Modified L-SHADE for Single Objective Real-Parameter Optimization

Contributed by

Jia-Fong Yeh, Ting-Yu Chen, and Tsung-Che Chiang

Department of Computer Science and Information Engineering,
National Taiwan Normal University, Taiwan

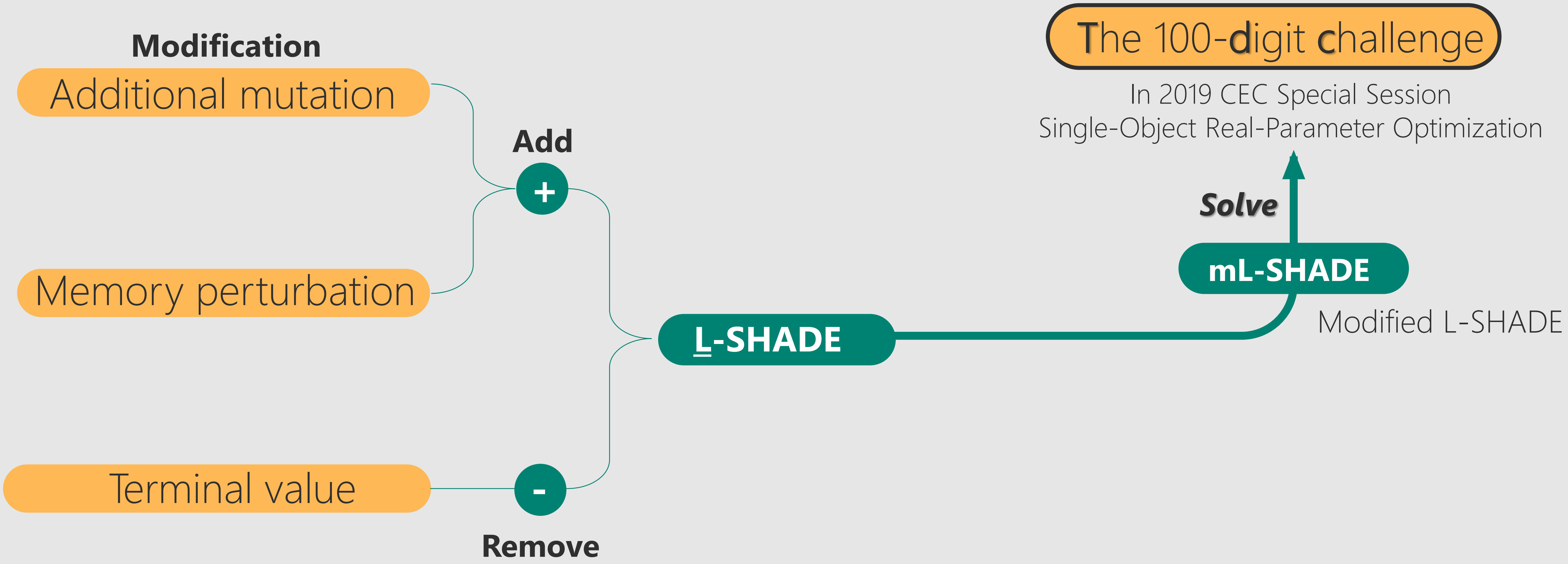
Personal contact

60647005S@ntnu.edu.tw, mazer0701@gmail.com, tcchiang@ieee.org

Contents

CEC 2019 Competitions

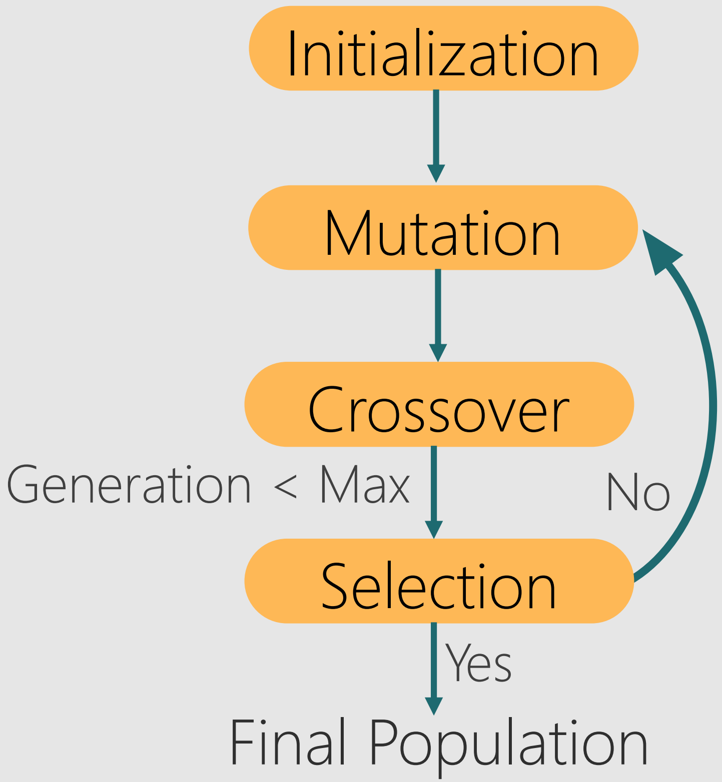
- CEC-C02 Competition on "Evolutionary Multi-task Optimization"
- CEC-C03 Competition on "Online Data-Driven Multi-Objective Optimization Competition"
- CEC-C04 Competition on "Smart Grid and Sustainable Energy Systems"
- CEC-C05 Competition on "Evolutionary Computation in Uncertain Environments: A Smart Grid Application"
- **CEC-C06 Competition on "100-Digit Challenge on Single Objective Numerical Optimization"**
- CEC-C07 FML-based Machine Learning Competition for Human and Smart Machine Co-Learning on Game of Go
- CEC-C08 General Video Game AI Single-Player Learning Competition
- CEC-C09 Strategy Card Game AI Competition
- CEC-C10 Nonlinear Equation Systems Competition



