Equality Constraint Handling for Surrogate-Assisted Constrained Optimization

Samineh Bagheri
and Wolfgang Konen
Department of Computer Science
TH Köln (University of Applied Sciences)
51643 Gummersbach, Germany
Email: {samineh.bagheri,wolfgang.konen}@th-koeln.de

Thomas Bäck
Department of Computer Science
Leiden University, LIACS,
2333 CA Leiden, The Netherlands
Email: T.H.W.Baeck@liacs.leidenuniv.nl

***Abstract*—Real-world optimization problems are often subject to many constraints. Often, as the volume of the feasible space gets smaller, the problems become more complex. The zero volume of feasible spaces for optimization problems with equality constraints makes them challenging. In this paper, we present an equality constraint handling approach embedded for the first time in a surrogate-assisted optimizer (SACOBRA). The proposed technique starts with an expanded feasible area which gradually shrinks to a volume close to zero. Several well-studied constrained test problems are used as benchmarks and the promising results in terms of efficiency and accuracy are compared with other state-of-the-art algorithms.**

1. Introduction

An optimization problem can be defined as the minimization of an objective function (fitness function) f subject to inequality constraint function(s) $g_1, \ldots, g_m$ and equality constraint function(s) $h_1, \ldots, h_r$:

$$\begin{array}{lll} \text{Minimize} & f(\vec{x}), & \vec{x} \in [\vec{a}, \vec{b}] \subset \mathbb{R}^d \\ \text{subject to} & g_i(\vec{x}) \leq 0, & i = 1, 2, \ldots, m \\ & h_j(\vec{x}) = 0, & j = 1, 2, \ldots, r \end{array} \tag{1}$$

where $\vec{a}$ and $\vec{b}$ define the lower and upper bounds of the search space (a hypercube). By negating the fitness function f a minimization problem can be transformed into a maximization problem without loss of generality.

Since most of the engineering optimization problems belong to the constrained optimization class, in the last years many efforts were devoted to design algorithms to handle this type of problems. The difficulty of solving a constrained optimization problem depends on the type of the fitness function, the number of variables, the number of constraints and the feasibility ratio ρ (ratio between feasible volume and search space volume). In the presence of equality constraints the feasibility ratio ρ is exactly zero. Therefore, it is difficult for numerical techniques to find a fully feasible solution for this class of problems. In general, most of the existing constraint handling algorithms need to modify the equality constraints in order to be able to address them.

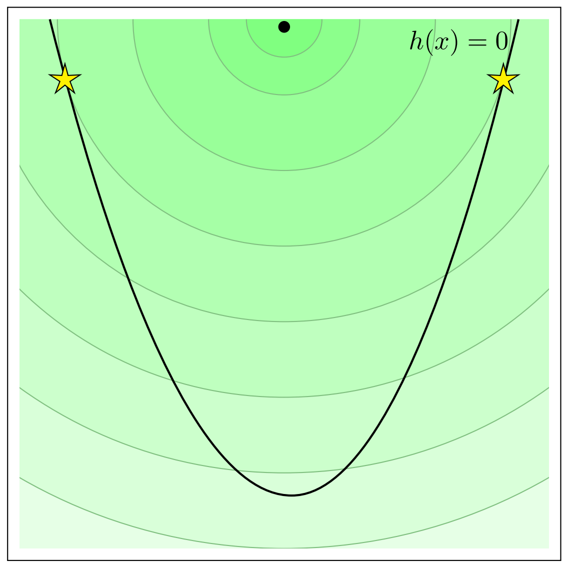

Figure 1. A simple 2-d optimization problem with one equality constraint. The shaded (green) contours depict the fitness function f (darker = smaller). The black curve shows the equality constraint. Feasible solutions are restricted to this line. The black point shows the global optimum of the fitness function which is different from the two optima of the constrained problem shown as the gold stars. If we transform the equality constraint to an inequality by selecting the area below the constraint line as the feasible region, we can expect to converge to one of the correct solutions.

The different modifications used for equality constraints can be divided into five categories of transforming equality constraints $h_j(\vec{x}) = 0$ to:

(a) a pair of inequality constraints

$$\begin{cases} h_j(\vec{x}) & \geq 0, \\ h_j(\vec{x}) & < 0, \end{cases} \tag{2}$$

(b) one inequality constraint (manually chosen),
(c) a tube-shaped region $|h_j(\vec{x})| - \mu_0 \leq 0$ in order to expand the feasible space, where μ_0 is a small positive constant,
(d) a shrinking tube-shaped region $|h_j(\vec{x})| - \mu^{(n)} \leq 0$, where $\mu^{(n)}$ is decaying in time,
(e) a repair mutation.

