Adaptive Simulated Annealing for Global Optimization in LS-OPT

Tushar Goel, Nielen Stander

Livermore Software Technology Corporation,

Livermore, CA, USA 94551

Summary:

The efficient search of global optimal solutions is an important contemporary subject. Different optimization methods tackle the search in different ways. The gradient based methods are among the fastest optimization methods but the final optimal solution depends on the starting point. The global search using these methods is carried out by providing many starting points. Other optimization methods like evolutionary algorithms that mimic the natural processes like evolution, and simulated annealing that emulates the metal cooling process via annealing can find the global optima but are criticized due to high computational expense. The adaptive simulated annealing algorithm has been proposed to be an efficient global optimizer. This algorithm is implemented in LS-OPT. A few analytical examples and meta-model based engineering optimization examples are used to demonstrate the efficiency of the global optimization using ASA. The optimization results are also compared with the existing LFOPC and genetic algorithm optimization methods.

Keywords:

Adaptive simulated annealing, Global constrained optimization, Genetic algorithm, LFOPC, Reliability based design optimization (RBDO), Crashworthiness optimization.

1 Introduction

Optimization is fast becoming a vital part of the design process in the engineering community due to ever increasing competitive pressure and other market forces. The efficient search of global optimal solution is much desired and many optimization methods have been proposed [1]. The gradient based algorithms are known to be the fastest algorithms however the convergence performance of these methods is dependent on the starting point. Often multiple starting points are used and then the best local optimal design is selected. One such method LFOPC [2] is available in LS-OPT® [3]. Among the global optimization methods, the genetic algorithms [4, 5], simulated annealing [6], particle swarm optimization [7] methods are the most popular but the search process is computationally very expensive. The genetic algorithms (GA) are particularly suited to find the Pareto optimal front for multi-objective optimization [8]. Ingber [9,10] modified the conventional simulated annealing algorithm to significantly improve the convergence rate. This algorithm is known as adaptive simulated annealing (ASA). In this paper, the performance of three optimization algorithms, LFOPC, GA and ASA as implemented in LS-OPT, are compared using a few examples.

2 Optimization Methods in LS-OPT

This section briefly describes different optimization methods available in LS-OPT.

2.1 LFOPC

LFOPC is a gradient based optimization method developed by Snyman [2] that differs conceptually from other gradient methods, such as SQP, in that no explicit line searches are performed. This algorithm generates a dynamic trajectory path, from a starting point, towards a local optimum. The LFOPC algorithm uses a penalty function formulation to incorporate constraints into the optimization problem and solves the optimization problem in three phases: Phase 0, Phase 1 and Phase 2. In Phase 0, the active constraints are introduced as mild penalties through the pre-multiplication of a moderate penalty parameter value. This allows for the solution of the penalty function formulation where the violation of the (active) constraints are premultiplied by the penalty value and added to the objective function in the minimization process. After the solution of Phase 0 through the leap-frog dynamic trajectory method, some violations of the constraints are inevitable because of the moderate penalty. In the subsequent Phase 1, the penalty parameter is increased to more strictly penalize violations of the remaining active constraints. Finally, and only if the number of active constraints exceed the number of design variables, a compromised solution is found to the optimization problem in Phase 2. Otherwise, the solution terminates having reached convergence in Phase 1.

2.2 Genetic Algorithm (GA)

The genetic algorithm is a population-based, probabilistic, global optimization method that emulates the Darwinian principle of 'survival of the fittest'. The concept of nature inspired algorithms was first envisaged by Prof. John Holland [4] at the University of Michigan in mid sixties. Later on this concept gained momentum in engineering optimization following the work of Prof. David Goldberg [5] and his students. The search is driven by three genetic operators, namely, selection, crossover and mutation. The search is carried out until a fixed number of generations are completed or when there are no significant improvements over a large number of generations. The constraints are handled using an efficient constraint handling strategy proposed by Deb [11] that emphasizes the feasible region before trying to minimize the objective function. The real-coded version [12] is implemented in LS-OPT [3] i.e., the variables are not mapped to binary space.

