**A Dual Center Approach to CMA-ES**

by

Dillon Bourne

A Thesis

Presented to

The University of Guelph

In partial fulfillment of requirements

For the degree of

Master of Science

In

Computer Science

Guelph, Ontario, Canada

© Dillon Bourne, December 2019

**ABSTRACT**

A DUAL CENTER APPROACH TO CMA-ES

Dillon Bourne Advisor:

University of Guelph, 2019 Dr. Mark Wineberg

Covariance Matrix Adaptation - Evolution Strategies (CMA-ES) is a renowned state-of-the-art black-box optimzation algorithm in the field of Evolutionary Computation. As real-world optimzation problems began to be characterized as multimodal, CMA-ES was subject to modifications to improve it’s performance on these problems. The rise of multimodal problems presented a new challenge for optimzation algorithms which is avoiding local optima while trying to find the global optimum.

The CMA-ES algorithm, although powerful, is not guaranteed to beat this challenge as it may be sampling in an area which contains basins of attraction where the global optimum does not reside. This could be attributed to the fact that CMA-ES uses a single-model Estimation Distribution Algorithm (EDA) to determine a single point in the problem landscape from which to perform sampling.

This research investigates the performance of CMA-ES on several multimodal and unimodal problems using two different EDAs with an overlapping model to perform sampling within the problem landscape. This proposed system, Dualcenter-CMA-ES (DC-CMA-ES) outperforms IPOP-CMA-ES on complex multimodal functions, especially as problem dimensionality increases.

**Dedication**

This thesis is dedicated to God, my friends and family, my advisor, the evolutionary computation community and to my beautiful island Barbados.

**Acknowledgements**

I would like to thank my advisor Dr. Mark Wineberg for helping me to achieve what at some points seemed impossible. I would also like to thank him for his patience, guidance and most importantly his time.

I would like to thank Dr. Deborah Stacey for taking an interest in my work. I would also like to thank her for her useful feedback and kind words and the time she has taken to read and analyse my thesis.

Last but not least, I’d like to thank my fellow researchers, Tamizhselvan Venkatesan, Adefunke Akinola and Omar Irfan Khan for all of their advice and company.

Table of Contents

ABSTRACT .................................................................................................................................. ii

Dedication ..................................................................................................................................... iii

Acknowledgements ....................................................................................................................... iv

List of Tables .............................................................................................................................. viii

List of Figures ............................................................................................................................... x

List of Appendices ........................................................................................................................ xi

[Chapter 1 1](#_Toc27410698)

[Introduction 1](#_Toc27410699)

[1.1 Layout of Thesis 3](#_Toc27410700)

[Chapter 2 4](#_Toc27410701)

[Background 4](#_Toc27410702)

[2.1 (1+1)-ES 5](#_Toc27410703)

[2.1.2 (1 + λ) - ES & (1, λ) – ES 7](#_Toc27410704)

[2.1.3 (µ/ρ, κ, λ) – ES 8](#_Toc27410705)

[2.2 (µw, λ) - CMA-ES 10](#_Toc27410706)

[2.2.1 LS-CMA-ES 18](#_Toc27410707)

[2.2.2 LR-CMA-ES 19](#_Toc27410708)

[2.2.3 IPOP-CMA-ES 20](#_Toc27410709)

[2.2.4 (µ + λ) - CMA-ES 20](#_Toc27410710)

[2.2.5 (1+1)-Cholesky-CMA-ES 20](#_Toc27410711)

[2.2.6 Active-CMA-ES 22](#_Toc27410712)

[2.2.7 (µ, λ)-CMSA-ES 22](#_Toc27410713)

[2.2.8 sep-CMA-ES 22](#_Toc27410714)

[2.2.9 (1+1)-Active-CMA-ES 23](#_Toc27410715)

[2.2.10 SPO-CMA-ES 23](#_Toc27410716)

[2.3 Niching Radius Algorithms 23](#_Toc27410717)

[2.4 Common Random Numbers (CRNs) 24](#_Toc27410718)

[Chapter 3 25](#_Toc27410719)

[Methodology 25](#_Toc27410720)

[3.1 Motivation 25](#_Toc27410721)

[3.2 Functionality of DC-CMA-ES 26](#_Toc27410722)

[3.3 Pseudocode 32](#_Toc27410723)

[Chapter 4 34](#_Toc27410724)

[Experiments 34](#_Toc27410725)

[4.1 Performance Measures & Test Functions 34](#_Toc27410726)

[4.2 Experimental Setup 37](#_Toc27410727)

[4.3 Integrating CRN Simulations and Restart ES Algorithms 38](#_Toc27410728)

[Chapter 5 40](#_Toc27410729)

[Results and Discussion 40](#_Toc27410730)

[5.1 Results of Fixed Ratio Systems 41](#_Toc27410731)

[5.2 Results of Conditional Ratio Systems 47](#_Toc27410732)

[5.3 Dimensionality 53](#_Toc27410733)

[Chapter 6 62](#_Toc27410734)

[Conclusion 62](#_Toc27410735)

[6.1 Future Works 65](#_Toc27410736)

[Bibliography 66](#_Toc27410737)

[Appendix A 69](#_Toc27410738)

[1. LS-CMA-ES 69](#_Toc27410739)

[2. (1+1)-Cholesky-CMA-ES 71](#_Toc27410740)

[3. Active-CMA-ES 73](#_Toc27410741)

[4. (µ, λ) - CMSA-ES 74](#_Toc27410742)

[5. sep-CMA-ES 75](#_Toc27410743)

[6. (1+1)-Active-CMA-ES 76](#_Toc27410744)

[7. SPO-CMA-ES 79](#_Toc27410745)

[Appendix B 81](#_Toc27410746)

**List of Tables**

[Table 3.1 - Description of terms relevant to DC-CMA-ES 33](#_Toc27410598)

[Table 5.1 – Description of Fixed Sampling Ratio Systems 40](#_Toc27410599)

[Table 5.2 - Main Effects of System Performance 41](#_Toc27410600)

[Table 5.3 – System performance on Ackley 42](#_Toc27410601)

[Table 5.4 – System performance on Griewank 42](#_Toc27410602)

[Table 5.5 – System performance on Levy 43](#_Toc27410603)

[Table 5.6 – System performance on Rastrigin 44](#_Toc27410604)

[Table 5.7 – System performance on Schwefel 44](#_Toc27410605)

[Table 5.8 – System performance on Elliptical 45](#_Toc27410606)

[Table 5.9 – System performance on Rosenbrock 45](#_Toc27410607)

[Table 5.10 – System performance on Zakharov 46](#_Toc27410608)

[Table 5.11 – Main Effects of System Performance with Conditional Metrics 47](#_Toc27410609)

[Table 5.12 – System Performance on Ackley with Conditional Metrics 48](#_Toc27410610)

[Table 5.13 – System Performance on Griewank with Conditional Metrics 48](#_Toc27410611)

[Table 5.14 – System Performance on Levy with Conditional Metrics 49](#_Toc27410612)

[Table 5.15 – System Performance on Rastrigin with Conditional Metrics 49](#_Toc27410613)

[Table 5.16 – System Performance on Schwefel with Conditional Metrics 50](#_Toc27410614)

[Table 5.17 – System Performance on Elliptical with Conditional Metrics 51](#_Toc27410615)

[Table 5.18 - System Performance on Rosenbrock with Conditional Metrics 51](#_Toc27410616)

[Table 5.19 – System Performance on Zakharov with Condition Metrics 52](#_Toc27410617)

[Table 5.20 – System performance at dimension 5 53](#_Toc27410618)

[Table 5.21 – System performance at dimension 10 54](#_Toc27410619)

[Table 5.22 – System performance at dimension 25 55](#_Toc27410620)

[Table 5.23 – System performance at dimension 50 56](#_Toc27410621)

[Table 5.24 – System performance at dimension 75 57](#_Toc27410622)

[Table 5.25 – System performance at dimension 100 58](#_Toc27410623)

**List of Figures**

[Figure 2.1 – Transformations of the Multivariate Normal Distribution [7] 4](#_Toc27060381)

[Figure 2.2 – (1+1)- ES taken from [2]. 7](#_Toc27060382)

[Figure 2.3 – (µ/ρ, κ, λ) – ES taken from [2]. 8](#_Toc27060383)

[Figure 2.4 – Measuring the length of the evolution path [7]. 13](#_Toc27060384)

[Figure 2.5 – (µw, λ) - CMA-ES Algorithm [2] 17](#_Toc27060385)

[Figure 3.1 – Master-Slave Relationship between the two EDAs 24](#_Toc27060386)

[Figure 3.2 - Functionality of the elite sliding window 26](#_Toc27060387)

[Figure 4.1 – Common Random Number Simulation between DC-CMA-ES & CMA-ES 35](#_Toc27060388)

[Figure 5.1 – 2-D plot of the Levy test function [14] 52](#_Toc27060389)

[Figure 5.2 – 2-D plot of the Rosenbrock test function [14] 52](#_Toc27060390)

[Figure 5.3 – 2-D plot of the Rastrigin test function [14] 53](#_Toc27060391)

**List of Appendices**

[Appendix A 61](#_Toc27060411)

[Appendix B 73](#_Toc27060412)

# 

## Introduction

Covariance Matrix Adaption – Evolution Strategies (CMA-ES) is one of the state-of-the-art algorithms in the field of Evolutionary Computation. It was originally designed in the mid-90s by N. Hansen & A. Ostermeier [1], and praised for its performance in solving unimodal functions as well as its ability to maintain that performance with increasing problem dimensionality. However, the importance of solving multimodal problems became more prevalent as real-world optimization scenarios began to reflect multimodal problems rather than unimodal [2]. Unimodal functions have only one optimum to reach, while multimodal functions have areas with many local optima, making it more difficult for an optimization algorithm to find the global optimum.

By 2005, CMA-ES was modified to better solve multimodal problems with the addition of an *increasing population* (IPOP) technique [3]. The IPOPtechnique is applied to CMA-ES after restarts to increase the number of samples being produced (IPOP-CMA-ES). The reasoning behind IPOP stems from the realization that test functions with different characteristics are better solved with different population sizes. For example, multimodal functions are usually better solved with a large population. The idea is to start with a relatively small default population size and successively increase it after every restart. This ensures that the algorithm doesn’t perform more function evaluations than necessary. When the best population size for the problem is found, the algorithm should no longer restart and will then converge towards the optimum solution. This provides IPOP with the extra power which makes CMA-ES even better equipped for solving multimodal problems. However, there is still an issue related to local optima.

Even though IPOP improves the performance of CMA-ES on multimodal problems, it is still possible that it may be sampling in an area which contains basins of attraction where the global optimum does not reside. Attempts have been made to identify topological aspects of multimodal fitness landscapes by applying a technique known as *niching* to CMA-ES [4] [5]. Niching techniques make assumptions about the fitness landscape in question, usually to identify areas which contain multiple ‘peaks’ where solutions can be found.

Never the less, the problem of identifying and avoiding the local optima of multimodal problems remains. To counteract this problem, we observe that at the heart of CMA-ES lies a single-model Estimation Distribution Algorithm (EDA). An EDA uses a model to sample a population which is then evaluated. The probabilistic model is updated based on the evaluated population and used to sample the new population [6]. We believe that the root of the problem is the use of a single-model EDA because it samples the entire population from one origin which limits the diversity of solutions and the exploration of multiple basins of attraction.

This thesis investigates the effect of diverging from the use of a single-model EDA towards two different EDAs with overlapping models. We hypothesize that using two EDAs for sampling which share certain model components will outperform single-model EDA algorithms on multimodal landscapes. In CMA-ES, the point that the EDA produces or the point from which sampling originates is often referred to as the center. With this taken into consideration, the proposed system in the thesis is referred to as *Dual-Center* CMA-ES (DC-CMA-ES).

## 1.1 Layout of Thesis

This thesis is divided into six chapters. Chapter 1 provides a brief introduction to the field of ES and discusses the problem in question along with the objectives of the paper. Chapter 2 gives a detailed background of the relevant material needed to understand this paper. This background consists of the fundamental and state-of-the-art algorithms in Evolution Strategies, the Common Random Number technique for variance reduction in experimentation and Niching techniques that are applied to optimization algorithms. Chapter 3 provides a detailed description and pseudocode of the proposed system DC-CMA-ES. In Chapter 4, the experimental setup and test functions used are discussed. Chapter 5, illustrates and analyzes the results obtained from the experiments. Lastly, Chapter 6 summarizes the major points of this thesis and lists some future research avenues to be considered.

# 

## Background

Before diving into the world of Evolution Strategies (ES), knowledge of multivariate normal distributions and covariance matrices is extremely necessary and will be discussed beforehand to aid in better understanding of the topics ahead. A multivariate normal distribution is an n-dimensional normal distribution which can be denoted as *N(m,C)*. Here, *m* is the mean vector of length *n* (the favored point in space from which sampling of new solutions takes place) and *C* is the *n* x *n* symmetric positive-definite covariance matrix [7]. A close up of a womans face

Description automatically generated

Figure 2.1 – Transformations of the Multivariate Normal Distribution [7]

The first image on left of *Figure 2.1* shows a multivariate normal distribution with mean of 0 and standard deviation as the identity matrix. The identity matrix is responsible for the circular shape around the mean, while *σ* is a scalar which can shrink or grow the search space. The middle image shows another multivariate normal distribution this time with a standard deviation of *D2*, where *D* is a diagonal matrix. Using this diagonal matrix to transform a sample from *N(0,I)* can result in “stretching” or “flattening” of the search space. The last image on the right of *Figure 2.1* uses the covariance matrix as it’s standard deviation. Transforming *N(0,I)* using the covariance matrix can also result in stretching or flattening, but also changes the direction of the search space.

## (1+1)-ES

The (1+1) – ES was the first Evolution Strategies algorithm for solving black-box optimization problems. This algorithm was developed in the 1960s by three students of the Technical University of Berlin, Hans-Paul Schwefel, Ingo Rechenberg and Peter Bienert [2]. This algorithm is the simplest in the field of ES. *Equation 2.1* shows how this algorithm generates one offspring from one parent, by utilizing a sample from the multivariate normal distribution *N(0,I)* and a scalar *σ* known as the step-size for mutation.