[L-SHADE] R. Tanabe, and A. S. Fukunaga, "Improving the Search Performance of SHADE Using Linear Population Size Reduction," in IEEE CEC, pp. 1658–1665, 2014.

- Algorithms overview

L-SHADE / mL-SHADE



(Uniform random initialization)

(Current-to-pbest/1 strategy + Gene repair)

(Binomial crossover)

(Fitness comparison)

L-SHADE

- Adaptive *F* and *CR* Parameter
- Population Size Reduction

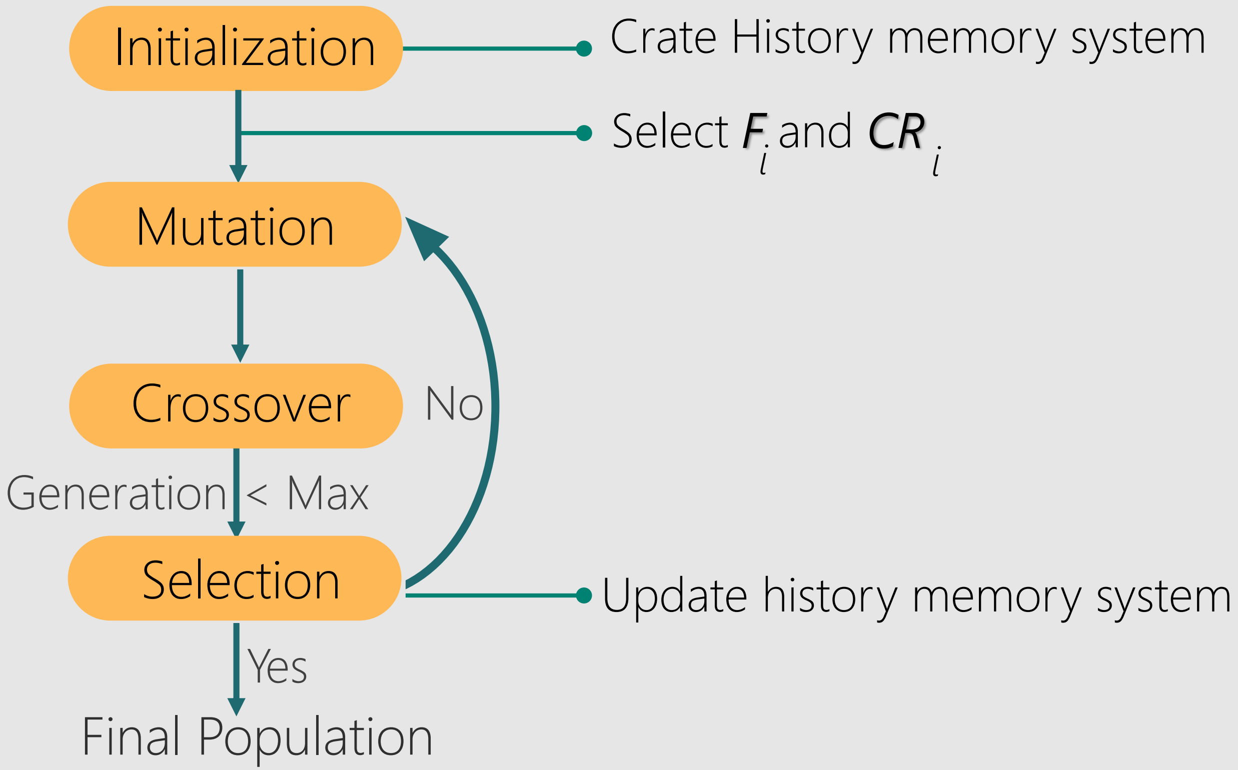
Adaptive Parameter

History memory system

M_F	$M_{F,1}$	$M_{F,2}$	……	$M_{F,H-1}$	$M_{F,H}$
M_{CR}	$M_{CR,1}$	$M_{CR,2}$	……	$M_{CR,H-1}$	$M_{CR,H}$

- *M* is mean value of the successful *F* and *CR* which can generate the better solution in each iteration
- Each mean value will be utilized to generate *F* and *CR* next iteration

L-SHADE



Adaptive Parameter

H

M_F	0.5	0.5	……	0.5	0.5
M_{CR}	0.5	0.5	……	0.5	0.5

In the initialization stage, all element of history table will be set to be 0.5, and each of them will be updated when the set of better solution is found

Initialization
Mutation
Crossover
Selection

Adaptive Parameter

H

M_F	$M_{F,1}$	$M_{F,2}$	……	$M_{F,H-1}$	$M_{F,H}$
M_{CR}	$M_{CR,1}$	$M_{CR,2}$	……	$M_{CR,H-1}$	$M_{CR,H}$

For each target vector x_i will generated F_i and CR_i as follow.

r_i index is selected randomly from [1,H]

$$F_i = \text{randc}_i(M_{F,r_i}, 0.1)$$

$$CR_i = \begin{cases} \text{randn}_i(M_{CR,r_i}, 0.1) & if\ M_{CR,r_i} \neq 0(\perp) \\ 0 & otherwise \end{cases}$$

$randc_i(\,)$ is a Cauchy distribution

$randn_i(\,)$ is a normal distribution

** 0 (⊥ terminal value)

Initialization
Mutation
Crossover
Selection

Adaptive Parameter

At the selection, if the trial vector's ($\vec{u}$) fitness is better than or equal to target vector's ($\vec{x}$), their fitness value, F_i, and CR_i will be stored in S table.

1. $$\Delta f_k = \left| f(u_{i,G}) - f(x_{i,G}) \right|$$

Fitness improvement

2. $$w_k = \frac{\Delta f_k}{\sum_{l=1}^{|S|} \Delta f_l}$$

Improvement-based weight

3. weighted Lehmer mean

F

$$mean_{wL}(F) = \frac{\sum_{k=1}^{|S|} w_k \cdot F_k^2}{\sum_{k=1}^{|S|} w_k \cdot F_k}$$

CR

$$mean_{wL}(\mathrm{CR}) = \frac{\sum_{k=1}^{|S|} w_k \cdot CR_k^2}{\sum_{k=1}^{|S|} w_k \cdot CR_k}$$