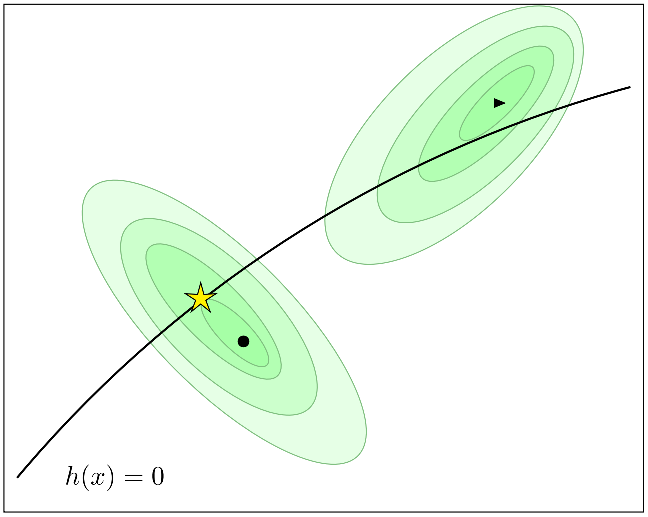

Figure 2. A 2-d optimization problem with a multimodal fitness function and one equality constraint (thick black line). Feasible solutions are restricted to this line. The shaded (green) contours depict the fitness function f (darker = smaller). The black points show the unconstrained optima of the fitness function which are different from the optimal solution of the constrained problem shown as the gold star. If we select for a category-(b)-type transformation the area below the constraint line as the feasible region, the optimal choice is the black dot and if we select the upper side of the equality constraint, the optimal solution is the black triangle. In both cases there is no chance to converge to the correct solution on the equality constraint line (gold star).

The first category does not enlarge the feasible volume. Therefore, the problem remains as challenging as before. This category is used in [1] for several constrained problems with inequalities and equalities. The approach fails on problems with equality constraints.

Category (b) is used by Regis [2] and in our previous work [3], [4], [5]. It chooses manually one side of the equality constraint(s) as the feasible region. This approach may work for simple problems (see Fig. 1) but it is problematic for two reasons: For each new problem the user has to find manually the correct transformations which can be difficult in the case of multiple equality constraints. Even worse, it is bound to fail in cases where the fitness function has several local optima on both sides of the equality constraint(s), as Fig. 2 shows for a concrete 2-d example.

The third category (c) is widely used in different studies and embedded in various algorithms e.g. [6], [7]. In this approach, a tube around each equality constraint is considered as the feasible space. The size of this tube is controlled by a (usually very small) parameter μ_0. However, a very small μ_0 makes it difficult to find feasible solutions at all and a larger choice of μ_0 makes it likely to return solutions which violate the equality constraints by a large amount.

Therefore, a fourth category (d) is suggested in different studies [8], [9] which recommends to start with a large tube-shaped margin around each equality constraint which gradually shrinks and converges to the actual equality constraint. The adaptive relaxing rule proposed in [8] introduces six new parameters to control the changing scheme of the margin. Later, Zhang [9] proposed a parameter-free adaptation rule for equality handling which decays the margin proportional to the fraction of feasible solutions in every iteration. Although the mentioned study reports good results for solving well-studied G-problems, the success is only achieved after many function evaluations (thousands of evaluations).

Category (e) does not modify the equality constraints, but it deduces repair mutations from them. As an example, RGA proposed by Chootinan [10] uses a gradient-based repair method to adapt a genetic algorithm (GA) for constrained problems. Also, Beyer and Finck [11] propose a CMA-ES algorithm which uses a problem-specific repair mutation to assure that the solutions always fulfill the equality constraint(s). This works well for equality constraints where one variable can be expressed as a function of the others.

In this work, we propose for the first time an equality handling (EH) scheme coupled with surrogate-assisted optimization techniques. We use several well-known benchmark problems (G-problem suite) to assess our algorithm in the context of the following research questions:

- **(H1)** Can we advice an efficient surrogate-assisted technique which can address problems with equality constraints?
- **(H2)** Does our proposed equality handling technique benefit from the dynamic shrinking feasible region?

In Section 2.1, the general idea of the efficient surrogate-assisted optimizer SACOBRA [5] is briefly described. In Section 2.2, the proposed algorithm for handling equality constraints is described in detail. We describe in Section 3 the experimental setup and the test functions. In the same section we report the results achieved by our proposed algorithm. In Section 4, the results are discussed and compared with the state of the art. Finally, Section 5 contains our conclusions and our answers to the research questions.

2. Methods

2.1. COBRA and SACOBRA

Regis proposed with COBRA [2] an algorithm which can solve high-dimensional constrained optimization problems very efficiently, i. e. with 100 – 2000 evaluations depending on the dimensionality of the problem. The main idea of COBRA is to use radial basis functions (RBF) as cheap surrogate models for objective and constraint functions. This greatly reduces the number of expensive real function evaluations. Although COBRA performs efficiently, it suffers from having numerous parameters which require to be tuned to the problem at hand. In our recent work [3], [4], [5], [12] we proposed several extensions to COBRA which culminated in our algorithm SACOBRA [5], [12]. SACOBRA performs on-line adaptations to the problem at hand and eliminates the need to tune any parameters.

