Assessing the Convergence Properties of NSGA-II for Direct Crashworthiness Optimization

Guangye Li[1], Tushar Goel[2], Nielen Stander[2]
[1]*IBM Deep Computing Group, Houston TX*
[2]*Livermore Software Technology Corporation, Livermore CA*

Abstract

The elitist non-dominated sorting genetic algorithm (NSGA-II) converges to the Pareto optimal front (POF) if a sufficient number of function evaluations are allowed. However, for expensive problems involving crash simulations, only a limited number of simulations might be affordable. It is observed that initially there are significant advances towards the POF but as the population matures, the improvements are relatively small. This means that one can probably limit the computational expense by terminating the search at the right point. The paper also demonstrates a successful use of IBM cluster for parallel processing that significantly reduces clock time for optimization.

Introduction

Genetic algorithms (GAs) are efficient to solve multi-objective optimization problems because they result in a diverse set of trade-off solutions. However, the GA requires a large number of simulations to converge to the global Pareto optimal front. This is a potential drawback for engineering problems like crashworthiness simulations that use long simulation times. One way of reducing the time for optimization is to exploit the parallelization capability of genetic algorithms in LS-OPT [1] and crash analysis using LS-DYNA [2]. Each crash analysis can be carried out using MPP version of LS-DYNA that utilizes many processors. This would reduce the time to analyze each design. Secondly, all designs in a GA generation can be simultaneously analyzed thus significantly reducing the clock time to perform optimization. This two level parallelization would have a multiplicative effect on the time-savings.

Another important factor in using GA for optimization is to assess the convergence properties of the algorithm for a given problem. It is well known that for single-objective optimization, there is a considerable improvement in the quality of solutions in the initial phases of the GA simulation. Incremental improvements are observed later on. One can terminate the GA search once a reasonable improvement is obtained. Thus saving a significant computational expense required to converge to the global optimal solution. While intuitively the same may be applicable for multi-objective optimization there has been no study to yet quantify the minimum computational effort.

In this paper, the above two issues are explored in context of multi-objective optimization using a GA. Firstly, the reduction in clock time is demonstrated by using a parallel architecture of processors using the IBM Poughkeepsie Benchmark Center’s x3455 cluster. Each node of this cluster has two Dual-Core AMD Opteron(tm) 2220 SE processors with the clock rate of 2.8 GHz. This cluster has totally 40 nodes (160 cores). Secondly, the convergence properties of the elitist non-dominated sorting algorithm (NSGA-II) [3] that is implemented in LS-OPT [4] are studied. A multi-disciplinary crashworthiness optimization of a full NHTSA vehicle is used for this study.

The paper is organized as follows. The multidisciplinary crashworthiness optimization problem is described in the next section. The test procedure adopted to study the issues is detailed in the Section **Test Procedure**. Information about the parallel architecture and other relevant details of running simulations on IBM clusters are delineated in the Section **Parallelization Details**. The main results of this study are shown in the Section **Results**, and major findings are summarized in the **Summary** Section.

Problem Formulation for Crashworthiness Optimization

The performance of a National Highway Transportation and Safety Association (NHTSA) vehicle is optimized for crashworthiness. The full frontal impact crash is simulated using a finite element model containing approximately 30,000 elements, shown in Figure 1. A modal analysis is performed on a so-called ‘body-in-white’ model that has approximately 18,000 elements. The vibration model is depicted in Figure 2 in the first torsional vibration mode. The tracking methodology applied to the torsional mode is described in Reference [4]. The design variables represent gauges of structural components in the engine compartment of the vehicle (Figure 3). Seven gauge variables namely apron, rail-inner, rail-outer, shotgun-inner, shotgun-outer, cradle rail and cradle cross member are selected to optimize the performance. Twelve parts comprising aprons, rails, shotguns, cradle rails and the cradle cross member were affected by selected design variables. LS-DYNA [2] is used for both the crash and modal analysis simulations, in explicit and implicit analysis modes respectively.

