An Evolutionary Algorithm based Pattern Search Approach for Constrained Optimization

Rituparna Datta* M. Fernanda P. Costa, † Kalyanmoy Deb‡ A. Gaspar-Cunha §

KANGAL REPORT 2013006

Abstract

Constrained optimization is one of the popular research areas since constraints are usually present in most real world optimization problems. The purpose of this work is to develop a gradient free constrained global optimization methodology to solve this type of problems. In the methodology proposed, the single objective constrained optimization problem is solved using a Multi-Objective Evolutionary Algorithm (MOEA) by considering two objectives simultaneously, the original objective function and a measure of constraint violation. The MOEA incorporates a penalty function where the penalty parameter is estimated adaptively. The use of penalty function method will enable to further improve the current best solution by decreasing the level of constraint violation, which is made using a gradient free local search method. The performance of the proposed methodology was assessed on a set of benchmark test problems. The results obtained allowed to conclude that the present approach is competitive when compared with other methods available.

1 Introduction and Motivation

There are many optimization problems, manly in the field of economics, engineering, decision science and operations research, where the objective function and/or some constraint functions can be formulated as non-convex and non-linear functions. Application examples includes areas like transportation, signal processing, production planning, scheduling, project management, structural optimization, and VLSI design [17, 2, 24] etc. to name a few. The main motivation of the present work is to develop an efficient methodology to obtain a global solution for these type of optimization problems.

The mathematical formulation of the problem is:

$$\begin{array}{ll} \text{minimize} & f(\boldsymbol{x}), \\ \text{subject to} & g(\boldsymbol{x}) \geq 0, \\ & x \in \Omega \end{array} \tag{1}$$

where $f : \mathbb{R}^n \to \mathbb{R}$ and $g : \mathbb{R}^n \to \mathbb{R}^m$ are nonlinear continuous functions defined on the search space $\Omega \subseteq \mathbb{R}^n$. Usually, the search space Ω is defined as $\Omega = \{x \in \mathbb{R}^n : -\infty < l \leq x \leq u < \infty\}$. Problems with equality constraints, $h(x) = 0$, are reformulated into the above form using a couple of inequality constraints $h(x) + \gamma \geq 0$ and $-h(x) + \gamma \geq 0$, where γ represents a positive small tolerance ($0 < \gamma \ll 1$). The set $\mathbf{F} = \{x \in \Omega : g(x) \geq 0\}$ defines the feasible region. Since, it is not assumed that the objective and constraint functions are convex, many global and local solutions can exist in the set **F**.

Initially, evolutionary algorithms (EAs) were designed to solve global unconstrained optimization problems, being, after, extended to handle constraints [16, 6, 21, 9]. One of the most popular and simple class of methods to solve globally non-convex constrained optimization problems are based on Penalty functions [16, 18]. In these methods, the penalty function is defined combining a measure of constraint violation with the objective function.

A penalty function method works by increasing the fitness value of the infeasible solutions proportionally to their level of constraint violation. Some of the penalty function based evolutionary research works are available in [20, 26, 35]. One of the drawback of penalty function method is that, it needs a proper estimation of penalty parameter to handle the constraints efficiently, throughout the iterative process. If the penalty parameter is too large, an arbitrary feasible solution

*Mechanical Engineering, IIT Kanpur, India, Email: rdatta@iitk.ac.in
†Centre of Mathematics, University of Minho, Guimaraes, Portugal, Email: mfc@math.uminho.pt
‡Mechanical Engineering, IIT Kanpur, India, Presently at: Michigan State University, East Lansing. Email: deb@iitk.ac.in
§Institute of Polymer and Composites/I3N, University of Minho, Guimaraes, Portugal, Email: agc@dep.uminho.pt

can be returned. On the other hand, if the parameter is too small, more emphasis is given to the objective function and, thus, the constraints can be neglected, which can result in an infeasible solution. These drawback of penalty function approach motivated researchers to develop alternative methods to deal with constraints in global optimization problems.

Deb in [9] proposed a penalty-parameter-less EA approach which efficiently handles constraints using the following criteria: (i) if there are two feasible solutions, the one with less objective function value is selected, (ii) if there are two solutions, of which one feasible and the other infeasible, the feasible solution is selected, (iii) if there are two infeasible solutions, the one with less constraint violation is selected. Some other penalty parameter less constraint handling approaches are available in [6, 5, 15, 34, 14].

In addition to penalty function approach, another idea, that received the attention among evolutionary research community, was to convert the constrained optimization problem into a bi-objective optimization problem. In the bi-objective approach two objectives are simultaneously minimized, one is a measure of the constraint violation and the other is the original objective function. Coello in [4] proposed an approach in which all constraints are treated as objectives. Herein, instead to solve bi-objective problem, the method solved a multi-objective problem. However, this idea is not always appropriate in real world scenario, since the complexity of the problem increases considerably with the number of constraints. Some other studies in bi-objective based constraint handling approaches can be found in [21, 31, 13, 8, 29].

