Model assisted (1+1)ES

Arash Kayhani

MAY 2017

1 Introduction

In my experiments I followed the model-assisted algorithm suggested in [1] with some changes in the evolutionary part and approximation approach.

1/5 **rule** Instead of the median selection with CMA-ES that is used in [1], my algorithm evolves using the 1/5 rule explained in[2].

Local training The surrogate algorithm in [1] only learns from the last 30 training data available for training, because of this the model error decrease as the evolutionary algorithm converges to the optimum point.However, my experiments show that increasing the number of training data makes the model error decrease faster for more complex problems.

RBF Network In this paper we are using the RBF network suggested in [1] which contains 20 hidden nodes, and a k-means core.

Experiments Case studies are evaluated by three factors:

1.Original fitness First case study is a simple Sphere function $Y = X^2$.In this case study the minimum fitness that the algorithms can show in python is $10^{-}323 * 10^{-}323 = 10^{-}646$.

Second case study is a sphere function shifted for one unit to the left or right side $Y = (X - 1)^2$.In this case study the minimum fitness that the algorithms can show in python is $10^{-}15 * 10^{-}15 = 10^{-}30$.

2.Model fitness error In all experiments I measured the absolute fitness error of the new generated population to evaluate if the model is accurate enough to select the best individual in the population:

$Error = |Fitness(ind) - Model(ind)|$

3.Original fitness slope Finding the absolute slope between the starting point and the best evaluated fitness can also be beneficial to describe the behaviour of the algorithms.

$Slope_{generation=N} = (log(Bestfitness_{generation=0}) - log(Bestfitness_{generation=N}))/N$

2 Algorithms

2.1 Algorithm 1

In the first two lines of the algorithm, a RBF network with a K-means core is initialized as was suggested in[1].In lines 5-12,the algorithm uses the σ calculated in the previous cycle to discover random points in sigma-neighbourhood of the current point, and then selects the best point via the best available approximated model of the fitness function.

In lines 14-18, the algorithm adds a new individual into the training population of the RBF model only if its real fitness is better than any other fitness available.(the training population only stores the last 30 individuals)

1: $N = numberofdimensions$
2: $initialize(individual, \sigma)$
3: $Model \leftarrow (RBFNetwork(Inputs = N, Hidden = 10, Output = 1))$
4: **for** $i \leq Originalgeneration$ **do**
5:     $individual2 \leftarrow individual$
6:     **for** $j \leq Modelgeneration$ **do**
7:         $individual3 \leftarrow individual2 + \sigma * I$
8:         $fit2 \leftarrow Modelfitness(individual2)$
9:         $fit3 \leftarrow Modelfitness(individual3)$
10:         **if** $fit3 \leq fit2$ **then**
11:             $individual2 \leftarrow individual3$
12:         **end if**
13:     **end for**
14:     $Newfitness = originalfitness(individual2)$
15:     $printerror \leftarrow \frac{(Newfitness - Model(individual2))}{Originalgeneration}$
16:     $\sigma \leftarrow \sigma * \exp^{\frac{1}{N}}((Newfitness \leq Bestfitness) - \frac{1}{5})$
17:     **if** $Newfitness \leq Bestfitness$ **then**
18:         $individual2 \leftarrow individual$
19:         $Bestfitness \leftarrow Newfitness$
20:         $trainingpop.append(individual, Bestfitness)$
21:         $Model.train(trainingpop)$
22:     **end if**
23: **end for**
24: $printfitness \leftarrow Bestfitness$

