WCCI 2012 IEEE World Congress on Computational Intelligence
June, 10-15, 2012 - Brisbane, Australia IEEE CEC

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 multi 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 process a novel probabilistic modeling is applied to the constraint ensemble, so that the stiff constraints are effectively taken care of, while the model parameter is adaptively determined during the evolutionary search. In this way the convergence to the solution is significantly accelerated and an accurate solution is established. The improvements are demonstrated by means of example problems including comparisons with the standard benchmark problems, the solutions of which are reported in the literature.**

I. Introduction

MANY real life problems involve decision makings and choices based on some compromises. Among the possible compromises to select the ones which are as good as possible is a natural way to proceed. In general this process is complex and it can be defined as optimization. The complexity arises due to a single or many conflicting objectives. Most real-life problems in science, engineering and optimization consist of one or many linear, non-linear, non-convex, and discontinuous constraints, which come into picture mainly due to some physical limitations or functional requirements to satisfy. Constraints can be subdivided into inequality type or equality type, but the challenge is to satisfy all constraints to make the solution be feasible. To solve the optimization problems traditionally a number of methods are developed in the realm of mathematics with their associated merits and limitations [1], [2], [3]. With modern advancements in science and engineering, from last two decades the solution of optimization problems are sought by means of evolutionary 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].

Rituparna Datta is doctoral student at Department of Mechanical Engineering, Indian Institute of Technology Kanpur, PIN 208016, India (email: rdatta@iitk.ac.in)

Michael S. Bittermann is post doctoral research scholar at at Chair of Design Informatics (TO&I), Department of Building Technology, Delft University of Technology, 2628 BL Delft (email: m.s.bittermann@tudelft.nl)

Kalyanmoy Deb (*Fellow, IEEE*) is professor, Department of Mechanical Engineering, Indian Institute of Technology Kanpur, PIN 208016, India (email: deb@iitk.ac.in)

Özer Ciftcioglu (*Sr. Member, IEEE*) is professor at Chair of Design Informatics (TO&I), Department of Building Technology, Delft University of Technology, 2628 BL Delft (email: O.Ciftcioglu@tudelft.nl)

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

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

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 $f(\boldsymbol{x})$ is the objective (cost) function, the j-th inequality constraint is $g_j(\boldsymbol{x})$ and $h_k(\boldsymbol{x})$ 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: $g_{J+k}(x) = |\epsilon_k - h_k(x)| \geq 0$, where ϵ_k represents a small tolerance for violation of the original equality constraint.

Identifying a solution $\boldsymbol{x}$ 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 formidably 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 generically 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

U.S. Government work not protected by U.S. copyright

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 aparent 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 $f(\boldsymbol{x})$ 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 'multiobjectivization' [15]. Through multiobjectivization 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:

$$\begin{array}{ll} \text{Minimize} & f(\boldsymbol{x}), \\ \text{Minimize} & \text{V}(\boldsymbol{x}), \\ \text{subject to} & x_i^{(L)} \leq x_i \leq x_i^{(U)},\ i = 1, .., n. \end{array} \tag{2}$$

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

Step 2: If $t > 0$ and ((t mod τ) = 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 $V(\boldsymbol{x})$, as given by

$$\begin{array}{ll} \text{Minimize} & P(\boldsymbol{x}) = f(\boldsymbol{x}) + R \times V(\boldsymbol{x}), \\ & x_i^{(L)} \leq x_i \leq x_i^{(U)}. \end{array} \tag{3}$$

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 δ_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].

$$\text{Minimize} \quad P(\boldsymbol{x}, R) = f(\boldsymbol{x}) + \textstyle\sum_{i=1}^{J} R_j \langle g_j(\boldsymbol{x}) \rangle. \tag{4}$$

where $f(\boldsymbol{x})$ is the objective function to be minimized; $\langle \alpha \rangle$ is the bracket operator and is equal to $-\alpha$, if $\alpha \langle 0$ and zero otherwise; $g_j(\boldsymbol{x})$ represents a general violation of the j-th constraint. R_j is the penalty parameter for j-th constraint. Since $\langle g_j(\boldsymbol{x}) \rangle$ is continually tried to be vanishing during the minimization process, probability density value of $\langle g_j(\boldsymbol{x}) \rangle$ 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