Equation 2.1

If the new offspring’s fitness is better than the parent’s, then itself becomes the new parent for the next generation, otherwise the parent remains the same. If the parent was replaced, then that would count as a successful mutation, otherwise it would count as a failure. The algorithm archives or keeps track of the number of successes and failures. This archive is used in the self-adaptation of the step-size *σ*. This self-adaptation method is known as the 1/5th success rule.

This self-adaptation method is based on the observation that the fastest convergence to the global optimum is achieved when approximately 1/5th of all mutations are successful, i.e. when the parent is replaced with the better offspring 1/5th of the time [2]. If the current success-rate is 1/5 then the value of *σ* stays the same. However, if the current success-rate is less than 1/5 then *σ* is increased by a constant factor, *c*. On the contrary, if the current success-rate is greater than 1/5 then *σ* is decreased by a constant factor, *c*. Therefore, to obtain the new step-size:

Equation 2.2

The value recommended for *c* in Equation 2.2 is and was recommended by Hans-Paul Schwefel using theoretical methods [8]. The probability of success *ps*, is measured based on the *10n* most recent mutations, where *n* is the problem dimension. The equation used to calculate the probability of success can be shown via the pseudocode of the (1+1) – ES algorithm in *Figure 2.2*.

A screenshot of a cell phone

Description automatically generated

Figure 2.2 – (1+1)- ES taken from [2].

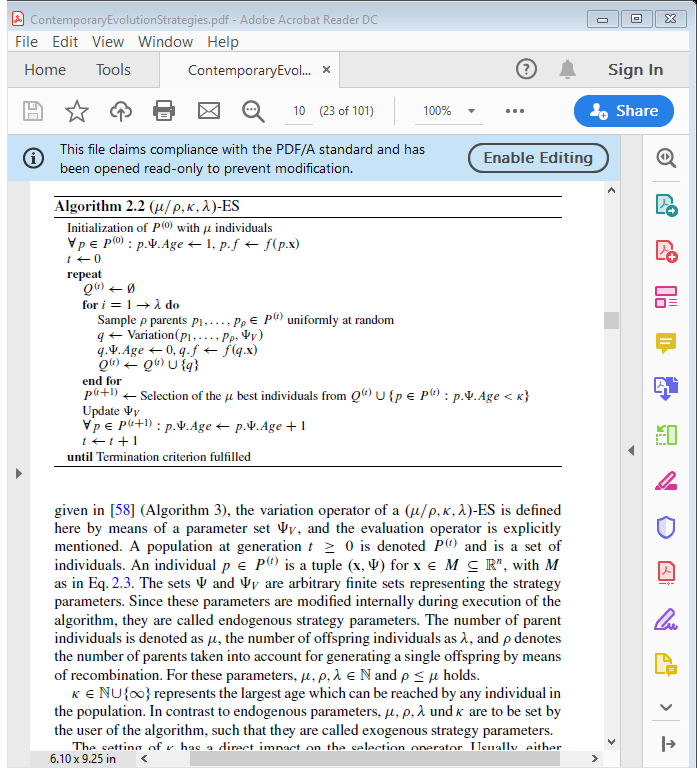
## 2.1.2 (1 + λ) - ES & (1, λ) – ES

The only difference between the (1 + λ) – ES and the (1 + 1) – ES is that instead of only generating one offspring from the parent, (1 + λ) – ES will generate λ offspring and replace the parent with the best offspring provided it is more fit than the current parent. However, in the (1, λ) – ES, λ offspring are also produced every generation, but the current parent is always replaced by the best offspring.

## 2.1.3 (µ/ρ, κ, λ) – ES

This algorithm is unique for introducing the concept that individual solutions have an *age* property. The age property of a solution is incremented by 1 each time it is selected to be in the next generation.

Figure 2.3 – (µ/ρ, κ, λ) – ES taken from [2].



As shown in *Figure 2.3*, the variation operator uses a parameter set *ѰV* and each individual solution has a set of attributes denoted *Ѱ*. The population at a specific generation is denoted *P*(*t*) for . Each individual in a given population is represented as a tuple *(x,Ѱ)* where represents the chromosome and *Ѱ* are its corresponding attributes. As usual in ES, the number of parent individuals are denoted by *µ* and the population size by *λ*. Additionally, *ρ* denotes the number of parents used to generate a single offspring using recombination. The maximum age that an individual solution can reach is denoted by *κ*.

Since *µ*, *λ*, *ρ*, κ, are determined by the user of the algorithm they are referred to as exogenous strategy parameters. The domain of these parameters is and . The most common settings are a one generation maximum lifetime (*κ* = 1) or an infinite maximum lifetime (*κ* = ) [2]. A one generation maximum lifetime for each solution is also referred to as comma-selection and an infinite maximum lifetime is referred to as plus-selection or elitist selection. Thus, when *κ* = 1 the title of the algorithm is (µ/ρ, λ) – ES and for *κ* = , (µ / ρ + λ) – ES where [2].

## 2.2 (µw, λ) - CMA-ES

Covariance Matrix Adaptation – Evolution Strategies is one of the most powerful black box optimization algorithms in the field of EC. What makes CMA-ES unique is the sampling of a covariance matrix to determine the mutation distribution, and using a method called step-size cumulation to determine the scale of the mutation distribution. The covariance matrix determines the direction of the search while the step size determines “how far” the search would be in that direction [7].

However, in CMA-ES the covariance matrix *C* undergoes an eigen decomposition. This decomposes the matrix into eigenvectors (which determine the direction of the search space) and eigenvalues (which determine the scale of the eigenvectors). Thus, the process of sampling new solutions is shown below in *Equation 2.3* [2].

Equation 2.3

In *Equation 2.3*, the eigenvectors and eigenvalues *BD* are used to shape the spherical noise *z* which is sampled from the multivariate normal distribution. The center ⟨x⟩ and the noise center ⟨y⟩ are updated using a method similar to an EDA, known as weighted-intermediate recombination. Weighted-intermediate recombination uses a set of *µ* weights where *w1 ≥ w2 ≥ . . . ≥ wµ* with for generating the new ⟨x⟩ and new ⟨y⟩ as follows [2]:

Equation 2.4

Equation 2.5

*Equation 2.4* shows how weighted intermediate recombination is used to determine the new point in space from which the next generation of sampling will take place. A weighted average was taken of the best *µ* solutions (*xi’s*) to create the new center. It’s important to note later that, the weighted average of the shaped noise, . In *Equation 2.5*, the best *µ* shaped-noise vectors (*yi’s*) associated with the best *µ* *xi* solutions from *Equation 2.4* also undergo weighted intermediate recombination to find the average of the best mutation steps.

Having covered the concepts such as multivariate normal distributions, covariance matrices and weighted intermediate recombination, the algorithm will be discussed in more detail starting with the cumulation methods for step-size adaptation (updating *σ*) and updating the covariance matrix *C*.

These cumulation paths can be thought of as history vectors since they accumulate parameter information over several generations [2]. In *Equation 2.6*, the first term is the decay factor where the cumulation time parameter is *cσ*<1 and approximately 1/*cσ* [9]. This means as new parameter information accumulates, the older information is gradually weened out of the cumulation path.

Equation 2.6

The expression under the square root of *Equation 2.6*, is a normalization constant associated with the first term, . The variance effective selection mass µ*eff* is necessary for adapting the strategy parameters [2]. It is important to note that *BD-1BT = C1/2* as this inverse matrix reverses the “shaping” of from *Equation 2.5* because *||pσ||* has to be compared with the expected length of a normally distributed random vector *E(||N(0,I)||)* in the step-size update in *Equation 2.7*.

Equation 2.7

The idea behind *Equation 2.7* is that if the length of *pσ* is greater than the expected length of a normally distributed random vector, then the value of σ will increase. For better understanding, consider *Figure 2.4* below which illustrates the length of the center (mean vector) in the generation sequence. The black arrow heads represent the position of the center at a specific generation, the arrows with white arrow heads represent the distance between the original center and the best center so far, and the arrow shafts represents the distance the center has moved.

A screenshot of a social media post

Description automatically generated

Figure 2.4 – Measuring the length of the evolution path [7].

In the left segment of Figure 2.4, you will notice that center moves around a lot less than the other segments. In this case the step-size is decreased to continue a tighter search around the current center as the algorithm may be converging on an optimum solution. However, in the right segment Figure 2.4, the opposite happens. The step size is increased because the mutation-steps are relatively in the same direction because the algorithm may still be looking for better solutions. The step-size increase is appropriate because it would be faster to make fewer big steps in the same direction rather than multiple small steps in the same direction.

Now we will move on to the covariance path update and the covariance matrix update.

⟨y⟩

Equation 2.8

*Equation 2.8* represents the covariance path update. Once again, the first term in this equation is the decay factor where the cumulation time parameter *cc*<1 and approximately 1/*cc* [9]. In the second term, *hσ* is either 1 or 0 based on conditions specified in *Equation 2.9*, where *t* is the generation number and *n* is the problem dimension [2].

Equation 2.9

Like the previous path update, the expression with the square root is a normalization constant associated with the first term, . The purpose of *hσ* is to choose whether to update *pc* based on information about the current generation *t* and the value of *||pc||* [2]. It’s also important to note that the length of a multivariate normally distributed random vector can be approximated by the gamma function shown in *Equation 2.10*

.

Equation 2.10

The variable *µeff* is the variance effective selection mass which is defined in *Equation 2.11*.

Equation 2.11

After the covariance path update, the next step of the algorithm is the covariance matrix update which is shown in *Equation 2.12*.

Equation 2.12

The covariance matrix update is a weighted matrix addition with 3 terms. The first term is the decay factor of the current covariance matrix. The second term is known as the *rank-one* update, which uses the covariance path and its transpose to form another matrix. According to the CMA-ES tutorial composed by Hansen & Auger [7], “the rank-one update uses the evolution path and reduces the number of function evaluations to learn straight ridges from *O(n2)* to *O(n)*”. The third term is known as the *rank-µ* update, also known as the weighted empirical covariance matrix, was added to the update equation to accommodate large population sizes. In essence, the covariance matrix is updated to increase the likelihood of successful mutation steps in the next generation [7]. The pseudocode for *(µw, λ)*-CMA-ES is shown in *Figure 2.5* and the following equations show how the strategy parameters from the algorithm are calculated:

*cc* is the parameter for the cumulation time of *pc* which is approximately 1/*pc*.

*cσ* is the parameter for the cumulation time of *pσ* which is approximately 1/*pσ*.

*dσ* is a dampening parameter that determines the change rate of the global step-size *σ* during the step-size update.

*c1* is the parameter that determines the change-rate of the covariance matrix with respect to the rank-one update.

*cµ* is the parameter that determines the change rate of the covariance matrix with respect to the rank-one update.

*wi:λ* is the set of logarithmic weights used for weighted-intermediate recombination of the sorted population to calculate the mean point. This method is also sometimes referred to as flattening.

A screenshot of a cell phone

Description automatically generated

Figure 2.5 – (µw, λ) - CMA-ES Algorithm [2]

## 2.2.1 LS-CMA-ES

The LS-CMA-ES algorithm is a CMA-ES variant that updates the covariance matrix based on an inverse Hessian matrix that is estimated via a solution to the appropriate least-squares (LS) estimation problem. Solving the LS estimation problem helps to find the optimal covariance matrix based on the fitness function being solved. Since solving this LS problem increases the computational complexity of the algorithm, it is only executed after a fixed number of generations. Finally, since solving the LS-problem isn’t guaranteed to make the covariance matrix more optimal, based on the error from the LS-problem, the algorithm switches between two modes, LS and CMA (Covariance Matrix Adaptation) [2]. More details can be found in *Appendix A* under *LS-CMA-ES*.

## 2.2.2 LR-CMA-ES

LR-CMA-ES is an extension of *(µw, λ)* - CMA-ES that introduces local restart conditions. The idea of restarting the algorithm is to avoid wasting function evaluations when the algorithm is facing stagnation in its optimization process. This algorithm provides five restart conditions for identifying stagnation, in which a new run of regular *(µw, λ)* - CMA-ES will commence [2]. There are two tolerance values, *Tx*= *σ*10-12 and *Tf* = 10-12, used within the restart conditions. The first restart condition, *equalfunvalhist* is satisfied if either the best fitness values of the last generations are the same, or the difference between the best fitness value and the weakest fitness values is smaller than *Tx* [2]. The second restart condition, *TolX*, is satisfied if for a vector *v = σ.pc* , *Tx* , where *i* ϵ {1,2,…,n}. The third restart condition, *noeffectaxis*, looks at the main coordinate axes formed by *C*. This condition is satisfied when, 0, where γi is the *ith* eigenvalue and *ui* is the *ith* eigenvector of *C* respectively, where . The fourth restart condition, *noeffectcoord*, also looks at the coordinate axis and is satisfied when . The fifth and final condition, *conditioncov*, is satisfied when the condition number of the covariance matrix exceeds *1014* [2].

## IPOP-CMA-ES

IPOP-CMA-ES is an extension of LR-CMA-ES with one small but very effective change. Should there be a restart of the *(µw, λ)* - CMA-ES, then the population size of the next run of *(µw, λ)* - CMA-ES is multiplied by a certain factor *η*, commonly set as *η* = 2 [2].

## 2.2.4 (µ + λ) - CMA-ES

*(µ + λ)* - CMA-ES is an extension of regular *(µw, λ)* - CMA-ES where the selection method is altered to perform elitism. Elitism is a well-known method used in the selection process of many Evolutionary Algorithms. What elitism does is it includes a certain number of the best selected offspring from the previous generation, in the selection process of the current generation. In *(µw, λ)* - CMA-ES, we selection the best *µ* offspring from the population, *λ*. Therefore, with elitism we add the *µ* offspring from the previous generation to the *λ* offspring of the current generation before performing a sort based on fitness and selecting the best *µ* offspring from the sorted *µ + λ* population. The main idea behind elitism in Evolutionary Algorithms is to ensure that there is always a set of the fittest solutions included in the population every generation, to maximize convergence.

## (1+1)-Cholesky-CMA-ES

The main property of this CMA-ES variant is that the covariance matrix is implicitly updated without using an eigen decomposition. As the name suggests, the algorithm uses a Cholesky decomposition instead which reduces the computational complexity in each generation from *O*(*n3*) to *O*(*n*2) [2].

