11th World Congress on Structural and Multidisciplinary Optimization
7th - 12th, June 2015, Sydney Australia

Many-objective Optimization in Engineering Design: Case Studies Using a Decomposition Based Evolutionary Algorithm

Hemant Kumar Singh, Tapabrata Ray

University of New South Wales, Canberra, Australia. Emails: {h.singh, t.ray}@adfa.edu.au

Abstract

Engineering design often involves simultaneous minimization/maximization of multiple conflicting objectives. The optimum solution of such problems comprises a set of designs representing best-tradeoff among objective values, known as Pareto optimal front (POF). It is well known that the existing multi-objective optimization algorithms can find POF for 2-3 objective problems successfully, but their performance deteriorates significantly for problems with 4 or more objectives, which are termed as "many-objective" optimization problems. There has been a significant recent interest in solving them. In this paper, we present a decomposition based approach for solving many-objective optimization problems. Further, we demonstrate that this improved capability can be exploited to solve various other intractable classes of problems. Two such classes presented are robust design optimization and "re-design" for robustness. In addition to the above, we also illustrate the benefits of multiobjective formulation for a special class of problems, where an user is interested in solving single objective optimization problems with different parameter values. We present numerical examples from various domains including mechanical, civil and aerospace industry to demonstrate the approaches and corresponding benefits.

1 Introduction

Optimization is an integral tool in engineering design used for pushing the boundaries of performance subject to several practical constraints. Multi-objective (MO) optimization algorithms are required where more than one conflicting objectives are simultaneously being optimized, for example, maximization of strength and minimization of weight of structural components, minimization of travel time and maximization of payload for space missions, maximization of engine efficiency and minimization of emissons etc. Optimum solutions of such problems is a set of designs representing best trade-off among the objectives, known as Pareto Optimal Front (POF). Population based methods such as Evolutionary Algorithms (EAs) [1] are commonly used to solve such problems, which "evolve" a population of solutions (designs) through a process similar to the Darwinian principle of natural selection. EAs simulate the evolution process through use of mathematical *crossover* and *mutation* (to generate new designs from "fitter parents"), *ranking* (to prioritize them based on performance) and *reduction* (to choose the designs to be retained for next generation). It is now well established in literature that the existing EAs can efficiently solve MO problems with 2-3 objectives but their performance deteriorates severely for problems with 4 or more objectives [2]. Such problems are referred to as "Many-objective" (MaO) optimization problems.

The poor performance of such algorithms for MaO problems is attributed to the ranking procedure based on Pareto-dominance. For MaO problems, most (or all) solutions in the population become *non-dominated* early in the search and hence there is no pressure to drive the solutions towards convergence. To deal with this challenge, a few different approaches have been reported in the literature. Some studies have focused on modifying the dominance relations [3] or adding a secondary ranking apart from Pareto dominance using various criteria such as sub-vector dominance, fuzzy dominance, ε-dominance etc. [4]. These methods may often result in reduced the diversity of solutions in POF in exchange for good convergence. Alternatively, the solutions can be ranked using an indicator such as hypervolume [5]. However, the calculation of hypervolume itself is computationally expensive. Interactive EAs [6, 7] have also been proposed, where a decision maker is asked to provide inputs/choices of preferred solutions during the search in an attempt to deliver designs in areas of interest. However, such user information may be hard to come by or at times misleading in absence of prior knowledge about the nature of the POF. For certain problems, reduction of original set of objectives to a manageable number has also been suggested [8]. Although each of these methods have their advantages and applicability, obtaining a well spread and converged POF for MaO problems still remains a challenge. In this regard, *decomposition* based methods [9, 10] have been most promising so far, which generate a set of diverse reference directions in the objective space and optimize linear combination of objectives along these directions to deliver diverse solutions on the POF.