2.3 Adaptive Simulated Annealing (ASA)

The simulated annealing (SA) is a global stochastic optimization algorithm that mimics the metallurgical annealing process (Kirkpatrick [6]). The objective function is often called 'energy' E and is assumed to be related to the state, popularly known as temperature T, by a probability distribution. During the course of optimization, new points are sampled and accepted using a probabilistic criterion such that inferior points also have non-zero probability of getting accepted. The temperature is also updated. The search terminates when the temperature has fallen substantially. While the original SA algorithm allowed a very slow rate for reducing the temperature and hence a very high cost, Ingber [9, 10] developed complex modifications to the sampling method that enabled the use of very high cooling rates and hence reduced the simulation cost. He used non-uniform sampling rates for variables and used different cooling rates in the variable and function spaces. He also introduced a concept called 'reannealing' that updated the cooling rate associated with each parameter by accounting the sensitivities of the objective function. The LS-OPT implementation uses a penalty function approach to handle constraints.

3 Test Examples

Two analytical and two engineering examples are used to demonstrate the applicability of the adaptive simulated annealing. The performance of the ASA is also compared with the LFOPC and GA optimization methods.

3.1 Analytical Examples

3.1.1 Schwefel Function

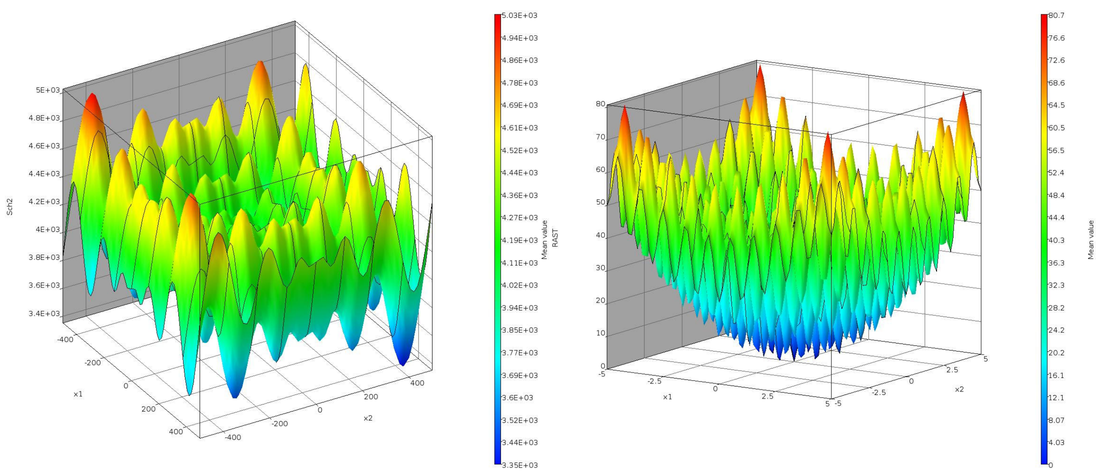
$$f(\mathbf{x}) = 418.9829n - \sum_{i=1}^{n} x_i \sin(\sqrt{|x_i|});\ -500 \le x_i \le 500. \quad (1)$$

The global optimal solution is $f^* = 0.0$, when all $x_i^* = 420.9687$.

3.1.2 Rastrigin Function

$$f(\mathbf{x}) = 10n + \sum_{i=1}^{n} (x_i^2 - 10\cos(2\pi x_i));\ -5 \le x_i \le 5. \quad (2)$$

The global optimal solution is $f^* = 0.0$, when all $x_i^* = 0$.



a) Schwefel b) Rastrigin

Figure 3-1: Function plots x_3 to x_{10} are fixed at 0.

Both analytical examples are solved for 10 variables each. The complexity of these functions is illustrated in Figure 3-1. One can clearly see the difficulty these examples would pose for gradient based methods.

