TPAM: A Simulation-Based Model for Quantitatively Analyzing Parameter Adaptation Methods

Ryoji Tanabe (SUSTech, CHN) and Alex Fukunaga (U. Tokyo, JP)

The first quantitative analysis of PAMs in isolation

- Parameter Adaptation Methods (PAMs) are poorly understood
- We propose a Target function-based PAM simulation (TPAM) framework for analyzing PAMs in adaptive DE
- TPAM measures the ability of PAMs to track predefined target parameters and enables quantitative analysis of PAMs
  - E.g., PAM-JADE tracks this particular target behavior 1.4 times better than PAM-SHADE

Differential Evolution (DE) [Storn 97]

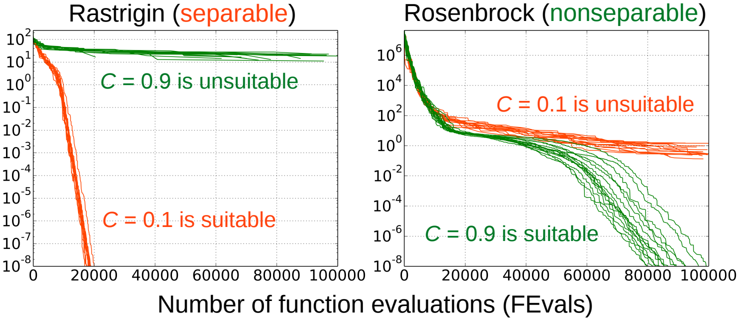
The two main control parameters of DE are:

1. Scale factor $F \in (0, 1]$:
   - F controls the magnitude of the differential mutation
2. Crossover rate $C \in [0, 1]$:
   - C controls the number of inherited variables from $\boldsymbol{x}$

```
t ← 1, initialize the population P^t = {x^{1,t}, ..., x^{N,t}};
while The termination criteria are not met do
    for i ∈ {1, ..., N} do
        v^{i,t} ← differentialMutation(P^t, F_{i,t});
        u^{i,t} ← crossover(x^{i,t}, v^{i,t}, C_{i,t});
    for i ∈ {1, ..., N} do
        if f(u^{i,t}) ≤ f(x^{i,t}) then x^{i,t+1} ← u^{i,t} ;
        else x^{i,t+1} ← x^{i,t} ;
    t ← t + 1;
```

The performance of DE depends on the setting of F and C

E.g., An appropriate setting of C depends on separability



The classical DE is not so efficient for black-box optimization

- DE needs automated parameter control methods for F and C

A large number of adaptive DE algorithms have been proposed

- jDE [Brest 06], JADE [Zhang 09], SaDE [Qin 09], EPSDE [Mallipeddi 11], MDE [Islam 12], SHADE [Tanabe 13], ...

Parameter Adaptation Methods (PAMs) are poorly understood

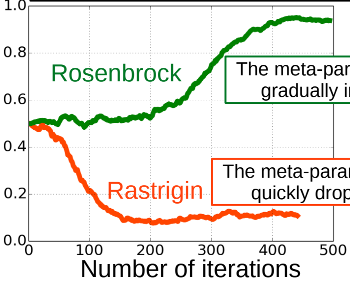
We are interested in PAMs in adaptive DE, not adaptive DE

	PAMs (our interest)	Mutation strategy	Cross.
jDE =	PAM of jDE	rand/1	binomial
JADE =	PAM of JADE	current-to-*pbest*/1	binomial

Several previous works have tried to analyze PAMs in adaptive DE

- Their results/analysis limited to plots of changes in F and C

E.g., Adaptation process of the meta-parameter of JADE for *C*



The meta-parameter of *C* gradually increases

The meta-parameter of *C* quickly drops down

Analysis limited to *qualitative* descriptions

Such qualitative results do not provide sufficient insight

Proposal: Target function-based PAM simulation (TPAM)

Traditional analyses based on plotting parameter values:

- Limited to *qualitative* descriptions
- May be useful for analyzing overall behavior of adaptive DEs, but cannot analyze the behavior of PAMs *in isolation*

Our proposed TPAM:

- *Quantitatively* evaluates the tracking performance of PAMs
- Enables analysis for PAMs in isolation
  - Independent of other mechanisms in adaptive DE
- Measures the ability of PAMs to track predefined target parameters and enables quantitative analysis of PAMs
  - E.g., PAM-JADE tracks this particular target behavior 1.4 times better than PAM-SHADE