$$f_\lambda(y) = \lambda\, e^{-\lambda y}. \tag{5}$$

where λ is the decay parameter. If we denote $\langle g_j(\boldsymbol{x}) \rangle$ by $v_j(\boldsymbol{x})$, namely

$$\langle g_j(\boldsymbol{x}) \rangle = v_j(\boldsymbol{x}). \tag{6}$$

the pdf in (5) becomes

$$f_{\lambda_j}(v_j) = \lambda\, e^{-\lambda_j v_j}. \tag{7}$$

The mean value of the exponential pdf function is equal to λ_j^{-1}. During the evolutionary search $\langle g_j(\boldsymbol{x}) \rangle$ 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

$$f_{\lambda_j}(v_{j,s}) = \lambda_j\, e^{-\lambda_j v_{j,s}}. \tag{8}$$

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

$$\lambda_j = \tfrac{1}{\bar{v}_j}. \tag{9}$$

so that (3) becomes

$$f_{\lambda_j}(v_j) = \tfrac{1}{\bar{v}_j} e^{-v_j/\bar{v}_j}. \tag{10}$$

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

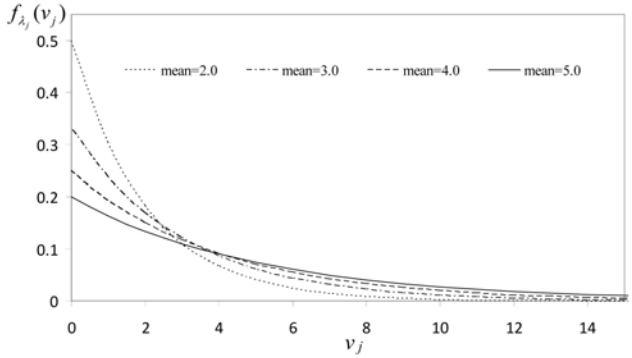

Fig. 1. Variation of exponential pdf for different decay constants versus v_j.

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

$$p(v_j) = \frac{1}{\bar{v}_j} \int_0^{v_j} e^{-\frac{v_j}{\bar{v}_j}} \, dv_j = 1 - e^{-\frac{v_j}{\bar{v}_j}}. \quad (11)$$

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.

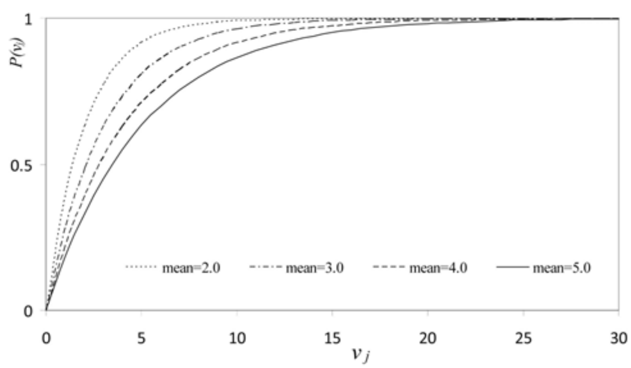

Fig. 2. Variation of $p(v_j)$ versus v_j for various mean values of $p(v_j)$.

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

$$\langle g_j(\boldsymbol{x}) \rangle = v_j(\boldsymbol{x}). \quad (12)$$

we can write

$$R_j \langle g_j(\boldsymbol{x}) \rangle = R_j v_j(\boldsymbol{x}) = R r_j(\boldsymbol{x}) v_j(\boldsymbol{x}). \quad (13)$$

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

$$r_j = f(v_j). \quad (14)$$

so that, we obtain

$$r_j v_j = f(v_j) v_j = p(v_j). \quad (15)$$

Using this result in (4), we write

$$\begin{aligned} & R_j v_j = R p(v_j), \\ & \textstyle\sum_{j=1}^{J} R_j v_j = R \sum_{j=1}^{J} p(v_j). \end{aligned} \quad (16)$$

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) $R_j(v_j)\langle g_j(\boldsymbol{x}) \rangle = R_j(v_j) v_j$ is explicitly defined by a single nonlinear 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 $\sum_1^J p(v_j)$ 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

