An Adaptive Surrogate-Assisted Strategy for Multi-Objective Optimization

Nielen Stander

Livermore Software Technology Corporation, Livermore, California, USA, nielen@lstc.com

1. Abstract

A sequential metamodel-based optimization method is proposed for multi-objective optimization problems. The algorithm, designated as Pareto Domain Reduction, is an adaptive sampling method and an extension of the classical Domain Reduction approach (also known as the Sequential Response Surface Method). In addition to standard benchmark examples, a Multidisciplinary Design Optimization (MDO) example involving a vehicle impact is used to demonstrate that the accuracy conforms to the Direct NSGA-II "exact" method while using a small fraction of its computational effort.

2. Keywords: Multi-objective optimization, metamodels, surrogates, crashworthiness optimization.

3. Introduction

Multi-objective optimization is typically conducted using a Direct Optimization approach such as the Non-Dominated Sorting Genetic Algorithm (NSGA-II) [1]. NSGA-II can be used to obtain "exact" results but requires a very large number of simulations so can be expensive, especially in crashworthiness optimization where simulations involve highly detailed Finite Element models and costly nonlinear dynamic analysis.

MOO is also possible with metamodel-assisted methods. A naïve approach is to conduct a large number of simulations by using a global Space Filling experimental design. This approach typically relies on uniform global sampling to construct the metamodel and can be faster than direct solution. The accuracy of the approach is severely affected by scaling as the number of simulations to obtain an accurate metamodel in a large variable space can be very large. For single objective problems, this deficiency can be addressed by an iterative Domain Reduction approach (known as the Sequential Response Surface Method or SRSM) [2,3] in which the search domain size is gradually reduced for each iteration. With suitable heuristics the SRSM approach can provide a sufficiently accurate result, but so far has not been adapted to multi-objective problems in which multiple solutions are possible.

Other studies of surrogate-assisted methods can be found in the literature. A well known method is ParEGO based on the Efficient Global Optimization (EGO) algorithm. In two studies [4,5], this algorithm was found to be competitive with NSGA-II, although the test problems were limited to 8 variables and the results were based on a fixed, limited number of runs rather than an attempt to converge to a fine tolerance.

In this study an adaptive domain reduction method designated as Pareto Domain Reduction (PDR) is introduced for improving efficiency and accuracy. The method employs heuristics which are similar to those of SRSM but since multiple optimal solutions are possible, it uses the irregular subregion of the Pareto Optimal Frontier (POF) as a sampling domain. The size of this subregion is iteratively reduced in order to intensify the exploration in the predicted neighborhood of the POF. The method is conservative in the sense that early sampling is global with a gradual convergence to the POF. Hence it can also be viewed as an adaptive sampling approach. The proposed method has the additional advantage that, if a multi-objective problem has only one optimal solution, it degenerates to SRSM which unifies and simplifies the methodology and user choice.

Since Radial Basis Function Networks [6,7] are typically used as metamodels, a Space Filling approach is used to obtain a well spaced sampling in the reduced domain. This avoids point duplication and maximizes the accuracy in the POF neighborhood.

The remainder of the paper deals with a detailed description of the methodology followed by two standard benchmark problems from the literature and a multi-objective multi-disciplinary (MDO) crashworthiness/modal analysis example of a vehicle. The model is not very large, but sufficiently realistic and representative of a typical model used in industry. The results show that the PDR method is highly accurate and potentially an order of magnitude cheaper than the direct NSGA-II approach.

4. Methodology

4.1 Sequential Response Surface method (SRSM)

To explain the methodology, the heuristics of a domain reduction approach for single-objective optimization are firstly explained. The purpose of the SRSM method is to allow convergence of the single-objective solution to a prescribed tolerance.

The SRSM method [2,3] uses a region of interest, a subspace of the design space, to determine an approximate optimum. A range is chosen for each variable to determine its initial size. A new region of interest centers on each successive optimum. Progress is made by moving the center of the region of interest as well as reducing its size. The starting point $\mathbf{x}^{(0)}$ will form the center point of the first region of interest. The lower and upper bounds $(x_i^{rL,0}, x_i^{rR,0})$ of the initial subregion are calculated using the specified initial range value $r_i^{(0)}$ so that

$$x_i^{rL,0} = x_i^{(0)} - 0.5r_i^{(0)} \text{ and } x_i^{rU,0} = x_i^{(0)} + 0.5r_i^{(0)}; \; i = 1,...,n \tag{1}$$

where n is the number of design variables. The modification of the ranges on the variables for the next iteration depends on the oscillatory nature of the solution and the accuracy of the current optimum.

