Viability Principles for Constrained Optimization Using a (1+1)-CMA-ES

Andrea Maesani and Dario Floreano

Laboratory of Intelligent Systems,
Institute of Microengineering,
Ecole Polytechnique Fédérale de Lausanne (EPFL),
Lausanne, Switzerland
{andrea.maesani,dario.floreano}@epfl.ch
http://lis.epfl.ch

Abstract. Viability Evolution is an abstraction of artificial evolution which operates by eliminating candidate solutions that do not satisfy viability criteria. Viability criteria are defined as boundaries on the values of objectives and constraints of the problem being solved. By adapting these boundaries it is possible to drive the search towards desired regions of solution space, discovering optimal solutions or those satisfying a set of constraints. Although in previous work we demonstrated the feasibility of the approach by implementing it on a simple genetic algorithm, the method was clearly not competitive with the current evolutionary computation state-of-the-art. In this work, we test Viability Evolution principles on a modified (1+1)-CMA-ES for constrained optimization. The resulting method shows competitive performance when tested on eight unimodal problems.

Keywords: Stochastic optimisation, constrained optimisation, evolution strategy, viability evolution, constraint handling.

1 Introduction

Evolutionary computation methods are often used to solve real-valued black-box optimization problems, a large number of which require satisfying constraints. Without loss of generality, solving a real-valued constrained optimization problem in $\mathbb{R}^n$ means minimizing the objective function $f(\boldsymbol{x})$, $\boldsymbol{x} \in \mathbb{R}^n$, subject to inequalities[1] defined on m constraints function $g_i(\boldsymbol{x}) \leq 0, i = 1, \ldots, m$.

Several approaches have been proposed to solve constrained problems using evolutionary algorithms [1], ranging from rejecting solutions that violate constraints (infeasible solutions) to more sophisticated strategies that modify the ranking of individuals by penalizing the fitness using a function of constraint violations (penalty functions). Other popular approaches include stochastic ranking of solutions [2], ϵ-constrained optimization [3], feasibility rules to

[1] Equality constraints can always be rewritten as inequalities by using a tolerance value on the equality.

T. Bartz-Beielstein et al. (Eds.): PPSN XIII 2014, LNCS 8672, pp. 272–281, 2014.

rank solutions [4], and transformation of constraints into objectives. Although these methods are necessary to handle infeasible solutions and constraints, an efficient optimizer is essential to progress during the search.

Currently, many state-of-the-art algorithms for unconstrained optimization are based on Covariance Matrix Adaptation Evolution Strategy (CMA-ES) [5]. In CMA-ES, a covariance matrix describing correlations between decision variables is learned and adapted during the search to maximize the likelihood of generating successful solutions. Although CMA-ES is a powerful optimizer in unconstrained settings [6], it may suffer from premature convergence in presence of constraints, a common problem in strategies with adaptive step-size control [7]. Furthermore, methods for constrained optimization based on CMA-ES often require providing a feasible solution as a starting point.

A different modelling of objectives and constraints in CMA-ES may offer novel possibilities for handling constraints and allow the initialization of the algorithm from infeasible solutions. Viability Evolution [8,9] is an abstraction of artificial evolution that models an optimization process using viability boundaries, which are modified over time to drive the search towards desirable regions of a search space, as shown in Figure 1. Under this abstraction, mutations can produce viable solutions, which survive, or non-viable solutions, which are eliminated from the population. Viability boundaries are generally defined as admissible ranges of problem objectives and constraints. At the beginning of the search the boundaries are relaxed to encompass all randomly generated initial solutions and then gradually tightened. Once viability boundaries reach the desired target boundaries they are not tightened further, and the evolutionary process is considered complete.

