Search Biases in Constrained Evolutionary Optimization

Thomas Philip Runarsson, *Member, IEEE,* and Xin Yao, *Fellow, IEEE,*

***Abstract*— A common approach to constraint handling in evolutionary optimization is to apply a penalty function to bias the search towards a feasible solution. It has been proposed that the subjective setting of various penalty parameters can be avoided using a multi-objective formulation. This paper analyses and explains in depth why and when the multi-objective approach to constraint handling is expected to work or fail. Furthermore, an improved evolutionary algorithm based on evolution strategies and differential variation is proposed. Extensive experimental studies have been carried out. Our results reveal that the unbiased multi-objective approach to constraint handling may not be as effective as one may have assumed.**

***Index Terms*— Nonlinear programming, multi-objective, penalty functions, evolution strategy.**

I. INTRODUCTION

THIS paper considers the general nonlinear programming problem formulated as

$$\text{minimize} \quad f(\boldsymbol{x}), \quad \boldsymbol{x} = (x_1, \ldots, x_n) \in \mathcal{R}^n, \tag{1}$$

where $f(\boldsymbol{x})$ is the objective function, $\boldsymbol{x} \in \mathcal{S} \cap \mathcal{F}$, $\mathcal{S} \subseteq \mathcal{R}^n$ defines the search space bounded by the parametric constraints

$$\underline{x}_i \leq x_i \leq \overline{x}_i, \tag{2}$$

and the feasible region $\mathcal{F}$ is defined by

$$\mathcal{F} = \{\boldsymbol{x} \in \mathcal{R}^n \mid g_j(\boldsymbol{x}) \leq 0 \quad \forall\, j\}, \tag{3}$$

where $g_j(\boldsymbol{x})$, $j = 1, \ldots, m$, are inequality constraints (equality constraints may be approximated by inequality constraints).

There have been many methods proposed for handling constraints in evolutionary optimization, including the penalty function method, special representations and operators, co-evolutionary method, repair method, multi-objective method, etc [1]. The penalty function method, due to its simplicity, is by far the most widely studied and used in handling constraints.

The introduction of a penalty term enables the transformation of a constrained optimization problem into a series of unconstrained ones. The common formulation is the following exterior penalty method,

$$\begin{aligned} \text{minimize} \quad \psi(\boldsymbol{x}) &= f(\boldsymbol{x}) + w_0 \sum_{j=1}^{m} w_j \big(g_j^+(\boldsymbol{x})\big)^\beta \\ &= f(\boldsymbol{x}) + w_0 \phi\big(\boldsymbol{g}^+(\boldsymbol{x})\big), \end{aligned} \tag{4}$$

where $\phi\big(\boldsymbol{g}^+(\boldsymbol{x})\big)$ is the penalty function and $\boldsymbol{g}^+(\boldsymbol{x}) = \{g_1^+(\boldsymbol{x}), \ldots, g_m^+(\boldsymbol{x})\}$ are the constraint violations,

$$g_j^+(\boldsymbol{x}) = \max[0, g_j(\boldsymbol{x})]. \tag{5}$$

Manuscript received September 1, 2003; revised February 1, 2004.

The exponent β is usually 1 or 2 and the weights $w_j; j = 0, \ldots, m$, are not necessarily held constant during search. In practice, it is difficult to find the optimal weights $w_j; j = 0, \ldots, m$ for a given problem. Balancing the objective function $f(\boldsymbol{x})$ and constraint violations $g_j^+(\boldsymbol{x})$ has always been a key issue in the study of constraint handling.

One way to avoid the setting of penalty parameters $w_j; j = 0, \ldots, m$ subjectively in (5) is to treat the constrained optimization problem as a multi-objective one [2, p. 403], where each of the objective function and constraint violations is a separate objective to be minimized,

$$\begin{aligned} &\text{minimize} \quad f(\boldsymbol{x}), \\ &\text{minimize} \quad g_j^+(\boldsymbol{x}), \quad j = 1, \ldots, m. \end{aligned} \tag{6}$$

Alternatively, one could approach the feasible region by considering only the constraint violations as objectives [3], [4],

$$\text{minimize} \quad g_j^+(\boldsymbol{x}), \quad j = 1, \ldots, m, \tag{7}$$

in which case the Pareto optimal set is the feasible region. These unbiased multi-objective approaches are compared with the penalty function method.

Although the idea of handling constraints through multi-objective optimization is very attractive, a search bias towards the feasible region must still be introduced in optimization if a feasible solution is to be found. When comparing the unbiased multi-objective approach to that of the biased penalty function method it becomes evident that the multi-objective approach does not work as well as one might first think. It does not solve the fundamental problem of balancing the objective function and constraint violations faced by the penalty function approach. The introduction of a search bias to the multi-objective approach would clearly be beneficial as illustrated in [5]. However, these search biases are also subjective and therefore defeat the purpose of the current study. The purpose of this study is not to present a new multi-objective approach for constraint handling. The main contribution lies in a new and clear exposition of how the multi-objective approach to constraint handling works and how to improve it in a principled way based on this new understanding. Furthermore, a search bias depends not only on selection but also on the chosen search operators. A significant improvement in performance can be achieved when the appropriate search distribution is applied. It will be shown that there exists a more suitable search distribution for some commonly studied benchmark functions.

The remainder of the paper is organized as follows. Section II introduces the test function used in this paper. Both artificial test functions with known characteristics and benchmark test functions widely used in the literature are included. Section III proposes an improved evolutionary algorithm used in this paper for constrained optimization. It combines evolution strategies with differential variation. Section IV presents our experimental results and discussions. Several different constraint handling techniques using the multi-objective approach are studied. Finally, Section V concludes the paper with a brief summary and some remarks.

II. Test Problems

Two types of test problems will be used in this paper. The first are artificial test functions with known characteristics. Such test functions enable one to understand and analyze experimental results. They also help in validating theories and gaining insights into different constraint handling techniques. The second type of test problems investigated are 13 widely-used benchmark functions [6], [7], [8].

Let's first introduce the artificial test functions as follows,

$$\text{minimize } f(\boldsymbol{x}) = \sum_{i=1}^{n} (x_i - c_{i,0})^2 \tag{8}$$

subject to

$$g_j(\boldsymbol{x}) = \sum_{i=1}^{n} (x_i - c_{i,j})^2 - r_j^2 \leq 0, \tag{9}$$

where $r_j > 0$, $j = 1, \ldots, m$, and n is the problem dimension. This problem is similar to that used in the test-case generator in [9]. A solution is infeasible when $g_j(\boldsymbol{x}) > 0, \forall j \in [1, \ldots, m]$, otherwise it is feasible. In other words,

$$g_j^+(\boldsymbol{x}) = \begin{cases} \max[0, g_j(\boldsymbol{x})] & \text{if } g_k(\boldsymbol{x}) > 0, \forall\ k \in [1, \ldots, m] \\ 0 & \text{otherwise.} \end{cases}$$

The local optima, $\forall\ j \in [1, \ldots, m]$, are

$$\boldsymbol{x}_j^* = \begin{cases} \boldsymbol{c}_j + r_j \frac{\boldsymbol{c}_0 - \boldsymbol{c}_j}{\|\boldsymbol{c}_0 - \boldsymbol{c}_j\|} & \text{when } \|\boldsymbol{c}_0 - \boldsymbol{c}_j\| > r_j \\ \boldsymbol{c}_0 & \text{otherwise.} \end{cases} \tag{10}$$

If any local optimum is located at $\boldsymbol{c}_0$, then this is the constrained as well as unconstrained global optimum. In the case where $\boldsymbol{c}_0$ is infeasible, the local minima with the smallest $\|\boldsymbol{c}_0 - \boldsymbol{c}_j\| - r_j$ is the constrained global minimum. For an infeasible solution $\boldsymbol{x}$, the sum of all constraint violations,

$$\phi(\boldsymbol{x}) = \sum_{j=1}^{m} w_j \left(\sum_{i=1}^{n} (x_i - c_{i,j})^2 - r_j^2 \right)^{\beta}, \tag{11}$$

form a penalty function which has a single optimum located at $\boldsymbol{x}_\phi^*$.

An example of the artificial test functions defined by equations (8) and (9) with $m = n = 2$ is shown in figure 1. Here the objective function's unconstrained global minimum is located at $\boldsymbol{c}_0$. Dotted contour lines for this function are drawn as circles around this point. The example has two constraints illustrated by the two solid circles. Any point within these two circles is feasible and the local minimum are $\boldsymbol{x}_1^*$ and $\boldsymbol{x}_2^*$. The larger circle contains the constrained global minimum, which is $\boldsymbol{x}_1^*$. The penalty function that uses $\beta = w_0 = w_1 = w_2 = 1$ has its minimum located at $\boldsymbol{x}_\phi^*$ in the infeasible region and the dashed contours for the penalty function are centered around this point. Figure 1 also shows two Pareto sets. The shaded sector represents the Pareto optimal set for (6). The global optimal feasible solution is located at $\boldsymbol{x}_1^*$ and belongs to this Pareto optimal set. Using this formulation the search may wander in and out of the feasible region. This could be avoided if all feasible solution were set to a special 0 Pareto level. Alternatively, an optimization level technique applied to find regions of preferred solution with small constraint violations would surely be located near $\boldsymbol{x}_\phi^*$. The Pareto optimal set for (7) is the feasible region but the next best (level 2) Pareto set is depicted figure 1. This is the line drawn from the centers of the feasible spheres and between the two feasible regions. Again, an optimization level technique biased towards a region for which all constraint violations are small would concentrate its search around $\boldsymbol{x}_\phi^*$. Notice also that a search guided by (7) enters the feasible region at a different point to that when guided by (6).