Although evolutionary based optimization methods have proven its efficiency in a large number of problems, they have the weakness of exact convergence. To over come this issue some hybrid evolutionary algorithms have been proposed . Usually, EAs are coupled with other optimization techniques or heuristic methods. To perform this hybridization both the techniques are integrated intelligently to retain the good properties of both techniques. Some hybrid evolutionary methods are available in [28, 12, 3, 27, 32].

Recently in [11], to solve non-convex and non-linear constrained global optimization using an evolutionary technique, the constrained optimization problem was converted into a bi-objective problem:

$$\min_{x \in \Omega} \quad (\, f(x),\ \theta(x)\,)\,,$$

where θ is a non-negative continuous aggregate constraint violation function defined by

$$\theta(x) = \sum_{j=0}^{m} \left|\min\{g_j(x), 0\}\right|.$$

In this approach, a penalty function method is applied to improve the performance EA. Herein, at pre-defined generations of EA, a penalty function is solved by a local approach. First, a cubic polynomial is fitted (using a nonlinear least square formulation) to a set of non-dominated solutions, that were obtained between the measure of the constraint violation and the objective function - the Pareto-optimal front. The slope of this polynomial is used as an approximation to the penalty parameter. Thereafter, given as initial point the best current point (the lesser infeasible point in the Pareto front), the penalty function is solved by a local gradient based approach. Finally, the minimizer of the penalty function is used to replace the worst point in the current Pareto front. This process is repeated until convergence is achieved.

The structure of the present paper is as follows; in section 2 the details of the proposed hybrid evolutionary coupled with pattern search method is described, hereafter called EA-PS method. In section 3, we report the results of the numerical experiments with a set of benchmark problems. Finally, the paper finishes with conclusions and future work in section 4.

2 Proposed Hybrid Evolutionary and Pattern Search Method

In this section the hybrid methodology (EA-PS) used to compute the global solution of problem (1) is described. The hybridization is made by coupling an evolutionary algorithm with a gradient free pattern search method to solve the penalized function.

2.1 No Gradient Information

In [11] the local search uses gradient information to solve the penalty function. However, often the gradient information may not be available. For instance, in *black-box* applications the gradient information of constraints and the objective function are not available and are forbidden to be used. In such situations the herein proposed derivative free local search integrated into the EA, target these type of optimization problems. Therefore, in this work, constrained optimization problems are solved using a derivative free method.

2.2 Pattern Search for Bound Constrained Problems

Direct search methods for unconstrained optimization problems generate a sequence of points $\{x_k\}$ in $\mathbb{R}^n$ with non-increasing objective function values. At each iteration, the objective function is computed at a finite set of trial points to try to find one that yields a lower objective function than the current point. Direct search methods works without using any gradient information and additionally not any derivative approximation is made. Pattern search are one of the popular direct methods in which trial points are computed follow an exact calculations. In the present work we apply a pattern search method, more specifically the Hooke and Jeeves pattern search method [22], to minimize the penalty function:

$$P(\boldsymbol{x}) = f(\boldsymbol{x}) + r\,\theta(\boldsymbol{x}), \tag{2}$$

where $r \geq 0$ is the penalty parameter.

In this section we describe details related to our implementation of this method, in particular, the scheme used to keep the iterates in the set Ω and the termination criteria. In the Hooke-Jeeves method two type of movements are performed iteratively, namely exploratory moves and heuristic pattern moves. In the exploratory move a coordinate search with a step length of Δ_k around the current point x_k is performed. Herein, one coordinated at time of the current point x_k is modified along of positive and negative coordinate directions and the best point (a point with a lower function value) is recorded. The point is updated to the best position at each variable modification. The iteration is considered successful if a best point $\hat{x}_{k+1}$ is found at the end of all variables modifications. Otherwise it is an unsuccessful iteration and the step length Δ_k is reduced.

When the iteration is successful the current and the best points are used to make a pattern move. The $\hat{x}_{k+1} - x_k$ entity defines a promising direction and the pattern search move jump from the best point $\hat{x}_{k+1}$ along that direction and it carries out an exploratory move around the new trial point $\hat{x}_{k+1} + (\hat{x}_{k+1} - x_k)$ instead of the current best $\hat{x}_{k+1}$. Thereafter, in case of successful exploratory move, a new best point is accepted. Otherwise, in case of unsuccessful exploratory move, the pattern search move is not accepted, and the method reduces to an exploratory move around $x_{k+1} \leftarrow \hat{x}_{k+1}$.