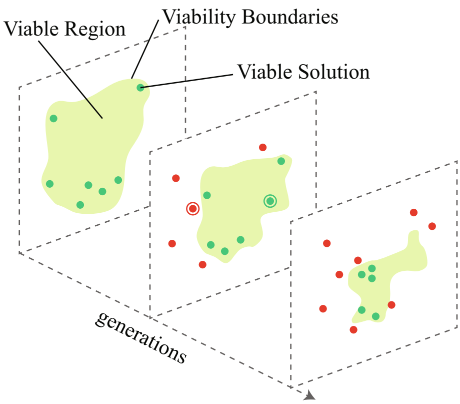

Fig. 1. Viability boundaries initially encompass all randomly generated solutions. We represent the viable region as a projection on a two-dimensional plane of the viability boundaries (shaded area). During the search, the boundaries are made more stringent. Viable solutions are retained in the population (dots in the shaded area), whereas solutions that do not satisfy viability boundaries are eliminated. Mutations can generate solutions (circled dots) that fall outside or inside the viable region.

In this work, we borrow concepts from Viability Evolution, and combine them with active covariance updates for CMA-ES [10], to derive a novel algorithm for constrained optimization. Here, we restrict ourselves to testing our method only in the case where it is started from a feasible solution, as done in [10] which reports current state-of-the-art performance on a set of eight unimodal functions.

The paper is structured as follows. In Section 2 we discuss the state-of-the-art of constraint handling in evolution strategies and we elucidate the workings of a (1+1)-CMA-ES with constraint handling proposed in [10]. In Section 3 we discuss Viability Evolution principles and the proposed approach for constrained optimization. Experimental setup and results of the proposed approach are presented in Section 4. Finally, we conclude with a brief discussion of the proposed approach in Section 5, and we propose future continuations of the work.

2 Related Work

Classical approaches to handle constraints in evolution strategies consist of simply discarding and resampling infeasible solutions [11] or using penalty functions. Penalty functions usually depend on the amount of constraints violation or number of violated constraints [12], and in some cases also on the fitness of selected feasible solutions [13]. The penalty functions can also be adaptive: for example the relative weight of each constraint in the penalty can be modified according to the number of iterations where infeasible solutions are discovered [14], or according to the ratio between feasible and infeasible individuals [15].

Other methods do not use penalty functions. An approach performs selection based on three feasibility rules [16]: feasible individuals are compared on objectives, infeasible ones are compared on total constraint violations, and feasible individuals are always ranked before infeasible ones. Similarly, a recently proposed method modifies the ranking of individuals based on three independent rankings: by objective function, by constraint violation amount, and by number of violated constraints depending on if the solution is feasible or infeasible [17]. Other approaches reduce the probability of generating infeasible solutions when in the proximity of the constraint, by moving the mean of the population [18] or by explicitly controlling the step size using a lower bound [7].

Another way in which constraints can be handled is learning surrogate models for linear constraints. One of these methods has been shown to be a promising approach to reduce the number of constraint function evaluations by predicting if solutions are feasible or infeasible, adapting directly the covariance matrix using the learned information, and repairing solutions that turn out to be infeasible [19]. The work has been recently extended to non-linear constraints, learning models using support vector machines [20]. Another recently proposed variant of CMA-ES [21] makes use of repair mechanisms, but the algorithm is very specific to the problem being solved (financial portfolio optimization).

2.1 (1+1)-CMA-ES with Active Covariance Matrix Adaptation

Among the various methods proposed for handling constraints in CMA-ES, Arnold and Hansen [10] recently proposed a modification of a (1+1)-CMA-ES that has displayed great performance improvements with respect to other methods on unimodal constrained problems when started from a feasible solution. The method maintains a (low-pass filtered) vector representing the direction of violations of steps with respect to each constraint. These vectors are used to update the covariance matrix such that the variance in the direction of violation is reduced. A (1+1)-CMA-ES combines (1+1) selection [22] with covariance matrix adaptation [5]. Given a parent solution $\boldsymbol{x} \in \mathbb{R}^n$, an offspring solution $\boldsymbol{y}$ is sampled according to $\boldsymbol{y} \leftarrow \boldsymbol{x} + \sigma \boldsymbol{A}\boldsymbol{z}$ where $\boldsymbol{A}$ is the Choleski decomposition of the covariance matrix $\boldsymbol{C} = \boldsymbol{A}^T\boldsymbol{A}$ and $\boldsymbol{z} \sim \mathcal{N}(0, \mathbf{I})$ is sampled from a normal distribution. The global step size $\sigma \in \mathbb{R}_+$ is changed according to a modified 1/5 rule proposed in [23]. The probability $P_{succ} \in [0, 1]$ of generating successful solutions and σ are updated at each iteration