Oscillation: A *contraction parameter* γ is firstly determined based on whether the current and previous designs $\mathbf{x}^{(k)}$ and $\mathbf{x}^{(k-1)}$ are on the opposite or the same side of the region of interest. Thus an *oscillation indicator c* may be determined in iteration k as

$$c_i^{(k)} = d_i^{(k)} d_i^{(k-1)} \tag{2}$$

where

$$d_i^{(k)} = 2\Delta x_i^{(k)} / r_i^{(k)}; \; \Delta x_i^{(k)} = x_i^{(k)} - x_i^{(k-1)}; d_i^{(k)} \in [-1;1] \tag{3}$$

The oscillation indicator (purposely omitting indices i and k) is normalized as $\hat{c}$ where

$$\hat{c} = \sqrt{|c|}\, sign(c). \tag{4}$$

The contraction parameter γ is then calculated as

$$\gamma = 0.5(\gamma_{pan}(1+\hat{c}) + \gamma_{osc}(1-\hat{c})) \,. \tag{5}$$

The parameter γ_{osc} is typically 0.5-0.8 representing shrinkage to dampen oscillation, whereas γ_{pan} represents the pure panning case and therefore unity is typically chosen.

Accuracy: The accuracy is estimated using the proximity of the predicted optimum of the current iteration to the starting (previous) design. The smaller the distance between the starting and optimum designs, the more rapidly the region of interest will diminish in size. If the solution is on the bound of the region of interest, the optimal point is estimated to be beyond the region. Therefore a new subregion, which is centered on the current point, does not change its size. This is called *panning*. If the optimum point coincides with the previous one, the subregion is stationary, but reduces its size (*zooming*). Both panning and zooming may occur if there is partial movement. The range $r_i^{(k+1)}$ for the new subregion in the (k + 1)-th iteration is then determined by:

$$r_i^{(k+1)} = \lambda_i r_i^{(k)}; i = 1,...,n; \; k = 0,..., niter \tag{6}$$

where λ_i represents the *contraction rate* for each design variable. To determine $\lambda_i, d_i^{(k)}$ is incorporated by scaling according to a *zoom parameter* η that represents pure zooming and the contraction parameter γ to yield the contraction rate

$$\lambda_i = \eta + \left|d_i^{(k)}\right|(\gamma - \eta) \tag{7}$$

for each variable.

When used in conjunction with neural networks or Kriging, the same heuristics are applied as described above but the nets are constructed using *all* the available points, including those belonging to previous iterations. Therefore the response surfaces are progressively updated in the region of the optimal point. Adaptive Continuous Space Filling experimental design is typically used for sampling.

4.2 Space Filling experimental design

In metamodel-based optimization two important methods for constructing approximations are the approximation method (metamodel type) and the experimental design (sampling method). In multi-objective optimization, suitable metamodel types are Radial Basis Function Networks, Feedforward Neural Networks or Kriging methods.

These metamodels are typically fitted using all available simulation results, also those of previous iterations. Because of the existence of fixed design points from earlier iterations, iterative metamodel-based methods ideally require adaptive sampling schemes. For adaptive sampling, Continuous Space Filling (CSF) experimental designs [8] are highly suitable.

The key to space-filling experimental designs is in generating 'good' random points and achieving reasonably uniform coverage of sampled volume for a given (user-specified) number of points. The constrained randomization method termed Latin Hypercube Sampling (LHS) [9], has become a popular strategy to generate points on the 'box' (hypercube) design region. The method implies that on each level of every design variable only one point is placed, and the number of levels is the same as the number of runs. The levels are assigned to runs either randomly or so as to optimize some criterion, e.g. so that the minimal distance between any two design points is maximized ('maximin distance' criterion). A probabilistic search technique, *adaptive simulated annealing* [11,12] is used to optimize the experimental design.

4.3 Adaptive sampling for multi-objective optimization

In the single-objective domain reduction scheme (Section 4.1), each new domain is centered on the optimum point of the previous iteration. If the sub-domain concept is to be extended to multi-objective optimization, the simple sub-domain can in general no longer be centered on a single point since it may exclude a large part of the POF. The POF, in this case, is a predicted POF produced as a result of a metamodel optimization of the previous iteration.

Hence, it is proposed that a POF subdomain, enveloping the entire POF, be created as the union of a suitable number of simple overlapping sub-domains of equal dimensions centered on chosen POF points denoted as *POF kernels* (Fig. 3). The number of POF kernels *m* depends on the current size of the sub-domain as well as the dimensionality of the objective function space. To provide a well distributed set of POF kernels, these are picked from the POF superset by maximizing the minimum distance between any two kernels in the variable space (a *discrete space filling* or DSF method). A simple procedure is to select the next point in a growing subset to be distant from the currently selected points. Hence, for each iteration, the set of POF kernels is a subset of the predicted POF. As a consequence of this strategy, it is anticipated that there will be relatively few, large sub-domains in the early iterations whereas the number of sub-domains will grow during the iterative process due to their gradual size reduction.

The domain reduction heuristics as described in Section 4.1 is applied using the motion (proximity and oscillation) of the optimum of the sum of the objectives. This approach ensures that the method devolves to SRSM for single objective optimization.

The next step is to fill all the sub-domains with a set of *diversity basis* points (Fig. 5). The purpose of the basis set is to serve as a pool of suitable points, within the POF region, from which to select design points for simulation. A continuous Space Filling principle is used [8,9]. This method maximizes the minimum distance between any two points, or between any point and a fixed point. Beginning with the first sub-domain (centered on a randomly chosen kernel), a number of points *p* is chosen within this sub-domain using CSF. The sampling process moves on to the second subdomain in which the CSF is repeated, filling the second sub-domain, but this time also considering the previously selected diversity points as part of the distance calculation. This process is continued until all the sub-domains are filled with points. The end result is an irregular sub-domain, enveloping the POF in the design space and which is space-filled with a large set of diversity basis points.

