

An Adaptive Normalization based Constrained Handling Methodology with Hybrid Bi-Objective and Penalty Function Approach

Rituparna Datta and Kalyanmoy Deb
Department of Mechanical Engineering,
Indian Institute of Technology
Kanpur, PIN: 208018, INDIA
Email: {rdatta,deb}@iitk.ac.in

KanGAL Report Number 2012005

Abstract—A hybrid adaptive normalization based constraint handling approach is proposed in the present study. In most constrained optimization problems, constraints may be of different scale. Normalization of constraints is crucial for the efficient performance of a constraint handling algorithm. A growing number of researchers have proposed different strategies using bi-objective methodologies. Classical penalty function approach is another common method among both evolutionary and classical optimization research communities due to its simplicity and ease of implementation. In the present study, we propose a hybrid approach of both bi-objective method and the penalty function approach where constraints are normalized adaptively during the optimization process. The proposed bi-objective evolutionary method estimates the penalty parameter and the starting solution needed for the penalty function approach. We test and compare our algorithm on seven mathematical test problems and two engineering design problems taken from the literature. We compare our obtained results with our previous studies in terms of function evaluations and solution accuracy. The obtained optima are also compared with those of other standard algorithms. In many cases, our proposed methodology perform better than all algorithms considered in this study. Results are promising and motivate further application of the proposed adaptive normalization strategy.

I. INTRODUCTION

OVER few decades noteworthy research has been performed on evolutionary algorithms and constraint handling. Constrained optimization problem occurs in many fields of science, engineering and structural design and economics etc. Objective function and constraints can be non-linear, non-convex and/or dis-continuous in nature. Due to existence of complex and non-linear nature (one or both objective function and constraints), classical mathematics based optimization methods are not able to perform well in these problems. The failures of classical mathematics based optimization methods open a room for evolutionary researchers to handle these problems.

Generally a standard constrained optimization problem can be formulated as follows:

$$\begin{aligned} & \text{Minimize} && f(\mathbf{x}), \\ & \text{Subject to} && g_j(\mathbf{x}) \geq 0, j = 1, \dots, J, \\ & && h_k(\mathbf{x}) = 0, k = 1, \dots, K, \\ & && x_i^l \leq x_i \leq x_i^u, i = 1, \dots, n. \end{aligned} \quad (1)$$

In the above problem, number of variables are n , J denotes the number of greater-than-equal-to type constraints, and K is the number of equality constraints. The function $f(\mathbf{x})$ is the objective function, $g_j(\mathbf{x})$ is the j -th inequality constraint, and $h_k(\mathbf{x})$ is the k -th equality constraint. The i -th variable varies in the range $[x_i^l, x_i^u]$. In this study, we restrict ourselves to handling inequality constraints only, however an equality constraint can be handled by converting it to an appropriate inequality constraint: $g_{J+k}(\mathbf{x}) = |\epsilon_k - h_k(\mathbf{x})| \geq 0$, with a small given value of ϵ_k .

One popular way to solve a constrained optimization problem is to use the penalty function methodology. Penalty function method is popular due its simple working principle. In earlier days researchers suggested different penalty function methods to deal with constraints [1], [2], [3], [4]. The main finding of their studies was the need of using an appropriate penalty term in order to scale the overall constraint violation with the objective function value. The second author [5] proposed a penalty-parameter-less approach where an infeasible solution always treated worse than a feasible solution, the approach is free from any penalty parameter. Coello [6] proposed a self-adaptive penalty approach by using a co-evolutionary model to adapt the penalty parameters.

In some studies the objective function and constraints are treated independently which is known as bi-objective optimization problem, where constraint violation measure is used to construct as another objective [7], [8], [9], [10], [11]. A study based on bi-objective approach was proposed in [12] that used a reference point approach to focus the search in the neighborhood of the constrained minimum solution. Wang et al. [13] proposed an adaptive trade off model (ATM) using a bi-objective approach with three-phase methodology for handling constraints in evolutionary optimization. A recent study reviewed [14] the concept of handling constraints using bi-objective approach. The readers are encouraged to refer these studies for more details.

A number of researchers use a hybrid approach for constraint handling. In many situations finding the exact constrained optimum using evolutionary methods are a difficulty. To overcome this problem, researchers hybridize evolutionary methods with a classical optimization method [15], [16],

[12], [17]. A constraint handling study hybridized genetic algorithm with artificial immune system (AIS), where the role of AIS was to help in pushing the population towards feasible region [18]. A recent study combined genetic algorithm with “complex search” algorithm [19] to improve the convergence and applied to constrained trajectory optimization. Optimal solution of genetic algorithm used as an initial parameter for the complex search method. Another recent methodology proposed a hybrid genetic algorithm with a flexible allowance technique (GAFAT) for solving constrained engineering design optimization problems by fusing center based differential crossover (CBDX), Levenberg Marquardt mutation (LMM) and non-uniform mutation (NUM) [20].

