Probabilistic Constraint Handling in the Framework of Joint Evolutionary-Classical Optimization with Engineering Applications

Rituparna Datta, Michael S. Bittermann, Kalyanmoy Deb (IEEE Fellow) and Özer Ciftcioglu

Abstract—Optimization for single main objective with multiple constraints is considered using a probabilistic approach coupled to evolutionary search. In this approach the problem is converted into a bi-objective problem, treating the constraint ensemble as a second objective subjected to multi-objective optimization for the formation of a Pareto front, and this is followed by a local search for the optimization of the main objective function. In this way the convergence to the formation of a Pareto front, and this is followed by a local search algorithms which are proved to be very effective [4], [5], [6]. Even though evolutionary algorithms mainly were developed to solve unconstrained problems, researchers successfully introduced many constraint handling mechanisms to solve constrained optimization problems [7], [8], [9].

A constrained optimization problem is generally formulated as the following non-linear programming (NLP) problem:

\[
\begin{align*}
\text{Minimize} / \text{Maximize} \quad & f(x), \\
\text{Subject to} \quad & g_j(x) \geq 0, \quad j = 1, \ldots, J, \\
& h_k(x) = 0, \quad k = 1, \ldots, K, \\
& x_i^L \leq x_i \leq x_i^U, \quad i = 1, \ldots, n.
\end{align*}
\]

In the NLP problem, \( n \) is the number of variables, \( J \) the number of inequality constraints, and \( K \) the number of equality constraints. The function \( f(x) \) is the objective (cost) function, the \( j \)-th inequality constraint is \( g_j(x) \) and \( h_k(x) \) is the \( k \)-th equality constraint. The range of \( i \)-th variable varies in between \([x_i^L, x_i^U]\). In this work exclusively inequality constraints are treated. To handle an equality constraint the same methodology can be used under the condition that the equality constraint is approximated by a corresponding inequality constraint as follows: \( g_{j+1}(x) = |\epsilon_k - h_k(x)| \geq 0 \), where \( \epsilon_k \) represents a small tolerance for violation of the original equality constraint.

Identifying a solution \( x \) that is both feasible and optimizes the objective at hand is particularly challenging when constraints and objective have a non-linear, non-convex, discrete, or non-differentiable nature. Solving such an engineering problem is formidable challenging when classical optimization algorithms and theorems are used, as these are developed mainly for well-behaved problems [4], [10], [11]. The reason for this is that the non-linearity in the objective and constraint functions gives rise to many local optima. Therefore a classical algorithm, being based on improving a single solution in objective space, is generally bound to be trapped in one of the local optima and not reach the global optimum. Evolutionary computation was found to be able to handle such problems due to its stochastic, population-based nature, where it probes the space of possible solutions at several points simultaneously, without making use of derivative information. The basic principle of an evolutionary algorithm (EA) is to improve individuals of a randomly generated initial population based on combining potentially successful ones. The combination yields new solutions in the vicinity of the original solutions, while the new solutions may outperform the previous ones. This principle turned out to be generally capable for optimization problems with non-linear or discrete objective functions, so that evolutionary algorithms have been used for various engineering applications, e.g. [7], [12], [13].

Deb [8] developed an evolutionary algorithm for constrained...
optimization problems. In this approach, during the tournament selection process, an infeasible solution is always treated as inferior compared to a feasible one, or as inferior compared to a solution that violates the constraints to a lesser extent. In this algorithm, among two infeasible solutions, exclusively the difference in the amount of violating a constraint determines the suitability of a solution, so that the information on objective function value is not taken into account. This may lead to feasible solutions, which however do not yield satisfactory objective function values. With increasing difficulty of a problem, i.e. as the degree of non-linearity of objectives and constraints increases, it becomes apparent that evolutionary computation has limited robustness and effectiveness, which is partially due to ineffective constraint handling in the algorithms. Namely, when the feasible region is sufficiently small compared to the space of possible solutions, then an initial population of an evolutionary algorithm generally does not contain any feasible solution. This lack of information in the population poses a challenge to the algorithm as it aims to reach to the feasible region in the search domain. The problem is to assess the potential of a solution in the population, for reaching the feasible region while satisfying the objective at the same time.

