Algorithm Configuration Data Mining for CMA Evolution Strategies

Sander van Rijn, Hao Wang, Bas van Stein and Thomas Bäck.
Introduction

• EA popularity → many variants

• Focus: CMA-ES

• Few combinations are tested
Modular CMA-ES Framework

Selected CMA-ES Modules

Active Update
Elitism
Mirrored Sampling
Orthogonal Sampling
Sequential Selection
Threshold Convergence
Two-Point step-size Adaptation (TPA)
Pairwise Selection
Recombination Weights
Quasi-Gaussian Sampling
Increasing Population

## Modular CMA-ES Framework

### Selected CMA-ES Modules

<table>
<thead>
<tr>
<th>Active Update</th>
<th>[0, 0, 0, 0, 0, 0, 0, 0, 0, 0]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Elitism</td>
<td>[0, 0, 0, 0, 0, 0, 0, 0, 0, 1]</td>
</tr>
<tr>
<td>Mirrored Sampling</td>
<td>[0, 0, 0, 0, 0, 0, 0, 0, 0, 2]</td>
</tr>
<tr>
<td>Orthogonal Sampling</td>
<td>[0, 0, 0, 0, 0, 0, 0, 0, 0, 2]</td>
</tr>
<tr>
<td>Sequential Selection</td>
<td></td>
</tr>
<tr>
<td>Threshold Convergence</td>
<td>[1, 1, 1, 1, 1, 1, 1, 1, 1, 0]</td>
</tr>
<tr>
<td>TPA</td>
<td>[1, 1, 1, 1, 1, 1, 1, 1, 1, 1]</td>
</tr>
<tr>
<td>Pairwise Selection</td>
<td>[1, 1, 1, 1, 1, 1, 1, 1, 1, 2]</td>
</tr>
<tr>
<td>Recombination Weights</td>
<td></td>
</tr>
<tr>
<td>Quasi-Gaussian Sampling</td>
<td></td>
</tr>
<tr>
<td>Increasing Population</td>
<td></td>
</tr>
</tbody>
</table>

![QR Code]


\[ 2^9 \times 3^2 = 4608 \]
Modular CMA-ES Framework

Selected CMA-ES Modules

<table>
<thead>
<tr>
<th>Module</th>
<th>Representation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Active Update</td>
<td>[0, 0, 0, 0, 0, 0, 0, 0, 0, 0]</td>
</tr>
<tr>
<td>Elitism</td>
<td>[0, 0, 0, 0, 0, 0, 0, 0, 0, 1]</td>
</tr>
<tr>
<td>Mirrored Sampling</td>
<td>[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 2]</td>
</tr>
<tr>
<td>Orthogonal Sampling</td>
<td>[1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 0]</td>
</tr>
<tr>
<td>Sequential Selection</td>
<td>[1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 1]</td>
</tr>
<tr>
<td>Threshold Convergence</td>
<td>[1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 2]</td>
</tr>
<tr>
<td>TPA</td>
<td></td>
</tr>
<tr>
<td>Pairwise Selection</td>
<td></td>
</tr>
<tr>
<td>Recombination Weights</td>
<td></td>
</tr>
<tr>
<td>Quasi-Gaussian Sampling</td>
<td></td>
</tr>
<tr>
<td>Increasing Population</td>
<td></td>
</tr>
</tbody>
</table>

\[2^9 \times 3^2 = 4608\]

Modular CMA-ES Framework

Algorithm 1 Modular CMA-ES Framework

1: options ← which modules are active
2: init-params ← initial/default parameter values
3: while not terminate do // Local restart loop
4: params ← Initialize(init-params)
5: t ← 0
6: \( \bar{x} \) ← randomly generated individual
7: while not terminate local do // ES execution loop
8: \( \bar{x} \) ← Mutate(\( \bar{x} \), options) // Sampler, Threshold
9: \( \bar{f} \) ← Evaluate(\( \bar{x} \), options) // Sequential
10: \( P^{(t+1)} \) ← Select(\( \bar{x} \), \( \bar{f} \), options) // Elitism, Pairwise
11: \( \bar{x} \) ← Recombine(\( P^{(t+1)} \), options) // Weights
12: UpdateParams(params, options) // Active, TPA
13: t ← t + 1
14: end while
15: AdaptParams(init-params, options) // (B)IPOP
16: end while

Problem: How to determine quality of a configuration $c$?

- FCE: arbitrary values
- ERT: only defined on $\text{FCE}(c) < \text{FCE}_{\text{target}}$
Algorithm Quality

Problem: How to determine quality of a configuration c?