The artificial test functions defined by equations (8) and (9) are simple yet capture many important characteristics of constrained optimization problems. It is scalable, easy to implement, and easy to visualize in low dimension cases. Because we know the characteristics, we can understand and analyze the experimental results much better than on an unknown test function. However, the artificial test functions are not widely used. They do not include all the characteristics of different constrained optimization problems. To evaluate our evolutionary algorithm and constraint handling techniques comprehensively in the remaining sections of this paper, we employ a set of 13 benchmark functions from the literature [6], [7] in our study, in addition to the artificial function. The 13 benchmark functions are described in the appendix and their main characteristics are summarized in table I.

As seen from Table I, the 13 benchmark functions represent a reasonable set of diverse functions that will help to evaluate

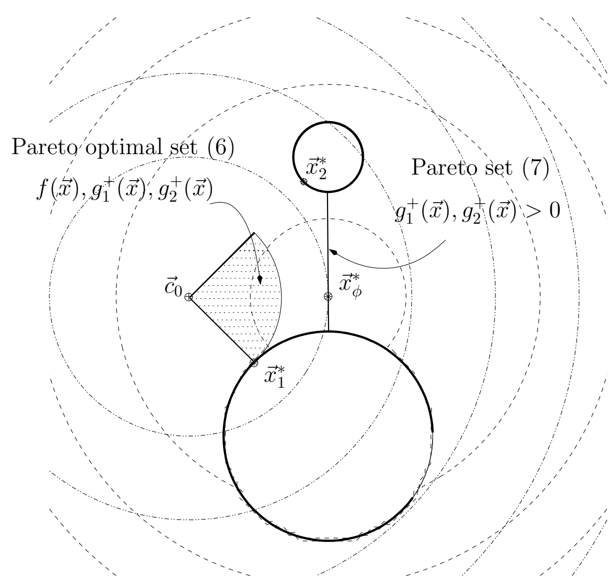

Fig. 1. A 2-D example of the artificial test function.

TABLE I

SUMMARY OF MAIN PROPERTIES OF THE BENCHMARK PROBLEMS (NE: NONLINEAR EQUALITY, NI: NONLINEAR INEQUALITY, LI: LINEAR INEQUALITY, THE NUMBER OF ACTIVE CONSTRAINTS AT OPTIMUM IS a.)

fcn	n	$f(\boldsymbol{x})$ type	$\lvert\mathcal{F}\rvert/\lvert\mathcal{S}\rvert$	LI	NE	NI	a
g01	13	quadratic	0.011%	9	0	0	6
g02	20	nonlinear	99.990%	1	0	1	1
g03	10	polynomial	0.002%	0	1	0	1
g04	5	quadratic	52.123%	0	0	6	2
g05	4	cubic	0.000%	2	3	0	3
g06	2	cubic	0.006%	0	0	2	2
g07	10	quadratic	0.000%	3	0	5	6
g08	2	nonlinear	0.856%	0	0	2	0
g09	7	polynomial	0.512%	0	0	4	2
g10	8	linear	0.001%	3	0	3	3
g11	2	quadratic	0.000%	0	1	0	1
g12	3	quadratic	4.779%	0	0	9^3	0
g13	5	exponential	0.000%	0	3	0	3

different constraint handling techniques and gain a better understand why and when some techniques work or fail.

III. EXPERIMENTAL SETUP

An evolutionary algorithm (EA) is based on the collective learning process within a population of individuals each of which represents a point in the search space. The EA's driving force is the rate at which the individuals are imperfectly replicated. The rate of replication is based on a quality measure for the individual. Here this quality is a function of the objective function and the constraint violations. In particular the population of individuals, of size λ, are ranked from best to worst, denoted $(\boldsymbol{x}_{1;\lambda}, \ldots, \boldsymbol{x}_{\mu;\lambda}, \ldots, \boldsymbol{x}_{\lambda;\lambda})$, and only the best μ are allowed to replicate λ/μ times. The different rankings considered are as follows:

1) Rank feasible individuals highest and according to their objective function value, followed by the infeasible solutions ranked according to penalty function value. This is the so called *over-penalized* approach and denoted as method A.
2) Treat the problem as an unbiased multi-objective optimization problem, either (6) or (7). Use a non-dominated ranking with the different Pareto levels determined using for example the algorithm described in [10]. All feasible solutions are set to a special 0 Pareto level. Applying the over-penalty approach, feasible individuals are ranked highest and according to their objective function value, followed by the infeasible solutions ranked according to their Pareto level. The ranking strategies based on (6) and (7) are denoted as methods B and C respectively.
3) Rank individuals such that neither the objective function value nor the penalty functions or Pareto level determine solely the ranking. An example of such a ranking would be the stochastic ranking [7] illustrated in figure 2. The ranking strategies above will in this case be marked by a dash, i.e. A', B', and C'.

The different ranking determine which parent individuals are to be replicated λ/μ times imperfectly. These imperfections or mutations have a probability density function (PDF) that can either dependent on the population and/or be self-adaptive. The evolution strategy (ES) is an example of a self-adaptive EA, where the individual represents a point in the search space as well as some strategy parameters describing the PDF. In mutative step-size self-adaptation the mutation strength is randomly changed. It is only dependent on the parent's mutation strength, that is the parent step-size multiplied by a random number. This random number is commonly log-normally distributed but other distributions are equally plausible [11], [12].

The *isotropic* mutative self-adaptation for a (μ, λ) ES, using the log-normal update rule, is as follows [13],

$$\begin{aligned} \sigma'_k &= \sigma_{i;\lambda} \exp(\tau_o N(0,1)), \qquad (12)\\ \boldsymbol{x}_k &= \boldsymbol{x}_{i;\lambda} + \sigma'_k \boldsymbol{N}(0,1), \quad k = 1, \ldots, \lambda \end{aligned}$$

for parent $i \in [1, \mu]$ where $\tau_o \simeq c_{(\mu,\lambda)}/\sqrt{n}$ [14]. Similarly, the *non-isotropic* mutative self-adaptation rule is,

$$\begin{aligned} \sigma'_{k,j} &= \sigma_{(i;\lambda),j} \exp\big(\tau' N(0,1) + \tau N_j(0,1)\big), \qquad (13)\\ x'_{k,j} &= x_{(i;\lambda),j} + \sigma'_{k,j} N_j(0,1), \; k = 1, \ldots, \lambda, j = 1, \ldots, n \end{aligned}$$

where $\tau' = \varphi/\sqrt{2n}$ and $\tau = \varphi/\sqrt{2\sqrt{n}}$ [13].

The primary aim of the step-size control is to tune the search distribution so that maximal progress in maintained. For this some basic conditions for achieving optimal progress must be satisfied. The first lesson in self-adaptation is taken from the $1/5$-*success rule* [15, p. 367]. The rule's derivation is based on the probability w_e that the offspring is better than the parent. This probability is calculated for the case where the optimal standard deviation is used $\hat{w}_e$, from which it is then determined that the number of trials must be greater than or equal to $1/\hat{w}_e$ if the parent using the optimal step-size is to be successful. Founded on the sphere and corridor models, this is the origin of the $1/5$ value.

In a mutative step-size control, such as the one given by (12), there is no single *optimal* standard deviation being tested, but rather a series of trial step sizes σ'_k, $k = 1, \ldots, \lambda/\mu$ centered (the expected median is $\sigma_{i;\lambda}$) around the parent step size $\sigma_{i;\lambda}$. Consequently, the number of trials may need to be greater than that specified by the $1/5$-success rule. If enough trial steps for success are generated near the optimal standard deviation then this trial step-size will be inherited via the corresponding offspring. This offspring will necessarily also be the most

```
I_j = j ∀ j ∈ {1, ..., λ}
for i = 1 to λ do
    for j = 1 to λ − 1 do
        sample u ∈ U(0,1) (uniform random number generator)
        if (φ(I_j) = φ(I_{j+1}) = 0) or (u < 0.45) then
            if (f(I_j) > f(I_{j+1})) then
                swap(I_j, I_{j+1})
            fi
        else
            if (φ(I_j) > φ(I_{j+1})) then
                swap(I_j, I_{j+1})
            fi
        fi
    od
    if no swap done break fi
od
```

Fig. 2. The stochastic ranking algorithm [7]. In the case of non-dominated ranking ϕ is replaced by the Pareto level.

likely to achieve the greatest progress and hence be the fittest. The fluctuations on $\sigma_{i;\lambda}$ (the trial standard deviations σ'_k) and consequently also on the optimal mutation strength, will degrade the performance of the ES. The theoretical maximal progress rate is impossible to obtain. Any reduction of this fluctuation will therefore improve performance [14, p. 315]. If random fluctuations are not reduced, then a larger number of trials must be used (the number of offspring generated per parent) in order to guarantee successful mutative self-adaptation. This may especially be the case for when the number of free strategy parameters increases, as in the non-isotropic case.

Reducing random fluctuations may be achieved using averaging or recombination on the strategy parameters. The most sophisticated approach is the *derandomized approach to self-adaptation* [16] which also requires averaging over the population. However, when employing a Pareto based method one is usually exploring different regions of the search space. For this reason one would like to employ a method which reduces random fluctuations without averaging over very different individuals in the population. Such a technique is described in [17] and implemented in our study. The method takes an *exponential recency-weighted average* of trial step sizes sampled via the lineage instead of the population.

In a previous study [7] the authors had some success applying a simple ES, using the *non-isotropic* mutative self-adaptation rule, on the 13 benchmark functions described in the previous section. The algorithm described here is equivalent but uses the exponential averaging of trial step sizes. Its full details are presented by the pseudocode in figure 3. As seen in the figure the exponential smoothing is performed on line 10. Other notable features are that the variation of the objective parameters $\boldsymbol{x}$ is retried if they fall outside of the parametric bounds. A mutation out of bounds is retried only 10 times after which it is set to its parent value. Initially (line 1) the parameters are set uniform and randomly within these bounds. The initial step sizes are also set with respect to the parametric bounds (line 1) guaranteeing initial reachability over the entire search space.