$$P_{succ} \leftarrow (1 - c_p)P_{succ} + c_p \mathbb{1}_{f(\boldsymbol{y}) \leq f(\boldsymbol{x})} \tag{1}$$

$$\sigma \leftarrow \sigma exp\left(\frac{1}{d}\left(P_{succ} - \frac{P_{target}}{1 - P_{target}}(1 - P_{succ})\right)\right) \tag{2}$$

where $\mathbb{1}_{f(\boldsymbol{y}) \leq f(\boldsymbol{x})}$ is 1 if the condition is true or 0 otherwise, the learning rate $c_p \in (0, 1]$ determines the fading of P_{succ} and the damping factor d controls the step size variation. P_{target} determines the probability threshold that decreases or increases σ. The covariance matrix is adapted using the original rank-one update rule of CMA-ES, $\boldsymbol{C}^{(g+1)} = \alpha \boldsymbol{C}^{(g)} + \beta \boldsymbol{v}^{(g)} \boldsymbol{v}^{(g)^T}$, which increases the variance in the direction of the provided vector $\boldsymbol{v}$ from one iteration g to the following one. Using a vector of fading successful steps $\boldsymbol{s}$, called the evolution path, in place of vector $\boldsymbol{v}$, allows the strategy to increase the likelihood of sampling new solutions in the direction of already successful steps. In fact, there is no need to maintain the covariance matrix $\boldsymbol{C}$, as updates can be performed directly on the Choleski factor $\boldsymbol{A}$ as proved in [23] according to

$$\boldsymbol{A} \leftarrow \sqrt{\alpha}\boldsymbol{A} + \frac{\sqrt{\alpha}}{\|\boldsymbol{w}\|^2}\left(\sqrt{1 + \frac{\beta}{\alpha}\|\boldsymbol{w}\|^2} - 1\right)\boldsymbol{s}\boldsymbol{w}^T \tag{3}$$

where $\boldsymbol{w} = \boldsymbol{A}^{-1}\boldsymbol{s}$ and $\beta = c_{cov}^+ \in \mathbb{R}^n$. In practice the evolution path $\boldsymbol{s}$ and α are updated depending on P_{succ}. If the probability of success is small ($P_{succ} < P_{thresh}$) then the covariance matrix is updated considering the current step $\boldsymbol{A}\boldsymbol{z}$, such that $\boldsymbol{s} \leftarrow (1 - c)\boldsymbol{s} + \sqrt{c(2 - c)}\boldsymbol{A}\boldsymbol{z}$ and $\alpha = 1 - c_{cov}^+$. Otherwise ($P_{succ} \geq P_{thresh}$), the update does not consider the current step in order to avoid the variance increasing too much in the direction of already successful mutations. In this case the covariance matrix is always updated using Equation 3 but the evolution path is set to $\boldsymbol{s} \leftarrow (1 - c)\boldsymbol{s}$ and $\alpha = 1 - c_{cov}^+ + c_{cov}^+ c(2 - c)$.

An alternative "active" covariance matrix update that also considers particularly unsuccessful steps worse than the fifth ancestor of the current solution was proposed in [24]. In this case, the covariance matrix is updated using the current unsuccessful step $\boldsymbol{Az}$ and the following rule that decreases the variance in the direction of that step[2]

$$\boldsymbol{A} \leftarrow \sqrt{\alpha}\boldsymbol{A} + \frac{\sqrt{\alpha}}{\|\boldsymbol{z}\|^2}\left(\sqrt{1 - \frac{\beta}{\alpha}\|\boldsymbol{z}\|^2} - 1\right)\boldsymbol{Azz}^T \tag{4}$$

where $\alpha = \sqrt{1 + c_{cov}^-}$ and $\beta = c_{cov}^-$.

