8th World Congress on Structural and Multidisciplinary Optimization
June 1 - 5, 2009, Lisbon, Portugal

Optimal Resource Allocation for Genetic Algorithm Based Multi-Objective Optimization with 1000 Simulations

Tushar Goel[1], Nielen Stander[1], Yih-Yih Lin[2], Martin Liebscher[3]

[1]Livermore Software Technology Corporation, Livermore, CA, tushar, nielen@lstc.com
[2]Hewlett-Packard, High Performance Computing Division, Richardson TX, yih-yih.lin@hp.com,
[3]DYNAmore GmBH, Stuttgart, Germany, martin.liebscher@dynamore.de

1. Abstract
This study pertains to practical application of the GA for industrial applications where only a limited number of simulations can be afforded. Specifically, an attempt is made to find the optimal allocation of the total simulation budget (population size and number of generations) for constrained multi-objective optimization. A study is conducted to seek improvements while restricting the number of simulations to 1000. Parallelization is exploited using concurrent simulations for each GA generation on a HP quad-core cluster, and resulted in a significant time savings. Furthermore, the optimal distribution of computational effort to achieve the greatest improvement in performance was explored. Two analytical examples as well as an automotive crashworthiness simulation of a finite element model with 58,000 elements were used as test examples. Various population sizes and numbers of generations were tried while limiting the total number of simulations to 1000. The optimization performance was compared with Monte-Carlo and space filling sampling methods. It was observed that using the GA, many feasible and trade-off solutions could be found. It is shown that it is beneficial to allow a large number of generations to get good trade-off solutions. For the vehicle design, significant improvements in the performance were observed while this example also suggests that, for problems with a small feasible region, the number of feasible solutions can be significantly increased in the first few generations involving about 200 simulations.

2. Keywords: Multi-objective evolutionary algorithm, crashworthiness, population sensitivity, resource allocation, NSGA-II, optimization.

3. Introduction
The genetic algorithm (GA) is a global optimization method for single- and multi-objective optimization [1]. Since the GA is a population-based method, it typically requires a large number of simulations to find an optimal solution. With expensive function evaluations such as vehicle crash simulations, the time to achieve a converged solution can therefore be unreasonably long. The only way to solve this predicament is to process the computational load in parallel with a large number of computer processors. The total computation time can be reduced a) by running the expensive simulation on many processors, and b) by running many simulations in parallel. The latter approach is highly suitable for population based optimization methods such as the GA. The genetic algorithm implementation in LS-OPT® ([2], [3]) enables the user to run many simulations in parallel while using many processors for each simulation using the MPP (parallel processing) version of LS-DYNA® [4]. Though most industries have adopted clusters of CPUs, the availability of resources for optimization is still finite. Hence, the issue of optimization with a reasonable budget of simulations is of very high significance.

The performance of the GA might depend on the interplay between diversity and evolution characterized by population size and the number of generations, respectively. Obviously, a small population size would have a lack of diversity in the population and would be prone to convergence to a local optimal design (or local Pareto optimal front). On the other hand, a large population size would require significant computational effort for evolution. The influence of population size on performance has been studied in literature for single objective optimization [5]-[6] but not for multi-objective optimization. This issue is particularly important for engineering problems like vehicle crash simulations, which almost always have multiple objectives and a limit on the number of simulations due to the high cost of each simulation.

This interplay between diversity, evolution, and computational expense is the subject of this paper. Three examples that include an analytical example, engineering example and a crashworthiness example are studied. The computational expense for all examples is limited by fixing the number of simulations to 1000. While the analytical and engineering examples are computationally inexpensive, the crashworthiness example is computationally intensive. The elapsed wall time for the crashworthiness example is reduced by using a cluster of HP ProLiant servers with quad-core Xeon processors (courtesy HP) to parallelize the optimization process. All individuals in the population at any generation are evaluated in parallel. The significant reduction in the

optimization time enabled the study of the convergence properties of a multi-objective optimizer with variation in the population size and number of generations.

The next section provides details of the simulation strategy and test metrics adopted in this study. The example problems are described in the following section. Next, the results obtained are presented and analyzed. Finally, the main findings are recapitulated.

4. Test Methodology and Test Metrics

4.1. Test Methodology

An elitist non-dominated sorting genetic algorithm (NSGA-II) [1] is one of the most popular multi-objective evolutionary optimization methods. To study the impact of diversity and evolution on the performance of NSGA-II, various combinations of population size and number of generations were considered, as given in Table 1. All other parameters of the NSGA-II algorithm are kept constant, i.e., tournament selection operator with a tournament size of two, and real-coded crossover and mutation operators are used to create child population. The crossover and mutation probabilities are taken as 0.99 and the inverse of the number of variables, respectively. The constraint-domination strategy proposed by Deb [1] is employed to handle constraints. Furthermore, the performance of different NSGA-II cases was compared with the random search method. Two strategies, Monte-Carlo method and Space filling design [7] (implemented in LS-OPT [2]), were used to sample 1000 random points in the design space. A maximum of 1000 simulations were allowed for each simulation to keep the computational expense close to practical limits and to provide equivalence among all cases.

Table 1: Different configurations of population-size and number of generations. *Not a GA simulation

Case	Population size	# of generations
Monte-Carlo*	1000	1
SpaceFilling*	1000	1
P20x50	20	50
P40x25	40	25
P50x20	50	20
P100x10	100	10

4.2. Test Metrics

Unlike single objective optimization problems where the optimum design is a single solution, multi-objective optimization results in a set of optimal solutions. Thus, special metrics are needed to compare the results from different simulations. Typically, two criteria, convergence to the Pareto optimal front and diversity on the Pareto optimal front, are used to compare the multi-objective optimization results. To compare the two sets of weakly non-dominated solutions[1] (candidate Pareto optimal designs), the number of solutions that are dominated in each set by the solutions in the other set is computed using a weak non-domination criterion [1]. The smaller the number of dominated solutions, the better is the convergence to the Pareto optimal front.

