

# Adaptive Strategy Selection in Differential Evolution

Wenyin Gong<sup>1</sup>   Álvaro Fialho<sup>2</sup>   Zhihua Cai<sup>1</sup>

<sup>1</sup>School of Computer Science, China University of Geosciences  
Wuhan, China

<sup>2</sup>Microsoft Research – INRIA Joint Centre  
Orsay, France

July 10, 2010

# Differential Evolution

- Population-based EA
- Each individual is used to generate a new offspring
- Mutation: weighted differences between several individuals
- Crossover: mix parts of the mutated and the original individual
- Mutation always applied, Crossover applied with rate (1-CR)
- Replacement: 1x1, if offspring better than parent, replace it

## User-defined parameters

- Population size NP
- Mutation scaling factor F
- Crossover rate CR
- **Which mutation strategies to apply?**

# DE Mutation Strategies

- Around a dozen of well-known existent strategies
  - Which is the base vector (e.g., current, best-so-far, random)
  - How many other vectors are considered
- As in other EAs, complex and problem-dependent choice

# DE Mutation Strategies

- Around a dozen of well-known existent strategies
    - Which is the base vector (e.g., current, best-so-far, random)
    - How many other vectors are considered
  - As in other EAs, complex and problem-dependent choice
- 
- Off-line tuning could be used to find the best one
  - Based on some statistics over several runs for each strategy
  - Expensive, providing the static single best strategy

# DE Mutation Strategies

- Around a dozen of well-known existent strategies
    - Which is the base vector (e.g., current, best-so-far, random)
    - How many other vectors are considered
  - As in other EAs, complex and problem-dependent choice
- 
- Off-line tuning could be used to find the best one
  - Based on some statistics over several runs for each strategy
  - Expensive, providing the static single best strategy
- 
- Best strategy depends on the region of the search space
  - Should be continuously adapted, while solving the problem
  - $\implies$  Adaptive Strategy Selection

# Adaptive Operator/Strategy Selection

## Objective

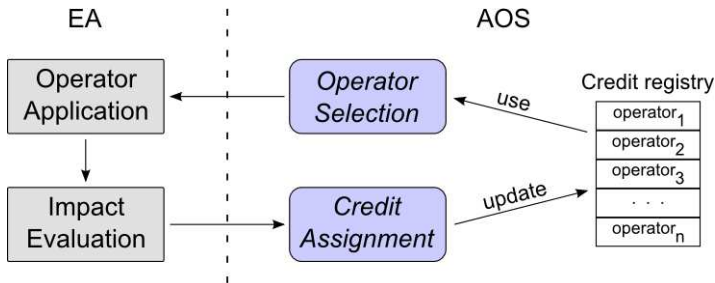
Autonomously select the operator to be applied between the available ones, based on its impact on the search up to now.



# Adaptive Operator/Strategy Selection

## Objective

Autonomously select the operator to be applied between the available ones, based on its impact on the search up to now.



- Credit Assignment: Relative fitness improvement
- Strategy Selection: Probability Matching

# Credit Assignment

- Very common: Fitness Improvement

## Which statistics to use?

- Instantaneous value likely to be unstable
- Average or Extreme value over a Window

## But...

- Ranges of rewards depend on the problem
- Very problem-dependent while raw fitness values are considered...
- Consequently, AdapSS becomes also very problem-dependent



# Credit Assignment

- Very common: Fitness Improvement

## Which statistics to use?

- Instantaneous value likely to be unstable
- Average or Extreme value over a Window

## But...

- Ranges of rewards depend on the problem
- Very problem-dependent while raw fitness values are considered...
- Consequently, AdapSS becomes also very problem-dependent
- $\implies$  Some kind of general normalization...