There is still one problem with the search algorithm described in figure 3. The search is biased toward a grid aligned with the coordinate system [18]. This could be solved by adapting the full covariance matrix of the search distribution for the function topology. This would require $(n^2+n)/2$ strategy parameters and is simply too costly for complex functions. However, to illustrate the importance of this problem a simple modification to our algorithm is proposed. The approach can be thought of as a variation of the Nelder-Mead method [19] or differential evolution [20]. The search is "helped" by performing one mutation per parents i as follows,

$$\boldsymbol{x}'_k = \boldsymbol{x}_{i;\lambda} + \gamma(\boldsymbol{x}_{1;\lambda} - \boldsymbol{x}_{i+1;\lambda}), \quad i \in \{1, \dots, \mu - 1\} \tag{14}$$

where the search direction is determined by the best individual in the population and the individual ranked one below the parent i being replicated. The step length taken is controlled by the new parameter γ. The setting of this new parameter will be described in the next section. The modified algorithm is described in figure 4 where the only change has been the addition of lines 8–10. For these trials the parent mean step size is copied unmodified (line 9). Any trial parameter outside the parametric bounds is generated anew by a standard mutation as before. Since this new variation involves other members of the population, it will only be used in the case where the ranking is based on the penalty function method.

1 *Initialize*: $\boldsymbol{\sigma}'_k := (\overline{\boldsymbol{x}}_k - \underline{\boldsymbol{x}}_k)/\sqrt{n}$, $\boldsymbol{x}'_k = \underline{\boldsymbol{x}}_k + (\overline{\boldsymbol{x}}_k - \underline{\boldsymbol{x}}_k)\boldsymbol{U}_k(0,1)$
2 **while** *termination criteria not satisfied* **do**
3 *evaluate*: $f(\boldsymbol{x}'_k)$, $\boldsymbol{g}^+(\boldsymbol{x}'_k), k = 1 \dots, \lambda$
4 *rank the λ points and copy the best μ in their ranked order*:
5 $(\boldsymbol{x}_i, \boldsymbol{\sigma}_i) \leftarrow (\boldsymbol{x}'_{i;\lambda}, \boldsymbol{\sigma}'_{i;\lambda}), i = 1, \dots, \mu$
6 **for** $k := 1$ to λ **do** (*replication*)
7 $i \leftarrow \mod(k-1, \mu) + 1$ (*cycle through the best μ points*)
8 $\sigma'_{k,j} \leftarrow \sigma_{i,j} \exp\big(\tau' N(0,1) + \tau N_j(0,1)\big), j = 1, \dots, n$
9 $\boldsymbol{x}'_k \leftarrow \boldsymbol{x}_i + \boldsymbol{\sigma}'_k \boldsymbol{N}(0,1)$ (*if out of bounds then retry*)
10 $\boldsymbol{\sigma}'_k \leftarrow \boldsymbol{\sigma}_i + \alpha(\boldsymbol{\sigma}'_k - \boldsymbol{\sigma}_i)$ (*exponential smoothing* [17])
11 **od**
12 **od**

Fig. 3. Outline of the simple (μ, λ) ES using exponential smoothing to facilitate self-adaptation (typically $\alpha \approx 0.2$).

IV. Experimental Study

The experimental studies are conducted in three parts. First of all, the behavior of six different ranking methods (A, B, C, A', B', C') for constraint handling are compared. In section IV-A, the search behavior for these methods is illustrated on the artificial test function using a simple ES. In section IV-B the search performance of these methods on commonly used benchmark functions is compared. Finally, the improved search algorithm presented in figure 4 is examined on the benchmark functions in section IV-C.

A. Artificial function

The search behavior resulting from the different ranking methods is illustrated using an artificial test function similar to the one depicted in fig. 1. Here both the objective function and constraint violations are spherically symmetric and so the isotropic mutation (12) is more suitable than (13) described on

1 *Initialize*: $\boldsymbol{\sigma}'_k := (\overline{\boldsymbol{x}}_k - \underline{\boldsymbol{x}}_k)/\sqrt{n}$, $\boldsymbol{x}'_k = \underline{\boldsymbol{x}}_k + (\overline{\boldsymbol{x}}_k - \underline{\boldsymbol{x}}_k)\boldsymbol{U}_k(0,1)$
2 **while** *termination criteria not satisfied* **do**
3 *evaluate*: $f(\boldsymbol{x}'_k)$, $\boldsymbol{g}^+(\boldsymbol{x}'_k), k = 1 \dots, \lambda$
4 *rank the λ points and copy the best μ in their ranked order*:
5 $(\boldsymbol{x}_i, \boldsymbol{\sigma}_i) \leftarrow (\boldsymbol{x}'_{i;\lambda}, \boldsymbol{\sigma}'_{i;\lambda}), i = 1, \dots, \mu$
6 **for** $k := 1$ to λ **do**
7 $i \leftarrow \mod(k-1, \mu) + 1$
8 **if** $(k < \mu)$ **do** (*differential variation*)
9 $\boldsymbol{\sigma}'_k \leftarrow \boldsymbol{\sigma}_i$
10 $\boldsymbol{x}'_k \leftarrow \boldsymbol{x}_i + \gamma(\boldsymbol{x}_1 - \boldsymbol{x}_{i+1})$
11 **else** (*standard mutation*)
12 $\sigma'_{k,j} \leftarrow \sigma_{i,j} \exp\big(\tau' N(0,1) + \tau N_j(0,1)\big), j = 1, \dots, n$
13 $\boldsymbol{x}'_k \leftarrow \boldsymbol{x}_i + \boldsymbol{\sigma}'_k \boldsymbol{N}(0,1)$
14 $\boldsymbol{\sigma}'_k \leftarrow \boldsymbol{\sigma}_i + \alpha(\boldsymbol{\sigma}'_k - \boldsymbol{\sigma}_i)$
15 **od**
16 **od**
17 **od**

Fig. 4. Outline of the improved (μ, λ) ES using the differential variation (lines $8 - 11$) performed once for each of the best $\mu - 1$ point.

line 8 in figure 3. Exponential smoothing is also not necessary, i.e. $\alpha = 1$. 1000 independent runs using a $(1, 10)$ ES are made and the algorithm is terminated once the search has converged. The termination criterion used is when the mean step size $\sigma < 10^{-7}$ the algorithm is halted. The initial step size is $\sigma = 1$ and the initial point used is $\boldsymbol{x} = [-1,\ 0]$ (marked by $*$). The artificial function is defined by the centers $\boldsymbol{c}_0 = [-1,\ 0]$, $\boldsymbol{c}_1 = [-1,\ 1]$, $\boldsymbol{c}_2 = [1,\ -1]$ and the radius $\boldsymbol{r} = [0.1,\ 0.8]$. This experiment is repeated for all six different ranking methods and plotted in figure 5. The final 1000 solutions are plotted as dots on the graphs.

The first two experiments are based on the penalty function method with $w_0 = w_1 = w_2 = \beta = 1$. Case A is an over-penalty and case A' uses the stochastic ranking [7] to balance the influence of the objective and penalty function on the ranking. In case A one observes that the search is guided to $\boldsymbol{x}^*_\phi$, but since the initial step size is large enough some parents happen upon a feasible region and remain there. Clearly the larger the feasible space is the likelier is this occurrence. From this example it should also be possible to visualize the case where $\boldsymbol{x}^*_\phi$ is located further away or closer to the global feasible optimum. The location of $\boldsymbol{x}^*_\phi$ is determined by the constraint violations and also the different penalty function parameters. In case A' the search is biased not only towards $\boldsymbol{x}^*_\phi$ but also towards c_0. Again it should be possible to imagine that theses attractors could be located near or far from the global feasible optimum. For the example plotted both are in the infeasible region creating additional difficulties for this approach.

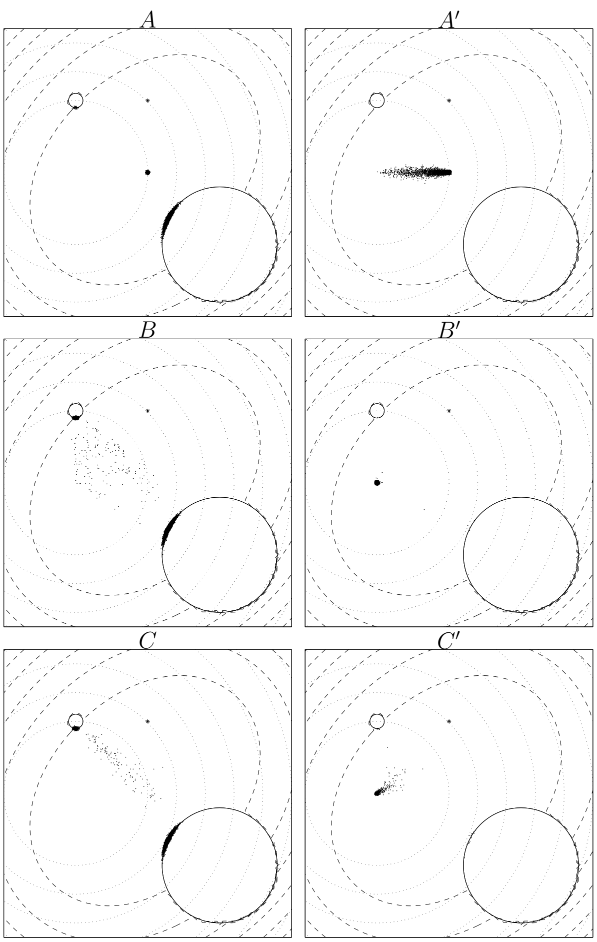

Fig. 5. The six different search behaviors resulting from the different ranking methods described in the text as cases A to C'. Some noise is added to make the density of the dots clearer on the figures.