An approach to make use of constraint information and objective information at the same time without merging them by means of a linear combination, is to consider the constrained single objective problem as a bi-objective optimization problem. This means, next to the original single objective \( f(x) \) a measure of overall constraint violation is used as an additional objective to be minimized [14]. This way the search process tolerates infeasible solutions as long as they are Pareto-optimal regarding function value as well as constraint violation. This approach is also referred to as ‘multiobjec-tivization’ [15]. Through multiobjec-tivization the information in the population is exploited more effectively compared to the previously mentioned approaches. Recently an extension of the bi-objective approach was proposed [16] using the reference point approach to focus the search in the vicinity of the constrained minimum solution. Wang et al. [17] proposed an adaptive trade off model (ATM) using bi-objective approach with three phase methodology for handling constraints in evolutionary optimization.

Studies on many different evolutionary algorithms for constrained optimization showed that finding the constrained optimum by an evolutionary algorithm alone is problematic for multiple, non-linear constraints, so that it became appealing to use evolutionary algorithms jointly with a classical local search procedure [18]. Recently, a bi-objective evolutionary optimization strategy has been proposed [9] to estimate the penalty parameter \( R \) for a problem from the obtained two-objective non-dominated front. Based on the information obtained through the bi-objective evolutionary algorithm, an appropriate penalized function is constructed and solved using a classical local search method. Another joint evolutionary-classical method used gradient information of the constraints to repair the solutions which are not feasible [19]. Yet another joint approach used the combination of Nelder-Mead simplex search with evolutionary algorithm [20]. More hybrid constraint handling studies can be found in [21], [22].

Presumably the most popular constraint handling approach is known as the penalty function approach, which was originally developed for the classical optimization methodologies. A penalty function penalizes a solution by worsening the fitness of a solution, when it violates constraints. This penalization is accomplished by adding a value to a solution’s objective function value in proportion to the amount of constraint violation. The proportionality factor is known as the penalty parameter, which balances the relative importance between constraint violation and objective function value. That is, in the penalty method a constrained optimization problem is converted into an unconstrained problem. This approach is very popular presumably due to the simplicity of the concept and ease of implementation. However, it is clearly noted that fixing a penalty parameter implies that the relative importance among constraint violation and satisfaction of the objective function should be known, which is a problematic issue in general due to non-linearity inherent to objective and constraint functions. To overcome this issue to some extend, Coello [23] proposed a self-adaptive penalty approach by using a co-evolutionary model to adapt the penalty factors.

In this paper an approach is proposed where constraint violation is treated in probabilistic terms. This way, the population based nature of the evolutionary algorithm is exploited, in order to estimate the relative significance of violating a constraint in perspective with the other constraints.

The organization of the paper is as follows. In the next section a probabilistic method developed in this work is described. Thereafter its effectiveness is verified by means of applications of the associated algorithm for solving a number of mathematical test problems, an engineering design problem, and a robotics problem. This is followed by conclusions.

II. Evolutionary Optimization and Probabilistic Constraint Handling

The bi-objective joint evolutionary-classical method used in this work is a combination of a multiobjectivized EA and the penalty function approach for inequality constraints. The method has been described elsewhere [9]. Additionally, in this work a probabilistic model is developed associated with the Pareto front formed by the objective function and the constraint violation. Based on this model the penalty parameter over all constraint violation is computed in a natural way, so that commensurate weighting of the constraint violation is maintained throughout the optimizations process. The anticipated outcome is the effective solution for the problem at hand with major improvements compared to the several approaches mentioned above. For convenience of readers the essential issues of the bi-objective joint approach are pointed out in the following section before addressing the novel probabilistic constraint handling technique.
A. Bi-objective approach

The working principle of evolutionary-classical algorithm proposed in this work is based on bi-objective method of handling constrained single objective optimization problem, where a penalty function approach is used. In both evolutionary and classical search, a novel probabilistic constraint handling approach is employed that will be described in the next subsection. The algorithm is described as follows, clarifying the role of its evolutionary, probabilistic and classical components.

First, the generation counter is set at \( t = 0 \).

