I-MODE: An Interactive Multi-Objective Optimization and Decision-Making using Evolutionary Methods

Kalyanmoy Deb and Shamik Chaudhuri

Kanpur Genetic Algorithms Laboratory (KanGAL)
Indian Institute of Technology, Kanpur
Kanpur, PIN 208016, India
{deb,shamik}@iitk.ac.in

KanGAL Report Number 2007003

Abstract. With the popularity of efficient multi-objective evolutionary optimization (EMO) techniques and the need for such problem-solving activities in practice, EMO methodologies and EMO research and application have received a great deal of attention in the recent past. The first decade of research in EMO area has been spent on developing efficient algorithms for finding a well-converged and well-distributed set of Pareto-optimal solutions, although EMO researchers were always aware of the importance of procedures which would help choose one particular solution from the Pareto-optimal set for implementation. In this paper, we address this long-standing issue and suggest an interactive EMO procedure by collating most salient research in EMO and putting together a step-by-step EMO and decision-making procedure. The idea is implemented in a GUI-based, user-friendly software which allows a user to supply the problem mathematically or by using user-defined macros and enables the user to evaluate solutions directly or by calling an executable software, such as popularly-used MATLAB software for a local search or ANSYS software for finite element analysis, etc. Starting with standard EMO applications, continuing to finding robust, partial, and user-defined preferred frontiers through standard MCDM procedures, the well-coordinated software allows the user to first have an idea of the complete trade-off frontier, then systematically focus in preferred regions, and finally choose a single solution for implementation.

1 Introduction

In the past decade of research and application activities of evolutionary multi-criterion optimization (EMO), major focus has been made in finding a set of trade-off solutions, representing the entire Pareto-optimal front. Although these efforts were the first steps in evaluating the potential of EMO methodologies as a true multi-objective optimizer, it is now time to address an equally important matter of choosing a single solution from the Pareto-optimal front for implementation. Such a task should involve a decision-making activity in which higher-level information must be provided by the decision-maker. It is obvious to realize that such a decision-making activity is subjective and must depend on the problem being solved. Thus, any effort in this direction must be spent on devising a procedure which will help a decision-maker (DM) to arrive at a solution of his/her choice, rather than one which will recommend a solution automatically. The multi-criterion decision-making (MCDM) approaches address a similar issue and some MCDM ideas can be borrowed to address the decision-making issue in an EMO study. Besides the higher-level decision-making approaches, there are some other more direct decisions which most decision-makers may like to follow. Some such decision-making ideas may include (i) preference of a robust frontier, instead of a Pareto-optimal frontier, (ii) preference of locally-optimal solutions obtained from EMO solutions, instead of simply choosing the EMO solutions, preference of knee solutions and preference of some specific regions detected by various means, instead of the entire trade-off Pareto-optimal frontier.

In this paper, we give shape to an earlier proposal by the authors [6] in combining EMO procedures with a number of direct (less subjective) decision-making tools and a number of higher-level (subjective) decision-making tools with a procedure which can go back and forth between many such tools and an EMO procedure. The main motivation behind such a repetitive procedure is that often the choice of a higher-level decision-making tool or fixation of parameter values associated with such a tool cannot be done a priori. When an idea of the entire trade-off frontier is obtained, a decision-making tool with all its associated parameters can be chosen adequately. The decision-making task is subjective to the DM and the final outcome of such a task will be dependent on the desires of the DM. To make the task of decision-making easier and possible, we also develop a GUI-based software (currently developed for a linux operating system) with visualization tools. Starting with a set of trade-off solutions, the developed I-MODE software will allow a decision-maker to finally choose a single preferred solution by performing a number of decision-making tasks. Currently, the procedure can be used for any number of objectives, but the software is restricted to a maximum of three objectives due to lack of suitable efficient visualization procedures. The working of the procedure is demonstrated on a welded-beam design problem having two objectives. The proposed methodology is one of many possible implementations of hybrid EMO and decision-making tools.

2 Existing Methodologies for Hybrid Multi-Objective Optimization and Decision-Making

There exist different interactive multi-objective optimization methods in the literature based on the classical optimization methods. Some popular methodologies, as described in [12] are as follows: Interactive Surrogate Worth Trade-off (ISWT) method [2], Reference point method [15], NIMBUS approach [12] etc. Each method is different from each other, but uses a single solution in each iteration. A guess solution is usually modified to another solution iteratively and by gathering some information from a DM. Since a single solution is used in an iteration, the DM only can find local information (such as a local trade-off or search direction) and cannot make a decision using a more global picture of the true Pareto-optimal front. However, in the context of an EMO, there do not exist many interactive studies. Tan and his students developed a GUI-based MOEA toolbox for multi-objective optimization [14]. The toolbox was designed with some classical decision-making aides, such as goal and priority settings. But a clear procedure of arriving at a single preferred solution was not present in the toolbox. Fonseca and Fleming [11] devised a GUI-based procedure which allowed some target values to be set for each objective and the trade-off objective information of different solutions found using an EMO procedure was demonstrated. However, the procedure lacked any quantitative statistical analysis of the solutions and also clearly did not provide any indication of the location of chosen solutions vis-a-vis the Pareto-optimal front. Another interactive GUI-based EMO software was *Guimoo*, developed by INRIA, but it lacked any decision-making facility.