Michalewicz [21] and Coello [22] did a comprehensive survey with the evolutionary based constraint handling methods executed in the last decade. Recently Mezura [23] edited a book on constraint handling in evolutionary optimization. The most recent study in constraint handling survey using nature-inspired optimization can be found in [24]. The following methodologies are briefly described in their paper:

- Feasibility rules
- Stochastic ranking
- ϵ -constrained method
- Novel penalty functions
- Novel special operators
- Multi-objective concepts
- Ensemble of constraint-handling techniques

The authors also showed a good future direction for the researchers in constraint handling areas. These areas will be helpful for researchers, novice and experts alike. The areas are as follows:

- Constraint-handling for multi-objective optimization
- Constraint Approximation
- Dynamic constraints
- Hyper-heuristics
- Theory

A recent methodology described a framework based on both genetic algorithm and differential evolution which consists of collective search operators in every generation and adaptively mixes them to solve constrained optimization problems [25]. Recently, we proposed a bi-objective evolutionary optimization based strategy [26] to approximate the penalty parameter (R) from the obtained bi-objective frontier between the objective function value and sum of normalized constraint violations. Based on the optimal solution (in terms of few or no constraint violation) and the approximated penalty parameter, the constructed penalty function problem is solved using a classical local search method.

In the remainder of this paper, we describe scaling issues in constraint handling and our proposed algorithm for adaptive normalization of constraints. Consequently, we describe proposed hybrid optimization with adaptive scaling. Then, we consider standard numerical and engineering design problems from the existing literature and show the results obtained from our proposed algorithm. We compare the obtained results with some of the best-known results from

the literature. Finally, the paper ends with conclusions.

II. SCALING ISSUES IN CONSTRAINED OPTIMIZATION

In our previous work [26], we briefly illustrated the bi-objective penalty function based hybrid methodology to handle inequality constraints. In the present work our motivation is to study the effect of scaling in constrained optimization and propose some methodology to scale the constraints adaptively.

As described earlier, penalty function approach is the simplest method to deal with constraints. The main concept in the penalty function approach is that a constrained optimization problem is converted into an unconstrained one by adding constraint violation to the objective function as penalty term. Any unconstrained optimization algorithm can then be applied after formulating it through penalty function methodology. In order to reduce the complexity and cut down the number of penalty parameters, often the constraints are normalized (generally between $[0, 1]$) and only one penalty parameter is introduced [27]. In order to use penalty function method effectively, all constraints should be scaled to take similar values. The idea of this scaling operation is to ensure that, no constraints has any artificial dominance on the other. If some constraints has more influence, then the algorithm will guide the search towards satisfying those constraints by neglecting the dominated ones. A study was proposed to scale each constraint in order to restore the described computational inefficiency by multiplying each constraint by an appropriate scale factor [28]. Another study on constraint scaling named *Evolian* [29] was proposed based on the concept of multi-phase optimization, where constraints were scaled in each phase.

Generally in most problems (specially in real-world problems) constraints are of different scale. Therefore in most situations all constraints are scaled or normalized based on problem formulation.

Considering equation (1), inequality constraints can be written in following form:

$$g_j(\mathbf{x}) \geq c_j, j = 1, \dots, J, \quad (2)$$

where c_j is the constant term (usually, the resource value) associated with j -th inequality constraint. Based on the problem information the j -th constraint can be normalized as:

$$\frac{g_j(\mathbf{x})}{c_j} - 1 \geq 0. \quad (3)$$

All constraints can be normalized manually in the similar way to make each constraint violation approximately on the same scale. This method has a deficiency in order to identify the appropriate scaling factor in many situations and also it is tedious to find the appropriate scaling factor to normalize the particular constraint. In most of the real-world situations, problem information may not be available in mathematical form, so the normalization task will be difficult.

Here, we modify our normalization procedure in an adaptive way. Before starting a generation, for each constraint (say j -th), constraint violation (V_j) of all population members are calculated. Population members are then sorted based on their constraint violation value. Then, we take top 50% members and an average is computed ($V_{j,avg}$). Every constraint is divided with this computed average value in both evolutionary and classical approaches in our algorithm. The scaling factor is updated at each generation. We describe the algorithm for adaptive normalization in the following algorithmic form.

Algorithm 1 Steps for adaptive normalization of constraints

Input: Population $[\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N]$

Output: $V_{j,avg}, j = 1, \dots, J$

```

1: for  $j = 1$  to  $J$  do
2:   viocount  $[j] = 0$ 
3:   sum  $[j] = 0$ 
4: end for
5:
6: for  $i = 1$  to  $N$  do
7:   for  $j = 1$  to  $J$  do
8:     Evaluate  $g_j(\mathbf{x}_i)$ 
9:     if  $g_j(\mathbf{x}_i) < 0$  then
10:      gcount  $[j]$   $[viocount[j]] = -g_j(\mathbf{x}_i)$ 
11:      viocount  $[j] = viocount [j] + 1$ 
12:     end if
13:   end for
14: end for
15:
16: for  $j = 1$  to  $J$  do
17:   Sort vector gcount  $[j]$   $[:]$  in increasing order
18:   for  $k = 1$  to  $\lceil 0.5 \times viocount[j] \rceil$  do
19:     sum  $[j] = \text{sum } [j] + \text{gcount } [j] [k]$ 
20:   end for
21:
22:   for  $j = 1$  to  $J$  do
23:     if sum  $[j] = 0$  then
24:        $V_{j,avg} = 1$ 
25:     else
26:        $V_{j,avg} = \text{sum } [j] / \lceil 0.5 \times viocount [j] \rceil$ 
27:     end if
28:   end for
29: end for
30: Send  $V_{j,avg}$  of each constraint to divide that constraint
    in both evolutionary and local search.

