(μ, λ)-CCMA-ES for Constrained Optimization with an Application in Pharmacodynamics

Georgios Arampatzis
Computational Science and Engineering Laboratory, ETH Zürich
Switzerland
Collegium Helveticum, ETH Zürich
Switzerland

Daniel Wälchli
Computational Science and Engineering Laboratory, ETH Zürich
Switzerland

Pascal Weber
Computational Science and Engineering Laboratory, ETH Zürich
Switzerland

Henri Rästas
Computational Science and Engineering Laboratory, ETH Zürich
Switzerland

Petros Koumoutsakos*
Computational Science and Engineering Laboratory, ETH Zürich
Switzerland
Collegium Helveticum, ETH Zürich
Switzerland

ABSTRACT

We present the algorithm CCMA-ES, an extension to CMA-ES, an evolution strategy that has shown to perform well in a broad range of black-box optimization problems. The (μ, λ)-CMA-ES effectively handles nonlinear nonconvex functions but faces difficulties in constrained optimization problems. We introduce viability boundaries to improve the search for an initial point in the valid domain and adapt the covariance matrix using normal approximations to maintain the inequality constraints. Using benchmark problems from 2006 CEC we compare the performance of CCMA-ES with a state of the art optimization algorithm (mViE) showing favorable results. Finally, CCMA-ES is applied to a pharmacodynamics problem describing tumor growth, and we demonstrate that CCMA-ES outperforms mViE in terms of the objective function value and total function evaluations.

CCS CONCEPTS

• **Mathematics of computing** → **Nonconvex optimization**; • **Applied computing** → *Mathematics and statistics*; Computational biology.

KEYWORDS

Stochastic optimization, constraint handling, evolution strategy, viability evolution, covariance matrix adaptation, pharmacodynamics

ACM Reference Format:
Georgios Arampatzis, Daniel Wälchli, Pascal Weber, Henri Rästas, and Petros Koumoutsakos. 2019. (μ, λ)-CCMA-ES for Constrained Optimization with an Application in Pharmacodynamics. In *Proceedings of the Platform for Advanced Scientific Computing Conference (PASC '19), June 12–14, 2019, Zurich, Switzerland.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3324989.3325725

1 INTRODUCTION

Covariance Matrix Adaptation – Evolutionary Strategy (CMA-ES) [5, 6] has been successfully deployed for a broad range of nonlinear, nonconvex optimization problems represented by black box functions with no access to derivatives. One frontier for CMA-ES has been the efficient handling of constraints. In [1] an adaptation mechanism for the covariance matrix on the constraint boundaries has been introduced for the (1+1)-CMA-ES. This technique has been further extended in [10] by the use of viability boundaries. Finally, in [11] the two previous approaches have been incorporated in a memetic algorithmic framework. In [2] the authors had amended a CMA-ES-like framework so that convergence results are possible, and they handle constraints by more standard optimization means.

In this work, we introduce the covariance adaptation [1] and the viability principles [10] in the general (μ, λ)-CMA-ES in order to address high dimensional constraint optimization problems. We assess the efficiency of our algorithm on the benchmark problems used by Maesani et al. [11]. This includes a subset of the problems proposed in the 2006 IEEE International Conference on Evolutionary Computation (2006 CEC).

We apply CCMA-ES to determine an optimal treatment schedule for diffuse, low-grade gliomas. We employ a parametric pharmacodynamics model that has been presented by Ribba [12] for this kind of tumors. The model is augmented by a set of constraints to account for treatment's toxicity to specific patients, thus increasing its clinical relevance [7]. Tumor growth is described by a linear system of parametric differential equations accompanied by a set of inequality constraints. The goal is to identify an optimal treatment schedule that minimizes tumor size.

[*] Corresponding author: petros@ethz.ch

PASC '19, June 12–14, 2019, Zurich, Switzerland

ACM ISBN 978-1-4503-6770-7/19/06…$15.00
https://doi.org/10.1145/3324989.3325725

2 BACKGROUND

We are interested in solving the constraint optimization problem,

$$\boldsymbol{x}^* = \min_{\boldsymbol{x} \in \Omega} f(\boldsymbol{x}),$$

for $\Omega \subset \mathbb{R}^n$, under the inequality constraints,

$$h_j(\boldsymbol{x}) \leq 0, \quad j \in \{1, \dots, m\}.$$

We assume that the objective function f and the constraints h_j are given by a black-box simulator. (μ, λ)-CMA-ES is an algorithm that is designed to work for black-box optimization problems with many local minima. Moreover, since the algorithm is based on independent evaluations of the objective function, it can be parallelized easily and is perfectly suited for computationally demanding problems that require high performance computing.

In this work, we present an extension of (μ, λ)-CMA-ES in order to effectively handle constraints. In the next section we give a brief presentation of the main characteristics of (μ, λ)-CMA-ES.

2.1 (μ, λ)-CMA-ES

CMA-ES approximates the minimum of an objective function f by sampling λ points $\boldsymbol{x}_i$ from a normal distribution $\mathcal{N}(\boldsymbol{m}, \Sigma)$ with an iteratively computed mean $\boldsymbol{m}$ and covariance Σ. Samples are evaluated and sorted based on their corresponding function values. The mean $\boldsymbol{m}$ and covariance matrix Σ are adapted based on the μ best samples in order to increase the probability of sampling future individuals in the direction of favorable samples. In the following we will give a brief overview of the different parts of the algorithm (see algorithm 1 and fig. 1). For a more detailed discussion we refer to [4] and [8].

Algorithm 1 CMA-ES overview

```
Initialize problem parameters                         ▷ fig. 1a
while Termination criteria not met do
    Sampling                                          ▷ fig. 1b
    Evaluation                    ▷ Evaluate individual fitness
    Selection and recombination                       ▷ fig. 1c
    Adaptation                                        ▷ fig. 1d
end while
Return solution
```

