Self-adjusting surrogate-assisted optimization techniques for expensive constrained black box problems

Bagheri, S.

Chapter 4

Handling Equality Constraints in SACOBRA

4.1 Outline

SACOBRA optimizer as introduced in Ch. 3 has the drawback of being only able to handle COPs with inequality constraints. However, real-world COPs are often subject to *equality* and inequality constraints.

In this chapter SACOBRA is enhanced to handle equality constraints. The analysis regarding SACOBRA's modification for COPs with equality constraints are taken from [16] and [18].

The rest of this chapter is organized as follows: After introducing the equality handling problem and its challenges in Sec. 4.2, we give an overview about the already existing techniques for handling black-box equality constraints and we categorize them into 5 different types, in Sec. 4.3. In the same section we describe a common dilemma of many equality handling approaches. In Sec. 4.4 we describe the equality handling approach embedded in SACOBRA. Sec. 4.5 reports the details of the experimental setup used in this chapter. SACOBRA with the proposed equality handling technique (SACOBRA+EH) is applied on a subset of G-problems with equality constraints. In Sec. 4.6, we report and analyze the results achieved by SACOBRA and we compare our results with several other constrained optimizers. This chapter is concluded and summarized in Sec. 4.7.

4.2 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) $h_1, \ldots, h_r$:

$$\begin{array}{rll} \text{Minimize} & f(\vec{x}), & \vec{x} \in [\vec{l}, \vec{u}] \subset \mathbb{R}^d \\ \text{subject to} & g_j(\vec{x}) \leq 0, & j = 1, 2, \ldots, m \\ & h_k(\vec{x}) = 0, & k = 1, 2, \ldots, r \end{array} \tag{4.1}$$

where $\vec{l}$ and $\vec{u}$ 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.

Handling equality constraints in black-box COPs is a challenging task. The presence of equality constraints causes the feasibility ratio ρ to be exactly zero[1]. Also, it is really demanding and sometimes impossible to bring such a problem in the *feasible subspace* formed by equality constraints; therefore, it is highly unlikely for most of the numerical optimizers to find a fully feasible solution.

Throughout this chapter the following research questions are addressed:

Q4.1 How can SACOBRA be extended to handle COPs with equality constraints *efficiently* and solve the common dilemma of margin-based equality handling methods?
After embedding an equality handling extension in SACOBRA framework, we evaluate them on a set of benchmarks with equality constraints from the G-problem function suite. In this chapter it will be answered if SACOBRA with equality handling (SACOBRA+EH) can be advised as an efficient surrogate-assisted technique for handling COPs with equality constraints. The equality handling approach in SACOBRA is a margin-based approach. After giving a description about the different equality handling approaches in Sec. 4.3, we describe the common dilemma of margin-based equality handling. Then we will show how to handle this dilemma.

Q4.2 Is a gradually shrinking feasibility margin an important ingredient for SACOBRA to produce high-quality results on COPs with equality constraints?
The equality handling approach designed for SACOBRA starts with a large artificially feasible area and the feasibility margin gradually shrinks to a narrow belt around the feasible subspace. In this chapter we investigate the effectiveness of this shrinking margin for SACOBRA.

[1] ρ : ratio between feasible volume and search space volume, described in Ch. 3.6.1

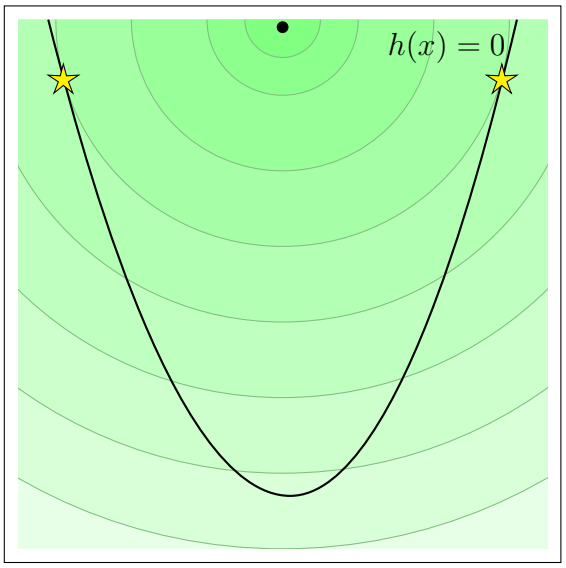

Figure 4.1: A simple $2D$ optimization problem with one equality constraint. The shaded (green) contours depict the fitness function f (darker = smaller). The black curve shows the equality constraint. Feasible solutions are restricted to this line. The black point shows the global optimum of the fitness 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.

4.3 Taxonomy of Equality Handling Techniques

We categorize the existing numerical constraint handling techniques into 5 categories (a)–(e). The last category assumes that constraint functions are not fully black-box. In general, most of the existing constraint handling algorithms for black-box COPs need to modify the equality constraints in order to be able to address them. The different modifications used for equality constraints can be divided into five categories of transforming equality constraints $h_k(\vec{x}) = 0$ to:

(a) a pair of inequality constraints

$$\begin{cases} h_k(\vec{x}) & \geq 0, \\ h_k(\vec{x}) & < 0, \end{cases} \tag{4.2}$$

(b) one inequality constraint (manually chosen),

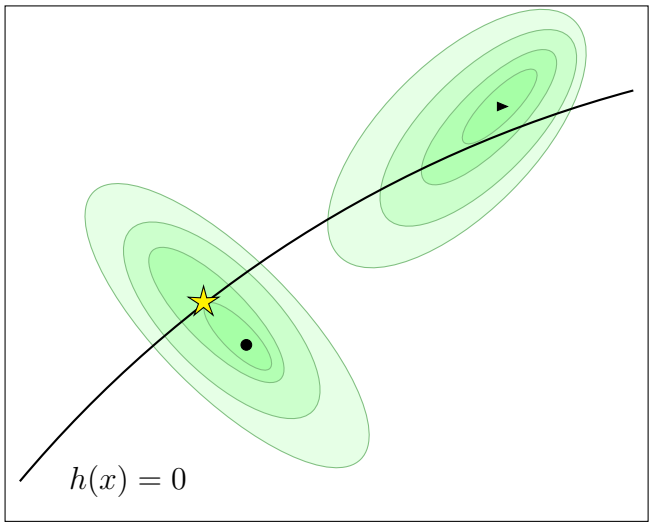

Figure 4.2: A $2D$ optimization problem with a multimodal fitness function and one equality constraint (thick black line). Feasible solutions are restricted to this line. The shaded (green) contours depict the fitness function f (darker = smaller). The black points show the unconstrained optima of the fitness function which are different from the optimal solution of the constrained problem shown as the gold star. If we select for a category-(a)-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).