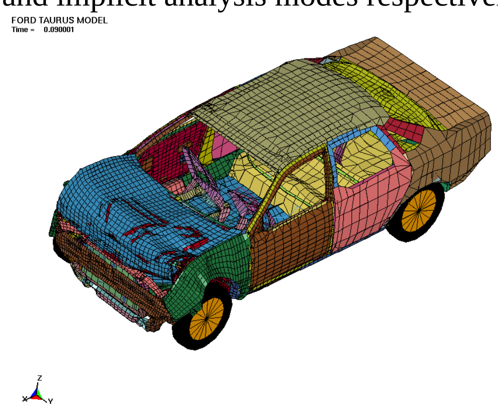

Fig. 1: Finite element crash model.

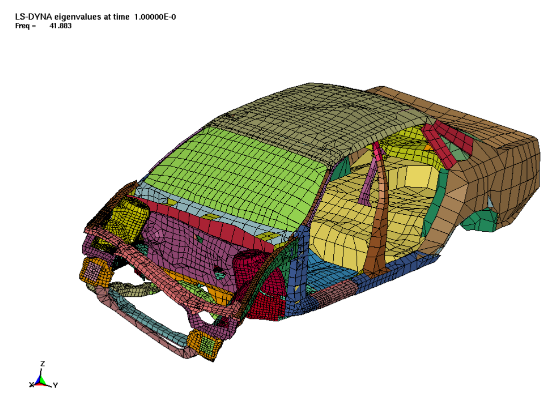

Fig. 2: Finite element vibration model.

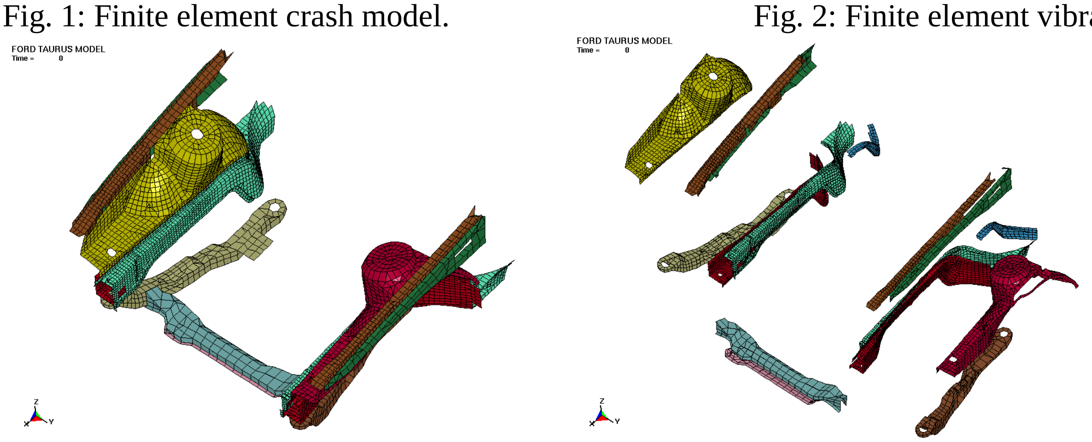

Fig. 3: Thickness design variables (with exploded view).

The vehicle performance is characterized by the intrusion, stage pulses, mass, and torsional frequency. A multi-disciplinary, multi-objective optimization problem is formulated as follows:

Minimize

Mass and Intrusion

Subject to:

Table 1 – Design constraints.

	Lower bound	Upper bound
Maximum intrusion (x_{crash})	-	551.27mm
Stage 1 pulse(x_{crash})	14.512 g	-
Stage 2 pulse(x_{crash})	17.586 g	-
Stage 3 pulse(x_{crash})	20.745 g	-
Torsional mode frequency(x_{NVH})	38.27Hz	39.27Hz

Table 2 – Starting values and bounds on design variables.

Variable name	Lower bound	Baseline design	Upper bound
Rail inner	1	2	3
Rail outer	1	1.5	3
Cradle rails	1	1.93	3
Aprons	1	1.3	2.5
Shotgun inner	1	1.3	2.5
Shotgun outer	1	1.3	2.5
Cradle cross member	1	1.93	3

The bounds on the design variables are given in Table 2 together with the starting baseline design.