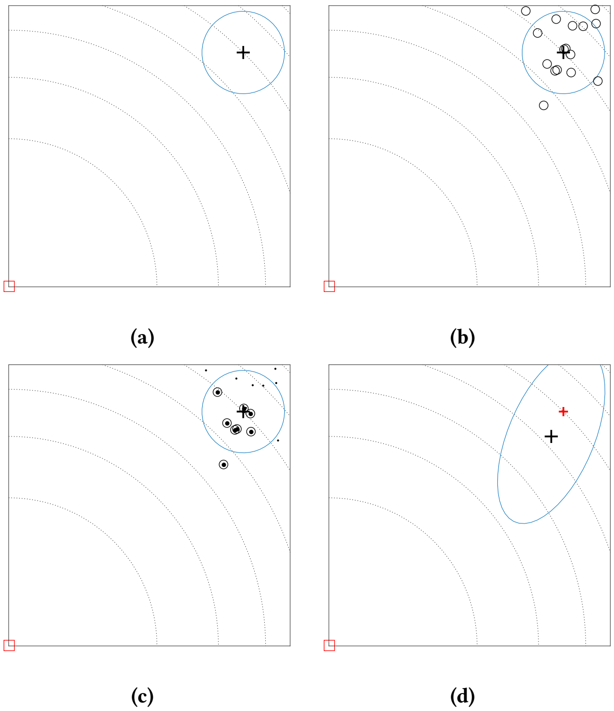
Sampling. Each generation g of the CCMA-ES starts by sampling a new population. The individuals $\boldsymbol{x}_i^{g+1}$ with $i \in \{1, \dots, \lambda\}$ are drawn from an n−dimensional normal distribution $\mathcal{N}(\boldsymbol{m}, \Sigma)$ according to

$$\boldsymbol{x}_i^{g+1} = \boldsymbol{m}^g + \sigma^g \boldsymbol{A}^g z_i, \quad \text{with} \quad z_i \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}). \tag{1}$$

Here $\sigma \in \mathbb{R}$ is the step size and $\boldsymbol{A}$= $\boldsymbol{BD}$ is the Cholesky factor of the covariance matrix,

$$\Sigma = \sigma^2 \boldsymbol{C} = \sigma^2 \boldsymbol{A}\boldsymbol{A}^\top.$$

Evaluation. The objective function f is evaluated at the obtained individuals $\boldsymbol{x}_i$ and sorted $f(\boldsymbol{x}_{1:\lambda}) \leq f(\boldsymbol{x}_{2:\lambda}) \leq \cdots \leq f(\boldsymbol{x}_{\lambda:\lambda})$. To denote the i-th fittest individual we introduce the notation $\boldsymbol{x}_{i:\lambda}$.


(a)

(b)

(c)

(d)

Figure 1: CMA-ES steps for the first generation with $f(x) = x_1^2 + x_2^2$, $\boldsymbol{m}^0 = [5, 5]$, $C^0 = \boldsymbol{I}$ and $\sigma^0 = 0.5$. The red square at the bottom left shows the location of the global minimum. Figure 1a shows the starting mean denoted by the cross and the circle of equal probability of the initial normal distribution. Figure 1b shows a population of 16 individuals sampled from the initial distribution. Figure 1c displays the selection step, with the fittest individuals presented as half-filled circles. Finally, the selected individuals are used to adapt the mean (new one black, old one red) and covariance matrix to produce the ellipse of equal probability shown in fig. 1d.

Selection and recombination. We choose the μ values with the smallest function value. The mean $\boldsymbol{m}^{g+1}$ is updated via

$$\boldsymbol{m}^{g+1} = \sum_{i=1}^{\mu} w_i \boldsymbol{x}_{i:\lambda}^{g+1}. \tag{2}$$

There are many ways to choose the weights w_i under the constraint that they are positive and sum up to 1. A reasonable choice is to assign decreasing weights such that favorable points have a greater impact on the new mean.

Adaptation. The crucial step of CMA-ES is the adaptation of the covariance matrix in a way that increases the probability of sampling individuals that are close to favorable individuals obtained over the previous generations. There are three adaptation steps.

In the first step, the covariance matrix is adapted using the μ fittest individuals to perform a rank-μ update, which reads [5]

$$C^{g+1} = (1 - c_\mu)\, C^g + c_\mu \sum_{i=1}^{\mu} w_i\, \boldsymbol{y}_{i:\lambda}^{g+1} \left(\boldsymbol{y}_{i:\lambda}^{g+1}\right)^\top ,$$

$$\text{with} \quad \boldsymbol{y}_i^{g+1} = \frac{\boldsymbol{x}_i^{g+1} - \boldsymbol{m}^g}{\sigma^g} ,$$

and c_μ is the learning rate.

In the second step, a rank-1 update is applied based on the evolution path $\boldsymbol{p}_c$ [6],

$$C^{g+1} = (1 - c_1)\, C^g + c_1 \boldsymbol{p}_c^{g+1} \left(\boldsymbol{p}_c^{g+1}\right)^\top ,$$

$$\text{with} \quad \boldsymbol{p}_c^{g+1} = (1 - c_c)\boldsymbol{p}_c^g + \sqrt{c_c(2 - c_c)\mu_{\text{eff}}}\, \frac{\boldsymbol{m}^{g+1} - \boldsymbol{m}^g}{\sigma^g},$$

where c_1 and c_c are the learning rates and the factor μ_{eff} normalizes $\boldsymbol{p}_\sigma^{g+1}$. For $c_c = \mu_{\text{eff}} = 1$ the evolution path p_c^g reduces to the most recent change of the mean.

For the final adaptation both rank−μ and rank−1 updates are combined by a weighted sum of the two updates. For the covariance matrix $\boldsymbol{C}$ the total update is given by

$$\begin{aligned} C^{g+1} = (1 - c_1 - c_\mu)\, C^g &+ c_1 \boldsymbol{p}_c^{g+1} \left(\boldsymbol{p}_c^{g+1}\right)^\top \\ &+ c_\mu \sum_{i=1}^{\mu} w_i \boldsymbol{y}_{i:\lambda}^{g+1} \left(\boldsymbol{y}_{i:\lambda}^{g+1}\right)^\top . \end{aligned} \tag{3}$$

The updates introduced do not explicitly control the overall scale of the distribution thus affecting the algorithm's performance. As a remedy, and the final component of the adaptation, we control the step-size σ via

$$\sigma^{g+1} = \sigma^g \exp\left(\frac{c_\sigma}{d_\sigma}\left(\frac{\|\boldsymbol{p}_\sigma^{g+1}\|}{E[\|\mathcal{N}(0, \boldsymbol{I})\|]} - 1\right)\right),$$

where

$$\boldsymbol{p}_\sigma^{g+1} = (1 - c_\sigma)\boldsymbol{p}_\sigma^g + \sqrt{c_\sigma(2 - c_\sigma)\mu_{\text{eff}}}\, \left(C^g\right)^{-\frac{1}{2}} \frac{\boldsymbol{m}^{g+1} - \boldsymbol{m}^g}{\sigma^g},$$

and

$$\left(C^g\right)^{-\frac{1}{2}} = \boldsymbol{B}\boldsymbol{D}^{-1}\boldsymbol{B}^\top .$$

Here we introduce further parameters, the learning rate c_σ and the damping factor d_σ. By including μ_{eff} we normalize $\boldsymbol{p}_\sigma^{g+1}$ in the same sense as before. The factor $(C^g)^{-\frac{1}{2}}$ is used to make the length $\|\boldsymbol{p}_\sigma\|$ comparable to a reference length $E[\|\mathcal{N}(0, \boldsymbol{I})\|]$, which is the length of $\boldsymbol{p}_\sigma$ under a random selection of points as opposed to the selection of fittest points [4].