3.2 Engineering Examples

3.2.1 Crash optimization (CRASH)

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 [13]), shown in Figure 3-2, has approximately 55K elements. Nine gauge thicknesses affecting the members listed in Table 1, are taken as design variables and affected parts are shown in Figure 3-2.

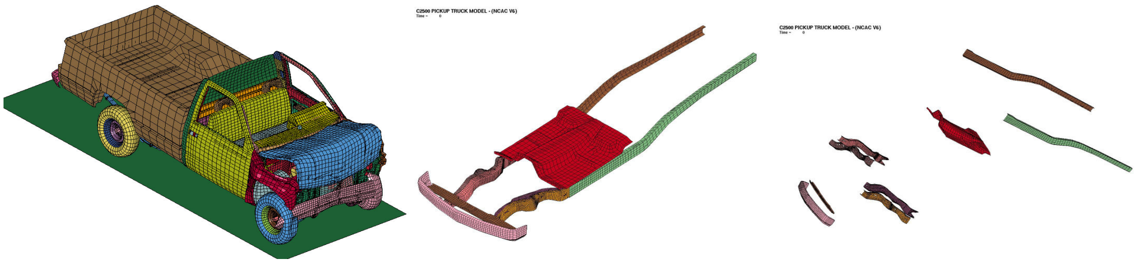


Figure 3-2: Finite element model, and thickness design variables for CRASH example.

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.

Table 1: Design variables for CRASH example

Variable description	Name	Lower bound	Baseline design	Upper bound
Rail front-right-inner	t_1	2.500	3.137	3.765
Rail front-right-outer	t_2	2.480	3.112	3.750
Rail front-left-inner	t_3	2.400	2.997	3.600
Rail front-left-outer	t_4	2.400	3.072	3.600
Rail right-back	t_5	2.720	3.400	4.080
Rail left-back	t_6	2.850	3.561	4.270
Bumper	t_{10}	2.160	2.700	3.240
Radiator bottom	t_{64}	1.000	1.262	1.510
Cabin bottom	t_{73}	1.600	1.990	2.400

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.

Table 2: Design constraints

	Upper bound
Maximum displacement (x_{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 design variable bounds are given in Table 1 and the performance constraints, namely maximum displacements and stage pulses, are specified in Table 2. 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:

$$-\frac{k}{(d_2 - d_1)}\int_{d_1}^{d_2} \ddot{x}\mathrm{d}x \quad k = 0.5 \text{ for } j = 1, k = 1.0 \text{ otherwise}; \tag{3}$$

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® [14] 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, was used to run simulations[1]. More details about running the simulation appear elsewhere [15].

3.2.2 Multi-disciplinary Reliability Based Design Optimization (RBDO)

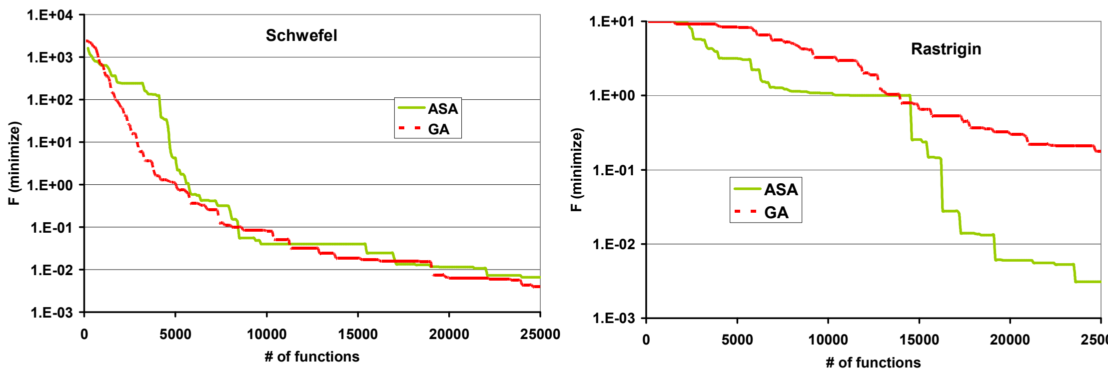
A RBDO example with six test cases is also considered. There are 12 design variables, 32 responses, seven objectives, and 25 probabilistic constraints. A weighted sum of all normalized objectives is used to formulate the single-objective optimization problem. The cost of evaluating each design is significant due to the probabilistic computations.