```

III. PROPOSED HYBRID APPROACH

Evolutionary algorithms are getting popularity for constrained handling, due to their efficiency, flexibility, robustness, ease of implementation and their ability to be integrate with any other algorithm. In our earlier study, we described the advantage of bi-objective EA and the penalty function approach [26]. In this section, we will describe our adaptive normalization based hybrid approach. The hybridization with

adaptive normalization is made between a bi-objective EA (using NSGA-II) and integrated with a penalty function based classical approach for faster convergence.

A. Proposed Algorithm

Our adaptive normalization based hybrid algorithm is a combination of bi-objective way of handling constraints using EA and a penalty function approach with classical optimization. The proposed algorithm is a modified version of our earlier study [26] integrated with adaptive normalization technique. The algorithm is as follows:

First, the generation counter is set at $t = 0$.

Step 1: An evolutionary algorithm (NSGA-II) is applied to the bi-objective optimization problem to find non-dominated Pareto-optimal front:

$$\begin{aligned}
& \text{minimize } f(\mathbf{x}), \\
& \text{minimize } CV(\mathbf{x}) = \sum_{j=1}^J \frac{g_j(\mathbf{x})}{V_{j,avg}}, \\
& \text{subject to } CV(\mathbf{x}) \leq c, \\
& \quad \mathbf{x}^{(L)} \leq \mathbf{x} \leq \mathbf{x}^{(U)}.
\end{aligned} \tag{4}$$

An artificial constraint is included in the Pareto-optimal front to focus search in the neighborhood of minimum constraint violation solution. Since $CV(\mathbf{x})$ is the adaptively normalized constraint violation, we have used $c = 0.2$ for this study.

Step 2: If $t > 0$ and $((t \bmod \tau) = 0)$, we estimate penalty parameter R from the current non-dominated Pareto-optimal front. A cubic curve ($f = a + b(CV) + c(CV)^2 + d(CV)^3$) is fitted in the non-dominated points and the penalty parameter is approximated by finding the slope at $CV=0$, that is, $R = -b$. Since this is a lower bound on R , we use twice this value as R i.e. $R \leftarrow -2b$.

Step 3: The following local search problem is solved with R computed from above and starting with the current minimum-CV solution:

$$\begin{aligned}
& \text{minimize } P(\mathbf{x}) = f(\mathbf{x}) + R \times CV(\mathbf{x}), \\
& \quad \mathbf{x}^{(L)} \leq \mathbf{x} \leq \mathbf{x}^{(U)}.
\end{aligned} \tag{5}$$

Say, the solution is $\bar{\mathbf{x}}$.

Step 4: If the value $\bar{\mathbf{x}}$ is feasible and the difference between $f(\bar{\mathbf{x}})$ and the objective value of the previous local searched solution is smaller than a small number δ_f (10^{-4} is used here), the algorithm is terminated and $\bar{\mathbf{x}}$ is declared as the optimized solution. Else, we increment t by one and proceed to Step 1.

Based on the above procedure, penalty parameter (R) is not a parameter defined by the user. It gets automatically adjusted from the obtained non-dominated front. We have introduced a new parameter τ which is frequency of local search. Our extensive parametric study on a number of problems show that the parameter τ does not have much effect on the outcome of our proposed method, except that if too small value of R is chosen the local search is called many times, thereby increasing the number of local search evaluations. On the other hand, if too large value of R is taken, the procedure does not get injected with good local

searched solutions, thereby delaying convergence. We show a parametric study for a problem described in Section IV-B. Based on our study, we recommend the use of $\tau = 5$.

In all our study, we use MATLAB's `fmincon()` procedure to solve the penalized function (equation 5) with the default parameter setting. Other classical optimization methodologies can also be used instead.

IV. EXPERIMENTAL STUDIES ON STANDARD TEST PROBLEMS

In this section, we perform experimental studies to check the efficiency of our approach on some standard constrained optimization test problems taken from [5].

The parameters used in these experiments are as follows:

Population size = $16n$ (unless stated otherwise),
 SBX probability= 0.9,
 SBX index = 10,
 Polynomial mutation probability = $1/n$, and
 Mutation index = 100.

The termination criterion of this approach is described in Section III-A. For every test problem, the algorithm is tested 50 times with different initial populations.

A. Problem P1

First, we consider the following two-variable problem having a non-convex feasible region [26]:

$$\begin{aligned} & \text{minimize} && f(\mathbf{x}) = (x_1 - 3)^2 + (x_2 - 2)^2, \\ & \text{subject to} && g_1(\mathbf{x}) \equiv 4.84 - (x_1 - 0.05)^2 - (x_2 - 2.5)^2 \geq 0, \\ & && g_2(\mathbf{x}) \equiv x_1^2 + (x_2 - 2.5)^2 - 4.84 \geq 0, \\ & && 0 \leq x_1 \leq 6, \quad 0 \leq x_2 \leq 6. \end{aligned} \quad (6)$$

First constraint g_1 is active at the optimal point. The optimal solution for this problem is at $\mathbf{x}^* = (2.219, 2.132)^T$ with a function value of 0.627. To achieve different scaling between two constraints, we multiply the first constraint by 100. So, the constraint g_1 becomes,

$$g_1(\mathbf{x}) \equiv 484 - 100 \times [(x_1 - 0.05)^2 + (x_2 - 2.5)^2] \geq 0.$$

Now both constraints are of different scale but the new optimum is the same as before. In our earlier study [26], we divided both the constraints manually by 4.84 for normalization. In the present study, to make the problem more challenging we change scaling (as above) between the constraints. With the introduced scaling and no problem specific normalization method, both the constraints are normalized adaptively during the optimization run, as described in the previous section.