Secondly, the diversity on the Pareto front is characterized by the spread of solutions and a uniformity measure [1], 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[2] of the solution in the function or variable space. As can be seen from Equation 1, the uniformity measure assesses how uniformly the points on the non-dominated front are spaced. A small value of this uniformity measure is desirable. Two uniformity measures, one in the function space and another in the variable space are considered in this study.

5. Test Examples

Three test problems, a mathematical function, an analytical engineering example, and a simulation based engineering problem, are used to study the interplay between evolution and diversity. The examples are described as follows:

---

[1] A solution **x** weakly dominates other solution **y** if **x** is not worse than **y** in any objective and **x** is strictly better than **y** for at least one objective. A solution **x** that is not dominated by any solution in a set of solutions is considered non-dominated with respect to that set.

[2] Crowding distance is defined as half the perimeter of the largest hypercube which can be allowed around a solution without including any other solution from the same non-dominated front. The crowding distance of the boundary points is taken as twice the regular crowding distance.

5.1. Analytical Example – OSY (Osyczka and Kundu [8])

This is a six variable analytical example, with two quadratic objective functions and six constraints.

$$\begin{aligned} f_1(\mathbf{x}) &= -\left[25(x_1-2)^2+(x_2-2)^2+(x_3-1)^2+(x_4-4)^2+(x_5-1)^2\right], \\ f_2(\mathbf{x}) &= x_1^2+x_2^2+x_3^2+x_4^2+x_5^2+x_6^2. \end{aligned} \tag{2}$$

Subject to:

$$\begin{aligned} &C_1(\mathbf{x}) \equiv x_1+x_2-2 \geq 0, \\ &C_2(\mathbf{x}) \equiv 6-x_1-x_2 \geq 0, \\ &C_3(\mathbf{x}) \equiv 2-x_2+x_1 \geq 0, \\ &C_4(\mathbf{x}) \equiv 2-x_1+3x_2 \geq 0, \\ &C_5(\mathbf{x}) \equiv 4-(x_3-3)^2-x_4 \geq 0, \\ &C_6(\mathbf{x}) \equiv (x_5-3)^2+x_6-4 \geq 0, \\ &0 \leq x_1, x_2, x_6 \leq 10, 1 \leq x_3, x_5 \leq 5, 0 \leq x_4 \leq 6. \end{aligned} \tag{3}$$

The global Pareto front for this example lies on the boundary of the constraints. There are five connected regions on the Pareto front that all satisfy the condition $x_4^* = x_6^* = 0.$

5.2. Analytical Engineering Example – WATER (Musselman and Talavage [9])

This engineering example describes planning for a storm drainage system in an urban area. The three design variables denote local detention storage (x_1), maximum treatment rate (x_2) and the maximum allowable overflow rate (x_3). The five objectives to be minimized are the drainage network cost (f_1), storage facility cost (f_2), treatment facility cost (f_3), expected flood damage cost (f_4), and expected economic loss due to flood (f_5). More details can be obtained from the original paper by Musselman and Talavage [9]. The description of the problem is as follows:

$$\begin{aligned} &f_1(\mathbf{x}) = 106780.37(x_2+x_3)+61704.67, \\ &f_2(\mathbf{x}) = 3000x_1, \\ &f_3(\mathbf{x}) = (305700)2289x_2/(0.06\times 2289)^{0.05}, \\ &f_4(\mathbf{x}) = (250)2289\exp(-39.75x_2+9.9x_3+2.74), \\ &f_5(\mathbf{x}) = 25\left[1.39/x_1x_2+4940x_3-80\right] \end{aligned} \tag{4}$$

Subject to:

$$\begin{aligned} &C_1(\mathbf{x}) \equiv 0.00139/(x_1x_2)+4.94x_3-0.08 \leq 1, \\ &C_2(\mathbf{x}) \equiv 0.000306/(x_1x_2)+1.082x_3-0.0986 \leq 1, \\ &C_3(\mathbf{x}) \equiv [12.307/(x_1x_2)+49408.24x_3+4051.02]/50000 \leq 1, \\ &C_4(\mathbf{x}) \equiv [2.098/(x_1x_2)+8046.33x_3-696.71]/16000 \leq 1, \\ &C_5(\mathbf{x}) \equiv [2.138/(x_1x_2)+7883.39x_3-705.04]/10000 \leq 1, \\ &C_6(\mathbf{x}) \equiv [0.417(x_1x_2)+1721.26x_3-136.54]/2000 \leq 1, \\ &C_7(\mathbf{x}) \equiv [0.164/(x_1x_2)+631.13x_3-54.48]/550 \leq 1. \\ &0.01 \leq x_1 \leq 0.45, 0.01 \leq x_2, x_2 \leq 0.1. \end{aligned} \tag{5}$$

All objectives in this example are scaled [f_1/8e4, f_2/1500, f_3/3e6, f_4/6e6, f_5/8e3] to avoid dimensionality issues.

5.3. Crashworthiness Example

The third example is a crashworthiness optimization problem that involves simulation of a National Highway Transportation and Safety Association (NHTSA) vehicle undergoing a full frontal impact. The finite element model for the full vehicle (obtained from NCAC website [10]), shown in Figure 1, has approximately 55K elements.

