IEEE WCCI 2020

IEEE World Congress on Computational Intelligence

Virtual Conference – July 19-24, 2020

Multi-population Modified L-SHADE for Single Objective Bound Constrained Optimization

Yann-Chern Jou, Shuo-Ying Wang, Jia-Fong Yeh and Tsung-Che Chiang

Department of Computer Science and Information Engineering,

National Taiwan Normal University, Taiwan

Outline

1. Introduction
2. DE / L-SHADE / mL-SHADE
3. Proposed algorithm – Multi-population Modified L-SHADE (mpmL-SHADE)
4. Experiments and Results
5. Conclusion

Introduction

- Differential evolution (DE) is one of the most popular evolutionary algorithms in solving real-parameter optimization problems.

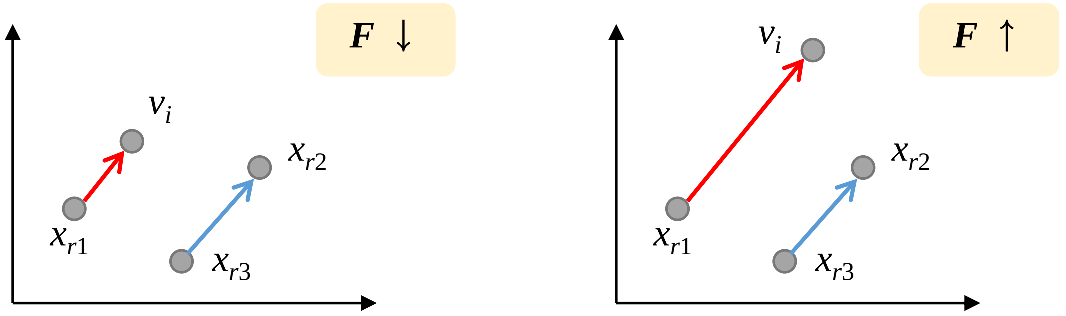
- We proposed a variant of DE with success history adaptive parameter control.
  - Multi-population Modified L-SHADE (mpmL-SHADE)

- Participate CEC2020 Single Objective Bound Constrained Optimization Competition.

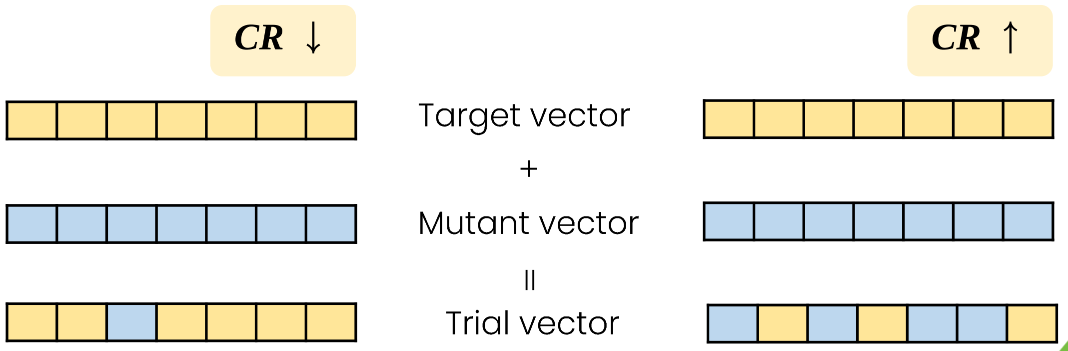
Differential Evolution (DE)