Our earlier approach [26], found the optimum solution of the original problem (equation (6)) after 677 function evaluations (600 needed by EMO and 77 by `fmincon()` procedure). In the current study, the proposed hybrid approach take 731 function evaluations (600 needed by EMO and 131 by `fmincon()` procedure) to solve the problem with introduced scaling and no problem specific normalization. Identical optimum was found.

TABLE I
FUNCTION EVALUATIONS, FE (NSGA-II AND LOCAL SEARCH) AND OPTIMAL SOLUTION, BY THE EARLIER APPROACH AND PROPOSED HYBRID APPROACH IN 50 RUNS.

		Best	Median	Worst
Earlier Approach	FE	677 (600 + 77)	733 (600 + 133)	999 (900 + 99)
	f	0.627380	0.627379	0.627379
Proposed Approach	FE	731 (600 + 131)	1297 (900 + 397)	1,623 (1200 + 423)
	f	0.627378	0.627393	0.627377

In Table I we tabulate best, median and worst performance of the hybrid algorithm in both earlier and present study with 50 different runs. In all these cases, our present approach finds the true optimal solution, despite having to deal with differently scaled constraints.

B. Problem TP3

This problem is a quadratic minimization problem with 13 variables and 9 linear inequality constraints, of which six are active constraints. The optimum objective function value is -15 . Table II shows the problem specific normalization value using problem information from our previous study [26] and adaptive normalization value ($V_{j,avg}$) proposed in the present study.

Table III presents best, median and worst function evaluation (FE) reached with the adaptive normalization based proposed hybrid approach. The least number of function evaluations required with our approach is 2,333, whereas 2,427 function evaluations was needed in the previous study [5]. If we compare in terms of the obtained optimum (Table III), in earlier study the optimum could not be found in worst performance whereas in the current study the best FE could not achieve the optima.

Table IV tabulates and compares the function values achieved by our approach with three other methodologies taken from literature [13], [30], [31]. In this problem, in terms of best function value, our approach perform successfully, whereas in term of median and worst function value other approaches outperform ours.

We perform a parametric study to investigate the frequency of local searches (τ). Figure 1 shows the overall function evaluations in vertical axis and τ in the horizontal axis. The parametric study on frequency of local search shows that $\tau = 1$ (local search in every generation) is the best

TABLE II
PROBLEM SPECIFIC NORMALIZATION VALUE FROM PREVIOUS STUDY [26] AND ADAPTIVE NORMALIZATION VALUE ($V_{j,avg}$) USING PROPOSED HYBRID APPROACH (AFTER 5 AND 10 GENERATIONS).

	Generation	g_1	g_2	g_3	g_4	g_5	g_6	g_7	g_8	g_9
Earlier Approach	Fixed	10	10	10	8	8	8	2	2	2
Proposed Approach	5	40.3	44.8	34.3	31.2	23.6	25.3	32.3	24.7	26.3
	10	21.1	23.2	20.6	16.1	17.8	14.9	14.5	15.3	14.4

TABLE III
COMPARISON OF FUNCTION EVALUATIONS (FE) NEEDED BY THE PROPOSED APPROACH AND THE EXISTING EARLIER APPROACH [26]. FUNCTION EVALUATIONS BY NSGA-II AND LOCAL SEARCH HAVE BEEN SHOWN SEPARATELY. THE BETTER ONE IN TERMS OF FE IS SHOWN IN BOLD.

Problem	Earlier Approach [26]			Proposed Hybrid Approach		
	Best	Median	Worst	Best	Median	Worst
TP3 (FE)	2,427	4,676	13,762	2,333	2,856	11,843
NSGA-II+Local	2,000 + 427	3,000 + 1,676	11,000 + 2,762	2,000 + 333	2,000 + 856	8,000 + 3,843
(f^*)	-15	-15	-12	-12	-15	-15
TP4 (FE)	54,946	31,367	1,00,420	2,705	27,235	1,07,886
NSGA-II+Local	24,600 + 30,346	4,400 + 16,967	45,600 + 54,820	1,200 + 1,505	7,200 + 20,035	45,600 + 62286
(f^*)	7,049.943	7,078.625	7,940.678	7,049.588	7,059.576	7,065.348
TP5 (FE)	6,332	15,538	38,942	1,961	11,729	42,617
NSGA-II+Local	3,920 + 2,412	9,520 + 6,018	25,200 + 13,742	1,120 + 841	7,280 + 4,449	27,440 + 15,177
(f^*)	680.630	680.634	680.876	680.635	680.631	680.646
TP6 (FE)	1,120	2,016	6,880	1,123	4,183	13,631
NSGA-II+Local	800 + 320	1,200 + 816	3,600 + 3,280	800 + 323	2,400 + 1,783	8,400 + 5,231
(f^*)	-30,665.539	-30,665.539	-30,649.552	-30,665.539	-30,665.539	-30,665.539
TP8 (FE)	4,880	23,071	83,059	7,780	68,977	3,54,934
NSGA-II+Local	3,200 + 1,680	18,000 + 5,071	44,800 + 38,259	5,600 + 2,180	41,600 + 27,377	2,20,800 + 1,34,134
(f^*)	24.308	25.651	31.254	24.565	24.306	24.306
Weld (FE)	1,067	2,261	9,115	1,607	5,761	13,636
NSGA-II+Local	800 + 267	600 + 661	6,400 + 2,715	1,200 + 407	4,000 + 1,761	10,000 + 3,636
(f^*)	2.38113	2.38216	2.46835	2.42551	2.38159	2.38113