(c) a tube-shaped region $|h_k(\vec{x})| - \epsilon_0 \leq 0$ in order to expand the feasible space, where ϵ_0 is a small positive constant,

(d) a shrinking tube-shaped region $|h_k(\vec{x})| - \epsilon^{(n)} \leq 0$, where $\epsilon^{(n)}$ is decaying iteratively,

(e) a repair mutation.

The first category does not enlarge the feasible volume. Therefore, the problem of zero feasibility rate remains as challenging as before. This category is used in [26] for several constrained problems with inequalities and equalities. The approach fails on problems with equality constraints.

Category (b) is used by Regis [141] and SACOBRA [97, 98, 20]. It chooses manually one side of the equality constraint(s) as the feasible region. This approach may work for simple problems (see Fig. 4.1) 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. 4.2 shows for a concrete $2D$ example.

The third category (c) is widely used in different studies and embedded in various algorithms e.g. [151, 150, 45]. 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 ϵ_0. However, a very small ϵ_0 makes it difficult to find feasible solutions at all and a larger choice of ϵ_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 [183, 187, 31] which recommends to start with a large tube-shaped margin around each equality constraint that gradually shrinks to a small final feasibility margin ϵ_f. The adaptive relaxing rule proposed in [183] introduces six new parameters to control the changing scheme of the margin. Later, Zhang [187] 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 the well-studied G-problems, the success is only achieved after many function evaluations (thousands of evaluations).

Choosing a suitable feasibility margin is not a trivial task and it is a problem dependent task. In addition, once there is a feasibility margin greater than zero, we have the following *dilemma*: Each COP solver has a whole set of possible solutions to choose from. Should we prefer a solution with better objective value but larger violation of the true equality constraint over another one with worse objective value but smaller violation? There is no clear answer to this, it is a multi-criteria problem. Nonetheless, most of the numerical constrained optimizers are designed to search for the solution with the best objective value $\vec{x}^*_{af}$ inside the artificially feasible region and they avoid approaching the true optimum $\vec{x}^*$ (see Fig. 4.3). The distance between these two solutions $\vec{x}^*_{af}$ and $\vec{x}^*$ can be pretty large, both in input space and objective space.[2] Of course the dilemma would disappear if ϵ_f approaches zero, but many optimizer have problems with too small margins (they may not find feasible solutions).

Not all the equality handling approaches used by numerical optimizers modify the equality constraints. There is a fifth equality handling category (e) that assumes only the objective function is black-box and the equality constraints are known explicitly. As an example, RGA proposed by Chootinan [38] uses a gradient-based repair method to adapt a genetic algorithm (GA) for constrained problems. Also,

[2]The distances depend on ϵ_f and the steepness of the equality functions $h_j(\vec{x})$ and the objective function $f(\vec{x})$.

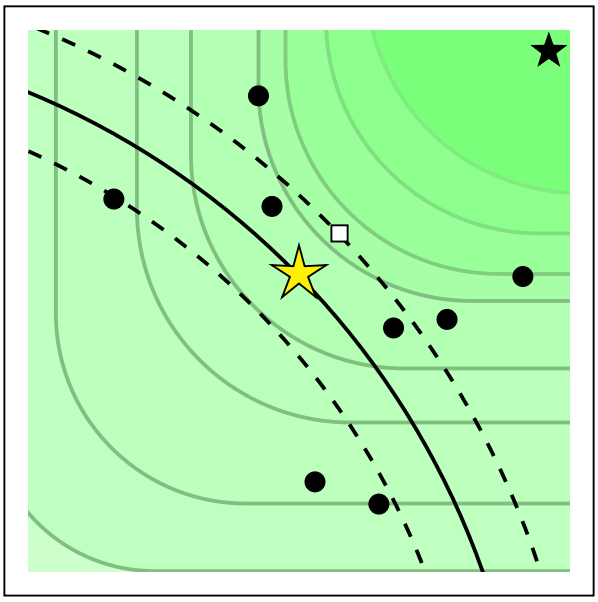

Figure 4.3: The shaded contours depict the objective function f (darker = smaller). The black curve shows the equality constraint. Feasible solutions are restricted to this line. The black star shows the global optimum of the objective function which is different from the optimum of the constrained problem shown as the yellow star $\vec{x}^*$. The area enclosed by the dashed curves is the artificially feasible area, given a fixed feasibility margin ϵ_0. The white square marks the optimal solution $\vec{x}^*_{af}$ in the artificial feasible region. Black dots mark some infill points.

Beyer and Finck [25] propose a CMA-ES algorithm which uses a problem-specific repair mutation to assure that the solutions always fulfill the equality constraint(s). [5] and [166] use different types of evolutionary strategies combined with a repairing mechanism which projects the solution(s) to the feasible subspace of the problem. The latter method is only applicable on linearly constrained problems. Since this category of algorithms are not suitable for fully black-box problems, they might have severe limitations in dealing with real-world COPs.

Here, we focus on solving fully black-box COPs efficiently with the assistance of surrogate models. In the following sections we introduce a shrinking margin-based equality handling method (category (d)) for SACOBRA. We use the well-known G-problems suite benchmark to assess our algorithm in the context of the research questions.