- Scaling Factor ***F***

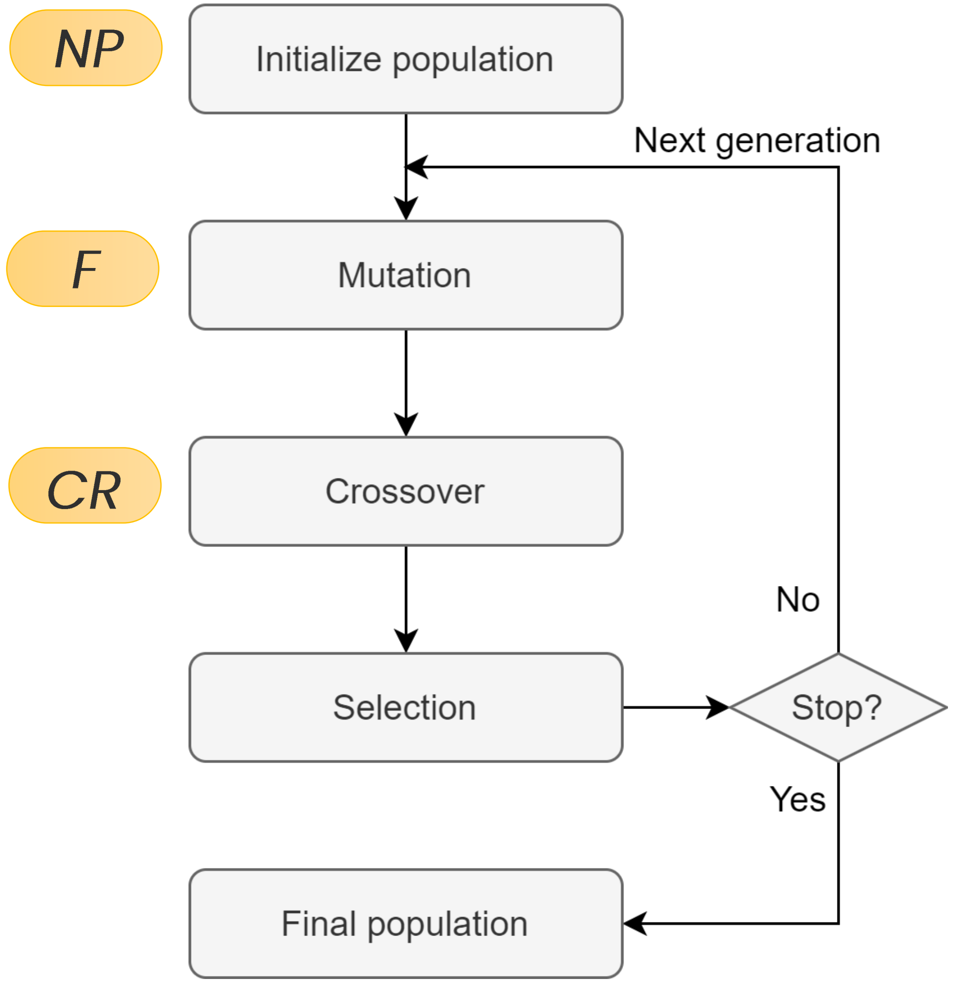
$$v_i = x_{r1} + \boldsymbol{F} \cdot (x_{r2} - x_{r3})$$



- Crossover Rate ***CR***



DE flowchart



Adaptive DE

Success history

- SaDE

  (Qin and Suganthan , 2005)