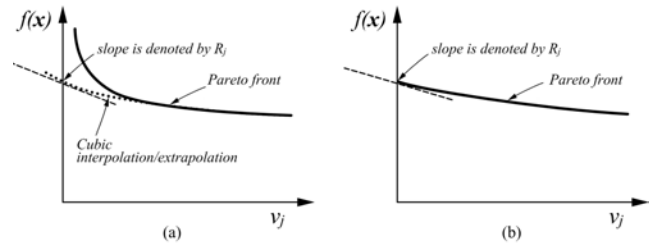

Fig. 3. Sketch of computing slope of the tangent in the presence of stiff constraint conditions. (a) Total overview of the Pareto front together with cubic interpolation & extrapolation to determine the slope of the tangent. (b) Cubic approximation merged with the Pareto front as a single curve (a) at the region where v_j is close to the origin.

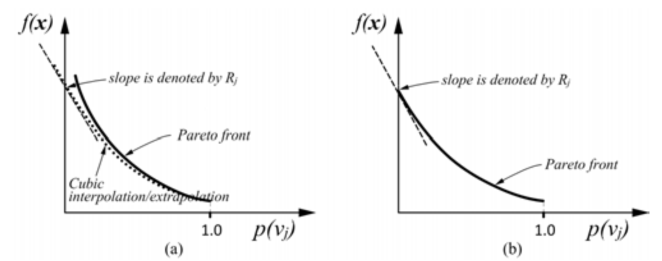

Fig. 4. Sketch of computing slope of the tangent in the presence of stiff constraint conditions. (a) Total overview of the Pareto front together with cubic interpolation & extrapolation to determine the slope of the tangent in the new approach. (b) Cubic approximation merged with the Pareto front as a single curve (a) at the region where v_j is close to the origin.

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 4b shows the cubic polynomial merged with the Pareto front and they are represented as a single curve.

It is noteworthy to mention that the change of the shape of the *degree of solution-unimportance* from generation to generation implies that the search processes uses the most actual violation information represented by the statistical properties of the population to grade the suitability of solutions, and the slope of the violation tends to go minus infinity. This way the local search benefits from the evolutionary algorithm having suitable starting point for the local search.

III. Applications

In order to study the effectiveness of the probabilistic-based bi-objective hybrid algorithm, the algorithm is applied to a number of well-known standard mathematical problems taken from constrained optimization literature [25], as well as a constrained welded beam design [8] and a robotics problem [26]. The problems have previously been tackled by researchers using different approaches.

The parameter values for the EA used are: population size = 100, simulated binary crossover (SBX) probability= 0.9, SBX index = 10, polynomial mutation probability = $1/n$, where n denotes the amount of decision variables, and mutation index = 100. It is to note that the termination criterion is described 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 mathetical 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:

$$\begin{array}{rl}
\text{min.} & f(x) = 5\sum_{i=1}^{4} x_i - 5\sum_{i=1}^{4} x_i^2 + 5\sum_{i=5}^{13} x_i,\\
\text{s.t.} & g_1(x) \equiv 2x_1 + 2x_2 + x_{10} + x_{11} - 10 \le 0,\\
& g_2(x) \equiv 2x_1 + 2x_3 + x_{10} + x_{12} - 10 \le 0,\\
& g_3(x) \equiv 2x_2 + 2x_3 + x_{11} + x_{12} - 10 \le 0,\\
& g_4(x) \equiv -8x_1 + x_{10} \le 0,\\
& g_5(x) \equiv -8x_2 + x_{11} \le 0,\\
& g_6(x) \equiv -8x_3 + x_{12} \le 0,\\
& g_7(x) \equiv -2x_4 - x_5 + x_{10} \le 0,\\
& g_8(x) \equiv -2x_6 - x_7 + x_{11} \le 0,\\
& g_9(x) \equiv -2x_8 - x_9 + x_{12} \le 0,
\end{array} \tag{17}$$