In order to maintain the iterates in set Ω in the Hooke Jeeves pattern search method, the iterates are project into this set component-wise, $(x_k)_i = \max(l_i, \min((x_k)_i, u_i))$ for $i = 1, \cdots, n$. To deal with variables with different magnitude, the Hooke Jeeves algorithm implementation uses a step length vector Δ. Given an initial guess $x_0 \in \Omega$, the vector Δ_0 is initialized component-wise as follows:

$$(\Delta_0)_i = \begin{cases} \rho\,(x_0)_i\,, & \text{if } (x_0)_i \neq 0, \\ \rho, & \text{otherwise} \end{cases} \tag{3}$$

where ρ is a positive parameter. Let $\alpha > 1$ be a step reduction factor. The stopping criterion of the pattern search method is defined by $\|\Delta_k\| < \epsilon$, where $\epsilon > 0$ is the termination parameter. The Hooke-Jeeves pattern search method is described in Algorithm 1.

2.3 Hybrid EA-PS method

Flowchart 1 describes the steps of the proposed approach. First, a single objective constrained optimization is converted into a bi-objective problem. Here, Non-dominated Sorting Genetic Algorithm-II (NSGA-II) [10] is used to solve the bi-objective problem and for obtaining the Pareto-optimal front. After every 5 generations, non-dominated solutions are identified and a cubic polynomial is fitted to those non-dominated solutions. The slope of this cubic polynomial is used to estimate the penalty parameter of (2). Taking the best current point as the initial guess, the penalty function (2) is minimized using the Hooke and Jeeves pattern search method. The optimal solution of the penalty function is used to replace the worst point in the current Pareto-optimal front. This process is repeated until two consecutive optimal local searched solutions of the penalty functions are less than small positive tolerance and the hybrid EA-PS stops.

3 Simulation Results and Discussions

To validate the proposed EA-PS, a set of six problems is used, out of which five are shown in the Appendix 4. One of these problems is shown below.

The C programming is used for the evolutionary algorithm and Hooke and Jeeves is implemented in Matlab. The simulations are performed on a PC with 2.1 GHz Intel core i3 and 2 GB of RAM. The parameters have been set as follows after an empirical study:

Algorithm 1 Hooke Jeeves Pattern Search Method

Input: Choose a starting point $x_0 \in \Omega$ and initialize Δ_0 using (3). Choose the step reduction factor $\alpha > 1$ and the termination parameter ϵ. Set $k = 0$.

while $\|\Delta_k\| \geq \epsilon$ **do**
    [**Exploratory move** (output: $\hat{x}_{k+1}$)]
    set $minP = P(x_k)$ and $flag = 0$
    set $\hat{x}_{k+1} = x_k$
    for $i = 1$ **to** n **do**
        set $(\hat{x}_{k+1})_i = \max(l_i, \min((x_k)_i + (\Delta_k)_i, u_i))$
        if $P(\hat{x}_{k+1}) < minP$ **then**
            set $minP = P(\hat{x}_{k+1})$
        else
            set $(\hat{x}_{k+1})_i = \max(l_i, \min((x_k)_i - (\Delta_k)_i, u_i))$
            if $P(\hat{x}_{k+1}) < minP$ **then**
                set $minP = P(\hat{x}_{k+1})$
            else
                set $(\hat{x}_{k+1})_i = (x_k)_i$
            end if
        end if
    end for
    set $x_{k+1} = \hat{x}_{k+1}$.
    if $P(x_{k+1}) < P(x_k)$ **then**
        set $flag = 1$ (Exploratory move was successful.)
    end if
    (If it makes some improvements, pursue that direction.)
    [**Pattern search move** (output: x_{k+1})]
    set $\hat{x}_k = x_k$
    while $P(\hat{x}_{k+1}) < P(\hat{x}_k)$ **do**
        set $x_{k+1} = \hat{x}_{k+1}$ and $minP = P(x_{k+1})$
        (Perform the exploratory move around the point $\hat{x}^p_k$.)
        set $\hat{x}^p_k = \max(l, \min(\hat{x}_{k+1} + (\hat{x}_{k+1} - \hat{x}_k), u))$
        set $\hat{x}_k = \hat{x}_{k+1}$
        for $i = 1$ **to** n **do**
            set $(\hat{x}^p_{k+1})_i = \max(l_i, \min((\tilde{x}^p_k)_i + (\Delta_k)_i, u_i))$
            if $P(\hat{x}^p_{k+1}) < minP$ **then**
                set $minP = P(\hat{x}^p_{k+1})$
            else
                $(\hat{x}^p_{k+1})_i = \max(l_i, \min((\hat{x}^p_k)_i - (\Delta_k)_i, u_i))$
                if $P(\hat{x}^p_{k+1}) < minP$ **then**
                    set $minP = P(\hat{x}^p_{k+1})$
                else
                    set $(\hat{x}^p_{k+1})_i = (\hat{x}^p_k)_i$
                end if
            end if
        end for
        set $\hat{x}_{k+1} = \hat{x}^p_{k+1}$
    end while
    if $flag \neq 1$ **then**
        set $\Delta_{k+1} = \Delta_k/\alpha$
    else
        set $\Delta_{k+1} = \Delta_k$
    end if
    set $k = k + 1$
end while