The final step before simulation again requires the DSF procedure to select *p* simulation design points for iteration *i*. The existence of all previous simulations is considered in the DSF execution. The set of points thus obtained is well distributed throughout the POF region and ready for simulation.

4.4 Algorithm

The algorithm steps are summarized below with references to Figures 1 to 7:

Initial conditions:

1. $k := 0$, choose *m* simulation designs in the full design space using CSF (Fig. 1).
2. Conduct the simulations, build the metamodel, construct an approximate POF using NSGA-II (Fig. 2).

For each iteration *k*:

1. $k := k + 1$
2. Select *m* kernels from the approximate POF generated in iteration (*k-1*) (Fig. 3).
3. Adapt the subregion size based on iteration (*k*-1).
4. Center a subregion on each kernel (Fig. 4).
5. Populate each subdomain with a number of *diversity* basis points (Fig. 5).
6. Select *p* points for simulation from the diversity basis points (Fig. 6).
7. Conduct *p* simulations (Fig. 7), build the metamodel and construct an approximate POF using NSGA-II

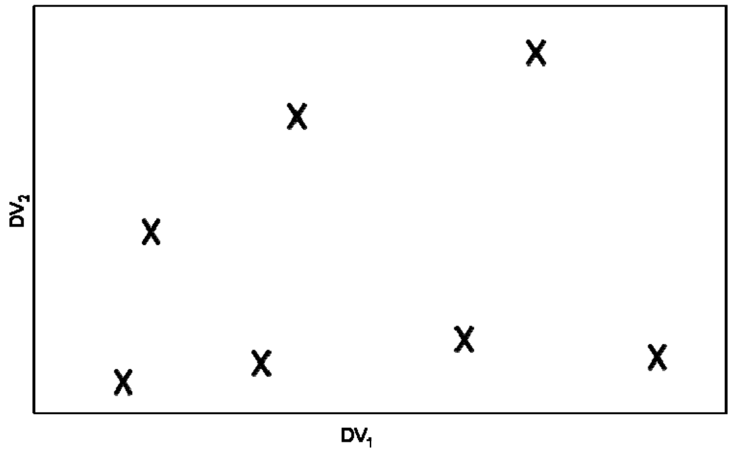

Figure 1: Sampling points (X) of the first iteration according to the Continuous Space Filling experimental design (schematic).

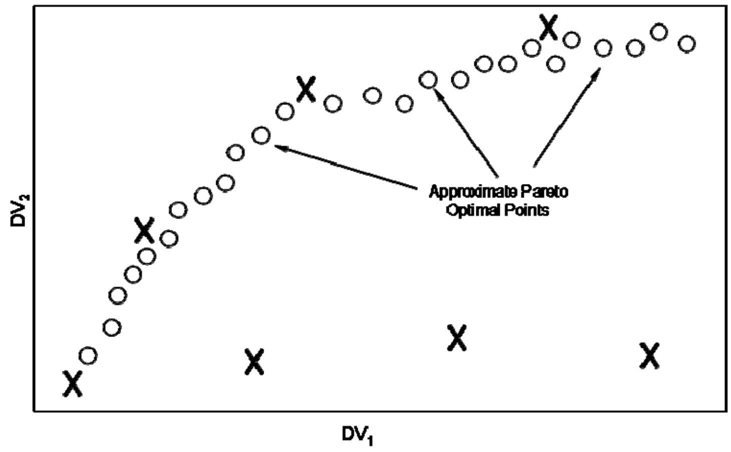

Figure 2: Conduct simulations at the X points. Construct a metamodel for each design criterion. Solve the approximate optimization problem to find the approximate Pareto Optimal Front (○).

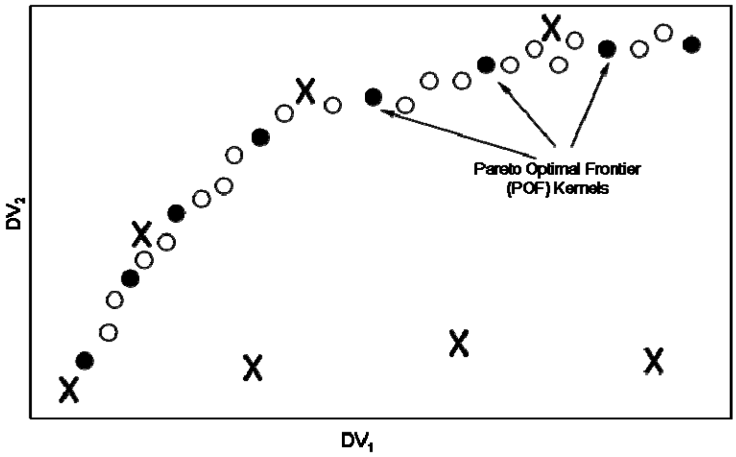

Figure 3: Using a Discrete Space Filling method, select a subset of the predicted POF as POF kernels (●).