3 Interactive Multi-objective Optimization and Decision-making using Evolutionary Methods (I-MODE)

In the proposed interactive EMO procedure, we attempt to put together some recent salient research results of EMO (described below) along with salient decision-making principles to constitute, for the first time, a hybrid interactive multi-criterion decision-making procedure. The existing EMO procedures used in I-MODE are as follows:

1. An EMO which is capable of finding the entire or a partial Pareto-optimal set, as desired [3, 1].
2. An EMO which capable of finding a preferred region of interest on the Pareto-optimal frontier using the reference point approach [10].
3. An EMO which is capable of finding a *robust* frontier [8], instead of Pareto-optimal frontier.
4. An EMO with a local search procedure which provides a better convergence properties [9, 5].
5. An EMO which is capable of handling multiple disconnected objective regions and constitute a parallel search.

I-MODE also uses the following single-objective optimization procedures, mostly for the purpose of verifying the multi-objective trade-off frontier obtained by an EMO:

1. A procedure for finding individual optimal solution(s) corresponding to each objective function subject to satisfaction of all supplied constraints [12].
2. The ϵ-constraint method of finding a single Pareto-optimal solution [12].

3. The multi-objective version of the ϵ-constraint method in which any number of original objectives can be kept as objectives and remaining original objectives can be constrained to some ϵ values. This procedure is expected to find a lower-dimensional Pareto-optimal front which would be a subset of the high-dimensional Pareto-optimal front.

Finally, for the decision-making purpose, we have borrowed a number of MCDM methodologies:

1. Tchebycheff methods with different "norms",
2. Reference point method [15],
3. Utility function method including weighted-sum approach and pseudo-weight method,
4. Surrogate worth trade-off method [12].

Using above procedures, we have designed an interactive procedure which allow a systematic procedure of performing any of the above tasks alone, in combination with each other or in sequence to each other in a manner which provides adequate flexibility to a decision-maker. We present the procedure in the step-by-step format. The parameters which are expected to be supplied by the decision-maker (DM) are mentioned in parenthesis.

Step 1: Obtain an approximate non-dominated front with following options:
- 1.1 Compute the complete front (DM: no parameter)
- 1.2 Compute a partial front (DM: limiting trade-off values)
- 1.3 Compute Pareto-optimal solutions near the reference points only (DM: reference point and limiting spread parameter)
- 1.4 Compute the *robust* Pareto-frontier (DM: robustness parameters)

Outcome: An approximate trade-off frontier

Step 2: Improve the obtained non-dominated front using other optimization methods:
- 2.1 Single-objective local searches from selected solutions:
  - 2.1.1 Automated selection: Clustering (DM: number of desired solutions)
  - 2.1.2 User-defined selection: (i) Weighted-sum approach (DM: weight vectors), (ii) Utility function based approach (DM: utility functions), (iii) Tchebycheff function approach (DM: ideal points and L_p norm), (iv) Using trade-off information between objectives (DM: Trade-off values).
- 2.2 Obtain a better trade-off frontier with specific solutions obtained using ϵ-constraint method (DM: ϵ values)

Outcome: A near-optimal and well-distributed trade-off frontier

Step 3: Verify obtained front with other optimization tasks:
- 3.1 ϵ-constraint method (single or multi-objective) (DM: ϵ-vector)
- 3.2 Optimization of individual objectives

Outcome: A verified (and confident) trade-off frontier

Step 4: Make decisions and choose regions of interest using one or more of the following methods:

4.1 Weighted-sum approach (DM: weight vectors)
4.2 Utility function based approach (DM: utility functions)
4.3 Tchebycheff function approach (DM: ideal points and L_p norm)
4.4 Using trade-off information between objectives (DM: Trade-off values)
4.5 Checking robustness of solutions (DM: robustness parameters)
Outcome: One or more regions of interest identified

Step 5: Until satisfied, go to Step 1 and focus further study in the above regions of interest, else declare the chosen solution(s)

3.1 Description of the I-MODE Procedure

The main difference between our proposed approach and the existing classical interactive methods described in [12] is that in our approach, we first attempt to find and show the DM the extent and shape of the Pareto-optimal front using a few representative solutions. This procedure, in addition to providing estimated ideal and nadir points of the problem, will also paint a good picture in the mind of the DM about the shape of the Pareto-optimal frontier which will help the DM later to concentrate on a particular region on the front. However, if the DM, for some reason, is interested in focusing on a particular region on the frontier, such information regarding his/her preference can be provided.