Interestingly, the same rule can be used to decrease variance in the direction of constraint violations. Similarly to what is done with the evolution path, Arnold and Hansen [10] proposed to use fading vectors of steps that violate constraints in combination with active covariance updates. Specifically, for each constraint i that is violated by step $\boldsymbol{Az}$, the vector $\boldsymbol{v}_i \leftarrow (1 - c_c)\boldsymbol{v}_i + c_c\boldsymbol{Az}$ is updated. Whenever even a single constraint is violated, the covariance matrix is updated according to

$$\boldsymbol{A} \leftarrow \boldsymbol{A} - \frac{B}{\sum_{i=1}^{m} \mathbb{1}_{g_i(\boldsymbol{y})>0}} \sum_{i=1}^{m} \mathbb{1}_{g_i(\boldsymbol{y})>0} \frac{\boldsymbol{v}_i\boldsymbol{w}_i^T}{\boldsymbol{w}_i\boldsymbol{w}_i^T} \tag{5}$$

where $\boldsymbol{w_i} = \boldsymbol{A}^{-1}\boldsymbol{v_i}$. The parameters used in the algorithm are set to the following [24]: $d = 1 + \frac{n}{2}$, $c = \frac{2}{n+2}$, $c_c = \frac{1}{n+2}$, $c_p = \frac{1}{12}$, $B = \frac{0.1}{n+2}$, $P_{target} = \frac{2}{11}$, $P_{thresh} = 0.44$, $c_{cov}^+ = \frac{2}{n^2+6}$, and $c_{cov}^- = \frac{0.4}{n^{1.6}+1}$. We will refer to this method in the following as (1+1)-acCMA-ES (active constrained CMA-ES).

3 Introducing Viability in CMA-ES

Modelling an evolutionary algorithm using the Viability Evolution abstraction offers novel possibilities. For example, in the case of constrained optimization viability boundaries can be defined to relax problem constraints at the beginning of the search, and be made more stringent over time to lead solutions into the feasible regions. The key idea proposed here is to use changing viability boundaries that define admissible regions of the search space (viable regions) in combination with the active covariance matrix updates proposed by Arnold and Hansen [10]. Active covariance updates are used to decrease the variance in the direction of boundary violations. As the boundaries defined on constraint functions values can be relaxed, the algorithm is compatible with infeasible starting solutions. On the other hand, whenever a viable solution is generated, the standard covariance matrix update rule of (1+1)-CMA-ES is employed to increase the variance in the direction that generated the viable solution. Because different boundaries may affect the global probability of generating viable solutions P_{succ}, we maintain a vector of probability of success $\boldsymbol{p}_{succ}$, that tracks which boundary is more likely to cause the generation of non viable solutions. As depicted in Figure 2A, when the covariance matrix is well adapted to a boundary, the probability of generating a new viable solution is greater or equal to 50%. Otherwise, when the probability of success is lower than 50% for at least one

[2] Note that the sign in the parenthesis is inverted. Furthermore, if $\|z\|^2 \geq \frac{1+c_{cov}^-}{2c_{cov}^-}$ then $c_{cov}^- = \frac{1}{2\|z\|^2-1}$.

Algorithm 1. (1+1)-VIE-CMA-ES pseudo-code. Problem objectives and constraints are modelled using the viability boundaries abstraction. Parameters $d, c, c_c, c_p, B, P_{target}, c^+_{cov}$ and c^-_{cov} are defined as in [24].