The last four experiments are equivalent to the previous two, however, now the ranking of infeasible solution is determined by a non-dominated ranking. In case B the non-dominated ranking is based on (6) where all feasible solution are set at the special highest Pareto level 0. In this case there is no collection of solutions around $\boldsymbol{x}^*_\phi$ but instead the search is spread over the Pareto front increasing the likelihood of falling into a feasible region and remaining there. Nevertheless, a number of parents were left scattered in the infeasible region at the end of their runs. When the objective function takes part in determining the ranking also, as shown by case B', the search is centered at c_0. Again the experiment is repeated but now the non-dominated ranking is based on (7). These cases are marked C and C' respectively. The results are similar, however, in case C' the parents are spread between c_0 and the Pareto front defined by the line between the centers of the two feasible spheres.

In general a feasible solution is either found by chance or when the search is biased by small constraint violations, and/or small objective function value, to a feasible region (or close to one). The global feasible optimum does not necessarily need to be in this region. Furthermore, one cannot guarantee that the attractors are located near a feasible region, as illustrated by the test case studied here. For the multi-objective approach, individuals will drift on the Pareto front and may chance upon a feasible region in an unbiased manner. The likelihood of locating a feasible region is higher when the size of the feasible search space is large.

It is possible that a method that minimizes each constraint violation independently would locate the two disjoint feasible regions. Such a multi-objective method was proposed in [21]. However, such a method may have difficulty finding feasible solutions to other problems such as `g13` [22].

B. Benchmarks functions

The artificial test function in the previous section illustrates the difficulties of finding a general method for constraint handling for nonlinear programming problems. However, in practice the class of problems studied may have different properties. For this reason it is also necessary to investigate the performance of our algorithms on benchmark functions typically originating from real world applications. These are the 13 benchmark functions summarized in table I and listed in the appendix.

For these experiments the non-isotropic mutation (13) with a smoothing factor $\alpha = 0.2$ is used. The expected rate of convergence φ is scaled up so that the expected change in the step size between generations is equivalent to when $\varphi = 1$ and $\alpha = 1$, see [17] for details. As it may be necessary for a multi-objective approach to maintain a greater diversity a larger than usual parent number is used, i.e. a $(60, 400)$ ES. Since the parent number has been doubled the number of generations

TABLE II
RESULTS USING THE OVER-PENALIZED APPROACH.

fcn/rnk	best	median	mean	st. dev.	worst	G_m
g01/	−15.000					
A	−15.000	−15.000	−15.000	4.7E−14	−15.000	874
C	−15.000	−15.000	−15.000	7.7E−14	−15.000	875
(3) B	−1.506	−1.502	−1.202	5.2E−01	−0.597	693
g02/	−0.803619					
A	−0.803474	−0.769474	−0.762029	2.9E−02	−0.687122	865
C	−0.803523	−0.773975	−0.764546	2.8E−02	−0.687203	858
B	−0.803517	−0.756532	−0.754470	3.1E−02	−0.673728	837
g03/	−1.000					
A	−0.400	−0.126	−0.151	1.1E−01	−0.020	54
C	−0.652	−0.103	−0.157	1.6E−01	−0.024	46
(11) B	−0.703	−0.085	−0.188	2.5E−01	−0.003	34
g04/	−30665.539					
A	−30665.539	−30665.539	−30665.539	1.1E−11	−30665.539	478
C	−30665.539	−30665.539	−30665.539	1.0E−11	−30665.539	472
B	−30665.539	−30665.539	−30665.539	1.4E−11	−30665.539	467
g05/	5126.498					
(22) A	5129.893	5274.662	5336.733	2.0E+02	5771.563	247
C	5127.351	5380.142	5415.491	2.8E+02	6030.261	440
B	–	–	–	–	–	–
g06/	−6961.814					
A	−6961.814	−6961.814	−6961.814	1.9E−12	−6961.814	663
C	−6961.814	−6961.814	−6961.814	1.9E−12	−6961.814	658
B	−6961.814	−6961.814	−6921.010	7.8E+01	−6702.973	646
g07/	24.306					
A	24.323	24.474	24.552	2.3E−01	25.284	774
C	24.336	24.500	26.364	9.5E+00	76.755	873
B	24.317	24.450	24.488	1.6E−01	25.092	816
g08/	−0.095825					
A	−0.095825	−0.095825	−0.095825	2.8E−17	−0.095825	335
C	−0.095825	−0.095825	−0.095825	2.8E−17	−0.095825	343
B	−0.095825	−0.095825	−0.095825	2.7E−17	−0.095825	295
g09/	680.630					
A	680.635	680.673	680.694	7.6E−02	681.028	208
C	680.632	680.667	680.693	8.4E−02	681.086	264
B	680.632	680.668	680.680	5.3E−02	680.847	235
g10/	7049.248					
A	7085.794	7271.847	7348.555	2.4E+02	8209.622	776
C	7130.745	7419.552	7515.000	4.7E+02	9746.656	827
(16) B	10364.402	14252.581	15230.161	4.4E+03	23883.283	57
g11/	0.750					
A	0.750	0.857	0.840	6.0E−02	0.908	19
C	0.750	0.821	0.827	5.5E−02	0.906	15
B	0.750	0.750	0.751	3.7E−03	0.766	478
g12/	−1.000000					
A	−1.000000	−1.000000	−0.999992	4.6E−05	−0.999750	85
C	−1.000000	−1.000000	−0.999992	4.6E−05	−0.999747	85
B	−1.000000	−1.000000	−1.000000	1.0E−11	−1.000000	85
g13/	0.053950					
A	0.740217	0.999424	0.974585	5.9E−02	0.999673	875
C	0.564386	0.948202	0.884750	1.3E−01	0.999676	875
B	–	–	–	–	–	–

TABLE III
RESULTS USING THE STOCHASTIC RANKING.

fcn/rnk	best	median	mean	st. dev.	worst	G_m
g01/	−15.000					
A'	−15.000	−15.000	−15.000	1.2E−13	−15.000	875
C'	−15.000	−15.000	−15.000	4.1E−12	−15.000	875
B'	–	–	–	–	–	–
g02/	−0.803619					
A'	−0.803566	−0.766972	−0.760541	3.8E−02	−0.642388	848
C'	−0.803542	−0.769708	−0.768147	2.2E−02	−0.729792	863
B'	−0.803540	−0.771902	−0.765590	2.7E−02	−0.697707	794
g03/	−1.000					
A'	−1.000	−0.999	−0.999	5.3E−04	−0.998	381
C'	−1.000	−0.999	−0.999	5.0E−04	−0.998	369
B'	–	–	–	–	–	–
g04/	−30665.539					
A'	−30665.539	−30665.539	−30665.539	1.1E−11	−30665.539	484
C'	−30665.539	−30665.539	−30665.539	1.1E−11	−30665.539	499
B'	−30665.539	−30665.539	−30665.539	1.0E−11	−30665.539	570
g05/	5126.498					
(28) A'	5126.509	5129.190	5134.550	1.0E+01	5162.690	432
C'	–	–	–	–	–	–
B'	–	–	–	–	–	–
g06/	−6961.814					
A'	−6961.814	−6830.074	−6828.072	8.5E+01	−6672.254	37
C'	−6961.814	−6961.814	−6961.795	1.0E−01	−6961.258	467
B'	−6927.754	−6756.593	−6753.254	1.2E+02	−6480.124	33
g07/	24.306					
A'	24.319	24.395	24.443	1.3E−01	24.948	870
C'	24.335	24.505	24.616	2.6E−01	25.350	872
(26) B'	63.951	106.811	135.417	9.3E+01	479.533	29
g08	−0.095825					
A'	−0.095825	−0.095825	−0.095825	2.8E−17	−0.095825	303
C'	−0.095825	−0.095825	−0.095825	2.5E−17	−0.095825	288
B'	−0.095825	−0.095825	−0.095825	2.6E−17	−0.095825	311
g09	680.630					
A'	680.634	680.674	680.691	6.5E−02	680.880	232
C'	680.634	680.659	680.671	3.3E−02	680.762	235
B'	680.677	781.001	788.336	7.8E+01	952.502	23
g10	7049.248					
A'	7086.363	7445.279	7563.709	4.0E+02	8778.744	873
(4) C'	14171.664	15484.311	15695.524	1.5E+03	17641.811	10
B'	–	–	–	–	–	–
g11/	0.750					
A'	0.750	0.750	0.750	4.1E−05	0.750	74
C'	0.750	0.750	0.750	9.2E−05	0.750	70
(15) B'	0.750	0.755	0.811	8.6E−02	0.997	4
g12/	−1.000000					
A'	−1.000000	−1.000000	−1.000000	7.3E−11	−1.000000	86
C'	−1.000000	−1.000000	−1.000000	2.1E−10	−1.000000	85
B'	−1.000000	−1.000000	−1.000000	4.6E−12	−1.000000	86
g13/	0.053950					
A'	0.053950	0.055225	0.107534	1.3E−01	0.444116	493
(2) C'	0.063992	0.075856	0.075856	1.7E−02	0.087719	874
B'	–	–	–	–	–	–

used has been halved, i.e. all runs are terminated after $G = 875$ generations except for g12, which is run for $G = 87$ generations, in this way the number of function evaluations is equivalent to that in [7].

For each of the test functions 30 independent runs are performed using the six different ranking strategies denoted by A to C', and whose search behavior was illustrated using the artificial test function in figure 5. The statistics for these runs are summarized in two tables. The first is table II where the feasible solutions are always ranked highest and according to their objective function value. The constraints are treated by A the penalty function $\boldsymbol{w} = \mathbf{1}$ and $\beta = 2$, then B and C for when the infeasible solutions are ranked according to their Pareto levels specified by problems (6) and (7) respectively. The second set of results are given in table III, with the constraint violations treated in the same manner but now the objective function plays also a role in the ranking, this is achieved using the stochastic ranking [7]. These runs are labeled as before by A', B' and C' respectively.