It is to be noted that all the studies mentioned above have been presented in the context of deterministic optimization, *i.e.*, the original problem formulation itself had many objectives and no uncertainties in variables/objectives are considered. However, certain other useful classes of problems may also be modeled as MaO problems. For example, when a robust optimum is sought instead of deterministic optimum, the metrics for robustness could be added as additional objectives to the problem, resulting in an MaO problem. Similarly, one can also consider the problem of re-design for robustness of an existing product as an MaO problem. However, such extensions of many-objective optimization have only recently been proposed by the authors [11].

In this paper, an enhanced Decomposition Based Evolutionary Algorithm (DBEA) is presented for solving MaO problems in Section 2. Thereafter, its applications are demonstrated using three different classes of problems – deterministic, robust and re-design in Section 3. An additional special case is also illustrated in Section 3. Numerical experiments are presented on examples from civil, mechanical and aerospace engineering domain to highlight the benefits of the proposed approach. A summary of the work is given in Section 4.

2 Decomposition based Evolutionary Algorithm (DBEA)

The proposed algorithm is referred to as Decomposition Based Evolutionary Algorithm (DBEA). It generates a set of uniform reference direction through systematic sampling. The quality of solution a solution is measured by its distance from an associated direction and the distance from ideal point (in scaled objective space). While the first version of the algorithm [10] used a steady state model, it has been continually improved and the DBEA presented here uses generational model which is amenable to parallelization and has a number of enhancements over its first version. The pseudo-code of DBEA is presented in Algorithm 1 and the details of its key components are described below.

Generate: A structured set W reference points is generated spanning a hyperplane with unit intercepts in each objective axis using normal boundary intersection method (NBI) [12]. The approach generates W points on the hyperplane with a uniform spacing of $\delta = 1/s$ for any number of objectives M with s unique sampling locations along each objective axis. The reference directions are formed by constructing a straight line from the origin to each of these reference points. A population size equal to the number of reference points is used, generated using Latin Hypercube Sampling (LHS).

Algorithm 1 Decomposition Based Evolutionary Algorithm (DBEA)

Require: Gen_{max} (maximum number of generations), W (number of reference points), p_c (probability of crossover), p_m (probability of mutation), η_c (crossover index), η_m (mutation index)

i=1;
Generate W reference points using Normal Boundary Intersection (NBI)
Construct W reference directions; Straight lines joining origin and W reference points
Initialize the population using LHS sampling P^i; $|P^i| = W$
Evaluate P^i and compute the ideal point z^I and Nadir point z^N
Scale the individuals of P^i using z^I and z^N
Compute d_1 and d_2 for all individuals in P^i
Assign individuals of P^i to the reference directions;
while $(i \leq Gen_{max})$ **do**
    Create C offspring from P^i via recombination;
    Evaluate C and compute Ideal point (z^I) and Nadir point (z^N)
    Scale the individuals of P^i+C using z^I and z^N
    Compute d_1 and d_2 for all individuals in C+P^i
    Replace individuals in P^i using C
    P^{i+1} =P^i
    i=i+1
end while

Scale: The objective values are scaled between between 0 and 1. If any coordinate of the ideal point matches with the corresponding coordinate of the Nadir point, the scaled value of the corresponding objective is set to 0.

Compute: Two measures d_1 and d_2 are computed for all feasible solutions. The first measure d_1 is the Euclidean distance between origin and the foot of the normal drawn from the solution to the reference direction, while the second measure d_2 is the length of the normal. Mathematically, d_1 and d_2 are computed as $d_1 = \mathbf{w}^T \mathbf{f}'(\mathbf{x})$; $\quad d_2 = \|\mathbf{f}'(\mathbf{x}) - \mathbf{w}^T \mathbf{f}'(\mathbf{x})\mathbf{w}\|$, where $\mathbf{w}$ is a unit vector along any given reference direction. A value of $d_2 = 0$ implies that the solutions are perfectly aligned along the required reference directions ensuring good diversity, while a smaller value of d_1 indicates superior convergence.

Assign: If all solutions in the population are infeasible, solutions are randomly assigned to the reference directions. Since the population size is equal to the number of reference directions, every direction has an associated solution. If the population has one or more feasible solutions, the feasible set of solutions are randomly shuffled (to avoid any bias). Each solution is assigned to a reference direction for which its d_2 is minimum. The assigned direction is removed from the list of available reference directions and feasible solutions are sequentially assigned following the shuffled list. Subsequently, infeasible solutions are randomly assigned to the remaining directions.