Require: $\sigma \in \mathbb{R}_+$ initial global step size
$\alpha \leftarrow 1 - c^+_{cov}$, $\beta \leftarrow c^+_{ccov}$, $s \leftarrow 0$
$\boldsymbol{A} \leftarrow \mathbf{I}$
for $i = 1 \ldots m + 1$ **do**
  $\boldsymbol{v}_i \leftarrow [0,\ \ldots,\ 0]_{n\times 1}$ ▷ The last $\boldsymbol{v}_i$ and b_i correspond to the objective
end for
$\boldsymbol{b} \leftarrow [max(0,\ g_1(\boldsymbol{x})),\ \ldots,\ max(0,\ g_m(\boldsymbol{x})),\ \infty]$
$\boldsymbol{p}_{succ} \leftarrow [\frac{1}{2}, \ldots, \frac{1}{2}]$
$\boldsymbol{x} \leftarrow$ randomly generate solution
while $\neg$ termination condition **do**
  $\boldsymbol{z} \sim \mathcal{N}(0, \mathbf{I})$
  $\boldsymbol{y} \leftarrow \boldsymbol{x} + \sigma \boldsymbol{A}\boldsymbol{z}$
  $\boldsymbol{V} \leftarrow [\mathbb{1}_{g_1(\boldsymbol{y})>b_1}, \ldots, \mathbb{1}_{g_m(\boldsymbol{y})>b_m}, \mathbb{1}_{f(\boldsymbol{y})>b_{m+1}}]$ ▷ Boundary violations
  if $\exists\ i : \boldsymbol{V}_i = 1$ **then**
    for all $i : \boldsymbol{V}_i = 1$ **do**
      $\boldsymbol{v}_i \leftarrow (1 - c_c)\boldsymbol{v}_i + c_c \boldsymbol{A}\boldsymbol{z}$
      $\boldsymbol{w}_i \leftarrow \boldsymbol{A}^{-1}\boldsymbol{v}_i$
    end for
    $\boldsymbol{A} \leftarrow \boldsymbol{A} - B \sum_{i=1}^{m} \mathbb{1}_{g_i(\boldsymbol{y})>0} \frac{\boldsymbol{v}_i \boldsymbol{w}_i^T}{\boldsymbol{w}_i \boldsymbol{w}_i^T}$ ▷ Decrease variance
    $\boldsymbol{p}_{succ} \leftarrow (1 - c_p)\boldsymbol{p}_{succ} + c_p[\mathbb{1}_{\boldsymbol{V_1}=0}, \ldots, \mathbb{1}_{\boldsymbol{V_{m+1}}=0}]$ ▷ Update success probability
    if $\exists\ i : \boldsymbol{p}_{succ_i} < \frac{1}{2}$ **then**
      $P_{succ} \leftarrow (1 - c_p)P_{succ}$ ▷ Decrease global P_{succ}
    end if
  else
    $P_{succ} \leftarrow (1 - c_p)P_{succ} + c_p$ ▷ Increase success probabilities
    $\boldsymbol{p}_{succ} \leftarrow (1 - c_p)\boldsymbol{p}_{succ} + c_p$
    $\sigma \leftarrow \sigma exp\left(\frac{1}{d}\left(P_{succ} - \frac{P_{target}}{1-P_{target}}(1 - P_{succ})\right)\right)$
    $\boldsymbol{s} \leftarrow (1 - c)\boldsymbol{s} + \sqrt{c(2 - c)}\boldsymbol{A}\boldsymbol{z}$
    $\boldsymbol{w} \leftarrow \boldsymbol{A}^{-1}\boldsymbol{s}$
    $\boldsymbol{A} \leftarrow \sqrt{\alpha}\boldsymbol{A} + \frac{\sqrt{\alpha}}{\|\boldsymbol{w}\|^2}\left(\sqrt{1 + \frac{\beta}{\alpha}\|\boldsymbol{w}\|^2} - 1\right)\boldsymbol{s}\boldsymbol{w}^T$
    $\boldsymbol{b}_{1..m} \leftarrow \Big[max\Big(0, min\Big(b_1, g_1(\boldsymbol{y}) + \frac{b_1 - g_1(\boldsymbol{y})}{2}\Big)\Big), \ldots,$
        $max\Big(0, min\Big(b_m, g_m(\boldsymbol{y}) + \frac{b_m - g_m(\boldsymbol{y})}{2}\Big)\Big)\Big]$
    if $V_{i:1,\ldots,m} = 0$ **then** ▷ Update boundary on objective when feasible
      $\boldsymbol{b}_{m+1} \leftarrow f(\boldsymbol{y}) + \frac{f(\boldsymbol{x}) - f(\boldsymbol{y})}{2}$
    end if
    $\boldsymbol{x} \leftarrow \boldsymbol{y}$
  end if
end while