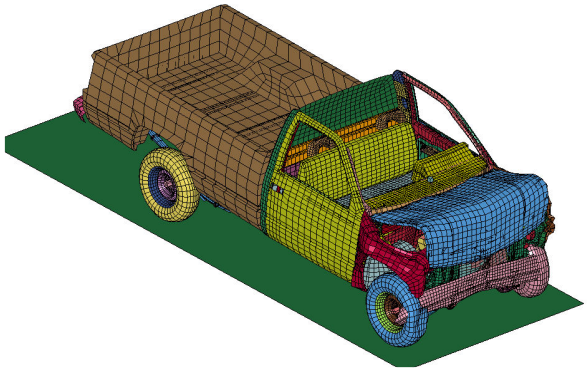

Figure 1: Finite element crash model of a pickup truck

Figure 2: Thickness design variables (with exploded view)

The gauges of structural components in the vehicle are parameterized directly in the solver input file. Nine gauge thicknesses associated with front-right-inner, front-right-outer, front-left-inner, front-left-outer, back-left and back-right rails, bumper, bottom-under radiator MTG, and bottom-center cabin member, are taken as design variables. The parts affected by the design variables are shown in Figure 2. The range of these design variables is chosen as within +/-20 % of the baseline design variable values. The baseline design and the bounds on the variables are given in Table 2.

Table 2: Baseline design and bounds on design variables

Variable description	Name	Lower bound	Baseline design	Upper bound
Rail front-right-inner	t1	2.500	3.137	3.765
Rail front-right-outer	t2	2.480	3.112	3.750
Rail front-left-inner	t3	2.400	2.997	3.600
Rail front-left-outer	t4	2.400	3.072	3.600
Rail right-back	t5	2.720	3.400	4.080
Rail left-back	t6	2.850	3.561	4.270
Bumper	t10	2.160	2.700	3.240
Radiator bottom	t64	1.000	1.262	1.510
Cabin bottom	t73	1.600	1.990	2.400

The crash performance of the vehicle is characterized by considering the maximum acceleration, maximum displacement that links to intrusion, time taken by the vehicle to reach zero velocity state, and different stage pulses. These responses are taken at the accelerometer mounted in the middle of the front seat. To reduce the influence of numerical noise, SAE filtered acceleration (filter frequency 60Hz) is used and different entities are averaged over two accelerometer nodes. While constraints are imposed on some of these crash performance criteria (stage pulses), it is desirable to optimize the performance with respect to other criteria. Thus a multi-objective optimization problem can be formulated as follows:

Minimize

Mass and peak acceleration;

Maximize

Time-to-zero-velocity and maximum displacement;

subject to constraints on variables and performance.

The design variable bounds are given in Table 2 and the performance constraints, namely maximum displacements and stage pulses, are specified in Table 3.

Table 3: Design constraints

	Upper bound
Maximum displacement ($\boldsymbol{x}_{\text{crash}}$)	721 mm
Stage 1 pulse(*SP1*)	7.48 *g*
Stage 2 pulse(*SP2*)	20.20 *g*
Stage 3 pulse(*SP3*)	24.50 *g*

The three stage pulses are calculated from the averaged SAE filtered (60Hz) acceleration $\ddot{x}$ and displacement x of the accelerometer nodes in the following fashion:

$$\text{Stage } j \text{ pulse} = -\frac{k}{(d_2 - d_1)}\int_{d_1}^{d_2} \ddot{x}\mathrm{d}x \quad k = 0.5 \text{ for } j = 1, k = 1.0 \text{ otherwise;} \tag{6}$$

The integration limits (d_1:d_2) = (0:200); (200:400); (400:Max(x_{crash})) for j = 1, 2, 3 respectively, represent different structural crash events. All displacement units are *mm* and the minus sign is used to convert acceleration to deceleration. During optimization, all objectives and constraints are scaled to avoid dimensionality issues.

The LS-DYNA [4] explicit solver is used to simulate the crash. Each crashworthiness simulation takes approximately 5 hours using one core of a fully-loaded quadcore Intel Xeon 5365 processor and generates an output of 225 MB. Obviously running 1000 simulations in serial would be very time-consuming. Fortunately, the genetic algorithms are very amenable to parallelization such that all individuals in a generation can be simultaneously analyzed. A 640-core HP XC cluster, comprising 80 ProLiant server nodes of two Intel Xeon 5365 quad-core processors (also known as Clovertown, with 2 processors/8 cores), with a 3.0 GHz clock rate and 8 GB memory, was used to run simulations[3]. More details about running the simulation appear elsewhere [11].

Table 4: Measured elapsed times for the six cases

Case	Population size	Number of generations	Elapsed time (h)	Ave. elapsed time per generation (h)
Monte-Carlo	1000	1	16	16
Space-Filling	1000	1	16	16
P20x50	20	50	334	6.68
P40x25	40	25	145	5.8
P50x20	50	20	120	6.0
P100x10	100	10	54	5.4

It is obvious from Table 4 that smaller population size resulted in higher wall time because only a single processor per simulation was used and hence many processors were left idle. However, the wall-time can be significantly reduced by running MPP version of LS-DYNA with multiple processors[4]. The variation in average time per generation reflects the waiting time due to sharing of the cluster.

5. Results and Discussion

The optimization results for all test examples are given in this section. It is noted that the global Pareto optimal front is not obtained with so few simulations instead the focus of this study is to assess the best resource allocation strategy to find the best non-dominated solution set. Table 5 summarizes the results for different cases for all examples.

