livermore Software Technology Corporation, "LS-OPT Manual Version 3.2", 2006, Livermore CA.
- [4] Deb, K, "Multi-Objective Optimization Using Evolutionary Algorithms", Wiley, 2001, New York.
- [5] Miettinen, K, "Nonlinear Multi-Objective Optimization", Kluwer, 1998, Boston.
- [6] Coello, CAC, VanVeldhuizen, DA, Lamont, GB, "Evolutionary Algorithms for Solving Multi-Objective Problems", Kluwer, 2002, New York.
- [7] Marler, RT, Arora, JS, "Survey of Multi-objective Optimization Methods for Engineering", Structural and Multidisciplinary Optimization, 26(6), 2004, pp. 369-395.
- [8] Messac, A., Ismail-Yahaya, A, "Multiobjective Robust Design using Physical Programming," Structural and Multidisciplinary Optimization, 23(5), 2002, pp. 357-371.
- [9] Messac, A, Mattson, CA, "Normal Constraint Method with Guarantee of Even Representation of Complete Pareto Frontier", AIAA Journal, 42(10), 2004, pp. 2101-2111.
- [10] Kim, IY, de Weck, O, "Adaptive weighted sum method for multiobjective optimization", Structural and Multidisciplinary Optimization, 31(2), 2006, 105-116.
- [11] Martínez, M, Sanchis, J, Blasco, X, "Global and well-distributed Pareto frontier by modified normalized normal constraint methods for bicriterion problems", Structural and Multidisciplinary Optimization, 34(3), 2007, pp. 197-209.
- [12] Deb, K, Pratap, A, Agarwal, S, Meyarivan, T, "A fast and elitist multi-objective genetic algorithm: NSGA-II", IEEE Transaction on Evolutionary Computation, 6(2), 2002, pp. 181-197.
- [13] Zitzler, E, Laumanns, M, Thiele, L, "SPEA2: Improving the Strength Pareto Evolutionary Algorithm for Multiobjective Optimization", In K.C. Giannakoglou et al., editors, Evolutionary Methods for Design, Optimisation and Control with Application to Industrial Problems (EUROGEN 2001), 2002, pp. 95-100.
- [14] EMOO webpage: "<http://www.lania.mx/~ccoello/EMOO/>" last accessed on August 24 2007.
- [15] Goel, T, Vaidyanathan, R, Haftka, RT, Shyy W, Queipo, NV, Tucker, PK, "Response Surface Approximation of Pareto Optimal Front in Multi-objective Optimization", Computer Methods in Applied Mechanics and Engineering, 196(4), 2007, pp. 879-893.
- [16] Livermore Software Technology Corporation, "LS-DYNA Manual Version 971", 2007, Livermore CA.

8 Appendix A: NSGA-II Parameters

Table 1: NSGA-II parameters for different problems. The selected parameters are not necessarily the optimal parameters to solve any problem but are used to demonstrate different available features in LS-OPT implementation.

	ZDT2	ZDT3	BNH	Car-MDO
Population size	100	100	100	100
# of generations	250	250	250	250
Selection operator	SUS	RLT	TRN-2	TRN-2
Encoding type	REAL	MIXED	REAL	MIXED
Chromosome size		183		35
Real x-over type	BLX- α	BLX- α	SBX	SBX
Real x-over probability	0.95	0.95	0.99	1.00
Parameter	$\alpha=0.5$	$\alpha=0.5$	DI=5	DI=5
Binary crossover type		Uniform		SPT
Binary x-over probability		1.00		1.00
Real mutation type	Polynomial	Polynomial	Polynomial	Polynomial
Real mutation probability	0.07	0.059	0.5	0.5
Distribution index	20	20	20	5
Binary mutation probability		0.0055		0.0286