# Normalized Credit Assignment

## SaDE [Qin et al., 2009]

- Number of successful trials of this operator w.r.t. others
- Not problem-dependent, reward always between 0 and 1

# Normalized Credit Assignment

## SaDE [Qin et al., 2009]

- Number of successful trials of this operator w.r.t. others
- Not problem-dependent, reward always between 0 and 1
- But amplitude and order lost:  $\Delta F$  of 1000 or 0.001  $\equiv$  1

# Normalized Credit Assignment

## SaDE [Qin et al., 2009]

- Number of successful trials of this operator w.r.t. others
- Not problem-dependent, reward always between 0 and 1
- But amplitude and order lost:  $\Delta F$  of 1000 or 0.001  $\equiv$  1

## ⇒ Relative Fitness Improvement (RFI) [Ding and Keane, 2004]

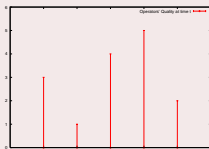
$$\text{RFI}_i = \frac{f(\text{best})}{f(\text{offspring}_i)} \cdot |f(\text{parent}_i) - f(\text{offspring}_i)| \quad (1)$$

- No improvement = null reward
- Credit assigned = Average of recent RFIs

# Strategy Selection: Probability Matching

## Quality Empirical Estimate $\hat{Q}$

- $\hat{Q}_{j,t}$  = estimate of reward of operator  $j$ , for time  $t$



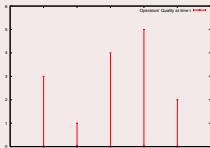
- At time  $t$ , op.  $j$  applied; gets reward  $r_{j,t}$ ;
- $\hat{Q}$  is updated by relaxation:

$$\hat{Q}_{j,t+1} = (1 - \alpha) \cdot \hat{Q}_{j,t} + \alpha \cdot r_{j,t}$$

# Strategy Selection: Probability Matching

## Quality Empirical Estimate $\hat{Q}$

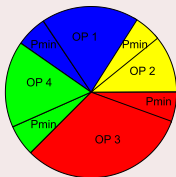
- $\hat{Q}_{j,t}$  = estimate of reward of operator  $j$ , for time  $t$



- At time  $t$ , op.  $j$  applied; gets reward  $r_{j,t}$ ;
- $\hat{Q}$  is updated by relaxation:

$$\hat{Q}_{j,t+1} = (1 - \alpha) \cdot \hat{Q}_{j,t} + \alpha \cdot r_{j,t}$$

## Selection Probability $s$



$s_i$  proportional to  $Q_i$

$$s_{i,t+1} = p_{min} + (1 - K * p_{min}) \frac{\hat{Q}_{i,t+1}}{\sum_{j=1}^K \hat{Q}_{j,t+1}}$$

# PM-AdapSS-DE: Prob. Matching + Relative $\Delta$ Fitness

- As meta-parameters, usually such an AdapSS would have:
  - Credit Assignment: the size of sliding window  $W$
  - Strategy Selection: the minimal selection probability  $p_{min}$
  - Strategy Selection: the quality adaptation rate  $\alpha$

# PM-AdapSS-DE: Prob. Matching + Relative $\Delta$ Fitness

- As meta-parameters, usually such an AdapSS would have:
  - Credit Assignment: the size of sliding window  $W$
  - Strategy Selection: the minimal selection probability  $p_{min}$
  - Strategy Selection: the quality adaptation rate  $\alpha$



# PM-AdapSS-DE: Prob. Matching + Relative $\Delta$ Fitness

- As meta-parameters, usually such an AdapSS would have:
  - Credit Assignment: the size of sliding window  $W$
  - Strategy Selection: the minimal selection probability  $p_{min}$
  - Strategy Selection: the quality adaptation rate  $\alpha$

## Simplifications

- Quality empirical estimate  $\hat{Q}$  and selection rate  $s$  updated just once every generation
  - Average of relative fitness improvements within generation

# PM-AdapSS-DE: Prob. Matching + Relative $\Delta$ Fitness

- As meta-parameters, usually such an AdapSS would have:
  - ~~Credit Assignment: the size of sliding window  $W$~~
  - Strategy Selection: the minimal selection probability  $p_{min}$
  - Strategy Selection: the quality adaptation rate  $\alpha$

## Simplifications

- Quality empirical estimate  $\hat{Q}$  and selection rate  $s$  updated just once every generation
  - Average of relative fitness improvements within generation

# PM-AdapSS-DE: Prob. Matching + Relative $\Delta$ Fitness

- As meta-parameters, usually such an AdapSS would have:
  - ~~Credit Assignment: the size of sliding window  $W$~~
  - Strategy Selection: the minimal selection probability  $p_{min}$
  - Strategy Selection: the quality adaptation rate  $\alpha$