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

$$\begin{array}{rl}
\text{min.} & f(x) = x_1^2 + x_2^2 + x_1x_2 - 14x_1 - 16x_2 + (x_3 - 10)^2\\
& +4(x_4 - 5)^2 + (x_5 - 3)^2 + 2(x_6 - 1)^2 + 5x_7^2\\
& +7(x_8 - 11)^2 + 2(x_9 - 10)^2 + (x_{10} - 7)^2 + 45,\\
\text{s.t.} & \\
& g_1(x) \equiv -105 + 4x_1 + 5x_2 - 3x_7 + 9x_8 \le 0,\\
& g_2(x) \equiv 10x_1 - 8x_2 - 17x_7 + 2x_8 \le 0,\\
& g_3(x) \equiv -8x_1 + 2x_2 + 5x_9 - 2x_{10} - 12 \le 0,\\
& g_4(x) \equiv 3(x_1 - 2)^2 + 4(x_2 - 3)^2 + 2x_2^3 - 7x_4 - 120 \le 0,\\
& g_5(x) \equiv 5x_1^2 + 8x_2 + (x_3 - 6)^2 - 2x_4 - 40 \le 0,\\
& g_6(x) \equiv x_1^2 + 2(x_2 - 2)^2 - 2x_1x_2 + 14x_5 - 6x_6 \le 0,\\
& g_7(x) \equiv 0.5(x_1 - 8)^2 + 2(x_2 - 4)^2 + 3x_5^2 - x_6 - 30 \le 0,\\
& g_8(x) \equiv -3x_1 + 6x_2 + 12(x_9 - 8)^2 - 7x_{10} \le 0,\\
& -10 \le x_i \le 10, \quad i = 1, \ldots, 10.
\end{array} \tag{18}$$

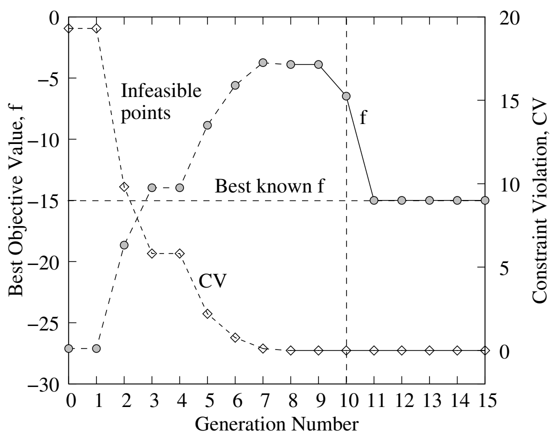

Fig. 5. Function value reduces with generation for problem 1. Figure taken from [9].

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:

$$\begin{array}{ll}
\text{min.} & f(\mathbf{x}) = -0.5\,(x_1x_4 - x_2x_3 + x_3x_9 - x_5x_9 \\
& +x_5x_8 - x_6x_7), \\
\text{s.t.} & g_1(\mathbf{x}) \equiv x_3^2 + x_4^2 - 1 \le 0, \\
& g_2(\mathbf{x}) \equiv x_9^2 - 1 \le 0, \\
& g_3(\mathbf{x}) \equiv x_5^2 + x_6^2 - 1 \le 0, \\
& g_4(\mathbf{x}) \equiv x_1^2 + (x_2 - x_9)^2 - 1 \le 0, \\
& g_5(\mathbf{x}) \equiv (x_1 - x_5)^2 + (x_2 - x_6) \le 0, \\
& g_6(\mathbf{x}) \equiv (x_1 - x_7)^2 + (x_2 - x_8) \le 0, \\
& g_7(\mathbf{x}) \equiv (x_3 - x_5)2 + (x_4 - x_6)^2 - 1‘0, \\
& g_8(\mathbf{x}) \equiv (x_3 - x_7)^2 + (x_4 - x_8)^2 - \le 0, \\
& g_9(\mathbf{x}) \equiv x_2 + (x_8 - x_9)^2 - 1 \le 0, \\
& g_{10}(\mathbf{x}) \equiv x_2x_3 - x_1x_4 \le 0, \\
& g_{11}(\mathbf{x}) \equiv -x_3x_9 \le 0, \\
& g_{12}(\mathbf{x}) \equiv x_5x_9 \le 0, \\
& g_{13}(\mathbf{x}) \equiv x_6x_7 - x_5x_8 \le 0, \\
& -10 \le x_i \le 10 \text{ for } i = 1, \ldots, 8, \quad 0 \le x_9 \le 20.
\end{array} \tag{19}$$

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

D. Problem 4

The problem is given as follows:

$$\begin{array}{ll}
\text{min.} & f(\mathbf{x}) = -x_1 - x_2, \\
\text{s.t.} & g_1(\mathbf{x}) \equiv -2x_1^4 + 8x_1^3 - 8x_1^2 + x_2 - 2 \le 0, \\
& g_2(\mathbf{x}) \equiv -4x_1^4 + 32x_1^3 - 88x_1^2 + 96x_1 + x_2 - 36 \le 0, \\
& 0 \le x_1 \le 3, \quad 0 \le x_2 \le 4.
\end{array} \tag{20}$$

The best-reported optimum lies at $x^* = (2.32952019747762, 3.17849307411774)$ and the corresponding objective value is $f(x^*) = -5.508013$.