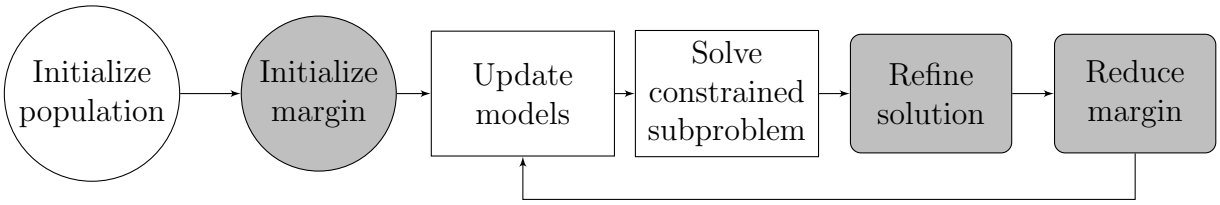

Figure 4.4: Main steps of SACOBRA with the equality handling mechanism.

4.4 Method

SACOBRA as described in Ch. 3 is an efficient surrogate assisted optimizer. The idea behind SACOBRA is to train RBF interpolants for objective and constraint functions and solve a constrained subproblem Eq.(4.3)–(4.5) in each iteration:

$$\begin{aligned} \text{Minimize} \quad & s_0^{(n)}(\vec{x}), && \vec{x} \in [\vec{l}, \vec{u}] \subset \mathbb{R}^d && (4.3) \\ \text{subject to} \quad & s_j^{(n)}(\vec{x}) + \delta^{(n)} \leq 0, && j = 1, 2, \ldots, m && (4.4) \\ & \rho^{(n)} - ||\vec{x} - \vec{x}_p|| \leq 0, && p = 1, 2, \ldots, n && (4.5) \end{aligned}$$

where $s_0^{(n)}$ is the fitness function surrogate model based on n points and $s_j^{(n)}$ stands for the model of the j-th inequality constraint in the n-th iteration. $\mu^{(n)}$ and $\rho^{(n)}$ are internal variables of the COBRA algorithm. More details can be found in Ch. 3.

SACOBRA's contribution on a series of COPs shown in Ch. 3 is significant. But a drawback of this algorithm is that it 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 (a) in Sec. 4.3). However, as it is illustrated with the example in Fig. 4.2, this approach is not viable in general. In the following section we show in detail how SACOBRA is extended to handle equality constraints.

As depicted in Fig. 4.4 the proposed method has three ingredients: (1) initializing the margin, (2) a decrementing feasibility margin, (3) a refine step which aims to move the solutions towards the subspace of equality constraint(s).

4.4.1 The Proposed Equality Handling Approach

Algorithm 8 shows the proposed approach 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 infill point is done by solving an internal optimization problem which minimizes the objective function (4.3) and tries to fulfill the constraints (4.4) – (4.5). Additionally, the expanded equality constraints, Eqs. (4.6) – (4.7), should be satisfied:

$$s_{m+k}^{(n)}(\vec{x}) - \epsilon^{(n)} \leq 0, \qquad k = 1, 2, \ldots, r \tag{4.6}$$

$$-s_{m+k}^{(n)}(\vec{x}) - \epsilon^{(n)} \leq 0, \qquad k = 1, 2, \ldots, r. \tag{4.7}$$

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 $\epsilon^{(n)}$ by moving $\vec{x}^{(n+1)}$ towards the equality line(s).[3] This is done to prevent losing the best solution in the next iteration when the equality margin ϵ is reduced (Fig. 4.5) and to produce solutions with low constraint violation. 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 intersection of the tube-shaped margins around the equality constraints.[4]

The proposed method benefits from having both feasible and infeasible solutions in the population, by gradually reducing the equality margin ϵ.

4.4.2 Initializing the Margin ϵ

To find an appropriate initial value for the margin $\epsilon^{(n)}$ we use the following procedure: we calculate for each initial design point the sum of its constraint violations. $\epsilon^{(0)}$ 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.

4.4.3 Decrementing Margin

The zero-volume feasible space attributed to the k-th equality constraint $h_k(\vec{x}) = 0$ is expanded to a tube-shaped region around it: $|h_k(\vec{x})| - \epsilon^{(n)} \leq 0$ by the proposed algorithm. By gradually reducing the margin $\epsilon^{(n)}$ the solutions are smoothly guided

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

[4] Theoretically, if inequalities are present, the refine step of Eq. (4.10) could make a former feasible solution infeasible in some of the inequality constraints. But this is only true during the initial iterations where $\epsilon^{(n)}$ is large. As $\epsilon^{(n)}$ approaches zero or the very small value ϵ_{final}, the probability of inducing infeasible solutions tends to zero.

Algorithm 4 Constrained optimization with equality handling (EH). Parameters: ϵ_{final}, β.

Choose initial population P by drawing $n = 3d$ points randomly from the search space, evaluate them on the real functions and select best point $\vec{x}^{(b)}$
Initialize EH margin $\epsilon^{(n)}$
Adapt SACOBRA parameters according to P
while $n <$ *budget* **do**
    Build surrogates $s_0^{(n)}, s_1^{(n)}, \ldots, s_{m+r}^{(n)}$ for $f, g_1, \ldots, g_m, h_1, \ldots, h_r$
    Perform SACOBRA optimization step: Minimize Eq. (4.3) subject to Eqs. (4.4) – (4.7), starting from the current best solution $\vec{x}_{best}$. Result: $\vec{x}^{(n+1)}$
    $\vec{x}^{(n+1)} \leftarrow \text{REFINE}(\vec{x}^{(n+1)})$
    Evaluate $\vec{x}^{(n+1)}$ on real functions
    $P \leftarrow P \cup \left\{\vec{x}^{(n+1)}\right\}$
    $\vec{x}_{best} \leftarrow$ Select the best solution so far from P
    $\epsilon^{(n+1)} \leftarrow \max\left\{\epsilon^{(n)}\beta, \epsilon_{final}\right\}$
    $n \leftarrow n + 1$
end while
return ($\vec{x}_{best}$, the best solution)

function REFINE($\vec{x}_{new}$)
    Starting from $\vec{x}_{new}$, minimize Eq. (4.10), the squared sum of all equality constraint violations.
    return ($\vec{x}_r$, the minimization result)
end function

towards the real feasible area. It is difficult to model the triangular shaped $|\ \cdot\ |$-function accurately with RBFs. Therefore we translate every equality constraint to two inequality constraints as follows:

$$\begin{cases} \quad h_k(\vec{x}) - \epsilon^{(n)} \leq 0, & k = 1, 2, \ldots, r \\ -h_k(\vec{x}) - \epsilon^{(n)} \leq 0, & k = 1, 2, \ldots, r. \end{cases}$$

The equality margin ϵ can be reduced in different fashions. Zhang [187] 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:

$$\epsilon^{(n+1)} \quad = \quad \epsilon^{(n)} \cdot \beta_Z = \epsilon^{(n)} \cdot \frac{|P_{inf}|}{|P|} \tag{4.8}$$

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 [187]. 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 8

$$\epsilon^{(n+1)} = \max(\epsilon_f,\ \epsilon^{(n)}\beta), \tag{4.9}$$

with decay factor $\beta > 0.5$. The decay factor β is constant for all problems, independent of the problem dimension d. The equality margin $\epsilon^{(n)}$ is bounded below by ϵ_f.