- Initialization
- Mutation
- Crossover
- **Selection**

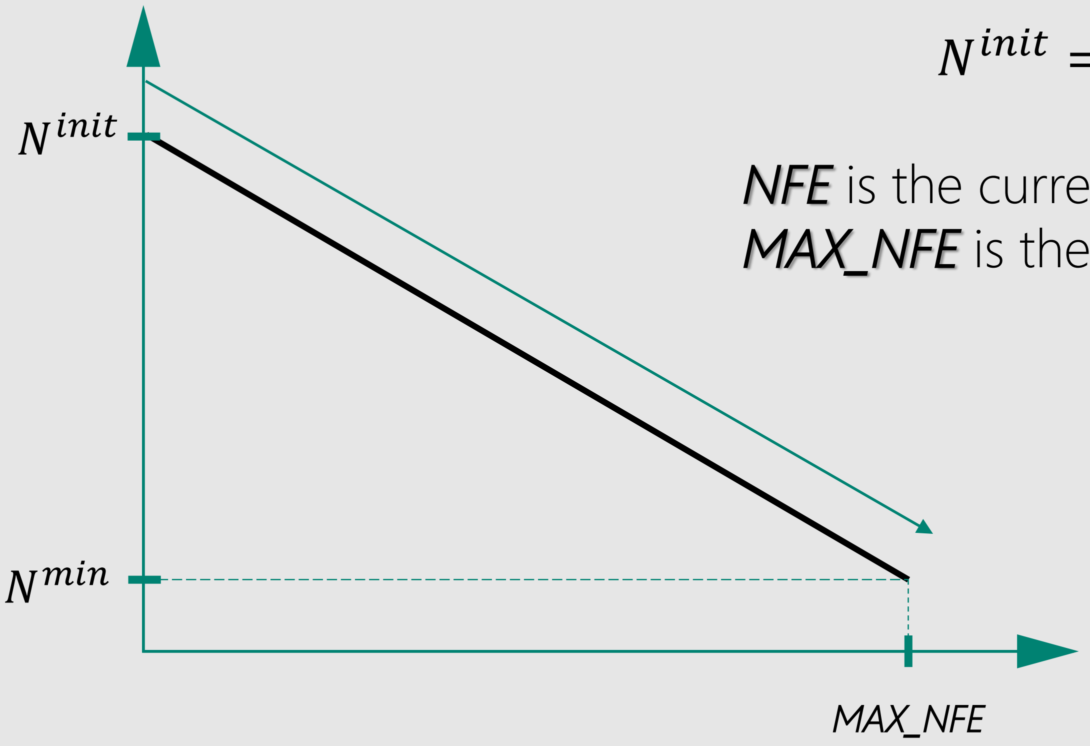
Adaptive Parameter

H

M_F	$M_{F,1}$	$M_{F,2}$	……	$M_{F,H-1}$	$M_{F,H}$
M_{CR}	$M_{CR,1}$	$\perp$	……	$M_{CR,H-1}$	$M_{CR,H}$

1. If *S* table is not empty, mean value of $M_{F,k}$ and $M_{CR,k}$ will be updated by new mean *F* and *CR*
2. If mean value of *CR* is 0, then $M_{CR,k}$ will be set as the terminal value ⊥ (0) and that element will never be changed to be the other number again.

- Linear population size reduction



$$NP_{G+1} = round\left(\left(\frac{N^{min} - N^{init}}{MAX\_NFE}\right) \times NFE + N^{init}\right)$$

$$N^{init} = 18 \times D,\ N^{min} = 4$$

NFE is the current number of fitness evaluations