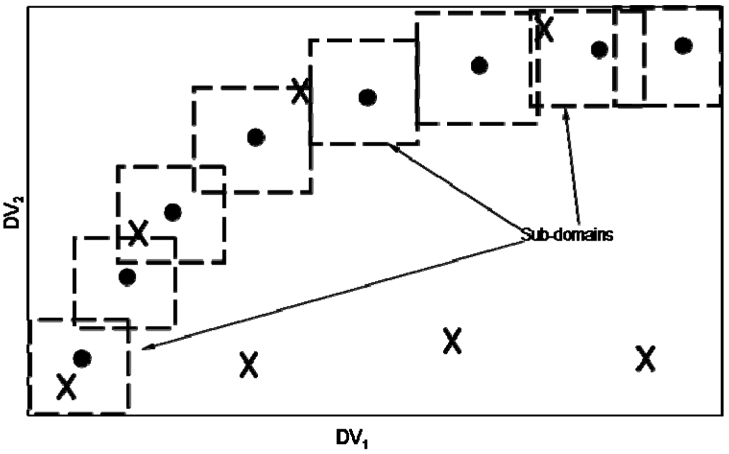

Figure 4: Center a sub-domain on each POF kernel (●).

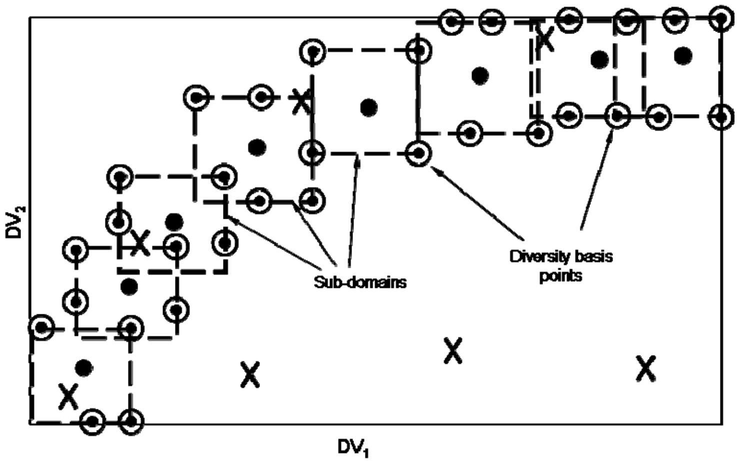

Figure 5: Fill the sub-domains with diversity basis points using a Continuous Space Filling procedure.

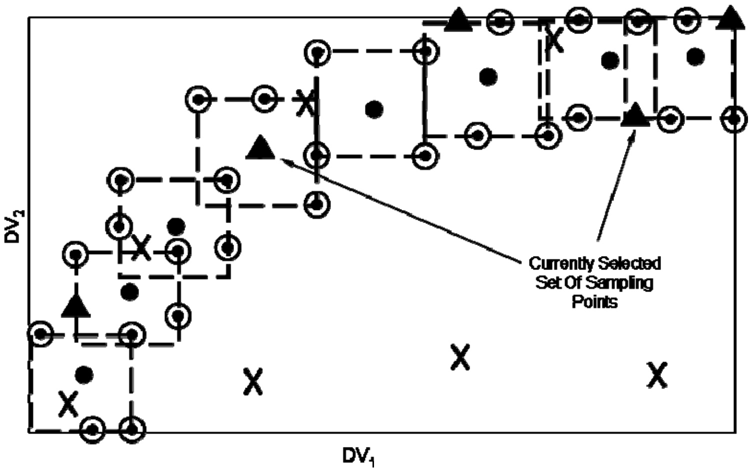

Figure 6: Sample *p* simulation points (▲) from the diversity basis set using the Discrete Space Filling approach.

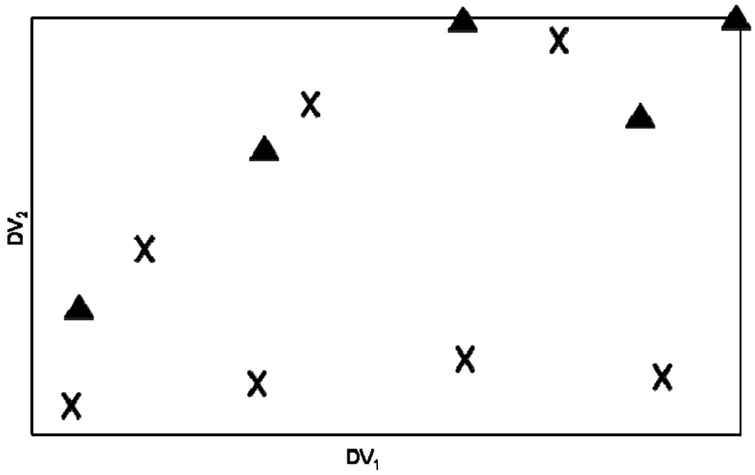

Figure 7: Final set of simulation points (▲) for iteration 2.