SACOBRA performs well on several black-box constraint optimization problems. For details the reader is referred to [12].[1]

In the n-th iteration COBRA and SACOBRA solve the following internal optimization problem:

$$\begin{aligned}
&\text{Minimize} && s_0^{(n)}(\vec{x}), && \vec{x} \in [\vec{a}, \vec{b}] \subset \mathbb{R}^d && (3)\\
&\text{subject to} && s_i^{(n)}(\vec{x}) + \epsilon^{(n)} \leq 0, && i = 1, 2, \ldots, m && (4)\\
& && \rho^{(n)} - ||\vec{x} - \vec{x}_k|| \leq 0, && k = 1, 2, \ldots, n && (5)
\end{aligned}$$

where $s_0^{(n)}$ is the fitness function surrogate model based on n points and $s_i^{(n)}$ stands for the model of the i-th inequality constraint in the n-th iteration. $\epsilon^{(n)}$ and $\rho^{(n)}$ are internal variables of the COBRA algorithm. For details the reader is referred to [2]. In a nutshell, Eq. (5) forces the internal optimizer to stay a distance $\rho^{(n)}$ away from all previous solutions. The idea is to avoid too frequent similar solutions. $\rho^{(n)}$ is selected cyclically from a distance requirement cycle (DRC) $\Xi = \{\xi^{(1)}, \ldots, \xi^{(p)}\}$ chosen by the user.

A drawback of COBRA and SACOBRA is that they can only handle inequality constraints. If problems with equality constraints are to be addressed, the user has to replace each equality constraint by the appropriate inequality constraint (category (b) in Sec. 1).

2.2. Proposed Equality Constraint Handling Approach

In this study, SACOBRA is extended to handle equality constraints. The zero-volume feasible space attributed to the j-th equality constraint $h_j(\vec{x}) = 0$ is expanded to a tube-shaped region around it: $|h_j(\vec{x})| - \mu^{(n)} \leq 0$. By gradually reducing the margin $\mu^{(n)}$ the solutions are smoothly guided towards the real feasible area. It is difficult to model with RBFs accurately the triangular shaped $|\cdot|$-function. Therefore we translate every equality constraint to two inequality constraints as follows:

$$\begin{cases} h_j(\vec{x}) - \mu^{(n)} \leq 0, & j = 1, 2, \ldots, r \\ -h_j(\vec{x}) - \mu^{(n)} \leq 0, & j = 1, 2, \ldots, r \end{cases}$$

To find an appropriate initial value for the margin $\mu^{(n)}$ we use the following procedure: We calculate for each initial design point the sum of its constraint violations. $\mu^{(n)}$ is set to be the median of these constraint violations. In this way we can expect to start with an initial design population containing roughly 50% artificially feasible and 50% infeasible points.

Algorithm 1 shows our complete algorithm in pseudo code. In every iteration RBFs are trained to model the objective function f and all constraint functions g and h. Finding the next iterate is done by solving an internal optimization problem which minimizes the objective function (3) and tries to fulfill the constraints (4) – (5). Additionally, the expanded equality constraints, Eqs. (6) – (7), should be satisfied:

1. SACOBRA is available as open-source R-package from CRAN via this URL: https://cran.r-project.org/web/packages/SACOBRA.

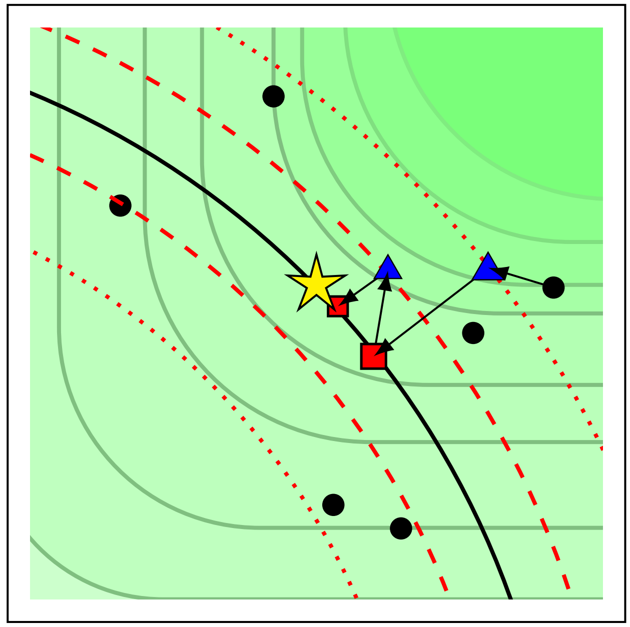

Figure 3. Refine Step. The shaded (green) contours, golden star and solid (black) line have the same meaning as in Fig. 1 and 2. In iteration n the dotted lines mark the current feasible tube. The optimization step will result in a point on the tube margin (rightmost blue triangle). The refine step moves this point to the closest point on the equality line (lower red square). In iteration $n+1$ the tube shrinks to the feasible region marked by the dashed lines. Now the optimization step will result in a point on the dashed line (leftmost blue triangle), and so on. If the refine steps were missing, we would lose the best feasible point when shrinking the margin.

$$s_{m+j}^{(n)}(\vec{x}) - \mu^{(n)} \leq 0, \quad j = 1, 2, \ldots, r \qquad (6)$$

$$-s_{m+j}^{(n)}(\vec{x}) - \mu^{(n)} \leq 0, \quad j = 1, 2, \ldots, r. \qquad (7)$$