The results for runs A and A' are similar to those in [7]. The only difference between the algorithm in [7] and the one here is the parent size and the manner by which self-adaptation is facilitated. In this case allowing the objective function to influence the ranking improves the quality of search for test functions g03, g05, g11, g13, which have non-linear equality constraints, and g12 whose global constrained optimum is also the unconstrained global optimum. In general the performance of the search is not affected when the objective function is allowed to influence the search. The exception is `g06`, however, in [8] this was shown to be the result of the rotational invariance of the non-isotopic search distribution used (see also results in next section).

The aim here is to investigate how the search bias introduced by the constraint violations influences search performance. The multi-objective formulation allows the feasible solutions to be found without the bias introduced by a penalty function. In cases C and C' the infeasible solution are ranked according to their Pareto level specified by problem (7). When comparing A with C in table II an insignificant difference is observed in search behavior with the exception of `g05` and `g07`. The difference is more apparent when A' with C' are compared in table III. The search behavior remains similar with the exception of `g05`, `g10` and `g13` where it has become difficult to find feasible solutions. The number of feasible solution found, for the 30 independent runs, are listed in parenthesis in the left most column when fewer than 30. The corresponding statistics is also based on this number. In tables II and III the variable G_m denotes the median number of generations needed to locate the best solution.

The problem of finding feasible solutions is an even greater issue when the infeasible solutions are ranked according to their Pareto levels specified by (6), these are case studies B and B'. An interesting search behavior is observed in case B for `g11` and `g12`. Because the objective function is now also used in determining the Pareto levels the search has been drawn to the location of the constrained feasible global optimum. This is the type of search behavior one desires from a multi-objective constraint handling method. Recall figure 1 where the feasible global optimum is part of the Pareto optimal set (hatched sector). However, these two problems are of a low dimension and it may be the case that in practice, as seen by the performance on the other benchmark functions, that this type of search is difficult to attain.

The difficulty in using Pareto ranking for guiding the search to feasible regions has also been illustrated in a separate study [22] where four different multi-objective techniques are compared. These methods fail to find feasible solutions to `g13`, in most cases for `g05` and in some for `g01`, `g07`, `g08` and `g10`.

C. Improved search

The purpose of the previous sections was to illustrate the effect different ranking (constraint handling) methods have on search performance. The results tend to indicate that the multi-objective methods are not as effective as one may at first have thought. They seem to spend too much time exploring the infeasible regions of the Pareto front. The penalty function method is more effective for the commonly used benchmark problems. It was also demonstrated that letting the objective function influence the ranking improves search performance for some of the problems without degrading the performance significantly for the others. However, the quality can only be improved so much using a proper constraint handling technique. The search operators also influence search performance. This is illustrated here using the improved ES version described in figure 4.

TABLE IV

THE IMPROVED $(60, 400)$ ES EXPERIMENT WITH STOCHASTIC RANKING D' AND OVER-PENALIZED APPROACH D.

`fcn/rnk`	best	median	mean	st. dev.	worst	G_m
`g01/`	−15.000					
D'	−15.000	−15.000	−15.000	5.8E−14	−15.000	875
D	−15.000	−15.000	−15.000	1.3E−15	−15.000	861
`g02/`	−0.803619					
D'	−0.803619	−0.793082	−0.782715	2.2E−02	−0.723591	874
D	−0.803619	−0.780843	−0.776283	2.3E−02	−0.712818	875
`g03/`	−1.000					
D'	−1.001	−1.001	−1.001	8.2E−09	−1.001	873
D	−0.747	−0.210	−0.257	1.9E−01	−0.031	875
`g04/`	−30665.539					
D'	−30665.539	−30665.539	−30665.539	1.1E−11	−30665.539	480
D	−30665.539	−30665.539	−30665.539	1.1E−11	−30665.539	403
`g05/`	5126.498					
D'	5126.497	5126.497	5126.497	7.2E−13	5126.497	489
D	5126.497	5173.967	5268.610	2.0E+02	5826.807	875
`g06/`	−6961.814					
D'	−6961.814	−6961.814	−6961.814	1.9E−12	−6961.814	422
D	−6961.814	−6961.814	−6961.814	1.9E−12	−6961.814	312
`g07/`	24.306					
D'	24.306	24.306	24.306	6.3E−05	24.306	875
D	24.306	24.306	24.307	1.3E−03	24.311	875
`g08`	−0.095825					
D'	−0.095825	−0.095825	−0.095825	2.7E−17	−0.095825	400
D	−0.095825	−0.095825	−0.095825	5.1E−17	−0.095825	409
`g09`	680.630					
D'	680.630	680.630	680.630	3.2E−13	680.630	678
D	680.630	680.630	680.630	1.7E−07	680.630	599
`g10`	7049.248					
D'	7049.248	7049.248	7049.250	3.2E−03	7049.270	872
D	7049.248	7049.248	7049.248	7.5E−04	7049.252	770
`g11/`	0.750					
D'	0.750	0.750	0.750	1.1E−16	0.750	343
D	0.750	0.754	0.756	6.9E−03	0.774	875
`g12/`	−1.000000					
D'	−1.000000	−1.000000	−1.000000	1.2E−09	−1.000000	84
D	−1.000000	−0.999954	0.999889	1.5E−04	−0.999385	78
`g13/`	0.053950					
D'	0.053942	0.053942	0.066770	7.0E−02	0.438803	559
D	0.447118	0.998918	0.964323	1.2E−01	0.999225	875

The new search distribution attempts to overcome the problem of a search bias aligned with the coordinate axis. The method introduces a new parameter γ. This parameter is used to scale the step length. In general a smaller step length is more likely to result in an improvement. Typically a step length reduction of around 0.85 is used in the ES literature. Using this value for γ the over-penalty method is compared with the stochastic ranking method in table IV. These experiments are labeled D and D' respectively. Here, like before, allowing the objective function to influence the ranking of infeasible solutions (using the stochastic ranking) is more effective. However, the results are of a much higher quality. Indeed global optimal solutions are found in all cases and consistently in 11 out of the 13 cases. Improved search performance typically means one has made some assumptions about the function studied. These assumptions may not hold for all functions and therefore the likelihood of being trapped in local minima is greater. This would seem to be the case for function `g13`. Although the global optimum is found

TABLE V

STATISTICS FOR 100 INDEPENDENT RUNS OF THE IMPROVED $(60, 400)$ ES WITH STOCHASTIC RANKING USING 3 DIFFERENT SETTINGS FOR γ.

fcn/γ	best	median	mean	st. dev.	worst	G_m
g01/	−15.000					
0.60	−15.000	−15.000	−15.000	1.9E−13	−15.000	873
0.85	−15.000	−15.000	−15.000	1.3E−13	−15.000	873
1.10	−15.000	−15.000	−15.000	1.4E−13	−15.000	873
g02/	−0.803619					
0.60	−0.803619	−0.780843	−0.775360	2.5E−02	−0.669854	871
0.85	−0.803619	−0.779581	−0.772078	2.6E−02	−0.683055	873
1.10	−0.803619	−0.773304	−0.768157	2.8E−02	−0.681114	873
g03/	−1.000					
0.60	−1.001	−1.001	−1.001	2.1E−11	−1.001	869
0.85	−1.001	−1.001	−1.001	6.0E−09	−1.001	873
1.10	−1.001	−1.001	−1.001	6.0E−09	−1.001	874
g04/	−30665.539					
0.60	−30665.539	−30665.539	−30665.539	2.9E−11	−30665.539	519
0.85	−30665.539	−30665.539	−30665.539	2.2E−11	−30665.539	507
1.10	−30665.539	−30665.539	−30665.539	2.2E−11	−30665.539	519
g05/	5126.498					
0.60	5126.497	5126.497	5126.497	6.3E−12	5126.497	525
0.85	5126.497	5126.497	5126.497	6.2E−12	5126.497	487
1.10	5126.497	5126.497	5126.497	6.1E−12	5126.497	474
g06/	−6961.814					
0.60	−6961.814	−6961.814	−6961.814	6.4E−12	−6961.814	486
0.85	−6961.814	−6961.814	−6961.814	6.4E−12	−6961.814	427
1.10	−6961.814	−6961.814	−6961.814	6.4E−12	−6961.814	404
g07/	24.306					
0.60	24.306	24.306	24.306	1.0E−04	24.307	874
0.85	24.306	24.306	24.306	2.7E−04	24.308	874
1.10	24.306	24.306	24.307	8.0E−04	24.313	874
g08/	−0.095825					
0.60	−0.095825	−0.095825	−0.095825	4.2E−17	−0.095825	364
0.85	−0.095825	−0.095825	−0.095825	4.2E−17	−0.095825	328
1.10	−0.095825	−0.095825	−0.095825	4.2E−17	−0.095825	285
g09/	680.630					
0.60	680.630	680.630	680.630	4.5E−13	680.630	706
0.85	680.630	680.630	680.630	4.6E−13	680.630	672
1.10	680.630	680.630	680.630	3.6E−11	680.630	715
g10/	7049.248					
0.60	7049.248	7049.248	7049.248	1.1E−12	7049.248	836
0.85	7049.248	7049.248	7049.249	4.9E−03	7049.296	874
1.10	7049.248	7049.253	7049.300	1.8E−01	7050.432	874
g11/	0.750					
0.60	0.750	0.750	0.750	1.8E−15	0.750	473
0.85	0.750	0.750	0.750	1.8E−15	0.750	359
1.10	0.750	0.750	0.750	1.8E−15	0.750	330
g12/	−1.000000					
0.60	−1.000000	−1.000000	−1.000000	1.8E−09	−1.000000	84
0.85	−1.000000	−1.000000	−1.000000	9.6E−10	−1.000000	84
1.10	−1.000000	−1.000000	−1.000000	7.1E−09	−1.000000	85
g13/	0.053950					
0.60	0.053942	0.053942	0.134762	1.5E−01	0.438803	593
0.85	0.053942	0.053942	0.096276	1.2E−01	0.438803	570
1.10	0.053942	0.053942	0.100125	1.3E−01	0.438803	574

TABLE VI

STATISTICS FOR 100 INDEPENDENT RUNS OF THE IMPROVED (μ, λ) ES WITH STOCHASTIC RANKING USING DIFFERENT POPULATION SIZES.