MAX_NFE is the maximum number of fitness evaluations

mL-SHADE

L-SHADE

↓

mL-SHADE

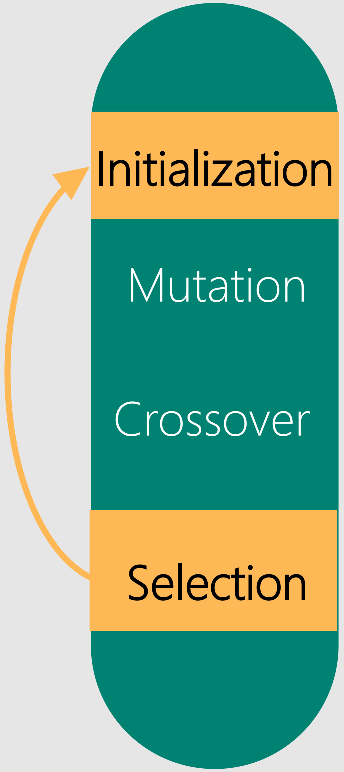
- − Terminal value
- \+ Memory perturbation
- \+ Additional mutation operator

Remove Terminal value

As L-SHADE will update the $M_{CR,k}$ element inside history memory table to be ⊥ every time, when it found the mean of *CR* equal 0 and never change to be the other value again. It also forces the target vector to *CR* as 0, it select the ⊥ from history table.

This can end the exploration and start to exploitation.

Binomial crossover



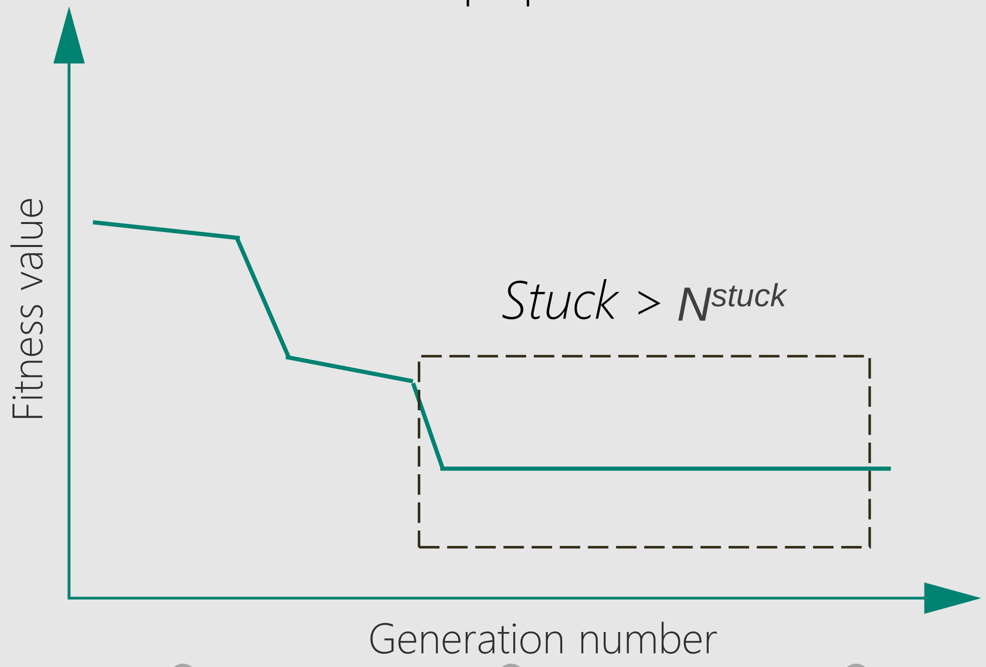
we found that in some cases, all M_{CR} element are set to terminal value when the evolution phase is *very early.*