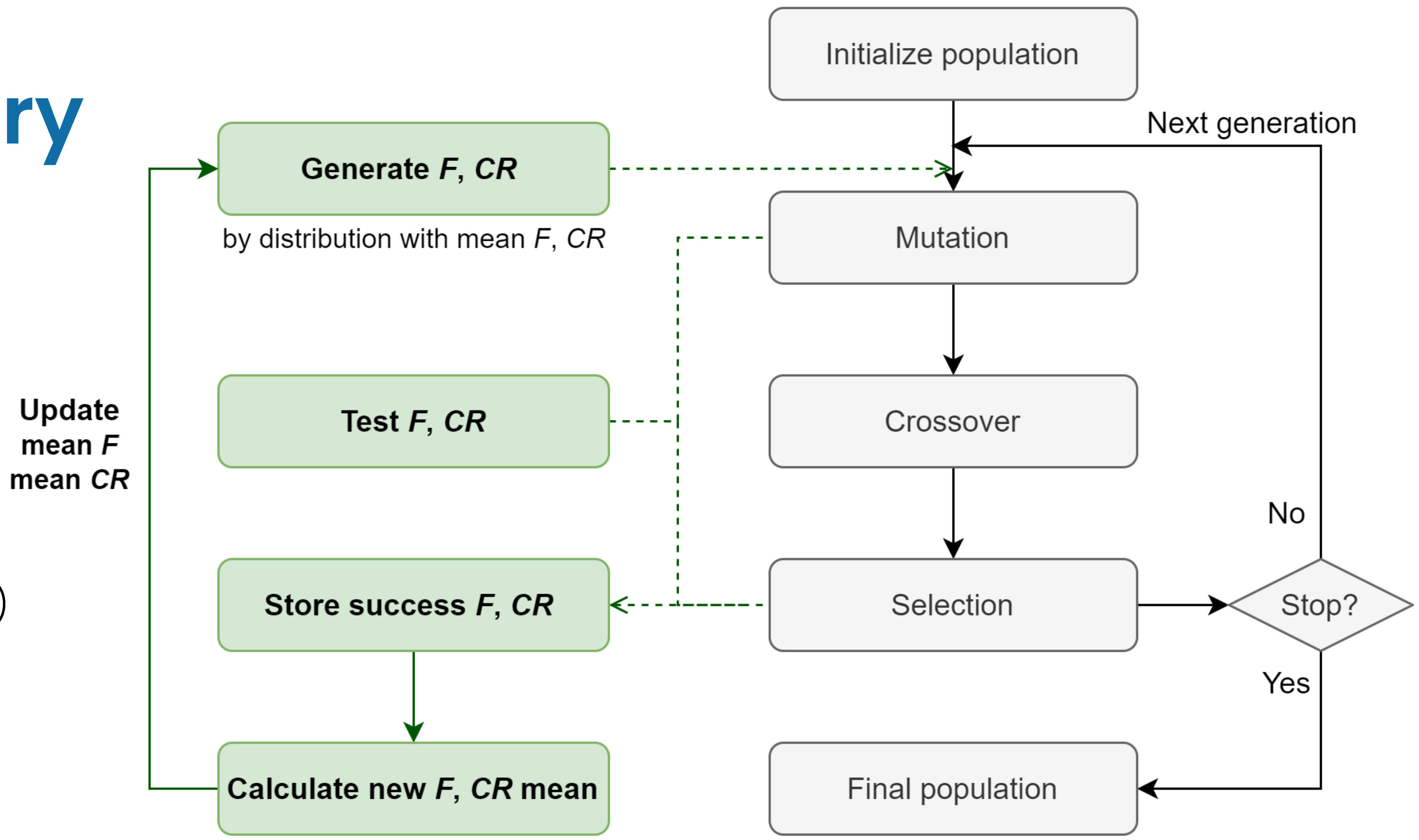
- JADE

  (Zhang and Sanderson, 2009)

- SHADE

  (Tanabe and Fukunaga, 2013)

Success history based DE flowchart



L-SHADE (Tanabe et al., 2014)

Success history memory

- Record **multiple** pairs of successful mean values of *F* and *CR*.
  - M_F : mean value of *F*
  - M_{CR} : mean value of *CR*
  - *H* : the size of success history memory

H

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

- Use different ranges of parameter values on different individuals in a generation.

SPONSORS:

L-SHADE (Tanabe and Fukunaga, 2014)