The global optimal solution for the engineering examples is not known so the improvements are measured with respect to the baseline designs.

4 Results

4.1 Analytical Examples

The convergence histories for the adaptive simulated annealing (ASA) and the genetic algorithm (GA) methods for the analytical examples are shown in Figure 4-1.



a) Schwefel b) Rastrigin

Figure 4-1: Convergence history of ASA and GA optimization methods on analytical examples.

4.1.1 Schwefel function

Table 3: Predicted optimal solution of Schwefel function from different optimization methods. The global optimal solution is $f^* = 0.0$, when all $x_i^* = 420.9687$.

	F	x_1	x_2	x_3	x_4	x_5
ASA	6.571e-3	420.9	421	421	420.9	421
GA	3.953e-3	421.1	421	420.9	421	420.9
LFOP	1.072e+3	420.5	420.5	420.5	-500	420.5
		x_6	x_7	x_8	x_9	x_{10}
ASA		421	420.9	421	421.1	421
GA		420.9	421	421	420.9	421
LFOP		-303	420.5	-500	-500	-500

The optimal solutions for the Schwefel function using different optimization methods are shown in Table 3. As expected, the ASA and the GA methods resulted in the optimal solutions close to the global optimal but the LFOP solution was very poor despite 7 different starting points. The

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

convergence history for the ASA and the GA methods shown in Figure 4-1(a) also shows comparable performance with continuous reduction in the objective function.

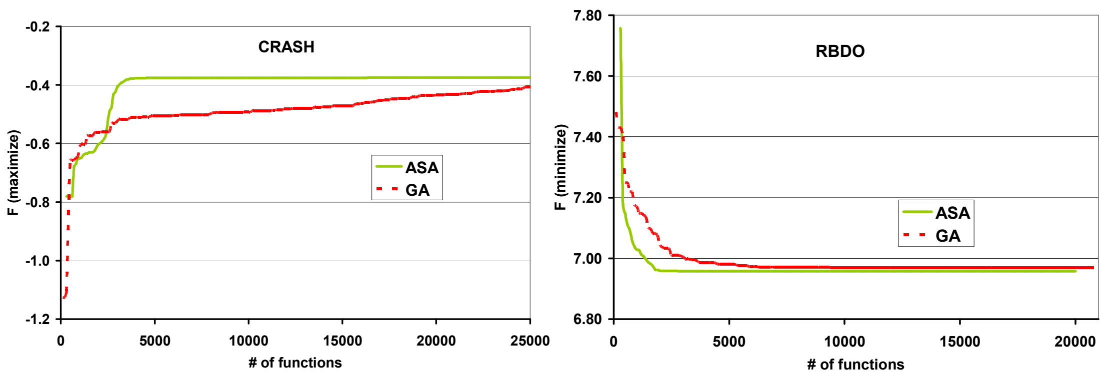
4.1.2 Rastrigin function

The optimal solutions from different optimization methods are tabulated in Table 3. As observed for the Schwefel function, the ASA and the GA methods resulted in the optimal solutions close to the global optimal solution but the LFOP solution was stuck in local optima. The performance of the ASA was the best among all optimization methods. Figure 4-1(b) also indicated better convergence rate for the ASA algorithm compared to the GA.