4.4.4 Refine Mechanism

The **refine step** is done by minimizing the squared sum of all equality constraint violations by means of a conjugate-gradient (CG) method. This minimization step as described in Eq. (4.10) is done based on the surrogates of the equality constraints and not the real equality functions; therefore, no extra real function evaluations are imposed by this step.

$$\text{Minimize} \quad \sum_{k=1}^{r} (s_{m+k}^{(n)}(\vec{x}))^2, \qquad \vec{x} \in [\vec{l}, \vec{u}] \subset \mathbb{R}^d, \tag{4.10}$$

where $s_{m+k}^{(n)}$ is the surrogate model for the k-th equality constraint h_k. Although in black-box COPs we do not have access to the feasible subspace formed by the equality constraints, the refine step tries to move the best found solution in each iteration towards the feasible subspace with assistance of the estimated models of the equality constraints. Furthermore, as visualized in Fig. 4.5, the refine step can be helpful in not losing a good feasible solution after shrinking the margin iteratively.

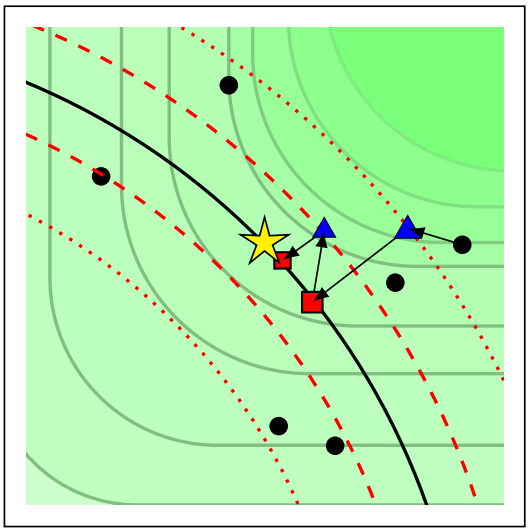

Figure 4.5: Refine step. The shaded (green) contours, golden star and solid (black) line have the same meaning as in Fig. 4.1 and 4.2. In iteration n the dotted lines mark the current feasible tube. The optimization step will result in a point on the tube margin (rightmost blue triangle). The refine 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.

4.5 Experimental Setup

The G-problem suite, described in [107], includes 11 problems with *equality constraints*. In this chapter, all 11 G-problems with equality constraints are considered, while problems containing only inequality constraint(s) are left out. The characteristics of these problems are listed in Tab. 4.1. Just the first four of the eight G-problems (named ‚training' in Tab. 4.1) were used during EH algorithmic development and for tuning the only free parameter β, the decay factor introduced in Sec. 4.4.3. Only after fixing all algorithmic details and parameters, SACOBRA+EH was run on the other seven G-problems (named ‚test' in Table 4.1) and the results were taken 'as-is'. This provides a first indication of the algorithm's performance on unseen problems.

We run the optimization process for every problem with 30 independent randomly initialized populations (using LHS) to have statistically sound results. The size of the initial population is $4 \cdot d$. The equality margin $\epsilon^{(n)}$ has the lower bound set to $\epsilon_f = 10^{-8}$.

Table 4.1: Characteristics of G-problems with equality constraints: d: dimension, LI: the number of linear inequalities, NI: the number of nonlinear inequalities, LE: the number of linear equalities, NE: the number of nonlinear equalities, a: the number of active constraints.

	Fct.	d	type	LI	NI	LE	NE	a
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
	G20	24	nonlinear	0	6	2	12	16
	G21	7	nonlinear	0	1	0	5	6
	G22	22	nonlinear	0	1	8	11	19
	G23	9	nonlinear	0	2	0	0	2

The internal COPs are addressed by `cobyla()` from the NLOPTR package in R [137]. The refine step is done with assistance of the `optim()` function from the STATS package in R, the method parameter is set to L-BFGS-B and the maximum number of refine iterations is set to 10^4.

Fig. 4.6 shows initial runs on the training problems to find a good choice for the decay factor β. Three of the training G-problems (G03, G05 and G11) show good performance for all values of β. The algorithm performs well on G13 with $\beta \in [0.90, 0.94]$. Larger values result in slower convergence, they would converge if the number of iterations were increased, but we allow here only a maximum of 400 iterations. For all subsequent results we fix the decay factor to $\beta = 0.92$.

Additionally, we compare our results with a differential evolution (DE) that also uses a decrementing equality margin. For DE results, we run our experiments using the DEOPTIMR package. The `JDEoptim()` in DEOPTIMR applies the same adaptive equality margin found in [187] but with a more aggressive updating scheme. We set the final feasibility margin to $\epsilon_f = 10^{-8}$.