## Active-CMA-ES

This modification to CMA-ES was inspired based on experiments which demonstrate that taking the worst offspring into account within the Estimation Distribution Algorithm (EDA) can increase convergence of Evolutionary Algorithms [10]. However, Active CMA-ES considers the worst offspring for the adaptation of the covariance matrix instead of the EDA by utilizing negative weights [2]. More details can be found in *Appendix A* under *Active-CMA-ES*.

## (µ, λ)-CMSA-ES

*(µ, λ)*-CMSA-ES aims to reduce the number of exogenous strategy parameters by reintroducing self-adaptation of the global step size *σ*. This simplifies the number of exogenous strategy parameters from five in *(µw, λ)-*CMA-ES down to two [2]. More details can be found in *Appendix A* under *(µ, λ) - CMSA-ES*.

## sep-CMA-ES

This CMA-ES variant uses a diagonal matrix in place of a covariance matrix in order to reduce the space and time complexity down to *O*(*n*) as compared to *O*(*n3*) since no eigen decompositions need to be done. Consequently, a diagonal matrix is not able to generate correlated mutations like a covariance matrix can. Therefore, the likelihood of producing successful mutations may be decreased. More details can be found in *Appendix A* under *sep-CMA-ES*.

## (1+1)-Active-CMA-ES

This algorithm is an extension of the *(1+1)-*Cholesky-CMA-ES that uses the Active-CMA-ES ideology of taking the worst offspring into account for the covariance matrix update. Rather than using the Cholesky decomposition for the covariance matrix , this algorithm uses the Cholesky factor A and its inverse A-1.

## SPO-CMA-ES

This algorithm is a restart-version of the (µw, λ)-CMA-ES that uses sequential parameter optimization (SPO) [11], to optimize the exogenous strategy parameters within the algorithm. SPO uses *design of experiments* (DoE) methods and *design and analysis of computer experiments* (DACE) [2]. This algorithm is concerned with optimizing the exogenous strategy parameters such as the number of offspring, the value of the initial step size and the selection pressure.

## 2.3 Niching Radius Algorithms

Niching techniques were created to improve the performance of EAs in multimodal optimization. This technique is governed by promoting population diversity and premature convergence which is achieved by ensuring certain properties exist within the population [4].

Niching is concerned with identifying certain aspects of the fitness landscape and leveraging this knowledge to find the global optimum. The most common aspects that are looked for in the fitness landscape are peaks because these are where the best solutions are found for multimodal problems [5].

The fundamental practice of niching is to divide the population into disjoint groups where each group is assigned to a peak or a basin of attraction which contains many peaks. Sampling then takes place from each of the peaks identified by the specific peak-detection algorithm. Some search points are replaced with newly discovered ones if they are not performing as well as the others.

## 2.4 Common Random Numbers (CRNs)

CRN is one of the most popular Variance Reduction Techniques (VRT). It is used when comparing similar systems with different configurations in hopes of investigating which system is better under the given circumstances [12]. The main idea is that two different configurations are compared while sharing the same experimental conditions, so that the results observed between the two system configurations are held with more integrity than if the two systems being compared did not share the same experimental conditions. For example, “experimental conditions” for computer simulations may be generated random variates or numbers that are used to feed both systems simultaneously as they carry out their separate operations. In other words, CRN is a VRT that attempts to induce positive correlation by using the same random numbers within simulation of both system configurations [12].

However, there is no guarantee that CRNs will always be a successful VRT. The success of CRNs depends heavily on the models under comparison. The reason for this is that different models may generate very different responses to the CRNs. Therefore, it is most effective when conducting CRN simulations with similar models. On the contrary, there are some classes of models where CRN is guaranteed to be successful as a VRT [12]. These classes of models and conditions are discussed in [13].

# 

## Methodology

## 3.1 Motivation

All CMA-ES variants discussed in the background use a single-model EDA and focus on manipulating the covariance matrix differently or optimizing exogenous strategy parameters. For this reason, this paper focuses on comparing DC-CMA-ES with IPOP-CMA-ES. ES Algorithms using a single point within the search space to perform sampling (such as a single-model EDA) have the potential to be trapped within areas which contain many local optima, even in instances of wide sampling. Naturally, ES algorithms decrease step-size when converging on good solutions. Even though this is a useful feature, this not only limits the chances of the algorithm being able to find the global optimum, but it also increases the likelihood of being trapped in areas where all local minima are relatively the same.

The main idea behind DC-CMA-ES is that the system does not have to rely on one EDA from which to perform sampling. If one EDA is getting trapped within local optima, the other can pick up the slack, making the system better at avoiding stagnation. For this reason, the DC-CMA-ES system seeks to diverge from the use of a single-model EDA with the goal of increasing the range of sampling and mitigating the occurrence of the system getting trapped within local optima.

## 3.2 Functionality of DC-CMA-ES

From our perspective, the simplest and most careful way of achieving this is by using two EDAs (one additional EDA) that have overlapping models. Overlapping models are used for two reasons: 1) manipulating the covariance matrix contributes to most of the computational complexity of a CMA-ES based algorithm (*O(n3)*), therefore having two covariance matrices would double the overall time complexity, 2) the additional EDA is dependent on the original EDA and can be synchronized to use the same covariance matrix and step-size. This method of synchronizing the sampling process is achieved by implementing a master-slave relationship between the two EDAs.

The master gives the slave its current best solution after every generation and the slave collects and manages these solutions in order to determine a point from which it will perform its own individual sampling. The second and third illustration in *Figure 3.1* shows how the master takes credit for all of the selected solutions found by the slave. Since the master uses a mixture of solutions from separate origins to determine the next search-point, it is less worried about being trapped in local optima.

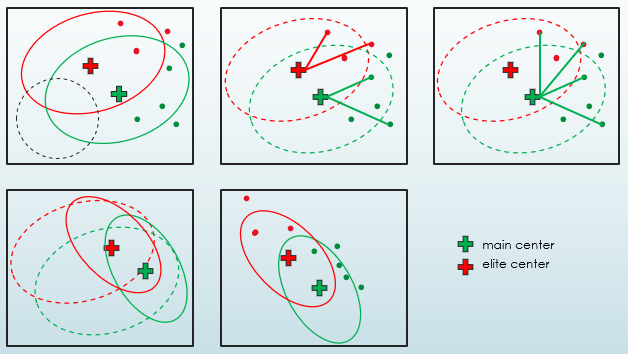


Figure 3.1 – Master-Slave Relationship between the two EDAs

Notice that since the two EDAs share a covariance matrix, they have the same mutation distribution shape. Since *Figure 3.1* is only an arbitrary example, the EDAs could also be searching in completely different areas of the problem landscape.

Before the synchronized sampling process is described in more detail, recall that DC-CMA-ES has a different EDA for computing the additional center(from now on referred to as the elite center) that is inspired by well-established techniques used in the field of EC namely, elitism, age and weighted-intermediate recombination.

Elitism is a technique where the best individual solutions from the previous generation are included with the current population to be considered in the selection process. The advantage of elitism in EC is that convergence is usually faster because good solutions are not lost unless better ones are produced.

The way elitism is used in DC-CMA-ES is inspired by the *(1+1)*-ES. The main idea was that the elite center should be replaced by the best solution of each generation. However, this method of updating the elite center was proven to be very unstable, especially in a multimodal fitness landscape.

This instability was solved with the introduction of a sliding window *ew*, which accumulates the best solution from each generation and operates in a First-in-First-out fashion. When *ew* becomes full, if the new elite solution is better than any of the solutions residing in the window, then the eldest elite solution exits while the new elite solution enters. The functionality of the elite sliding window resembles the age property used by the *(µ/ρ, κ, λ)* – ES as the *life-span* of an elite solution partially depends on the size of the sliding window, *s*. *Figure 3.2* illustrates the functionality of an elite sliding window with a window size of *s* = 8.

A close up of a logo

Description automatically generated

Figure 3.2 - Functionality of the elite sliding window

The rationale behind this sliding window is that it would improve the stability of the elite center update if we average the elite solutions that currently exist in the sliding window. Performing a weighted-average of the elite solutions in the sliding window is analogous to the use of weighted-intermediate recombination. *Equation 3.1* shows how the weights are calculated, where and *n* is the problem dimension. This equation produces log weights that are skewed so the first few terms are not as heavily weighted as compared to the weights used for weighted-intermediate recombination in *(µw, λ) –* CMA-ES. These weights take the most recent elite solutions into higher consideration to ensure steady convergence but not so much as to quickly forget the eldest elite solutions in the window.

Equation 3.1

This way of computing the elite-center leverages the history of elite solutions with the aim of obtaining an optimum position in the search space from which to sample solutions. The time period that elite solutions can stay in the window, paired with the fact that only better elite solutions can enter, ensures that the elite-center is less likely to be trapped in local minima because solutions in the window often change and never get worse. For these reasons, the EDA which produces the elite center can be referred to as the elite history EDA (EH-EDA).

The job of sampling the new solutions *λx* and *λe* can be split between the main center and the elite center at any ratio, *rx*and *re* as long as *λx + λe = λ*. Since these two centers share a covariance matrix and step-size, every solution sampled from the elite center is tied back to the main center , almost as if it were sampled from originally as follows:

Equation 3.2

where is the *ith* solution sampled from and is set of mutations or noise vectors corresponding to the solutions in . This allows the two centers to use the same covariance matrix for sampling because from the perspective of the covariance matrix, every mutation step originated from the main center. As a result, no extra covariance matrix is necessary.

Aside, using separate covariance matrices would pose a problem to DC-CMA-ES. With separate covariance matrices, there would be no reason to combine the solutions and tie them back to the main center. This would conflict with the functionality of the additional EDA since it uses the best solutions generated from itself or the original EDA to calculate which point it will sample from next. Changing the way that the elite history EDA is built could potentially collapse the entire algorithm. If the elite history EDA was not dependent on the original, then this would not be a problem.

In the case where the sampling ratios *rx* and *re* are *1.0* and *0.0* respectively, DC-CMA-ES is identical to CMA-ES because no solutions are sampled from . It is because of this adjustable ratio that DC-CMA-ES can be thought of as a layer on top of CMA-ES. However, the choice of these sampling ratios is important as you may want one center to dominate the sampling process or evenly share with the other.

It is intuitive that the currently best performing center should dominate the sampling process to minimize the risk of generating poor solutions. Instead of utilizing complex niching radius techniques to estimate good search points in the fitness landscape, we naturally came up with two conditional metrics for determining which center is more likely to generate better solutions.

The first one is referred to as the *evaluation* metric. The evaluation metric checks after every generation to see if the elite center has a better fitness value than the main center. If this is the case, then an appropriate sampling ratio is chosen for the next generation. For example, if the elite center is the currently the better one, you may want it to generate more samples than it was before. However, if the elite center is no longer the better one, the ratios will default to what they were originally. The drawback of the evaluation metric is that it requires two extra function evaluations to compare the fitness of both centers every generation.

The second metric is referred to as the *rank-sum*. As the name may suggest, this metric performs a rank sum on the selected *µ* solutions to see which center is producing better offspring (the minimum rank-sum value). These *µ* solutions that are used to update the main center could be made up of any ratio of solutions from the two centers. Thus, the possibility exists that only solutions sampled from one of the centers are selected. When using the rank-sum metric, the ratio of solutions cannot favor one center completely because if one center generates all of the solutions, then the other center will not have a rank-sum. Having determined if the elite center produced a smaller rank-sum than the main center, the ratios would be changed appropriately. If the elite center has not produced the smaller rank-sum than the main center, then the default ratios would be reapplied.

The conditional and rank-sum metrics are good alternatives to niching radius techniques as they help the algorithm to determine a good opportunity for the elite center to contribute more to the sampling of solutions. They also help the algorithm to avoid generating bad solutions in the case where one of the centers may be unfit.

## 3.3 Pseudocode

Algorithm: *(µw, λ, rx, re)* – DC-CMA-ES

initialize , initialize

**repeat**

*\* λ*

**for**

**end**

**for**

**end**

and *x1* is better than any solution in *ew*

removeOldest()

**else**

addNewest()

**end**

⟨y⟩

The additional terms used in the DC-CMA-ES that are not recognized in 2.2 *(µw, λ)* - CMA-ES are described in *Table 3.1*.

|  |  |
| --- | --- |
| Term | Description |
|  | elite center |
| *ew* | sliding window which stores elite solutions |
| *s* | sliding window size |
| *rx* | ratio of solutions to be sampled from the main center |
| *re* | ratio of solutions to be sampled from the elite center |
| *λx* | number of solutions sampled from the main center |
| *λe* | number of solutions sampled from the elite center |
|  | a set containing the solutions sampled from the main center |
|  | a set containing the solutions sampled from the elite center |
|  | the set of noise vectors added to to obtain the solutions in *px* |
|  | the set of noise vectors added to to obtain the solutions in *px* |
| *a1:s* | the set of weights used to calculate the elite center |

Table 3.1 - Description of terms relevant to DC-CMA-ES

# 

## Experiments

## 4.1 Performance Measures & Test Functions

The performance of an ES is measured by the best solution found within a fixed budget of evaluations. When systems are compared, they are given the same budget of evaluations. The best performing system in that round would be the one with the best solution found. A total of 8 test functions were used to run the experiments, 6 of which are multimodal and defined as follows [14]:

Equation 4.1 - Ackley - evaluated on the hyper-cube xi ϵ [-32.768, 32.768] for all i = 1 to d, global minimum f(x)=0 at x=(0,...,0)

Equation 4.2 - Griewank - evaluated on the hyper-cube xi ϵ [-600,600], for all i = 1 to d, global minimum f(x)=0 at x = (0, …, 0)

Equation 4.3 – Rastrigin – evaluated on the hyper-cube xi ϵ [-5.12,5.12], for all i = 1 to d, global minimum f(x)=0 at x = (0, …, 0)

Equation 4.4 – Levy – evaluated on the hyper-cube xi ϵ [-10,10], for i = 1 to d, global minimum f(x)=0 at x = (0, …, 0)

Equation 4.5 - Schwefel – evaluated on the hypercube xi ϵ [-500,500], for all i = 1 to d, global minimum f(x)=0 at x=(420.969,…,420.969)

Equation 4.6 – Rosenbrock – evaluated on the hypercube xi ϵ [-5,10] for all i = 1 to d, global minimum f(x) = 0 at x= (1, …, 1)

According to Kok & Sandrock [15], the Rosenbrock function forms additional local minima and many saddle points when the dimensionality is ≥ 4. Since the lowest dimensionality experimented on in this thesis is 5, we consider it as a multimodal function.