5. Examples

The method was implemented in the LS-OPT optimization code [3] and applied to a number of test problems. The domain reduction algorithm uses parameters:

$$\gamma_{\text{pan}} = 1, \ \ \gamma_{\text{osc}} = \eta = 0.8 \ .$$

5.1 Analytical benchmark tests

To validate the method, the results of two standard benchmark examples ZDT1 and ZDT2 with 30 variables (n = 30) each [1] are first presented. 25 Iterations using a population of 110 per iteration were required resulting in a total of 2850 simulations. 100 of these simulations were executed to construct the final Pareto Optimal Frontier. The Direct GA method required about 20,000 simulations The Pareto solutions are shown in Figs. 8 and 9. RBF networks were used as metamodels.

It should be noted that the predicted Pareto Optimal Frontier features a few errant Pareto solutions in the range $f_2 >$ 1. These occur as a result of a slight misalignment of the final metamodel which causes some solutions to be found on the f_2 – axis.

Minimize $$f_1(\mathbf{x})$$

Minimize $$f_2(\mathbf{x}) = g(\mathbf{x})h(f_1(\mathbf{x}), g(\mathbf{x}))$$

Example ZDT1:
$$f_1(\mathbf{x}) = x_1$$
$$g(\mathbf{x}) = 1 + \frac{9}{n-1}\sum_{i=2}^{n} x_i$$
$$h(f_1, g) = 1 - \sqrt{f_1 / g_1}$$

Example ZDT2:
$$f_1(\mathbf{x}) = x_1$$
$$g(\mathbf{x}) = 1 + \frac{9}{n-1}\sum_{i=2}^{n} x_i$$
$$h(f_1, g) = 1 - (f_1 / g)^2$$

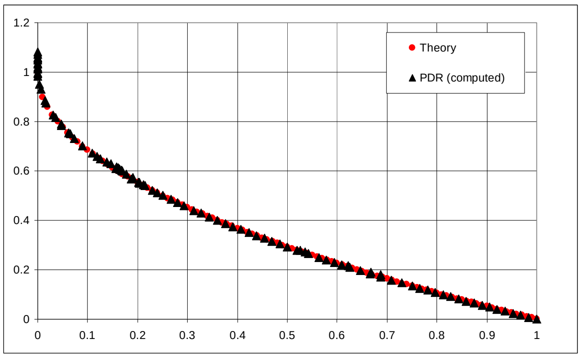

Figure 8: Pareto Optimal Frontier of ZDT1. Comparison with theoretical result

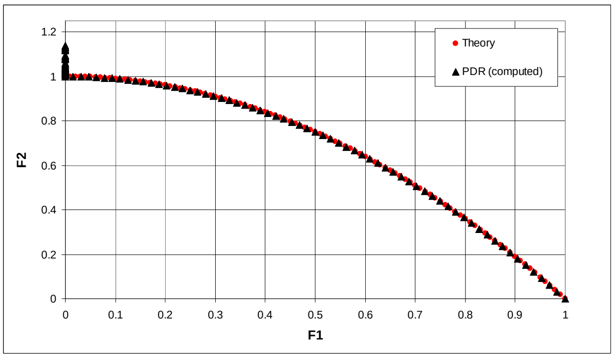

Figure 9: Pareto Optimal Frontier of ZDT2 Comparison with theoretical result.

5.2. Full vehicle MDO

The crashworthiness simulation also used for optimization in [13] considers a model containing approximately 30,000 finite elements of a National Highway Transportation and Safety Association (NHTSA) vehicle undergoing a full frontal impact. A modal analysis is performed on a ‘body-in-white’ model containing approximately 18,000 elements. The crash model for the full vehicle is shown in Figure 10 for the undeformed and deformed (time = 78ms) states, and with only the structural components affected by the design variables, both in the undeformed and deformed (time = 72ms) states, in Figure 11. The NVH model is depicted in Figure 12 in the first torsion vibrational mode. Only body parts that are crucial to the vibrational mode shapes are retained in this model. The design variables are all thicknesses or gages of structural components in the engine compartment of the vehicle (Figure 11), parameterized directly in the LS-DYNA input file. Twelve parts are affected, comprising aprons, rails, shotguns, cradle rails and the cradle cross member (Figure 11). LS-DYNA v.971 [14] is used for both the crash and NVH simulations, in explicit and implicit modes respectively.

The design formulation is as follows:

$$\text{Minimize Mass}$$

$$\text{Minimize Maximum intrusion}$$

subject to

$$\text{Maximum intrusion}(\boldsymbol{x}_{\text{crash}}) < 551.27\text{mm}$$

$$\text{Stage 1 pulse}(\boldsymbol{x}_{\text{crash}}) > 14.51\text{g}$$

$$\text{Stage 2 pulse}(\boldsymbol{x}_{\text{crash}}) > 17.59\text{g}$$

$$\text{Stage 3 pulse}(\boldsymbol{x}_{\text{crash}}) > 20.75\text{g}$$

$$41.38\text{Hz} < \text{Torsional mode frequency}(\boldsymbol{x}_{\text{NVH}}) < 42.38\text{Hz}$$

$\mathbf{x}_{\text{crash}} = \mathbf{x}_{\text{NVH}}$ = [rail_inner, rail_outer, cradle_rails, aprons, shotgun_inner, shotgun_outer, cradle_crossmember]^T

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

$$\text{Stage } i \text{ pulse} = \frac{-k}{d_2 - d_1}\int_{d_1}^{d_2} \ddot{x}\,\mathrm{d}x\ ; \quad k = 2 \text{ for } i = 1,\ 1.0 \text{ otherwise;}$$

with the limits [d_1;d_2] = [0;184]; [184;334]; [334;Max(displacement)] for i = 1,2,3 respectively, all displacement units in mm and the minus sign to convert acceleration to deceleration. The Stage 1 pulse is represented by a triangle with the peak value being the value used.