Thus, in the very first step of the I-MODE, in most situations, the DM applies an EMO (NSGA-II, an efficient multi-objective optimizer [4], is used here) on the problem to obtain a non-dominated front (Step 1). The EMO algorithm can start with two types of initial population. If some problem information is available then a biased population honoring the problem information can be generated, otherwise a completely random set of solutions can be chosen. Without any preference to any particular region on the trade-off frontier, the DM can find a representative set of solutions on the entire Pareto-optimal frontier. If, however, the DM is interested in a portion of the entire frontier, a number of options are available. A useful procedure would be to suggest a surrogate worth trade-off information (such as a 100% sacrifice in one objective must bring in at least a ζ% gain in another objective and so on) and find a partial frontier using the guided-domination based EMO [1]. Another useful way to find a biased set of trade-off optimal solutions is to use a number of reference points (or aspiration points) and use I-MODE to find optimal solutions close to these reference points [10]. This way, the DM gets to know Pareto-optimal solutions which are near his/her chosen reference points and are not on the entire Pareto-optimal frontier. In practice, solutions are only possible to be implemented with a finite precision. If this uncertainty in decision variables cause the objective and constraint function values to change by a large amount, the solution is declared as a non-robust solution. As another alternative, right in the beginning, the DM can opt for finding *robust* solutions which are less sensitive to parameter perturbations using a robust EMO procedure [8]. To find a robust frontier (which would be, in general, different from the original Pareto-optimal frontier), the user needs to define a range of likely perturbation in decision variables or parameters and an allowable change in functions (called as the robustness parameter) for defining a robust solution. The DM chooses one of the two robust optimization procedures described in [8] and can find the corresponding robust frontier.

Once the preliminary front is established through Step 1, the next step is to improve the obtained frontier by means of other optimization concepts. This step is necessary simply because evolutionary algorithms do not have a mathematical convergence proof for any arbitrary problem and certain portion of the frontier may not converge close to the true optimal frontier. One of the ways to have confidence about the optimality of obtained solutions is to try various optimization concepts and check to see if no further improvement on the obtained solutions are possible by various optimization runs. We attempt to improve the solutions using a local search procedure. For this task, a few solutions are picked from the non-dominated set and an individual local search is initiated from each of these solutions. For each solution, a combined single objective is constructed by computing a normalized pseudo-weight vector based on the location of the solution in the Pareto-optimal frontier [3, 7]. After several local searches are performed, the new non-dominated frontier is constructed. The solutions chosen for a local search can be randomly picked from the obtained set in Step 1 or some preferred solutions picked using some decision-making tools. In some occasions, there may exist wide gaps in the frontier obtained after the local searches and new single-objective ϵ-constraint procedures can be initiated with ϵ vectors chosen inside such gaps to find a number of representative solutions there. At the end of Step 2, the DM expects to come up with a well-converged and well-distributed set of trade-off solutions.

The next step (Step 3) is to verify the obtained frontier by a number of single-objective optimizations. The extreme solutions of the non-dominated front can be verified by using a single-objective genetic algorithm on each objective independently. The intermediate trade-off solutions can be verified by using the ϵ-constraint method in which only j $(\in [1, M])$ objectives can be kept as objectives and the remaining $(M - j)$ objectives can be converted into constraints [2]. If $j = 1$ is chosen, a single-objective optimization and if $j > 1$ is chosen, a multi-objective optimization procedure can be used to find one or more optimal solutions. These solutions should theoretically fall on the trade-off frontier obtained by I-MODE.

After obtaining the non-dominated frontier in Step 1, improving it through local searches in Step 2 and verifying the frontier through several other optimizations in Step 3, the DM is confident enough on the near-optimality of the obtained frontier and is ready to perform some decision-making tasks by getting an idea of the range of trade-off objective values. In Step 4, the DM can use a number of decision-making tools to concentrate one or more regions of preference by analyzing different regions of the trade-off frontier. For this purpose, pseudo-weight selection, Tchebycheff metric method with different norms, reference point method, surrogate worth trade-off method, etc. can be used depending on the appropriateness of the procedure to a particular problem. The DM can simply decide to choose the robust solutions from the frontier. In the case of a systematic evaluation of the frontier, multiple regions of interest can be selected

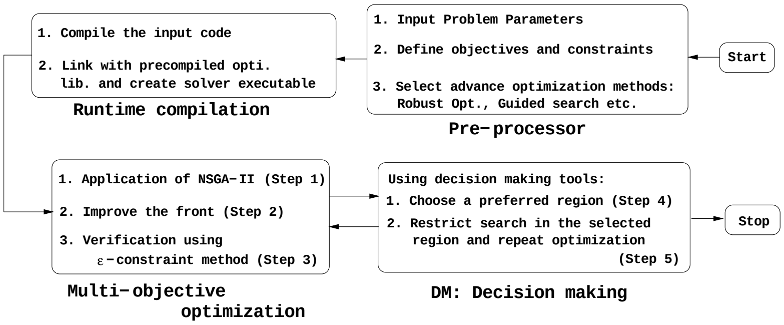
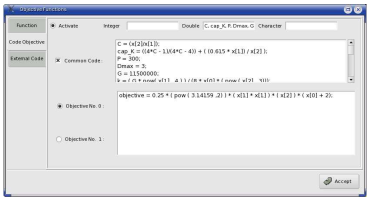