## Simplifications

- Quality empirical estimate  $\hat{Q}$  and selection rate  $s$  updated just once every generation
  - Average of relative fitness improvements within generation
- No need of minimal selection probability
  - Operator must receive “infinite” null rewards to get “lost”

# PM-AdapSS-DE: Prob. Matching + Relative $\Delta$ Fitness

- As meta-parameters, usually such an AdapSS would have:
  - ~~Credit Assignment: the size of sliding window  $W$~~
  - ~~Strategy Selection: the minimal selection probability  $p_{min}$~~
  - Strategy Selection: the quality adaptation rate  $\alpha$

## Simplifications

- Quality empirical estimate  $\hat{Q}$  and selection rate  $s$  updated just once every generation
  - Average of relative fitness improvements within generation
- No need of minimal selection probability
  - Operator must receive “infinite” null rewards to get “lost”

# PM-AdapSS-DE: Prob. Matching + Relative $\Delta$ Fitness

- As meta-parameters, usually such an AdapSS would have:
  - ~~Credit Assignment: the size of sliding window  $W$~~
  - ~~Strategy Selection: the minimal selection probability  $p_{min}$~~
  - Strategy Selection: the quality adaptation rate  $\alpha$

## Simplifications

- Quality empirical estimate  $\hat{Q}$  and selection rate  $s$  updated just once every generation
  - Average of relative fitness improvements within generation
- No need of minimal selection probability
  - Operator must receive “infinite” null rewards to get “lost”
- Adaptation rate  $\alpha$  not sensitive
  - $\alpha \in \{.1, .3, .6, .9\}$  tried,  $\{.3, .6\}$  showed good performance

# PM-AdapSS-DE: Prob. Matching + Relative $\Delta$ Fitness

- As meta-parameters, usually such an AdapSS would have:
  - ~~Credit Assignment: the size of sliding window  $W$~~
  - ~~Strategy Selection: the minimal selection probability  $p_{min}$~~
  - ~~Strategy Selection: the quality adaptation rate  $\alpha \equiv 0.5$~~

## Simplifications

- Quality empirical estimate  $\hat{Q}$  and selection rate  $s$  updated just once every generation
  - Average of relative fitness improvements within generation
- No need of minimal selection probability
  - Operator must receive “infinite” null rewards to get “lost”
- Adaptation rate  $\alpha$  not sensitive
  - $\alpha \in \{.1, .3, .6, .9\}$  tried,  $\{.3, .6\}$  showed good performance

# Experimental Settings

## Differential Evolution

- Population size  $NP = 10 \times DIM$
- Mutation scaling factor  $F = 0.5$
- No crossover at all, search guided just by mutation
- No tuning of these parameters, focus on AdapSS

# Experimental Settings

## Differential Evolution

- Population size  $NP = 10 \times DIM$
- Mutation scaling factor  $F = 0.5$
- No crossover at all, search guided just by mutation
- No tuning of these parameters, focus on AdapSS

## Mutation Strategies

- 1 rand/1:  $\mathbf{v}_i = \mathbf{x}_{r_1} + F \cdot (\mathbf{x}_{r_2} - \mathbf{x}_{r_3})$
- 2 rand/2:  $\mathbf{v}_i = \mathbf{x}_{r_1} + F \cdot (\mathbf{x}_{r_2} - \mathbf{x}_{r_3}) + F \cdot (\mathbf{x}_{r_4} - \mathbf{x}_{r_5})$
- 3 rand-to-best/2:  
 $\mathbf{v}_i = \mathbf{x}_{r_1} + F \cdot (\mathbf{x}_{best} - \mathbf{x}_{r_1}) + F \cdot (\mathbf{x}_{r_2} - \mathbf{x}_{r_3}) + F \cdot (\mathbf{x}_{r_4} - \mathbf{x}_{r_5})$
- 4 current-to-rand/1:  $\mathbf{v}_i = \mathbf{x}_i + F \cdot (\mathbf{x}_{r_1} - \mathbf{x}_i) + F \cdot (\mathbf{x}_{r_2} - \mathbf{x}_{r_3})$