A generalized Parameter Adaptation Method (PAM) in adaptive DE

1. At the beginning of each iteration t, generate $F_{i,t}$ and $C_{i,t}$ for each individual $\boldsymbol{x}^{i,t}$ using meta-parameters
2. Decide whether $\{F_{i,t}, C_{i,t}\}$ is a success or a failure
    - Success: if the child $\boldsymbol{u}^{i,t}$ is better than the parent $\boldsymbol{x}^{i,t}$
        - I.e., if $f(\boldsymbol{u}^{i,t}) \leq f(\boldsymbol{x}^{i,t})$
    - Failure: Otherwise
3. At the end of each iteration t, update the meta-parameters based on the success/failure decisions

Strictly speaking:

- Some PAMs do not use meta-parameters
    - E.g., PAM-jDE, PAM-EPSDE,...
- But, such PAMs can also be generalized into the above framework

Example: PAM-JADE [Zhang 09]

- PAM-JADE uses two meta-parameters μ_F and μ_C for parameter adaptation of F and C, respectively

$t \leftarrow 1$, initialize the population $\boldsymbol{P}^t$, $\mu_F, \mu_C \leftarrow 0.5$;
while *The termination criteria are not met* **do**
  $\boldsymbol{S}^F \leftarrow \emptyset$, $\boldsymbol{S}^C \leftarrow \emptyset$;
  for $i \in \{1, ..., N\}$ **do**
    $F_{i,t} \leftarrow$ CauchyRand$(\mu_F, 0.1)$, $C_{i,t} \leftarrow$ NomalRand$(\mu_C, 0.1)$;
    $\boldsymbol{v}^{i,t} \leftarrow$ differentialMutation$(\boldsymbol{P}^t, F_{i,t})$;
    $\boldsymbol{u}^{i,t} \leftarrow$ crossover$(\boldsymbol{x}^{i,t}, \boldsymbol{v}^{i,t}, C_{i,t})$;
  for $i \in \{1, ..., N\}$ **do**
    if $f(\boldsymbol{u}^{i,t}) \leq f(\boldsymbol{x}^{i,t})$ **then**
      $\boldsymbol{x}^{i,t+1} \leftarrow \boldsymbol{u}^{i,t}$, $\boldsymbol{S}^F \leftarrow F_{i,t}$, $\boldsymbol{S}^C \leftarrow C_{i,t}$
    else $\boldsymbol{x}^{i,t+1} \leftarrow \boldsymbol{x}^{i,t}$;
  $\mu_F \leftarrow (1-c)\mu_F + c\,\mathrm{mean}_{\mathrm{A}}(\boldsymbol{S}^F)$, $\mu_C \leftarrow (1-c)\mu_C + c\,\mathrm{mean}_{\mathrm{L}}(\boldsymbol{S}^C)$;
  $t \leftarrow t+1$;

Example: PAM-SHADE [Tanabe 13]

- PAM-SHADE uses $\boldsymbol{M}^F = (M_1^F, ..., M_H^F)^{\mathrm{T}}$ and $\boldsymbol{M}^C = (M_1^C, ..., M_H^C)^{\mathrm{T}}$

$t \leftarrow 1$, initialize the population $\boldsymbol{P}^t$, $\boldsymbol{M}^F, \boldsymbol{M}^C \leftarrow \mathbf{0.5}$, $k \leftarrow 1$;

while *The termination criteria are not met* **do**

  $\boldsymbol{S}^F \leftarrow \emptyset$, $\boldsymbol{S}^C \leftarrow \emptyset$;

  for $i \in \{1, ..., N\}$ **do**

    $r_{i,t} \leftarrow$ Randi$\{1, ..., N\}$, $F_{i,t} \leftarrow$ CauchyRand$(M^F_{r_{i,t}}, 0.1)$, $C_{i,t} \leftarrow$ NomalRand$(M^C_{r_{i,t}}, 0.1)$;

    $\boldsymbol{v}^{i,t} \leftarrow$ differentialMutation$(\boldsymbol{P}^t, F_{i,t})$;