strategy in comparison to all others. The resulting algorithm pushes population members to converge towards the constrained optimum quickly. But in terms of median and worst performance, $\tau = 5$ (local searches after every 5 generations) is better than all others. If we compare the spread (out of 50 runs), the performance of $\tau = 5$ is best compared to all other frequency of local searches. A high frequency of local search is not suggested as the efficiency of the algorithm decreases with an increase in τ . Based on our study, a suggested range of τ is between 1 to 10.

C. Problem TP4

This problem consists of eight decision variables with six constraints (three linear and three non-linear). The objective function is linear in nature. The optimum solution is at $\mathbf{x}^* = (579.307, 1359.971, 5109.971, 182.018, 295.601, 217.982, 286.417, 395.601)^T$ with an objective function value of $f^* = 7049.280$. All constraints are active at this optimum.

Table III shows the efficiency of the present approach. A better objective function value is achieved using our proposed adaptive normalization approach than that of the existing study [26]. In terms of all the performance measures, the present approach outperforms the earlier study. If we compare them with respect to the objective function, the present approach performs better in terms of all (best, median and worst) performance. In Table IV, we compare our optimum

function values with three existing approaches. In all these cases our solution dominates all the other solutions.

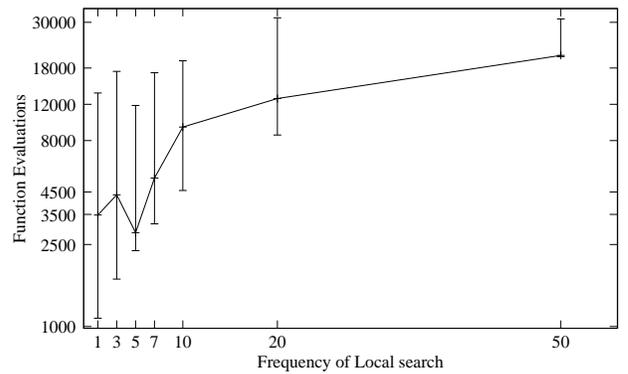


Fig. 1. A parametric study of τ for problem TP3.

D. Problem TP5

This is a non-linear minimization problem which consists of seven design variables and four inequality constraints. The minimum value of variable is at $\mathbf{x}^* = (2.330, 1.951, -0.478, -4.366, -0.624, 1.038, 1.594)^T$ with an objective function value $f = 680.630$. Both g_1 and g_4 are active constraint. In Table III we present the function

evaluations with corresponding objective values. If we compare in terms of best and median performance (with respect to FE) the performance of present approach is better than the previous one but if we compare in terms of function values, earlier approach is slightly better than the present one. This is because of the normalization of constraints. In earlier study normalization was done manually based on problem information; on the other hand constraints are normalized automatically in the present study. From Table IV, we observe that in terms of best performance two existing studies [13], [31] outperform our method, but in terms of mean and worst performance, our results are better than all others.

E. Problem TP6

This is a non-linear problem having five decision variables. Total number of constraints effecting in this problem are six. Constraints g_1 and g_6 are active at this optimum. The minimum is at $\mathbf{x}^* = (78, 33, 29.995, 45, 36.776)^T$ with a function value $f^* = -30665.539$. Even though performance of earlier approach [26] is slightly better than the present study (from Table III), in the earlier case the constraints were normalized manually.

Table IV shows that all four algorithms perform (in terms of optimum function value) equally well. From the results it is clear that the proposed adaptive normalization method performs equally well.

F. Problem TP8

This problem has 10 design variables and eight non-linear inequality constraints. The minimum is at $\mathbf{x}^* = (2.172, 2.364, 8.774, 5.096, 0.991, 1.431, 1.322, 9.829, 8.280, 8.376)^T$ with a objective function value of 24.306. All the six constraints from g_1 to g_6 are active at the minimum point. Table V shows the problem specific normalization value using problem information from our previous study [26] and adaptive normalization value ($V_{j,avg}$) proposed in the present study. In terms of FE (from Table III) earlier approach [26] outperforms current one, but the previous approach never found the optima. In median and worst FE, present approach found the solution optimally.

Figure 2 shows the performance of our present approach for a particular typical simulation run. The algorithm was not able to find any feasible solution up to 10-th generation. The first local search is performed after 10-th generation. From Table IV, in terms of best objective function value both present approach and ATMES [13] performed equally well. In terms of mean performance, our approach is slightly better than other three methods. In terms of worst performance the performance of ATMES is better than all other algorithms.