***This may affect the performance of the algorithm, so we remove the terminal value in mL-SHADE algorithm.*

Memory perturbation

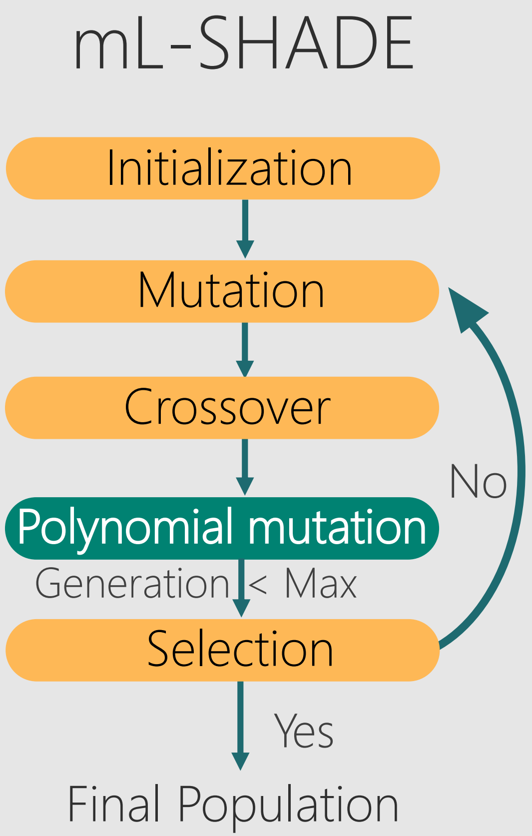
We found that the memory may not be updated for a long time, which means that the fitness value has not improved.

One of the reasons why fitness value stops improving is that the control parameters are not suitable for the current population.