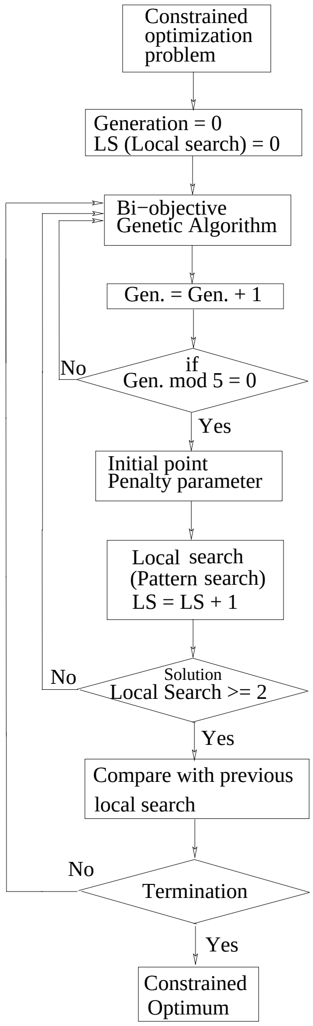

Figure 1: Flowchart of the proposed EA-PS method.

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

The hybrid algorithm is allowed to runs 50 times with different initial populations. First, EA-PS is tested in a two variable problem. Thereafter, the efficiency of the algorithm is tested with the remaining five problems. When difference between the absolute values of two consecutive local searched solutions are less than 10^{-4} we terminated the algorithm.

3.1 Problem P1

First, the following two-variable problem is tested. The problem has two inequality constraints. The constraints are non-linear and non-convex and the first one is active at the optimum [11]:

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

Table 1 shows the total number of function evaluations (FE), which is the sum of the number of function evaluations taken by EA and the Hooke-Jeeves method, and the corresponding objective function values (f). We compare the results with the previous hybrid method [11] that uses gradient information. The Table 1 clearly shows that our best number of function evaluation is better than the previous reported one. However, in terms of median and worst of the number of function evaluations the previous method outperform the EA-PS, which is expected since EA-PS does not use gradient information. But the results are comparable. We can conclude that EA-PS method performs successfully.

Table 1: Function evaluations, FE (NSGA-II and local search) and optimal solution, by the Earlier approach and EA-PS in 50 runs.

		Best	Median	Worst
Single	FE	677 (600 + 77)	733 (600 + 133)	999 (900 + 99)
Penalty [11]	f	0.627380	0.627379	0.627379
EA-PS	FE	672 (600 + 72)	1,342 (1,200 + 142)	3,332 (3,000 + 332)
	f	0.627485	0.627424	0.628774

Table 2 shows similar results for other five problems. In Table 2 we compare our results with the results obtained by three previously developed evolutionary algorithms based constraint handling techniques. This comparison is again made in terms of total number of function evaluations and the corresponding objective values.

Table 3 reports similar results obtained by seven proposed approaches namely HM: Homomorphous Mapping, SR: Stochastic Ranking, ASCHEA: Adaptive Segregational Constraint Handling Evolutionary, SMES: Simple Multi-membered Evolution Strategy, FSA: Filter Simulated Annealing, ATMES: Adaptive Trade-off Model Evolution Strategy, and NM-PSO: Nelder-Mead Particle Swarm Optimization [23, 30, 1, 25, 19, 33, 36]. Based on the results we may conclude that EA-PS has a good performance. EA-PS is able to reach the global optimal solution with the desired accuracy, beside using any gradient information, except with TP4 and TP8 problems.

4 Conclusions

A hybrid evolutionary approach coupled with a pattern search method for global nonlinear constrained optimization is proposed. The advantage of the this method lies on the fact that the local search does not need any gradient information, which may not be available in many instances. In the proposed hybrid method, evolutionary algorithm is used to generate the Pareto-optimal front. At each five generations, a penalty function is minimized, in which its penalty parameter is estimated by the slope of a cubic polynomial that is fitted to the points defined by the Pareto-front. To minimize the penalty function, Hooke and Jeeves pattern search method is used, taking as initial point the best current point in the Pareto-optimal (the point which has the lesser constraint violation measure). The minimizer of the penalty function is used to replace the worst point in the Pareto-front. The proposed method is tested with a set of six constrained optimization problems very well known in literature. In test, the robustness of the hybrid algorithm is tested using different initial population. The total number of function evaluations are compared with three evolutionary based constraint handling methods. In addition to that the best, average and the worst objective function value is also compared with seven previously developed methods. Results shows that the proposed hybrid method is efficient. Since most practical problems are expected to be non-differentiable and discrete, evolutionary algorithms are better off in hybridizing with gradient-free methods, such as Hooke-Jeeves method. The results here are promising and the combined method needs further testing and analysis. In future we plan to apply it to solve problems having equality constraints and some real life constrained optimization problems.