Create: The process of offspring creation involves two steps, *i.e.*, identification of participating parents for recombination and the recombination process itself. Two participating parents are recombined using simulated binary crossover (SBX) and polynomial mutation (PM). The first offspring (out of two) is considered as a member of C. Thus to generate W offspring solutions, W participating parents and their corresponding mating partners need to be identified. The selection of participating parents and their corresponding mating partner depends on the state of the population. If all the solutions in the parent population P are infeasible, the participating parents are identified via binary tournament, where their fitness measure is based on their sum of constraint violation. If all the solutions in the population are feasible, the first set of participating parents are the members of P itself, while their partners are identified using a roulette wheel selection. The use of roulette wheel induces a stochastic selection pressure to prefer partners that are close to a reference direction. In the event there is a mix of feasible and infeasible solutions in the population, mating partner selection for a feasible individual follows the schemes described above. To identify the rest of participating parents and their partners, the following strategy is adopted. The infeasible solutions in the population are first sorted based on their sum of constraint violation values and assigned to the set of weight directions (take note that these weight directions had infeasible solutions associated with them and this process alters the previous assignment). Following that, binary tournaments are conducted among all individuals in the population to identify the remaining participating parents and their mating partners. Such a process encourages the recombination between feasible and infeasible solutions.
Replace: The above **Create** process will result in C, *i.e.*, a set of W offspring. Since the individuals in P are already assigned to reference directions, such a process essentially looks through the list of C to identify potential candidates for replacement. In the event there are infeasible solutions in P, a sequential replacement scheme is adopted, wherein a solution from C with better fitness replaces the worst infeasible individual in P. Such a process will continue till all members of P become feasible. In the event all individuals in P are feasible, a sequential replacement scheme is once again adopted. In such a process, an individual from C replaces a dominated individual of P. In the event the offspring solution fails to dominate any individual in P, the replacement is carried out based on their fitness computed using the d_2 measure (smaller d_2 preferred). The process also adopts a single replacement policy, wherein an offspring can only replace one individual of P.

The constraint handling approach used in this work is based an on epsilon level comparison and has been reported earlier in [10]. The process adaptively controls an allowable constraint violation measure, which offers the marginally infeasible solutions to be selected as opposed to a feasibility first principle.

3 Numerical Experiments

In this section, numerical experiments are conducted using the proposed DBEA on a number of engineering problems. As mentioned previously, three different kind of problems are considered, namely deterministic, robust and re-design problems; along with an additional special case. These experiments are discussed next. For each of the experiments, the parameters chosen are: $Gen_{max} = 100$, $p_c = 1$, $p_m = 0.1$, $\eta_c = 30$, $\eta_m = 20$.

3.1 Deterministic optimization

For deterministic MaO optimization, two problems are chosen, which are discussed below.

The **water resource management (WRM)** problem was proposed in [13] in the context of urban planning and has emerged from the environmental engineering domain. The problem has three design variables: local detention storage capacity , maximum treatment rate and the maximum allowable overflow rate. The objective functions to be minimized are f_1 = drainage network cost, f_2 = storage facility cost, f_3 = treatment facility cost, f_4 = expected flood damage cost, and f_5 = expected economic loss due to flood.

The **general aviation aircraft (GAA)** is an example from aerospace engineering, first introduced by Simpson *et al.* [14]. The problem involves 9 design variables: cruise speed, aspect ratio, sweep angle, propeller diameter, wing loading, engine activity factor, seat width, tail length/ diameter ratio and taper ratio and the aim is to optimize 10 objectives: Minimize f_1 =takeoff noise, f_2 =empty weight, f_3 =direct operating cost, f_4 =ride roughness, f_5 =fuel weight, f_6 =purchase price, f_7 =product family dissimilarity and maximize f_8 = the flight range, f_9 =lift/drag ratio and f_{10} =cruise speed.