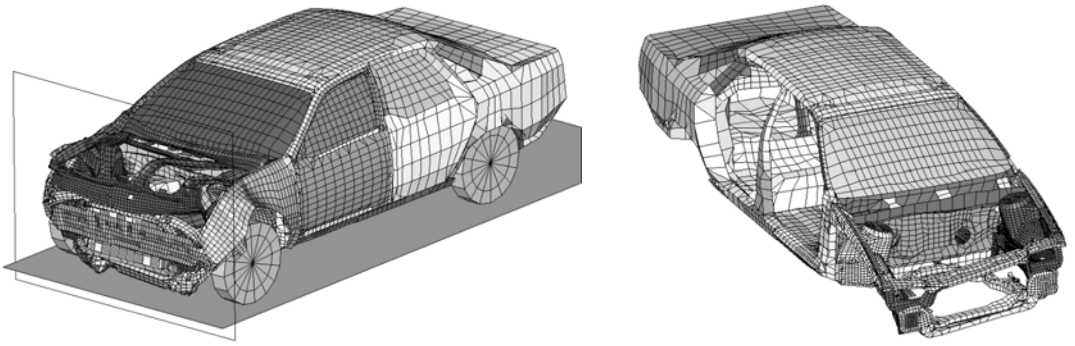

Figure 10: (a) Crash model of vehicle in the deformed state (78ms), (b) body-in-white model of vehicle in torsional vibration mode (38.7Hz)

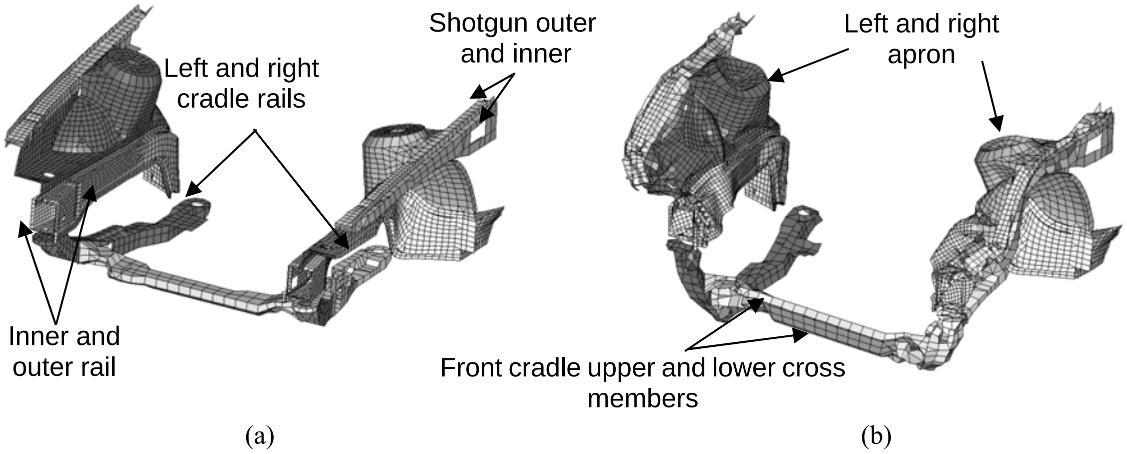
Shotgun outer and inner
Left and right cradle rails
Left and right apron
Inner and outer rail
Front cradle upper and lower cross members
(a)
(b)

Figure 11: Structural components affected by design variables – a) Undeformed and (b) deformed (time = 72ms)

Three optimization strategies were used to verify and compare the results:

- A direct optimization using the NSGA-II algorithm. A tolerance on the hypervolume change [15] of 10^{-4} was used to create a realistic benchmark result.
- A metamodel-assisted optimization using sequential global Radial Basis Function networks. A stopping criterion of 30 iterations was selected.
- A metamodel-assisted optimization using the Pareto Domain Reduction approach. RBF networks are used as metamodels. A stopping criterion of 30 iterations was selected.

The parameters for each run are shown in Table 1.

Table 1: Run statistics for Example 1 [16]. *Note that the number of simulations for the metamodel assisted methods includes a 100 final verification runs to construct a full trade-off curve. This number determines the size and diversity of the trade-off curve.