Table 2: Comparison of function evaluations (FE) needed by the EA-PS and three existing earlier approaches [9, 11, 7]. Function evaluations by NSGA-II and local search have been shown separately.

Problem	Penalty Parameter Less Approach [9]			Single Penalty Approach [11]		
	Best	Median	Worst	Best	Median	Worst
TP3 (FE)	65,000	65,000	65,000	2,427	4,676	13,762
NSGA-II+Local				2,000+427	3,000+1,676	11,000+2,762
(f^*)	**−15**	**−15**	−13	**−15**	**−15**	−12
TP4 (FE)	320,080	320,080	320,080	31,367	54,946	100,420
NSGA-II+Local				14,400+16,967	24,600+30,346	45,600+54,820
(f^*)	7,060.221	7,220.026	10,230.834	7,078.625	7,049.943	7,940.678
TP5 (FE)	350,070	350,070	350,070	6,332	15,538	38,942
NSGA-II+Local				3,920+2,412	9,520+6,018	25,200+13,742
(f^*)	680.634	680.642	680.651	**680.630**	680.634	680.876
TP6 (FE)	250,000	250,000	250,000	1,120	2,016	6,880
NSGA-II+Local				800+320	1,200+816	3,600 + 3,280
(f^*)	−30, 665.537	−30, 665.535	−29, 846.654	**−30, 665.539**	**−30, 665.539**	−30, 649, 552
TP8 (FE)	350,000	350,000	350,000	4,880	23,071	83,059
NSGA-II+Local				3,200+1,680	8,000+5,071	44,800+38,259
(f^*)	24.372	24.409	25.075	**24.308**	25.651	31.254
Problem	Adaptive Normalization Approach [7]			EA-PS		
	Best	Median	Worst	Best	Median	Worst
TP3 (FE)	2,333	**2, 856**	**11, 843**	2,959	5,752	32,292
NSGA-II+Local	2,000+ 333	2,000 + 856	8,000 + 3,843	2,000 + 959	3,000 + 1,702	25,000 + 7,292
(f^*)	-12	**−15**	**−15**	-14.968	**−14.993**	**−14.992**
TP4 (FE)	**2, 705**	**27, 235**	**1, 07, 886**	10,064	37,724	1,24,128
NSGA-II+Local	1,200 + 1,505	7,200 + 20,035	45,600 + 62286	9,600 + 464	36,000 + 1,724	87,600 + 36,528
(f^*)	7,049.588	7,059.576	7,065.348	8, 200.0697	7078.2195	7117.6887
TP5 (FE)	**1,961**	11,729	42,617	3,222	**6,682**	**13,379**
NSGA-II+Local	1,120 + 841	7,280 + 4,449	27,440 + 15,177	2,800 + 422	5,040 + 1,582	8,960 + 4,419
(f^*)	680.635	680.631	680.646	680.6387	681.6397	681.0874
TP6 (FE)	1,123	4,183	13,631	**8,396**	**12,679**	**18,327**
NSGA-II+Local	800 + 323	2,400 +1,783	8,400 + 5,231	8,000 + 396	12,000 + 679	16,000 + 2,327
(f^*)	**−30, 665.539**	**−30, 665.539**	**−30, 665.539**	-30665.530	**−30665.540**	**−30665.540**
TP8 (FE)	7,780	68,977	3,54,934	8,712	**85,324**	**1,85,273**
NSGA-II+Local	5,600 + 2,180	41,600 + 27,377	1,600+1673	7,200+1,512	64,000+21,324	1,28,000 + 57,273
(f^*)	24.565	**24.306**	**24.306**	**25.889**	27.309	31.146

Table 3: Comparison of obtained objective function values using EA-PS and seven existing constraint handling approach [a] [23, 30, 1, 25, 19, 33, 36].