Here, $s_{m+j}^{(n)}$ is the n-point surrogate model for the j-th equality constraint h_j. Before conducting the expensive real function evaluation of the new iterate $\vec{x}^{(n+1)}$, we try to refine the suggested solution: We eliminate the infeasibility caused by the current margin $\mu^{(n)}$ by moving $\vec{x}^{(n+1)}$ towards the equality line.[2] This **refine step** is done by minimizing with a conjugate-gradient (CG) method the squared sum of all equality constraint violations:

$$\text{Minimize} \sum_{j=1}^{r} (s_{m+j}^{(n)}(\vec{x}))^2, \quad \vec{x} \in [\vec{a}, \vec{b}] \subset \mathbb{R}^d \qquad (8)$$

The refine step moves the returned solution towards the equality constraints. This is done to prevent losing it in the next iteration when the equality margin μ is reduced (Fig. 3). The refined point is evaluated and the so-far best solution is updated. The best solution is the one with the best fitness value which satisfies the inequality constraints and lies in the

2. More precisely: towards the intersection of the constraint model hypersurfaces $s_{m+j}^{(n)}$ in the case of multiple equality constraints.

Algorithm 1 Constrained optimization with equality handling (EH). Parameters: μ_{final}, β.

```
1: Choose initial population P by drawing n = 3d points randomly from the search space, evaluate them on the real functions and select best point x⃗^(b)
2: Initialize EH margin μ^(n)
3: Adapt SACOBRA parameters according to P
4: while n < budget do
5:     Build surrogates s_0^(n), s_1^(n), ..., s_{m+r}^(n) for f, g_1, ..., g_m, h_1, ..., h_r
6:     Perform SACOBRA optimization step: Minimize Eq. (3) subject to Eqs. (4) – (7), starting from the current best solution x⃗^(b). Result: x⃗^(n+1)
7:     x⃗^(n+1) ← REFINE(x⃗^(n+1))
8:     Evaluate x⃗^(n+1) on real functions
9:     P ← P ∪ {x⃗^(n+1)}
10:    x⃗^(b) ← Select the best solution so far from P
11:    μ^(n+1) ← max {μ^(n) β, μ_final}
12:    n ← n + 1
13: end while
14: return (x⃗^(b), the best solution)

15: function REFINE(x⃗_new)
16:    Starting from x⃗_new, minimize Eq. (8), the squared sum of all equality constraint violations.
17:    return (x⃗_r, the minimization result)
18: end function
```

intersection of the tube-shaped margins around the equality constraints.

Our proposed method benefits from having both feasible and infeasible solutions in the population, by gradually reducing the equality margin μ. The equality margin μ can be reduced in different fashions. Zhang [9] proposes an adaptive scheme: in every iteration the equality margin is multiplied by a factor $\beta_Z \in [0, 1]$ which is proportional to the ratio of current infeasible solutions P_{inf} within the set of all solutions P:

$$\mu^{(n+1)} = \mu^{(n)} \cdot \beta_Z = \mu^{(n)} \cdot \frac{|P_{inf}|}{|P|} \tag{9}$$

That is, if there are no feasible solutions, no reduction of the margin takes place. On the other hand, if 50% of the population is feasible, the margin is halved in every iteration, i. e. a very rapid decrease. This scheme may work well for algorithms having a population of solutions and infrequent updates at the end of each generation as in [9]. But for our algorithm with only one new solution in each iteration this scheme may decay too rapidly. Therefore, we use an exponential decay scheme shown in Step 11 of Algorithm 1

$$\mu^{(n+1)} = \mu^{(n)} \beta \tag{10}$$

with decay factor $\beta > 0.5$. The decay factor β is constant for all problems, independent of the problem dimension d.

Table 1. CHARACTERISTICS OF G FUNCTIONS WITH EQUALITY CONSTRAINTS: d: DIMENSION, LI: THE NUMBER OF LINEAR INEQUALITIES, NI: NUMBER OF NONLINEAR INEQUALITIES, LE: THE NUMBER OF LINEAR EQUALITIES, NE: THE NUMBER OF NONLINEAR EQUALITIES, α: NUMBER OF ACTIVE CONSTRAINTS.

	Fct.	d	type	LI	NI	LE	NE	α
training	G03	20	nonlinear	0	0	0	1	1
	G05	4	nonlinear	2	0	0	3	3
	G11	2	nonlinear	0	0	0	1	1
	G13	5	quadratic	0	0	0	3	3
test	G14	10	nonlinear	0	0	3	0	3
	G15	3	quadratic	0	0	1	1	2
	G17	6	nonlinear	0	0	0	4	4
	G21	7	nonlinear	0	1	0	5	6

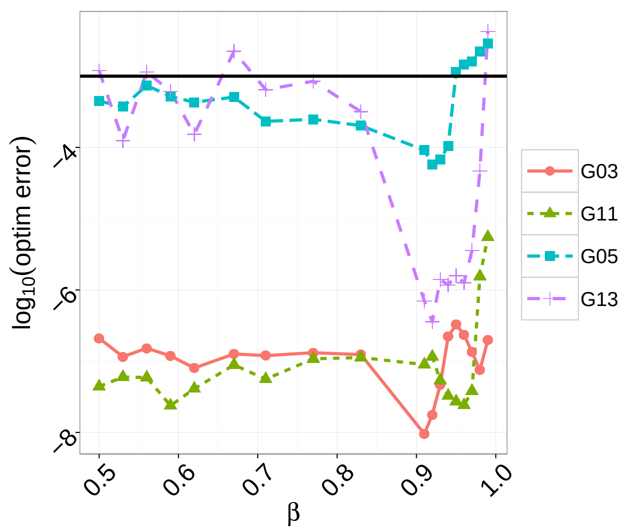

Figure 4. Impact of varying parameter β (Eq. (10)). Shown is the median of the final optimization error for all training G-problems. The black horizontal line depicts the threshold 10^{-3}. For $\beta \in [0.90, 0.94]$ all problems have a significantly smaller error than this threshold.