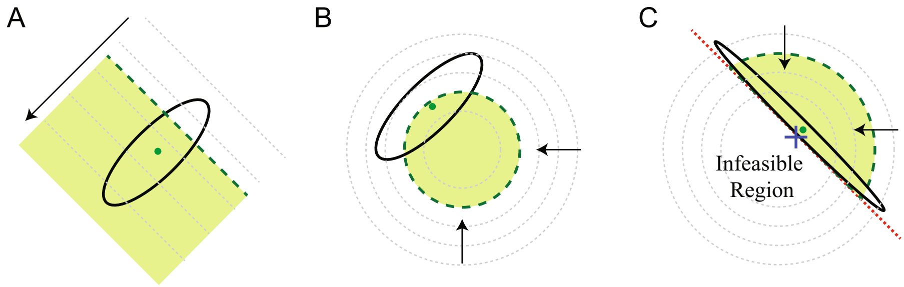

Fig. 2. Possible scenarios encountered during a search. A) The covariance matrix (ellipsoid in solid line) is well adapted with respect to a boundary (dashed line). The probability of generating a successful solution in the viability region (shaded area) is greater than 50%. Isocline of the objective function are shown as thin dotted lines and the gradient direction is shown by the arrow. The mean of the search distribution is represented as a dot. B) The covariance matrix should be adapted. Probability of generating successful solutions is lower than 50%. C) The method encounters difficulties when the direction to reach the optimum (shown as a cross) is the same that generates infeasible solutions that violate the constraint (thick dotted line).

boundary, as shown in Figure 2B, the covariance matrix should be modified and the global step size reduced. To achieve this, we reduce the global P_{succ} probability. Conversely, the overall P_{succ} probability and all elements of the $\boldsymbol{p}_{succ}$ vector are increased whenever a viable solution is generated. Note that in the method presented in [10] not adapting P_{succ} on failure may lead to the use of outdated information for step-size adaptation.

The pseudo-code of our method, referred to as (1+1)-VIE-CMA-ES, is presented in Algorithm 1. The user must only provide an initial step size σ. The algorithm sets the initial viability boundaries b as either the target boundary (0 for the constraints) or a relaxed value if an infeasible solution is provided. The initial boundary for the objective is set to ∞. At each iteration, boundary violations V are checked. The active covariance matrix update for feasible solutions (Equation 4) and the stall of updates of the original method in presence of high probability of success are not used. A single update rule is applied whenever a viable solution (that does not violates the boundaries b) is generated. When this happens, the mean of the population is updated to the new viable solution and the boundaries are tightened.

4 Results

The proposed method was tested on all the eight benchmark functions used in [10]. These benchmark functions include problems from two to ten dimensions with up to eight non-linear constraints. The experimental setup is identical to the one reported in [10], including the same number of repetitions, equivalent generation of initial solutions, the same termination condition, and the same parameter settings for the (1+1)-CMA-ES. For each benchmark function we counted the total number of objective function and constraints function evaluations. We tested the method starting it 99 times from different initial solutions, uniformly sampled from the

Table 1. Experimental results of the (1+1)-VIE-CMA-ES and comparison against the (1+1)-acCMA-ES proposed in [10]

	g06		g07		g09		g10	
	VIE-CMA	acCMA	VIE-CMA	acCMA	VIE-CMA	acCMA	VIE-CMA	acCMA
Function Evaluations								
10th	282	**272**	**1578**	1939	**1305**	1430	**1387**	2794
50th	333	**308**	**1794**	2211	**1452**	1674	**1697**	3976
90th	385	**364**	**2049**	2703	**1595**	2074	**2554**	5369
Constraint Evaluations								
10th	**797**	827	**7184**	10435	**3474**	3626	**7360**	15621
50th	**900**	1060	**7545**	11283	**3660**	4106	**8295**	18781
90th	**986**	1223	**8032**	12704	**3913**	5075	**11322**	23088

	TR2		2.40		2.41		HB	
	VIE-CMA	acCMA	VIE-CMA	acCMA	VIE-CMA	acCMA	VIE-CMA	acCMA
Function Evaluations								
10th	465	**376**	**863**	1326	**820**	1483	638	**623**
50th	520	**443**	**1023**	1990	**954**	2271	**734**	768
90th	561	**510**	**1209**	3326	**1100**	3581	**841**	1150
Constraint Evaluations								
10th	751	**616**	**3166**	4551	**3183**	5235	2659	**2338**
50th	812	**708**	**3570**	6994	**3449**	8108	**2893**	2912
90th	884	**839**	**3899**	11114	**3801**	12056	**3185**	3970

solution space until a feasible solution is found. Iterations needed to obtain the starting feasible solution are not counted in the results, as in [10].