# Baseline Methods

- Each strategy alone, and naive uniform strategy selection

# Baseline Methods

- Each strategy alone, and naive uniform strategy selection

## Dynamic Multi-Armed Bandit (DMAB) [Da Costa et al., 2008]

- Original UCB MAB algorithm
- Dynamics: Page-Hinkley change-detection test

# Baseline Methods

- Each strategy alone, and naive uniform strategy selection

## Dynamic Multi-Armed Bandit (DMAB) [Da Costa et al., 2008]

- Original UCB MAB algorithm
- Dynamics: Page-Hinkley change-detection test

## Adaptive Pursuit (AP) [Thierens, 2008]

- Winner-take-all strategy to update the operators rates
- Best operator with rate  $p_{max}$ , others  $p_{min}$

# Baseline Methods

- Each strategy alone, and naive uniform strategy selection

## Dynamic Multi-Armed Bandit (DMAB) [Da Costa et al., 2008]

- Original UCB MAB algorithm
- Dynamics: Page-Hinkley change-detection test

## Adaptive Pursuit (AP) [Thierens, 2008]

- Winner-take-all strategy to update the operators rates
- Best operator with rate  $p_{max}$ , others  $p_{min}$
- Both rewarded by Extreme raw  $\Delta F$  values (window size  $W$ )
- DMAB: scaling factor  $C$ , change detection threshold  $\gamma$
- AP: adaptation rate  $\alpha$ , learning rate  $\beta$ , min. prob.  $p_{min}$

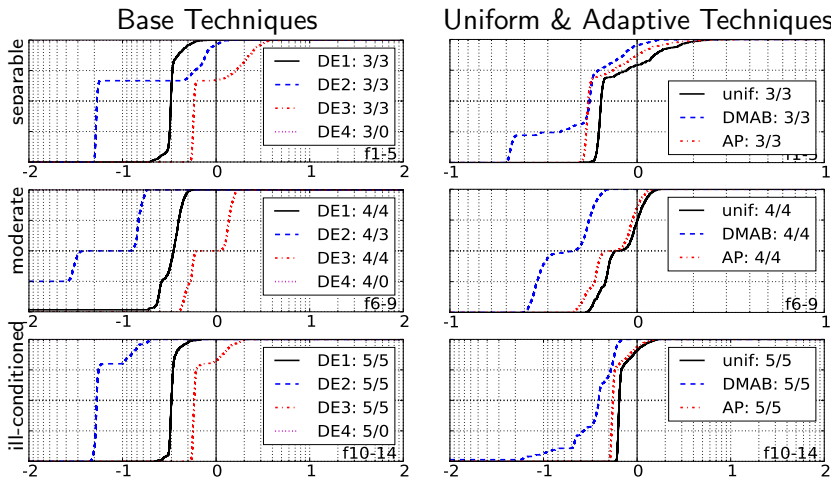
# Meta-Parameters Tuning

- Tuned off-line by F-RACE [\[Birattari et al., 2002\]](#)
  - 1 racing lap = 1 run over all functions
  - Elimination after each lap, based on Friedman test at 95%
  - $\implies$  Comparisons of AdapSS algorithms using best configurations
- 
- PM-AdapSS-DE, 16 configs: 4  $p_{min}$ , 4  $\alpha$
  - Ext-AP, 192 configs: 4  $p_{min}$ , 4  $\alpha$ , 4  $\beta$ , 3  $W$
  - Ext-DMAB, 714 configs: 14  $C$ , 17  $\gamma$ , 3  $W$

# Black-Box Optimization Benchmark (BBOB)

- Standardize the benchmarking of continuous optimizers
  - Workshops at GECCO'09 and GECCO'10 [[Hansen et al., 2010](#)]
- 24 single-objective **continuous functions** divided in 5 classes
  - Separable, Moderate, Ill-conditioned, Multi-modal and Weak-structure
- Several problem dimensions (2, 3, 5, 10, 20, 40)
- Post-processing scripts – standard tables and plots for comparisons

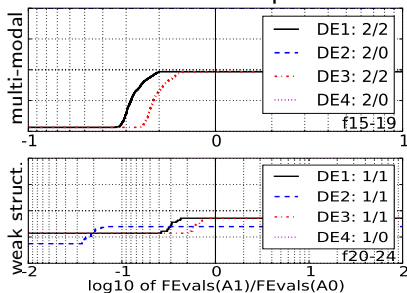
# Parwise comparisons of PM-AdapSS-DE with ...