E. Problem welded beam design

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

$$\begin{array}{ll}
\text{min.} & f_1(\mathbf{x}) = 1.10471h^2l + 0.04811tb(14.0 + l), \\
\text{s. t.} & g_1(\mathbf{x}) \equiv 13,600 - \tau(\mathbf{x}) \ge 0, \\
& g_2(\mathbf{x}) \equiv 30,000 - \sigma(\mathbf{x}) \ge 0, \\
& g_3(\mathbf{x}) \equiv b - h \ge 0, \\
& g_4(\mathbf{x}) \equiv P_c(\mathbf{x}) - 6,000 \ge 0, \\
& g_5(\mathbf{x}) \equiv 0.25 - \delta(\mathbf{x}) \ge 0, \\
& 0.125 \le h, b \le 5, \\
& 0.1 \le l, t \le 10,
\end{array} \tag{21}$$

where,

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

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

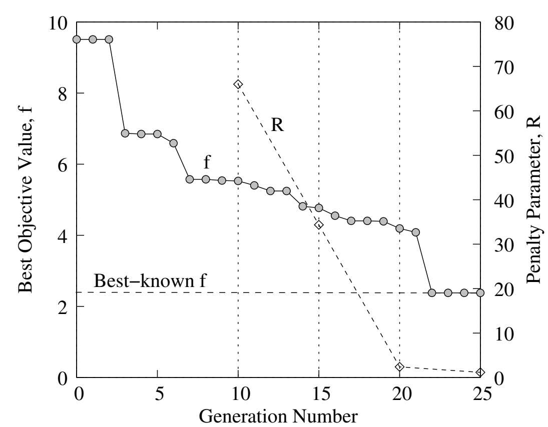

Fig. 6. Function value reduces with generation for welded beam design problem. Figure taken from [9].

With reference to problems 1-4, table I presents the comparison of function evaluations needed by the probabilistic joint evolutionary-classical approach and an existing approach [27]. The number of required evolutionary and classical function evaluations are shown separately.

C. Robotics problem

We emphasize that many real-world engineering design optimization problems involve multiple constraints to satisfy, while few is known about expected magnitude of violation to be reached for the constraints simultaneously during the search

TABLE I

Comparison of function evaluations needed by the probabilistic hybrid approach and an existing approach [27]; the number of required evolutionary and classical evaluations are shown separately.

Problem	Best known optima (f^*)	Function Evaluations (FEs)					
		Zavala, Aguirre & Diharce [27]			Proposed Hybrid Approach		
Mathematical		Best	Median	Worst	Best	Median	Worst
Problem 1 NSGA-II+Local	**−15.0**	80,776	90,343	96,669	**1,475** 1,000+475	**4,274** 3,500+774	**22,367** 14,000+8,367
Problem 2 NSGA-II+Local	**24.306209**	1,14,709	1,38,767	2,08,751	**4,069** 2,400+1,669	**23,842** 14,400+9,442	**55,721** 34,400+21,321
Problem 3 NSGA-II+Local	**−0.866025**	97,157	1,07,690	1,24,217	**2,107** 1,000+1,107	**14,082** 9,000+5,082	**36,547** 19,000+17,547
Problem 4 NSGA-II+Local	**−5.508013**	11,081	18,278	6,33,378	**1,263** 1,000+263	**7,695** 6,500+1,195	**41,762** 39,000+2,762
Engineering		Approach [8]			Proposed Hybrid Approach		
Welded beam design NSGA-II+Local	**2.38119**	3,20,000	3,20,000	3,20,000	**2,778** 1,500+1,278	**21,287** 18,000+3,287	**44,617** 37,000+7,617

process. Therefore proposed probabilistic based methodology is also applied to a constrained robot gripper design optimization problem. The problem concerns the design of a robot gripper optimally, which is commonly used in industry as an interaction device between environment, to pick and place object and to perform grasping and manipulation tasks. The vector of seven design variables are $\mathbf{x} = (a, b, c, e, f, l, \delta)^T$, where a, b, c, e, f, l are dimensions (link lengths) of the robot gripper and δ is the angle between link length b with c shown in figure 7 [26].