Results are reported in Table 1. The method is competitive on seven out of eight problem. Our method has medians lower than what were reported by Arnold and Hansen [10] for constraint function calls by a factor of 0.15, 0.33, 0.11, 0.56, 0.49, 0.57 on `g06`, `g07`, `g09`, `g10`, `2.40`, `2.41` respectively and almost identical performance on `HB`. In the linear constrained sphere function problem `TR2`, our method exceeds values reported by Arnold and Hansen [10] by a factor 0.15. In one problem, `g06`, our method, while being better on the overall number of constraint evaluations, performs slightly worse on number of objective function evaluations.

In our view, one of the reasons of decreased performance in the `TR2` problem probably lies in the specific orientation of the constraint. From experimental investigation, we observed that the mean of the search distribution tends to align to the normal direction to the optimum (a situation similar to the one depicted in Figure 2C), which in this case is also the same direction that is most likely to violate the constraint. Probably, in cases like this one when the direction of constraint violation is very close to the direction of viable solutions generation, the covariance matrix update should be stalled, or the variance should be decreased along the other axis.

5 Discussion and Future Work

In this paper, we proposed (1+1)-VIE-CMA-ES, a method that combines viability boundaries and active covariance matrix updates in a (1+1)-CMA-ES. Our algorithm showed competitive performance with respect to state-of-the-art methods on all the benchmark problems except on the constrained sphere function problem TR2. Further investigations are needed to solve the lower performance experienced on TR2. Here, we tested the method only when starting from feasible initial solutions, but our algorithm is also compatible with infeasible starting solutions. In the future, we will proceed with a rigorous evaluation of the method when initialized from infeasible solutions. Also, more research will be needed for tackling multimodal problems using the approach presented here.

It is important to note that dealing with constraints and objectives using the same algorithmic framework allows one to readily extend the method to situations not directly manageable by standard CMA-ES. We anticipate that the coupling of changing viability boundaries and active covariance updates could also potentially be used in multi-objective optimization. For example, a "virtual" boundary may be learned on the Pareto front and made more stringent over time to push solutions towards the optimal Pareto front. The combined use of viability boundaries and active covariance updates might pave the way for a new class of powerful algorithms that can manage unconstrained, constrained and multi-objective problems under the same algorithmic scheme.

Acknowledgements. The authors would like to thank Joshua E. Auerbach and Giovanni Iacca for precious advice and useful comments on the manuscript. This research has been supported by the Swiss National Science Foundation, grant 200021_127143 and the FET-Open Grant 308943 within the 7th Framework Programme for Research of the European Commission.

References

1. Mezura-Montes, E., Coello Coello, C.A.: Constraint-handling in nature-inspired numerical optimization: Past, present and future. Swarm and Evolutionary Computation 1(4), 173–194 (2011)
2. Runarsson, T.P.: Stochastic ranking for constrained evolutionary optimization. IEEE Transactions on Evolutionary Computation 4(3), 284–294 (2000)
3. Takahama, T., Sakai, S.: Constrained optimization by the ϵ constrained differential evolution with an archive and gradient-based mutation. In: IEEE Congress on Evolutionary Computation (CEC 2010), pp. 1–9. IEEE Press (2010)
4. Deb, K.: An efficient constraint handling method for genetic algorithms. Computer Methods in Applied Mechanics and Engineering 186(2-4), 311–338 (2000)
5. Hansen, N., Müller, S.D., Koumoutsakos, P.: Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (CMA-ES). Evolutionary Computation 11(1), 1–18 (2003)
6. Hansen, N., Auger, A., Ros, R., Finck, S., Pošík, P.: Comparing results of 31 algorithms from the black-box optimization benchmarking BBOB-2009. In: Proceedings of the 12th Annual Conference on Genetic and Evolutionary Computation (GECCO 2010), pp. 1689–1696. ACM Press (2010)
7. Kramer, O., Schwefel, H.: On three new approaches to handle constraints within evolution strategies. Natural Computing 5(4), 1–22 (2006)