1. Compile the input code
2. Link with precompiled opti. lib. and create solver executable
Runtime compilation
1. Input Problem Parameters
2. Define objectives and constraints
3. Select advance optimization methods: Robust Opt., Guided search etc.
Pre-processor
Start
1. Application of NSGA-II (Step 1)
2. Improve the front (Step 2)
3. Verification using ε-constraint method (Step 3)
Multi-objective optimization
Using decision making tools:
1. Choose a preferred region (Step 4)
2. Restrict search in the selected region and repeat optimization (Step 5)
DM: Decision making
Stop

Fig. 1. Structure of I-MODE software.

simultaneously for further investigation. A completely new multi-objective optimization run with or without considering all the above-discussed steps (robust optimization, local searches, decision-making etc.) can be repeated to find more trade-off solutions in the chosen regions of interest and to help choose one or more subregions for further investigation. This procedure can be continued till the DM is satisfied (in Step 5) with a preferred solution.

3.2 I-MODE Software Implementation

I-MODE procedure is extensively user-dependent, where the DM has to interact with the software frequently for an effective run. To carry out such a rigorous interactive activity, we need a software with a powerful GUI, through which the DM can specify his/her preferences. At the time of development of the above features, we kept the GUI simple but effective for the decision-maker. The whole software is developed using C-language on the Linux platform and GUI is developed using GTK toolkit. This provides a robust structure of the code which can handle a large optimization problem where the memory requirement may be high. The I-MODE software has three broad modules, namely the pre-processor module, the optimization module, and the decision-making module (Figure 1).

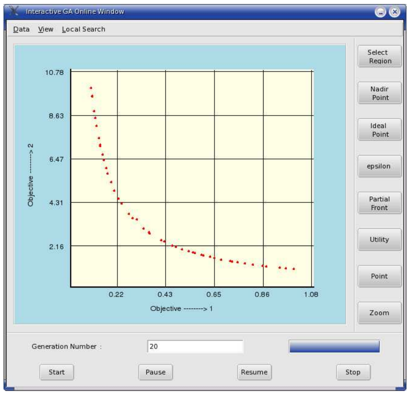


Double: C, cap_K, P, Dmax, G

```
C = (x[2]/x[1]);
cap_K = ((4*C - 1)/(4*C - 4)) + ( (0.615 * x[1]) / x[2] );
P = 300;
Dmax = 3;
G = 11500000;
k = ( G * pow( x[1] , 4 ) ) / (8 * x[0] * ( pow ( x[2] , 3)));
```

```
objective = 0.25 * ( pow ( 3.14159 ,2) ) * ( x[1] * x[1] ) * ( x[2] ) * ( x[0] + 2);
```

Fig. 2. Window for coding objective functions.



Generation Number : 20

Fig. 3. Plot window for two-objective problem.

In the pre-processor module, the DM specifies the optimization problem by setting the number of objectives, variables, and constraints. The DM also codes the objective function in a GUI window (Figure 2) once for any subsequent operations or can supply the program through a C-code or can be linked with an external evaluation software, such as a finite element software, MATLAB or others. Different GA parameters are also specified in the preprocessor module. The next module is the optimization module in which the DM can execute various optimization runs (Steps 1, 2 and 3 of the I-MODE procedure). Finally in the decision-making module, the DM uses different decision-making tasks (Step 4) to choose preferred solutions or regions. Figure 3 shows the online visualization window where the DM can observe the real-time animation of the optimization run. On this window, several `Menu buttons` are available, such as `Point Menu`, `Utility Menu`, ϵ`-constraint Menu`, `Ideal and Nadir point Menu`, and `Select region Menu`. The DM can choose one or more such menus and proceed with the software.

4 Case Study: A Welded Beam Design Problem

In this problem, a beam is welded on another beam and carry a certain load. This design problem with one objective is a particularly well-studied [13] one, but here we modify the problem to include a second objective:

$$\begin{array}{ll}
\text{Minimize} & f_1(\mathbf{x}) = 1.10471h^2\ell + 0.04811tb(14.0+\ell), \\
\text{Minimize} & f_2(\mathbf{x}) = \delta(\mathbf{x}) = \frac{2.1952}{t^3 b}, \\
\text{Subject to} & g_1(\mathbf{x}) \equiv 13,600 - \tau(\mathbf{x}) \geq 0, \quad g_2(\mathbf{x}) \equiv 30,000 - \sigma(\mathbf{x}) \geq 0, \\
 & g_3(\mathbf{x}) \equiv b - h \geq 0, \quad g_4(\mathbf{x}) \equiv P_c(\mathbf{x}) - 6,000 \geq 0, \\
 & 0.125 \leq h, b \leq 5.0, \quad 0.1 \leq \ell, t \leq 10.0.
\end{array} \tag{1}$$

The first objective is the cost of fabrication and second objective is the end deflection, both of which are to be minimized. Four non-linear constraints are related to limitations on normal stress $\sigma(\mathbf{x})$, shear stress $\tau(\mathbf{x})$, buckling load $P_c(\mathbf{x})$ and a dimensional practicality. There are four design variables: thickness of the beam b, width of the beam t, length of weld ℓ, and weld thickness h, each bounded between lower and upper bounds. The non-linear terms for stress and buckling are given elsewhere [13]. The problem is coded in the pre-processor phase of I-MODE and following systematic procedure is used to obtain a single solution from a two-objective consideration.