Since no problem knowledge or trial-and-error approach is used to normalize the constraints in our present approach, the performance is expected to be somewhat worse than previous approaches. However, in most test problems, equivalent or better results are obtained using our approach. This speaks of the generality of our adaptive normalization approach. After

TABLE V
PROBLEM SPECIFIC NORMALIZATION VALUE FROM PREVIOUS STUDY [26] AND ADAPTIVE NORMALIZATION VALUE ($V_{j,avg}$) USING PROPOSED HYBRID APPROACH (AFTER 5 AND 10 GENERATIONS).

	Generation	g_1	g_2	g_3	g_4	g_5	g_6	g_7	g_8
Earlier Approach	Fixed	105	10	12	120	40	14	30	12
Proposed Approach	5	37.5	104.3	67.6	153.3	237.4	101.0	162.8	1515.6
	10	16.3	25.8	0.3	57.9	61.0	42.9	39.3	139.4
	15	11.8	24.1	22.4	21.5	23.4	41.4	17.2	73.4
	20	5.8	12.3	0.4	13.7	10.8	6.5	10.2	1
	25	3.7	4.8	7.1	8.9	4.5	3.5	0.1	1
	30	3.8	4.3	0.4	2.5	4.7	6.1	0.1	1
	35	3.7	3.7	7	18.9	4.7	5.2	1	1
	40	3.3	4.2	0.7	9.0	4.3	3.1	1	1

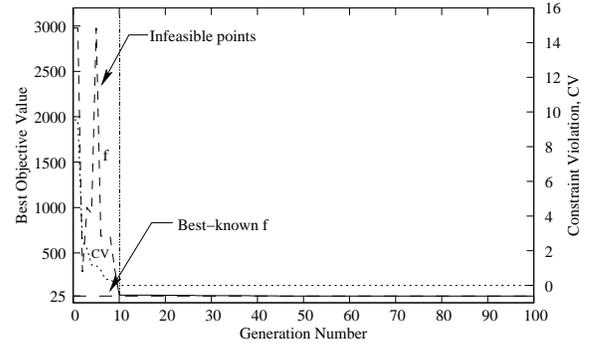


Fig. 2. Objective value reduces with generation for problem TP8.

solving five mathematical test problems we shall now investigate two engineering design problems. These two problems are well studied by evolutionary algorithm researchers.

G. Problem Weld

The objective of the problem is to minimize the fabrication cost of the welded beam [5]. This problem is non-linear with four variables and five constraints. We use a population size of 80. The minimum solution is found to be $\mathbf{x}^* = (0.244368, 8.29150, 6.218586, 0.244368)^T$ with a function value 2.38113. Table III shows that in terms of FE earlier approach is better than present one but that was achieved based on problem specific normalization which is usually less available in real-world scenario.

H. Three-bar truss design

The volume of the truss structure is to be minimized subject to the stress constraints [32]. Figure 3 shows variation of the objective function value with respect to generations. Starting from initial population all population members are feasible. The updation of R with respect to generation is also shown. First local search is started at the end of 5-th generation and near-optimal solution is achieved. The algorithm takes 5 local searches to meet our termination

TABLE IV

COMPARISON OF FUNCTION VALUES ACCOMPLISHED BY THE PROPOSED APPROACH AND THREE EXISTING CONSTRAINT HANDLING APPROACH [13], [30],[31]. THE BEST ONES IN TERMS OF GLOBAL OPTIMUM OR IN TERMS OF BEST PERFORMANCE AMONG THE FOUR ARE SHOWN IN BOLD. IN SOME OF THE PROBLEMS, THE PROPOSED HYBRID APPROACH OUTPERFORMS THE EXISTING THREE APPROACHES.

Problem	Optima (f^*)	Proposed Hybrid Approach			Adaptive Trade-off Model Evolution Strategy (ATMES) [13]		
		Best	Mean	Worst	Best	Mean	Worst
TP3	-15	-15	-14.6	-12	-15	-15	-15
TP4	7,049.280	7,049.588	7,063.311	7,065.348	7,052.253	7,250.437	7,560.224
TP5	680.630	680.631	680.635	680.646	680.630	680.639	680.673
TP6	-30665.539	-30665.539	-30665.539	-30665.539	-30665.539	-30665.539	-30665.539
TP8	24.306	24.306	24.313	24.565	24.306	24.316	24.359
		Simple Multi-membered Evolution Strategy (SMES) [30]			Stochastic Ranking (SR) [31]		
		Best	Mean	Worst	Best	Mean	Worst
TP3	-15	-15	-15	-15	-15	-15	-15
TP4	7,049.280	7,051.903	7,253.047	7,638.366	7,054.316	7,559.192	8,835.655
TP5	680.630	680.632	680.643	680.719	680.630	680.656	680.763
TP6	-30665.539	-30665.539	-30665.539	-30665.539	-30665.539	-30665.539	-30665.539
TP8	24.306	24.327	24.475	24.843	24.307	24.374	24.642

criteria. For this simulation run, value of R at the end of generation 25 is found to be 340.35. Table VI shows comparative results of ours with three existing studies taken from literature [33], [34], [32]. In addition to total function evaluations (FE), NSGA-II and local search FE has shown separately (Table VI). Among statistical results, for best performance our algorithm requires only 1,122 evaluations which is very lower as compared to all, other shown in Table VI.