Problem	Optima (f^*)		HM [23]	SR [30]	ASCHEA [1]	SMES [25]	FSA [19]	ATMES [33]	NM-PSO [36]	EA-PS
TP3	-15.0	Best	-14.7864	-15.0	-15.0	-15.0	-14.9991	-15.0	-15.0	-14.997
		Mean	-14.7082	-15.0	-14.84	-15.0	-14.9933	-15.0	-15.0	-14.989
		Worst	-14.6154	-15.0		-15.0	-14.9799	-15.0	-15.0	-14.968
TP4	7,049.248	Best	7147.9	7054.316	7061.13	7051.9028	7059.8635	7052.253	7049.2969	7051.03931
		Mean	8163.6	7559.192	7497.434	7253.0470	7509.3210	7250.437	7049.5652	7298.1528
		Worst	9659.3	8835.655		7638.3662	9398.6492	7560.224	7049.9358	8200.0697
TP5	680.630	Best	680.91	680.630	680.630	680.6316	680.6301	680.630	680.6301	680.6387
		Mean	680.16	680.656	680.641	680.6434	680.6364	680.639	680.6301	681.6473
		Worst	683.18	680.763		680.7192	680.6983	680.673	680.6301	682.9548
TP6	-30665.539	Best	-30664.5	-30665.539	-30665.5	-30665.539	-30665.538	-30665.539	-30665.5386	-30665.539
		Mean	-30665.3	-30665.539	-30665.5	-30665.539	-30665.4665	-30665.539	-30665.5386	-30665.261
		Worst	-30645.9	-30665.539		-30665.539	-30664.6880	-30665.539	-30665.5386	-30663.498
TP8	24.306	Best	24.620	24.307	24.3323	24.3267	24.3105	24.306	24.3062	25.889
		Mean	24.826	24.374	24.6636	24.4749	24.3795	24.316	24.4883	28.880
		Worst	25.069	24.642		24.8428	24.6444	24.359	24.7195	31.146

[a] HM: Homomorphous Mapping SR: Stochastic Ranking ASCHEA: Adaptive Segregational Constraint Handling Evolutionary Algorithm SMES: Simple Multi-membered Evolution Strategy FSA: Filter Simulated Annealing ATMES: Adaptive Trade-off Model Evolution Strategy NM-PSO: Nelder-Mead Particle Swarm Optimization

.1 Problem TP3

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

where $0 \leq x_i \leq 1$ for $i = 1, \ldots, 9$, $0 \leq x_i \leq 100$ for $i = 10, 11, 12$, and $0 \leq x_{13} \leq 1$. The minimum point is $\mathbf{x}^* = (1, 1, 1, 1, 1, 1, 1, 1, 1, 3, 3, 3, 1)^T$, where six constraints (g_1, g_2, g_3, g_7, g_8 and g_9) are active and $f(\mathbf{x}^*) = -15$.

.2 Problem TP4

The problem is given as follows:

$$\begin{array}{rl} \text{min.} & f(\mathbf{x}) = x_1 + x_2 + x_3, \\ \text{s.t.} & g_1(\mathbf{x}) \equiv -1 + 0.0025(x_4 + x_6) \leq 0, \\ & g_2(\mathbf{x}) \equiv -1 + 0.0025(x_5 + x_7 - x_4) \leq 0, \\ & g_3(\mathbf{x}) \equiv -1 + 0.01(x_8 - x_5) \leq 0, \\ & g_4(\mathbf{x}) \equiv -x_1x_6 + 833.33252x_4 + 100x_1 - 83333.333 \leq 0, \\ & g_5(\mathbf{x}) \equiv -x_2x_7 + 1250x_5 + x_2x_4 - 1250x_4 \leq 0, \\ & g_6(\mathbf{x}) \equiv -x_3x_8 + 1250000 + x_3x_5 - 2500x_5 \leq 0, \\ & 100 \leq x_1 \leq 10000, 1000 \leq (x_2, x_3) \leq 10000, \\ & 10 \leq (x_4, \ldots, x_8) \leq 1000. \end{array}$$

The minimum point lies at $\mathbf{x}^* = (579.307, 1359.971, 5109.971, 182.018, 295.601, 217.982, 286.417, 395.601)^T$ with a function value $f^* = 7049.280$. All constraints are active at this point.

.3 Problem TP5

The problem is given as follows:

$$\begin{array}{rl} \text{min.} & f(\mathbf{x}) = (x_1 - 10)^2 + 5(x_2 - 12)^2 + x_3^4 + 3(x_4 - 11)^2 \\ & \quad +10x_5^6 + 7x_6^2 + x_7^4 - 4x_6x_7 - 10x_6 - 8x_7, \\ \text{s.t.} & g_1(\mathbf{x}) \equiv -127 + 2x_1^2 + 3x_2^4 + x_3 + 4x_4^2 + 5x_5 \leq 0, \\ & g_2(\mathbf{x}) \equiv -282 + 7x_1 + 3x_2 + 10x_2^3 + x_4 - x_5 \leq 0, \\ & g_3(\mathbf{x}) \equiv -196 + 23x_1 + x_2^2 + 6x_6^2 - 8x_7 \leq 0, \\ & g_4(\mathbf{x}) \equiv 4x_1^2 + x_2^2 - 3x_1x_2 + 2x_3^2 + 5x_6 - 11x_7 \leq 0, \\ & -10 \leq x_i \leq 10, \quad i = 1, \ldots, 7. \end{array}$$

The minimum is at $\mathbf{x}^* = (2.330, 1.951, -0.478, -4.366, -0.624, 1.038, 1.594)^T$ with $f = 680.630$. Constraints g_1 and g_4 are active at the minimum point.