4.1 Step 1: Find an Approximate Front

First, we find an idea of the Pareto-optimal front using I-MODE. We set following parameter values: Population size=100; maximum generation=100; crossover probability=0.9; mutation probability=0.1; distribution indices for SBX recombination and polynomial mutation are 10 and 20, respectively. Figure 4 shows the obtained front (solutions marked from A to B) using NSGA-II. To validate the obtained front, we independently find the estimated ideal point by individual minimization of each objective and the estimated nadir point by using the

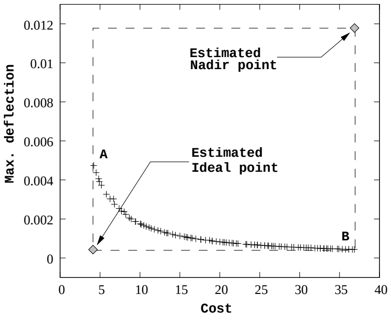


Fig. 4. Pareto-optimal front after original NSGA-II run.

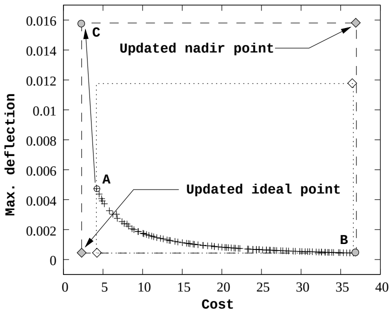


Fig. 5. Updated extreme solutions of the non-dominated front.

nadir point estimation procedure of I-MODE. Estimated ideal and nadir points are joined to show the range of obtained trade-off optimal solutions. It is clear from the figure that the non-dominated points obtained through NSGA-II do not cover the entire range defined by the nadir and ideal objective vectors. Thus, we can conclude that the NSGA-II procedure is unable to find the entire optimal front and we need to improve this frontier.

4.2 Step 2: Improve the Trade-off Frontier

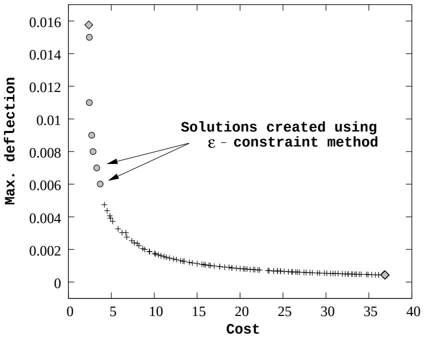


Fig. 6. Updated Pareto-optimal front of welded beam problem.

To obtain a better front, we use the local search procedure on the end points. For this purpose, we use the `fmincon` optimization procedure of MATLAB procedure, which is a classical SQP method. We link MATLAB with the I-MODE software through the local search option: `User defined`. Figure 5 shows the solutions after the local searches. The minimum-cost solution A obtained through NSGA-II gets largely improved to solution C, but the minimum-deflection solution B gets improved slightly. The updated ideal and nadir points are found to be different from those obtained earlier, due to the difficulty in obtaining the minimum-cost solution in this problem. Now from these updated results, we observe that there is a gap between solutions A and C. So we try to find the missing part of the Pareto-optimal front by using the ϵ-constraint method by minimizing f_1 and constraining f_2 to several ϵ_2 values. The obtained solutions are then passed through a non-domination check with the original NSGA-II solutions and the front is modified with new solutions.

Figure 6 shows the updated front consisting of previous NSGA-II run and ϵ-constraint single-objective solutions.

4.3 Step 3: Verify Obtained Front

Since the new ϵ-constraint optimization runs find solutions which are well-matched with the NSGA-II frontier, we skip the verification process at this first iteration of the proposed I-MODE procedure.

4.4 Step 4: Make Decisions and Choose Regions of Interest

The above steps helped us get an idea of the range of Pareto-optimal solutions. The next step is to find one or more regions of interest based on a higher-level consideration. Here, we use two criteria. First we are interested in concentrating in a region which is robust (less sensitive to the variable perturbation). We have already seen in subsection 4.2 that minimum-cost region is sensitive to parameter values and difficult to optimize. To perform the robustness study, we assume that the beam dimensions t and b are expected to vary with $\pm 2\%$ from their chosen values and weld dimensions h and l vary with $\pm 4\%$. These values are kept this way to take into account the fact that parameters t and b are obtained by a machining operation and are expected to have a better control on dimensional tolerance compared to the weld dimensions. We use the robustness of type II [8] procedure and obtain the robust frontier with robustness parameter $\eta = 0.01$, meaning that a maximum of 1% difference in average perturbation in objective values from their original values due to uncertainty is allowed. Figure 7 shows the robust frontier. It is interesting to note that minimum-cost solutions are sensitive.