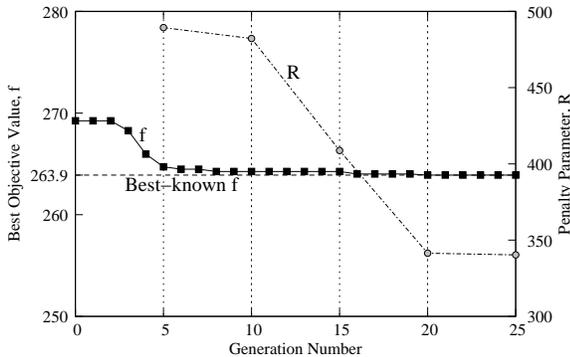


Fig. 3. Function value reduces with generation for three-bar truss design problem..

Table VII, tabulates different values of the penalty parameter with respect to generations. From Figure 3 and Table VII we observe that penalty parameter (R) starts with a high value, then it decreases with generation to a suitable value needed to find the optimal solution.

TABLE VI

COMPARISON OF RESULTS FOR THREE-BAR TRUSS DESIGN.

Algorithms	Best	Median	Worst	FE
ADE[33]	263.89584	263.89584	263.89584	45,000
DSS-MDE[34]	263.89584	263.89584	263.89584	15,000
Ray & Liew[32]	263.89584	263.8989	263.9697	17,610
proposed Method	263.89584	263.89584	263.89585	
FE	1, 122	3, 896	5, 101	
NSGA-II+Local	600 + 522	2,000 + 1,896	2,600 + 2,501	

TABLE VII

ADAPTATION OF PENALTY PARAMETER FOR THREE-BAR TRUSS DESIGN.

Generation	Penalty Parameter
0	0.0000
5	489.3803
10	482.2213
15	408.7732
20	341.4576
25	340.3497

V. CONCLUSIONS

In this paper, we have suggested an adaptive normalization based hybrid bi-objective evolutionary and penalty function based classical optimization procedure where constraints are normalized adaptively during the optimization process. The strength of this method stems from the complimentary combination of the evolutionary and the classical optimization

method which tend to overcome the overall drawbacks of the individual methods when applied alone. The adaptive normalization ensures that no constraint is dominant thereby providing an equal importance to all constraints. We have tested this adaptive normalization based hybrid algorithm on a number of numerical optimization problems and two engineering design problems. To assess the robustness of the proposed procedure 50 independent runs have been performed resulting in a small variance on the obtained solution. Results have been compared with respect to function evaluations with our previous study achieved with a manual scaling of constraints. The optimum function values have been compared with three existing approaches as well. The study indicates that the proposed adaptive normalization procedure performs well and therefore eliminates the need of any manual normalization of constraints. Future work in this area would be to test the adaptive normalization approach on more real-world problems, where explicit mathematical formulation of constraints may not be available. We plan to extend this method to multi-objective constrained optimization problems as well.

ACKNOWLEDGMENT:

The study is funded by Department of Science and Technology, Government of India under SERC-Engineering Sciences scheme (No. SR/S3/MERC/091/2009).