.4 Problem TP6

The problem is given as follows:

$$
\begin{aligned}
\text{min.} \quad & f(\mathbf{x}) = 5.3578547x_3^2 + 0.8356891x_1x_5 + 37.293239x_1 \\
& -40792.141, \\
\text{s.t.} \quad & g_1(\mathbf{x}) \equiv 85.334407 + 0.0056858x_2x_5 + 0.0006262x_1x_4 \\
& \quad -0.0022053x_3x_5 - 92 \leq 0, \\
& g_2(\mathbf{x}) \equiv -85.334407 - 0.0056858x_2x_5 - 0.0006262x_1x_4 \\
& \quad +0.0022053x_3x_5 \leq 0, \\
& g_3(\mathbf{x}) \equiv 80.51249 + 0.0071317x_2x_5 + 0.0029955x_1x_2 \\
& \quad +0.0021813x_3^2 - 110 \leq 0, \\
& g_4(\mathbf{x}) \equiv -80.51249 - 0.0071317x_2x_5 - 0.0029955x_1x_2 \\
& \quad -0.0021813x_3^2 + 90 \leq 0, \\
& g_5(\mathbf{x}) \equiv 9.300961 + 0.0047026x_3x_5 + 0.0012547x_1x_3 \\
& \quad +0.0019085x_3x_4 - 25 \leq 0, \\
& g_6(\mathbf{x}) \equiv -9.300961 - 0.0047026x_3x_5 - 0.0012547x_1x_3 \\
& \quad -0.0019085x_3x_4 + 20 \leq 0, \\
& 78 \leq x_1 \leq 102, 33 \leq x_2 \leq 45, 27 \leq (x_3, x_4, x_5) \leq 45.
\end{aligned}
$$

The minimum is at $\mathbf{x}^* = (78, 33, 29.995, 45, 36.776)^T$ with a function value $f^* = -30665.539$. Constraints g_1 and g_6 are active at the minimum point.

.5 Problem TP8

The problem is given as follows:

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

The minimum is at $\mathbf{x}^* = (2.172, 2.364, 8.774, 5.096, 0.991,$
$1.431, 1.322, 9.829, 8.280, 8.376)^T$ with a function value 24.306. Constraints g_1 to g_6 are active at the minimum point.

References

[1] S Ben Hamida and Marc Schoenauer. Aschea: New results using adaptive segregational constraint handling. In *Evolutionary Computation, 2002. CEC'02. Proceedings of the 2002 Congress on*, volume 1, pages 884–889. IEEE, 2002.

[2] Julian Benjamin. An analysis of inventory and transportation costs in a constrained network. *Transportation Science*, 23(3):177–183, 1989.

[3] HS Bernardino, HJC Barbosa, and ACC Lemonge. A hybrid genetic algorithm for constrained optimization problems in mechanical engineering. In *Evolutionary Computation, 2007. CEC 2007. IEEE Congress on*, pages 646–653. IEEE, 2007.

[4] A Carlos and Coello Coello. Treating constraints as objectives for single-objective evolutionary optimization. *Engineering Optimization+ A35*, 32(3):275–308, 2000.

[5] C.A.C. Coello. Use of a self-adaptive penalty approach for engineering optimization problems. *Computers in Industry*, 41(2):113–127, 2000.

[6] D.W. Coit, A.E. Smith, and D.M. Tate. Adaptive penalty methods for genetic optimization of constrained combinatorial problems. *INFORMS Journal on Computing*, 8:173–182, 1996.

[7] R Datta and K Deb. An adaptive normalization based constrained handling methodology with hybrid bi-objective and penalty function approach. In *Evolutionary Computation (CEC), 2012 IEEE Congress on*, pages 1–8. IEEE, 2012.

[8] Rituparna Datta. Constrained engineering design optimization using a hybrid bi-objective evolutionary-classical methodology. *Simulated Evolution and Learning*, pages 633–637, 2010.

[9] K. Deb. An efficient constraint handling method for genetic algorithms. *Computer Methods in Applied Mechanics and Engineering*, 186(2–4):311–338, 2000.

[10] K. Deb, S. Agrawal, A. Pratap, and T. Meyarivan. A fast and elitist multi-objective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6(2):182–197, 2002.

[11] 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)*, pages 1–8, 2010.

[12] K. Deb and T. Goel. A hybrid multi-objective evolutionary approach to engineering shape design. In *Proceedings of the First International Conference on Evolutionary Multi-Criterion Optimization (EMO-01)*, pages 385–399, 2001.