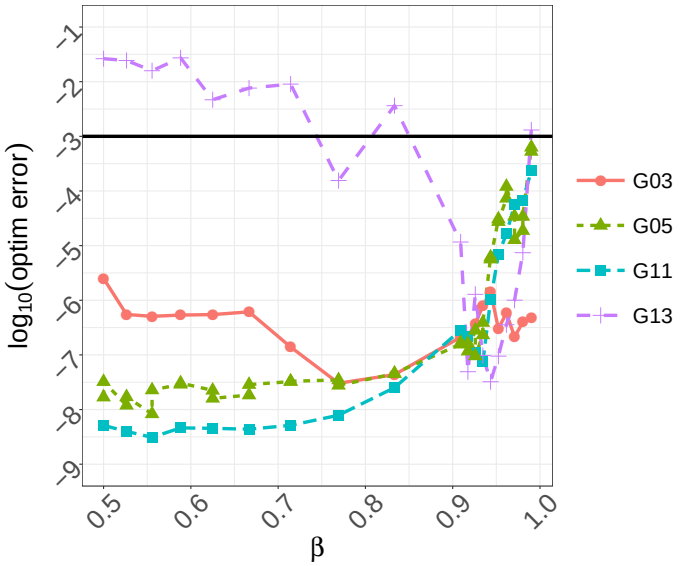

Figure 4.6: Impact of varying parameter β (Eq. (4.9)). Shown is the median of the final optimization error for all training G-problems. The black horizontal line depicts the threshold 10^{-3}. For $\beta \in [0.90, 0.94]$ all problems have a significantly smaller error than this threshold.

4.6 Results & Discussion

4.6.1 Convergence Curves

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 ϵ, two things will happen: (a) There can be intermediate solutions which are apparently better than the optimum.[5] To avoid ‚negative' errors we use the absolute value

$$E_{abs} = |f(\vec{x}_{best}) - f(\vec{x}^*)| \tag{4.11}$$

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

[5] They are on the side of the tube where f is lower than the constrained optimum.

To make the former and the new solutions comparable we form the sum

$$E_{combined} = |f(\vec{x}_{best}) - f(\vec{x}^*)| + V \tag{4.12}$$

where $V = max_{j,k}\{g_j(x), |h_k(x)|, 0\}$ is the maximum violation. This sum $E_{combined}$ is shown as *Combined* curve in the plots of Fig. 4.7 and 4.8.

Fig. 4.7 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 400 function evaluations. The final maximum violation is less than 10^{-4} for all of these problems, i. e. the infeasibility level is negligible according to [107]. The optimization error converges to 10^{-6} or smaller. This means that the algorithm is efficient in locating solutions very close to the true optimum after only a very limited number of evaluations.

Fig. 4.8 shows the optimization process on the five of the ‚test' G-problems (G14, G15, G17, G21 and G23) which were not taken into account during the 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. 4.8 for all five problems. The G14 convergence curve in Fig. 4.8 shows that 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 but 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 small level of infeasibility. Our algorithm can find almost feasible solutions (max. violation $< 10^{-4}$) for all ‚test' G-problems shown in Fig. 4.8 with a reasonably small optimization error except for G21 where the best maximum violation is in order of 10^{-2}. In Fig. 4.8 we can see that the initial solutions of G21 have a very high maximum violation (in the order of 10^3) and after 500 iterations, although the maximum violation is reduced by a large factor 10^5, still did not reach a desired small violation. In this case a faster decaying scheme could be useful. It has to be noted that the worst-case errors for the ‚test' problems are worse than for the training problems. This has to be expected since the parameters were not explicitly tuned to the ‚test' problems. As shown in Fig. 4.9, G20 and G22 are the only problems where our algorithm has difficulties to find solutions with a small constraint violation.

G20 and G22

The G20 and G22 problems are known as very challenging COPs in the literature [179, 171, 88]. Both problems are high-dimensional ($d > 20$) with many equality

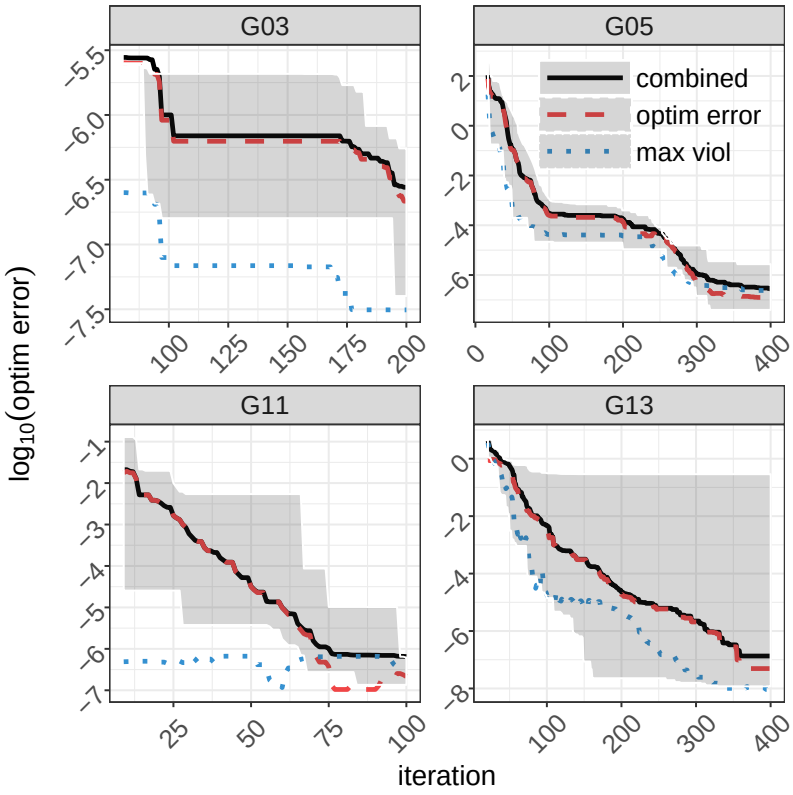

Figure 4.7: Optimization progress for G03, G05, G11, and G13. The dashed (red) curve is the absolute optimization error $|f(\vec{x}_{best}) - f(\vec{x}^*)|$ in every iteration. The dotted (blue) curve is the maximum constraint violation V of the the so-far best solution. The solid (black) line is the combined sum $|f(\vec{x}_{best}) - f(\vec{x}^*)| + V$ 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*.