	Number of iterations/ generations	Number of simulations per iteration	Total number of simulations per case, including 100 verification simulations for the last iteration	Run configuration (Number of concurrent jobs × MPP cores per simulation)
Direct NSGA-II	75	160	12,000	80×4
Global	30	13	390+100*	13×4
PDR	30	13	390+100*	13×4

All runs were conducted on a 768-core Xeon-based cluster running up to 80×4-core MPP-DYNA simulations in parallel. The SunGrid Engine queuing system was used with job monitoring using the LS-OPT Graphical User Interface.

Although there is no mathematical criterion for global optimality, the Direct GA run is considered to be "exact" in the sense that it was run to a very fine tolerance (10^{-4}) of the dominated hyper-volume change. The choice of a population size of 160 was the result of some experimentation as an earlier choice of 100 produced a sub-optimal result.

Fig. 13 shows a comparison of the trade-off curves produced by the Direct GA optimization and the PDR run. The PDR results are simulation results of the final trade-off curve (i.e. *not* the approximate results). Note that the trade-off curve is discontinuous (two major discontinuities) with several concave and convex sections. In Fig. 14, the PDR result can be seen to closely approximate the exact result. The wall clock time for the PDR method was approximately 7 hours for the entire optimization.

Fig. 14 shows a comparison of the predicted trade-off curve and the computed curve using the PDR method which confirms the accuracy of the metamodel and sampling strategy. Note that the metamodel is based on all 390 (13×30) simulations, and not only the last 13.

The new PDR method was also compared to the existing global sampling approach (Fig. 15) in which the design points for building the metamodel are evenly distributed globally using a Space Filling sampling approach. The benefit of using the PDR method is clearly visible. Note that finer features such as discontinuities are not properly modeled by the global sampling, obviously because of a sparsity of simulation results near the optimum.

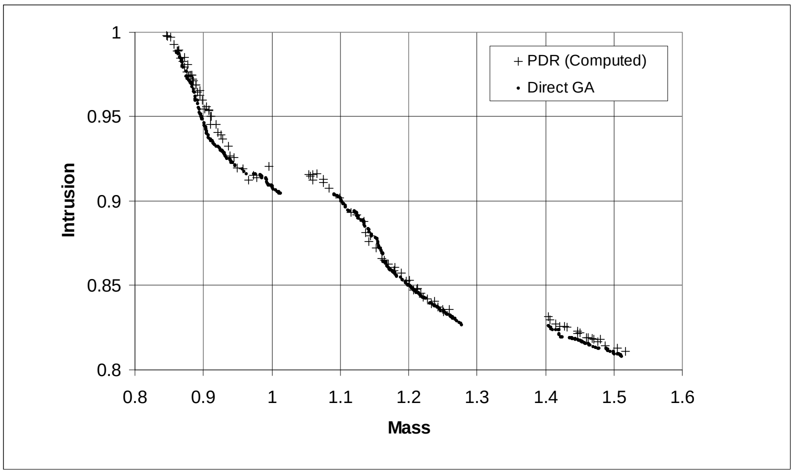

Figure 13: Pareto Optimal Frontier: Comparison of the *computed* POF (100 designs) obtained using the PDR approach with the reference POF computed using the Direct NSGA-II algorithm

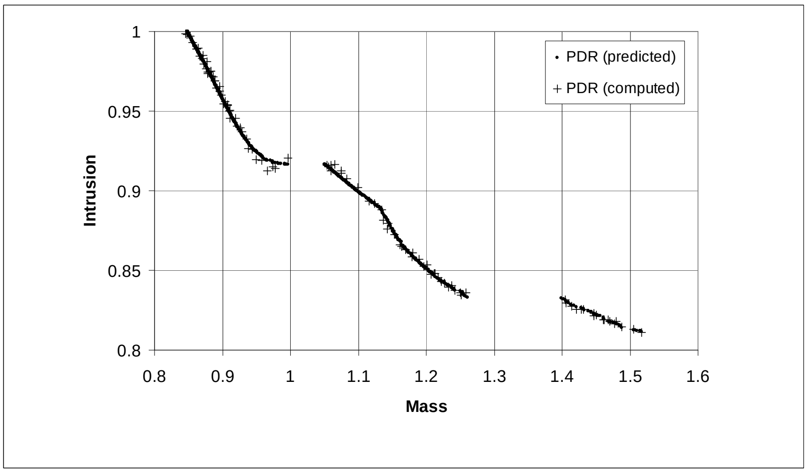

Figure 14: Pareto Optimal Frontier: Comparison of the PDR predicted POF and the PDR computed POF.