The three stage pulses are calculated from the SAE filtered (60Hz) acceleration $\ddot{x}$ and displacement x of a left rear sill node in the following fashion:

$$\text{Stage } j \text{ pulse} = -\frac{k}{(d_2 - d_1)}\int_{d_1}^{d_2} \ddot{x}\,\mathrm{d}x;\, k = 0.5 \text{ for } j = 1, k = 1.0 \text{ otherwise;}$$

with the limits (d_1;d_2) = (0;184); (184;334); (334;Max(x)) for i = 1,2,3 respectively, all displacement units are given in mm and the minus sign converts acceleration to deceleration.

In summary, the optimization problem aims to minimize the mass and intrusion without compromising the crash and vibration criteria. The constraints are scaled using the target values to balance the violations of the different constraints. This scaling is important in cases where multiple constraints may be violated as in the current problem.

Test Procedure

Each crash simulation takes about 3,400 to 3,900 seconds on one core of the IBM x3455 machine and the modal analysis takes around 40 seconds. The optimization of this example is setup in LS-OPT. To identify the Pareto optimal front, a direct GA simulation that uses NSGA-II as the multi-objective optimizer [4] is used. A population of 80 individuals is evolved over 100 generations. For selection, a tournament selection operator with a tournament size of two is used. Real-coded SBX crossover and polynomial mutation operators with a distribution index of five are used for crossover and mutation. The crossover probability is taken as 0.99 and mutation is carried out with a probability of 0.15.

To compare the sets of non-dominated solutions[1] [3] from various generations, the number of solutions that are dominated in each set is computed using a weak non-domination criterion. The smaller the number of dominated solutions, the better is the convergence property. Secondly, the diversity is characterized using a uniformity measure [3] defined as,

$$\Delta = \sum_{i=1}^{N} \frac{\left| d_i - \bar{d} \right|}{N}; \bar{d} = \frac{1}{N} \sum_{i=1}^{N} d_i . \qquad (1)$$

where d_i is the crowding distance of the solution in function or variable space. The boundary points are assigned a crowding distance of twice the distance to the nearest neighbor. Two uniformity measures, one in the function space and another in the variable space are considered in this study. A small value of the uniformity measure is desired.

Parallelization Details

To handle a large number of simulations required for direct optimization, LS-OPT is run on an IBM cluster that is managed using the queuing system Tivoli Workload Scheduler LoadLeveler®. LoadLeveler is a parallel job scheduling system that allows users to run more jobs in less time by matching each job's processing needs and priority with the available resources, thereby maximizing resource utilization. Since the chosen population size is 80 in our experiment, at each generation, LS-OPT submits 80 crash jobs and 80 NVH jobs to LoadLeveler, and LoadLeveler arranges all jobs to optimize the use of the computing resource.

Results

The main results of this study are outlined in this section. A total of 7956 distinct designs were analyzed using LS-DYNA and 3823 designs were found feasible. The total number of designs is not 8000 because simulations for duplicate designs are not repeated. The savings by distributing function evaluations on a cluster using IBM clustering system were about 7,800 hours. That is, we do 80 simulations on 80 cores of the cluster concurrently, and the elapsed simulation time was reduced from about 7950 hours on a single core to 100 hours.

[1] A non-dominated solution is not dominated by any other solution in the set. A solution is considered to be dominated if it is not better than comparing solution in all objectives and is worse in at least one objective.