Step 1: The evolutionary component is an elitist, non-dominated-sorting based multi-objective genetic algorithm (NSGA-II [6]). It is applied to the bi-objective optimization problem [9]. This means the Pareto-optimal solutions in the objective space formed by function value and constraint violation are to be identified. That is, the bi-objective problem is defined as follows:

\[
\begin{align*}
\text{Minimize} & \quad f(x), \\
\text{Minimize} & \quad V(x), \\
\text{subject to} & \quad x^L_i \leq x_i \leq x^U_i, \quad i = 1, \ldots, n.
\end{align*}
\]

where \( n \) denotes the number of variables, \( x^L_i \) and \( x^U_i \) are the lower and upper variable bounds of the \( i \)-th component of \( x \) respectively, and \( V(x) \) denotes the overall constraint violation whose computation we will describe in the next section.

Step 2: If \( t > 0 \) and \( (t \mod \tau) = 0 \), the penalty parameter \( R \) is obtained from the current non-dominated front as follows [9]. A cubic curve is fitted for the non-dominated points (\( f = a + b \times V + c \times V^2 + d \times V^3 \)). This way the penalty parameter is estimated by finding the slope at \( V=0 \), that is \( R = -b \). Since this is a lower bound on \( R \), after some trial-and-error experiments on standard test problems, twice this value is chosen as \( R \), i.e. \( R = -2b \) [9].

Step 3: Thereafter, using \( R \) computed in Step 2, the following local search problem is solved starting with the most feasible solution, i.e. the solution having minimum \( V(x) \), as given by

\[
\begin{align*}
\text{Minimize} & \quad P(x) = f(x) + R \times V(x), \\
\text{subject to} & \quad x^L_i \leq x_i \leq x^U_i.
\end{align*}
\]

The solution from the local search is denoted by \( \bar{x} \).

Step 4: The algorithm is terminated in case \( \bar{x} \) is feasible, and the difference between two consecutive local search solutions is smaller than a small number \( \delta_f \). In the applications of this paper, \( \delta_f = 10^{-4} \) is used. Then \( \bar{x} \) is identified as the optimized solution. Else, \( t \) is incremented by one, and we continue with Step 1.

It is noted that due to Step 2, the penalty parameter \( R \) is not a user-tunable parameter as it is determined from the obtained non-dominated front. For the local search procedure in Step 3 Matlab’s `fmincon()` procedure with reasonable parameter settings is used to solve the penalized function.

B. Probabilistic constraint handling

Typically the penalty function approach is used to solve the following unconstrained problem [9].

\[
\begin{align*}
\text{Minimize} & \quad P(x, R) = f(x) + \sum_{j=1}^{J} R_j(g_j(x)).
\end{align*}
\]

where \( f(x) \) is the objective function to be minimized; \( \langle \alpha \rangle \) is the bracket operator and is equal to \( -\alpha \), if \( \alpha(0 \text{ and zero otherwise); } g_j(x) \text{ represents a general violation of the } j \text{-th constraint. } R_j \text{ is the penalty parameter for } j \text{-th constraint. Since } \langle g_j(x) \rangle \text{ is continually tried to be vanishing during the minimization process, probability density value of } \langle g_j(x) \rangle \text{ is highest at zero and its values gradually diminish. With this information we can confidently surmise a probabilistic model for this probability density (pdf) which is exponential pdf given by}

\[
\begin{align*}
f_j(y) = \lambda \frac{e^{-\lambda y}}{y^\alpha},
\end{align*}
\]

where \( \lambda \) is the decay parameter. If we denote \( \langle g_j(x) \rangle \) by \( v_j(x) \), namely

\[
\langle g_j(x) \rangle = v_j(x).
\]

the pdf in (5) becomes

\[
\begin{align*}
f_{j,\langle \alpha \rangle}(v_j) = \lambda \frac{e^{-\lambda v_j}}{v_j^\alpha}.
\end{align*}
\]

The mean value of the exponential pdf function is equal to \( \lambda^{-1} \). During the evolutionary search \( \langle g_j(x) \rangle \) is a general form of violation which applies to any member \( s \) of the population although this is not explicitly denoted. In explicit form, we can write

\[
\begin{align*}
f_{j,\langle \alpha \rangle}(v_j) = \lambda_j \frac{e^{-\lambda_j v_j}}{v_j^\alpha}.
\end{align*}
\]

We can characterize the exponential pdf function according to the constraint \( j \) simply by equating the mean value of the violations to the mean of the exponential pdf, namely

\[
\lambda_j = \frac{1}{v_j}.
\]

so that (3) becomes

\[
\begin{align*}
f_{j,\langle \alpha \rangle}(v_j) = \frac{1}{v_j^\alpha} e^{-\lambda_j v_j/(v_j)^\alpha}.
\end{align*}
\]

One should note that the mean of the exponential probability density of \( v_j \) is equivalent to the mean of a uniform probability density applied to the violations \( v_j \). Therefore the mean of the exponential density function is estimated by taking the mean of the violations which are from a uniform probability density and they are independent. This is closely connected to exponential averaging [24]. Variation of the exponential pdf for different decay parameters is shown in figure 1.