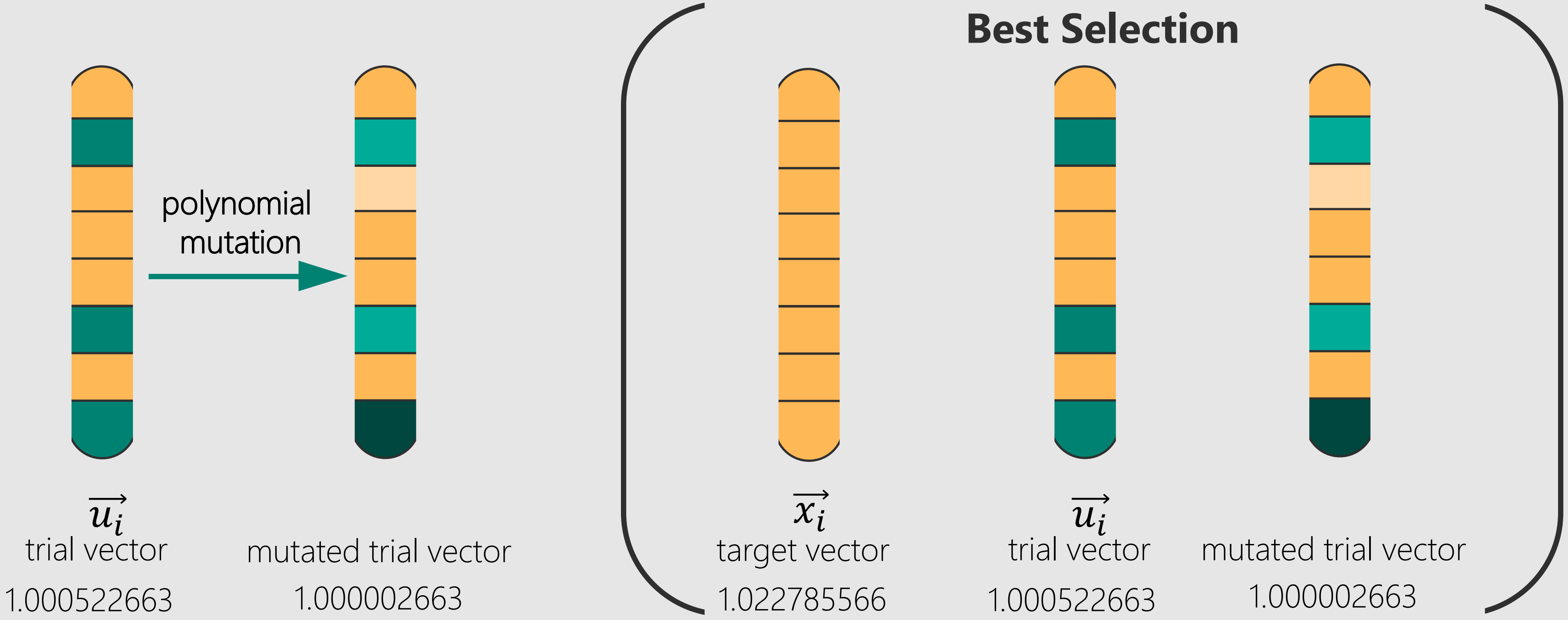


$$M_{CR,k} = 1.0 - M_{CR,k}$$
$$M_{F,k} = 1.0 - M_{F,k}$$

Additional mutation operation (polynomial mutation)



After the trial vector is generated, **polynomial mutation (PM)** is applied to generate a mutated trial vector, and choose the better one to be the final trial vector.



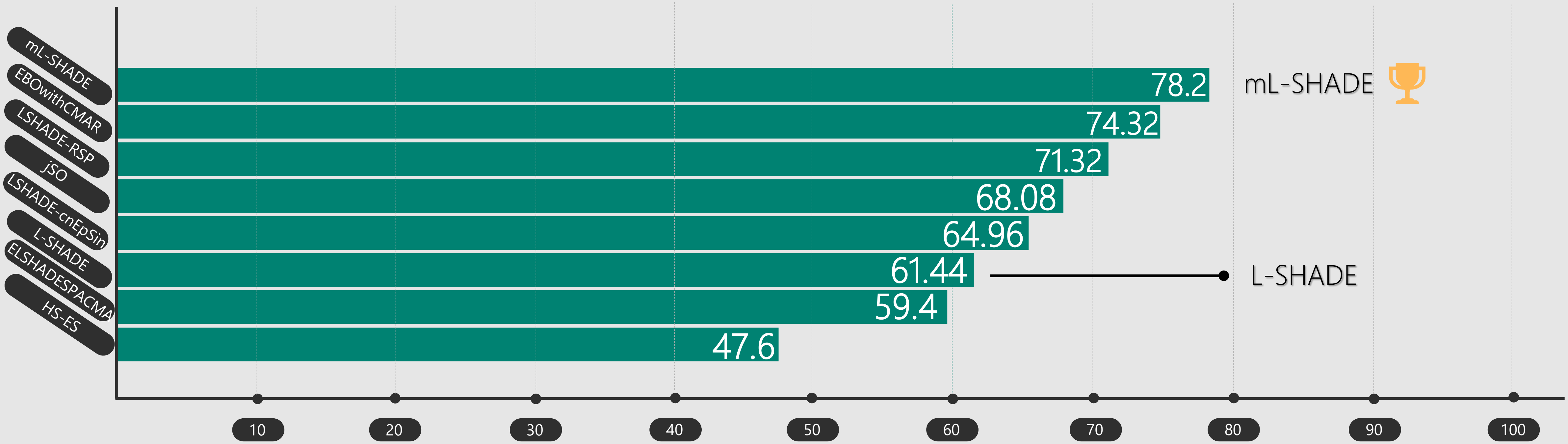
- 100-Digit Challenge on Single Objective Numerical Optimization (CEC C06) was utilized to test the performance of our algorithm
- We compared mL-SHADE with the other seven algorithms including L-SHADE
- The source code of those algorithms can be downloaded from organizer's website.

Parameter setting

Parameter	Meaning	mL-SHADE	L-SHADE
N^{init}	size of the initial population	$18 \cdot D$	$18 \cdot D$
N_{min}	minimal population size	4	4
H	size of the history memory	6	6
r^{arc}	archive size $\|A\|$ = round($r^{arc} \cdot N^{init}$)	1.0	2.6
p	required in the cur-to-$p$best/1 mutation	0.11	0.11
m_r	probability of polynomial mutation	0.05	N/A
p_m, η	parameters of polynomial mutation	$1/D$, 10	N/A
$MaxNFE$	maximum number of fitness evaluations	$2 \cdot 10^6$	$10000 \cdot D$

No.	N^{stuck}	No.	N^{stuck}
1	400	6	400
2	400	7	400
3	6 (same as H)	8	400
4	400	9	6 (same as H)
5	400	10	400