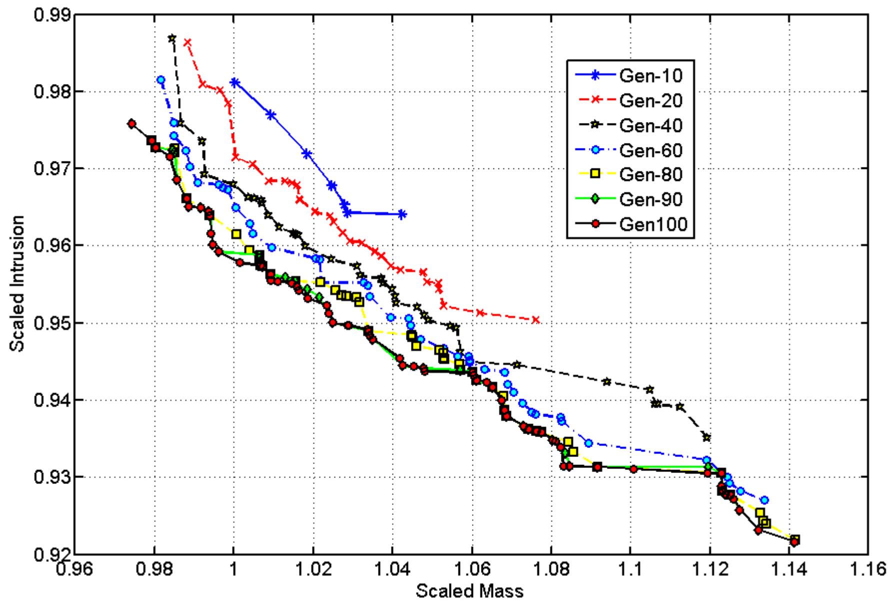

Fig. 4: Non-dominated solutions at different generations. Gen-10 corresponds to 10th generation.

The evolution of the trade-offs between objectives is shown in Figure 4 where the scaled mass and intrusion are plotted on the x- and y-axis respectively for different generations. As expected, there were significant improvements in the quality of trade-off solutions in the initial phase (notice the difference in trade-off curves obtained at 10th and 40th generations). Subsequently, smaller but still noteworthy improvements in trade-off solutions were observed until the 90th generation. The improvements between trade-off solutions obtained between the 90th and 100th generation were relatively small. Apparently, the trade-off solutions at the 100th generation were better distributed at the front than the solutions obtained at the 90th generation.

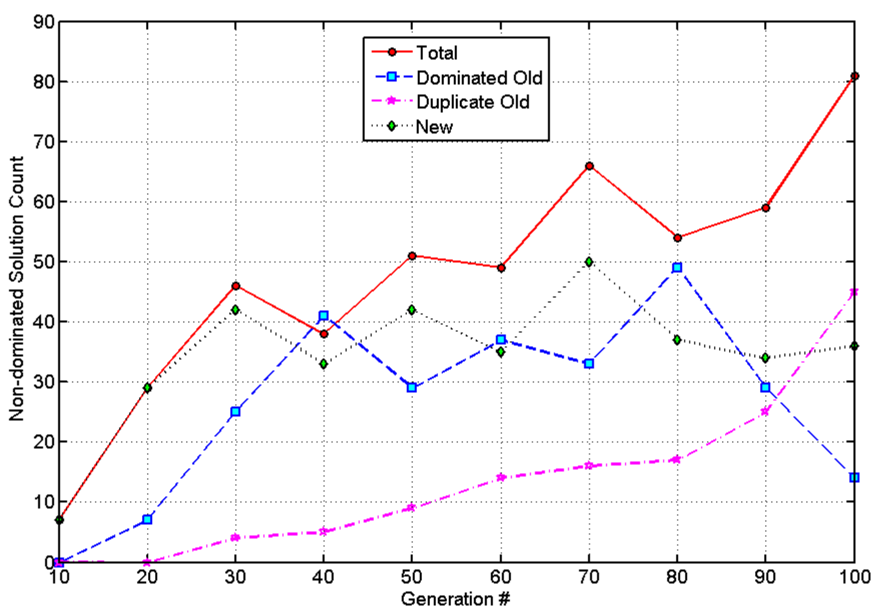

Fig. 5: Non-dominated solutions at different generations. Gen-10 corresponds to 10th generation.

The improvements in the quality of the trade-off solutions in ten generations are quantified by computing the total number of non-dominated solutions and the changes in the trade-off solution set over the last ten generations. The total number of non-dominated solutions (Total), the number of duplicate solutions (Duplicate Old) that were carried to next generations, the count of dominated solutions that were removed (Dominated Old), and the number of new non-dominated solutions that are added (New) are shown as a function of generation number in Figure 5. It is observed that the total number of non-dominated solutions increased with generations. The occasional decrease in the total non-dominated solutions (NDS) indicates evolution of a few designs that were better than many NDS found in earlier generations. The number of duplicate solutions continuously increased with the evolution of the population. A steep increase in duplicate solutions is noted after 80 generations. The number of dominated old NDS also decreased after 80 generations. This signifies the convergence of the designs to a (local[2]) Pareto optimal front (POF). At each generation, many new trade-off solutions evolved that characterized continuous improvement either in terms of convergence or diversity on trade-off curves. It is obvious from Figure 4 that many new solutions at the 100th generation contributed to improvement in diversity of the local POF.