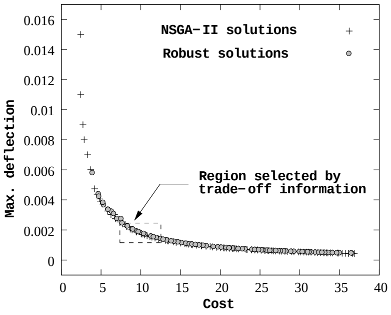


Fig. 7. Robust solutions obtained using robust NSGA-II.

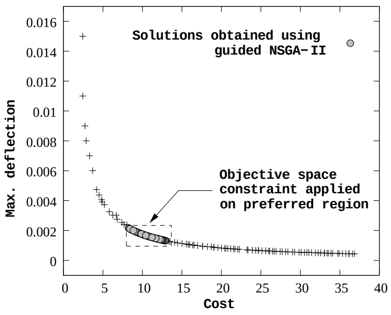


Fig. 8. Solutions obtained using guided NSGA-II.

Since the minimum-cost solution corresponds to minimal use of materials, the solution tends to make most constraints active. With an expected fluctuation in design variables, such solutions can easily become infeasible and cannot qualify to be robust. Thus, the robust consideration as a direct decision-making tool enables us to keep away from choosing a solution close to the minimum-cost solution. However, still in this problem we observe a wide variety of solutions which qualify as robust solutions.

To reduce our focus further, we now use a subjective decision-making procedure of surrogate worth trade-off. Of the robust solutions, we are interested in solutions for which a 100% sacrifice in the cost value, at-least 150% improvement in deflection occurs. That is, from a solution if we double the cost value, we are interested in solutions which reduces the deflection 2.5 times. Simultaneously, we would also like to ensure that a saving of at-least 25% cost for a 100% sacrifice in deflection. To find such solutions from the remaining portion of the trade-off frontier, we specify the following matrix and obtain a partial frontier by the I-EMO software (Figure 8):

$$\texttt{Tradeoff matrix} = \begin{bmatrix} 1.0 & 1.5 \\ 0.25 & 1.0 \end{bmatrix} \tag{2}$$

It is interesting to note that only a small portion in the intermediate portion of the robust frontier becomes the preferred region of solutions corresponding to above trade-off information.

4.5 Step 5: Termination Criterion

This completes one iteration of the I-EMO procedure. Since we have not converged to a single solution yet, we move to Step 1 for another round of I-EMO but concentrate only in the trade-off region obtained at the end of Step 4.

4.6 Step 1: Find More Solutions in Preferred Region by NSGA-II

We run NSGA-II with the guided-domination concept and obtained more solutions in the preferred region of interest. Figure 9 shows 100 solutions obtained with a rerun of guided NSGA-II.

4.7 Step 2: Improve the Front

We ignore this step due to a robustness study planned in subsection 4.9.

4.8 Step 3: Verify Obtained Front

Here, we perform five ϵ-constraint single-objective minimizations of f_2 by constraining f_1 into different cost values in the current range $[8.2, 13.0]$. The obtained solutions are shown in Figure 9 with circles, which suggests that the NSGA-II front and these ϵ-constraint solutions more or less agree, thereby gaining confidence on the obtained NSGA-II solutions.

We perform another verification process here. Since in the previous iteration we expected certain trade-off (given in equation 2), we compute the pseudo-weight vector of five widely-separated solutions (in diamonds). Table 1 shows the objective function values and weight vectors for these selected points.

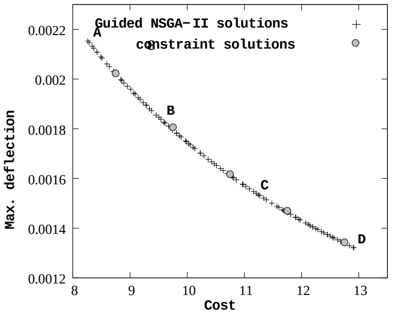


Fig. 9. Guided NSGA-II run is confirmed by ϵ-constraint method.

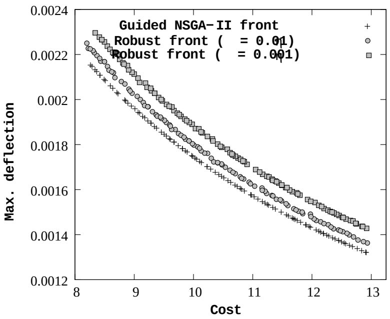


Fig. 10. Robust optimal fronts.

Table 1. Pseudo-weight for selected solutions.

Solution	Cost	Max. deflection	Pseudo-weight	
	f_1	f_2	w_1	w_2
A	8.262	0.002153	0.48	0.52
B	9.541	0.001838	0.47	0.53
C	11.087	0.001558	0.45	0.55
D	12.914	0.001321	0.42	0.58

Recall that for an identical proportion of loss in either objective a more stringent gain in deflection objective was set by the matrix. From the above table, we observe that the second objective is given more importance than the first objective in the selected region. These calculations give us confidence in our approach and we now proceed to make further decisions to choose a single preferred solution.