$\frac{\text{fcn}}{(\mu,\lambda)}$	best	median	mean	st. dev.	worst	$\frac{\text{feval}}{400}$
g01/	−15.000					
(60, 400)	−15.000	−15.000	−15.000	1.3E−13	−15.000	873
(30, 200)	−15.000	−15.000	−15.000	0.0E+00	−15.000	520
(15, 100)	−15.000	−15.000	−15.000	1.6E−16	−15.000	305
g02/	−0.803619					
(60, 400)	−0.803619	−0.779581	−0.772078	2.6E−02	−0.683055	873
(30, 200)	−0.803619	−0.770400	−0.765099	2.9E−02	−0.686574	875
(15, 100)	−0.803619	−0.760456	−0.753209	3.7E−02	−0.609330	874
g03/	−1.000					
(60, 400)	−1.001	−1.001	−1.001	6.0E−09	−1.001	873
(30, 200)	−1.001	−1.001	−1.001	7.0E−07	−1.001	875
(15, 100)	−1.001	−1.001	−1.001	1.7E−05	−1.001	849
g04/	−30665.539					
(60, 400)	−30665.539	−30665.539	−30665.539	2.2E−11	−30665.539	507
(30, 200)	−30665.539	−30665.539	−30665.539	2.2E−11	−30665.539	228
(15, 100)	−30665.539	−30665.539	−30665.539	2.2E−11	−30665.539	166
g05/	5126.498					
(60, 400)	5126.497	5126.497	5126.497	6.2E−12	5126.497	487
(30, 200)	5126.497	5126.497	5126.497	6.0E−12	5126.497	264
(15, 100)	5126.497	5126.497	5126.497	5.8E−12	5126.497	155
g06/	−6961.814					
(60, 400)	−6961.814	−6961.814	−6961.814	6.4E−12	−6961.814	427
(30, 200)	−6961.814	−6961.814	−6961.814	6.4E−12	−6961.814	247
(15, 100)	−6961.814	−6961.814	−6961.814	6.4E−12	−6961.814	140
g07/	24.306					
(60, 400)	24.306	24.306	24.306	2.7E−04	24.308	874
(30, 200)	24.306	24.308	24.310	7.1E−03	24.355	875
(15, 100)	24.306	24.323	24.337	4.1E−02	24.635	875
g08	−0.095825					
(60, 400)	−0.095825	−0.095825	−0.095825	4.2E−17	−0.095825	328
(30, 200)	−0.095825	−0.095825	−0.095825	4.2E−17	−0.095825	214
(15, 100)	−0.095825	−0.095825	−0.095825	4.2E−17	−0.095825	124
g09	680.630					
(60, 400)	680.630	680.630	680.630	4.6E−13	680.630	672
(30, 200)	680.630	680.630	680.630	1.5E−06	680.630	798
(15, 100)	680.630	680.630	680.630	7.4E−04	680.635	775
g10	7049.248					
(60, 400)	7049.248	7049.248	7049.249	4.9E−03	7049.296	874
(30, 200)	7049.248	7049.375	7050.109	2.7E+00	7073.069	875
(15, 100)	7049.404	7064.109	7082.227	4.2E+01	7258.540	860
g11/	0.750					
(60, 400)	0.750	0.750	0.750	1.8E−15	0.750	359
(30, 200)	0.750	0.750	0.750	1.8E−15	0.750	206
(15, 100)	0.750	0.750	0.750	1.8E−15	0.750	116
g12/	−1.000000					
(60, 400)	−1.000000	−1.000000	−1.000000	9.6E−10	−1.000000	84
(30, 200)	−1.000000	−1.000000	−1.000000	2.4E−15	−1.000000	85
(15, 100)	−1.000000	−1.000000	−1.000000	0.0E+00	−1.000000	51
g13/	0.053950					
(60, 400)	0.053942	0.053942	0.096276	1.2E−01	0.438803	570
(30, 200)	0.053942	0.053942	0.100125	1.2E−01	0.438803	314
(15, 100)	0.053942	0.053942	0.111671	1.4E−01	0.438804	273

consistently for this function, still one or two out of the runs are trapped in a local minimum with a function value of 0.438803. Another function whose optimum was not found consistently is g02. This benchmark function is known to have a very rugged fitness landscape and is in general the most difficult to solve of these functions.

In order to illustrate the effect of the new parameter γ on performance another set of experiments are run. This time 100 independent runs are performed for $\gamma = 0.6$, 0.85 and 1.1 using the $(60, 400)$ ES and stochastic ranking, these results are depicted in table V. From this table it may be seen that a smaller value for γ results in even further improvement for g10, however, smaller steps sizes also means slower convergence. In general the overall performance is not sensitive to the setting of γ. Finally, one may be interested in the effect of the parent number. Again three new runs are performed, $(15, 100)$, $(30, 200)$, $(60, 400)$, each using the same number of function evaluations as before. Here $\gamma \approx 0.85$ and the stochastic ranking is used. The results are given in table VI. These results show that most functions can be solved using a fewer number of function evaluations, i.e a smaller population. The exceptions are g02, g03, g07, g09 and g10 which benefit from using larger populations.

V. Conclusions

This paper shows in depth the importance of search bias [23] in constrained optimization. Different constraint handling methods and search distributions create different search biases for constrained evolutionary optimization. As a result, infeasible individuals may enter a feasible region from very different points depending on this bias. An artificial test function was created to illustrate this search behavior.

Using the multi-objective formulation of constrained optimization, infeasible individuals may drift into a feasible region in a bias-free manner. Of the two multi-objective approaches presented in this paper, the one based solely on constraint violations (7) in determining the Pareto levels is more likely to locate feasible solutions than (6), which also includes the objective function. However, in general finding feasible solutions using the multi-objective technique is difficult since most of the time is spent on searching infeasible regions. The use of a non-dominated rank removes the need for setting a search bias. However, this does not eliminate the need for having a bias in order to locate feasible solutions. Introducing a search bias is equivalent to making hidden assumptions about a problem. It turns out that these assumptions, i.e. using the penalty function to bias the search towards the feasible region, is a good idea for 13 test functions but a bad idea for our artificial test function. These results give us some insights into when the penalty function can be expected to work in practice.

A proper constraint handling method often needs to be considered in conjunction with an appropriate search algorithm. Improved search methods are usually necessary in constrained optimization as illustrated by our improved ES algorithm. However, an improvement made in efficiency and effectiveness for some problems, whether due to the constraint handling method or search operators, comes at the cost of making some assumptions about the functions being optimized. As a consequence, the likelihood of being trapped in local minima for some other functions may be greater. This is in agreement with the no-free-lunch theorem [24].

References

[1] C. A. C. Coello, "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, January 2002.

[2] K. Deb, *Multi-Objective Optimization using Evolutionary Algorithms*. Wiley & Sons, Ltd, 2001.

[3] 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. Selected Papers*, T. C. Fogarty, Ed. Sheffield, U.K.: Springer-Verlag, 1995, pp. 166–180.

[4] T. Ray, K. Tai, and K. C. Seow, "An evolutionary algorithm for constrained optimization," in *Genetic and Evolutionary Computing Conference (GECCO)*, 2000.

[5] A. Hernández Aguirre, S. Botello Rionda, G. Lizárraga Lizárraga, and C. A. Coello Coello, "IS-PAES: A Constraint-Handling Technique Based on Multiobjective Optimization Concepts," in *Evolutionary Multi-Criterion Optimization. Second International Conference, EMO 2003*. Faro, Portugal: Springer. Lecture Notes in Computer Science. Volume 2632, April 2003, pp. 73–87.

[6] S. Koziel and Z. Michalewicz, "Evolutionary algorithms, homomorphous mappings, and constrained parame ter optimization," *Evolutionary Computation*, vol. 7, no. 1, pp. 19–44, 1999.

[7] T. P. Runarsson and X. Yao, "Stochastic ranking for constrained evolutionary optimization," *IEEE Transactions on Evolutionary Computation*, vol. 4, no. 3, pp. 284–294, 2000.

[8] ——, *Evolutionary Optimization*. Kluwer Academic Publishers, 2002, ch. Constrained Evolutionary Optimization: The penalty function approach., pp. 87–113.

[9] Z. Michalewicz, K. Deb, M. Schmidt, and T. Stidsen, "Test-case generator for nonlinear continuous parameter optimization techniques," *IEEE Transactions on Evolutionary Computation*, vol. 4, no. 3, pp. 197–215, 2000.

[10] H. T. Kung, F. Luccio, and F. P. Preparata, "On finding the maxima of a set of vectors," *Journal of the Association for Computing Machinery*, vol. 22, no. 4, pp. 496–476, 1975.

[11] H.-G. Beyer, "Evolutionary algorithms in noisy environments: theoretical issues and guidelines for practice," *Computer Methods in Applied Mechanics and Engineering*, vol. 186, no. 2–4, pp. 239–267, 2000.

[12] N. Hansen and A. Ostermeier, "Completely derandomized self-adaptation in evolution stategies," *Evolutionary Computation*, vol. 2, no. 9, pp. 159–195, 2001.