The multimodal test functions chosen for the experiments are the most commonly used in the field of optimization. Some of the multimodal functions differ in terms of the spread and number of local optima and are scalable to high dimensions.

The remaining 2 test functions elliptical and zakharov are unimodal and from the classes bowl-shaped and plate-shaped respectively. These unimodal test functions are defined as follows [14]:

Equation 4.7 – Elliptical – evaluated on the hypercube xi ϵ [-64.536, 65.536], for all i = 1 to d, global optimum f(x)=0 at x=(0, …, 0)

Equation 4.8 – Zakharov – evaluated on the hypercube xi ϵ [-5,10] for all i = 1 to d, global minimum f(x) = 0 at x = (0, …, 0)

## 4.2 Experimental Setup

Due to the different ratios and metrics you can use within DC-CMA-ES, the total number of systems being compared is 17 as shown in *Table 5.1*. Each system was run 50 times for each of the 8 test functions at dimensions 5, 10, 25, 50, 75 and 100. The reproduction parameters were set as follows [2]:

Regarding experimentation, DC-CMA-ES was run simultaneously with IPOP-CMA-ES using a Variance Reduction Technique (VRT) known as Common Random Numbers (CRNs), briefly described in the backgrond [13] [12]. Under this technique, IPOP-CMA-ES and DC-CMA-ES share one common random number stream. More specifically, they use the same samples from the multivariate normal distribution *N(0,I)*, also known as *spherical noise*, which are then shaped with the same eigenvectors and eigenvalues *BD* as previously shown in *Equation 2.3,* since they share the same covariance matrix.

## 4.3 Integrating CRN Simulations and Restart ES Algorithms

A close up of a map

Description automatically generated

Figure 4.1 – Common Random Number Simulation between DC-CMA-ES & CMA-ES

*Figure 4.1* illustrates how the CRN system was modified to work with restarting systems and a fixed budget of evaluations. Recall that with the application of the IPOP technique to CMA-ES systems may need to restart and increase their population size based on certain restart conditions. The challenge that arises when performing CRN simulations on systems that restart is that if a system needs to restart it will no longer share the same CRNs with the other system.

For example, from *Figure 4.1* DC-CMA-ES needs to restart at *50n* evaluations when the fixed budget of evaluations is *100n* (where n is the problem dimensionality). At this point, DC-CMA-ES must stop running and wait until CMA-ES reaches *100n* evaluations or also needs to restart. In this arbitrary example, CMA-ES runs to the fixed budget of evaluations without restarting. Since CMA-ES is finished, DC-CMA-ES will continue from the *50n* evaluation point until the *100n* evaluations point, with no other system to share CRNs with. If DC-CMA-ES needed to restart again before the fixed budget of evaluations, it would do so immediately since it does not have to wait on the other system. Therefore, we can assume that CRN simulations would not be as effective if there are frequent occurrences of restarts.

Due to time constraints, no analysis has been done to demonstrate that the CRN technique reduces the variance between these two systems. However, if these restarting CMA-ES systems were not assigned a fixed budget of evaluations and given a fitness value goal to reach instead, some time-saving benefits can be reaped. For example, if one system found the goal first at *x* evaluations, then the other system would be given a chance to find the goal using no more than the *x* evaluations if not already surpassed. This method is more efficient than running the systems independently and measuring success based on the number of evaluations used to reach the goal.

An analysis of variance (ANOVA) and Tukey post-hoc corrections of the different interacting factors were performed on the simulation output data. The best fitness values found for each run was transformed for normality using the Box-Cox transform before performing the ANOVA and Tukey post-hoc corrections.

# 

## Results and Discussion

In this chapter, even though 17 different systems were tested, we have filtered the irrelevant results out of each table for readability. The full details of each table can be found in *Appendix B*. *Table 5.*1 contains the names given to systems based on their sampling ratios representing the original EDA and the elite history EDA respectively.

|  |  |  |
| --- | --- | --- |
| **System** | **Sampling Ratio** | **Description** |
| CMA-ES | 1.0 - 0.0 | the original IPOP-CMA-ES algorithm |
| CMA:eh | 0.75 - 0.25 | the dualcenter system favoring sampling from original EDA |
| cma:eh | 0.5 - 0.5 | the dualcenter system with equal sampling from both EDAs |
| cma:EH | 0.25 - 0.75 | the dualcenter system favoring the elite history EDA |
| EH | 0.0 - 1.0 | sampling from the elite history EDA only |
| - | all | all possible ratios of the system type |

Table 5.1 – Description of Fixed Sampling Ratio Systems

## 5.1 Results of Fixed Ratio Systems

|  |  |  |
| --- | --- | --- |
| **system** | **means** | **group** |
| CMA:eh | 0.237818369 | ab |
| cma:eh | 0.264913244 | b |
| CMA-ES | 0.461258262 | c |
| cma:EH | 0.47168142 | c |
| EH | 0.980726836 | d |

Table 5.2 - Main Effects of System Performance

Regarding the system names, excluding CMA-ES, the capitalization indicates which EDA does most of the sampling. Systems are ordered based on their means and the last column, group, is provided as a visual aid to decipher which systems are better, worse or no different from each other. Based on a 95% confidence level, systems not sharing a letter are statistically significant, while those sharing a letter are statistically insignificant.

From the main effects shown in *Table 5.2*, we can concur that the dualcenter system with an equal ratio (cma:eh), or a ratio in which the main center dominates (CMA:eh) performs better than CMA-ES.

To facilitate an accurate and detailed analysis we will look at the interactions between the three factors of interest, namely system, function and dimension. The system factor is important because we want to compare the systems based on performance. The function and dimension factor are of particular interest because our objective is to highlight the performance of the systems on multimodal functions at various levels of dimensionality. We will observe and discuss the interactions between the two factors, system and function, for the test functions defined in *Equation 4.1* - *Equation 4.8*. After this, the interaction between the two factors system and dimensionality will be shown and discussed.

|  |  |  |  |
| --- | --- | --- | --- |
| **function** | **system** | **means** | **group** |
| ackley | cma:eh | 0.015111253 | a |
| ackley | CMA:eh | 0.018326322 | a |
| ackley | CMA-ES | 0.045523428 | b |
| ackley | cma:EH | 0.050402009 | bc |
| ackley | EH | 0.17258676 | d |

Table 5.3 – System performance on Ackley

From *Table 5.3*, it is shown that the performance of the Ackley function shadows the main effects shown in *Table 5.2*. The dualcenter systems with equal ratios and favouring the main center performed better than CMA-ES, while the dualcenter system with ratios strictly favouring the elite center (EH) performed worse than CMA-ES.

|  |  |  |  |
| --- | --- | --- | --- |
| **function** | **system** | **means** | **group** |
| griewank | cma:eh | 0.000179814 | a |
| griewank | CMA:eh | 0.000196035 | ab |
| griewank | cma:EH | 0.000599728 | cd |
| griewank | CMA-ES | 0.00111369 | d |
| griewank | EH | 0.003230298 | e |

Table 5.4 – System performance on Griewank

*Table 5.4* shows that the majority of the dualcenter systems perform better than CMA-ES and the strictly elite center sampling dualcenter system. It is also observed that this function is insensitive to most variations in sampling ratios.

|  |  |  |  |
| --- | --- | --- | --- |
| **function** | **system** | **means** | **group** |
| levy | CMA:eh | 0.042312906 | a |
| levy | CMA-ES | 0.048092132 | ab |
| levy | cma:eh | 0.094950956 | c |
| levy | cma:EH | 0.212533793 | d |
| levy | EH | 0.35366576 | de |

Table 5.5 – System performance on Levy

*Table 5.5* shows that CMA-ES and the dualcenter system with sampling ratio *0.75-0.25* performs better than most of the other dualcenter systems but no systems perform better than CMA-ES. Levy is the only multimodal function that does not reflect the advantage gained by DC-CMA-ES. The only thing that separates the levy function from the other multimodal functions is that is has fewer local minima compared to the other ones chosen for the tests. The other multimodal functions such as Rastrigin and Griewank have many widespread local minima [14].

|  |  |  |  |
| --- | --- | --- | --- |
| **function** | **system** | **means** | **group** |
| rastrigin | cma:eh | 52.10532943 | a |
| rastrigin | CMA:eh | 53.29764564 | a |
| rastrigin | cma:EH | 54.35393177 | a |
| rastrigin | EH | 60.93550937 | ab |
| rastrigin | CMA-ES | 147.1763114 | c |

Table 5.6 – System performance on Rastrigin

*Table 5.6* shows that much like the Griewank function, Rastrigin is also insensitive to changes in the sampling ratios. We can conclude this because every dualcenter system performs better than CMA-ES on the Rastrigin test function. Both Rastrigin and Griewank have many widespread local minima which are regularly distributed [14].

|  |  |  |  |
| --- | --- | --- | --- |
| **function** | **system** | **means** | **groups** |
| schwefel | cma:EH | 0.000101271 | a |
| schwefel | cma:eh | 0.000116985 | a |
| schwefel | CMA:eh | 0.000208435 | abc |
| schwefel | CMA-ES | 0.000297597 | bcd |
| schwefel | EH | 0.000477601 | cd |

Table 5.7 – System performance on Schwefel

From the data in *Table 5.7*, we noticed that on the Schwefel function, the sampling ratios favoring the elite center as well as equal sampling ratios perform better than CMA-ES.

|  |  |  |  |
| --- | --- | --- | --- |
| **function** | **system** | **means** | **group** |
| elliptical | CMA:eh | 0.097608199 | ab |
| elliptical | cma:eh | 0.145873704 | bc |
| elliptical | CMA-ES | 0.239338247 | c |
| elliptical | cma:EH | 0.461751442 | d |
| elliptical | EH | 1.410394786 | e |

Table 5.8 – System performance on Elliptical

*Table 5.8* shows that on elliptical the only dualcenter system that performs better than CMA-ES is the one with a *0.75-0.25* ratio (CMA:eh).

|  |  |  |  |
| --- | --- | --- | --- |
| **function** | **system** | **means** | **group** |
| rosenbrock | CMA:eh | 35.3719264 | a |
| rosenbrock | CMA-ES | 36.86061248 | ab |
| rosenbrock | cma:eh | 45.60300689 | abc |
| rosenbrock | cma:EH | 67.23614793 | c |
| rosenbrock | EH | 112.6584486 | d |

Table 5.9 – System performance on Rosenbrock

*Table 5.9* shows that all dual-center systems, except for those two with ratios *0.0-1.0* and *0.25-0.75* (EH & cma:EH), do not hinder or improve the performance of its underlying CMA-ES mechanism on Rosenbrock.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **function** | **system** | **rx** | **re** | **means** | **group** |
| zakharov | dualcenter | 0.75 | 0.25 | 9.503907295 | a |
| zakharov | dualcenter | 0.5 | 0.5 | 9.579555485 | a |
| zakharov | dualcenter | 0.25 | 0.75 | 13.29205583 | abc |
| zakharov | dualcenter | 0.0 | 1.0 | 15.71439568 | bc |
| zakharov | CMA-ES | 1.0 | 0.0 | 16.24145573 | c |

Table 5.10 – System performance on Zakharov

*Table 5.10* shows that the best sampling ratios for the Zakharov function are *0.5-0.5* (cma:eh) and *0.75-0.25* (CMA:eh) as they both perform better than CMA-ES.

## 5.2 Results of Conditional Ratio Systems

Recall that these systems choose between two sets of sampling ratios based on which center is expected to be more successful. If the elite center is believed to be better than the main center, then the ratios chosen for that system would be the ones listed in the 3rd column, otherwise the ratios used would be the ones listed in the 2nd column.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **system** | **original EDA better** | **elite history EDA better** | **means** | **group** |
| rank-sum | CMA:eh | cma:eh | 0.196919405 | a |
| evaluation | CMA-ES | cma:eh | 0.222463855 | ab |
| rank-sum | CMA:eh | cma:EH | 0.228289752 | abc |
| evaluation | CMA-ES | cma:EH | 0.229297767 | abc |
| evaluation | CMA-ES | CMA:eh | 0.232724025 | bc |
| evaluation | CMA-ES | EH | 0.23457734 | bc |
| CMA:eh | - | - | 0.237818369 | bc |
| cma:eh | - | - | 0.264913244 | cd |
| CMA-ES | - | - | 0.461258262 | e |
| cma:EH | - | - | 0.47168142 | e |
| EH | - | - | 0.980726836 | f |

Table 5.11 – Main Effects of System Performance with Conditional Metrics

Overall, the conditional metrics do have a significant advantage over the regular dualcenter systems as well as CMA-ES.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **function** | **system** | **original EDA better** | **elite history EDA better** | **means** | **group** |
| ackley | rank-sum | CMA:eh | cma:eh | 0.010339798 | a |
| ackley | rank-sum | CMA:eh | cma:EH | 0.014231207 | ab |
| ackley | evaluation | CMA-ES | cma:EH | 0.014807807 | abc |
| ackley | cma:eh | - | - | 0.015111253 | abc |
| ackley | evaluation | CMA-ES | CMA:eh | 0.015860541 | abc |
| ackley | evaluation | CMA-ES | EH | 0.015897679 | abc |
| ackley | CMA:eh | - | - | 0.018326322 | abc |
| ackley | rank-sum | cma:eh | cma:EH | 0.018462547 | abc |
| ackley | evaluation | CMA-ES | cma:eh | 0.018771639 | abc |
| ackley | evaluation | CMA:eh | cma:eh | 0.020970899 | cd |
| ackley | evaluation | CMA:eh | cma:EH | 0.021943994 | cde |
| ackley | CMA-ES | - | - | 0.045523428 | f |
| ackley | cma:EH | - | - | 0.050402009 | fg |
| ackley | EH | - | - | 0.17258676 | h |