    $\boldsymbol{u}^{i,t} \leftarrow$ crossover$(\boldsymbol{x}^{i,t}, \boldsymbol{v}^{i,t}, C_{i,t})$;

  for $i \in \{1, ..., N\}$ **do**

    if $f(\boldsymbol{u}^{i,t}) \leq f(\boldsymbol{x}^{i,t})$ **then**

      $\boldsymbol{x}^{i,t+1} \leftarrow \boldsymbol{u}^{i,t}$, $\boldsymbol{S}^F \leftarrow F_{i,t}$, $\boldsymbol{S}^C \leftarrow C_{i,t}$

    else $\boldsymbol{x}^{i,t+1} \leftarrow \boldsymbol{x}^{i,t}$;

  $M_k^F \leftarrow \mathrm{mean_A}(\boldsymbol{S}^F)$, $M_k^C \leftarrow \mathrm{mean_L}(\boldsymbol{S}^C)$, $k \leftarrow \mathrm{mod}(k+1, M)$;

  $t \leftarrow t + 1$;

Basic idea of our proposed TPAM: Space reduction

A traditional approach must consider three complex spaces

Control parameter space ⇄ Solution space ⇄ Objective function space

- *ALL* previous work used the objective function value of the solution $f(\boldsymbol{x})$ to evaluate the generated control parameters
- Since $\boldsymbol{x}$ is also affected by variation operators, such approaches cannot evaluate the control parameters *in isolation*

PROPOSAL:

Let's remove the solution space from optimization problems!

The TPAM approach considers only the two simplified spaces

Control parameter space ⇄ TPAM success/failure space

- The solution space is eliminated, and the objective function space is replaced by the TPAM success/failure space
- Control parameters can be directly evaluated *in isolation*

Some notes on this presentation

Recall: We are NOT interested in "the whole adaptive DE"

- We want to focus only on Parameter Adaptation Methods
    - We are interested in PAM-jDE, not jDE

TPAM is NOT a class of benchmark function for optimization

- The TPAM is a simulation framework to analyze PAMs

TPAM does NOT seek to optimize a static objective function f

- TPAM measures the target-tracking behavior of a PAM
- Individuals in TPAM only have F, C values but no base level genome, so variation operators (e.g., mutation) are irrelevant

TPAM can deal with both F and C simultaneously

- We investigated (i) C, (ii) F, and (iii) $\{F, C\}$
    - But, the tendency of their results is not different
- For simplicity, we focus on C in this talk

The process of PAMs in adaptive DE only depend on whether each child generation is a success or a failure

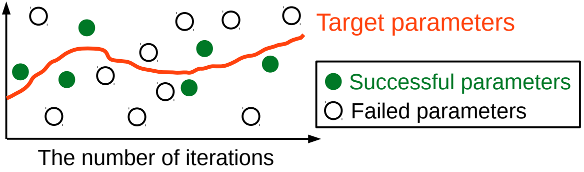
- Analyzing the PAM behavior does not require modeling the absolute objective function values of the individuals
- A simulation model of the success/failure decisions is sufficient
- Parameter adaptation of PAMs can be simulated by using a surrogate model deciding whether $\{F_{i,t}, C_{i,t}\}$ is a success or not

$t \leftarrow 1$;
while *The termination criteria are not met* **do**
  $\boldsymbol{S}^F \leftarrow \emptyset$, $\boldsymbol{S}^C \leftarrow \emptyset$;
  for $i \in \{1, ..., N\}$ **do**
    Generate $F_{i,t}$ and $C_{i,t}$ according to meta-parameters;
  for $i \in \{1, ..., N\}$ **do**
    if *The pair of $F_{i,t}$ and $C_{i,t}$ is successful* **then**
      $\boldsymbol{S}^F \leftarrow F_{i,t}$, $\boldsymbol{S}^C \leftarrow C_{i,t}$
  Update the meta-parameters based on $\boldsymbol{S}^F$ and $\boldsymbol{S}^C$;
  $t \leftarrow t + 1$;

A model of the success/failure decisions in the proposed TPAM

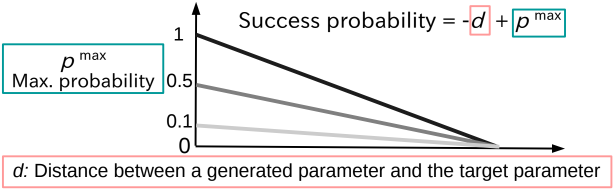
Target parameters $\theta_1^{\text{target}}, \theta_2^{\text{target}}, \ldots$ are introduced in TPAM