3. Experiments

3.1. Benchmark Functions and Experimental Setup

The G-problem suite, described in [13], is a set of well-known constrained optimization problems. In this study, 8 G-problems with equality constraints are considered. The characteristics of these G-problems are listed in Table 1. These problems have very diverse characteristics. Therefore, they provide a suitable testbed for constrained optimization algorithms.

Just the first four of the eight G-problems (named ‚training‘ in Table 1) were used during EH algorithmic development and for tuning the only free parameters β, the decay factor, and μ_{final}. Only after fixing all algorithmic details and parameters, SACOBRA-EH was run on the other four G-problems (named ‚test‘ in Table 1) and the results were taken ’as-is’. This provides a first indication of the algorithm’s performance on unseen problems.

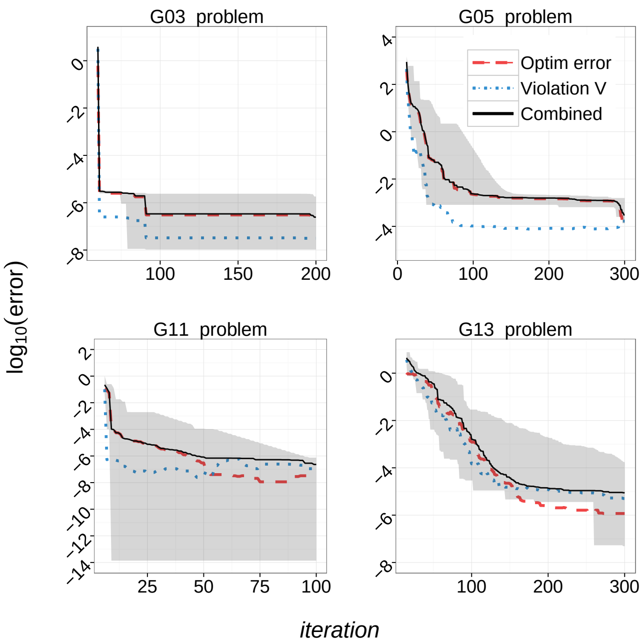

Figure 5. Optimization progress for G03, G05, G11, and G13. The dashed (red) curve is the absolute optimization error $|f(\vec{x}^{(b)}) - f(\vec{x}^{(opt)})|$ in every iteration. The dotted (blue) curve is the maximum constraint violation V of the the so-far best solution. The solid (black) line is the combined sum $|f(\vec{x}^{(b)}) - f(\vec{x}^{(opt)})| + V$ of absolute optimization error and maximum constraint violation V. Each of the three curves shows the median value from 30 independent runs. The gray bands around the black curves show the worst and the best runs for *Combined*.

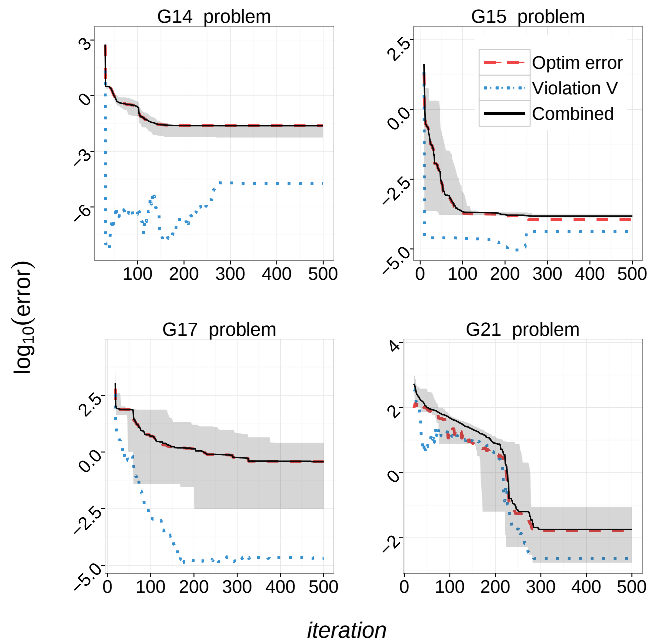

Figure 6. Same as Fig. 5 for problems G14, G15, G17, and G21 (‚test' problems).

Fig. 4 shows initial runs on the training problems to find a good choice for the decay factor β. Two of the training G-problems (G03 and G11) show good performance for all values of β. The other two problems (G05 and G13) show good performance for $\beta \in [0.90, 0.94]$. Larger values result in slower convergence, they would converge if the number of iterations were increased, but we allow here only a maximum of 300 iterations. For all subsequent results we fix the decay factor to $\beta = 0.94$.

We run the optimization process for every problem with 30 independent randomly initialized populations (using LHS) to have statistically sound results.