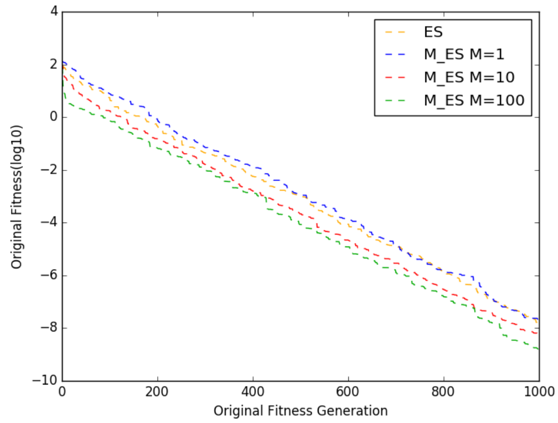

Figure 1: simple(1+1) ES fitness compared to M-ES(model assisted ES) using three different number of model generation(M) per original fitness evaluation on 10D $Y = (X)^2$ (5 trials)

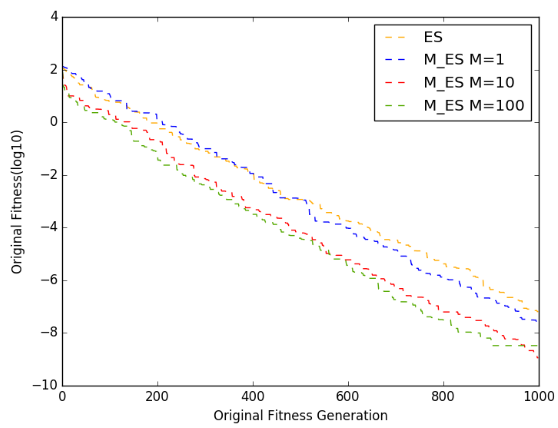

Figure 2: simple ES fitness compared to M-ES(model assisted ES) using three different number of model generation(M) per original fitness evaluation on 10D $Y = (X - 1)^2$ (5 trials)

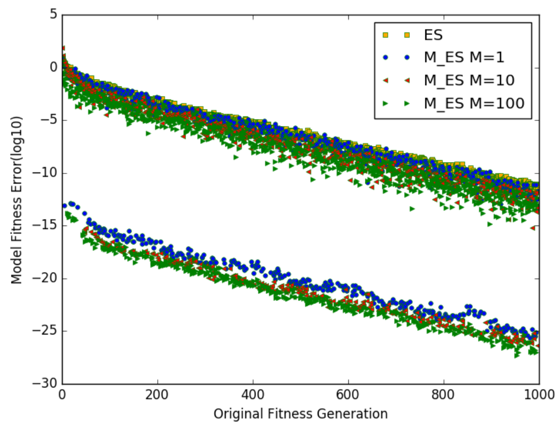

Figure 3: simple ES error compared to M-ES(model assisted ES) using three different number of model generation(M) per original fitness evaluation on 10D $Y = (X)^2$ (5 trials)

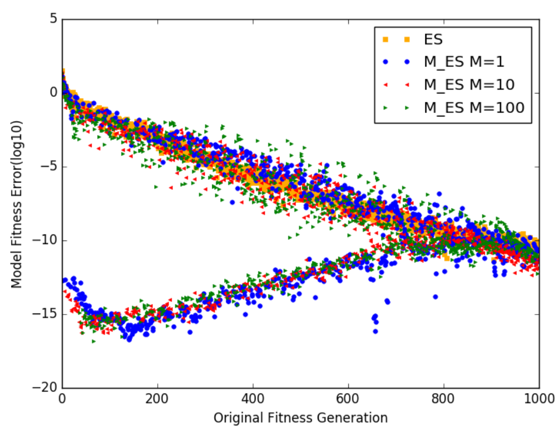

Figure 4: simple ES error compared to M-ES(model assisted ES) using three different number of model generation(M) per original fitness evaluation on 10D $Y = (X-1)^2$ (5 trials)