- The dicision is made based on the distance from θ^{target}
- The closer θ is from θ^{target}, the higher its success probability



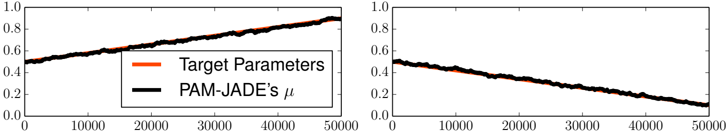
Success probability depends on two parameters: $p^{\max}$ and $d \in [0, 1]$

- A smaller $p^{\max}$ value makes a simulation model difficult

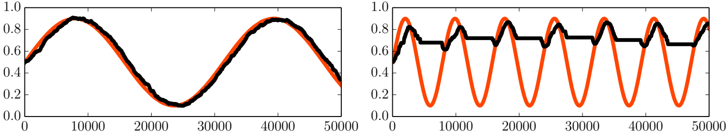


Target parameters $\theta_1^{\mathrm{target}}, \theta_2^{\mathrm{target}}, \ldots$ are given by a target function g

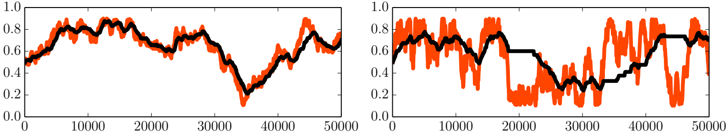
1. Two types of the linear functions $g^{\mathrm{lin/inc}}$ and $g^{\mathrm{lin/dec}}$



2. The sinusoidal function $g^{\sin}$ (left: $\omega = 10$, right: $\omega = 40$)



3. The random-walk function g^{ran} (left: $s = 0.04$, right: $s = 0.1$)



The overall TPAM framework

$t \leftarrow 1$, initialize a meta-parameter;
while *The termination criteria are not met* **do**
    $\theta_t^{\mathrm{target}} \leftarrow g(t)$;
    $\boldsymbol{S}^{\theta} \leftarrow \emptyset$;
    for $i \in \{1, ..., N\}$ **do**
        Generate $\theta_{i,t}$ according to the meta-parameter;
    for $i \in \{1, ..., N\}$ **do**
        if $\mathrm{isParametersSuccessful}(\theta_{i,t}, \theta_t^{\mathrm{target}})$ = *successful* **then**
            $\boldsymbol{S}^{\theta} \leftarrow \theta_{i,t}$;
    Update the meta-parameter based on $\boldsymbol{S}^{\theta}$;
    $t \leftarrow t + 1$;

- A parameter θ is (i) C, (ii) F, or (iii) $\{F, C\}$

Experimental settings

Settings for Parameter Adaptation Methods (PAMs)

- PAM-jDE, PAM-EPSDE, PAM-JADE, PAM-MDE, PAM-SHADE
  - For each PAM, the hyperparameters recommended by the original authors were used
- Population size = 50

Settings for the proposed TPAM

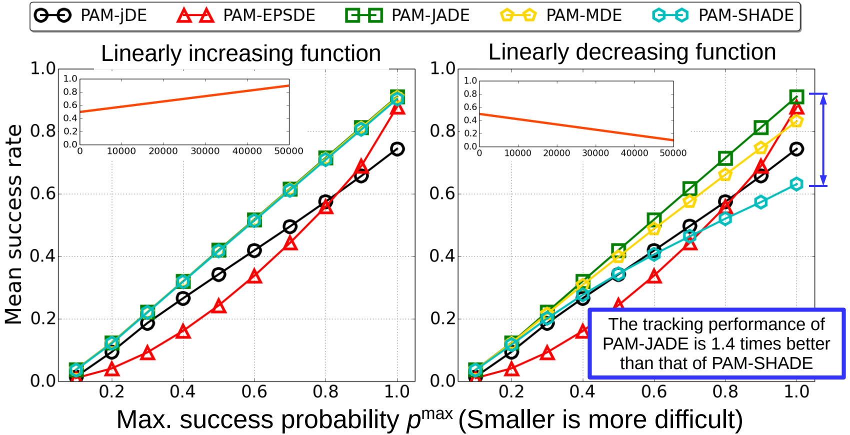
- Number of parameter sampling steps $= 50,000$
  - To evaluate PAMs over a large window of activity
  - This does not correspond to any specific number of search steps executed by a DE
- 101 independent runs. $p^{\max} \in \{0.1, 0.2, ..., 1.0\}$