The CMA-ES algorithm, though very good at finding minima in a black box setting, has a few drawbacks. Most notably it cannot efficiently account for constraints and has difficulties finding optima close to constraint bounds. In the following we address this issue by proposing a new variant of the (μ, λ)–CMA-ES.

2.2 Related Work

A straightforward modification of the CMA-ES algorithm is to account for constraints by resampling all individuals that violate a given constraint. This strategy works well when the minima of the function do not lie near the constraint boundary. If not, or if the objective function dimension is high, resampling individuals that are not in the feasible region will result in many rejections, respectively many repeated resamplings, and consequently very long algorithm runtimes. Other approaches to account for constraints in CMA-ES include penalty functions [3] or modified ranking methods [13]. An algorithm recently proposed is mViE [11], an extension of $(1 + 1)$-Vie-CMA-ES [10], developed to handle multi-modal constrained optimization problems. Its performance has been compared in a comprehensive manner to most of the current methods and therefore we refer to this paper for an extended overview on algorithms for constrained optimization problems.

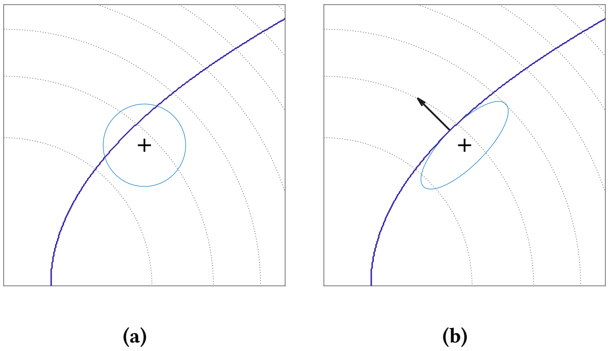


Figure 2: The region towards the lower right below the constraint boundary (highlighted in blue) represents the feasible region. Given a constraint normal (black arrow) the ideal modification of the normal distribution is given in fig. 2a. The variance of the distribution gets compressed in the direction of the given constraint normal as shown in fig. 2b.

The present method builds on the methodology proposed by Arnold and Hansen [1]. It uses invalid individuals to reshape the covariance matrix, in the spirit of rank-1-update of the unconstrained CMA-ES algorithm. This approach has shown to be very effective at solving constrained optimization problems. In this work we aim to extend this constraining method from $(1 + 1)$-CMA-ES to apply to the general (μ, λ)-CMA-ES.

3 CCMA-ES

The underlying idea of Arnold and Hansen's constraining method is to extend the $(1 + 1)$-CMA-ES to modify the covariance matrix such that the variance in the direction of the constraint normal is reduced (see fig. 2). Before the covariance matrix can be adapted, the constraint normal or an approximation of it is needed. For this purpose each of the constraints h_j for $j \in \{1, \ldots, m\}$ will maintain an exponentially fading record $\boldsymbol{v}_j$ that approximates the constraint normal. The normal $\boldsymbol{v}_j$ is initialized to zero and updated whenever a constraint h_j is violated by any sample $\boldsymbol{x_i}$,

$$\boldsymbol{v}_j = (1 - c_v)\boldsymbol{v}_j + c_v \boldsymbol{y_i} , \tag{4}$$

where $\boldsymbol{y_i} = \boldsymbol{A}\boldsymbol{z_i}$ with $\boldsymbol{z_i} \sim \mathcal{N}(0, \boldsymbol{I})$ is the vector used to sample the i-th population individual, c_v is an adjustable parameter reflecting how quickly the information in the current normal fades. This type of update results in a cancellation of the mean components of $\boldsymbol{Az}$

that are tangential to the local constraint boundary. If a candidate violates the constraint we update the covariance matrix accoring to

$$C = C - \frac{\beta}{\alpha_0(\boldsymbol{x})} \sum_{j=1}^{m} \alpha_j(\boldsymbol{x}) \frac{\boldsymbol{v}_j \boldsymbol{v}_j^\top}{\|\boldsymbol{v}_j\|^2}\,, \tag{5}$$

where $a_0(\boldsymbol{x}) = \sum_{j=1}^{m} \alpha_j(\boldsymbol{x})$ and $\alpha_j(\boldsymbol{x})$ is equal to 1 if $h_j(\boldsymbol{x}) > 0$ and 0 otherwise. Here, β is an adjustable parameter that determines the size of the update. After the update, the individuals are resampled according to the updated covariance matrix. The process is continued until the whole population is located in the feasible region. Algorithm 2 summarizes the constraint handling loop which is performed straight after sampling the population in Algorithm 1 and Figure 3 gives a visual interpretation of the constraining method.

Algorithm 2 Pseudo code for constraint handling in CCMA-ES

```
Require: Sampled individuals $x_i = m + \sigma y_i$          ▷ see eq. (1)
while Constraints violated do
    for $i = 1, \dots, \lambda$ do                 ▷ for all offspring individuals
        for $j = 1, \dots, m$ do                         ▷ for all constraints
            if $\alpha_j(x_i)$ then
                $v_j \leftarrow (1 - c_v) v_j + c_v y_i$
                $C \leftarrow C - \frac{\beta}{\alpha_0(x_i)} \frac{v_j v_j^\top}{\|v_j\|^2}$
            end if
        end for
    end for
    for $i = 1, \dots, \lambda$ do                 ▷ for all offspring individuals
        for $j = 1, \dots, m$ do                         ▷ for all constraints
            if $\alpha(x_i)$ then
                Resample offspring $x_i$
            end if
        end for
    end for
end while
```