Results


mL-SHADE
EBOwithCMAR
LSHADE-RSP
jSO
LSHADE-cnEpSin
L-SHADE
ELSHADESPACMA
HS-ES
78.2
74.32
71.32
68.08
64.96
61.44
59.4
47.6
mL-SHADE
L-SHADE
10
20
30
40
50
60
70
80
90
100

Results

Ranking:

1. mL-SHADE
2. EBOwithCMAR
3. L-SHADE-RSP
4. jSO
5. L-SHADE-cnEpSin
6. L-SHADE
7. ELSHADESPACMA
8. HS-ES

No.	mL-SHADE	L-SHADE	ELSHADESPACMA	jSO
1	10	10	10	10
2	10	10	10	10
3	10	7.16	5.44	10
4	10	0.24	0.84	3.88
5	10	10	10	10
6	10	10	10	10
7	5.04	1	1.08	1
8	1	1	1.16	1.08
9	2.16	2.04	3	2.12
10	10	10	7.88	10
Score	78.2	61.44	59.4	68.08

No.	LSHADE-RSP	LSHADE-cnEpSin	EBOwithCMAR	HS-ES
1	10	10	10	7.6
2	10	10	10	0
3	10	6.12	10	1.72
4	6.76	4.6	10	4.96
5	10	10	10	10
6	10	10	10	10
7	1.36	0.84	0.56	0.16
8	1	1.04	1.68	0.16
9	2.2	2.36	2.08	3
10	10	10	10	10
Score	71.32	64.96	74.32	47.6

Results

- We made a minor changes after submitting the paper, which let us get a higher score.

The repair method of CR value is modified.

- Rule 1: If CR is bigger than 1, CR is set as 1.
- Rule 2: If CR is negative, it will be set as its absolute value, then apply the Rule 1.

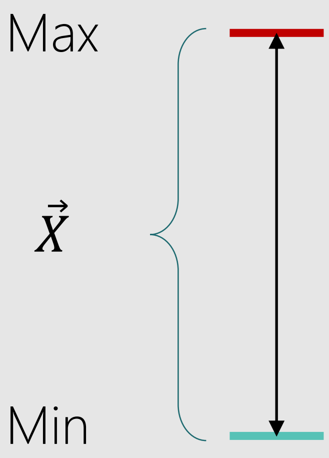
No.	mL-SHADE
1	10
2	10
3	10
4	10
5	10
6	10
7	8.12
8	1.08
9	2.12
10	10
Score	81.32

- In our experiments, we found three problems (7-9) that are difficult to all tested algorithms. No single algorithm can get the highest score in more than one of them.
- For the future work, we will continue our research to study how different algorithms fit different functions and then to propose a better integration.
- Another direction is to develop an adaptive method to adjust the value of the important parameter N^{stuck} in our memory perturbation mechanism.

Thanks for your attention

Department of Computer Science and Information Engineering,
National Taiwan Normal University, Taiwan

Uniform random initialization



- Basically, the parameter of each initial vector is be selected randomly by using the below equation.
- The variable $x_{j,min}$ and $x_{j,max}$ are the minimum and maximum value of each parameter. $rand_{i,j}$ is the random real value between 0 to 1.

$$x_{j,i,G} = x_{j,min} + rand_{i,j}[0,1] * (x_{j,max} - x_{j,min})$$

Algorithms overview

Current-to-pbest/1 strategy

$$v_{i,g} = x_{i,g} + F_i \cdot (x_{pbest,g} - x_{i,g}) + F_i \cdot (x_{r1,g} - x_{r2,g})$$

Where $v_{i,g}$ is the mutated vector of $x_{i,g}$

$x_{pbest,g}$ is randomly selected from top 100p% solutions in the current population.

$x_{r1,g}$ is randomly selected from current population.

$x_{r2,g}$ is randomly selected from the combination of current population and the set of repents which are replaced by trial vector.

Algorithms overview

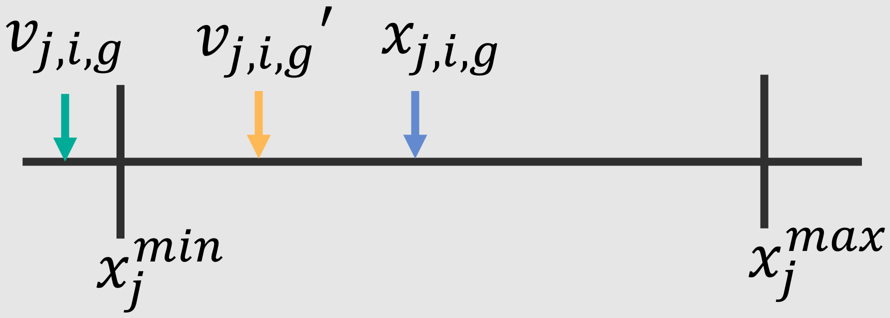
- Gene repair

$$v_{j,i,g}' = \begin{cases} (x_j^{min} + x_{j,i,g})/2 & if \ v_{j,i,g} < x_j^{min} \\ (x_j^{max} + x_{j,i,g})/2 & if \ v_{j,i,g} > x_j^{max} \end{cases}$$