5.1. Number of Simulations

While the total number of budgeted simulations was 1000 for each example and case, the actual number of simulations for NSGA-II was slightly lower because duplicate designs were simulated only once. This resulted in cost savings for computationally expensive problems like those encountered in crashworthiness simulations. The analytical engineering example (WATER) yielded the highest number of duplicate solutions for GA simulations.

[3] It is important to note that the cluster was shared by many users and each node was fully populated by the queuing system.

[4] Only SMP processor with a single-processor was used in this study to ensure complete equivalence of all simulations.

Table 5: Comparison of different cases for the three examples. Δ_{var}-uniformity measure in variable space, Δ_{obj}-uniformity measure in function space (smaller value is desired).

		# of simulations	# of feasible points	# of local non-dominated points	# of true non-dominated points	Δ_{var}	Δ_{obj}
OSY	Monte-Carlo	1000	18	6	0	0.89	26.26
	SpaceFilling	1000	22	3	0	4.93	99.35
	P20x50	994	787	7	7	0.47	2.34
	P40x25	999	721	32	10	0.23	0.57
	P50x20	994	617	23	17	0.50	1.40
	P100x10	995	509	19	1	0.57	4.45
WATER	Monte-Carlo	1000	919	230	86	0.013	0.024
	SpaceFilling	1000	921	228	87	0.011	0.020
	P20x50	958	857	422	386	0.011	0.014
	P40x25	964	896	475	373	0.009	0.009
	P50x20	977	905	449	323	0.008	0.009
	P100x10	962	904	449	259	0.009	0.009
CRASH	Monte-Carlo	1000	2	2	0	-	-
	SpaceFilling	1000	5	4	0	0.076	0.542
	P20x50	997	150	26	25	0.405	0.180
	P40x25	996	67	31	20	0.349	0.191
	P50x20	995	99	32	9	0.274	0.121
	P100x10	995	46	20	15	0.270	0.204

5.2. Number of Feasible Solutions

The random sampling techniques like Monte-Carlo method and space-filling methods provided an estimate of the size of the feasible region. It could be concluded from the results (Table 5) that the analytical example OSY and the crashworthiness example had very small feasible regions (2.5% and less than 0.5% of the entire design space, respectively), whereas the analytical engineering example was largely feasible (~92%).

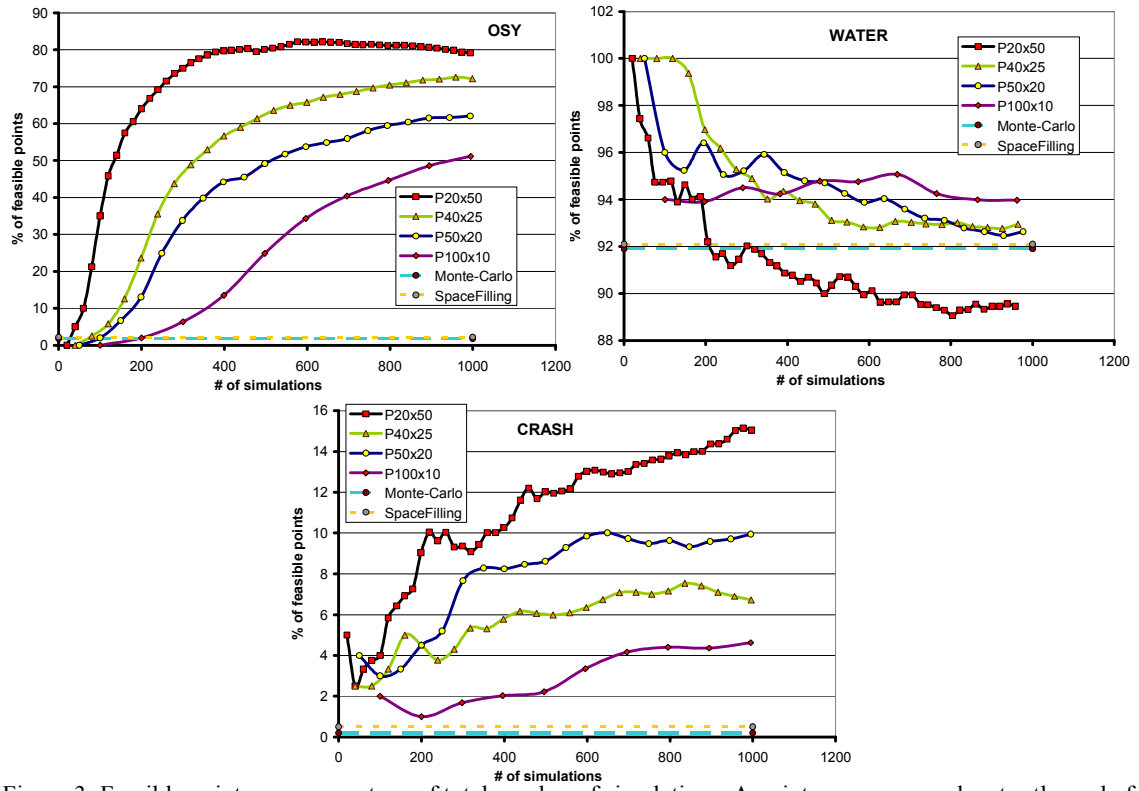

Figure 3: Feasible points as a percentage of total number of simulations. A point on any curve denotes the end of GA generation

A graphical view of the number of feasible points as a percentage of the total number of simulations is shown in Figure 3. As expected, the total number of feasible solutions obtained from the NSGA-II simulations was significantly higher compared to the random sampling method because the former focuses on finding feasible solutions. The proportion of feasible points generally increased because with evolution the GA search learned to avoid generating infeasible designs. The results of analytical and crash examples indicated that evolution (lower population size) was favorable for finding feasible designs, particularly for the problems with very small feasible regions. For the WATER example, the relatively poor performance of low population size was observed because the GA search was focused near the constraint boundary where optimal solutions are located and hence many infeasible solutions were found.