[13] H.-P. Schwefel, *Evolution and Optimum Seeking*. New-York: Wiley, 1995.

[14] H.-G. Beyer, *The Theory of Evolution Strategies*. Berlin: Springer-Verlag, 2001.

[15] I. Rechenberg, *Evolutionstrategie '94*. Stuttgart: Frommann-Holzboog, 1994.

[16] A. Ostermeier, A. Gawelczyk, and N. Hansen, "A derandomized approach to self-adaptation of evolution strategies," *Evolutionary Computation*, vol. 2, no. 4, pp. 369–380, 1994.

[17] T. P. Runarsson, "Reducing random fluctuations in mutative self-adaptation," in *Parallel Problem Solving from Nature VII (PPSN-2002)*, ser. LNCS, vol. 2439. Granada, Spain: Springer Verlag, 2002, pp. 194–203.

[18] R. Salomon, "Some comments on evolutionary algorithm theory," *Evolutionary Computation Journal*, vol. 4, no. 4, pp. 405–415, 1996.

[19] J. A. Nelder and R. Mead, "A simplex method for function minimization," *The Computer Journal*, vol. 7, pp. 308–313, 1965.

[20] R. Storn and K. Price, "Differential evolution - a simple and efficient heuristic for global optimization over continuous spaces," *Journal of Global Optimization*, vol. 11, pp. 341 – 359, 1997.

[21] C. A. Coello Coello, "Treating Constraints as Objectives for Single-Objective Evolutionary Optimization," *Engineering Optimization*, vol. 32, no. 3, pp. 275–308, 2000.

[22] E. Mezura-Montes and C. A. C. Coello, "A numerical comparison of some multiobjective-based techniques to handle constraints in genetic algorithms," Departamento de Ingeniera Elctrica, CINVESTAV-IPN, México, Tech. Rep. Technical Report EVOCINV-03-2002, 2002.

[23] X. Yao, Y. Liu, and G. Lin, "Evolutionary programming made faster," *IEEE Transactions on Evolutionary Computation*, vol. 3, no. 2, pp. 82–102, July 1999.

[24] D. H. Wolpert and W. G. Macready, "No free lunch theorems for optimization," *IEEE Transactions on Evolutionary Computation*, vol. 1, no. 1, pp. 67–82, 1997.

Appendix

`g01`

Minimize:

$$f(\boldsymbol{x}) = 5\sum_{i=1}^{4} x_i - 5\sum_{i=1}^{4} x_i^2 - \sum_{i=5}^{13} x_i \qquad (15)$$

subject to:

$$\begin{aligned}
\mathrm{g}_1(\boldsymbol{x}) &= 2x_1 + 2x_2 + x_{10} + x_{11} - 10 \le 0 \\
\mathrm{g}_2(\boldsymbol{x}) &= 2x_1 + 2x_3 + x_{10} + x_{12} - 10 \le 0 \\
\mathrm{g}_3(\boldsymbol{x}) &= 2x_2 + 2x_3 + x_{11} + x_{12} - 10 \le 0 \\
\mathrm{g}_4(\boldsymbol{x}) &= -8x_1 + x_{10} \le 0 \\
\mathrm{g}_5(\boldsymbol{x}) &= -8x_2 + x_{11} \le 0 \\
\mathrm{g}_6(\boldsymbol{x}) &= -8x_3 + x_{12} \le 0 \\
\mathrm{g}_7(\boldsymbol{x}) &= -2x_4 - x_5 + x_{10} \le 0 \\
\mathrm{g}_8(\boldsymbol{x}) &= -2x_6 - x_7 + x_{11} \le 0 \\
\mathrm{g}_9(\boldsymbol{x}) &= -2x_8 - x_9 + x_{12} \le 0
\end{aligned}$$

where the bounds are $0 \le x_i \le 1$ ($i = 1, \dots, 9$), $0 \le x_i \le 100$ ($i = 10, 11, 12$) and $0 \le x_{13} \le 1$. The global minimum is at $\boldsymbol{x}^* = (1, 1, 1, 1, 1, 1, 1, 1, 1, 3, 3, 3, 1)$ where six constraints are active ($\mathrm{g}_1, \mathrm{g}_2, \mathrm{g}_3, \mathrm{g}_7, \mathrm{g}_8$ and g_9) and $f(\boldsymbol{x}^*) = -15$.

`g02`

Maximize:

$$f(\boldsymbol{x}) = \left| \frac{\sum_{i=1}^{n} \cos^4(x_i) - 2\prod_{i=1}^{n} \cos^2(x_i)}{\sqrt{\sum_{i=1}^{n} i x_i^2}} \right| \quad (16)$$

subject to:

$$\mathrm{g}_1(\boldsymbol{x}) = 0.75 - \prod_{i=1}^{n} x_i \le 0$$

$$\mathrm{g}_2(\boldsymbol{x}) = \sum_{i=1}^{n} x_i - 7.5n \le 0$$

where $n = 20$ and $0 \le x_i \le 10$ ($i = 1, \dots, n$). The global maximum is unknown, the best we found is $f(\boldsymbol{x}^*) = 0.803619$, constraint g_1 is close to being active ($\mathrm{g}_1 = -10^{-8}$).

`g03`

Maximize:

$$f(\boldsymbol{x}) = (\sqrt{n})^n \prod_{i=1}^{n} x_i \quad (17)$$

$$\mathrm{h}_1(\boldsymbol{x}) = \sum_{i=1}^{n} x_i^2 - 1 = 0$$

where $n = 10$ and $0 \le x_i \le 1$ ($i = 1, \dots, n$). The global maximum is at $x_i^* = 1/\sqrt{n}$ ($i = 1, \dots, n$) where $f(\boldsymbol{x}^*) = 1$.

`g04`

Minimize:

$$\begin{aligned} f(\boldsymbol{x}) &= 5.3578547x_3^2 + 0.8356891x_1x_5 + \\ &\quad 37.293239x_1 - 40792.141 \end{aligned} \quad (18)$$

subject to:

$$\begin{aligned}
\mathrm{g}_1(\boldsymbol{x}) &= 85.334407 + 0.0056858x_2x_5 + \\ &\quad 0.0006262x_1x_4 - 0.0022053x_3x_5 - 92 \le 0 \\
\mathrm{g}_2(\boldsymbol{x}) &= -85.334407 - 0.0056858x_2x_5 - \\ &\quad 0.0006262x_1x_4 + 0.0022053x_3x_5 \le 0 \\
\mathrm{g}_3(\boldsymbol{x}) &= 80.51249 + 0.0071317x_2x_5 + \\ &\quad 0.0029955x_1x_2 + 0.0021813x_3^2 - 110 \le 0 \\
\mathrm{g}_4(\boldsymbol{x}) &= -80.51249 - 0.0071317x_2x_5 - \\ &\quad 0.0029955x_1x_2 - 0.0021813x_3^2 + 90 \le 0 \\
\mathrm{g}_5(\boldsymbol{x}) &= 9.300961 + 0.0047026x_3x_5 + \\ &\quad 0.0012547x_1x_3 + 0.0019085x_3x_4 - 25 \le 0 \\
\mathrm{g}_6(\boldsymbol{x}) &= -9.300961 - 0.0047026x_3x_5 - \\ &\quad 0.0012547x_1x_3 - 0.0019085x_3x_4 + 20 \le 0
\end{aligned}$$

where $78 \le x_1 \le 102$, $33 \le x_2 \le 45$ and $27 \le x_i \le 45$ ($i = 3, 4, 5$). The optimum solution is $\boldsymbol{x}^*$ =(78, 33, 29.995256025682, 45, 36.775812905788) where $f(\boldsymbol{x}^*) = -30665.539$. Two constraints are active (g_1 and g_6).

`g05`

Minimize:

$$f(\boldsymbol{x}) = 3x_1 + 0.000001x_1^3 + 2x_2 + (0.000002/3)x_2^3 \quad (19)$$

subject to:

$$\begin{aligned}
\mathrm{g}_1(\boldsymbol{x}) &= -x_4 + x_3 - 0.55 \le 0 \\
\mathrm{g}_2(\boldsymbol{x}) &= -x_3 + x_4 - 0.55 \le 0 \\
\mathrm{h}_3(\boldsymbol{x}) &= 1000\sin(-x_3 - 0.25) + 1000\sin(-x_4 - 0.25) \\ &\quad +894.8 - x_1 = 0 \\
\mathrm{h}_4(\boldsymbol{x}) &= 1000\sin(x_3 - 0.25) + 1000\sin(x_3 - x_4 - 0.25) \\ &\quad +894.8 - x_2 = 0 \\
\mathrm{h}_5(\boldsymbol{x}) &= 1000\sin(x_4 - 0.25) + 1000\sin(x_4 - x_3 - 0.25) \\ &\quad +1294.8 = 0
\end{aligned}$$

where $0 \le x_1 \le 1200$, $0 \le x_2 \le 1200$, $-0.55 \le x_3 \le 0.55$ and $-0.55 \le x_4 \le 0.55$. The best known solution [6] $\boldsymbol{x}^* = (679.9453, 1026.067, 0.1188764, -0.3962336)$ where $f(\boldsymbol{x}^*) = 5126.4981$.

`g06`

Minimize:

$$f(\boldsymbol{x}) = (x_1 - 10)^3 + (x_2 - 20)^3 \quad (20)$$

subject to:

$$\mathrm{g}_1(\boldsymbol{x}) = -(x_1 - 5)^2 - (x_2 - 5)^2 + 100 \le 0$$

$$\mathrm{g}_2(\boldsymbol{x}) = (x_1 - 6)^2 + (x_2 - 5)^2 - 82.81 \le 0$$

where $13 \le x_1 \le 100$ and $0 \le x_2 \le 100$. The optimum solution is $\boldsymbol{x}^* = (14.095, 0.84296)$ where $f(\boldsymbol{x}^*) = -6961.81388$. Both constraints are active.