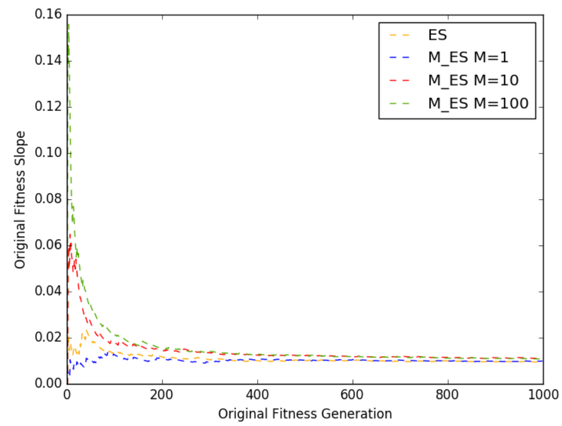

Figure 5: simple ES slope compared to M-ES(model assisted ES) using three different number of model generation(M) per original fitness evaluation on 10D $Y = (X)^2$ (5 trials)

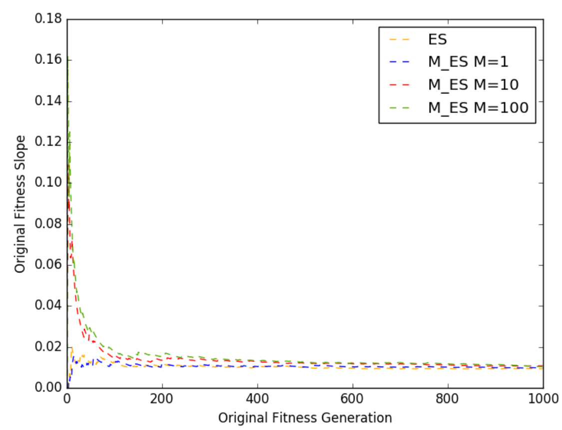

Figure 6: simple ES slope compared to M-ES(model assisted ES) using three different number of model generation(M) per original fitness evaluation on 10D $Y = (X - 1)^2$ (5 trials)

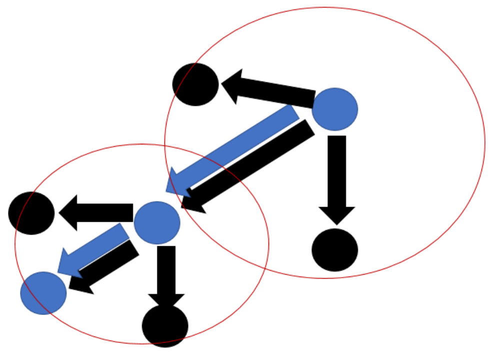

Figure 7: Shows how A1 algorithm works.Black arrow are the model evaluations in the sigma neighbourhood and the blue arrows are the real fitness evaluations on the best points selected by the model evolution

Looking at the results generated at figures 1 -6 we can say that the model-assisted algorithms helps the evolutionary algorithm by choosing the best point according to the approximated function.In Figure 4 however it can be seen that the fitness error fails to decrease the error after$10^{-}15$ probably because that's the minimum value that python can evaluate.

2.2 Algorithm 2

In algorithm 1 Model only finds the best point in the sigma neighbourhood which increases the convergence speed of the ES in early iterations where sigma is big(figure 5 and 6).However, after 200 iteration sigma is so small that having more approximated points can not make ES much faster than what it already is.Therefore, in the second algorithm in line 13 I added a sigma update function.In A2 algorithms ES evolves for a number of iteration on the approximated model before original fitness evaluation.