5.3. Convergence/Non-dominated Solutions

The number and quality of non-dominated solutions offered insights about the convergence of different optimization cases. Non-dominated solutions for each case were identified using a weak non-domination criterion on the respective set of solutions (~1000 simulations) for all example problems (Column 5, Table 5). The resulting non-dominated set was referred as local non-dominated solution set.

It was noted that the large number of feasible points did not guarantee a large number of local non-dominated points. While the random search strategies expectedly performed poorly, the NSGA-II simulation based on population sizes of 40 and 50 resulted in the highest number of locally non-dominated solutions (non-dominated with respect to corresponding set of 1000 simulations). This result suggested that there was a need to balance evolution and diversity in the GA search.

OSY	Monte-Carlo	Space-Filling	P20x50	P40x25	P50x20	P100x10
Monte-Carlo	0.00	0.00	0.00	0.00	0.00	0.00
Space-Filling	16.67	0.00	0.00	0.00	0.00	15.79
P20x50	66.67	33.33	0.00	18.75	13.04	31.58
P40x25	100.00	66.67	0.00	0.00	26.09	63.16
P50x20	100.00	100.00	0.00	50.00	0.00	63.16
P100x10	100.00	66.67	0.00	6.25	13.04	0.00
WATER	**Monte-Carlo**	**Space-Filling**	**P20x50**	**P40x25**	**P50x20**	**P100x10**
Monte-Carlo	0.00	21.93	1.42	2.74	8.24	7.57
Space-Filling	19.57	0.00	1.18	3.16	4.45	9.58
P20x50	34.35	33.33	0.00	11.16	13.14	19.60
P40x25	42.17	40.79	5.21	0.00	17.59	22.05
P50x20	37.83	34.65	4.50	8.00	0.00	17.82
P100x10	32.17	34.21	3.79	6.53	8.02	0.00
TRUCK	**Monte-Carlo**	**Space-Filling**	**P20x50**	**P40x25**	**P50x20**	**P100x10**
Monte-Carlo	0.00	0.00	0.00	0.00	0.00	0.00
Space-Filling	0.00	0.00	0.00	0.00	0.00	0.00
P20x50	100.00	50.00	0.00	22.58	37.50	25.00
P40x25	100.00	75.00	3.85	0.00	50.00	25.00
P50x20	100.00	50.00	0.00	6.45	0.00	25.00
P100x10	50.00	50.00	0.00	12.90	28.13	0.00

Figure 4: Percentage of local non-dominated points (case shown in column) that were dominated by the local non-dominated solution set from other strategies (shown in rows). Lower %age is desired.

Non-dominated solution sets from different cases were compared according to the weak non-domination criterion. Figure 4 shows the percentage of local non-dominated points that were dominated by the local non-dominated solution set identified using other cases. The space-filling sampling strategy was slightly better than the Monte-Carlo sampling because fewer local non-dominated solutions were dominated by local non-dominated solutions identified using other methods. Among all NSGA-II configurations tested, the lower population size cases were better than the higher population size NSGA-II cases. The case of 50 individuals evolved for 50 generations performed the best, and the case of 100 individuals evolved for 10 generations was the worst among all the NSGA-II cases. This result justified the benefits of evolution over diversity.

The relative quality of these local non-dominated solutions from different cases was determined by identifying the non-dominated solution set over entire set of simulation points (~6000 points) for each problem.

This set of non-dominated solutions was referred as the true non-dominated solution set[5]. Many local non-dominated solutions were found dominated by the solutions in the true non-dominated solution set. The number of points from the local non-dominated set that were included in the true non-dominated solution set is furnished in Table 5.

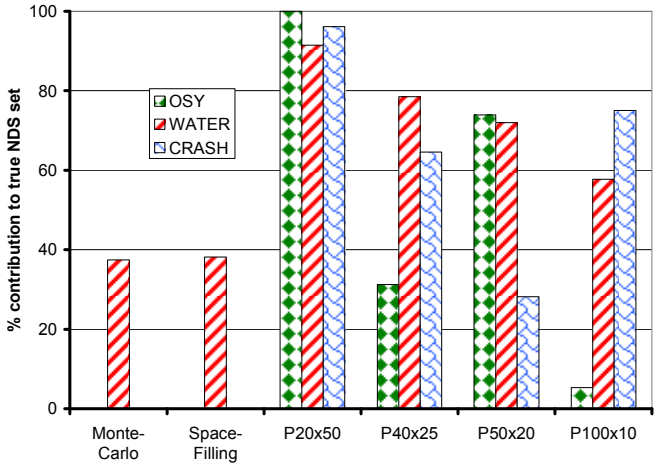

Figure 5: Percentage of local non-dominated solutions from different cases in the true non-dominated solution set (NDS) identified using all simulation points for each problem

The quality of local non-dominated solutions from different cases was quantified by computing the ratio of the number of local non-dominated points that were included in the true non-dominated solution set, to the total number of true non-dominated solutions. This ratio was plotted for all cases and examples in Figure 5. It is obvious that the best quality local non-dominated solutions are obtained when a population of 20 individuals is evolved for 50 generations. Lower population sizes are apparently more desirable than the large population size though the results for higher population sizes were dependent on the example problem.

Nevertheless, the results presented in this section clearly indicate that evolution is more important than diversity (high population size) in order to find good convergence properties for NSGA-II.