4.9 Step 4: Make Decisions and Choose Subregions of Interest

To choose a subregion of interest, we first investigate the robustness of the current trade-off frontier using two different robustness parameter values of 0.01 and 0.001. Figure 10 shows that with a stricter requirement in fluctuation in function values due to perturbations in design variables, the robust frontier gets worse. To investigate how the solutions (design variables) change with a more strict requirement on objective fluctuations, we plot two design variables (t and b) versus the cost objective in Figure 11. Other two variables are found to have a similar 'almost constant' behavior as that of t. The robust solutions are different from the original solutions and it is interesting to note that the variable b of robust and Pareto-optimal solutions follow a relationship with the cost objective: the beam thickness variable must be increased linearly with large-cost solutions. Interestingly, the change in beam thickness does not seem to depend much on the chosen robustness parameter, but demand a significant change in t. These informations are interesting and useful and are found as a by-product of I-MODE procedure. Based on these plots, we decide to fix the robustness parameter to $\eta = 0.01$ and proceed with the rest of the study.

To narrow down the preferred region, next we consider a subjective decision-making tool with reference points. Say, we are interested in solutions towards two

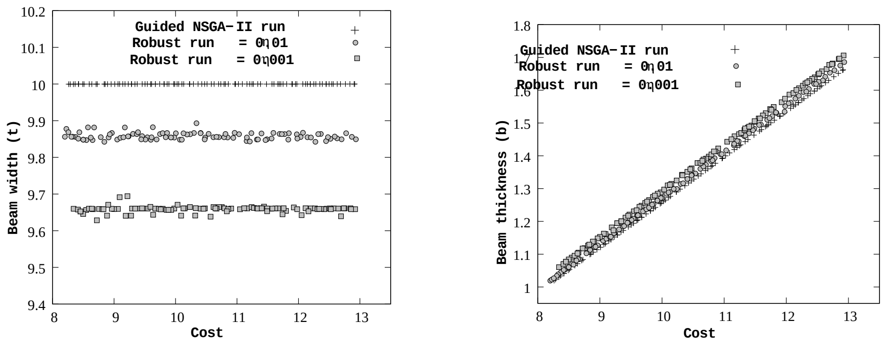


Fig. 11. Variable sensitivity of the welded beam problem.

extreme regions of the remaining trade-off front and specify following two reference (aspiration) points: $(8.7, 0.00195)^T$ and $(12.0, 0.0014)^T$. To get a reasonable spread of solutions, we choose (by trial-and-error here) a spread parameter of $\epsilon = 0.001$. Figure 12 shows the final solutions obtained by the reference NSGA-II run on both reference points simultaneously. Reference points are also shown in the figure.

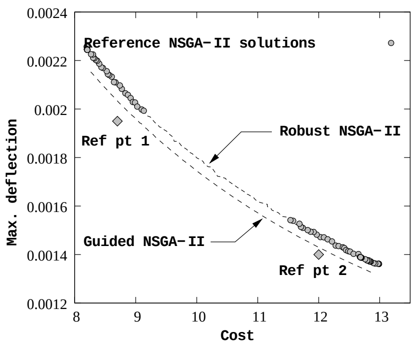


Fig. 12. Robust Pareto-optimal solutions based on the target values.

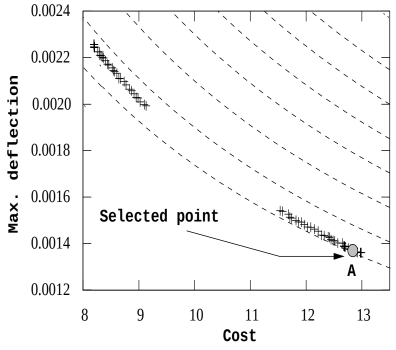


Fig. 13. Final solution selected based on utility function.

Finally, we decide to use another subjective decision-making tool based on the utility function approach. We decide to use the following utility function: Minimize $U(f_1, f_2) = f_1 \times f_2$. Since cost and deflection are con-

Table 2. Most preferred solution of the welded-beam problem with supplied decision-making aides.

Design Variables (in)				Objective Values	
h	l	t	b	Cost	Deflection
0.917	1.009	9.856	1.672	12.838	0.00137

flicting to each other, a product of the two objective values in the regions of our interest may be thought as a combined utility measure, minimizing which may result a solution having small values of both objectives. Figure 13 shows the contour plot of the above utility function and reference point based NSGA-II solutions. The utility function is tangential with the reference NSGA-II solutions at point A, thereby meaning that the solution A is the most preferred solution with respect to the chosen utility. The decision variables and objective function values of this solution are shown in Table 2. It is important to note here that the above task of finding the best solution based on a utility function is performed on a chosen *range* of *robust* solutions and on solutions exhibiting certain *trade-off* information and is not to be confused with finding the best solution for a fixed utility function on the entire Pareto-optimal solution.