[13] 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)*, pages 36–45. Springer-Verlag, 2007.

[14] AE Eiben and Zsófia Ruttkay. Self-adaptivity for constraint satisfaction: Learning penalty functions. In *Evolutionary Computation, 1996., Proceedings of IEEE International Conference*, pages 258–261. IEEE, 1996.

[15] Raziyeh Farmani and Jonathan A. Wright. Self-adaptive fitness formulation for constrained optimization. *Evolutionary Computation, IEEE Transactions on*, 7(5):445–455, 2003.

[16] R. Fletcher. An ideal penalty function for constrained optimization. *IMA Journal of Applied Mathematics*, 15(3):319–342, 1975.

[17] Otis Lamont Frost III. An algorithm for linearly constrained adaptive array processing. *Proceedings of the IEEE*, 60(8):926–935, 1972.

[18] R.T. Haftka and JH Starnes. Applications of a quadratic extended interior penalty function for structural optimization. In *AIAA, ASME, and SAE, Structures, Structural Dynamics, and Materials Conference, 16 th, Denver, Colo*, page 1975, 1975.

[19] Abdel-Rahman Hedar and Masao Fukushima. Derivative-free filter simulated annealing method for constrained continuous global optimization. *Journal of Global Optimization*, 35(4):521–549, 2006.

[20] A. Homaifar, S. H.-V. Lai, and X. Qi. Constrained optimization via genetic algorithms. *Simulation*, 62(4):242–254, 1994.

[21] Sung-Chul Hong and Yang-Byung Park. A heuristic for bi-objective vehicle routing with time window constraints. *International Journal of Production Economics*, 62(3):249–258, 1999.

[22] Robert Hooke and To A Jeeves. Direct search solution of numerical and statistical problems. *Journal of the ACM (JACM)*, 8(2):212–229, 1961.

[23] Slawomir Koziel and Zbigniew Michalewicz. Evolutionary algorithms, homomorphous mappings, and constrained parameter optimization. *Evolutionary computation*, 7(1):19–44, 1999.

[24] Daniel Merkle, Martin Middendorf, and Hartmut Schmeck. Ant colony optimization for resource-constrained project scheduling. *Evolutionary Computation, IEEE Transactions on*, 6(4):333–346, 2002.

[25] E. Mezura-Montes and C.A.C. Coello. A simple multimembered evolution strategy to solve constrained optimization problems. *Evolutionary Computation, IEEE Transactions on*, 9(1):1–17, 2005.

[26] Z. Michalewicz and C. Z. Janikow. Handling constraints in genetic algorithms. In *Proceedings of the Fourth International Conference on Genetic Algorithms*, pages 151–157, 1991.

[27] S. Mitsi, K.-D. Bouzakis, D. Sagris, and G. Mansour. Determination of optimum robot base location considering discrete end-effector positions by means of hybrid genetic algorithm. *Robotics and Computer-Integrated Manufacturing*, 24(1):50 – 59, 2008.

[28] H. Myung and J.H. Kim. Hybrid interior-lagrangian penalty based evolutionary optimization. In *Evolutionary Programming VII*, pages 85–94. Springer, 1998.

[29] T. Ray, H. Singh, A. Isaacs, and W. Smith. Infeasibility driven evolutionary algorithm for constrained optimization. In Efrn Mezura-Montes, editor, *Constraint-Handling in Evolutionary Computation*, pages 145–165. Springer, Berlin, 2009.

[30] T.P. Runarsson and X. Yao. Stochastic ranking for constrained evolutionary optimization. *Evolutionary Computation, IEEE Transactions on*, 4(3):284–294, 2000.

[31] 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*, pages 166–180. Springer-Verlag, 1995.

[32] Y. Wang, Z. Cai, Y. Zhou, and Z. Fan. Constrained optimization based on hybrid evolutionary algorithm and adaptive constraint-handling technique. *Structural and Multidisciplinary Optimization*, 37(4):395–413, 2009.

[33] Y. Wang, Z. Cai, Y. Zhou, and W. Zeng. An adaptive tradeoff model for constrained evolutionary optimization. *IEEE Transactions on Evolutionary Computation*, 12(1):80–92, 2008.

[34] ZY Wu and T Walski. Self-adaptive penalty approach compared with other constraint-handling techniques for pipeline optimization. *Journal of water resources planning and management*, 131(3):181–192, 2005.

[35] Ozgur Yeniay. Penalty function methods for constrained optimization with genetic algorithms. *Mathematical and Computational Applications*, 10(1):45–56, 2005.

[36] Erwie Zahara and Chia-Hsin Hu. Solving constrained optimization problems with hybrid particle swarm optimization. *Engineering Optimization*, 40(11):1031–1049, 2008.