5.4. Diversity of Non-dominated Solutions

While the convergence in multi-objective optimization is important to reach the global Pareto front, the diversity of the Pareto front is also very important. The diversity reflects different trade-off solutions that might be interesting for a designer.

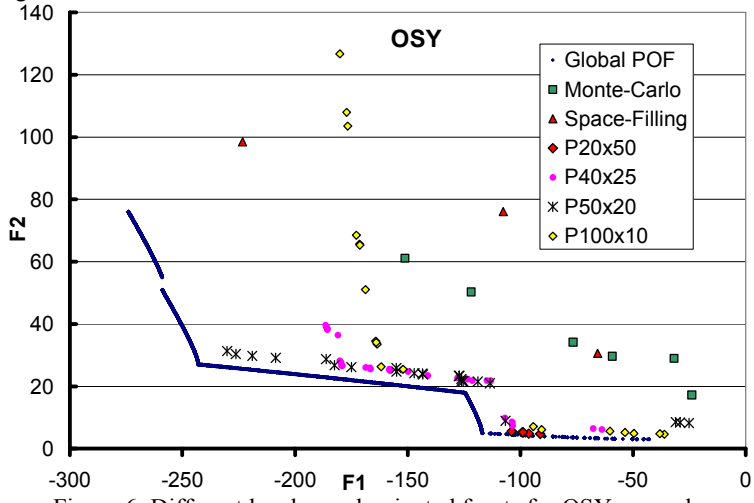

Figure 6: Different local non-dominated fronts for OSY example

The local non-dominated solutions obtained from different cases for the OSY example were visualized in the

[5] This is not the global Pareto optimal front. The term 'true' is used here to indicate that all simulations were used to identify non-dominated solutions.

function space (Figure 6). The global Pareto optimal front for this example was also plotted. Random sampling methods had very poor convergence but offered very good spread and uniformity in distribution. The results from different NSGA-II cases revealed that while population size 20 yielded the best convergence, the simulations points were largely focused in one region of the Pareto front (poor spread). This behavior is due to the fact that NSGA-II tries to converge to the global Pareto optimal front before spreading the points on the front. The local non-dominated solutions from the case with population size 50 resulted in the most reasonable convergence and spread. The lack of evolution was obvious for the case with population 100 as many local non-dominated solutions were found very far from the global Pareto front.

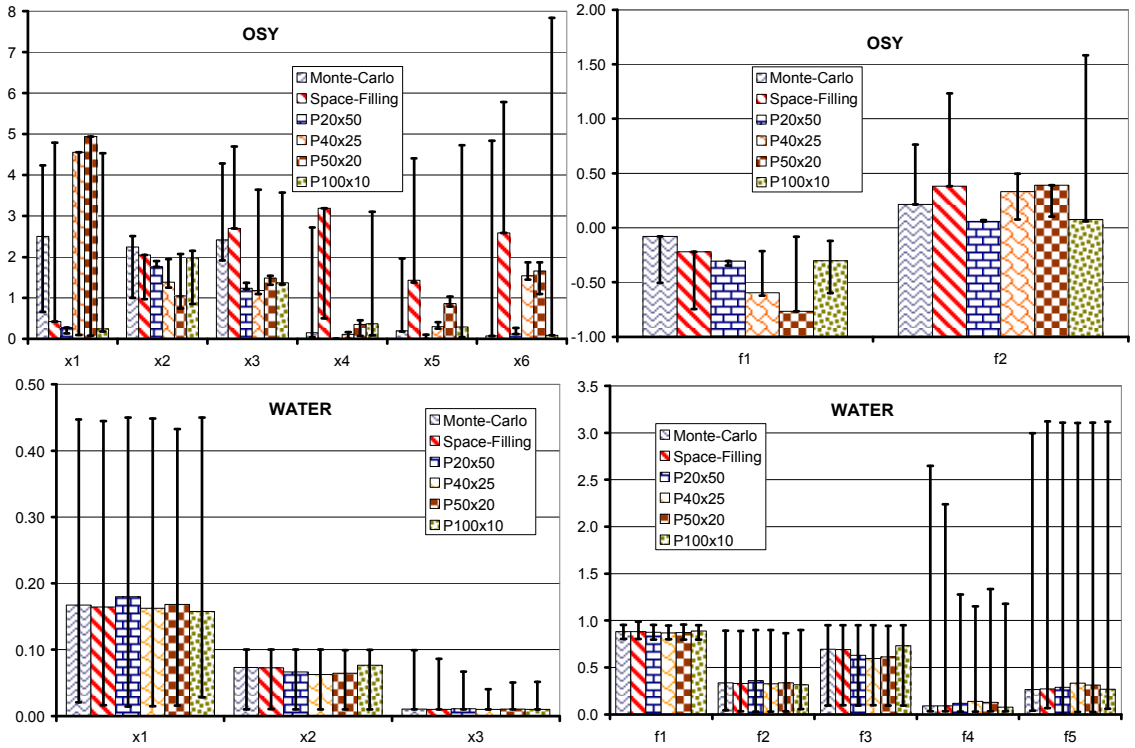

Figure 7: Best solution (sum of objectives) for each case and the range of local non-dominated solutions (shown by error bars) for different examples. Objectives f_1 and f_2 for the OSY problem are scaled by 300 and 80, respectively.