The equality margin $\mu^{(n)}$ has the lower bound $\mu_{final} = 10^{-6}$. The DRC parameter is set to $\Xi = \{0.3, 0.0\}$.

3.2. Results

Visualizing the optimization process for optimization problems with equality constraints is not always straightforward because early or intermediate solutions are usually never strictly feasible. If we artificially increase the feasible volume with the help of a (shrinking) margin μ, two things will happen: (a) There can be intermediate solutions which are apparently better than the optimum.[3] To avoid ‚negative' errors we use the absolute value

$$E_{abs} = |f(\vec{x}^{(b)}) - f(\vec{x}^{(opt)})| \tag{11}$$

as a measure to evaluate our algorithm where $\vec{x}^{(opt)}$ is the location of the true optimum in input space. (b) Secondly, when the margin μ shrinks, former feasible solutions become infeasible and new feasible solutions often have larger optimization errors. To make the former and the new solutions comparable we form the sum

$$E_{combined} = |f(\vec{x}^{(b)}) - f(\vec{x}^{(opt)})| + V \tag{12}$$

where V is the maximum violation. This sum is shown as *Combined* curve in the plots of Fig. 5 and 6.

Fig. 5 shows the optimization process for G03, G05, G11 and G13. It is clearly seen that the median of the optimization error as well as the median of the maximum violation reaches very small values for all four problems within less than 300 function evaluations. The final maximum violation is less than 10^{-4} for these problems, i. e. the infeasibility level is negligible. The optimization error converges to 10^{-3} for G05 and to 10^{-6} or smaller for the three other problems. This means that the algorithm is efficient in locating solutions very close to the true optimum after only a few function evaluations.

Fig. 6 shows the optimization process on our four ‚test' G-problems (G14, G15, G17, and G21) which were not taken into account during algorithmic development and tuning. We choose a maximum budget of 500 function evaluations. The decreasing trend for the *Optim error* and *Combined* curves is clearly seen in Fig. 6 for all four

3. They are on the side of the tube where f is lower than the constrained optimum.

problems. G14 is the only problem with an increasing maximum violation over iterations. In Fig. 6, left-top, the solutions in early iterations are often almost feasible (max. violation $=10^{-8}$, due to the refine step) but they have large objective values. The maximum violation increases in later iterations. At the same time the objective value is reduced by a factor of more than 100 and the final solution has a reasonable low objective value and also a very small level of infeasibility. Our algorithm can find almost feasible solutions (max. violation $< 10^{-4}$) for all four ‚test' G-problems with reasonably small optimization error $< 10^{-2}$ except for G17 where the error is 0.26. Note that G17 has a quite high optimum value and we can infer from Table 2 that G17 is solved in 50% of the cases with a *relative* error better than $3 \cdot 10^{-5}$. It has to be remarked that, however, the worst-case error for the two ‚test' problems G17 and G21 is a little bit worse than for the four training problems. This has to be expected since the parameters were not explicitly tuned to the ‚test' problems.

In Table 2 we compare our algorithm with other state-of-the-art constrained optimization solvers. We show the results from the efficient differential evolutionary algorithm by Zhang [9], an evolutionary algorithm with a novel selection scheme by Jiao [14], the repair genetic algorithm (RGA) by Chootinan [10] and the improved stochastic ranking evolutionary strategy (ISRES) by Runarsson [6]. The results shown in column 2–5 of Table 2 are taken from the original papers. We present in the last column results for differential evolution (DE) [16] with automatic parameter adjustment as proposed by Brest [15] This was done by running own experiments using the DEoptimR package in R [17]. DE does not perform well on G03 and G13. It has to be noted that 25 of the 30 runs for G13 did not terminate by reaching a tolerance threshold but reached the maximum of allowed iterations. Note that DE has on G17 after 37 500 function evaluations an error larger than ours after 500 function evaluations.

Zhang [9] uses an adaptive equality margin to tackle equality constraints. RGA [10] applies a gradient based repair technique to handle equality and inequality constraints. All other methods make use of a constant and small equality margin μ_0.

Our proposed algorithm can achieve results with better (G03) or similar accuracy as compared to the evolutionary algorithms (ES, GA, DE) with a significantly lower number of function evaluations.

4. Discussion

Although the refine step is an algorithmically very simple step, it is essential for our EH algorithm. This is because only one new point will be added to the population in each iteration. Usually this point will sit at the border of the artificial feasible region after the optimization step. If we now shrink the artificial feasible region without a refine step, we would lose this point and jump to another feasible point, if any, probably with a much larger objective value. The refine step has two advantages: It often produces a solution