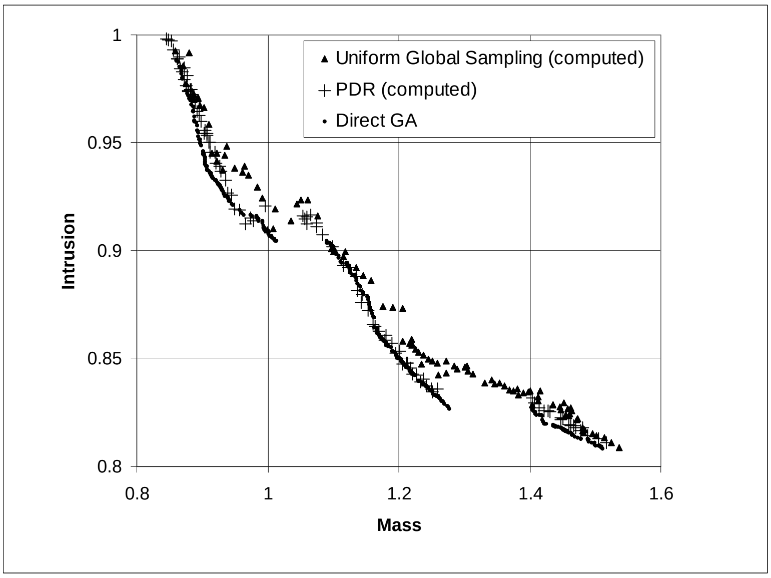

Figure 15: Pareto Optimal Front: Comparison of the simulation results for uniform global sampling (▲) and PDR sampling (+) with the reference solution obtained using the Direct NSGA-II algorithm (●).

6. Conclusions

This study presents an adaptive metamodel-based technique for multi-objective optimization. Although the set of test examples is sparse, the results seem to indicate that the method holds promise for being efficient as well as accurate. For a 7-variable problem only about 4% of the effort required by a direct method could provide a result of similar accuracy. For a 30-variable problem this number rose to about 15%. It is apparent that the larger the problem (in terms of the number of variables), the less efficient the PDR method becomes. Industrial crash problems typically require 30-50 design variables and in this range the PDR method should still be competitive. A significant advantage is that PDR devolves to SRSM for a single-objective optimization problem.

It should be noted that a large number of simulations per iteration was required for the analytical problems whereas the much smaller (in terms of the variables) crash application required significantly less. As a rule it is recommended to use $1.5n$ simulations per iteration (as was done for the crash example), but that number was obviously too limited for the 30-variable test problems. It is difficult to recommend a specific number of simulation points per iteration. This dilemma is not unique to PDR, but also appears in direct solution methods. For example the population size of a Genetic Algorithm such as NSGA-II is also typically problem dependent. As noted above, the author had to experiment with population size when applying the direct GA solver to the crash problem to produce an “exact” result. It is therefore recommended that a large array of test problems be tested as future research to determine a robust population size for PDR.

7. References

[1] Deb, K. Multi-Objective Optimization using Evolutionary Algorithms. Wiley. 2001.

[2] Stander, N., Craig, K.J. On the robustness of a simple domain reduction scheme for simulation-based optimization, *Engineering Computations*, 19(4), pp. 431-450, 2002.

[3] Stander, N., Roux, W.J., Basudhar, A., Eggleston, T., Goel, T., Craig, K.J. *LS-OPT User’s Manual Version 5.0*, April 2013.

[4] Knowles, J. ParEGO: A hybrid algorithm with on-line landscape approximation for expensive multiobjective optimization problems. Technical report TR-COMPSYSBIO-2004-01, September 2004.

[5] Fu, G., Khu, S.-T., Butler, D. Multiobjective optimization of urban wastewater systems using ParEGO: a comparison with NSGA II, Proceedings of the 11th Conference on Urban Drainage, Edinburgh, Scotland, UK, 2008.

[6] Hartman, E.J., Keeler, J.D., Kowalski, J.M. Layered neural networks with Gaussian hidden units as universal approximations. *Neural Computation,* 2(2), pp. 210-215, 1990.

[7] Park, J., Sandberg, I.W. Approximation and radial basis function networks. *Neural Computation,* 5(2), pp. 305-316, 1993.

[8] Morris, M. D. and Mitchell, T. J. Exploratory Designs for Computer Experiments. *J. Statist. Plann. Inference* 43, 381-402, 1995.

[9] Johnson, M.E., Moore, L.M., Ylvisaker, D. Minimax and Maximin distance designs. *J. Statist. Plann. Inference,* 26, 131-148, 1990.

[10] McKay, M.D., Conover, W.J., Beckman, R.J. A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics*, pp. 239-245, 1979.

[11] Ingber, L. Very fast simulated re-annealing, Mathematical Computer Modeling, 12, pp. 967-983,1989.

[12] Ingber, L., Adaptive simulated annealing (ASA), [ftp.alumni.caltech.edu: /pub/ingber/ASA.tar.gz], Lester Ingber Research, McLean VA, 1993.

[13] Craig K.J., Stander, N., Dooge, D., Varadappa, S. MDO of automotive vehicle for crashworthiness and NVH using response surface methods. Paper AIAA2002_5607, 9th AIAA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, 4-6 Sept 2002, Atlanta, GA.

[14] Hallquist, J.O. LS-DYNA User's Manual, Livermore Software Technology Corporation.

[15] While, L., Bradstreet, L. Barone, L., Hingston, P. Heuristics for Optimizing the calculation of hypervolume for multi-objective optimization problems. 2005 IEEE Congress on Evolutionary Computation, CEC 2005, Vol. 3, 2225-2232, IEEE, September 2005.

[16] Stander, N. An Efficient New Sequential Strategy for Multi-Objective Optimization using LS-OPT. *Proceedings of the 12th International LS-DYNA User's Conference*, Detroit, Michigan, June 3-5, 2012.