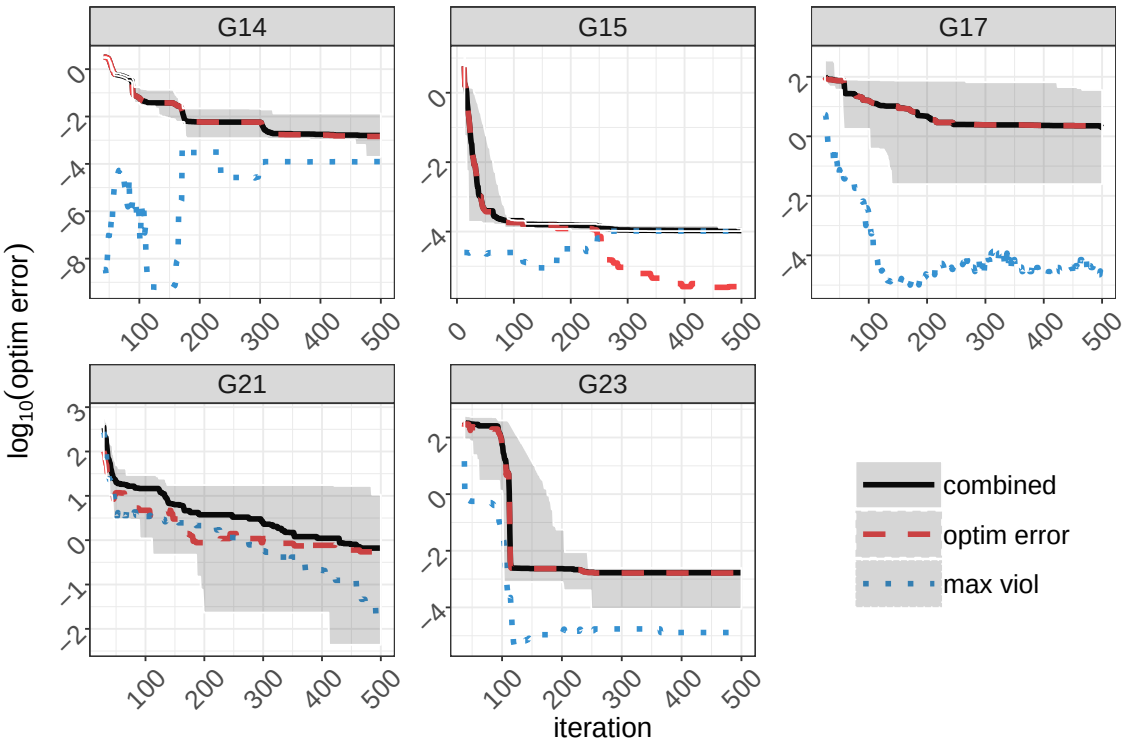

Figure 4.8: Optimization progress for G14, G15, G17, G21, G22 and G23. The dashed (red) curve is the absolute optimization error $|f(\vec{x}_{best}) - f(\vec{x}^*)|$ in every iteration. The dotted (blue) curve is the maximum constraint violation V of the the so-far best solution. The solid (black) line is the combined sum $|f(\vec{x}_{best}) - f(\vec{x}^*)| + V$ 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*.

constraints ($r > 10$). These problems are especially challenging because the feasible subspace formed by the equality constraints is significantly smaller than the original search space of the problem. For example, G22 has a 3-dimensional feasible subspace, while the original problem is defined in 22 dimensions. If we modify G22 under the assumption that equality constraints are not black-box, then SACOBRA can find fully feasible solutions (see Appendix B for G22 derivations). We find a fully feasible solution with the fitness value of 241.609, while the best known solution reported in CEC2006 [107] has a fitness value of 236.431 and a constraint violation in order of 10^{-4}. It is important to mention that often it possible to find the feasible sub-

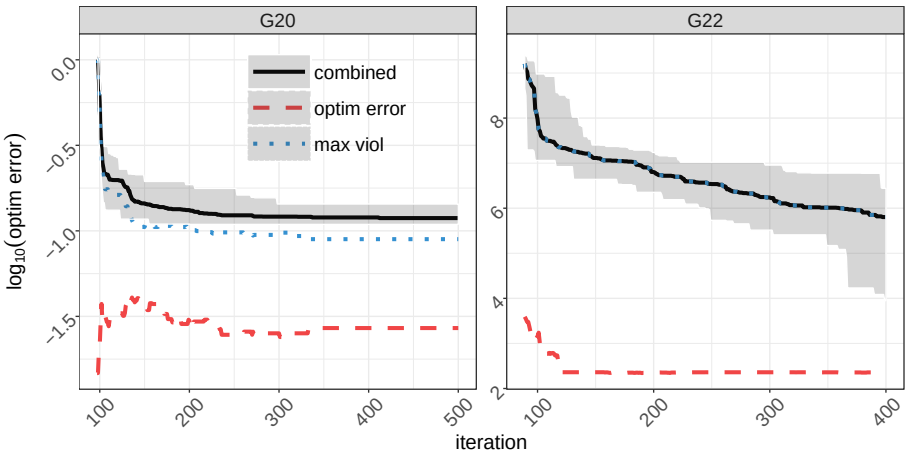

Figure 4.9: Optimization progress for G20, G22. The dashed (red) curve is the absolute optimization error $|f(\vec{x}_{best}) - f(\vec{x}^*)|$ in every iteration. The dotted (blue) curve is the maximum constraint violation V of the the so-far best solution. The solid (black) line is the combined sum $|f(\vec{x}_{best}) - f(\vec{x}^*)| + V$ 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*.

space formed by equality constraints, even if the equality constraints are analytically described. No feasible solution is known so-far for the G20 problem.

4.6.2 Analyzing Update Scheme for Margin ϵ

In order to investigate the effectiveness of the gradual shrinkage of the equality margin ϵ compared to the other update schemes, we have embedded the Zhang update scheme, Eq. (4.8), and a constant scheme with a small equality margin $\epsilon_0 = 10^{-6}$ in our algorithm. In Fig. 4.10, the final optimization results achieved by the different schemes are compared.