The introduced strategy for handling constraints works only if the initial mean of the generation is located in the feasible region. However, the search for an initial mean vector can be difficult especially when the problem dimension is high. An unguided random search suffers from the curse of dimensionality leading to high rejection rates and hence the algorithm stalls. A remedy to this problem has been proposed by Maesani and Floreano [10] and referred as viability boundaries. The $(1+1)$-Vie-CMA-ES is an extension of $1+1$-Constrained-CMA-ES that benefits from an initially relaxed constraint boundary instead of the original tight boundary. As the offspring is guided towards the optimum, the boundary is contracted until it matches the original constraint boundary. Generalizing the ideas presented to (μ, λ)-CMA we initialize the relaxed boundary as

$$\begin{aligned} \boldsymbol{b} = [&\max\{0, h_1(\boldsymbol{x}_1), \dots, h_1(\boldsymbol{x}_\lambda)\}, \dots, \\ &\max\{0, h_m(\boldsymbol{x}_1), \dots, h_m(\boldsymbol{x}_\lambda)\}], \end{aligned}$$

As before we denote $\boldsymbol{x}_i$ as the location of the offsprings with $i \in \{1, \dots, \lambda\}$. Intuitively it follows that each constraint gets replaced by a relaxed boundary that follows the sample which violates said constraint the most. The relaxed boundary is illustrated in fig. 4c. If no individual violates the relaxed boundary we update the relaxed boundary using

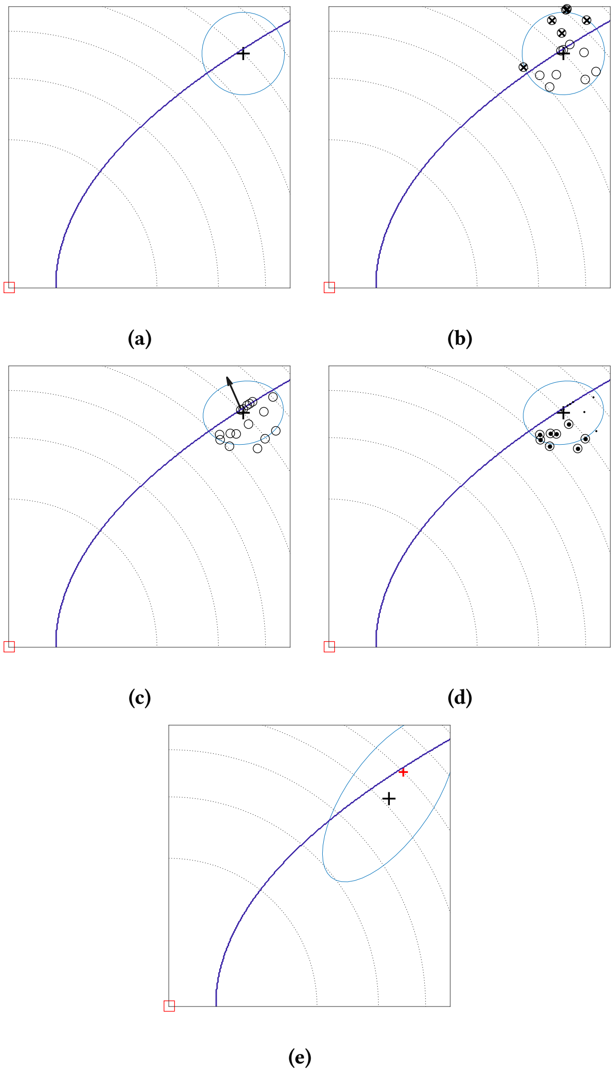

(a)
(b)
(c)
(d)
(e)

Figure 3: Constrained CMA-ES steps for the first generation with the same setup as in fig. 1 and an added constraint function $h(y) = y^2/6.5 + 1 < x$. Figure 3b shows a population of 16 individuals sampled from the initial distribution. The 6 individuals that violate the boundary are marked. Figure 3c displays the added step for constraining: The constraint normal approximation (black arrow), resampled population that is inside the feasible region and the adapted distribution equiprobability line.

$$\begin{aligned} \boldsymbol{b} = [&\max\left\{0, \min\left\{b_1, h_1(\boldsymbol{x}_{\mathrm{c},1}) + \frac{b_1 - h_1(\boldsymbol{x}_{\mathrm{c},1})}{2}\right\}\right\}, \dots, \\ &\max\left\{0, \min\left\{b_m, h_m(\boldsymbol{x}_{\mathrm{c},m}) + \frac{b_m - h_m(\boldsymbol{x}_{\mathrm{c},m})}{2}\right\}\right\}], \end{aligned}$$

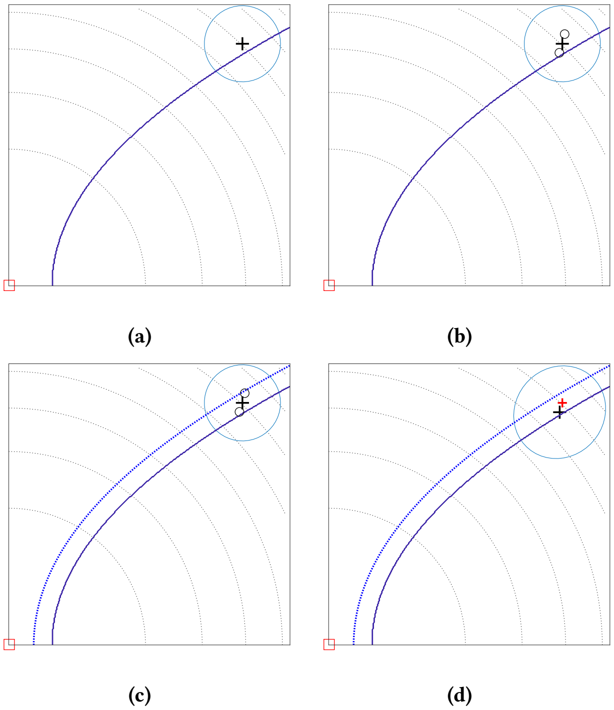


Figure 4: CCMA-ES steps for the first generation with the same setup as in previous figures. The sampling step in fig. 4b includes 2 individuals with mean outside the constraint boundary (solid black line). The viability boundary (dashed blue line) in fig. 4c is set to the point which most violates the true boundary. Figure 4d shows the updated mean with respect to the old (red) one.

where $\boldsymbol{x}_{c,j}$ is the sample closest to the relevant boundary j with respect to the Euclidean distance. Once the viability boundary matches the constraints no further updates on the boundary are performed and the algorithm continues to function as the previously proposed constraint (μ, λ)-CMA-ES.