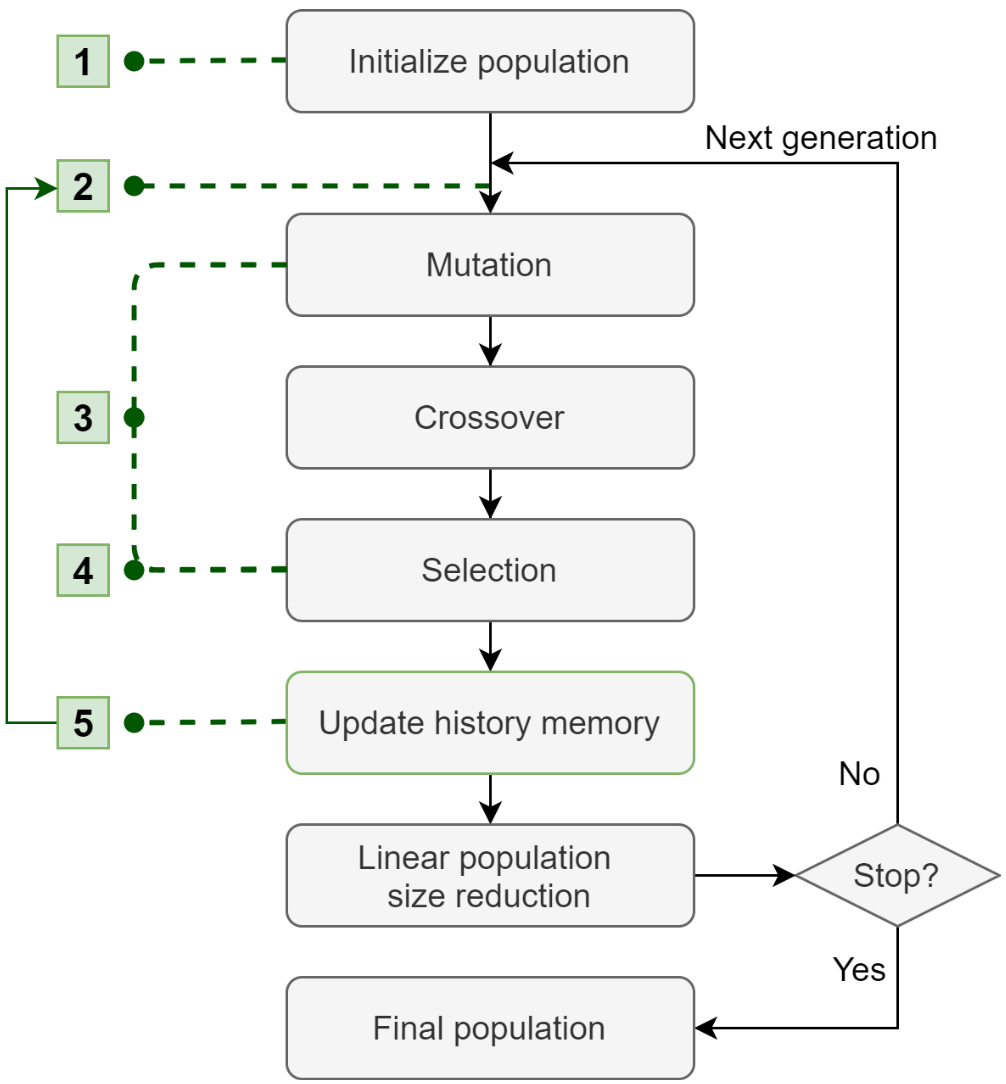
Success history memory

H

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

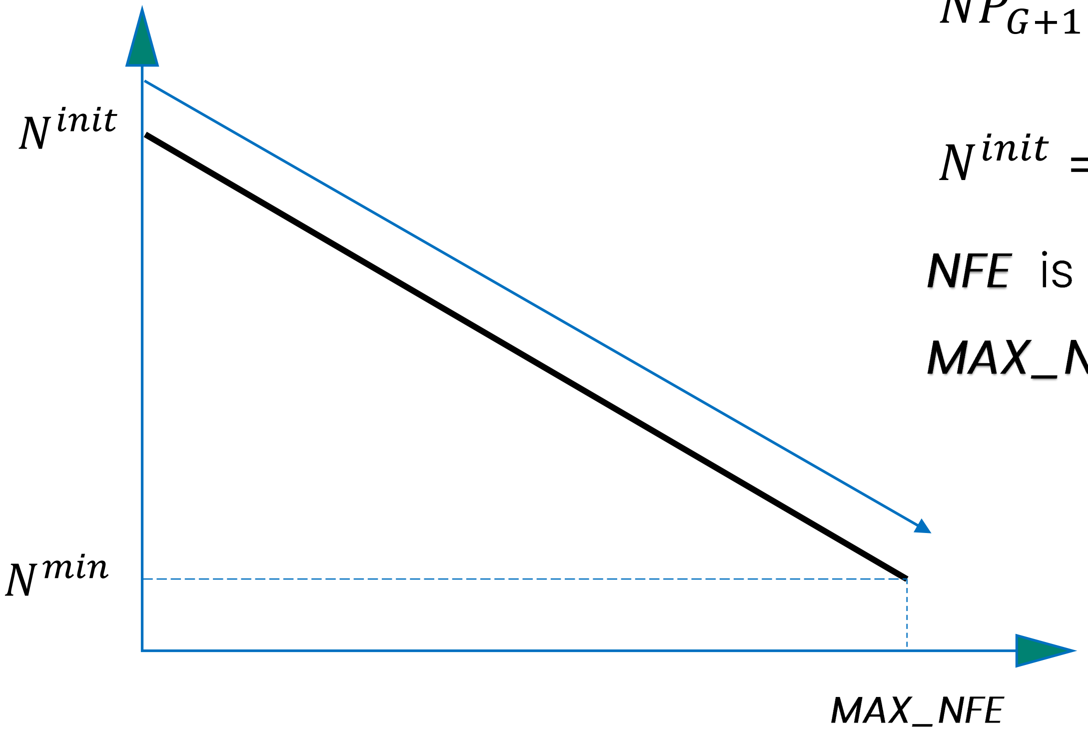
1. Creating history memory table.
2. Randomly select a pair of M_F and M_{CR} from history memory for each target vector to generate F and CR.
3. Testing F and CR in each iteration.
4. Store the successful F and CR value.
5. Update by calculating weighted Lehmer mean

L-SHADE flowchart



L-SHADE (Tanabe and Fukunaga, 2014)

Linear population size reduction (LPSR)



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

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