Table 5.12 – System Performance on Ackley with Conditional Metrics

From *Table 5.12*, we observe that the conditional metrics did not improve the dualcenter systems with equal or original EDA dominating sampling ratios (cma:eh & CMA:eh) on Ackley, but still performed better than CMA-ES. The same analysis stands for the Griewank function as well, shown in *Table 5.13*.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **function** | **system** | **original EDA better** | **elite history EDA better** | **means** | **group** |
| griewank | rank-sum | CMA:eh | cma:eh | 0.000126535 | abc |
| griewank | evaluation | CMA-ES | cma:eh | 0.00015207 | bcd |
| griewank | evaluation | CMA-ES | cma:EH | 0.0001623 | bcd |
| griewank | rank-sum | CMA:eh | cma:EH | 0.00016949 | bcd |
| griewank | cma:eh | - | - | 0.000179814 | cd |
| griewank | rank-sum | cma:eh | cma:EH | 0.000195362 | cd |
| griewank | CMA:eh | - | - | 0.000196035 | cde |
| griewank | cma:EH | - | - | 0.000599728 | fg |
| griewank | CMA-ES | - | - | 0.00111369 | g |
| griewank | EH | - | - | 0.003230298 | h |

Table 5.13 – System Performance on Griewank with Conditional Metrics

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **function** | **system** | **original EDA better** | **elite history EDA better** | **means** | **group** |
| levy | CMA:eh | - | - | 0.042312906 | abc |
| levy | CMA-ES | - | - | 0.048092132 | bcd |
| levy | rank-sum | CMA:eh | cma:eh | 0.058640452 | cde |
| levy | evaluation | CMA-ES | CMA:eh | 0.068530702 | cdef |
| levy | evaluation | CMA-ES | cma:eh | 0.069312447 | cdef |
| levy | evaluation | CMA:eh | cma:eh | 0.077151729 | cdef |
| levy | cma:eh | - | - | 0.094950956 | efg |
| levy | cma:EH | - | - | 0.212533793 | hi |
| levy | EH | - | - | 0.35366576 | ij |

Table 5.14 – System Performance on Levy with Conditional Metrics

Regarding the Levy function in *Table 5.14*, the conditional metrics show no improvements over CMA-ES or CMA:eh. It is also noticed that this function does not perform well when the elite history EDA contributes to sampling half or more of the population size because the cma:eh, cma:EH and EH systems perform worse than CMA-ES.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **function** | **system** | **original EDA better** | **elite history EDA better** | **means** | **group** |
| rastrigin | evaluation | all | all | - | abc |
| rastrigin | rank-sum | all | all | - | abc |
| rastrigin | cma:eh | - | - | 52.10532943 | bc |
| rastrigin | CMA:eh | - | - | 53.29764564 | bc |
| rastrigin | cma:EH | - | - | 54.35393177 | bc |
| rastrigin | EH | - | - | 60.93550937 | c |
| rastrigin | CMA-ES | - | - | 147.1763114 | d |

Table 5.15 – System Performance on Rastrigin with Conditional Metrics

The analysis for the Rastrigin function is rather straight-forward. All systems perform better than CMA-ES, but systems with conditional metrics have no advantage over the regular dualcenter systems.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **function** | **system** | **original EDA better** | **elite history EDA better** | **means** | **groups** |
| schwefel | evaluation | CMA-ES | EH | 3.06771E-05 | a |
| schwefel | rank-sum | CMA:eh | cma:EH | 3.37462E-05 | a |
| schwefel | rank-sum | cma:eh | cma:EH | 3.65182E-05 | ab |
| schwefel | rank-sum | CMA:eh | cma:eh | 3.65884E-05 | ab |
| schwefel | cma:EH | - | - | 0.000101271 | c |
| schwefel | cma:eh | - | - | 0.000116985 | c |
| schwefel | CMA:eh | - | - | 0.000208435 | cd |
| schwefel | CMA-ES | - | - | 0.000297597 | cde |
| schwefel | EH | - | - | 0.000477601 | ef |

Table 5.16 – System Performance on Schwefel with Conditional Metrics

In *Table 5.16*, the conditional dualcenter systems performed better than the best performing regular dualcenter systems. The evaluation dualcenter system has ratios of *1.0-0.0* (CMA-ES) when the elite center is not as fit as the main center and *0.0-1.0* otherwise. This system switches between CMA-ES and pure elite history (EH) sampling depending on which center has the better fitness. Interestingly, this evaluation dualcenter system performed just as well as the rank-sum systems even though it consumed two extra evaluations per generation. This shows that sometimes trading evaluations to determine the best origin to perform sampling from can be ideal.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **function** | **system** | **original EDA better** | **elite history EDA better** | **means** | **group** |
| elliptical | evaluation | CMA-ES | cma:eh | 0.077045025 | ac |
| elliptical | evaluation | CMA-ES | CMA:eh | 0.08806114 | abc |
| elliptical | evaluation | CMA-ES | EH | 0.088961955 | abc |
| elliptical | rank-sum | CMA:eh | cma:eh | 0.09671144 | acd |
| elliptical | CMA:eh | - | - | 0.097608199 | acd |
| elliptical | evaluation | CMA-ES | cma:eh | 0.106912199 | acde |
| elliptical | cma:eh | - | - | 0.145873704 | cdefg |
| elliptical | CMA-ES | - | - | 0.239338247 | g |
| elliptical | cma:EH | - | - | 0.461751442 | h |
| elliptical | EH | - | - | 1.410394786 | i |

Table 5.17 – System Performance on Elliptical with Conditional Metrics

Overall, one of the rank-sum dualcenter systems and three of the evaluation systems also perform better than CMA-ES. However, they are not significantly better some of the regular dualcenter systems. Even though this test function is unimodal and one of the easiest functions to solve, improvements are still noticed.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **function** | **system** | **original EDA better** | **elite history EDA better** | **means** | **group** |
| rosenbrock | evaluation | CMA-ES | EH | 30.55174864 | a |
| rosenbrock | evaluation | CMA-ES | cma:EH | 30.58473897 | a |
| rosenbrock | evaluation | CMA-ES | cma:EH | 31.41133945 | ab |
| rosenbrock | evaluation | CMA-ES | CMA:eh | 32.05685651 | abc |
| rosenbrock | CMA:eh | - | - | 35.3719264 | abcd |
| rosenbrock | CMA-ES | - | - | 36.86061248 | abcde |
| rosenbrock | rank-sum | CMA:eh | cma:EH | 37.33553933 | abcde |
| rosenbrock | cma:eh | - | - | 45.60300689 | abcdef |
| rosenbrock | cma:EH | - | - | 67.23614793 | f |
| rosenbrock | EH | - | - | 112.6584486 | g |

Table 5.18 - System Performance on Rosenbrock with Conditional Metrics

On the Rosenbrock function, conditional metrics provide no significant advantage to regular dualcenter systems. Although not a statistically significant statistic, it is still interesting to see that the systems which go back and forth between CMA-ES and dualcenter systems conditionally have the lowest means. Much like the Levy function, the elite history EDA is not the best at solving the Rosenbrock function.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **function** | **system** | **original EDA better** | **elite history EDA better** | **means** | **group** |
| zakharov | rank-sum | some | some | - | a |
| zakharov | CMA:eh | - | - | 9.503907295 | a |
| zakharov | cma:eh | - | - | 9.579555485 | a |
| zakharov | rank-sum | some | some | - | ab |
| zakharov | evaluation | all | all | - | abc |
| zakharov | cma:EH | - | - | 13.29205583 | abc |
| zakharov | EH | - | - | 15.71439568 | bc |
| zakharov | CMA-ES | - | - | 16.24145573 | c |

Table 5.19 – System Performance on Zakharov with Condition Metrics

The rank-sum dualcenter systems also perform better than CMA-ES, but not better or worse than the dualcenter systems which have equal sampling as well as sampling favoring the elite center (cma:eh & CMA:eh). Lastly, none of the evaluation dualcenter systems performed better or worse CMA-ES. This indicates that the function evaluation trade-off is not ideal on Zakharov.

## 5.3 Dimensionality

Up until this point, we have only looked at the system performance for each function. The other factor to consider is the dimensionality of the problem. Thus, we look at the system performance for each of the dimensions used in the experimentation.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **dimension** | **system** | **original EDA better** | **elite history EDA** | **means** | **group** |
| 5 | cma:eh | - | - | 0.002805 | a |
| 5 | cma:EH | - | - | 0.003117 | a |
| 5 | EH | - | - | 0.003442 | a |
| 5 | CMA:eh | - | - | 0.003644 | a |
| 5 | CMA-ES | - | - | 0.004102 | a |
| 5 | rank-sum | all | all | - | a |
| 5 | evaluation | all | all | - | b |

Table 5.20 – System performance at dimension 5

Based on the data in *Table 5.20*, it is observed that the regular dualcenter systems at any ratio do not perform better or worse than CMA-ES at dimension 5. It is also highly-noticeable that the evaluation dualcenter systems all perform worse than CMA-ES at this dimension. This shows that sacrificing extra evaluations to determine the best center is not useful when dimensionality is low.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **dimension** | **system** | **original EDA better** | **elite history EDA better** | **means** | **group** |
| 10 | rank-sum | CMA:eh | cma:eh | 0.010649 | ab |
| 10 | rank-sum | CMA:eh | cma:EH | 0.011782 | ab |
| 10 | cma:eh | - | - | 0.013229 | abc |
| 10 | CMA:eh | - | - | 0.013727 | abc |
| 10 | cma:EH | - | - | 0.018359 | acd |
| 10 | rank-sum | cma:eh | cma:EH | 0.020118 | cd |
| 10 | EH | - | - | 0.024983 | de |
| 10 | evaluation | all | all | - | de |
| 10 | CMA-ES | - | - | 0.03433 | e |

Table 5.21 – System performance at dimension 10

Improvements due to dualcenter sampling are first witnessed at dimension 10. Excluding the conditional dualcenter systems, it is shown that all of the dualcenter sampling systems perform better than CMA-ES on the function test suite at dimension 10. Similarly, in this dimension the dualcenter systems with the evaluation metric does not perform better than CMA-ES due to the sacrifice of function evaluations.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **dimension** | **system** | **original EDA better** | **elite history EDA better** | **means** | **group** |
| 25 | rank-sum | CMA:eh | all | 0.090101 | a |
| 25 | evaluation | CMA-ES | cma:eh | 0.112087 | ab |
| 25 | CMA:eh | - | - | 0.116607 | ab |
| 25 | evaluation | CMA-ES | all | 0.120512 | abc |
| 25 | evaluation | CMA:eh | cma:EH | 0.132566 | abc |
| 25 | cma:eh | - | - | 0.137899 | abcd |
| 25 | cma:EH | - | - | 0.248323 | e |
| 25 | CMA-ES | - | - | 0.325685 | f |
| 25 | EH | - | - | 0.591967 | g |

Table 5.22 – System performance at dimension 25

Table 5.22 shows that the majority of the dualcenter systems and conditional dualcenter systems perform better than CMA-ES. Regarding the evaluation metric, the function sacrifice starts to become profitable around this dimension.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **dimension** | **system** | **original EDA better** | **elite history EDA better** | **means** | **groups** |
| 50 | evaluation | CMA-ES | all | - | ab |
| 50 | cma:eh | - | - | 0.585818 | bc |
| 50 | rank-sum | CMA:eh | all | 0.690185 | bcde |
| 50 | CMA:eh | - | - | 0.698816 | cde |
| 50 | evaluation | CMA:eh | all | - | cdef |
| 50 | rank-sum | cma:eh | all | - | efg |
| 50 | CMA-ES | - | - | 1.30453 | gh |
| 50 | cma:EH | - | - | 1.762478 | hi |
| 50 | EH | - | - | 4.083197 | j |

Table 5.23 – System performance at dimension 50

The results from *Table 5.23* highly reflect those from *Table 5.22*. At this dimension, notice that the conditional dualcenter systems that switch to and from the ratio *1.0-0.0* (CMA-ES) are performing the best. Even some of these conditional dualcenter systems are shown to be not just better than CMA-ES but also some regular dualcenter systems. For example, the evaluation dualcenter system in group ‘ab’ is significantly better than the dualcenter system with ratios *0.75-0.25* (CMA:eh).

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **dimension** | **system** | **original EDA better** | **elite history EDA better** | **means** | **group** |
| 75 | evaluation | CMA-ES | CMA:eh | 1.130551 | ab |
| 75 | evaluation | CMA-ES | cma:eh | 1.170769 | ab |
| 75 | evaluation | CMA-ES | cma:EH | 1.181608 | ab |
| 75 | evaluation | CMA-ES | EH | 1.342304 | abcd |
| 75 | rank-sum | CMA:eh | cma:eh | 1.567011 | bcde |
| 75 | CMA:eh | - | - | 2.067891 | defg |
| 75 | rank-sum | CMA:eh | cma:EH | 2.106456 | efg |
| 75 | rank-sum | cma:eh | cma:EH | 2.653593 | fgh |
| 75 | cma:eh | - | - | 2.890998 | gh |
| 75 | CMA-ES | - | - | 4.134013 | hi |
| 75 | cma:EH | - | - | 5.622925 | ij |
| 75 | EH | - | - | 14.38816 | k |

Table 5.24 – System performance at dimension 75

As compared to *Table 5.23*, *Table 5.24* has more of the conditional dualcenter systems being statistically significant. The three evaluation dualcenter systems that perform significantly better than any of the regular dualcenter systems and CMA-ES share interesting similarities. It is evident that they all switch between *1.0-0.0* (CMA-ES) and regular dualcenter at sampling ratios *0.25-0.75, 0.5-0.5 and 0.75-0.25* (cma:EH, cma:eh & CMA:eh). The ratio among the best performing evaluation dualcenter systems that comes as a surprise is *0.25-0.75* (cma:EH) since the regular dualcenter system at that ratio has previously been shown to perform worse or the same as CMA-ES.

Based on this data we can conclude that at high dimension, aggressive dualcenter sampling ratios such as *0.25-0.75* (cma:EH) can be successful if there is reliable information used to determine the best point to be aggressive with elite center sampling or to default back to *1.0-0.0* (CMA-ES).