The viability extension suffers from a significant drawback; if we choose a large λ we increase the probability of samples far from the constraint boundary. Since we adapt the relaxed boundary to the worst sample, it may happen that the mean of the generation moves away from the feasible region. As a result the constraint boundary may not be reached by the samples. This premature convergence is shown in fig. 5.

Premature convergence can be alleviated by:

(1) choosing an initial population mean inside the constraint boundary (or close to it),
(2) reducing population size λ.

The first idea is not applicable since we do not know yet any points inside the viable domain. The second idea, to reduce λ, is undesirable as it increases the probability of missing the global minimum and converging towards a local minimum. A possible solution is to alternate the population size and choose λ to be small when the population mean is outside the constraint boundary and sample a large population λ if the mean complies with the constraints. Our empirical tests showed that best results have been achieved when $\lambda = 2$ if the mean violates the constraints and $\lambda = 1.5(4 + 3\ln(n))$ if the mean resides within the feasible region. This approach has been implemented and its performance is evaluated in the next section. The other parameters used in our implementation can be found in table 1.

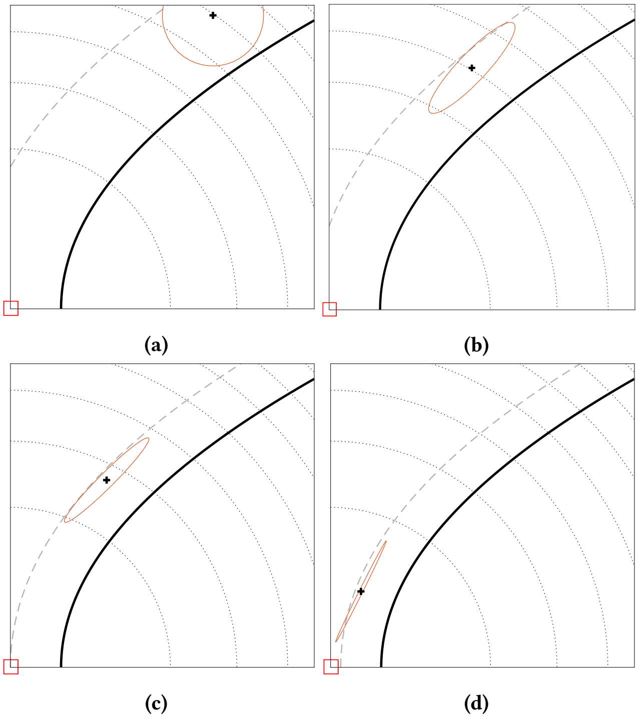


Figure 5: Idealized case of premature convergence for viability–CMA-ES.

4 EVALUATION

The evaluation of CCMA-ES consists of two parts:

(1) Test problems with known solutions [1, 9].
(2) A pharmacodynamics model [12], where the true solution is not know.

For both cases the results are compared to a state of the art optimization strategy – the memetic viability evolution CMA-ES (mViE) [11]. We use the Matlab implementation of the algorithm that accompanies the publication. For the CEC 2006 test problems we set the stopping criteria such that the algorithm terminates if $f(\boldsymbol{x}) - f(\boldsymbol{x}^*) \leq 0.0001$, where $\boldsymbol{x}^*$ is the known global minimum.

4.1 2006 CEC Test Problems

In our numerical experiment we use a subset of the test problems from the 2006 IEEE International Conference on Evolutionary Computation (2006 CEC). The 2006 CEC problem set includes equality, inequality and mixed constraints out of which only the problems with solely inequality constraints are chosen. We note that for test

λ	$4 + 3\ln(n)$ or $1.5(4 + 3\ln(n))$ when constrained
μ	$\frac{\lambda}{2}$
w_i	$\frac{w'_i}{\sum_{j=1}^{\mu} w'_j}$
w'_i	$\ln\frac{\lambda+1}{2i}, \quad i = 1, \ldots, \mu$
μ_{eff}	$\frac{1}{\sum_{i=1}^{\mu} w_i^2}$
c_σ	$\frac{\mu_{\text{eff}}+2}{n+\mu_{\text{eff}}+5}$ or $\frac{\sqrt{\mu_{\text{eff}}}}{\sqrt{\mu_{\text{eff}}}+\sqrt{n}}$ when constrained
d_σ	$1 + 2\max(0, \sqrt{\frac{\mu_{\text{eff}}-1}{n+1}} - 1) + c_\sigma$
c_c	$\frac{4+\mu_{\text{eff}}/n}{4+n+2\mu_{\text{eff}}/n}$
c_1	$\frac{2}{\mu_{\text{eff}}+(n+1.3)^2}$
c_μ	$\min(1 - c_1, 2\frac{\mu_{\text{eff}}-2+1/\mu_{\text{eff}}}{(n+2)^2+\mu_{\text{eff}}})$
β	$\frac{0.1}{n+2}$
c_v	$\frac{1}{n+2}$

Table 1: Parameters for the CCMA-ES

Prob.	Best	Median	Worst	Mean	Std
g04	3049	4030	13226	5313	2880
g06	909	1402	38480	3061	6237
g07	5375	6258	16418	6729	1902
g08	192	425	2831	735	609
g09	2553	3038	8280	3656	1410
g10	8253	10944	24710	12378	3963
g16	1451	2938	7303	3086	1320
g19	19430	23506	47838	24459	4641
g24	597	807	9998	1342	1658

Table 2: Best, median, worst, mean and standard deviation of objective function evaluations for the benchmark functions for mVie.

Prob.	Best	Median	Worst	Mean	Std
g04	1833	2080	2418	2099	117
g06	525	632	792	634	46
g07	2377	2658	3251	2686	148
g08	63	183	270	181	37
g09	399	556	758	564	66
g10	2703	3510	4956	3532	381
g16	894	1554	2736	1573	317
g19	18288	30180	69060	31979	10011
g24	255	408	501	395	43

Table 3: Best, median, worst, mean and standard deviation of objective function evaluations for the benchmark functions for CCMA-ES.

functions g01, g02 and g18 CCMA-ES did not achieve a 100% success probability of finding the global minimum. The reason for this remains a subject of ongoing research. Finally, g12 is also excluded since the viable domain is the union of disjoint sets and CCMA-ES works only on connected domains.

All the problems were linearly transformed such that the domain of f is $[0, 1]^n$. This is done because the CMA-ES algorithm is know to exhibit a subpar performance when the problem is stretched in at least one of the dimensions. Furthermore, for a given box boundary the suggested initial step-size σ^0 is 30% of boundary length [4]. This allowed us to use the same initial step-size for all problems.