The importance of (10) can be seen in the following way. Since a violation \( v_j \) spans all the violation starting from zero up to the point \( v_j \), the probability of the violation is expressed
as cumulative distribution function whose implication is easy to comprehend by considering the extremes. The cumulative distribution function of (10) is given by the following equation

\[ p(v_j) = \frac{1}{v_j} \int_{0}^{v_j} \exp\left(-\frac{v_j}{\sigma_j}\right) dv_j = 1 - \exp\left(-\frac{v_j}{\sigma_j}\right). \]  

(11)

For \( v_j = 0 \) violation is zero and for \( v_j = \infty \), violation is 1.0, i.e., 100%. The variation of \( p(v_j) \) vs \( v_j \) with respect to the mean of \( v_j \) is shown in figure 2.

Fig. 1. Variation of exponential pdf for different decay constants versus \( v_j \).

The function expressed by equation 11 tells us how probable it is that a solution with equal or less degree of violation is expected to occur. That is, it is the probability to find an equal or better solution. This can be defined as the degree of solution-unimportance. The smaller the probability \( p(v) \) the more important the solution is. This is because, we want the search process to arrive at the region where the probability to find equal or better solutions is low. In the extreme case the probability of the violation is zero. Consequently this is the most important, i.e. ideal solution. We value a solution with a low probability as important, because it is relatively close to the most important point where the violation is zero, and the algorithm tries to find solutions to confirm this. For a solution with a high probability \( p(v) \) the occurrence of this solution is quite common, so that such a solution is not significant for the search process to reach the feasible region. Namely it is relatively far from the point where the violation is zero, and to find a better solution than the one at hand is foregone conclusion. The corresponding chromosomes are to be favoured according to their degree of importance namely according to their closeness to the point where the violation is zero in the evolutionary search process. It is interesting to note that, from the figures, for zero constraint violation the probability density is maximum, while the corresponding probability, namely to find an equal or better solution, is minimum, i.e. zero. In (4) denoting

\[ \langle g_j(x) \rangle = v_j(x). \]  

(12)

we can write

\[ R_j \langle g_j(x) \rangle = R_j v_j(x) = R_j(v_j(x). \]  

(13)

where \( R \) is the penalty parameter common for all the constraints; \( r_j \) is given as a non-linear function of \( v_j \) in (14) in a general form

\[ r_j = f(v_j). \]  

(14)

so that, we obtain

\[ r_j v_j = f(v_j)v_j = p(v_j). \]  

(15)

Using this result in (4), we write

\[ R_j v_j = R p(v_j), \]
\[ \sum_{j=1}^{J} R_j v_j = R \sum_{j=1}^{J} p(v_j). \]  

(16)

where \( J \) is the number of constraints; \( R \) is a common penalty parameter which is determined as described in the preceding section. The probability \( p(v_j) \) controls the common penalty parameter which varies theoretically between zero and minus infinity. Equation (16) points out two important items.

1) \( R_j(v_j)\langle g_j(x) \rangle = R_j(v_j)v_j \) is explicitly defined by a single non-linear function \( p(v_j) \).

2) The entire \( v_j \) region is transformed between zero and one, where probable stiffness of the constraints is naturally and effectively handled. Especially in the presence of stiff violations the determination of \( R_j \) contains much uncertainty, since it is computed as a slope at the point where constraint violations vanish. In this case the \( R_j \) is relatively small and \( v_j \) is relatively large so that the product \( R_j v_j \) is precarious. This is a typical manifestation of stiff constraint. This is sketched in figure 3. Figure 3a presents the total overview about the Pareto front for stiff conditions. The cubic approximation is carried out to impose curve fitting on the Pareto front. The slope of the tangent is computed at the point where cubic approximation and the vertical axis intersect. Figure 3b is cubic approximation merged with the Pareto front as a single curve.