NFE is the current number of fitness evaluations

MAX_NFE is the maximum number of fitness evaluations

mL-SHADE (Yeh et al., 2019)

Removal of terminal value

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

Observation (L-SHADE):

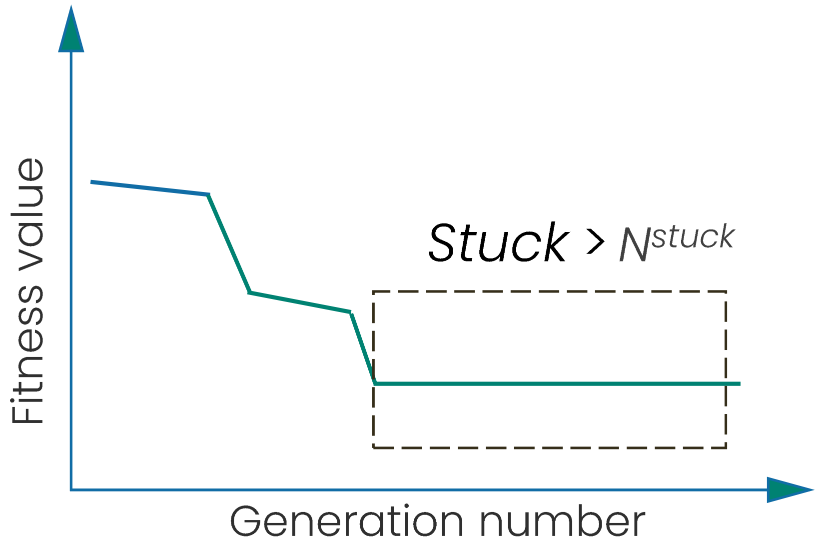
If the new mean of *CR* is zero, L-SHADE will set the $M_{CR,k}$ by a **terminal value** $\perp$ and will never update it. When $\perp$ is chosen, *CR* is set by zero and thus only a single gene is exchanged. Sometimes all M_{CR} elements are set to $\perp$ at the early stage of the evolutionary process. This leads to too strong exploitation.