The only input parameter needed for the CCMA-ES algorithm is an initial population mean $\boldsymbol{m}^0$ which was uniformly chosen from the domain. In order to account for the randomness each test problem was solved a 100 times. All other parameters of the algorithm remained fixed throughout the verification and validation process. For comparison of the algorithms the number of objective and constraint function evaluations were counted and recorded.

In table 2 and table 3 we summarize the performance of mVie and CCMA-ES on the test problem set whereas in fig. 6a the ratio of the median of the objective function evaluations of mViE over CCMA-ES is presented. In all the cases, except g19, CCMA-ES required less objective function evaluations to reach the minimum. In fig. 6b the same ratio for constraint function evaluations is presented. CCMA-ES required more constraint evaluations in most of the test problems. This feature makes the proposed algorithm best suited for problems where the evaluation of constraints is computationally cheap compared to the evaluation of the objective function. In the next section we discuss an example with this property.

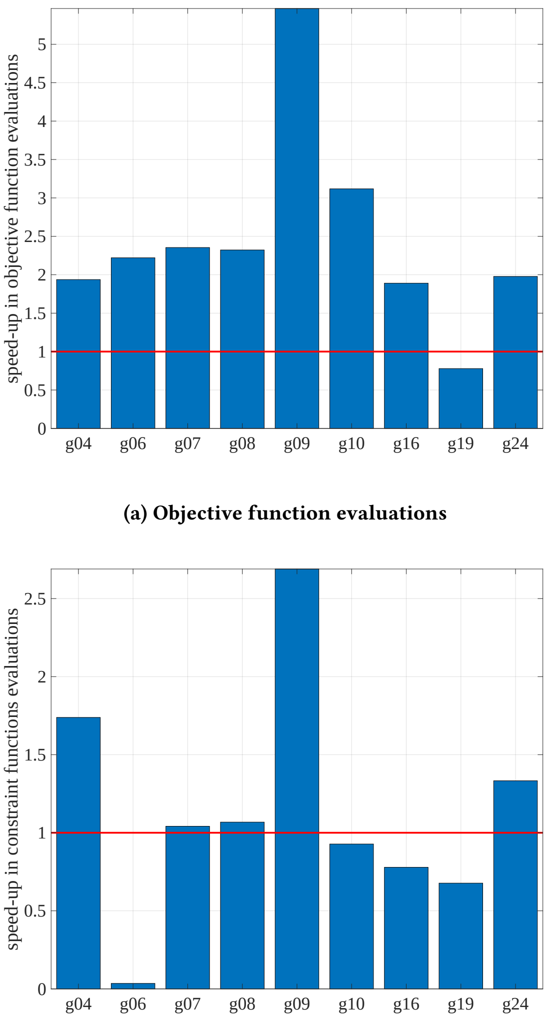
4.2 Pharmacodynamics for Tumor Growth

The proposed algorithm is applied to the the tumor growth inhibition model by Ribba [12]. The model has been proposed in order to predict the growth of diffuse low-grade gliomas in adults, under radiation or chemotherapy. It is described by the system of ordinary differential equations,

$$\begin{aligned}
\frac{\mathrm{d}C}{\mathrm{d}t} &= -\vartheta_1 C \\
\frac{\mathrm{d}P}{\mathrm{d}t} &= \vartheta_4 P(1 - \frac{P + Q + Q_P}{K}) + \vartheta_5 Q_P - \vartheta_3 P - \vartheta_1 \vartheta_2 CP \\
\frac{\mathrm{d}Q}{\mathrm{d}t} &= \vartheta_3 P - \vartheta_1 \vartheta_2 CQ \\
\frac{\mathrm{d}Q_P}{\mathrm{d}t} &= \vartheta_1 \vartheta_2 CQ - \vartheta_5 Q_P - \vartheta_6 Q_P \,,
\end{aligned} \tag{6}$$

with initial conditions

$$C(0) = 0, \quad P(0) = \vartheta_7, \quad Q(0) = \vartheta_8, \quad Q_P(0) = 0. \tag{7}$$



(a) Objective function evaluations

(b) Constraint function evaluations

Figure 6: Ratio of the median of function evaluations for mViE over CCMA-ES.

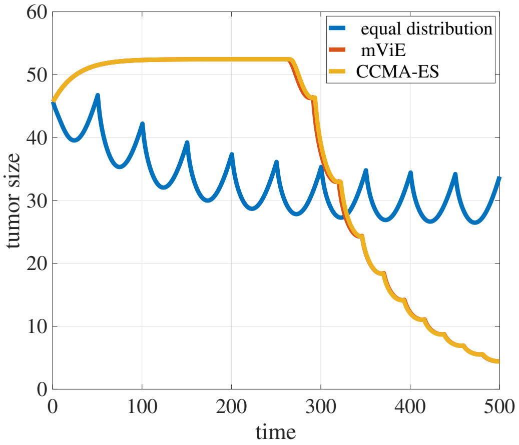


Figure 7: Evolution of the tumor size for the non-optimized treatment schedule and the optimized schedule given by mViE and CCMA-ES. In this case both optimization algorithms converge to the same solution.

The volume of the tumor, $P^* = P + Q + Q_P$, consists of proliferative cells (P), quiescent cells (Q) and damaged quiescent cells (Q_P). The drug used for treatment (C) directly eliminates proliferative cells while damaging the quiescent cells which in turn can repair themselves and turn into proliferative cells or die. The parameters $\vartheta_1, \ldots, \vartheta_8$ correspond to the following parameters: $\vartheta_1 = KDE$, the rate of constant decay of the drug concentration (C), $\vartheta_2 = \gamma$, linear rate of damages to both proliferative tissue (P) and quiescent tissue (Q), $\vartheta_3 = k_{PQ}$, the rate of constant transition from proliferative tissue (P) to quiescent tissue (Q), $\vartheta_4 = \lambda_p$, the rate of constant growth of the proliferative tissue (P), $\vartheta_5 = k_{Q_pP}$, the rate of constant transfer from damaged quiescent tissue (Q_p) to proliferative tissue (P), $\vartheta_6 = \delta_{Q_p}$, the rate of constant elimination of the damaged quiescent tissue (Q_p), $\vartheta_7 = P_0$, initial amount of proliferative tissue (P), $\vartheta_8 = Q_0$, initial amount of quiescent tissue (Q).