1: $N = numberofdimensions$
2: $initialize(individual, \sigma)$
3: $Model \leftarrow (RBFNetwork(Inputs = N, Hidden = 10, Output = 1))$
4: **for** $i \leq Originalgeneration$ **do**
5:     $individual2 \leftarrow individual$
6:     **for** $j \leq Modelgeneration$ **do**
7:         $individual3 \leftarrow individual2 + \sigma * I$
8:         $fit2 \leftarrow Modelfitness(individual2)$
9:         $fit3 \leftarrow Modelfitness(individual3)$
10:         **if** $fit3 \leq fit2$ **then**
11:             $individual2 \leftarrow individual3$
12:         **end if**
13:         $\sigma \leftarrow \sigma * \exp^{\frac{1}{N}}((Newfitness \leq Bestfitness) - \frac{1}{5})$
14:     **end for**
15:     $Newfitness = originalfitness(individual2)$
16:     $printerror \leftarrow \frac{(Newfitness - Model(individual2))}{Originalgeneration}$
17:     **if** $Newfitness \leq Bestfitness$ **then**
18:         $individual2 \leftarrow individual$
19:         $Bestfitness \leftarrow Newfitness$
20:         $trainingpop.append(individual, Bestfitness)$
21:         $Model.train(trainingpop)$
22:     **end if**
23: **end for**
24: $printfitness \leftarrow Bestfitness$

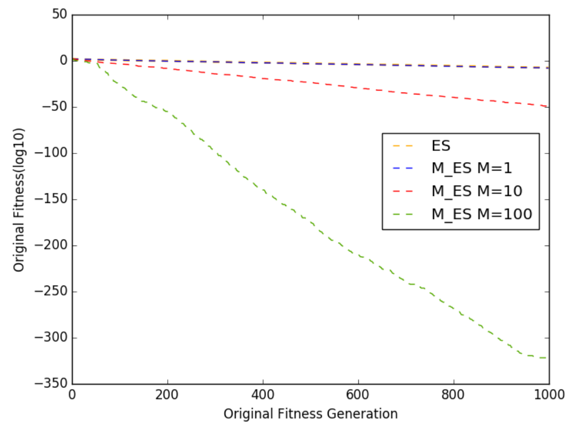

Figure 8: simple(1+1) ES fitness compared to M-ES(model assisted ES) using three different number of model generation(M) per original fitness evaluation on 10D $Y = (X)^2$ (5 trials)

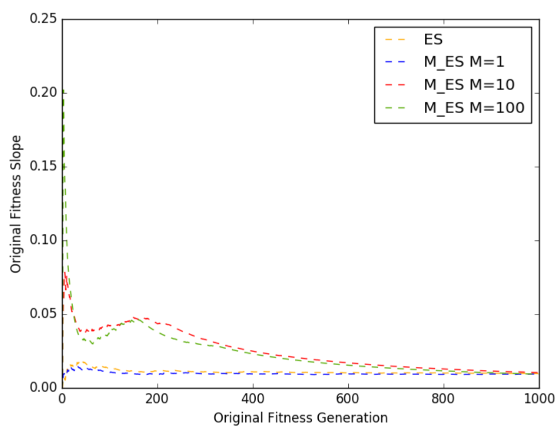

Figure 9: simple ES fitness compared to M-ES(model assisted ES) using three different number of model generation(M) per original fitness evaluation on 10D $Y = (X-1)^2$ (5 trials)

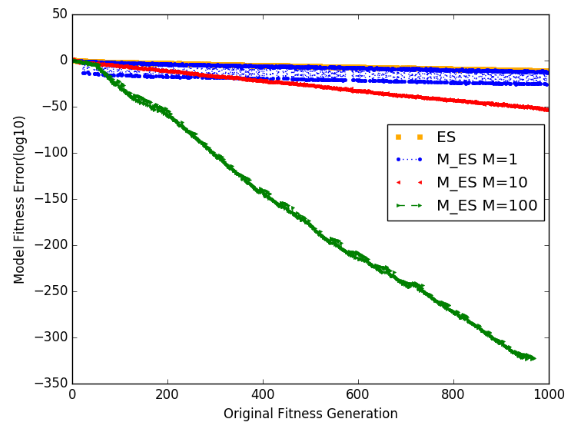

Figure 10: simple ES error compared to M-ES(model assisted ES) using three different number of model generation(M) per original fitness evaluation on 10D $Y = (X)^2$ (5 trials)