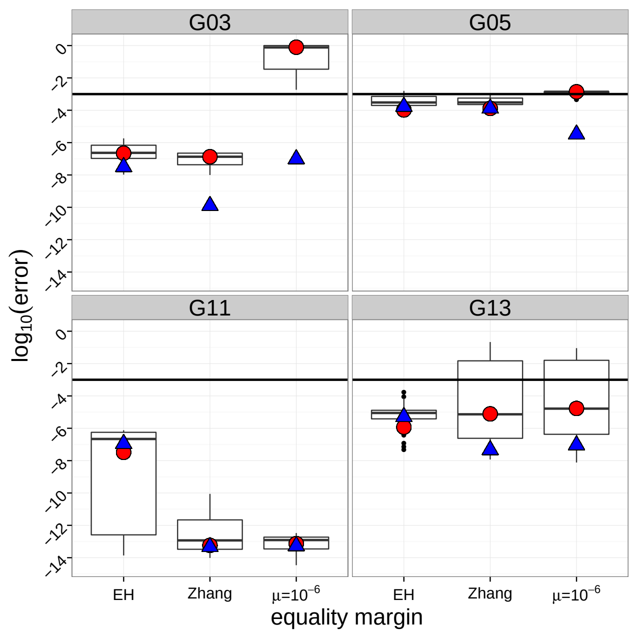

Figure 7. Impact of different update schemes for the equality margin μ.The circle points (red) are the median of E_{abs} and the triangle points (blue) are the median of the final maximum constraint violations V for 30 independent runs. The box-plots depict the combined sum $E_{combined}$ acc. to Eq. (12) for different margin update schemes: (I) our algorithm (EH), (II) Zhang update scheme described in Eq. (9), and (III) constant margin $\mu = 10^{-6}$. In case of G13, both other cases (II) and (III) have more than a quarter of runs worse than the threshold 10^{-3} (black horizontal line).

which fulfills all equality constraints to machine accuracy. Even more important, it brings the new point into the inner part of the artificial feasible region, so that it will remain feasible after shrinking has taken place.

The results in Sec. 3 show that our extended algorithm SACOBRA+EH with a dynamic equality margin (category (d)) can provide reasonable solutions for 8 of the G-problems, while most of these problems were not solvable with the older version of SACOBRA which used the equality-to-inequality transformation scheme (category (b)). Additionally, SACOBRA+EH boosts up the results for the G03 problem. As the worst-case behavior in Fig. 5 shows, the current algorithm can solve 100% of the runs for the G03 problem within only 100 iterations. The older SACOBRA algorithm required 3 times more function evaluations and could solve only 77% of the runs [3], [12]. The G03 problem benefits from the gradually shrinking tube-shaped margin of SACOBRA+EH in two ways: (i) Infeasible points with good fitness values are included in the population since they are the best solutions in early iterations. This helps to improve the surrogate model of the fitness function in the interesting region. (ii) The feasible region is a tube around the equality hypersurface. This is in contrast to the former category-(b)-type solution where the considerably larger region on one side of the hypersurface was feasible. The tube-shaped region helps the optimizer to concentrate on the interesting

Table 2. Different optimizers: median (m) of best feasible results and (fe) average number of function evaluations. Results from 30 independent runs with different random number seeds. Numbers in **boldface (blue)**: distance to the optimum ≤ 0.001. Numbers in ***italic (red)***: reportedly better than the true optimum.

Fct.	Optimum		SACOBRA+EH [this work]	Zhang [9]	ISRES [6]	RGA 10% [10]	Jiao [14]	DE [15]
G03	-1.0	m	**-1.0**	***-1.0005***	***-1.001***	-0.9999	***-1.0005***	-0.8414
		fe	100	25493	349200	399804	19534	13325
G05	5126.497	m	**5126.497**	**5126.497**	**5126.497**	5126.498	**5126.497**	5126.498
		fe	300	21363	195600	39459	2050	8108
G11	0.750	m	**0.750**	***0.749***	**0.750**	**0.750**	***0.749***	**0.750**
		fe	100	6609	137200	7215	135	2099
G13	0.0539	m	**0.0539**	**0.0539**	**0.0539**	–	**0.0539**	0.068
		fe	300	19180	223600	–	3103	23637
G14	-47.765	m	-47.759	**-47.765**	–	–	**-47.765**	-47.761
		fe	500	34825	–	–	6093	72015
G15	961.715	m	**961.715**	**961.715**	–	–	**961.715**	**961.715**
		fe	500	11706	–	–	757	5666
G17	8853.534	m	8853.794	8868.539	–	–	**8853.534**	8867.606
		fe	500	43369	–	–	3203	37532
G21	193.724	m	193.725	193.735	–	–	**193.724**	193.790
		fe	500	23631	–	–	46722	35559
		average fe	350	23272	226400	148826	10200	24742

area, regardless of the initial population.

In order to study the impact of the update scheme for the equality margin μ, we have embedded for comparison the Zhang update scheme, Eq. (9), and a constant scheme with a small equality margin $\mu = 10^{-6}$ in our algorithm. In Fig. 7 the final optimization results achieved by the different schemes are compared.

Fig. 7 shows that the adaptive update scheme proposed by Zhang [9] appears to have a similar behavior as the constant scheme on 3 G-problems. This is because the Zhang decay factor in Eq. (9) usually results in a fast decay ($\beta \approx 0.5$), i. e. the margin μ gets small in early iterations. Therefore, infeasible solutions with good objective values have a smaller chance to be found. This may result in less accurate surrogate models, especially for problems with a nonlinear objective function and several local optima. G13 is such an example with several local optima. Here, Zhang's fast shrinking equality margin or a constant small margin are problematic and more than 25% of the runs would not converge to the optimum (Fig. 7).