Evaluation metric for TPAM: The percentage of successful parameters $r^{\mathrm{succ}} \in [0, 1]$

- A higher r^{succ} represents that the PAM is able to track given target parameters $\theta_1^{\mathrm{target}}, \theta_2^{\mathrm{target}}, ...$

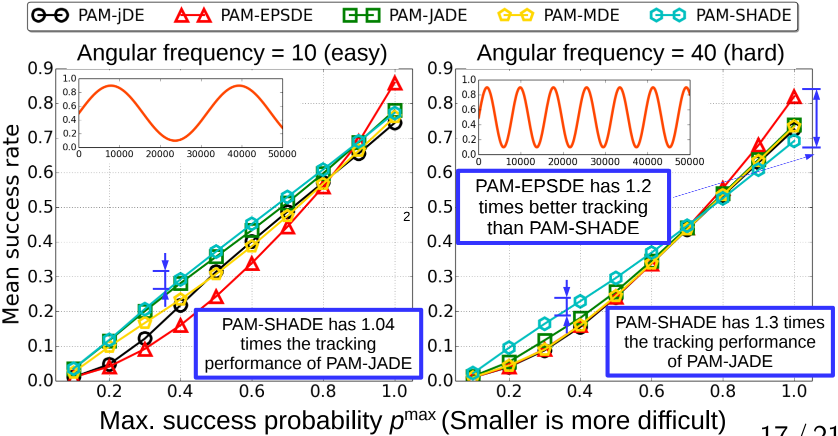
The tracking performance of the five PAMs on the linear functions

- PAM-JADE has best tracking for all the $p^{\max}$ values
- For the smaller $p^{\max}$ values, PAM-EPSDE cannot track θ^{target}
- PAM-MDE and PAM-SHADE have a hidden bias



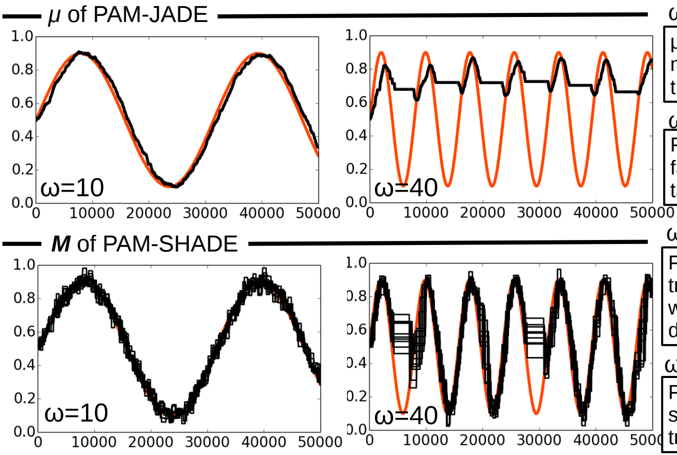
The tracking performance of the five PAMs on the sinusoidal function

- PAM-EPSDE has best tracking for the larger $p^{\max}$ values
- PAM-SHADE has best tracking for the smaller $p^{\max}$ values
- The tracking performance of PAM-SHADE on hard TPAM instances is better than that of the other PAMs



Why does PAM-SHADE track the target better than PAM-JADE?

The diversity of values in $\boldsymbol{M}$ enables PAM-SHADE to be much more robust than PAM-JADE on hard TPAM instances