Thirty independent runs are conducted on the above two problems. Population sizes chosen for WRM and GAA are 210 and 715, respectively. Two standard metrics are used for measuring the quality of non-dominated set of solutions obtained by the algorithm, namely *Hypervolume* and *Inverse Generational Distance (IGD)*. Both of these measures provide a quantification of overall convergence *and* diversity of the solutions obtained and are hence commonly used for benchmarking MO algorithms [15]. A high value of hypervolume and low value of IGD indicates better quality of the POF obtained by an algorithm.

The results are compared with a well established decomposition based algorithm called MOEA/D (Multiobjective Evolutionary Algorithm Based on Decomposition) [9], shown in Table 1. It is observed that the best, mean and median metric values (both Hypervolume and IGD) across 30 runs achieved using DBEA are better than those obtained using MOEA/D. Furthermore, the standard deviations (Std.) obtained using DBEA is also lower than MOEA/D for all cases except Hypervolume for GAA, which reflects its consistency in obtaining the competitive results. For the case of Hypervolume of GAA, the reason for DBEA's standard deviation being higher is that the Hypervolume values themselves are significantly higher than those from MOEA/D. In terms of magnitude of Std. relative to mean values, DBEA performs much better.

Table 1: Performance metrics values for deterministic problems across 30 runs

		Water resource management				
Algorithm	Metric	Best	Mean	Median	Worst	Std
DBEA	Hypervolume	**0.43913**	**0.43628**	**0.43613**	0.43378	0.00131
MOEA/D		0.35660	0.29864	0.29803	0.22559	0.03601
DBEA	IGD	**0.08324**	**0.09417**	**0.09084**	0.11909	0.01038
MOEA/D		0.07773	0.13413	0.13386	0.18848	0.02925
		General aviation aircraft				
Algorithm	Metric	Best	Mean	Median	Worst	Std
DBEA	Hypervolume	**0.03460**	**0.02583**	**0.02582**	0.01647	0.00450
MOEA/D		0.01011	0.00303	0.002722	0.00094	0.00180
DBEA	IGD	**0.28656**	**0.30918**	**0.30514**	0.38737	0.02019
MOEA/D		0.30685	0.41841	0.41088	0.51788	0.05346

3.2 Robust design optimization

In the previous section (and in most literature), the optimization problems are considered *deterministic*, *i.e.*, it is assumed that performance estimates are exact and variable values can be achieved to an arbitrary precision. However, in practice, most engineering designs operate under uncertainties that may emerge from varying ambient conditions, material imperfections, inaccuracies in analyses/simulations, manufacturing precision etc. This work considers uncertainties in design variables only. For practical implementation, designs need to be robust, *i.e.*, less sensitive in the presence of uncertainties.

To obtain robust solutions, the deterministic problem needs to be transformed to a robust formulation by using *expected* values for performance and constraints instead of deterministic values. Two kinds of robustness measures need to be considered under uncertainty: *feasibility robustness* (the design should not become infeasible) and *performance robustness* (the performance should not deteriorate). They are quantified in terms of *sigma* levels, a terminology used in industry to judge the reliability of design [11]. Most of the past studies have either considered these measures as *constraints* rather than *objectives*, or have only considered one of these robustness measures instead of both. This has been partially motivated by lack of good techniques for solving MaO problems, because even if the original problem has only 2-3 objectives, the reformulated robust problem will become a MaO problem.

Using the robustness as constraints requires one to set the level of desired robustness level (say $> 4\sigma$) before optimization, which delivers only a partial solution set for the problems. Furthermore, it is not possible to predict the robustness levels expected from the optimization exercise without a priori knowledge about optimum solutions, in which case the optimization may not return any solutions (as there may not exist any 4σ designs). Only by considering the robustness measures as objectives can one obtain a full range of solutions with all possible trade-offs among the performance objectives, feasibility robustness and performance robustness. Towards this goal, in this paper, the deterministic problem is reformulated as an MaO problem as shown in Equation 1. This formulation is referred to as Feasibility and Performance Robust (FPR) formulation. For illustration of the concept, a mechanical design example from automotive industry is considered next.