In the new approach the sum \( \sum_{j=1}^{J} p(v_j) \) has a well defined outcome which corresponds to a well established \( R \) so that the product of these two yields a stable outcome. The improvement by the new approach is illustrated in figure 4. Figure 4a presents the overview about the Pareto front in stiff constraint
in the algorithm. For every problem the algorithm was run 25 times from different initial populations. As result the number of function evaluations is presented in the form of best, median and worst number of evaluations.

A. Test problems

In this section, we are providing the problem formulation of both the mathematical and engineering design test problems. The probabilistic-based hybrid algorithm is applied to four difficult test problems, that are named g01, g07, g18, and g24 in [25]. The mathematical formulation for each problem is given with the corresponding best-known optimum solution. In Table I the function evaluations needed by the probabilistic-based hybrid approach are presented and compared with an existing approach taken from the literature [27]. From the results of the applications it is seen that our approach outperforms the existing one. Considering the average amount of function evaluations, the existing approach requires more evaluation by factor 21.1 for 1; factor 5.8 for problem 2; factor 7.6 for problem 3; and factor 2.4 for problem 4.

B. Test problem description

A. Problem 1

The problem is given as follows:

\[
\begin{align*}
\text{min.} & \quad f(x) = 5 \sum_{i=1}^{4} x_i - 5 \sum_{i=1}^{4} x_i^2 + 5 \sum_{i=5}^{13} x_i^2, \\
\text{s.t.} & \quad g_1(x) \equiv 2x_1 + 2x_2 + x_{10} + x_{11} - 10 \leq 0, \\
& \quad g_2(x) \equiv 2x_1 + 2x_3 + x_{10} + x_{12} - 10 \leq 0, \\
& \quad g_3(x) \equiv 2x_2 + 2x_3 + x_{11} + x_{12} - 10 \leq 0, \\
& \quad g_4(x) = -8x_1 + x_{10} \leq 0, \\
& \quad g_5(x) = -8x_2 + x_{11} \leq 0, \\
& \quad g_6(x) = -8x_3 + x_{12} \leq 0, \\
& \quad g_7(x) = -2x_4 - x_5 + x_{10} \leq 0, \\
& \quad g_8(x) = -2x_6 - x_7 + x_{11} \leq 0, \\
& \quad g_9(x) = -2x_8 - x_9 + x_{12} \leq 0,
\end{align*}
\]

(17)

where \(0 \leq x_i \leq 1\) for \(i = 1, \ldots, 9\), \(0 \leq x_i \leq 100\) for \(i = 10,11,12\), and \(0 \leq x_{13} \leq 1\). The minimum point is \(x^* = (1, 1, 1, 1, 1, 1, 1, 3, 3, 3, 1)^T\) and \(f(x^*) = -15\).

In Figure 5 for a typical simulation among the 25 runs of problem 1, the history of the best objective value of the population and the corresponding constraint violation value are shown.

B. Problem 2

The problem is given as follows:

\[
\begin{align*}
\text{min.} & \quad f(x) = x_1^2 + x_2^2 + x_1x_2 - 14x_1 - 16x_2 + (x_3 - 10)^2 \\
& \quad + 4(x_4 - 5)^2 + (x_5 - 3)^2 + 2(x_6 - 1)^2 + 5x_7^2 \\
& \quad + 7(x_8 - 11)^2 + 2(x_9 - 10)^2 + (x_{10} - 7)^2 + 45, \\
\text{s.t.} & \quad g_1(x) \equiv -105 + 4x_1 + 5x_2 - 3x_3 + 9x_4 \leq 0, \\
& \quad g_2(x) \equiv 10x_1 - 8x_2 - 17x_3 + 2x_4 \leq 0, \\
& \quad g_3(x) \equiv -8x_1 + 2x_2 + 5x_3 - 2x_5 - 12 \leq 0, \\
& \quad g_4(x) \equiv 3(x_1 - 2)^2 + 4(x_2 - 3)^2 + 2(x_3 - 2)^2 - 2x_4 - 40 \leq 0, \\
& \quad g_5(x) \equiv 5x_1^2 + 8x_2 + (x_3 - 6)^2 - 9x_4 - 30 \leq 0, \\
& \quad g_6(x) \equiv (x_7 + 2(x_9 - 2)^2 - 2x_1x_2 + 14x_6 - 6x_7 \leq 0, \\
& \quad g_7(x) \equiv 0.5(x_1 - 8)^2 + 2x_2 - 4y^2 + 3x_3^2 - x_8 - 30 \leq 0, \\
& \quad g_8(x) \equiv -3x_1 + 6x_2 + 12(x_9 - 8)^2 - 7x_{10} \leq 0, \\
& \quad -10 \leq x_i \leq 10, \quad i = 1, \ldots, 10.
\end{align*}
\]
The best-reported optimum lies at \( x^* = (2.3295201974776, 3.17849307411774)^T \) and the corresponding objective value is \( f(x^*) = -5.508013 \).