The parameter K denotes the largest tumor size ($K = 100$ mm) which also conforms with largest low-grade glioma tumors observed in patients. At time t_j, $j \in \{1, \ldots, n_q\}$ the drug at a dosage $a_j = a(t_j)$ is administered to the patient and the equations get reinitialized by adding a_j to the current drug concentration (C).

The tumor growth inhibition model is complemented by the following set of constraints introduced by Harrold [7]: The maximum amount of drug that can be administered per injection

$$h_j = a_j - 1 \leq 0, \quad j \in \{1, \ldots, n_q\}, \tag{8}$$

where n_q is the total number of injections and t_j is the time of a particular injection. The next two constraints are related to the drug toxicity: instantaneous and cumulative drug concentration. Violating those constraints would harm a patient. The maximal instantaneous drug concentration takes into account the lethal dose

$$h_{\max} = \max_t C(t) - v_{\max} \leq 0. \tag{9}$$

In accordance, the parameter $v_{\max}$ is the highest drug concentration a human body can be exposed to. The constraint on cumulative drug concentration reflects the rate at which the drag can be absorbed by the body

$$h_c = \int_0^{t_{\text{end}}} C(t)dt - v_{\text{cum}} \leq 0. \tag{10}$$

The introduced parameter v_{cum} corresponds to the overall amount of drug the body can be exposed to over a certain time frame $t \in [0, t_{\text{end}}]$. Notice that $C(t)$ can be explicitly calculated from eq. (6).

The goal in a treatment is to minimize the tumor size P^* at a given time in the future by finding optimal values for the parameter vector,

$$\boldsymbol{x} = (t_1, \ldots, t_{n_q}, a_1, \ldots, a_{n_q}). \tag{11}$$

The dimension of the parameter space in the optimization problem is $2n_q$.

We performed the experiments with $n_q = 4$ and $n_q = 10$. The tumor size $P^*(t_{\text{end}})$ is calculated using a black-box simulator integrating the system of differential equations from eq. (6) up to t_{end}. Notice that the derivatives of the final tumor size with respect to $t_1, \ldots, t_{n_q}$ are not readily available and thus gradient based optimization cannot be applied in the case. The parameters used for the system of eq. (6) were set to,

$$\boldsymbol{\vartheta} = [0.045, 4.52, 0.09, 0.11, 0.04, 0.00001, 0.09, 1]. \tag{12}$$

The maximal instantaneous drug concentration has been set to $v_{\max} = 1.1$ for both cases, whereas the limit for the cumulative drug amount in the patient has been chosen as $v_{\text{cum}} = 65$ and $v_{\text{cum}} = 196$ for each problem, respectively. The final times $t_{\text{end}} = 200$ and $t_{\text{end}} = 400$ were chosen such that $v_{\max}$ remains constant whilst increasing the number of equally spaced injections of a drug amount $a_j = 1$. Finally, the cumulative drug concentration has been calculated to ensure the satisfaction of the constraint described in eq. (10). In order to compare the performance of the two algorithms we conducted 100 runs with each algorithm. The initial point for both optimization algorithms has been set to equally spaced administration times t_j with dosages $a_j = 0.8$ which corresponds to 72.7% of $v_{\max}$.

The tumor size after optimization with $n_q = 4$ and $n_q = 10$ injections is presented in fig. 8 and fig. 9. We conclude that CCMA-ES converges towards an optimum using less objective function evaluations. The number of constraint function evaluations is slightly greater for CCMA-ES but remains comparable in both experiments.

The shaded areas in fig. 8 and fig. 9 correspond to one standard deviation. It is evident that for both $n_q = 4$ and $n_q = 10$ CCMA-ES can find better solutions than mVie.

The mean tumor size evolution resulting from the optimization with $n_q = 10$ injections is presented in fig. 7 together with an equally spaced injection treatment plan as a reference. Both algorithms find a minimum that corresponds to a rather similar treatment schedule and results in a much smaller tumor size compared to the unoptimized case. We note that while both algorithms perform a similar number of constraint function evaluations, CCMA-ES outperforms mViE in objective function evaluations.

5 CONCLUSIONS

In this article we present CCMA-ES, an extension of the (μ, λ)-CMA-ES algorithm, that is able to effectively handle inequality constraints. The algorithm combines both the idea of of viability boundaries and the adaptation of the covariance matrix to the boundary given by the constraints. We demonstrate that our algorithm outperforms the state of the art mViE algorithm in terms of number of objective function evaluations but not on number of constraint function evaluations. This property makes CCMA-ES best suited for problems with computationally demanding objective functions and constraints that are computationally cheap to evaluate.

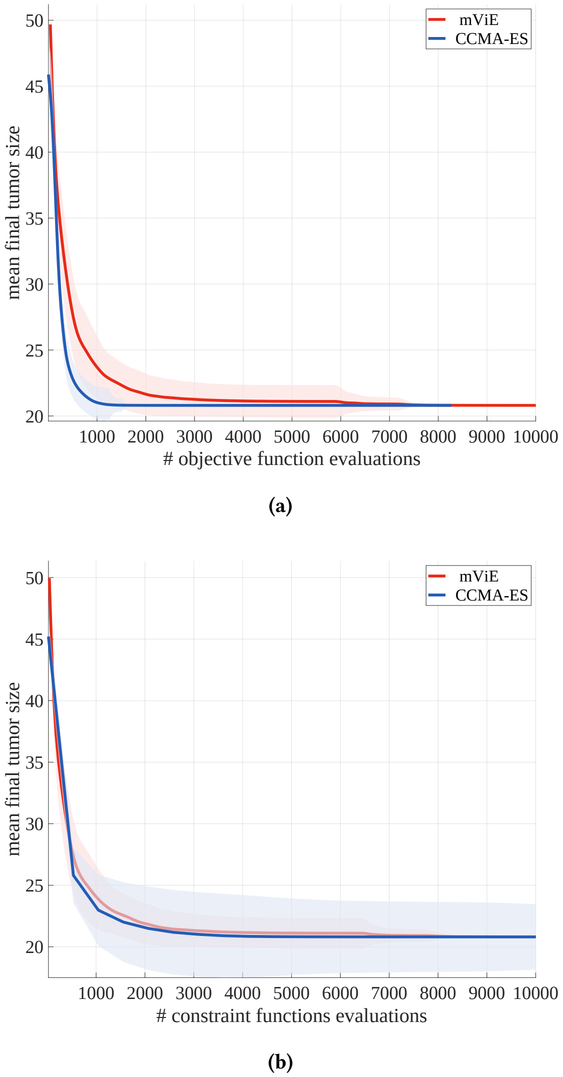
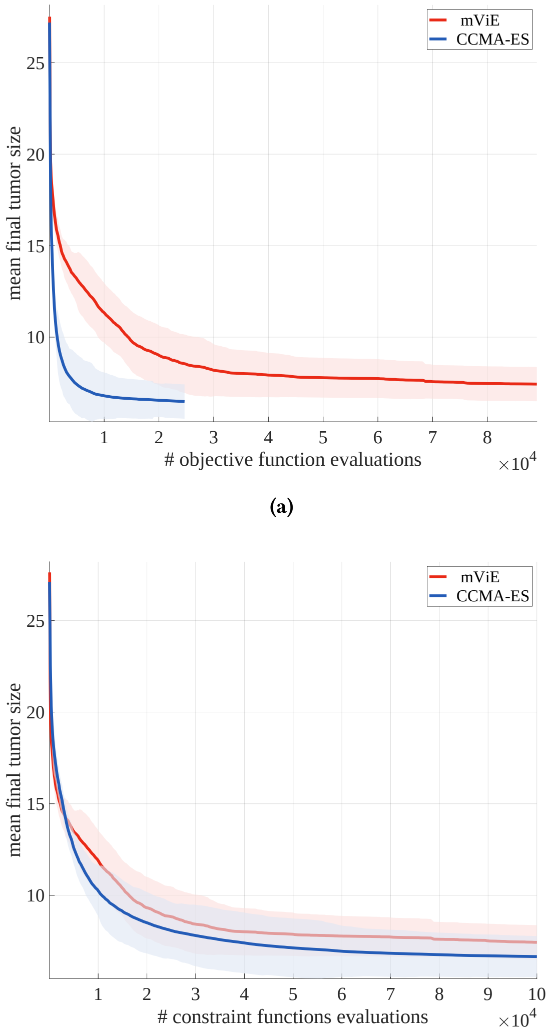