4.10 Step 5: Select the Most Preferred Solution

Since the outcome is a single solution, we terminate the I-MODE procedure and declare solution A is the preferred outcome of the complete multi-objective optimization and decision-making procedure.

Here, we have followed a sequence of steps with some subjective decision-making tools to come up with a preferred solution, which is *robust*, *near-optimal*, having desired *trade-off* in objectives, close to preferred *aspiration* points, and possessing *optimal desired utility.* It is obvious that the outcome of the study would change if any major change in the sequence of operation of steps is chosen or a different decision-making tool or different parameter values are chosen. There are certainly other procedures possible which will result in a different solution. This is the unique feature of multi-objective optimization. But, what we have demonstrated here is a systematic procedure of using such mixed optimization-cum-decision-making strategies for arriving at a preferred solution.

5 Conclusions

In this paper, we, for the first time, have proposed an interactive optimization and decision-making procedure for solving two and three-objective optimization problems. In order to arrive at the procedure with a GUI-based software, we have used salient research results from classical and evolutionary multi-objective optimization literatures in a synergistic manner. The procedure not only finds near Pareto-optimal fronts and then helps the DM to choose a particular solution, the procedure provides options for doing *checks and balances* at various stages so that the DM is more confident in arriving at a particular solution. Till now, no such combined (classical and EMO) software is available for this task. With the ground-breaking research and application studies using EMO so far, it is now time for researchers to think and develop such interactive hybrid methodologies which will give multi-objective optimization studies a real practical flavor which they rightfully deserve.

Acknowledgement

Authors acknowledge the financial support provided by STMicroelectronics, Italy, Singapore and India during the course of this study.

References

1. J. Branke, T. Kauβler, and H. Schmeck. Guidance in Evolutionary Multi-objective Optimization. *Advances in Engineering Software*, 32:499–507, 2001.
2. V. Chankong and Y. Y. Haimes. *Multiobjective Decision Making Theory and Methodology*. New York: North-Holland, 1983.
3. K. Deb. *Multi-objective optimization using evolutionary algorithms*. Chichester, UK: Wiley, 2001.
4. K. Deb, S. Agrawal, A. Pratap, and T. Meyarivan. A fast and elitist multi-objective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6(2):182–197, 2002.
5. K. Deb and S. Chaudhuri. Automated discovery of innovative designs of mechanical components using evolutionary multi-objective algorithms. In N. Nedjah and L. de Macedo M, editors, *Evolutionary Machine Design: Methodology and Applications*, chapter 6, pages 143–168. Nova Science Publishers, Inc, New York, 2005.
6. K. Deb and S. Chaudhuri. I-EMO: An interactive evolutionary multi-objective optimization tool. In *Pattern Recognition and Machine Intelligence: First International Conference (PReMI-2005)*, pages 690–695. Springer, 2005.
7. K. Deb and T. Goel. A hybrid multi-objective evolutionary approach to engineering shape design. In *Proceedings of the First International Conference on Evolutionary Multi-Criterion Optimization (EMO-01)*, pages 385–399, 2001.
8. K. Deb and H. Gupta. Searching for robust Pareto-optimal solutions in multi-objective optimization. In *Proceedings of the Third Evolutionary Multi-Criteria Optimization (EMO-05) Conference (Also Lecture Notes on Computer Science 3410)*, pages 150–164, 2005.
9. K. Deb and S. Jain. Running performance metrics for evolutionary multi-objective optimization. In *Proceedings of the Fourth Asia-Pacific Conference on Simulated Evolution and Learning (SEAL-02)*, pages 13–20, 2002.
10. K. Deb, J. Sundar, U. B. Rao N., and S. Chaudhuri. Reference point based multi-objective optimization using evolutionary algorithms. *International Journal of Computational Intelligence Research*, 2(3):273–286, 2006.
11. C. M. Fonseca and P. J. Fleming. Multiobjective optimization and multiple constraint handling with evolutionary algorithms–Part II: Application example. *IEEE Transactions on Systems, Man, and Cybernetics: Part A: Systems and Humans*, 28(1):38–47, 1998.
12. K. Miettinen. *Nonlinear Multiobjective Optimization*. Kluwer, Boston, 1999.
13. G. V. Reklaitis, A. Ravindran, and K. M. Ragsdell. *Engineering Optimization Methods and Applications*. New York : Wiley, 1983.
14. K. C. Tan, T. H. Lee, D. Khoo, and E. F. Khor. A multiobjective evolutionay algorithm toolbox for computer-aided multiobjective optimization. *IEEE Transactions on Systems,Man, and Cybernetics - Part B: Cybernetics*, 31(4):537–556, 2001.
15. A. P. Wierzbicki. The use of reference objectives in multiobjective optimization. In G. Fandel and T. Gal, editors, *Multiple Criteria Decision Making Theory and Applications*, pages 468–486. Berlin: Springer-Verlag, 1980.