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **dimension** | **system** | **original EDA better** | **elite history EDA better** | **means** | **group** |
| 100 | evaluation | CMA-ES | CMA:eh | 2.943785 | a |
| 100 | evaluation | CMA-ES | cma:EH | 3.252008 | a |
| 100 | evaluation | CMA-ES | cma:eh | 3.494681 | a |
| 100 | evaluation | CMA-ES | EH | 3.662443 | b |
| 100 | rank-sum | CMA:eh | cma:eh | 5.359464 | bc |
| 100 | rank-sum | CMA:eh | cma:EH | 5.993764 | cd |
| 100 | CMA:eh | - | - | 6.088878 | cd |
| 100 | rank-sum | cma:eh | cma:EH | 8.361673 | def |
| 100 | cma:eh | - | - | 9.298715 | ef |
| 100 | CMA-ES | - | - | 10.46242 | fg |
| 100 | cma:EH | - | - | 15.10568 | g |
| 100 | EH | - | - | 40.28188 | h |

Table 5.25 – System performance at dimension 100

*Table 5.25* shows that dimension 100 reiterates the results seen in *Table 5.24*. One interesting difference between them is that an additional evaluation dualcenter system joins the group which perform significantly better than the regular dualcenter systems. This conditional dualcenter system switches between *1.0-0.0* (CMA-ES) and *0.0-1.0* pure elite center sampling (EH) based on which of the two centers is more fit. This once again proves that the elite history EDA can be powerful at high dimensions when used with conditional metrics.

To summarize the results section, we must revisit several points. Of most importance, out of the 6 multimodal functions chosen for the experiments, 4 of them performed better than CMA-ES. The multimodal functions in which DC-CMA-ES did not improve or decrease performance on is Levy and Rosenbrock, shown in *Equation 4.4* and *Equation 4.6* respectively. Since DC-CMA-ES was designed to improve performance on complex multimodal landscapes we believe that this is not a surprise given that the Levy and Rosenbrock (shown in *Figure 5.1* and *Figure 5.2*) have few local minima as compared to the other functions such as Rastrigin (shown in *Figure 5.3*).

A close up of a map

Description automatically generated

Figure 5.2 – 2-D plot of the Levy test function [14]

A picture containing text

Description automatically generated

Figure 5.3 – 2-D plot of the Rosenbrock test function [14]

A screenshot of a cell phone

Description automatically generated

Figure 5.4 – 2-D plot of the Rastrigin test function [14]

It’s important to note that the Rosenbrock function is multimodal as dimensionality increases from 4 onwards [15]. Even though the 2-dimensional plot shown in *Figure 5.2* is in fact unimodal, it is still useful to compare it with 2-dimensional multimodal functions. Since all experiments in this thesis were run on dimensions > 4, we consider it as a multimodal function.

However, the 2 unimodal functions defined in *Equation 4.7*and *Equation 4.8* also showed improved performance with DC-CMA-ES. This could be attributed to the elitist nature of the elite center in addition to the conditional metrics discussed in the methodology.

From the main effects, the fact that the systems where both EDAs participate in the sampling process outperforms the systems where just one EDA participates in sampling (CMA-ES & EH), suggests that using two EDAs is a beneficial approach to solving multimodal problems.

In terms of the problem dimension, DC-CMA-ES begins to show improvements over CMA-ES as early as dimension 10 and even up to dimension 100. As dimensionality increases, solving these functions becomes more difficult. However, it was proven that the conditional metric and the rank-sum metric are useful modifications as they help DC-CMA-ES to maintain its performance in higher dimensions.

Moreover, these conditional metrics applied to DC-CMA-ES solidifies that CMA-ES is a very robust optimization algorithm since the best performing ratios in the conditional DC-CMA-ES are those that default back to *1.0­-0.0* (CMA-ES) when the main center is more fit than the elite center. This setup allows you to extract the power from CMA-ES and combine it with the local-minima avoiding advantage from DC-CMA-ES.

# 

## Conclusion

Improving optimization performance on complex multimodal problems is still an ongoing problem within the field of optimization. This study gave evidence that a synchronized method of performing sampling with two EDAs can mitigate the problem of being trapped within the local optima of multimodal problems.

The system discussed in this thesis is a modification of CMA-ES in which an additional Estimation Distribution Algorithm (EDA) was used within the sampling process and updated differently than the original EDA. The model of this additional EDA took inspiration from other successful evolutionary techniques within Evolution Strategies, such as elitism, age and weighted-intermediate recombination.

It is important to note that since the EDAs share the same covariance matrix and step-size, every solution sampled by this additional EDA is tied back to the original EDA so that the use of a single covariance matrix can be maintained to help synchronize the two EDAs. This is beneficial because performing an eigen decomposition on a covariance matrix has a time complexity of *O(n3)*. Having to perform this decomposition on anextra covariance matrix would double the time complexity of the algorithm.

DC-CMA-ES can also be thought of as a layer on top of CMA-ES or a controlled variant of CMA-ES because it can perform synchronized sampling at different ratios. If the original EDA is set to do all sampling, then it is identical to CMA-ES. Thus, if the ratios were chosen to favor the original EDA but still allow the additional EDA to perform some sampling then DC-CMA-ES is unlikely to hinder the power of CMA-ES.

For example, let the main center sample ¾ of the current population and the elite center sample the remaining ¼. Since the main EDA selects the best solutions to update the center; if all solutions sampled from the elite center have poor fitness, they would be thrown away having little or no effect on the convergence of the algorithm.

It was found that sharing the sampling process evenly or having it slightly favor the original EDA showed the best results. Since the systems in which both EDAs participated in sampling outperformed those systems where either one of the EDAs solely performed sampling, we can conclude that using more than one EDA to sample solutions is an effective technique in improving the performance of the optimization algorithms which use EDAs.

We also learned that ratios favoring the additional EDA can benefit performance as long as the algorithm can cast good judgement about which EDA to sample most of its solutions from. This judgement was formulated by two checks known as the evaluation metric and the rank-sum metric. The latter of these checks essentially looks at the quality of solutions produced and decides. The evaluation metric performs a function evaluation on both centers and is not beneficial at low dimensions but is very much so at higher dimensions. This is because this metric cost extra evaluations per generation and is only worthwhile when the problem becomes more complex to solve.

Overall, DC-CMA-ES outperformed CMA-ES on 4 out of 6 multimodal test functions and 2 out of 2 unimodal test functions. The two multimodal test functions which DC-CMA-ES could not improve was Levy and Rosenbrock, which have the fewest multimodal characteristics out of the group. DC-CMA-ES performed best on multimodal test functions which contain areas of many widespread local optima such Rastrigin and Griewank. Although targeted mainly for multimodal problems, it still managed to improve performance in solving some simple unimodal functions, Elliptical and Zakharov.

## 6.1 Future Works

Having already mentioned that the two EDAs of DC-CMA-ES share the same covariance matrix for computational efficiency, it would be interesting to see how the system would react if these EDAs used their own covariance matrix to perform sampling. This would potentially increase the sampling success of the additional EDA. Subsequently, the use of separate step-sizes would also be an interesting experiment. However, the functionality of the additional EDA may be compromised since it partially relies on solutions coming from the main EDA. In addition to this, this collapses the overlapping model between the two EDAs, essentially running two independent systems.

In this paper, a fixed sliding window-size of 10 was used for updating the additional EDA. We believe that this window-size should be dynamic based on the dimensionality of the problem. This may explain why DC-CMA-ES did not show improvements at dimension 5 but showed improvements from dimension 10 and onward. Since dimension 5 is not particularly difficult to solve, an average of the previous 10 best solutions may be unnecessary.

As mentioned in section *4.2 Experimental Setup*, the CRN simulation was not analyzed in order to prove there is a reduction in variance between DC-CMA-ES and CMA-ES. This is likely to be our next course of research.

Lastly, an interesting research avenue would be to push the system beyond two EDAs. The only challenge regarding this is coming up with a different method for building each additional EDA. Moreover, creating multiple EDAs that have an evolutionary nature and provide good posterior probabilities about where in the problem landscape to perform sampling will be a challenging endeavor.

# Bibliography

|  |  |
| --- | --- |
| [1] | N. Hansen and A. Ostermeier, "Adapting arbitrary normal mutation distributions in evolution strategies: the covariance matrix adaptation," in *IEEE International Conference on Evolutionary Computation*, Nagoya, 1996. |
| [2] | T. Bäck, C. Foussette and P. Krause, Contemporary Evolution Strategies, Berlin: Springer Berlin, 2013. |
| [3] | A. Auger and N. Hansen, "A restart CMA evolution strategy with increasing population size," in *IEEE Congress on Evolutionary Computation (CEC'05)*, Edinburgh, 2005. |
| [4] | O. M. Shir, M. Emmerich and T. Back, "Adaptive Niche Radii and Niche Shapes Approaches for Niching with the CMA-ES," *Evolutionary Computation,* vol. 18, no. 1, pp. 97-99, 2010. |
| [5] | Z. Xu, H. Iizuka and M. Yamamoto, "Attraction bsin sphere estimation approach for niching CMA-ES," *Soft Computing,* vol. 21, no. 5, pp. 1327-1345, 2017. |
| [6] | A. Agapie, "Estimation of distribution algorithms on non-separable problems," *International Journal of Computer Mechanics,* vol. 87, no. 3, p. 1, 2010. |
| [7] | A. Auger and N. Hansen, "Tutorial CMA-ES - Evolutionary Strategies and Covariance Matrix," *Proceeding of the Fifteenth Annual Conference Companion on Genetic and Evolutionary Computation Conference Companion - Gecco 13 Companion,* pp. 1-83, 2013. |
| [8] | H.-P. Schwefel, "Numberische Optimisrung von Computer-Modellen Mittles der Evolution-strategie," *Journal of Applied Mathematics and Mechanics,* vol. 60, no. 5, pp. 1-272, 1977. |
| [9] | N. Hansen and A. Ostermeier, "Completey Derandomized Self-Adaptation in Evolution Strategies," *Evolutionary Computation,* vol. 9, no. 2, pp. 1-195, 2001. |
| [10] | G. Rudolph, "Convergence Properties of Evolutionary Algorithms," 1997. |
| [11] | T. Bäck, D. B. Fogel and Z. Michalewicz, Evolutionary Computation 2: Advanced Algorithms and Operators, New York: Taylor & Francis, 2000. |
| [12] | A. M. Law, Simulation Modeling & Analysis, 4th ed., New York: McGraw-Hill, 2007. |
| [13] | P. Lecuyer, "Efficiency improvement and variance reduction," in *Proceedings of Winter Simulation Conference*, Montreal, 1994. |
| [14] | S. Surjanovic and D. Bingham, "Virtual Library of Simulation Experiments," Simon Fraser University, August 2017. [Online]. Available: https://www.sfu.ca/~ssurjano/index.html. [Accessed 4 November 2019]. |
| [15] | S. Kok and C. Sandrock, "Locating and Characterizing the Stationary Points of the Extended Rosenbrock Function," *Evolutionary Computation,* vol. 17, no. 3, pp. 437-453, 2009. |
| [16] | C. Igel, N. Suttorp and N. Hansen, "A computational efficient covariance matrix update and a (1+1)-CMA for evolution strategies," *Proceedings of the 8th annual conference on Genetic and Evolutionary Computation,* pp. 453-460, 2006. |
| [17] | G. A. Jastrebski, "Improving evolution strategies through active covariance matrix adaptation," *Master's thesis, Faculty of Computer Science, Dalhousie University,* 2005. |
| [18] | B. Tang, "Orthogonal array-based latin hypercubes.," *J. Am. Stat. Assoc.,* vol. 88, no. 424, pp. 1392-1397, 1993. |
| [19] | H.-G. Beyer and H.-P. Schwefel, "Evolution strategies: a comprehensive introduction," *Natural Computing,* vol. 1, no. 1, pp. 3-52, 2002. |

Appendix A

**Modern CMA-ES Variants**

### LS-CMA-ES

The LS-CMA-ES algorithm is a CMA-ES variant that updates the covariance matrix based on an inverse Hessian matrix that is estimated via a solution to the appropriate least-squares estimation problem. To solve this least-squares estimation problem, the algorithm collects an archive of tuples *(x, f(x))* where *x* is a candidate solution and *f(x)* is the fitness value of that candidate solution [2]. As a result, the least-squares estimation problem is defined by the following minimization task:

Equation A.1

The result of this minimization in *Equation A.1* produces two estimators, for the gradient and for the Hessian matrix. The Taylor series expansion up to the quadratic term only provides an estimation of the true fitness landscape at *x0*­. Therefore, it is of interest to obtain an error measure of the estimate to determine if should be used for the covariance matrix update. The error measure is shown in *Equation A.2* [2].

Equation A.2

Since solving *Equation A.2* and computing H-1 uses numerical methods algorithms that have a time complexity of *O*(*n*6), executing this step at each generation is not ideal for performance. As a result, LS-CMA-ES has two different modes known as LS and CMA, used for performing the covariance matrix update [2].

In LS mode, *Equation A.2* is executed once every *nupd* generations. Additionally, if the error measure falls below a defined threshold, *Q*t, the covariance matrix until the new update after another *nupd* generations. If Q > Qt, the algorithm switches into CMA mode. In CMA mode the covariance matrix is updated using the rank-one update (the 2nd term in *Equation 2.12*). In this algorithm, sampling offspring from the parent is shown in *Equation A.3* below.

Equation A.3

A global step size *σ* is used and updated via mutative step size adaptation. For example, if *b* is the index of the best offspring after selection, then the new global step size is updated as Finally, the covariance matrix is updated using the rank-one update (the 2nd term in *Equation 2.12)*. It is important to note that the evolution path *pc* is still updated regardless of which mode the algorithm is in so that the covariance matrix is always up-to-date before switching into or staying in CMA mode [2]. The recommended exogenous strategy parameters are shown below:

### (1+1)-Cholesky-CMA-ES

The main property of this CMA-ES variant is that the covariance matrix is implicitly updated without using an eigen decomposition. As the name suggests, the algorithm uses a Cholesky decomposition instead. This method reduces the computational complexity in each generation from *O*(*n3*) to *O*(*n*2) [2]. It is proven that the Cholesky factors *A* can be updated without explicit knowledge of the covariance matrix [16]. The theorem will be discussed without the proof. However, consider this lemma which is required for the proof of the following theorem [2]. For any vector and , the following equation holds:

Equation A.4

The theorem then states, let be a symmetric positive definite matrix with the Cholesky decomposition where . Additionally, let defined the update of *C* with . Finally, the update Cholesky factor is given below [2]:

Equation A.5

An offspring is produced via the parent *x* with the following equation:

Equation A.6

Using the previously discussed theorem, the Cholesky factor *A* is adapted with a constant exogenous strategy parameter denoted by *ca*.

Equation A.7

The adaptation show in *Equation A.7* is executed if the value of (shown in *Equation A.8*) is less than a threshold value, denoted *pt*.