Solution (mL-SHADE) :

mL-SHADE removes the terminal value to keep the exploration ability.

mL-SHADE (Yeh et al., 2019)

History Memory perturbation



Observation (L-SHADE):

- The fitness value stops improving because the control parameter values are not suitable for the current population.
- The memory **may not be updated for a long time**.

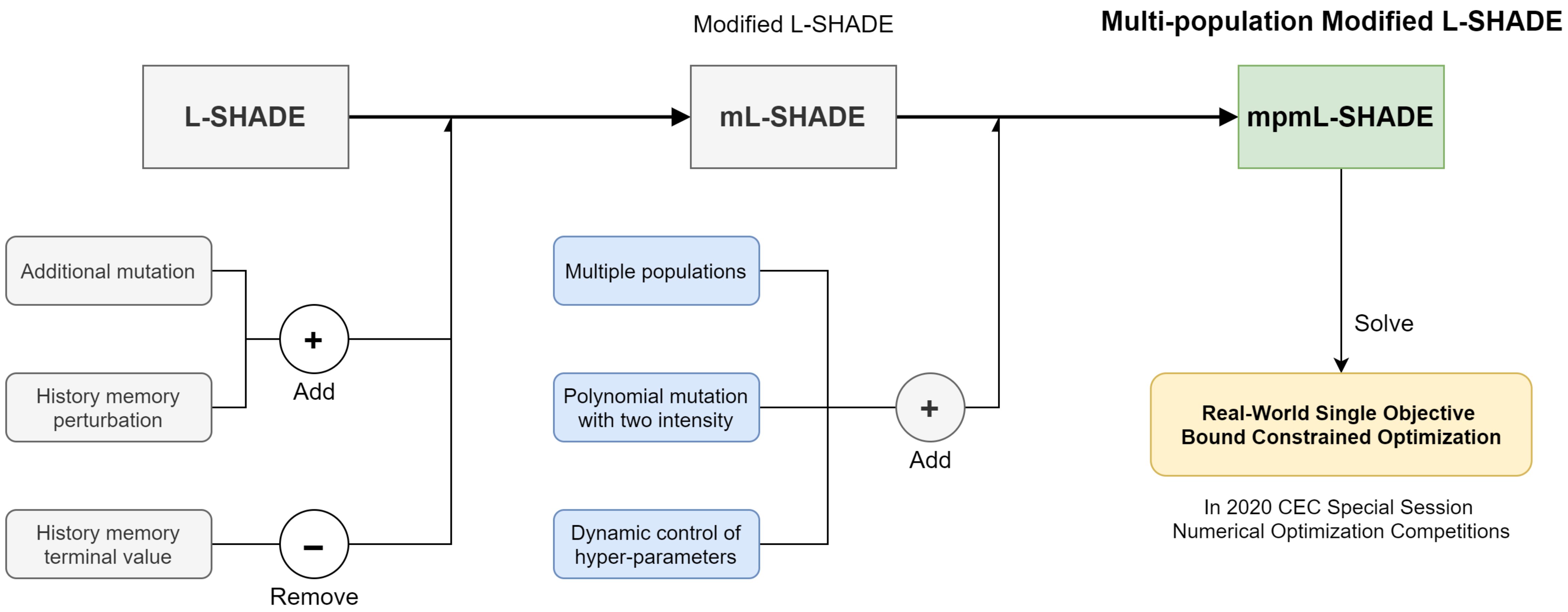
Solution (mL-SHADE):

- If the evolution gets stuck for N^{stuck} generations, mL-SHADE perturbs the success history memory by resetting one of them.