Similar graphical analysis is not possible for other examples and the variable space of OSY example due to high dimensionality. So the spread (ranges of all-objectives/variables) of local non-dominated solutions from different configurations are shown in both variable and objective function space in Figure 7 (analytical examples) and Figure 8 (crash example). The so-called "best" design is chosen by defining a weighted sum of all scaled objectives with equal importance to each objective function. The spread of non-dominated solutions in the function and variable space was quite comparable for all configurations on WATER example. This is perhaps due to a sufficient number of local non-dominated points for all cases. The spread of local non-dominated points for the CRASH example was comparable for all objectives except scaled acceleration which was also the most dominant objective function. The objective Mass (which is the total vehicle mass) did not vary much because only a part of the total mass was linked to the variables. The performance of low population size (particularly 20) was the most reasonable. There was ample diversity in the variable space also.

The uniformity of the distribution of the points on the local non-dominated front was reasonable for all examples and the same is reflected in Table 5. The lack of points on the non-dominated front for both random sampling techniques and NSGA-II case with population size 20 resulted in high values on the uniformity metrics due to dominant boundary effects. Relatively, moderate population-sized NSGA-II simulations resulted in more uniform distribution of non-dominated solutions for all examples.

5.5. Selection of the Best Available Design for the Crashworthiness Example

To compare the performance of a single design selected from the corresponding trade-off solution sets, a weighted sum of objectives was maximized, when unit weight was assigned to each objective function. The design variables, objectives, and constraints corresponding to the selected best designs for all cases are shown in Table 6 and Table 7, respectively. The performance of the baseline design was also provided for comparison.

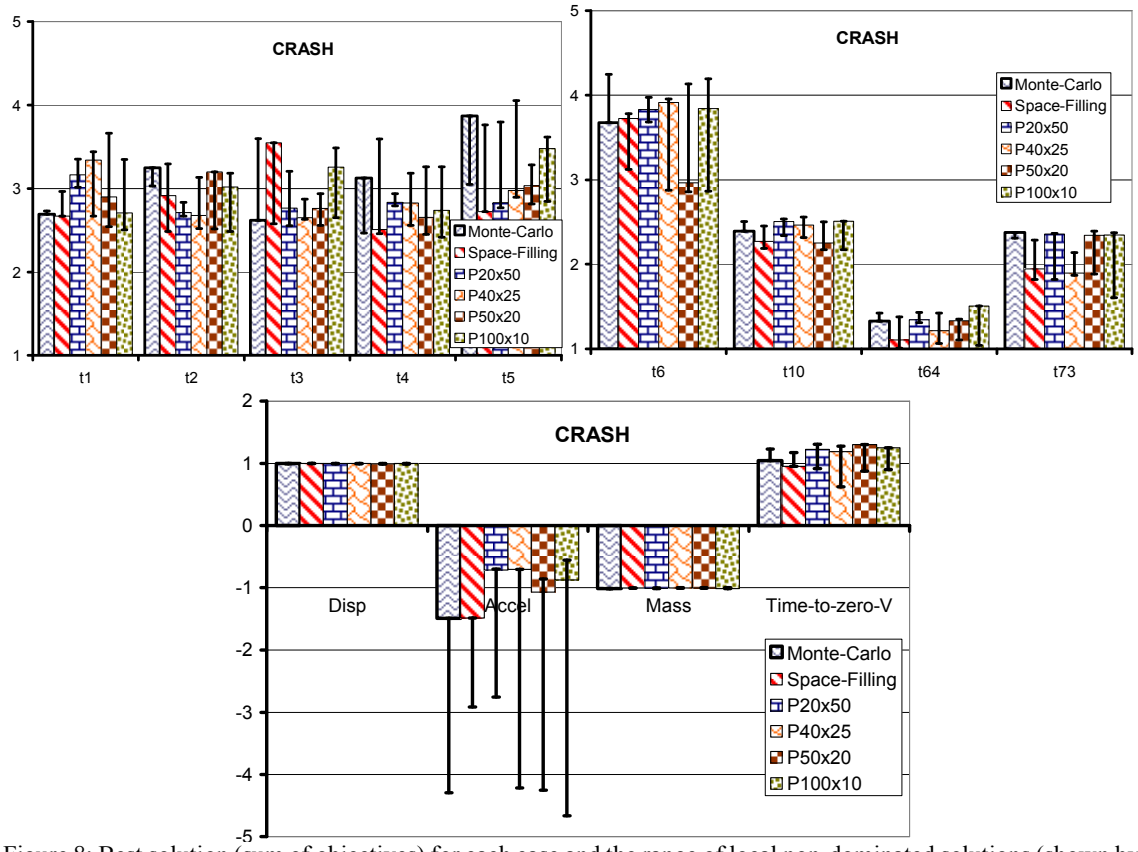

Figure 8: Best solution (sum of objectives) for each case and the range of local non-dominated solutions (shown by error bars) for crash examples. All objectives and the weighted sum are maximized.

Table 6: Design variables corresponding to the selected optimal design obtained using different configurations

Case	t1	t2	t3	t4	t5	t6	t10	t64	t73
Monte-Carlo	2.693	3.247	2.618	3.125	3.870	3.678	2.391	1.330	2.376
Space Filling	2.670	2.917	3.548	2.509	2.724	3.727	2.271	1.104	1.946
P20x50	3.165	2.710	2.767	2.834	2.829	3.829	2.505	1.345	2.357
P40x25	3.340	2.679	2.652	2.828	2.978	3.916	2.464	1.215	1.894
P50x20	2.899	3.197	2.762	2.655	3.036	2.963	2.250	1.331	2.343
P100x10	2.708	3.018	3.256	2.737	3.478	3.843	2.510	1.504	2.347
Baseline	3.137	3.112	2.997	3.072	3.400	3.561	2.700	1.262	1.990

The baseline design was infeasible and resulted in high peak acceleration. The designs obtained from random search methods were feasible and had slightly better performance compared to the baseline design. However, major improvements were obtained by using NSGA-II simulations. All designs were not only feasible but also, significant reductions in peak acceleration (32–55%) and increase in time-to-zero-velocity (4-6%) were obtained with same or lower mass and maximum displacement values. While no design variable hit the bound, no patterns were identified among all feasible designs.