Equation A.8

The global step size adaptation in this algorithm possesses similarities with the 1/5th success rule of the (1+1)-ES shown earlier in *Equation 2.2*. Therefore, if the new offspring has a better fitness value than that of its parent, *λs* = 1, otherwise *λs* = 0. These measures of success are accumulated over generations by using the parameter *cp* as a learning rate as shown in *Equation A.8* [2]. Now, with the success rate , the global step size is updated as follows:

Equation A.9

where is the target value for the success rate [2].

The recommended exogenous strategy parameter settings are as follows:

### Active-CMA-ES

The idea behind Active CMA-ES is that the worst offspring should be considered for the adaptation of the covariance matrix by utilizing negative weights. As a result, Active-CMA-ES modifies the covariance matrix update of *(µw, λ)* - CMA-ES as follows [2]:

Equation A.10

Equation A.11

The parameter *cc* is also modified as . The parameter *β* has been tuned with empirical investigation, described in [17]. The parameter was chosen as a compromise between improving the convergence velocity and ensuring the covariance matrix *C* is positive definite, in order to make the system more robust [2].

### (µ, λ) - CMSA-ES

*(µ, λ)*-CMSA-ES aims to reduce the number of exogenous strategy parameters by reintroducing self-adaptation of the global step size *σ*. This simplifies the number of exogenous strategy parameters from five in *(µw, λ)-*CMA-ES down to two [2]. Offspring solutions each have their own individual step sizes which are calculated using the global step size *σ* as follows [2]:

The *µ* best offspring go through recombination with identical weights and is applied to the vectors and for to obtain the vectors and the new global step size *σ*. Therefore, the new parent is updated as . The vector is necessary for updating the covariance matrix *C*, using the learning rate as follows [2]:

Equation A.12

The recommended settings of the exogenous strategy parameters are:

### sep-CMA-ES

This CMA-ES variant reduces the space and time complexity of the *(µw, λ)-*CMA-ES algorithm down to *O*(*n*) as compared to the original *O*(*n3*). The reason for this increase in performance is because it uses a diagonal matrix D which only contains the square root of the main diagonal elements of the covariance matrix *C*, along its own diagonal. This means that no computationally expensive eigen decompositions need to be done. The drawback is that this matrix is unable to generate correlated mutations like a covariance matrix. Hence, the covariance matrix update is modified from *(µw, λ)-*CMA-ES as follows [2]:

Equation A.13

The learning rate is increased because of the reduced complexity of the covariance matrix and is newly determined by:

Equation A.14

All other parameters from *(µw, λ)-*CMA-ES remain unchanged.

### (1+1)-Active-CMA-ES

This algorithm is an extension of the *(1+1)-*Cholesky-CMA-ES that uses the Active-CMA-ES ideology of taking the worst offspring into account for the covariance matrix update. Rather than using the Cholesky decomposition for the covariance matrix , this algorithm uses the Cholesky factor A and its inverse A-1 [2]. Thus, the theorem discussed in section *2.2.7* is modified. Let be a positive definite symmetric matrix with Cholesky decomposition . In addition, let be the update equation for *C* where . Let with and let be the Cholesky decomposition of the updated covariance matrix [2]. Then is updated as follows:

Equation A.15

and is updated as follows:

Equation A.16

New offspring are generated the same way, . The success rate from *(1+1)-*Cholesky-CMA-ES is now modified with a condition as follows:

Equation A.17

With this new success rate, a dampening parameter and the target success rate *pt*, the global step size is now:

Equation A.18

The target success rate is set as , which is similar to the 1/5th success rule employed by the *(1+1)-*ES. If the fitness value of the offspring is better than that of the parent, then a positive Cholesky update is done. *(1+1)*-Active-CMA-ES relies on a search path *s*, which accumulates the successful mutation steps with a learning rate *c* [2]:

Equation A.19

With the constant and vector , the positive update of *A* and *A-1* are defined as follows:

Equation A.20

Equation A.21

The variables *a* and *b* in *Equation A.20* and *Equation A.21* are defined as follows:

In Active-CMA-ES, the *λ* - *µ* worst solutions are used in the covariance matrix update. With the strategy introduced in this algorithm, previous fitness values are stored and checked to see if the offspring is inferior to its parent. If the offspring is inferior to its parent, then the positive updates in *Equation A.20* and *Equation A.21* are done using *a* and *b*. However, the transformed search path and the vector *z* is used for the negative update [2]:

It is important that holds for the constant , in order to ensure that the covariance matrix is positive definite. Additionally, if the value of , the convergence of the algorithm can be come very unstable. To stop this effect, when , is given and upper bound of [2]. The recommended settings of the exogenous strategy parameters are:

### SPO-CMA-ES

This algorithm is a restart-version of the (µw, λ)-CMA-ES that uses sequential parameter optimization (SPO) [11], to optimize the exogenous strategy parameters within the algorithm. SPO uses *design of experiments* (DoE) methods and *design and analysis of computer experiments* (DACE) [2]. This algorithm is concerned with the exogenous strategy parameters such as the number of offspring λ, the value of the initial step size and the selection pressure .

The first step of the algorithm is *latin hypercube sampling* (LHS) [18] in which an initial design of experiments (DoE) for the exogenous strategy parameters is created. In the second step, independent runs of *(µw, λ)*-CMA-ES are done using the parameters that were created in step one of the DoE plan. The best solution along with its fitness value from each independent run is stored in a set, *Y*. This first phase of the algorithm is referred to as the *exploration phase* and is followed by the *exploitation phase*.

The exploitation phase is repeated until the function evaluation budget is reached. A performance measure *y* is computed for each of the results in *Y*. Based on the performance measure values and the corresponding exogenous parameter sets from the DoE, a Kriging model *M* is trained. Subsequently, this model is used to find a new design point by running an optimizing the Kriging model and using the resulting point. This new design point *d* is then added to experimental plan *D* and the algorithm loops again [2].

Appendix B

**Full Statistical Significance Grouping of System Performance**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **system** | **original EDA better** | **elite history EDA better** | **means** | **group** |
| rank-sum | CMA:eh | cma:eh | 0.196919405 | i |
| rank-sum | CMA-ES | cma:eh | 0.222463855 | hi |
| rank-sum | CMA:eh | cma:EH | 0.228289752 | ghi |
| evaluation | CMA-ES | cma:EH | 0.229297767 | ghi |
| evaluation | CMA-ES | CMA:eh | 0.232724025 | gh |
| evaluation | CMA-ES | EH | 0.23457734 | gh |
| CMA:eh | - | - | 0.237818369 | gh |
| cma:eh | - | - | 0.264913244 | fg |
| evaluation | CMA:eh | cma:eh | 0.287354569 | ef |
| evaluation | CMA:eh | CMA:eh | 0.29230512 | ef |
| rank-sum | cma:eh | cma:EH | 0.317761093 | e |
| evaluation | CMA:eh | EH | 0.318215438 | de |
| evaluation | cma:eh | cma:EH | 0.372839137 | cd |
| evaluation | cma:eh | EH | 0.38650352 | c |
| CMA-ES | - | - | 0.461258262 | b |
| cma:EH | - | - | 0.47168142 | b |
| EH | - | - | 0.980726836 | a |

B.1 – Main Effects of System Performance

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **function** | **system** | **original EDA better** | **elite history EDA better** | **means** | **group** |
| ackley | rank-sum | CMA:eh | cma:eh | 0.0103398 | H |
| ackley | rank-sum | CMA:eh | cma:EH | 0.01423121 | GH |
| ackley | evaluation | CMA-ES | cma:EH | 0.01480781 | FGH |
| ackley | cma:eh | - | - | 0.01511125 | FGH |
| ackley | evaluation | CMA-ES | CMA:eh | 0.01586054 | FGH |
| ackley | evaluation | CMA-ES | EH | 0.01589768 | FGH |
| ackley | CMA:eh | - | - | 0.01832632 | FGH |
| ackley | rank-sum | cma:eh | cma:EH | 0.01846255 | FGH |
| ackley | evaluation | CMA-ES | cma:eh | 0.01877164 | FGH |
| ackley | evaluation | CMA:eh | cma:eh | 0.0209709 | FG |
| ackley | evaluation | CMA:eh | cma:EH | 0.02194399 | EFG |
| ackley | evaluation | CMA:eh | EH | 0.02474928 | DEFG |
| ackley | evaluation | cma:eh | EH | 0.02785522 | CDEFG |
| ackley | evaluation | cma:eh | cma:EH | 0.02867824 | CDEF |
| ackley | CMA-ES | - | - | 0.04552343 | BCD |
| ackley | cma:EH | - | - | 0.05040201 | zABC |
| ackley | EH | - | - | 0.17258676 | qrstuv |

B.2 – System Performance on Ackley

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **function** | **system** | **original EDA better** | **elite history EDA better** | **means** | **group** |
| griewank | rank-sum | CMA:eh | cma:eh | 0.000126535 | OPQRST |
| griewank | evaluation | CMA-ES | cma:eh | 0.00015207 | NOPQRS |
| griewank | evaluation | CMA-ES | cma:EH | 0.0001623 | NOPQRS |
| griewank | rank-sum | CMA:eh | cma:EH | 0.00016949 | NOPQRS |
| griewank | cma:eh | - | - | 0.000179814 | NOPQR |
| griewank | rank-sum | cma:eh | cma:EH | 0.000195362 | NOPQR |
| griewank | CMA:eh | - | - | 0.000196035 | MNOPQR |
| griewank | evaluation | CMA-ES | CMA:eh | 0.000209169 | LMNOPQ |
| griewank | evaluation | CMA-ES | EH | 0.000234601 | LMNOP |
| griewank | evaluation | CMA:eh | cma:eh | 0.000326331 | KLMN |
| griewank | evaluation | CMA:eh | cma:EH | 0.000327066 | KLMN |
| griewank | evaluation | CMA:eh | EH | 0.000367983 | KLMN |
| griewank | evaluation | cma:eh | EH | 0.000457673 | KLM |
| griewank | evaluation | cma:eh | cma:EH | 0.000492237 | JKL |
| griewank | cma:EH | - | - | 0.000599728 | JK |
| griewank | CMA-ES | - | - | 0.00111369 | J |
| griewank | EH | - | - | 0.003230298 | I |

B.3 – System performance on Griewank

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **function** | **system** | **original EDA better** | **elite history EDA better** | **means** | **group** |
| levy | CMA:eh | - | - | 0.042312906 | BCDE |
| levy | CMA-ES | - | - | 0.048092132 | ABCD |
| levy | rank-sum | CMA:eh | cma:eh | 0.058640452 | yzAB |
| levy | evaluation | CMA-ES | CMA:eh | 0.068530702 | xyzAB |
| levy | evaluation | CMA-ES | cma:eh | 0.069312447 | xyzAB |
| levy | evaluation | CMA:eh | cma:eh | 0.077151729 | xyzAB |
| levy | cma:eh | - | - | 0.094950956 | vwxyz |
| levy | rank-sum | CMA:eh | cma:EH | 0.09594786 | vwxy |
| levy | evaluation | CMA-ES | cma:EH | 0.096831509 | vwxy |
| levy | rank-sum | cma:eh | cma:EH | 0.110784515 | uvwx |
| levy | evaluation | CMA:eh | cma:EH | 0.119693404 | tuvwx |
| levy | evaluation | CMA-ES | EH | 0.122342305 | tuvwx |
| levy | evaluation | CMA:eh | EH | 0.124427824 | tuvwx |
| levy | evaluation | cma:eh | EH | 0.183294987 | qrstu |
| levy | evaluation | cma:eh | cma:EH | 0.190381567 | qrst |
| levy | cma:EH | - | - | 0.212533793 | pqrst |
| levy | EH | - | - | 0.35366576 | op |

B.4 – System Performance on Levy

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **function** | **system** | **original EDA better** | **elite window EDA better** | **means** | **groups** |
| schwefel | evaluation | CMA-ES | EH | 3.06771E-05 | X |
| schwefel | rank-sum | CMA:eh | cma:EH | 3.37462E-05 | X |
| schwefel | rank-sum | cma:eh | cma:EH | 3.65182E-05 | WX |
| schwefel | rank-sum | CMA:eh | cma:eh | 3.65884E-05 | WX |
| schwefel | evaluation | CMA-ES | CMA:eh | 4.50668E-05 | VWX |
| schwefel | evaluation | CMA-ES | cma:EH | 4.76935E-05 | UVWX |
| schwefel | evaluation | CMA-ES | cma:eh | 5.03215E-05 | TUVWX |
| schwefel | evaluation | cma:EH | cma:eh | 6.85641E-05 | STUVWX |
| schwefel | evaluation | CMA:eh | EH | 7.7823E-05 | RSTUVWX |
| schwefel | evaluation | CMA:eh | cma:eh | 9.13708E-05 | QRSTUVW |
| schwefel | cma:EH | - | - | 0.000101271 | PQRSTUV |
| schwefel | evaluation | CMA:eh | cma:EH | 0.000102031 | PQRSTUV |
| schwefel | cma:eh | - | - | 0.000116985 | PQRSTUV |
| schwefel | evaluation | cma:eh | EH | 0.000121029 | OPQRSTU |
| schwefel | CMA:eh | - | - | 0.000208435 | LMNOPQ |
| schwefel | CMA-ES | - | - | 0.000297597 | KLMNO |
| schwefel | EH | - | - | 0.000477601 | KL |

B.5 – System Performance on Schwefel

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **function** | **system** | **original EDA better** | **elite history EDA better** | **means** | **group** |
| rastrigin | rank-sum | CMA:eh | cma:eh | 46.6110405 | bcdefghij |
| rastrigin | rank-sum | CMA:eh | cma:EH | 46.6180427 | bcdefghij |
| rastrigin | rank-sum | cma:eh | cma:EH | 47.4409174 | bcdefghi |
| rastrigin | evaluation | CMA:eh | cma:EH | 48.1678292 | bcdefghi |
| rastrigin | evaluation | CMA:eh | cma:eh | 49.2731794 | bcdefgh |
| rastrigin | evaluation | CMA-ES | cma:EH | 49.9292556 | bcdefg |
| rastrigin | cma:eh | - | - | 52.1053294 | bcdefg |
| rastrigin | evaluation | CMA-ES | cma:eh | 52.8324448 | bcdefg |
| rastrigin | CMA:eh | - | - | 53.2976456 | bcdefg |
| rastrigin | evaluation | cma:eh | cma:EH | 53.502552 | bcdefg |
| rastrigin | evaluation | CMA:eh | EH | 54.2896083 | bcdefg |
| rastrigin | cma:EH | - | - | 54.3539318 | bcdefg |
| rastrigin | evaluation | CMA-ES | CMA:eh | 56.2001293 | bcdef |
| rastrigin | evaluation | CMA-ES | EH | 56.6764121 | bcde |
| rastrigin | evaluation | cma:eh | EH | 59.4797021 | bcd |
| rastrigin | EH | - | - | 60.9355094 | bc |
| rastrigin | CMA-ES | - | - | 147.176311 | a |