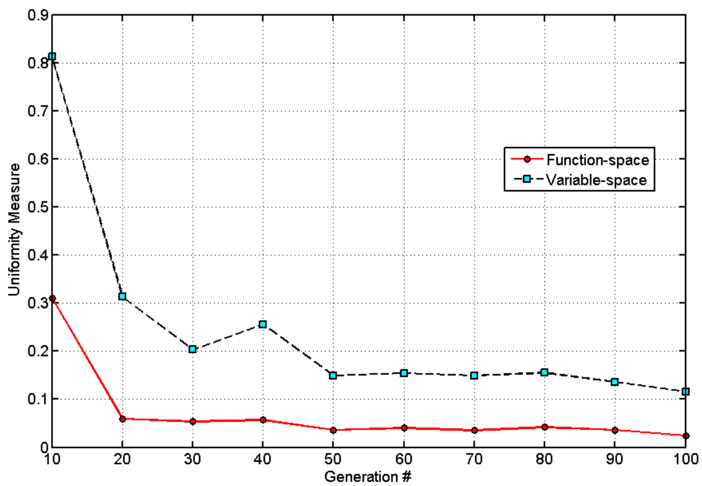

Fig. 6: Uniformity measures versus generation number.

The diversity in the POF/trade-off curves is quantified using the uniformity measure defined earlier. Figure 6 shows the uniformity measure in the function and variable spaces as a function of number of generations. It is noted that the solutions obtained have a reasonable diversity in both spaces and minor improvements in diversity are obtained over generations.

Finally, the improvements in the performance of a single design that corresponds to equal importance of all objectives are shown in Figure 7. This design is identified by allocating unit weight to all the objectives and minimizing the weighted sum of the objectives at each generation. A gradual improvement in the weighted sum of the objectives with generations is observed. The step plot is typical of a GA where no improvement in a design can be observed for a few generations. Corresponding values of individual objectives and design variables are plotted in Figure 8 and Figure 9, respectively. Figure 8 indicates that the best design (according to unit weight criterion) at any generation is not obtained by monotonous improvements in individual objectives. Instead, the trade-off between two objectives is reflected such that occasional increase in one objective is observed compared to the previous generation.

[2] For engineering problems, the global Pareto optimal front is unknown. It must be noted that a global Pareto optimal front is also a local Pareto optimal front but vice-versa is not true.

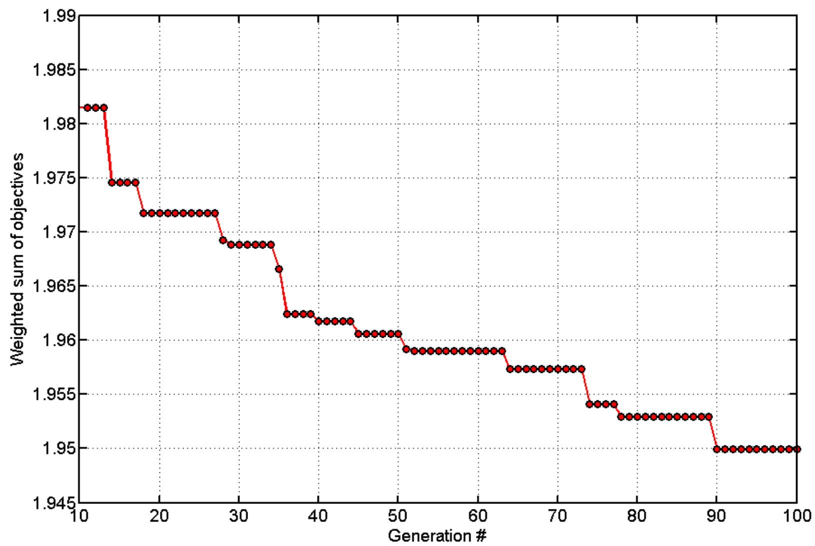

Fig. 7: Generation number vs. best equally weighted sum of objectives.