- FCE: arbitrary values
- ERT: only defined on FCE(c) < FCE_{target}

Solution: scaled combination $ERT \times FCE \rightarrow [0, 2]$

$$q(c) = \begin{cases} \frac{ERT(c)}{ERT_{\text{max}}}, & \text{if } ERT(c) \text{ exists} \\ \frac{\log(FCE(c)/FCE_{\text{target}})}{1 + \frac{\log(FCE_{\text{max}}/FCE_{\text{target}})}} & \text{otherwise,} \end{cases}$$
Algorithm quality vs. Rank for F10

Quality

Rank

Example
Random Forest Regression

Forest predicts $q$ per experiment

Forest of 250 trees

Mean feature importance\(^1\) over all experiments

\(^1\)A measure of how pure the split according to a feature is
Random Forest Regression

Forest predicts $q$ per experiment

Forest of 250 trees

Mean feature importance\(^1\) over all experiments

<table>
<thead>
<tr>
<th>Module</th>
<th>Importance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Active</td>
<td>0.04</td>
</tr>
<tr>
<td>Elitism</td>
<td>0.06</td>
</tr>
<tr>
<td>Mirrored</td>
<td>0.04</td>
</tr>
<tr>
<td>Orthogonal</td>
<td>0.05</td>
</tr>
<tr>
<td><strong>Sequential</strong></td>
<td><strong>0.16</strong></td>
</tr>
<tr>
<td><strong>Threshold</strong></td>
<td><strong>0.31</strong></td>
</tr>
<tr>
<td>TPA</td>
<td>0.03</td>
</tr>
<tr>
<td>Pairwise</td>
<td>0.02</td>
</tr>
<tr>
<td>Weights</td>
<td>0.02</td>
</tr>
<tr>
<td><strong>Base-Sampler</strong></td>
<td><strong>0.20</strong></td>
</tr>
<tr>
<td>(B)IPOP</td>
<td>0.07</td>
</tr>
</tbody>
</table>

\(^1\)A measure of how *pure* the split according to a feature is
Impact

Compare $x_{on}$ and $x_{off}$:

$$I_x = \bar{q}(C_{off}^x) - \bar{q}(C_{on}^x)$$

$\bar{q}(C)$: mean $q$ of set $C$
Impact

Compare $x_{on}$ and $x_{off}$:

$$I_x = \bar{q}(C^x_{off}) - \bar{q}(C^x_{on})$$

$\bar{q}(C)$: mean $q$ of set $C$
Impact per Module

Discover the world at Leiden University
Impact: Module Interaction

Compare $x_{on} \land y_{on}$ and $\neg(x_{on} \land y_{on})$
Impact: Module Interaction

Compare $x_{on} \land y_{on}$ and $\neg(x_{on} \land y_{on})$
Impact: Module Interaction

Impact of module interaction for F2

Impact of module interaction for F7

Impact of module interaction for F23

Impact of module interaction for F24
Module Progression

Configuration ranking:

#1: [0 0 1 1 0 1 1 0 0 2 0]
#2: [0 0 1 1 0 1 1 0 0 2 2]
#3: [0 0 1 1 0 1 1 0 0 2 1]
#4: [0 0 1 1 0 1 1 1 0 2 2]

... 

#4608: ...

Which modules are active

• in the best configuration?
Module Progression

Configuration ranking:

#1: [0 0 1 1 0 1 1 0 0 2 0]
#2: [0 0 1 1 0 1 1 0 0 2 2]
#3: [0 0 1 1 0 1 1 0 0 2 1]
#4: [0 0 1 1 0 1 1 1 0 2 2]
...
#4608: ...

Which modules are active

• in the best configuration?
• in the 10 best configurations?
Module Progression

Configuration ranking:

#1: [0 0 1 1 0 1 1 0 0 2 0]
#2: [0 0 1 1 0 1 1 0 0 2 2]
#3: [0 0 1 1 0 1 1 0 0 2 1]
#4: [0 0 1 1 0 1 1 1 0 2 2]
...
#4608: ...

Which modules are active

• in the best configuration?
• in the 10 best configurations?
• in the 100 best configurations?
• ...

Discover the world at Leiden University
Progression Correlation

Progression for experiment similarity cluster (c > 0.978)

Progression for experiment similarity cluster (c > 0.993)

Progression for experiment similarity cluster (c > 0.997)
Progression Correlation

Progression for module cooperation cluster (c > 0.921)

Progression for module cooperation cluster (c > 0.962)

Impact of module interaction for 5D F14

Impact of module interaction for 5D F15
Summary & Outlook

Summary

• We can successfully identify useful options
• Similar landscapes show similar impact/progression behavior

Outlook

• Combine with landscape features
• Include parameters tuning
• Expand to include more modules

Code on Github
Summary & Outlook

Summary

• We can successfully identify useful options
• Similar landscapes show similar impact/progression behavior

Outlook

• Combine with landscape features
• Include parameters tuning
• Expand to include more modules
Summary & Outlook

Summary

• We can successfully identify useful options
• Similar landscapes show similar impact/progression behavior

Outlook

• Combine with landscape features
• Include parameters tuning
• Expand to include more modules

Code on Github

Appendix: Module Impact p-values

All \( l_x \) outside \([-0.144, 0.229]\) have \( p < 0.01 \) (442 / 2640 values)
Appendix: Interaction Impact p-values

All $I_x$ outside $[-0.551, 0.371]$ have $p < 0.01$ (1383 / 29040 values)