B.6 – System performance on Rastrigin

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **function** | **system** | **original EDA better** | **elite window EDA better** | **means** | **group** |
| elliptical | evaluation | CMA-ES | cma:eh | 0.077045025 | xyzAB |
| elliptical | evaluation | CMA-ES | CMA:eh | 0.08806114 | wxyzA |
| elliptical | evaluation | CMA-ES | EH | 0.088961955 | wxyzA |
| elliptical | rank-sum | CMA:eh | cma:eh | 0.09671144 | vwxy |
| elliptical | CMA:eh | - | - | 0.097608199 | vwxy |
| elliptical | evaluation | CMA-ES | cma:EH | 0.106912199 | uvwxy |
| elliptical | cma:eh | - | - | 0.145873704 | stuvw |
| elliptical | evaluation | CMA:eh | cma:EH | 0.146826973 | stuvw |
| elliptical | rank-sum | CMA:eh | cma:EH | 0.14776988 | stuvw |
| elliptical | rank-sum | cma:eh | cma:EH | 0.171752866 | rstuv |
| elliptical | evaluation | CMA:eh | cma:eh | 0.173988598 | qrstuv |
| elliptical | evaluation | CMA:eh | EH | 0.186460778 | qrstu |
| elliptical | CMA-ES | - | - | 0.239338247 | pqrs |
| elliptical | evaluation | cma:eh | EH | 0.287194417 | opqr |
| elliptical | evaluation | cma:eh | cma:EH | 0.289695365 | opq |
| elliptical | cma:EH | - | - | 0.461751442 | o |
| elliptical | EH | - | - | 1.410394786 | n |

B.7 – System Performance on Elliptical

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **function** | **system** | **original EDA better** | **elite history EDA better** | **means** | **group** |
| rosenbrock | evaluation | CMA-ES | EH | 30.55174864 | j |
| rosenbrock | evaluation | CMA-ES | cma:EH | 30.58473897 | j |
| rosenbrock | evaluation | CMA-ES | cma:eh | 31.41133945 | ij |
| rosenbrock | evaluation | CMA-ES | CMA:eh | 32.05685651 | hij |
| rosenbrock | CMA:eh | - | - | 35.3719264 | ghij |
| rosenbrock | evaluation | CMA:eh | cma:EH | 36.7522971 | fghij |
| rosenbrock | CMA-ES | - | - | 36.86061248 | efghij |
| rosenbrock | rank-sum | CMA:eh | cma:EH | 37.33553933 | efghij |
| rosenbrock | evaluation | cma:eh | EH | 39.07435192 | defghij |
| rosenbrock | evaluation | CMA:eh | cma:eh | 39.44106963 | defghij |
| rosenbrock | rank-sum | CMA:eh | cma:eh | 39.70366715 | cdefghij |
| rosenbrock | evaluation | CMA:eh | EH | 41.57827142 | cdefghij |
| rosenbrock | evaluation | cma:eh | cma:EH | 42.04211074 | cdefghij |
| rosenbrock | cma:eh | - | - | 45.60300689 | bcdefghij |
| rosenbrock | rank-sum | cma:eh | cma:EH | 50.02356048 | bcdefg |
| rosenbrock | cma:EH | - | - | 67.23614793 | b |
| rosenbrock | EH | - | - | 112.6584486 | a |

B.8 – System Performance on Rosenbrock

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **function** | **system** | **original EDA better** | **elite history EDA better** | **means** | **group** |
| zakharov | CMA:eh | - | - | 9.503907295 | m |
| zakharov | cma:eh | - | - | 9.579555485 | m |
| zakharov | rank-sum | cma:eh | cma:EH | 10.00664763 | lm |
| zakharov | rank-sum | CMA:eh | cma:EH | 10.01420532 | lm |
| zakharov | rank-sum | CMA:eh | cma:eh | 10.12630348 | lm |
| zakharov | evaluation | CMA-ES | EH | 11.93447709 | klm |
| zakharov | evaluation | cma:EH | CMA:eh | 12.58019552 | klm |
| zakharov | evaluation | CMA-ES | cma:EH | 12.72348326 | klm |
| zakharov | evaluation | CMA:eh | EH | 13.28066326 | klm |
| zakharov | cma:EH | - | - | 13.29205583 | klm |
| zakharov | evaluation | CMA-ES | cma:eh | 13.49306437 | klm |
| zakharov | evaluation | CMA:eh | cma:eh | 13.5408201 | klm |
| zakharov | evaluation | CMA-ES | CMA:eh | 14.99357513 | klm |
| zakharov | evaluation | cma:eh | cma:EH | 15.48952502 | kl |
| zakharov | evaluation | cma:eh | EH | 15.7023013 | kl |
| zakharov | cma:EH | - | - | 15.71439568 | kl |
| zakharov | CMA-ES | - | - | 16.24145573 | k |

B.9 – System Performance on Zakharov

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **dimension** | **system** | **original EDA better** | **elite history EDA better** | **means** | **group** |
| 5 | cma:eh | - | - | 0.002805 | N |
| 5 | rank-sum | CMA:eh | cma:eh | 0.002858 | N |
| 5 | rank-sum | CMA:eh | cma:EH | 0.003012 | N |
| 5 | cma:EH | - | - | 0.003117 | N |
| 5 | EH | - | - | 0.003442 | N |
| 5 | CMA:eh | - | - | 0.003644 | N |
| 5 | CMA-ES | - | - | 0.004102 | N |
| 5 | rank-sum | cma:eh | cma:EH | 0.004289 | N |
| 5 | evaluation | CMA-ES | EH | 0.009169 | M |
| 5 | evaluation | CMA:eh | EH | 0.00918 | M |
| 5 | evaluation | CMA:eh | cma:EH | 0.009465 | M |
| 5 | evaluation | cma:eh | EH | 0.009818 | M |
| 5 | evaluation | CMA:eh | cma:eh | 0.009834 | M |
| 5 | evaluation | CMA-ES | cma:EH | 0.009855 | M |
| 5 | evaluation | cma:eh | cma:EH | 0.009964 | M |
| 5 | evaluation | CMA-ES | cma:eh | 0.010888 | LM |
| 5 | evaluation | CMA-ES | CMA:eh | 0.011237 | LM |

B.10 – System Performance on Dimension 5

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **dimension** | **system** | **original EDA better** | **elite history EDA better** | **means** | **group** |
| 10 | rank-sum | CMA:eh | cma:eh | 0.010649 | LM |
| 10 | rank-sum | CMA:eh | cma:EH | 0.011782 | LM |
| 10 | cma:eh | - | - | 0.013229 | KLM |
| 10 | CMA:eh | - | - | 0.013727 | KLM |
| 10 | cma:EH | - | - | 0.018359 | JKL |
| 10 | rank-sum | cma:eh | cma:EH | 0.020118 | JK |
| 10 | EH | - | - | 0.024983 | IJ |
| 10 | evaluation | CMA-ES | cma:EH | 0.026157 | IJ |
| 10 | evaluation | CMA-ES | cma:eh | 0.026374 | IJ |
| 10 | evaluation | CMA-ES | EH | 0.027327 | IJ |
| 10 | evaluation | CMA:eh | EH | 0.027562 | IJ |
| 10 | evaluation | CMA:eh | cma:EH | 0.028341 | IJ |
| 10 | evaluation | cma:eh | EH | 0.02985 | IJ |
| 10 | evaluation | CMA:eh | cma:eh | 0.03063 | IJ |
| 10 | evaluation | cma:eh | cma:EH | 0.032124 | IJ |
| 10 | CMA-ES | - | - | 0.03433 | I |
| 10 | evaluation | CMA-ES | CMA:eh | 0.034572 | I |

B.11 – System Performance on Dimension 10

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **dimension** | **system** | **original EDA better** | **elite history EDA better** | **means** | **group** |
| 25 | rank-sum | CMA:eh | cma:eh | 0.090101 | H |
| 25 | rank-sum | CMA:eh | cma:EH | 0.101983 | GH |
| 25 | evaluation | CMA-ES | cma:eh | 0.112087 | FGH |
| 25 | CMA:eh | - | - | 0.116607 | FGH |
| 25 | evaluation | CMA-ES | EH | 0.120512 | EFGH |
| 25 | evaluation | CMA-ES | cma:EH | 0.129862 | EFGH |
| 25 | evaluation | CMA-ES | CMA:eh | 0.13016 | EFGH |
| 25 | CMA:eh | CMA:eh | cma:EH | 0.132566 | EFGH |
| 25 | cma:eh | - | - | 0.137899 | DEFGH |
| 25 | rank-sum | cma:eh | cma:EH | 0.1442 | DEFG |
| 25 | evaluation | CMA:eh | cma:eh | 0.157517 | CDEFG |
| 25 | evaluation | CMA:eh | EH | 0.182599 | CDEF |
| 25 | evaluation | cma:eh | EH | 0.197425 | CDE |
| 25 | evaluation | cma:eh | cma:EH | 0.220933 | BCD |
| 25 | cma:EH | - | - | 0.248323 | BC |
| 25 | CMA-ES | - | - | 0.325685 | AB |
| 25 | EH | - | - | 0.591967 | wxyz |

B.12 – System Performance on Dimension 25

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **dimension** | **system** | **original EDA better** | **elite history EDA better** | **means** | **group** |
| 50 | evaluation | CMA-ES | cma:eh | 0.435083 | zA |
| 50 | evaluation | CMA-ES | CMA:eh | 0.479242 | yzA |
| 50 | evaluation | CMA-ES | EH | 0.503505 | yzA |
| 50 | evaluation | CMA-ES | cma:EH | 0.528891 | xyz |
| 50 | evaluation | - | - | 0.585818 | wxyz |
| 50 | evaluation | CMA:eh | cma:eh | 0.678047 | vwxyz |
| 50 | rank-sum | CMA:eh | cma:eh | 0.690185 | uvwxyz |
| 50 | CMA:eh | - | - | 0.698816 | uvwxy |
| 50 | evaluation | CMA:eh | cma:EH | 0.738633 | tuvwxy |
| 50 | rank-sum | CMA:eh | cma:EH | 0.808048 | stuvwx |
| 50 | evaluation | CMA:eh | EH | 0.820638 | stuvwx |
| 50 | evaluation | cma:eh | cma:EH | 0.923587 | rstuvw |
| 50 | evaluation | cma:eh | EH | 1.007965 | rstuv |
| 50 | rank-sum | cma:eh | cma:EH | 1.03152 | rstu |
| 50 | CMA-ES | - | - | 1.30453 | pqr |
| 50 | cma:EH | - | - | 1.762478 | nopq |
| 50 | EH | - | - | 4.083197 | ghi |

B.13 – System Performance on Dimension 50

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **dimension** | **system** | **original EDA better** | **elite history EDA better** | **means** | **group** |
| 75 | evaluation | CMA-ES | CMA:eh | 1.130551 | qrst |
| 75 | evaluation | CMA-ES | cma:eh | 1.170769 | qrs |
| 75 | evaluation | CMA-ES | cma:EH | 1.181608 | qrs |
| 75 | evaluation | CMA-ES | EH | 1.342304 | opqr |
| 75 | rank-sum | CMA:eh | cma:eh | 1.567011 | nopq |
| 75 | evaluation | CMA:eh | cma:EH | 1.718551 | nopq |
| 75 | evaluation | CMA:eh | cma:eh | 1.764421 | nopq |
| 75 | evaluation | CMA:eh | EH | 1.852074 | mnop |
| 75 | CMA:eh | - | - | 2.067891 | lmno |
| 75 | rank-sum | CMA:eh | cma:EH | 2.106456 | lmn |
| 75 | evaluation | cma:eh | cma:EH | 2.315674 | klmn |
| 75 | rank-sum | cma:eh | cma:EH | 2.653593 | jklm |
| 75 | cma:eh | - | - | 2.890998 | ijkl |
| 75 | evaluation | cma:eh | EH | 3.015878 | ijkl |
| 75 | CMA-ES | - | - | 4.134013 | ghi |
| 75 | cma:EH | - | - | 5.622925 | fg |
| 75 | EH | - | - | 14.38816 | b |

B.14 – System Performance on Dimension 75

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **dimension** | **system** | **original EDA better** | **elite history EDA better** | **means** | **group** |
| 100 | evaluation | CMA-ES | CMA:eh | 2.943785 | ijkl |
| 100 | evaluation | CMA-ES | cma:EH | 3.252008 | ijk |
| 100 | evaluation | CMA-ES | cma:eh | 3.494681 | ijk |
| 100 | evaluation | CMA-ES | EH | 3.662443 | hij |
| 100 | evaluation | CMA:eh | cma:eh | 4.286006 | ghi |
| 100 | rank-sum | CMA:eh | cma:eh | 5.359464 | fgh |
| 100 | evaluation | CMA:eh | cma:EH | 5.54501 | fgh |
| 100 | evaluation | CMA:eh | EH | 5.812736 | fg |
| 100 | rank-sum | CMA:eh | cma:EH | 5.993764 | efg |
| 100 | CMA:eh | - | - | 6.088878 | efg |
| 100 | evaluation | cma:eh | cma:EH | 6.839608 | def |
| 100 | evaluation | cma:eh | EH | 6.895625 | def |
| 100 | rank-sum | cma:eh | cma:EH | 8.361673 | cde |
| 100 | cma:eh | - | - | 9.298715 | cd |
| 100 | CMA-ES | - | - | 10.46242 | bc |
| 100 | cma:EH | - | - | 15.10568 | b |
| 100 | EH | - | - | 40.28188 | a |

B.15 – System Performance on Dimension 100