### E. Problem welded beam design

The problem is given as follows (\( x = (h, l, t, b)^T \)):

\[
\begin{align*}
\text{min. } & \quad f_1(x) = 1.10471 h^2 l + 0.04811 l (14.0 + l), \\
\text{s. t. } & \quad g_1(x) \equiv 13,600 - \tau(x) \geq 0, \\
& \quad g_2(x) \equiv 30,000 - \sigma(x) \geq 0, \\
& \quad g_3(x) \equiv b - h \geq 0, \\
& \quad g_4(x) \equiv P_r(x) - 6,000 \geq 0, \\
& \quad g_5(x) \equiv 0.25 - \delta(x) \geq 0, \\
& \quad 0.125 \leq h, b \leq 5, \\
& \quad 0.1 \leq l, t \leq 10,
\end{align*}
\]

where,

\[
\begin{align*}
\tau(x) &= \sqrt{(\tau')^2 + (\tau'')^2 + (l\tau' \tau'')/\sqrt{0.25(l^2 + (h + t)^2)}}, \\
\tau' &= \frac{6,000}{\sqrt{2hl}}, \\
\tau'' &= \frac{6,000(14 + 0.5l) \sqrt{0.25(l^2 + (h + t)^2)}}{2[0.707hl(l^2/12 + 0.25(h + t)^2)]}, \\
\sigma(x) &= \frac{504,000}{t^2b}, \\
\delta(x) &= \frac{2.1952}{t^2b}, \\
P_r(x) &= 64,746.022(1 - 0.0282346t)tb^3.
\end{align*}
\]

Reduction of function value with generation is shown in Figure 6.

---

The best-reported minimum is at \( x^* = (2.172, 2.364, 8.774, 5.096, 0.991, 1.431, 1.322, 9.829, 8.280, 8.376)^T \) with a function value 24.306.

### C. Problem 3

The problem is given as follows:

\[
\begin{align*}
\min. \quad & f(x) = -0.5 (x_1 x_4 - x_3 x_9 + x_3 x_9 - x_5 x_9) \\
\text{s. t. } & g_1(x) \equiv x_1^2 + x_2^2 - 1 \leq 0, \\
& g_2(x) \equiv x_3^2 - 1 \leq 0, \\
& g_3(x) \equiv x_4^2 + (x_5 - x_9)^2 - 1 \leq 0, \\
& g_4(x) \equiv (x_1 - x_5)^2 + (x_2 - x_6) \leq 0, \\
& g_5(x) \equiv (x_3 - x_7)^2 + (x_4 - x_8)^2 \leq 0, \\
& g_6(x) \equiv (x_3 - x_7)^2 + (x_4 - x_8)^2 - 1 \leq 0, \\
& g_7(x) \equiv (x_3 - x_7)^2 + (x_4 - x_8)^2 \leq 0, \\
& g_8(x) \equiv (x_1 - x_5)^2 + (x_2 - x_6)^2 \leq 0, \\
& g_9(x) \equiv x_2 (x_3 - x_9)^2 - 1 \leq 0, \\
& g_{10}(x) \equiv x_3 x_9 - x_1 x_4 \leq 0, \\
& g_{11}(x) \equiv -x_3 x_9 \leq 0, \\
& g_{12}(x) \equiv x_5 x_9 \leq 0, \\
& g_{13}(x) \equiv x_6 x_9 - x_5 x_8 \leq 0, \\
& -10 \leq x_i \leq 10 \text{ for } i = 1, \ldots, 8, \quad 0 \leq x_9 \leq 20.
\end{align*}
\]

The best-reported constrained minimum lies at \( x^* = (-0.657776, -0.153419, 0.323414, -0.946258, -0.657776, -0.753213, 0.323414, -0.346463, 0.599795) \) with an objective value of \( f(x^*) = -0.866025 \).