Table 7: Performance of the selected optimal design obtained using different configurations. SP: Stage Pulse

	Objectives				Constraints			
Case	X_{crash}	Accel	Mass	Time-to-0V	X_{crash}	SP-1	SP-2	SP-3
Monte-Carlo	719.3	112019	1.8187	0.0836	719.3	6.96	19.87	23.82
Space Filling	719.0	111248	1.8030	0.0761	719.0	7.02	20.15	23.57
P20x50	719.3	53645	1.8123	0.0979	719.3	7.29	19.96	24.28
P40x25	720.2	52667	1.8061	0.0950	720.2	7.43	20.02	22.15
P50x20	717.3	80132	1.8059	0.1042	717.3	6.86	20.13	23.61
P100x10	720.0	65615	1.8181	0.1000	720.0	7.21	19.84	24.08
Baseline	711.1	116601	1.8122	0.0936	711.1	7.90	21.18	25.23

Overall, the results showed merits of using evolution for multi-objective optimization. For the same amount of computational effort, the evolution helped find more feasible solutions and yielded many trade-off solutions. Probably, the GA simulation P20x50 performed the best. In general, there might be some variability in the performance because of the dependence on initialization procedures.

6. Summary

The practical operability of the elitist non-dominated sorting genetic algorithm is studied in this paper. Specifically, the focus is on finding the best compromise in evolution (number of generations) and diversity (population size) when the total number of simulations is fixed. Three examples that include one analytical, one engineering, and one crashworthiness simulation problem are used to study different combinations of population size and number of generations. The maximum number of simulations for each case was fixed at 1000. Four cases of NSGA-II simulations and two random sampling methods were studied.

As expected, it was observed that all NSGA-II simulations performed significantly better than Monte-Carlo sampling and space-filling methods. Low population size resulted in the most number of feasible points particularly when the size of feasible region was very small. Moderate population size also resulted in a high number of local non-dominated solutions. However, the NSGA-II case where a population of 20 individuals evolved for 50 generations resulted in the best convergence characteristics. In general, more evolution led to better convergence.

While all cases resulted in fairly uniformly distributed points on respective local non-dominated fronts, the spread of the local non-dominated solutions was the best for moderately sized populations. It can be concluded that evolution is more important to obtain good convergence and large population size are needed to preserve diversity. As a compromise, one can use moderate population size (e.g. 40) to balance convergence and diversity on the Pareto front.

The simulations of the crashworthiness example also demonstrated the significance of using parallel processing to reduce the wall clock time. Furthermore, it was observed that multi-objective optimization resulted in significant improvement in the performance. While the baseline design was infeasible, the best optimal design (obtained using a population of size 20 evolved for 50 generations) resulted in more than 50% reduction in the peak-acceleration and nearly 6% increase in the time to reach zero velocity while keeping the mass of the vehicle and maximum displacement nearly constant. All constraints were also satisfied. This example also demonstrated that, for problems with a small feasible region, the number of feasible solutions can be significantly increased in the first few generations involving about 200 simulations.

7. Acknowledgements

The authors would like to thank Subramani Balasubramanyam, Dilip Bhalsod, and Marko Thiele for their help in setting up the crash test problem.

8. References

[1] K. Deb, *Multi-Objective Optimization Using Evolutionary Algorithms*, Wiley, 2001, New York.

[2] N. Stander, W.J. Roux, T. Goel, T. Eggleston and K.J. Craig, *LS-OPT® Version 3.4 User's Manual*, Livermore Software Technology Corporation, April 2009.

[3] T. Goel and N. Stander, Multi-objective optimization with LS-OPT, *6th German LS-DYNA Forum*, Frankenthal Germany, Oct 11-12 2007.

[4] J.O. Hallquist, *LS-DYNA® Version 971 User's Manual,* Livermore Software Technology Corporation, October 2007.

[5] D.E. Goldberg, K. Deb and J.H. Clark, Genetic algorithms, noise, and the sizing of the populations, *Complex Systems*, 6(4), 333-362, 1992.

[6] G. Harik, E. Cantú-Paz, D.E. Goldberg and B.L. Miller, The gambler's ruin problem, genetic algorithms, and the sizing of the populations, Evolutionary Computation Journal, 7(3), 231-254, 1999.

[7] ME Johnson, LM Moore, D Ylvisaker, Minimax and maximin distance designs. *Journal of Statistical Planning and Inference*, 26, 131-148, 1990.

[8] A. Osyczka and S. Kundu, A new method to solve generalized multicriteria optimization problems using the simple genetic algorithm, *Structural Optimization*, 10(2), 94-99, 1995.

[9] K. Musselman and J. Talavage, A trade-off cut approach to multiple objective optimization, *Operations Research*, 28(6), 1424-1435, 1980.

[10] National Crash Analysis Center (NCAC), Public Finite Element Model Archive, www.ncac.gmu.edu/archives/model/index.html, 2001, (last accessed on Mar 13, 2009).

[11] T. Goel, Y.-Y. Lin and N. Stander, Direct multi-objective optimization through LS-OPT using small number of crashworthiness simulations, 10th International LS-DYNA User's Conference, Detroit MI, Jun 8-10, 2008.