$$
\begin{aligned}
\text{Minimize:}\quad & f_i'(\mathbf{x}) = \mu_{f_i(\mathbf{x})}, i = 1,2,......M \qquad \{\textit{expected} \text{ performance objectives; } M\text{= no. of objectives}\}\\
\text{Maximize:}\quad & f_{M+1}'(\mathbf{x}) = \text{Min}(sigma_g, R_c) \qquad \{\text{feasibility robustness}\}\\
\text{Maximize:}\quad & f_{M+2}'(\mathbf{x}) = \text{Min}(sigma_f, R_f) \qquad \{\text{performance robustness}\}\\
\text{Subject to:}\quad & sigma_g \geq 0\\
\text{where:}\quad & sigma_g \equiv \text{Min}(\mu_{g_j(\mathbf{x})}/\sigma_{g_j(\mathbf{x})}), j = 1\ldots K \text{ (no. of constraints)}\\
& sigma_f \equiv \text{Min}(\sigma_{0,j}/\sigma_{f_j(\mathbf{x})}), j = 1\ldots M \text{ (no. of objectives)}\\
& \sigma_{0,j}, j = 1\ldots M \text{ are allowable performance variations in each objective}
\end{aligned}
\tag{1}
$$

Vehicle Crash worthiness Optimization Problem (VCOP): VCOP has been studied in various works including Sun *et al.* [16]. A modified bi-objective formulation of the problem is studied in this paper which seeks to maximize the post-impact energy absorption (U) of the vehicle structure and aims to minimize the structural weight (M_s), subject to the constraint on peak deceleration (a). The calculations for U, M and a are done as suggested in [16]. A higher energy absorption lowers the risk to the occupants of the car. However, increase in energy absorption often leads to unwanted increase in the structural weight. To limit impact severity, a constraint on maximum deceleration is imposed in this formulation which is assumed to be 40 g (g = 9.81 m/s^2). Part thicknesses of three key members – inner rail, outer rail and the cradle rails – of the vehicle front end structure have been chosen as design variables for the crash worthiness optimization problem. For each variable, the uncertainty is assumed to follow a Gaussian distribution with $\sigma = 0.05$. For each objective, maximum allowable $\sigma_{f,0}$ is set as 2.1. The results obtained using DBEA on the problem are shown in Figure 1. It can be clearly seen that the delivered set of designs span a range of sigma levels achievable, between 0 and 6σ. This gives the user a choice to select the design which is most suitable for the application.

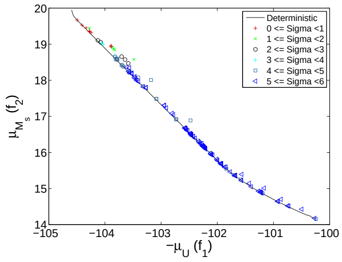
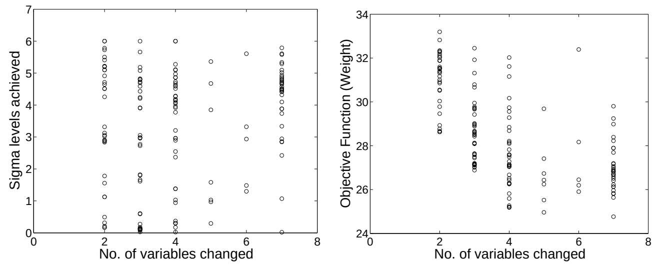


Figure 1: Robust solutions obtained for VCOP using DBEA. Here, Sigma is overall, *i.e.*, both feasibility and performance robustness achieve the indicated level