Table 2 shows that sometimes the other algorithms find solutions which are slightly infeasible and have a better objective value than the real optimum. The level of permitted infeasibility is controlled by margin parameters for the algorithms ISRES, Jiao and Zhang. Therefore, the quality of the results is dependent on the final value μ_{final} of the equality margin. In contrast to that, our algorithm is – due to the refine step – much less sensitive to the parameter μ_{final}.

5. Conclusion

We developed an equality constraint handling technique coupled with a surrogate-assisted optimization algorithm [12] which benefits from a gradual shrinking of the expanded feasible region. The gradual shrinking smoothly guides the infeasible solutions toward the feasible hypersurfaces. Our algorithm requires **less than 500 function evaluations** to provide high-quality optimization results on both sets of training and test G-problems with **equality constraints**. The state-of-the-art solutions reach similar accuracy, but they require on average 250 times more function evaluations. Therefore, our answer to the first posed research question **(H1)** is positive.

Our results have shown that a gradually shrinking equality margin μ is essential for an effective algorithm working on a diverse set of problems. Although a small constant equality margin may work well for very simple problems, we found that for other cases, where the objective function is nonlinear or multimodal, a constant equality margin often causes early stagnation of the optimization process. On the other hand, if we use a dynamic shrinking feasible region (considered in our research question **(H2)**), we observe a much more robust behavior on a variety of challenging equality constraint problems.

Acknowledgments

This work has been supported by the Bundesministerium für Wirtschaft (BMWi) under the ZIM grant MONREP (AiF FKZ KF3145102, Zentrales Innovationsprogramm Mittelstand).

Supported by:

Federal Ministry for Economic Affairs and Energy

on the basis of a decision by the German Bundestag

References

[1] K. S. Bhattacharjee and T. Ray, "A novel constraint handling strategy for expensive optimization problems," *World Congress on Structural and Multidisciplinary Optimization*, 2015.

[2] R. G. Regis, "Constrained optimization by radial basis function interpolation for high-dimensional expensive black-box problems with infeasible initial points," *Engineering Optimization*, vol. 46, no. 2, pp. 218–243, 2014.

[3] P. Koch, S. Bagheri, W. Konen, C. Foussette, P. Krause, and T. Bäck, "Constrained optimization with a limited number of function evaluations," in *Proc. 24. Workshop Computational Intelligence*, F. Hoffmann and E. Hüllermeier, Eds. Universitätsverlag Karlsruhe, 2014, pp. 119–134.

[4] ——, "A new repair method for constrained optimization," in *Proceedings of the 2015 on Genetic and Evolutionary Computation Conference (GECCO)*. ACM, 2015, pp. 273–280.

[5] S. Bagheri, W. Konen, C. Foussette, P. Krause, T. Bäck, and P. Koch, "SACOBRA: Self-adjusting constrained black-box optimization with RBF," in *Proc. 25. Workshop Computational Intelligence*, F. Hoffmann and E. Hüllermeier, Eds. Universitätsverlag Karlsruhe, 2015, pp. 87–96.

[6] T. P. Runarsson and X. Yao, "Search biases in constrained evolutionary optimization," *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, vol. 35, no. 2, pp. 233–243, 2005.

[7] ——, "Stochastic ranking for constrained evolutionary optimization," *IEEE Transactions on Evolutionary Computation*, vol. 4, no. 3, pp. 284–294, 2000.

[8] X.-F. Xie, W.-J. Zhang, and D.-C. Bi, "Handling equality constraints by adaptive relaxing rule for swarm algorithms," *Congress on Evolutionary Computation (CEC)*, pp. 2012–2016, 2004.

[9] H. Zhang and G. Rangaiah, "An efficient constraint handling method with integrated differential evolution for numerical and engineering optimization," *Computers & Chemical Engineering*, vol. 37, pp. 74–88, 2012.

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

[11] H.-G. Beyer and S. Finck, "On the design of constraint covariance matrix self-adaptation evolution strategies including a cardinality constraint," *Trans. Evol. Comp*, vol. 16, no. 4, pp. 578–596, Aug. 2012.

[12] S. Bagheri, W. Konen, M. Emmerich, and T. Bäck, "Solving the G-problems in less than 500 iterations: Improved efficient constrained optimization by surrogate modeling and adaptive parameter control," *Applied Soft Computing*, 2016, submitted.

[13] J. J. Liang, T. P. Runarsson *et al.*, "Problem definitions and evaluation criteria for the CEC 2006 special session on constrained real-parameter optimization," Nanyang Technological University, Singapore, Tech. Rep., May 2006.

[14] L. Jiao, L. Li, R. Shang, F. Liu, and R. Stolkin, "A novel selection evolutionary strategy for constrained optimization," *Information Sciences*, vol. 239, pp. 122 – 141, 2013.

[15] J. Brest, S. Greiner, B. Boskovic, M. Mernik, and V. Zumer, "Self-adapting control parameters in differential evolution: A comparative study on numerical benchmark problems," *Trans. Evol. Comp*, vol. 10, no. 6, pp. 646–657, Dec. 2006.

[16] R. Storn and K. Price, "Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces," *Journal of global optimization*, vol. 11, no. 4, pp. 341–359, 1997.

[17] R Core Team, *R: A Language and Environment for Statistical Computing*, R Foundation for Statistical Computing, Vienna, Austria, 2014. [Online]. Available: http://www.R-project.org