Fig. 4.10 shows that the adaptive update scheme proposed by Zhang [187] appears to have a similar behavior as the constant scheme on 3 G-problems (G05, G11, G13). This is because the Zhang decay factor in Eq. (4.8) usually results in a fast decay ($\beta \approx 0.5$), i. e. the margin ϵ 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

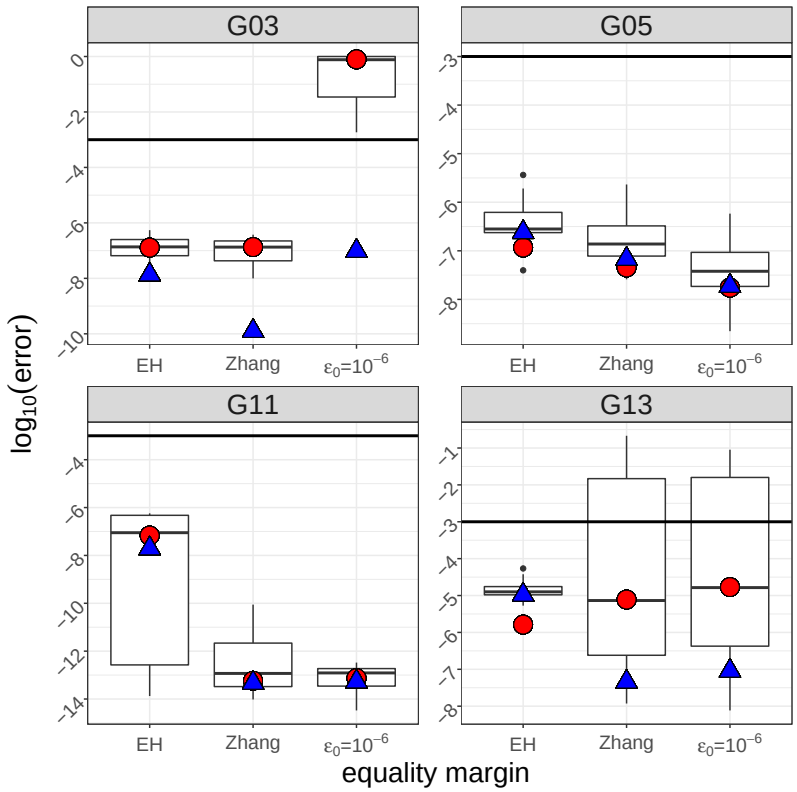

Figure 4.10: Impact of different update schemes for the equality margin ϵ.The circle points (red) are the median of E_{abs} and the triangle points (blue) are the median of the final maximum constraint violations V for 30 independent runs. The box-plots depict the combined sum $E_{combined}$ acc. to Eq. (4.12) for different margin update schemes: (I) our algorithm (EH), (II) Zhang update scheme described in Eq. (4.8), and (III) constant margin $\epsilon_0 = 10^{-6}$. 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).

are problematic and more than 25% of the runs would not converge to the optimum (Fig. 4.10). For two of the tested problems the EH decrementing scheme is clearly the best and for the other two problems is not significantly worse. As illustrated, with the EH decrementing scheme, SACOBRA framework finds solutions with an optimization error smaller than 10^{-3} for all runs of each problem.

In Table 4.2, we compare the proposed algorithm with other state-of-the-art constrained optimization solvers. We show the results from the efficient differential evolutionary algorithm by Zhang [187], an evolutionary algorithm with a novel selection scheme by Jiao [88], the repair genetic algorithm (RGA) by Chootinan [38] and

Table 4.2: 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. Numbers in (brackets): solution violates some constraints.

Fct.	Optimum		SACOBRA+EH [this work]	Zhang [187]	ISRES [151]	RGA 10% [38]	Jiao [88]	DE [31]
G03	-1.0	m	**-1.0**	***-1.0005***	***-1.001***	-0.9999	***-1.0005***	-0.8414
		fe	100	25493	349200	399804	19534	13325
G05	5126.498	m	**5126.498**	***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	-47.763	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	8855.519	8868.539	–	–	**8853.534**	8867.606
		fe	500	43369	–	–	3203	37532
G20	(0.0721)	m	(0.124)	–	–	–	–	–
		fe	500	–	–	–	–	–
G21	193.724	m	(194.270)	193.735	–	–	**193.724**	193.790
		fe	500	23631	–	–	46722	35559
G22	241.09	m	(457.71)	–	–	–	–	–
		fe	500	–	–	–	–	–
G23	-400	m	-400	***-400.055***	–	–	***-400.055***	?
		fe	500	41000	–	–	9410	?
		average fe	350	23272	226400	148826	10200	24742

the improved stochastic ranking evolutionary strategy (ISRES) by Runarsson [151]. The results shown in column 2–5 of Table 4.2 are taken from the original papers. We present in the last column results for differential evolution (DE) [167] with automatic parameter adjustment as proposed by Brest [31]. This was done by running own experiments using the DEoptimR package in R [137]. DE does not perform well on G03 and G13. This being said, 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 [187] and DE use an adaptive equality margin to tackle equality constraints. RGA [38] applies a gradient based repair technique to handle equality and