Figure 9 indicates that the thickness of cradle rail and inner shotgun were at the lower bounds, and the inner rail thickness was at the upper bound for the best weighted design with unit weights. Low values of outer shotgun, outer rails, and apron were preferred.

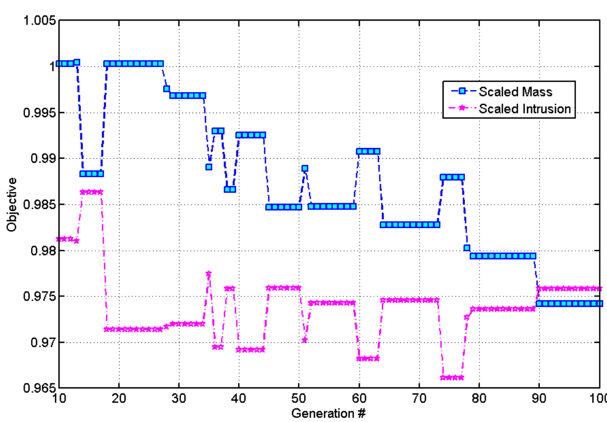

Fig. 8: Objectives vs. Generation number.

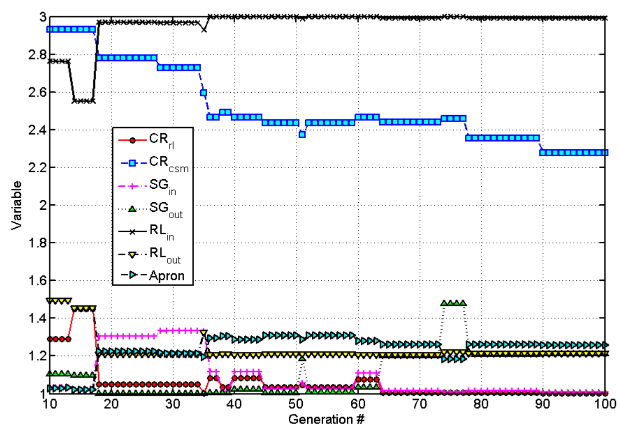

Fig. 9: Optimal design variables.

Summary

The present study demonstrated an application of a direct GA to multi-disciplinary crashworthiness optimization of a NHTSA vehicle. It was observed that allowing sufficient computational resources may help GA to converge to a local Pareto optimal front. The improvements in the quality of solutions diminish as the simulations converge and there may be a cut-off to terminate search to save computational resources. The relatively continuous nature of the local Pareto front suggests that different trade-offs exist but one has to judiciously select the design that suits the requirements. One such trade-off design with equal importance to all objectives was selected and gradual improvements with generations were observed. The trade-off nature of the design was reflected as individual objectives were not monotonically reduced during the course of the optimization. A look into the design variables corresponding to this trade-off design suggested that the inner rail and shotgun thicknesses need to be minimized and the cradle rail thickness has to be maximized. All thicknesses except the cradle cross member thickness are found to favor low values. Finally, a very important result was the effective and successful use of parallel hardware that significantly reduced the computational time.

Acknowledgement

Authors would like to thank the IBM Poughkeepsie Benchmark Center for providing computing equipment and for their strong technical support. Authors also thank David Bjorkevik for enormous help with parallel processing.

References

1. Stander, N., Roux, W.J., Goel, T., Eggleston T. and Craig, K.J. LS-OPT® Version 3.3 User's Manual, Livermore Software Technology Corporation, October 2007.
2. Hallquist, J.O. LS-DYNA® User's Manual, Version 971.
3. Deb, K, Multi-Objective Optimization Using Evolutionary Algorithms, Wiley, 2001, New York.
4. Goel T, Stander N, Multi-objective Optimization with LS-OPT, 6th German LS-DYNA Forum, Frankenthal Germany, Oct 11-12 2007.
5. Craig, K.J., Stander, N., Dooge, D.A,. Varadappa, S. Automotive crashworthiness design using response surface-based variable screening and optimization, Engineering Computations, Vol. 22:38-61, 2005.