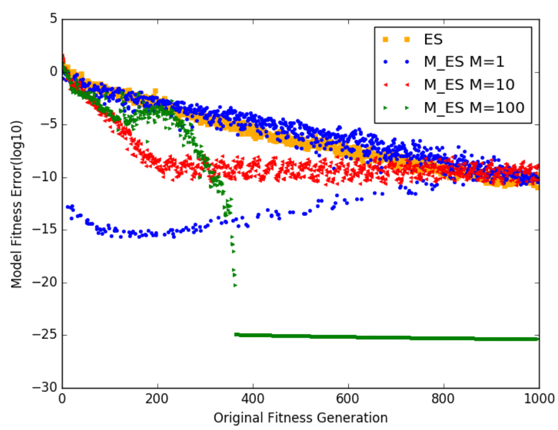

Figure 11: simple ES error compared to M-ES(model assisted ES) using three different number of model generation(M) per original fitness evaluation on 10D $Y = (X-1)^2$ (5 trials)

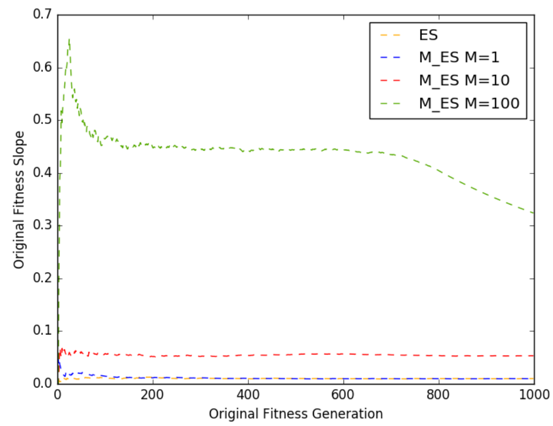

Figure 12: simple ES slope compared to M-ES(model assisted ES) using three different number of model generation(M) per original fitness evaluation on 10D $Y = (X)^2$ (5 trials)

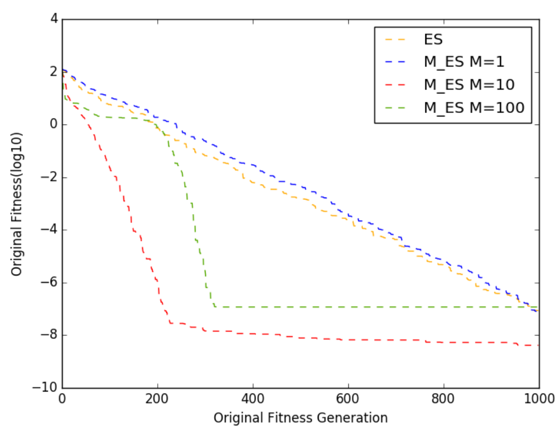

Figure 13: simple ES slope compared to M-ES(model assisted ES) using three different number of model generation(M) per original fitness evaluation on 10D $Y = (X-1)^2$ (5 trials)

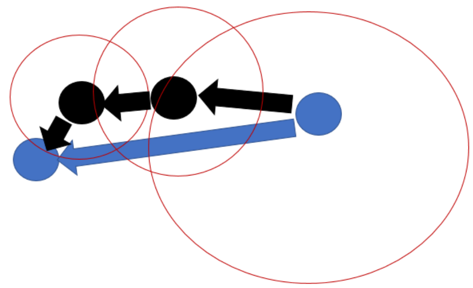

Figure 14: Shows how A2 algorithm works.Black arrow are the model evaluations in the sigma neighbourhood and the blue arrows are the real fitness evaluations on the best points selected by the model evolution

References

[1] Ulmer, H., Streichert, F. and Zell, A., 2003, July. Model-assisted steady-state evolution strategies. In Genetic and Evolutionary Computation Conference (pp. 610-621). Springer Berlin Heidelberg.

[2] Hansen, N., Arnold, D.V. and Auger, A., 2015. Evolution strategies. In Springer handbook of computational intelligence (pp. 871-898). Springer Berlin Heidelberg.