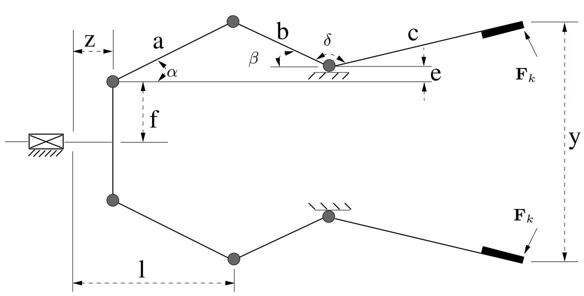

Fig. 7. A sketch of robot Gripper-I. Figure taken from [26]

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

$$f(\mathbf{x}) = \max_z F_k(\mathbf{x}, z) - \min_z F_k(\mathbf{x}, z). \tag{22}$$

The minimization of robot gripper design is subject to the following geometric and force constraints:

$$\begin{aligned}
g_1(\mathbf{x}) &\equiv Y_{\min} - y(\mathbf{x}, Z_{\max}) \geq 0,\\
g_2(\mathbf{x}) &\equiv y(\mathbf{x}, Z_{\max}) \geq 0,\\
g_3(\mathbf{x}) &\equiv y(\mathbf{x}, 0) - Y_{\max} \geq 0,\\
g_4(\mathbf{x}) &\equiv Y_G - y(\mathbf{x}, 0) \geq 0,\\
g_5(\mathbf{x}) &\equiv (a+b)^2 - l^2 - e^2 \geq 0.\\
g_6(\mathbf{x}) &\equiv (l - Z_{\max})^2 + (a-e)^2 - b^2 \geq 0,\\
g_7(\mathbf{x}) &\equiv l - Z_{\max} \geq 0.
\end{aligned} \tag{23}$$

where,

$$\begin{aligned}
g &\equiv \sqrt{(l-z)^2 + e^2},\\
b^2 &\equiv a^2 + g^2 - 2.a.g.\cos(\alpha - \phi),\\
\alpha &\equiv \arccos(\frac{a^2 + g^2 - b^2}{2.a.g}) + \phi,\\
a^2 &\equiv b^2 + g^2 - 2.b.g.\cos(\beta + \phi),\\
\beta &\equiv \arccos(\frac{b^2 + g^2 - a^2}{2.b.g}) - \phi,\\
\phi &\equiv \arctan(\frac{e}{l-z}),\\
F_k &\equiv \frac{P.b\sin(\alpha+\beta)}{2.c\cos\alpha},\\
y(x, z) &\equiv 2(e + f + c sin(\beta + \delta)),\\
\\
10 &\leq a \leq 250,\\
10 &\leq b \leq 250,\\
100 &\leq c \leq 300,\\
0 &\leq e \leq 50,\\
10 &\leq f \leq 250,\\
100 &\leq l \leq 300,\\
1.0 &\leq \delta \leq 3.14,\\
\\
Y_{\min} &= 50\, mm,\\
Y_{\max} &= 100\, mm,\\
Y_G &= 150\, mm,\\
Z_{\max} &= 50\, mm,\\
P &= 100\, N,\\
FG &= 50\, N.
\end{aligned} \tag{24}$$

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.

	Best	Median	worst
Teaching-learning-based optimization (TLBO) [26]	4.247644	4.93770095	8.141973
Artificial Bee Colony optimization [26]	4.247644	5.086611	6.784631
Proposed methodology	0.592738	0.623837	0.673636

abilistic 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

[1] J. Kuester, J. Mize, and D. Griffin, *Optimization Techniques with Fortran*. McGraw-Hill (New York), 1973.

[2] R. Fletcher, *Practical Methods of Optimization, volume 1*. Wiley, 1987.