ω=10

μ of PAM-JADE mostly overlaps the target

ω=40

PAM-JADE clearly fails to track the target parameters

ω=10

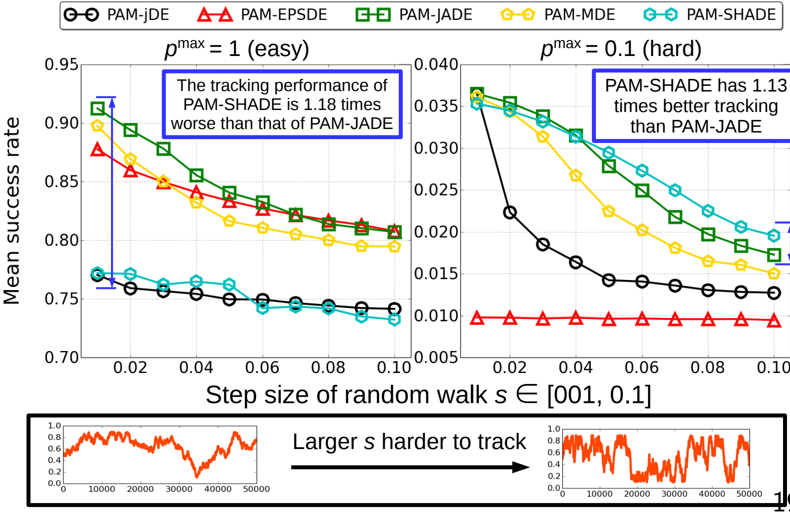
PAM-SHADE tracks the target with broadly distributed values

ω=40

PAM-SHADE succeeds in tracking the target

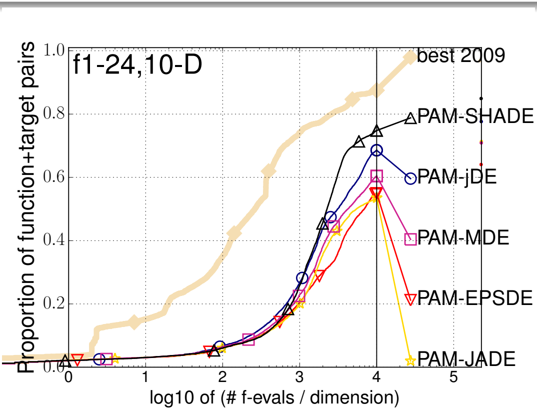
The tracking performance of the PAMs on the random-walk function

- Easy instances: PAM-JADE and PAM-EPSDE track well
- Hard instances: PAM-SHADE has the best tracking performance



How relevant are the target tracking accuracy of PAMs to the search performance of adaptive DEs? Results on the BBOB functions [Hansen 09]

- The rand/1/bin operators are used for all PAMs
- An adaptive DE using PAM-SHADE performs well
- The tracking performance of a PAM in the TPAM model is correlated with search performance of DE using that PAM



Conclusion:

The first quantitative investigation of the adaptation ability of PAMs

We proposed a Target function-based PAM simulation (TPAM)

- TPAM evaluates the tracking performance of PAMs and enables quantitative comparison of PAMs
  - E.g., PAM-SHADE tracks this particular target behavior 1.3 times better than PAM-JADE
- TPAM can provide important insights on PAMs,
  - E.g., Why an adaptive DE using PAM-SHADE performs well

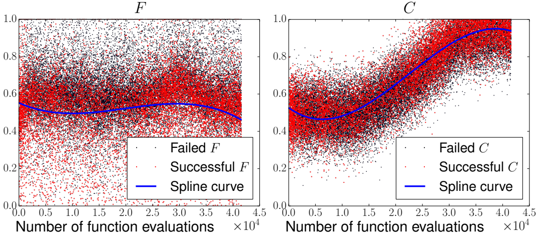
TPAM is a promising simulation framework for analyzing PAMs

Future work

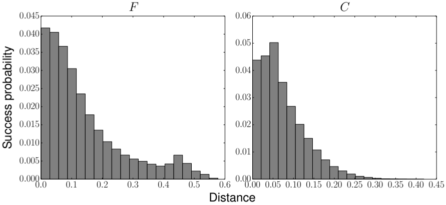
- Apply TPAM to analysis of PAMs in other EAs (e.g., ESs)
- Design an unified, systematic simulation framework
  - TPAM evaluates only the tracking performance of PAMs
  - Other simulation models that evaluate other important aspects of PAMs (e.g., diversity) are necessary

Is our definition of success probability appropriate?

All of the parameter values generated by an adaptive DE using the current-to-$p$best/1/bin and PAM-JADE on the Rosenbrock



Success probability as a function of the distance of the F (left) and C (right) from the spline curve



The assumption that the probability of generating successful trial vectors is highly correlated with the ability to generate control parameters values which accurately track a target parameter is justifiable