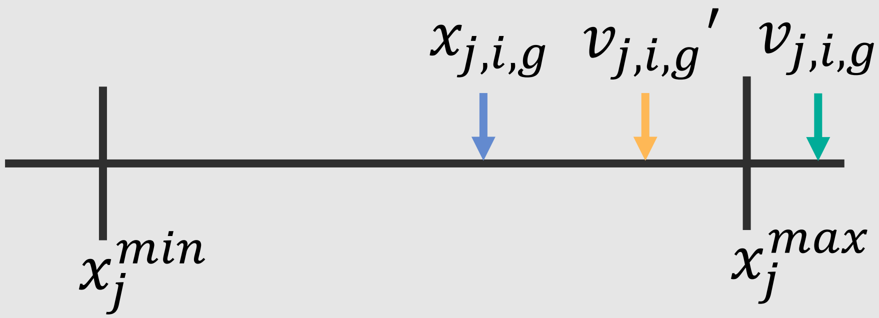
Case 1

Case 2

Case 1



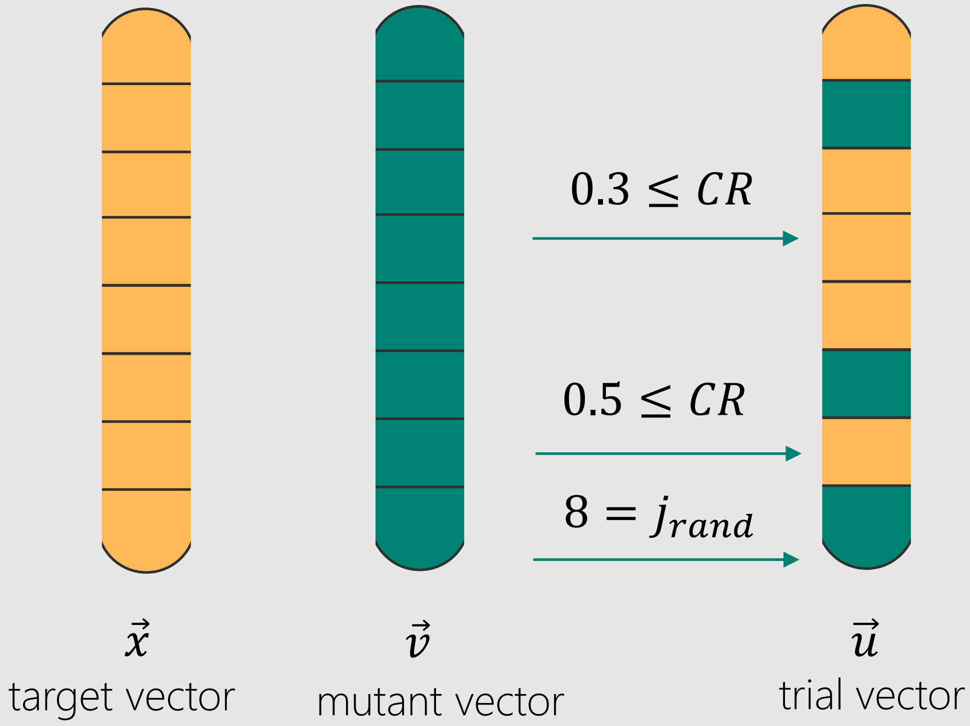
Case 2



Algorithms overview

Binomial crossover

$$u_{j,i,g} = \begin{cases} v_{j,i,g} & \text{if } rand[0,1) \leq CR_i \text{ or } j = j_{rand} \\ x_{j,i,g} & \text{otherwise} \end{cases}$$



$CR = 0.5$

$j_{rand} = 8$

Algorithms overview

Remove Terminal value

selection

The trial vector $\overrightarrow{u_{i,g}}$ will be selected to be the candidate solution in the iteration if its fitness is not better than the target vector $\overrightarrow{x_{i,g}}$.

$$x_{i,g+1} = \begin{cases} u_{i,g} & if\ f\left(u_{i,g}\right) \leq f(x_{i,g}) \\ x_{i,g} & otherwise \end{cases}$$

Algorithms overview