3.3 Re-design for robustness

Next, the applicability of presented DBEA is considered for "re-design for robustness (RDR)". RDR is an endeavor to improve the robustness of an *existing* design by doing minimal changes to it. This need could arise, for example, in case operating conditions of the product have changed due to unforeseen circumstances. In such situations, instead of discarding all inventory of components for the product and re-designing it from scratch, it would be of great value to investigate whether only a few components could be replaced for making the product robust. In order to solve this problem, apart from the objectives considered in previous subsection, one more objective, f_{M+3}, is added for minimization, which is the number of components different from the base design. This formulation is referred to as FPRR formulation (FPR for Re-design), and contains $M+3$ objectives, where M is the number of objectives in the deterministic formulation. To evaluate the objective f_{M+3}, a binary string is maintained in which 0 indicates the variable value is same as base design, whereas 1 indicates it's different. This binary string is also evolved along with the variable vectors. For illustration of the concept, a mechanical design example from automotive industry is considered next where the objective is to reduce the weight of a car subject to constraints on satisfaction of several safety criteria in the event of side impact [17].



(a) Sigma levels achieved (b) Objective values (Weight)

Figure 2: Solutions obtained for FPRR formulation of CSIP using DBEA.

Car Side Impact Problem (CSIP): In CSIP, the uncertainties associated with 7 (out of 11) variables are modeled as Gaussian distribution with $\sigma_{x_i} = 0.0408, i = 1, 2 \ldots 7$, and $\sigma_{f,0}$ is set as 2.5. The base design is taken from [18] as $\{1.0,1.0,1.0,1.0,1.0,1.0,1.0,0.345,0.192,0,0\}$, with a weight of 29.05 units; whereas the deterministic optimum for the problem is reported to be 23.59. The base design is actually *infeasible*, violating two of the constraints (lower rib deflection and pubic force criteria) marginally. The results obtained using DBEA on the FPRR formulation of CSIP are shown in Figure 2. It can be seen that by merely changing 2 variables out of 7, one can achieve *up to* 6σ designs, at the cost of corresponding weight values going up to 33.2 units. As the number of changed components increase, the same can be achieved with lower compromises in weight values.

3.4 Special class of single objective optimization problems with varying parameters

Lastly, we consider a special case where the original problem contains only single objective (SO), but this necessitates it being solved for several parameter settings according to the specific design case. Instead, it can be solved as multi-objective problem to get entire range of solutions for all parameter settings in one run. This is illustrated using a reinforced beam design example.

The **reinforced concrete beam design problem (RCB)** has been studied as a constrained, single objective optimization problem in [19]. The problem was also solved for various values of a parameter a_6 (minimum allowable width)[19].

Instead of independent runs of SO optimization with different parameter values, we reformulate the problem and solve it as a bi-objective optimization problem, *i.e.*, minimize $-a_6$ (minimum allowable width) and minimize total cost. The results using DBEA for a single run are presented in Figure 3 for various values of the minimum allowable width parameter. The values are in agreement with those reported in [19] for minimum allowable width of 30. While most studies focus on the ability of MO formulations to deliver a set of trade-off solutions for problems involving conflicting set of objectives, the principle presented here can be exploited for problems such as above to deliver solutions spanning the region of interest (different parameter values) when objectives are not conflicting.

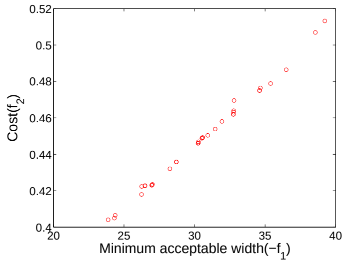


Figure 3: Optimal designs for RCB problem

4 Summary

In this paper, we have introduced an improved generational form of a decomposition based evolutionary algorithm for the solution of optimization problems involving many objectives. Apart from its ability to efficiently solve deterministic MaO problems, we have highlighted its extended use for two other important class of problems i.e. robust optimization and re-design for robustness. Numerical examples have been selected from various domains (mechanical, civil and aerospace) to highlight the benefits of proposed approach.

References

[1] Deb, K., 2001. *Multi-objective optimization using evolutionary algorithms*. John Wiley & Sons.

[2] Ishibuchi, H., Tsukamoto, N., and Nojima, Y., 2008. "Evolutionary many-objective optimization: A short review". In Proc. IEEE World Congress Computational Intelligence, pp. 2419–2426.