Table 4: Predicted optimal solution of Rastrigin function from different optimization methods. The global optimal solution is f = 0.0, when all $x_i^* = 0$.*

	F	x_1	x_2	x_3	x_4	x_5
ASA	3.119e-3	2.80e-3	-7.09e-3	4.48e-4	-4.01e-4	-1.32e-3
GA	1.790e-1	-1.51E-02	1.32E-03	9.47E-04	-1.31E-02	6.06E-03
LFOP	9.999e0	0.99	0.99	0.99	0.99	0.99
		x_6	x_7	x_8	x_9	x_{10}
ASA		1.18e-3	-2.70e-4	-4.68e-5	-1.95e-3	-1.32e-4
GA		-7.04E-03	-4.04E-03	-9.22E-03	-8.29E-03	-1.56E-02
LFOP		0.99	0.99	0.99	0.99	0.99

4.2 Engineering Examples



a) CRASH b) RBDO

Figure 4-2: Convergence history of ASA and GA methods for engineering optimization problems.

4.2.1 CRASH Example

Table 5: Predicted optimal and the baseline designs for the CRASH example.

	Objectives				Constraints				
	F	Disp	Accel	Mass	Time	Disp	SP1	SP2	SP3
ASA	-0.374	1.000	-1.576	-1.006	1.208	1.000	1.000	0.952	0.963
GA	-0.408	1.000	-1.602	-1.006	1.201	1.000	0.999	0.965	0.963
LFOP	-0.371	1.000	-1.580	-1.006	1.215	1.000	1.000	0.942	0.964
Baseline	-1.228	0.975	-2.250	-1.007	1.054	0.975	1.051	1.054	1.003
	t_1	t_2	t_3	t_4	t_5	t_6	t_{10}	t_{64}	t_{73}
ASA	2.509	3.635	2.400	2.576	2.720	4.270	2.602	1.510	2.148
GA	2.708	3.416	2.400	2.695	2.721	4.269	2.584	1.489	2.095
LFOP	2.500	3.706	2.400	2.401	2.720	4.270	2.615	1.510	2.200
Baseline	3.137	3.112	2.297	3.072	3.400	3.561	2.700	1.262	1.990

Table 5 enlists the predicted optimal solutions for the CRASH example. The baseline design was infeasible and caused high peak acceleration and had a low time to reach velocity. Optimization significantly reduced the peak acceleration significantly and improved the time to reach zero velocity. While the LFOPC (with 20 multi-starts) resulted in the best performance, the ASA result was comparable. The GA resulted in the worst performance of the three optimization methods though by tuning some parameters, the optimal solution was comparable. That result is not reported here to ensure fair comparison among all optimization methods. The design variable values for the optimal solution obtained from different optimization methods were in the same vicinity. The convergence history (Figure 4-2(b)) comparison of the ASA and the GA also indicated that the ASA converged much faster than the GA.

4.2.2 RBDO Example

Finally, the optimization results for the RBDO example are presented in Table 6. As was observed for other problems, the ASA performed very well for the RBDO example too. The LFOP algorithm (with 23 starting points) performed the best. The performance of the GA was slightly poor than other algorithms but with some parameter tuning it could also be improved (result not shown here). The convergence history (Figure 4-2(b)) also showed that the ASA converged to the optima faster than the GA.

Table 6: Optimization results for the RBDO example.

	F	F_1	F_2	F_3	F_4	F_5	F_6	F_7	Feasibility
ASA	6.957	1.000	1.000	1.203	0.877	1.075	0.789	1.013	YES
GA	6.969	1.000	1.000	1.209	0.872	1.063	0.789	1.036	YES
LFOP	6.956	1.000	1.000	1.202	0.878	1.076	0.789	1.011	YES
Baseline	8.586	1.235	1.687	1.243	0.957	1.031	1.081	1.352	NO