[3] J. Nocedal and S. Wright, *Numerical optimization*. Springer verlag, 1999.

[4] K. Deb, *Optimization for Engineering Design: Algorithms and Examples*. New Delhi: Prentice-Hall, 1995.

[5] L. Davis, K. Jong, L. Whitley, and M. Vose, *Evolutionary Algorithms*. Springer-Verlag New York, Inc., 1999.

[6] K. Deb, *Multi-Objective Optimization using Evolutionary Algorithms*. Chichester, UK: Wiley, 2001.

[7] Z. Michalewicz and M. Schoenauer, "Evolutionary Algorithms for Constrained Parameter Optimization Problems," *Evolutionary computation*, vol. 4, no. 1, pp. 1–32, 1996.

[8] 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.

[9] 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.

[10] G. Reklaitis, A. Ravindran, and K. Ragsdell, *Engineering optimization: Methods and applications*. Wiley, 1983.

[11] S. Rao, *Engineering Optimization: Theory and Practice*. New Age Publishers, 1996.

[12] D. E. Goldberg, *Genetic Algorithms for Search, Optimization, and Machine Learning*. Reading, MA: Addison-Wesley, 1989.

[13] M. Gen and R. Cheng, *Genetic Algorithms and Engineering Optimization*. Wiley-interscience, 2000.

[14] 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.

[15] J. D. Knowles, R. A. Watson, and D. W. Corne, "Reducing local optima in single-objective problems by multi-objectivization," in *Proceedings of the First International Conference on Evolutionary Multi-Criterion Optimization (EMO-01)*, 2001, pp. 269–283.

[16] 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 (ISICA 2007)*. Springer-Verlag, 2007, pp. 36–45.

[17] Y. Wang, Z. Cai, Y. Zhou, and W. Zeng, "An adaptive tradeoff model for constrained evolutionary optimization," *IEEE Transactions on Evolutionary Computation*, vol. 12, no. 1, pp. 80–92, 2008.

[18] H. Myung and J. Kim, "Hybrid interior-lagrangian penalty based evolutionary optimization," in *Evolutionary Programming VII*. Springer, 1998, pp. 85–94.

[19] P. Chootinan and A. Chen, "Constraint handling in genetic algorithms using a gradient-based repair method," *Computers & Operations Research*, vol. 33, no. 8, pp. 2263–2281, 2006.

[20] E. Zahara and Y. Kao, "Hybrid nelder–mead simplex search and particle swarm optimization for constrained engineering design problems," *Expert Systems with Applications*, vol. 36, no. 2, pp. 3880–3886, 2009.

[21] A. Mani and C. Patvardhan, "A hybrid quantum evolutionary algorithm for solving engineering optimization problems," *International Journal of Hybrid Intelligent Systems*, vol. 7, no. 3, pp. 225–235, 2010.

[22] R. Datta, "Constrained engineering design optimization using a hybrid bi-objective evolutionary-classical methodology," *Simulated Evolution and Learning*, pp. 633–637, 2010.

[23] C. Coello, "Use of a self-adaptive penalty approach for engineering optimization problems," *Computers in Industry*, vol. 41, no. 2, pp. 113–127, 2000.

[24] T. Peeters and O. Ciftcioglu, "Statistics on exponential averaging of periodograms," *Signal Processing, IEEE Transactions on*, vol. 43, no. 7, pp. 1631–1636, 1995.

[25] J. J. Liang, T. P. Runarsson, E. Mezura-Montes, M. Clerc, P. N. Suganthan, C. A. C. Coello, and K. Deb, "Problem definitions and evaluation criteria for the CEC 2006: Special session on constrained real-parameter optimization," Nanyang Technological University, Singapore, Tech. Rep., 2006.

[26] R. V. Rao, V. J. Savsani, and D. P. Vakharia, "Teaching-learning-based optimization: A novel method for constrained mechanical design optimization problems," *Computer Aided Design*, vol. 43, pp. 303–315, March 2011.

[27] A. Zavala, A. Aguirre, and E. Diharce, "Continuous constrained optimization with dynamic tolerance using the copso algorithm," in *Constraint-Handling in Evolutionary Computation*, E. Mezura-Montes, Ed. Berlin: Springer, 2009, ch. 1, pp. 1–23.