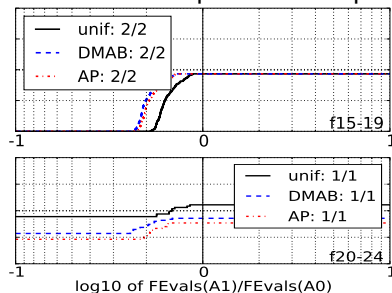
X-axis:  $\log\left(\frac{\text{effort}(\text{PM-AdapSS-DE})}{\text{effort}(\text{Baseline})}\right)$  – Y-axis: Proportion of runs

# Parwise comparisons of PM-AdapSS-DE with ...

## Base Techniques



## Uniform & Adaptive Techniques

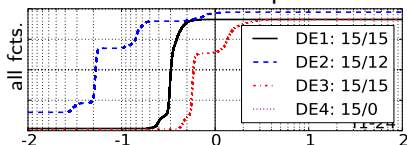


X-axis:  $\log\left(\frac{\text{effort}(\text{PM-AdapSS-DE})}{\text{effort}(\text{Baseline})}\right)$  – Y-axis: Proportion of runs

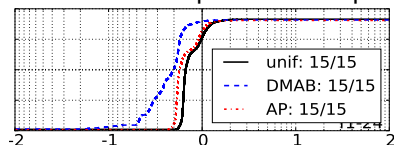


# Parwise comparisons of PM-AdapSS-DE with ...

## Base Techniques



## Uniform & Adaptive Techniques



X-axis:  $\log\left(\frac{\text{effort}(\text{PM-AdapSS-DE})}{\text{effort}(\text{Baseline})}\right)$  – Y-axis: Proportion of runs

# Conclusion and Perspectives

- Very simple and efficient AdapSS technique
- Abstracts complex decisions from the user's burden
  - Which strategies should be applied, and at which rate
- Mix of strategies is always better than the use of a single one
  - Continuously adapts according to the search progress
- Although much simpler, more robust to different situations
  - Consistently better than more complex AdapSS techniques

# Conclusion and Perspectives

- Very simple and efficient AdapSS technique
  - Abstracts complex decisions from the user's burden
    - Which strategies should be applied, and at which rate
  - Mix of strategies is always better than the use of a single one
    - Continuously adapts according to the search progress
  - Although much simpler, more robust to different situations
    - Consistently better than more complex AdapSS techniques
- 
- Further assessment
    - Method is generic, apply with other meta-heuristics
    - Assess on SAT, real-world problems, ...
  - Real-world problems are often multi-modal
    - Diversity should also be considered for the reward
    - As seen by the low performance on the multi-modal problems

# References I



Birattari, M., Stützle, T., Paquete, L., and Varrentrapp, K. (2002).

A racing algorithm for configuring metaheuristics.

In Proc. GECCO'02.



Da Costa, L., Fialho, A., Schoenauer, M., and Sebag, M. (2008).

Adaptive operator selection with dynamic multi-armed bandits.

In M. Keijzer et al., editor, Proc. GECCO'08, pages 913–920. ACM Press.



Hansen, N., Auger, A., Finck, S., and Ros, R. (2010).

Real-parameter black-box optimization benchmarking 2010: Experimental setup.

Technical Report RR-7215, INRIA.



Ong, Y.-S. and Keane, A. J. (2004).

Meta-Lamarckian learning in memetic algorithms.

IEEE Trans. on Evol. Comput., 8(2):99–110.



Qin, A. K., Huang, V. L., and Suganthan, P. N. (2009).

Differential evolution algorithm with strategy adaptation for global numerical optimization.

IEEE Trans. on Evol. Comput., 13(2):398–417.



Thierens, D. (2005).

An adaptive pursuit strategy for allocating operator probabilities.

In Beyer, H.-G., editor, Proc. GECCO'05, pages 1539–1546. ACM Press.

# Adaptive Strategy Selection in Differential Evolution

Wenyin Gong<sup>1</sup>   Álvaro Fialho<sup>2</sup>   Zhihua Cai<sup>1</sup>

<sup>1</sup>School of Computer Science, China University of Geosciences  
Wuhan, China

<sup>2</sup>Microsoft Research – INRIA Joint Centre  
Orsay, France

July 10, 2010