### D. Problem 4

The problem is given as follows:

\[
\begin{align*}
\min. \quad & f(x) = -x_1 - x_2, \\
\text{s. t. } & g_1(x) \equiv -2x_1^4 + 8x_1^3 - 8x_1^2 + x_2 - 2 \leq 0, \\
& g_2(x) \equiv -4x_1^3 + 32x_1^2 - 88x_1^2 + 96x_1 + x_2 - 36 \leq 0, \\
& 0 \leq x_1 \leq 3, \quad 0 \leq x_2 \leq 4.
\end{align*}
\]

The best-reported optimum lies at \( x^* = (2.32952019747762, 3.17849307411774) \) and the corresponding objective value is \( f(x^*) = -5.508013 \).
The minimization of robot gripper design is subject to the geometric and force constraints:

\begin{align}
g_1(x) & \equiv Y_{\text{min}} - y(x, Z_{\text{max}}) \geq 0, \\
g_2(x) & \equiv y(x, Z_{\text{max}}) \geq 0, \\
g_3(x) & \equiv y(x, 0) - Y_{\text{max}} \geq 0, \\
g_4(x) & \equiv Y_G - y(x, 0) \geq 0, \\
g_5(x) & \equiv (a + b)^2 - l^2 - e^2 \geq 0, \\
g_6(x) & \equiv (l - Z_{\text{max}})^2 + (a - e)^2 - b^2 \geq 0, \\
g_7(x) & \equiv l - Z_{\text{max}} \geq 0.
\end{align}

Fig. 7. A sketch of robot Gripper-I. Figure taken from [26]

The objective is to minimize the difference between maximum and minimum force in the gripper. Minimize

\[ f(x) = \max_z F_k(x, z) - \min_z F_k(x, z). \]  

Table II shows the best, median and worst objective function values obtained using probabilistic constraint handling methodology and compared with an existing approach taken from literature [26]. It is noted that due to the multi-modality of the robotics problems at hand the algorithm is executed without local search component involved in order to ensure robustness of the algorithm. Despite this, as for the mathematical problems, also for the robotics problems the prob-
TABLE II
COMPARISON OF RESULTS FOR THE ROBOT GRIPPER DESIGN.

<table>
<thead>
<tr>
<th>Method</th>
<th>Best</th>
<th>Median</th>
<th>Worst</th>
</tr>
</thead>
<tbody>
<tr>
<td>Teaching-learning-based</td>
<td>4.247644</td>
<td>4.93770095</td>
<td>8.141973</td>
</tr>
<tr>
<td>optimization (TLBO) [26]</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Artificial Bee Colony</td>
<td>4.247644</td>
<td>5.086611</td>
<td>6.784631</td>
</tr>
<tr>
<td>optimization [26]</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Proposed methodology</td>
<td>0.592738</td>
<td>0.623837</td>
<td>0.673636</td>
</tr>
</tbody>
</table>

A probabilistic based approach performed better than the previous one. Namely the existing approach used about 10 times more function evaluations in average compared to the probabilistic hybrid approach, and the probabilistic based approach reached a solution that is significantly better compared to the best known optimum up till now. This improved performance of robot gripper design optimization is indicated in Table II.

IV. CONCLUSIONS

A probabilistic constraint handling approach for evolutionary-classical constrained optimization is presented to deal effectively with challenging constrained single objective optimization problems. The novelty of the probabilistic approach is to handle the stiffness present among the constraints. This is accomplished by employing a bi-objective constraint handling approach based on multi-objectivization of the constrained single objective problem. This entails that the information obtained from the evolutionary algorithm is used effectively in order to bring the population near to the feasible region with pressure. This is accomplished by using the constraint violation information in terms of the degree of solution importance that is quantified by a probability. The evolutionary search provides the information for estimation of the appropriate penalty parameter for the local search, so that it is able to arrive at the exact global optimum. From the applications of the approach on several mathematical test problems, a welded beam design and a robotics problem, it is noted that the probabilistic hybrid approach outperforms existing approaches by significant factors in terms of the amount of function evaluations required to reach the optima, reported in Table I as well as with respect to best known optima, reported in Table II, where it is seen that the probabilistic hybrid algorithm yields very accurate results as it generally arrives at the best-known optimum with exactness.

ACKNOWLEDGMENTS

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