REFERENCES

- [1] R. Fletcher, "An ideal penalty function for constrained optimization," *IMA Journal of Applied Mathematics*, vol. 15, no. 3, pp. 319–342, 1975.
- [2] A. Homaifar, S. H.-V. Lai, and X. Qi, "Constrained optimization via genetic algorithms," *Simulation*, vol. 62, no. 4, pp. 242–254, 1994.
- [3] Z. Michalewicz and C. Z. Janikow, "Handling constraints in genetic algorithms," in *Proceedings of the Fourth International Conference on Genetic Algorithms*, 1991, pp. 151–157.
- [4] J. Joines and C. Houck, "On the use of non-stationary penalty functions to solve nonlinear constrained optimization problems with ga's," in *Evolutionary Computation, 1994. IEEE World Congress on Computational Intelligence., Proceedings of the First IEEE Conference on*. IEEE, 1994, pp. 579–584.
- [5] K. Deb, "An efficient constraint handling method for genetic algorithms," *Computer Methods in Applied Mechanics and Engineering*, vol. 186, no. 2–4, pp. 311–338, 2000.
- [6] C. Coello and A. Carlos, "Use of a self-adaptive penalty approach for engineering optimization problems," *Computers in Industry*, vol. 41, no. 2, pp. 113–127, 2000.
- [7] M. Schoenauer and S. Xanthakis, "Constrained ga optimization," in *Proceedings of the 5th International Conference on Genetic Algorithms*, vol. 580. Citeseer, 1993.
- [8] C. A. C. Coello, "Treating objectives as constraints for single objective optimization," *Engineering Optimization*, vol. 32, no. 3, pp. 275–308, 2000.
- [9] B. Tessema and G. Yen, "A self adaptive penalty function based algorithm for constrained optimization," in *Evolutionary Computation, 2006. CEC 2006. IEEE Congress on*. IEEE, 2006, pp. 246–253.
- [10] P. D. Surry, N. J. Radcliffe, and I. D. Boyd, "A multi-objective approach to constrained optimisation of gas supply networks : The COMOGA method," in *Evolutionary Computing. AISB Workshop*. Springer-Verlag, 1995, pp. 166–180.
- [11] K. Masuda and K. Kurihara, "A constrained global optimization method based on multi-objective particle swarm optimization," *IEEJ Transactions on Electronics, Information and Systems*, vol. 131, no. 5, pp. 990–999, 2011.
- [12] K. Deb, S. Lele, and R. Datta, "A hybrid evolutionary multi-objective and SQP based procedure for constrained optimization," in *Proceedings of the 2nd international conference on Advances in computation and intelligence*. Springer-Verlag, 2007, pp. 36–45.
- [13] Y. Wang, Z. Cai, Y. Zhou, and W. Zeng, "An adaptive tradeoff model for constrained evolutionary optimization," *Evolutionary Computation, IEEE Transactions on*, vol. 12, no. 1, pp. 80–92, 2008.
- [14] E. Mezura-Montes and C. Coello, "Constrained optimization via multi-objective evolutionary algorithms," *Multiobjective Problem Solving from Nature*, pp. 53–75, 2008.
- [15] H. Myung and J. Kim, "Hybrid interior-lagrangian penalty based evolutionary optimization," in *Evolutionary Programming VII*. Springer, 1998, pp. 85–94.
- [16] T. Takahama, S. Sakai, and N. Iwane, "Constrained optimization by the ϵ constrained hybrid algorithm of particle swarm optimization and genetic algorithm," *AI 2005: Advances in Artificial Intelligence*, pp. 389–400, 2005.
- [17] R. Datta, "Constrained engineering design optimization using a hybrid bi-objective evolutionary-classical methodology," *Simulated Evolution and Learning*, pp. 633–637, 2010.
- [18] H. Bernardino, H. Barbosa, and A. Lemonge, "A hybrid genetic algorithm for constrained optimization problems in mechanical engineering," in *Evolutionary Computation, 2007. CEC 2007. IEEE Congress on*. IEEE, 2007, pp. 646–653.
- [19] J. Sha and M. Xu, "Applying hybrid genetic algorithm to constrained trajectory optimization," in *Electronic and Mechanical Engineering and Information Technology (EMEIT), 2011 International Conference on*, vol. 7. IEEE, 2011, pp. 3792–3795.
- [20] J. Zhao, L. Wang, P. Zeng, and W. Fan, "An effective hybrid genetic algorithm with flexible allowance technique for constrained engineering design optimization," *Expert Systems with Applications*, 2011.
- [21] Z. Michalewicz and M. Schoenauer, "Evolutionary algorithms for constrained parameter optimization problems," *Evolutionary computation*, vol. 4, no. 1, pp. 1–32, 1996.
- [22] C. Coello and A. Carlos, "Theoretical and numerical constraint-handling techniques used with evolutionary algorithms: a survey of the state of the art," *Computer methods in applied mechanics and engineering*, vol. 191, no. 11–12, pp. 1245–1287, 2002.
- [23] E. Mezura-Montes, *Constraint-handling in evolutionary optimization*. Springer Verlag, 2009, vol. 198.
- [24] E. Mezura-Montes and C. A. C. Coello, "Constraint-handling in nature-inspired numerical optimization: Past, present and future," *Swarm and Evolutionary Computation*, vol. 1, no. 4, pp. 173 – 194, 2011. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S2210650211000538>
- [25] S. Elsayed, R. Sarker, and D. Essm, "Multi-operator based evolutionary algorithms for solving constrained optimization problems," *Computers & operations research*, vol. 38, no. 12, pp. 1877–1896, 2011.
- [26] K. Deb and R. Datta, "A fast and accurate solution of constrained optimization problems using a hybrid bi-objective and penalty function approach," in *Proceedings of the Congress on Evolutionary Computation (CEC-2010)*, 2010, pp. 1–8.
- [27] K. Deb, *Optimization for Engineering Design: Algorithms and Examples*. New Delhi: Prentice-Hall, 1995.
- [28] D. Keefer and B. Gottfried, "Differential constraint scaling in penalty function optimization," *AIEE Transactions*, vol. 2, no. 4, pp. 281–289, 1970.
- [29] H. Myung and J. Kim, "Evolian: evolutionary optimization based on lagrangian with constraint scaling," in *Evolutionary Programming VI*. Springer, 1997, pp. 177–187.
- [30] E. Mezura-Montes and C. Coello, "A simple multimembered evolution strategy to solve constrained optimization problems," *Evolutionary Computation, IEEE Transactions on*, vol. 9, no. 1, pp. 1–17, 2005.
- [31] T. Runarsson and X. Yao, "Stochastic ranking for constrained evolutionary optimization," *Evolutionary Computation, IEEE Transactions on*, vol. 4, no. 3, pp. 284–294, 2000.
- [32] T. Ray and K. Liew, "Society and civilization: An optimization algorithm based on the simulation of social behavior," *Evolutionary Computation, IEEE Transactions on*, vol. 7, no. 4, pp. 386–396, 2003.
- [33] Y. Ao and H. Chi, "An adaptive differential evolution algorithm to solve constrained optimization problems in engineering design," *Engineering*, vol. 2, no. 1, pp. 65–77, 2010.
- [34] M. Zhang, W. Luo, and X. Wang, "Differential evolution with dynamic stochastic selection for constrained optimization," *Information Sciences*, vol. 178, no. 15, pp. 3043–3074, 2008.