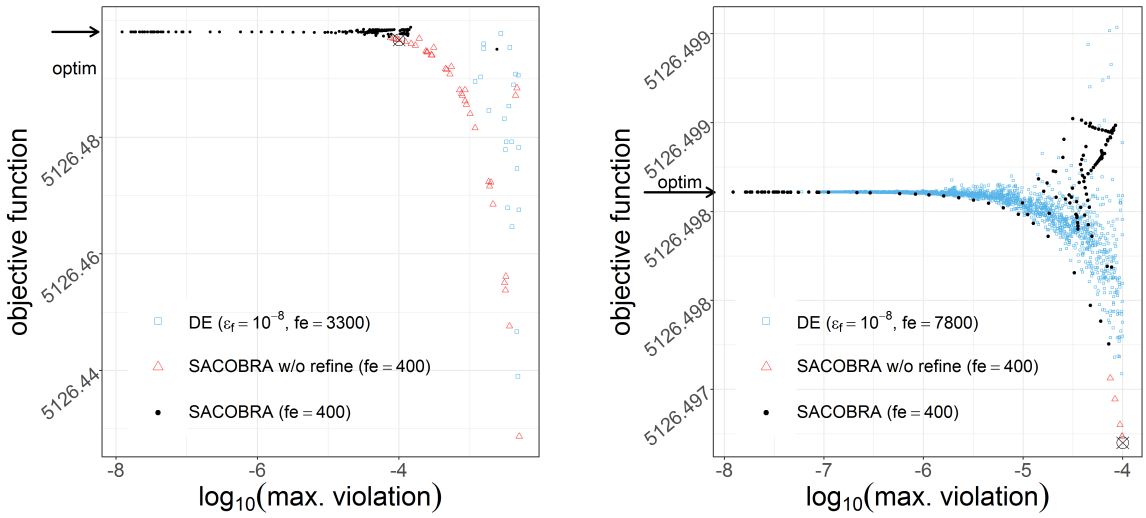

Figure 4.11: Pareto-optimal solutions for G05, minimizing both fitness function and maximum violation, generated by different optimizers. Each point indicates a solution generated by one of the three algorithms SACOBRA, SACOBRA w/o refine and DE for the G05 problem. This plot shows only a subset of solutions generated by each algorithm which are very close to the optimal solution. The gray points are generated by SACOBRA with the exact same configuration described in Sec. 4.5. The red points are the solutions generated by SACOBRA without the refine mechanism and the blue points show the solutions found by DE. The real optimal solution of G05 has a fitness value of 5126.4981. SACOBRA unlike the other two methods, is able to approach the real optimum and generates many solutions on the Pareto front with very limited number of function evaluations.

inequality constraints. Jiao and ISRES make use of a constant and small equality margin ϵ_0.

4.6.3 Pareto Set of Solutions or a Single Solution?

Table 4.2 shows that sometimes the other algorithms find solutions which are slightly infeasible and have a better objective value than the real optimum. These algorithms use the category (c) or (d) equality handling approaches (Sec. 4.3) for searching the best solution within a given margin, so basically they solve an inequality constrained problem. The main dilemma with most of the margin-based equality handling techniques is the fact that often solutions with a fitness value better than the optimum and a violation in order of ϵ, are preferred over the feasible ground truth solution. We tend to partly overcome this problem with applying the refine step. Our algorithm first solves an inequality optimization problem within a margin but this solution is

not evaluated on the real function before refine step tries to project this solution on the feasible subspace.

Although the refine step is a very simple step, it is essential for our algorithm. As shown in Fig. 4.11, SACOBRA with refine is doing a better job in reducing the constraint violation in comparison with the SACOBRA without the refine step. On the one hand, the refine step helps us to move the best solution found in each margin towards the real feasible subspace in each iteration. Although a wrong approximation of the equality constraints in the early iterations can cause a shift towards more violated regions, the model(s) of equality constraint(s) gradually improve by learning about these regions and eventually the refine step can guide the solutions towards the correct direction. On the other hand, since 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 refining, we would lose this point and jump to another feasible point, if any, probably with a much larger objective value.

It is almost impossible to compare our results with the state-of-the-art in a fair way with the information from Tab. 4.2. Therefore, we suggest reporting a Pareto set of solutions minimizing the objective function and the constraint violation, instead of one single solution. Fig. 4.11 shows the infill points found for the G05 problem in the neighborhood of the real optimum in terms of maximum violation and fitness value. We can see that SACOBRA, unlike SACOBRA without refine, is able to approach the real optimum. Combining the results achieved from SACOBRA and SACOBRA w/o refine, we can generate many solutions on the Pareto front with very limited real function calls.

Comparing SACOBRA with DE on Fig. 4.11 shows that SACOBRA is remarkably better in approaching optimum with only 400 fe (function evaluations). SACOBRA and SACOBRA w/o refine together can populate the Pareto front nicely, with in total 800 fe. DE needs more function evaluations (3300 fe, Fig. 4.11-left) to come into the vicinity of the Pareto front, however, many DE points have a constraint violation larger than 10^{-4}. Only if we add more fe to DE (7800 fe, Fig. 4.11-right), then the DE points will more or less densely populate the region near the Pareto front. A more detailed discussion in this regards and similar figures for other G-problems can be found in [18].

It is important to mention that comparing a set of found solutions instead of only the final one as done in Fig. 4.11, helps us to have a better comparison for the performance of the equality constrained algorithms. Additionally, providing a set of solutions for COPs with equality constraints can be beneficial in practice, because the user then can decide to take a solution which fits best to his/her application.

4.7 Conclusion

The results in Sec. 4.6 show that our extended algorithm SACOBRA+EH with a shrinking equality margin (category (d)) can provide reasonable solutions for 8 out of 11 G-problems with equality constraints, while most of these problems were not solvable with the older version of SACOBRA which used the equality-to-inequality transformation scheme (category (b)). SACOBRA+EH cannot solve G20, G21 and G22 problems. To the best of our knowledge these COPs are challenging problems for all constrained optimizers. No other constrained can find feasible solutions for G20 and G22.

SACOBRA benefits from a gradual shrinking of the expanded feasible region as discussed in Sec. 4.6.2. The gradual shrinking smoothly guides the infeasible solutions toward the feasible subspace. 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 as illustrated in Fig. 4.10. Therefore, the second research question **Q4.2** can be answered positively.

We showed that SACOBRA is capable of approaching the true solution of many equality-constrained optimization problems in **less than 500 function evaluations**, this is remarkably more efficient than other optimizers we compared with. Moreover, SACOBRA avoids – at least to a large extent – the often seen dilemma of equality-constraint optimization that a margin leads to solutions 'better than the optimum' by violating some of the equality constraints. We have seen that the refine step – which may be also a useful building block for other optimization schemes – is essential for achieving this goal. As illustrated in Fig. 4.11, SACOBRA finds better solutions for G05 in terms of constraint violation, vicinity to the optimum and efficiency, compared to DE, a well-known algorithm from evolutionary strategies. This being said, we can answer the first research question **Q4.1** and conclude that SACOBRA can be advised as an efficient solver for COPs with equality constraints.

In this chapter we indicated why showing a set of best solutions minimizing both, maximum constraint violation and objective function, is helpful to have a fair comparison of different algorithms. Furthermore, reporting the Pareto front solutions is more practical for real-world applications because the user has a chance to select the solution which suits the application best.