8. Mattiussi, C., Floreano, D.: Viability Evolution: Elimination and Extinction in Evolutionary Computation. Technical Report (April 2003)
9. Maesani, A., Fernando, P.R., Floreano, D.: Artificial evolution by viability rather than competition. Plos One 9(1), e86831 (2014)
10. Arnold, D.V., Hansen, N.: A (1+1)-CMA-ES for constrained optimisation. In: Proceedings of the Fourteenth International Conference on Genetic and Evolutionary Computation Conference (GECCO 2012), pp. 297–304. ACM Press (2012)
11. Schwefel, H.-P.P.: Evolution and optimum seeking: the sixth generation. John Wiley & Sons, Chichester (1993)
12. Hoffmeister, F., Sprave, J.: Problem-Independent Handling of Constraints by Use of Metric Penalty Functions. In: Proceedings of the Fifth Annual Conference on Evolutionary Programming, pp. 289–294. MIT Press (1996)
13. Oyman, A., Deb, K., Beyer, H.G.: An alternative constraint handling method for evolution strategies. In: Proceedings of the 1999 IEEE Congress on Evolutionary Computation, pp. 612–619. IEEE Press (1999)
14. Collange, G., Delattre, N., Hansen, N., Quinquis, I., Schoenauer, M.: Multidisciplinary Optimization in the Design of Future Space Launchers. In: Multidisciplinary Design Optimization in Computational Mechanics, pp. 487–496. John Wiley & Sons, Inc. (2010)
15. Kramer, O., Schlachter, U., Spreckels, V.: An adaptive penalty function with metamodeling for constrained problems. In: IEEE Congress on Evolutionary Computation (CEC 2013), pp. 1350–1354. IEEE Press (June 2013)
16. Mezura-Montes, E., Coello Coello, C.A.: A simple multimembered evolution strategy to solve constrained optimization problems. IEEE Transactions on Evolutionary Computation 9(1), 1–17 (2005)
17. Kusakci, A.O., Can, M.: A novel evolution strategy for constrained optimization in engineering design. In: XXIV International Conference on Information, Communication and Automation Technologies (ICAT), pp. 1–6. IEEE Press (October 2013)
18. Kramer, O., Ting, C.K., Büning, H.K.: A New Mutation Operator for Evolution Strategies for Constrained Problems. In: IEEE Congress on Evolutionary Computation (CEC 2005), vol. 3, pp. 2600–2606. IEEE Press (2005)
19. Kramer, O., Barthelmes, A., Rudolph, G.: Surrogate constraint functions for CMA evolution strategies. In: Mertsching, B., Hund, M., Aziz, Z. (eds.) KI 2009. LNCS, vol. 5803, pp. 169–176. Springer, Heidelberg (2009)
20. Gieseke, F., Kramer, O.: Towards non-linear constraint estimation for expensive optimization. In: Esparcia-Alcázar, A.I. (ed.) EvoApplications 2013. LNCS, vol. 7835, pp. 459–468. Springer, Heidelberg (2013)
21. Beyer, H.G., Finck, S.: On the Design of Constraint Covariance Matrix Self-Adaptation Evolution Strategies Including a Cardinality Constraint. IEEE Transactions on Evolutionary Computation 16(4), 578–596 (2012)
22. Beyer, H.G., Schwefel, H.P.: Evolution strategies – a comprehensive introduction. Natural Computing 1(1), 3–52 (2002)
23. Igel, C., Suttorp, T., Hansen, N.: A Computational Efficient Covariance Matrix Update and a (1+1)-CMA for Evolution Strategies. In: Proceedings of the 8th Annual Conference on Genetic and Evolutionary Computation (GECCO 2006), pp. 453–460. ACM Press (2006)
24. Arnold, D.V., Hansen, N.: Active Covariance Matrix Adaptation for the (1+1)-CMA-ES. In: Proceedings of the 12th Annual Conference on Genetic and Evolutionary Computation (GECCO 2010), pp. 285–392. ACM Press (2010)