Figure 8: Mean final tumor size versus number of function evaluations for 4 injections and 100 independent optimization runs. Figure 8a shows the number of objective function evaluations, and fig. 8b shows the number of constraint function evaluation. The shaded regions correspond to one standard deviation.

6 ACKNOWELDEGMENTS

G.A. acknowledges financial support from Collegium Helveticum, ETH, Zurich. P.K. acknowledges support by the European Research Council Advanced Investigator Award 341117. The authors acknowledge the computational time at Swiss National Supercomputing Center (CSCS) under the project s659.



(a)

(b)

Figure 9: Mean final tumor size versus number of function evaluations for 10 injections over 100 independent optimization runs. Figure 9a shows the number of object function evaluations, where fig. 9b the number of constraint function evaluation. The shaded regions correspond to one standard deviation.

REFERENCES

[1] Dirk V. Arnold and Nikolaus Hansen. 2012. A (1+1)-CMA-ES for Constrained Optimisation. In *Proceedings of the 14th Annual Conference on Genetic and Evolutionary Computation (GECCO '12)*. ACM, New York, NY, USA, 297–304.

[2] Y. Diouane, S. Gratton, and L. N. Vicente. 2012. *Globally convergent evolution strategies and CMA-ES.* Technical Report.

[3] Collange Guillaume, Delattre Nathalie, Hansen Nikolaus, Quinquis Isabelle, and Schoenauer Marc. 2013. *Multidisciplinary Optimization in the Design of Future Space Launchers.* Wiley-Blackwell, Chapter 12, 459–468.

[4] Nikolaus Hansen. 2016. The CMA Evolution Strategy: A Tutorial. *CoRR* abs/1604.00772 (2016). arXiv:1604.00772

[5] Nikolaus Hansen, Sibylle D. Müller, and Petros Koumoutsakos. 2003. Reducing the Time Complexity of the Derandomized Evolution Strategy with Covariance Matrix Adaptation (CMA-ES). *Evolutionary Computation* 11, 1 (2003), 1–18.

[6] Nikolaus Hansen and Andreas Ostermeier. 2001. Completely Derandomized Self-Adaptation in Evolution Strategies. *Evolutionary Computation* 9, 2 (2001), 159–195.

[7] John M. Harrold and Robert S. Parker. 2009. Clinically relevant cancer chemotherapy dose scheduling via mixed-integer optimization. *Computers and Chemical Engineering* 33, 12 (2009), 2042 – 2054. https://doi.org/10.1016/j.compchemeng.2009.06.005 FOCAPO 2008 – Selected Papers from the Fifth International Conference on Foundations of Computer-Aided Process Operations.

[8] Christian Igel, Thorsten Suttorp, and Nikolaus Hansen. 2006. 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 '06)*. ACM, New York, NY, USA, 453–460.

[9] Jing Liang, Thomas Runarsson, Efrén Mezura-Montes, M Clerc, Ponnuthurai Suganthan, C A. C. Coello, and Kalyan Deb. 2006. *Problem definitions and evaluation criteria for the CEC 2006 special session on constrained real-parameter optimization.* Technical Report 41. Nangyang Technological University, Singapore.

[10] Andrea Maesani and Dario Floreano. 2014. Viability Principles for Constrained Optimization Using a (1+1)-CMA-ES. In *Parallel Problem Solving from Nature – PPSN XIII.* Springer International Publishing, Cham, 272–281.

[11] A. Maesani, G. Iacca, and D. Floreano. 2016. Memetic Viability Evolution for Constrained Optimization. *IEEE Transactions on Evolutionary Computation* 20, 1 (2016), 125–144.

[12] Benjamin Ribba, Gentian Kaloshi, Mathieu Peyre, Damien Ricard, Vincent Calvez, Michel Tod, Branka Čajavec-Bernard, Ahmed Idbaih, Dimitri Psimaras, Linda Dainese, Johan Pallud, Stéphanie Cartalat-Carel, Jean-Yves Delattre, Jérôme Honnorat, Emmanuel Grenier, and François Ducray. 2012. A Tumor Growth Inhibition Model for Low-Grade Glioma Treated with Chemotherapy or Radiotherapy. *Clinical Cancer Research* 18, 18 (2012), 5071–5080. https://doi.org/10.1158/1078-0432.CCR-12-0084

[13] T. P. Runarsson and Xin Yao. 2000. Stochastic ranking for constrained evolutionary optimization. *IEEE Transactions on Evolutionary Computation* 4, 3 (Sep 2000), 284–294. https://doi.org/10.1109/4235.873238