`g07`

Minimize:

$$\begin{aligned} f(\boldsymbol{x}) &= x_1^2 + x_2^2 + x_1x_2 - 14x_1 - 16x_2 + \\ &\quad (x_3 - 10)^2 + 4(x_4 - 5)^2 + (x_5 - 3)^2 + \\ &\quad 2(x_6 - 1)^2 + 5x_7^2 + 7(x_8 - 11)^2 + \\ &\quad 2(x_9 - 10)^2 + (x_{10} - 7)^2 + 45 \end{aligned} \quad (21)$$

subject to:

$$\begin{aligned}
\mathrm{g}_1(\boldsymbol{x}) &= -105 + 4x_1 + 5x_2 - 3x_7 + 9x_8 \le 0 \\
\mathrm{g}_2(\boldsymbol{x}) &= 10x_1 - 8x_2 - 17x_7 + 2x_8 \le 0 \\
\mathrm{g}_3(\boldsymbol{x}) &= -8x_1 + 2x_2 + 5x_9 - 2x_{10} - 12 \le 0 \\
\mathrm{g}_4(\boldsymbol{x}) &= 3(x_1 - 2)^2 + 4(x_2 - 3)^2 + 2x_3^2 - 7x_4 - 120 \le 0 \\
\mathrm{g}_5(\boldsymbol{x}) &= 5x_1^2 + 8x_2 + (x_3 - 6)^2 - 2x_4 - 40 \le 0 \\
\mathrm{g}_6(\boldsymbol{x}) &= x_1^2 + 2(x_2 - 2)^2 - 2x_1x_2 + 14x_5 - 6x_6 \le 0 \\
\mathrm{g}_7(\boldsymbol{x}) &= 0.5(x_1 - 8)^2 + 2(x_2 - 4)^2 + 3x_5^2 - x_6 - 30 \le 0 \\
\mathrm{g}_8(\boldsymbol{x}) &= -3x_1 + 6x_2 + 12(x_9 - 8)^2 - 7x_{10} \le 0
\end{aligned}$$

where $-10 \le x_i \le 10$ $(i = 1, \ldots, 10)$. The optimum solution is $\boldsymbol{x}^* = $ (2.171996, 2.363683, 8.773926, 5.095984, 0.9906548, 1.430574, 1.321644, 9.828726, 8.280092, 8.375927) where $\texttt{g07}(\boldsymbol{x}^*) = 24.3062091$. Six constraints are active ($\mathrm{g}_1, \mathrm{g}_2, \mathrm{g}_3, \mathrm{g}_4, \mathrm{g}_5$ and g_6).

`g08`

Maximize:

$$f(\boldsymbol{x}) = \frac{\sin^3(2\pi x_1)\sin(2\pi x_2)}{x_1^3(x_1 + x_2)} \tag{22}$$

subject to:

$$\mathrm{g}_1(\boldsymbol{x}) = x_1^2 - x_2 + 1 \le 0$$
$$\mathrm{g}_2(\boldsymbol{x}) = 1 - x_1 + (x_2 - 4)^2 \le 0$$

where $0 \le x_1 \le 10$ and $0 \le x_2 \le 10$. The optimum is located at $\boldsymbol{x}^* = (1.2279713, 4.2453733)$ where $f(\boldsymbol{x}^*) = 0.095825$. The solution lies within the feasible region.

`g09`

Minimize:

$$f(\boldsymbol{x}) = (x_1 - 10)^2 + 5(x_2 - 12)^2 + x_3^4 + 3(x_4 - 11)^2 + 10x_5^6 + 7x_6^2 + x_7^4 - 4x_6x_7 - 10x_6 - 8x_7 \tag{23}$$

subject to:

$$\mathrm{g}_1(\boldsymbol{x}) = -127 + 2x_1^2 + 3x_2^4 + x_3 + 4x_4^2 + 5x_5 \le 0$$
$$\mathrm{g}_2(\boldsymbol{x}) = -282 + 7x_1 + 3x_2 + 10x_3^2 + x_4 - x_5 \le 0$$
$$\mathrm{g}_3(\boldsymbol{x}) = -196 + 23x_1 + x_2^2 + 6x_6^2 - 8x_7 \le 0$$
$$\mathrm{g}_4(\boldsymbol{x}) = 4x_1^2 + x_2^2 - 3x_1x_2 + 2x_3^2 + 5x_6 - 11x_7 \le 0$$

where $-10 \le x_i \le 10$ for $(i = 1, \ldots, 7)$. The optimum solution is $\boldsymbol{x}^* = $ (2.330499, 1.951372, −0.4775414, 4.365726, −0.6244870, 1.038131, 1.594227) where $f(\boldsymbol{x}^*) = 680.6300573$. Two constraints are active (g_1 and g_4).

`g10`

Minimize:

$$f(\boldsymbol{x}) = x_1 + x_2 + x_3 \tag{24}$$

subject to:

$$\mathrm{g}_1(\boldsymbol{x}) = -1 + 0.0025(x_4 + x_6) \le 0$$
$$\mathrm{g}_2(\boldsymbol{x}) = -1 + 0.0025(x_5 + x_7 - x_4) \le 0$$
$$\mathrm{g}_3(\boldsymbol{x}) = -1 + 0.01(x_8 - x_5) \le 0$$
$$\mathrm{g}_4(\boldsymbol{x}) = -x_1x_6 + 833.33252x_4 + 100x_1 - 83333.333 \le 0$$
$$\mathrm{g}_5(\boldsymbol{x}) = -x_2x_7 + 1250x_5 + x_2x_4 - 1250x_4 \le 0$$
$$\mathrm{g}_6(\boldsymbol{x}) = -x_3x_8 + 1250000 + x_3x_5 - 2500x_5 \le 0$$

where $100 \le x_1 \le 10000$, $1000 \le x_i \le 10000$ $(i = 2, 3)$ and $10 \le x_i \le 1000$ $(i = 4, \ldots, 8)$. The optimum solution is $\boldsymbol{x}^* = $ (579.3167, 1359.943, 5110.071, 182.0174, 295.5985, 217.9799, 286.4162, 395.5979) where $f(\boldsymbol{x}^*) = 7049.3307$. Three constraints are active (g_1, g_2 and g_3).

`g11`

Minimize:

$$f(\boldsymbol{x}) = x_1^2 + (x_2 - 1)^2 \tag{25}$$

subject to:

$$\mathrm{h}(\boldsymbol{x}) = x_2 - x_1^2 = 0$$

where $-1 \le x_1 \le 1$ and $-1 \le x_2 \le 1$. The optimum solution is $\boldsymbol{x}^* = (\pm 1/\sqrt{2}, 1/2)$ where $f(\boldsymbol{x}^*) = 0.75$.

`g12`

Maximize:

$$f(\boldsymbol{x}) = (100 - (x_1 - 5)^2 - (x_2 - 5)^2 - (x_3 - 5)^2)/100 \tag{26}$$

subject to:

$$\mathrm{g}(\boldsymbol{x}) = (x_1 - p)^2 + (x_2 - q)^2 + (x_3 - r)^2 - 0.0625 \le 0$$

where $0 \le x_i \le 10$ $(i = 1, 2, 3)$ and $p, q, r = 1, 2, \ldots, 9$. The feasible region of the search space consists of 9^3 disjointed spheres. A point (x_1, x_2, x_3) is feasible if and only if there exist p, q, r such that the above inequality holds. The optimum is located at $\boldsymbol{x}^* = (5, 5, 5)$ where $f(\boldsymbol{x}^*) = 1$. The solution lies within the feasible region.

`g13`

Minimize:

$$f(\boldsymbol{x}) = e^{x_1x_2x_3x_4x_5} \tag{27}$$

subject to:

$$\mathrm{h}_1(\boldsymbol{x}) = x_1^2 + x_2^2 + x_3^2 + x_4^2 + x_5^2 - 10 = 0$$
$$\mathrm{h}_2(\boldsymbol{x}) = x_2x_3 - 5x_4x_5 = 0$$
$$\mathrm{h}_3(\boldsymbol{x}) = x_1^3 + x_2^3 + 1 = 0$$

where $-2.3 \le x_i \le 2.3$ $(i = 1, 2)$ and $-3.2 \le x_i \le 3.2$ $(i = 3, 4, 5)$. The optimum solution is $\boldsymbol{x}^* = $ (−1.717143, 1.595709, 1.827247, −0.7636413, −0.763645) where $f(\boldsymbol{x}^*) = 0.0539498$.

Thomas Philip Runarsson received the M.Sc. in mechanical engineering and Dr. Scient. Ing. degrees from the University of Iceland, Reykjavik, Iceland, in 1995 and 2001, respectively. Since 2001 he is a research professor at the Applied Mathematics and Computer Science division, Science Institute, University of Iceland and adjunct at the Department of Computer Science, University of Iceland. His present research interests include evolutionary computation, global optimization, and statistical learning.

Xin Yao is a professor of computer science in the Natural Computation Group and Director of the Centre of Excellence for Research in Computational Intelligence and Applications (CERCIA) at the University of Birmingham, UK. He is also a Distinguished Visiting Professor of the University of Science and Technology of China in Hefei, and a visiting professor of the Nanjing University of Aeronautics and Astronautics in Nanjing, the Xidian University in Xi'an and the Northeast Normal University in Changchun. He is an IEEE fellow, the editor in chief of IEEE Transactions on Evolutionary Computation, an associate editor or an editorial board member of ten other international journals, and the past chair of IEEE NNS Technical Committee on Evolutionary Computation. He is the recipient of the 2001 IEEE Donald G. Fink Prize Paper Award and has given more than 27 invited keynote/plenary speeches at various international conferences. His major research interests include evolutionary computation, neural network ensembles, global optimization, computational time complexity and data mining.