4.3 Simulation Time

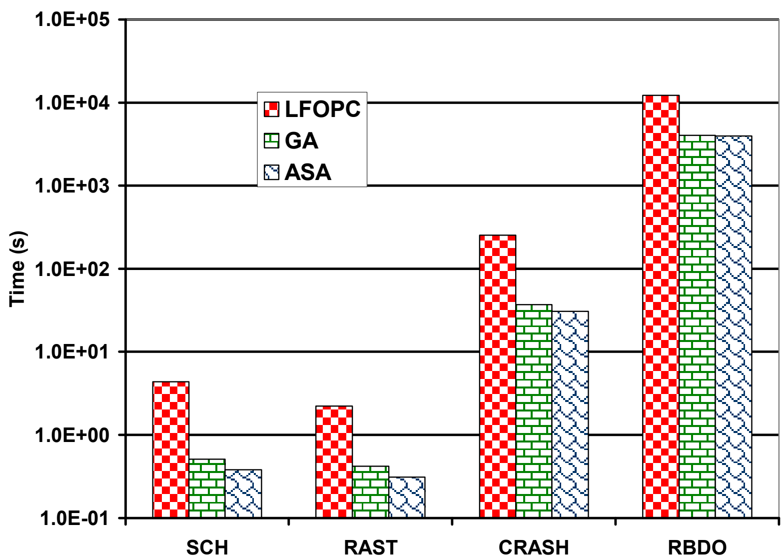


Figure 4-3: Simulation time.

While the convergence performance of different algorithms was studied in the previous two subsections, the computational expense to obtain the optimal solution is shown in Figure 4-3. All simulations were carried on the Intel Xeon 2.66 GHz processor with 4 GB memory. The RBDO problem took the maximum time because the estimation of the reliability constraints required an expensive procedure. Among all optimization algorithms, the computational expense of LFOPC was the highest due to the multi-start procedure. On the other hand the adaptive simulated annealing and the genetic algorithm resulted in optimal solution in an order of magnitude lesser time than LFOPC. The ASA was the fastest algorithm.

5 Conclusions

In this paper, a comparison of different optimization algorithms available in LS-OPT is carried out using two analytical examples and two engineering problems including RBDO examples. It was observed that the adaptive simulated annealing algorithm was the best optimizer for all test problems.

This method converged closest to the optimal solution with the least computational time. The LFOPC algorithm converged to the optimal solutions for engineering problems but was poorer for analytical problems. Also the search using LFOPC was an order of magnitude more expensive than the genetic algorithm and the ASA methods. The genetic algorithm is a good alternative global optimization method to ASA as it uses slightly higher computational expense while producing similar accuracy.

6 Literature

[1] SS Rao, *Engineering Optimization: Theory and Practice*, Wiley, 1996.

[2] JA Snyman, The LFOPC Leap-Frog Algorithm for Constrained Optimization, *Computers and Mathematics with Applications*, 40, 2000, 1085-1096.

[3] N Stander, WJ Roux, T Goel, T Eggleston, KJ Craig, *LS-OPT® Version 3.4 User's Manual*, Livermore Software Technology Corporation, April 2009.

[4] JH Holland, *Adaptation in Natural and Artificial Systems*, MIT Press, 1992.

[5] DE Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley Publishing Company, 1989.

[6] S Kirkpatrick, CD Gelatt, MP Vecchi, Optimization by Simulated Annealing, *Science*, 220, 1983, 671-680.

[7] J Kenndy, R Eberhart, Particle Swarm Optimization, *In Proceedings of IEEE International Conference on Neural Networks*, Perth Australia, Nov 1995, 1942-1948.

[8] K Deb, *Multiobjective Optimization using Evolutionary Algorithms*, Wiley Chichester UK, 2001.

[9] L Ingber, Very Fast Simulated Re-annealing, *Mathematical Computer Modeling*, 12, 1989, 967-983.

[10] L Ingber, *Adaptive Simulated Annealing (ASA)*, [ftp.alumni.caltech.edu: /pub/ingber/ASA.tar.gz], Lester Ingber Research, McLean VA, 1993.

[11] K Deb, An Efficient Constraint Handling Method for Genetic Algorithms, *Computer Methods in Applied Mechanics and Engineering*, 186, 2000, 311-338.

[12] K Deb, RB Agrawal, Simulated Binary Crossover for Continuous Search Space, *Complex Systems*, 9, 1995, 115-148.

[13] 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).

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

[15] T Goel, Y-Y Lin, 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.