[3] Sato, H., Aguirre, H., and Tanaka, K., 2007. "Controlling dominance area of solutions and its impact on the performance of moeas". In Lecture notes in Computer Science 4403: Evolutionary Multi-Criterion Optimization-EMO, Springer-Verlag, Berlin, pp. 5–20.

[4] Koppen, M., and Yoshida, K., 2007. "Substitute distance assignments in NSGA-II for handling many-objective optimization problems". In Lecture notes in Computer Science 4403: Evolutionary Multi-Criterion Optimization, Springer Berlin/Heidelberg, pp. 727–741.

[5] Zitzler, E., and Kunzli, S., 2004. "Indicator-based selection in multiobjective search". In Lecture notes in Computer Science 3242: Parallel Problem Solving from Nature - PPSN VIII, Springer Berlin/Heidelberg, pp. 832–842.

[6] Deb, K., Sinha, A., Korhonen, P., and Wallenius, J., 2010. "An interactive evolutionary multiobjective optimization method based on progressively approximated value functions". *Evolutionary Computation, IEEE Transactions on,* **14**(5), Oct, pp. 723–739.

[7] Thiele, L., Miettinen, K., Korhonen, P. J., and Molina, J., 2009. "A preference-based evolutionary algorithm for multi-objective optimization". *Evolutionary Computation,* **17**(3), pp. 411–436.

[8] Singh, H. K., Isaacs, A., and Ray, T., 2011. "A Pareto corner search evolutionary algorithm and dimensionality reduction in many-objective optimization problems". *IEEE Transactions on Evolutionary Computation,* **15**(4), pp. 539–556.

[9] Zhang, Q., and Li, H., 2007. "MOEA/D: A multiobjective evolutionary algorithm based on decomposition". *IEEE Transactions on Evolutionary Computation,* **11**, pp. 712–731.

[10] Asafuddoula, M., Ray, T., and Sarker, R., 2013. "A decomposition based evolutionary algorithm for many objective optimization with systematic sampling and adaptive epsilon control". In Evolutionary Multi-Criterion Optimization, R. Purshouse, ed., Vol. 7811 of *Lecture Notes in Computer Science*, Springer Berlin Heidelberg, pp. 413–427.

[11] Asafuddoula, M., Singh, H., and Ray, T., 2014. "Six-sigma robust design optimization using a many-objective decomposition based evolutionary algorithm". *IEEE Transactions on Evolutionary Computation*(99). Available online.

[12] Das., I., and Dennis, J. E., August 1998. "Normal-bounday intersection: A new method for generating Pareto optimal points in multicriteria optimization problems". *SIAM J. Optim.,* **8**(3), pp. 631–657.

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

[14] Simpson, T. W., Chen, W., Allen, J. K., and Mistree, F., 1996. "Conceptual design of a family of products through the use of the robust concept exploration method". *AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization,* **2**, pp. 1535–1545.

[15] Singh, H., 2011. "Development of optimization methods to deal with current challenges in engineering design optimization". PhD thesis, School of Engineering and Information Technology, University of New South Wales Canberra, Australia.

[16] Sun, G., Li, G., Zhou, S., Li, H., Hou, S., and Li, Q., 2011. "Crashworthiness design of vehicle by using multiobjective robust optimization". *Structural and Multidisciplinary Optimization,* **44**(1), pp. 99–110.

[17] Saxena, D. K., and Deb, K., 2007. "Trading on infeasibility by exploiting constraints criticality through multi-objectivization: A system design perspective". In Evolutionary Computation, 2007. CEC 2007. IEEE Congress on, IEEE, pp. 919–926.

[18] Gu, L., Yang, R., Tho, C., Makowskit, M., Faruquet, O., and Y. Li, Y. L., 2001. "Optimisation and robustness for crashworthiness of side impact". *International Journal of Vehicle Design,* **26**(4), pp. 348–360.

[19] Coello, C. C., Christiansen, A. D., and Hernandez, F. S., 1997. "A simple genetic algorithm for the design of reinforced concrete beams". *Engineering with Computers,* **13**(4), pp. 185–196.