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

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

Multi-population Modified L-SHADE



[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.
[mL-SHADE] J. Yeh, T. Chen and T. Chiang, "Modified L-SHADE for single objective real-parameter optimization," in IEEE CEC, pp. 381-386, 2019.

mpmL-SHADE

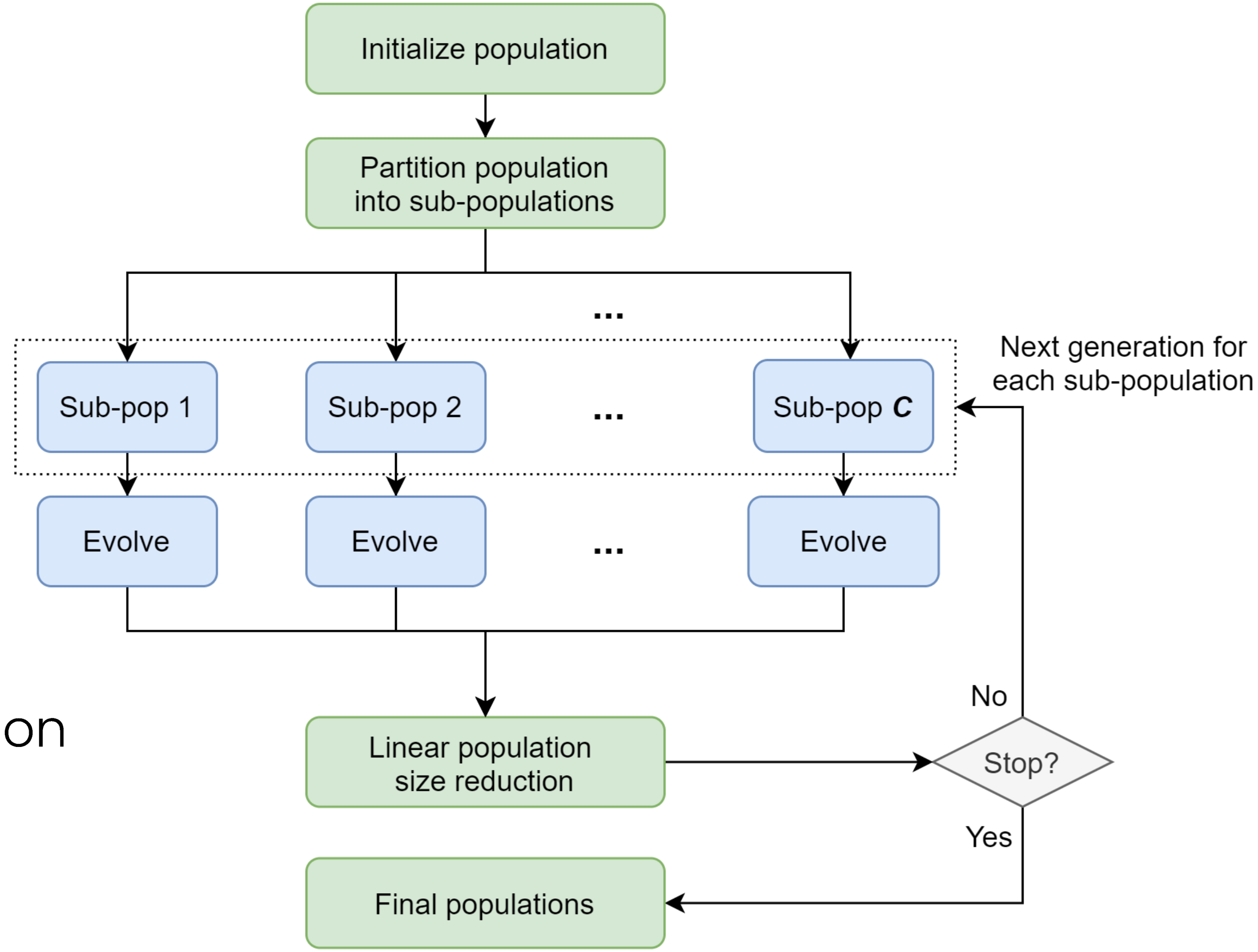
Algorithm overview

- Multiple population

  with cluster partition

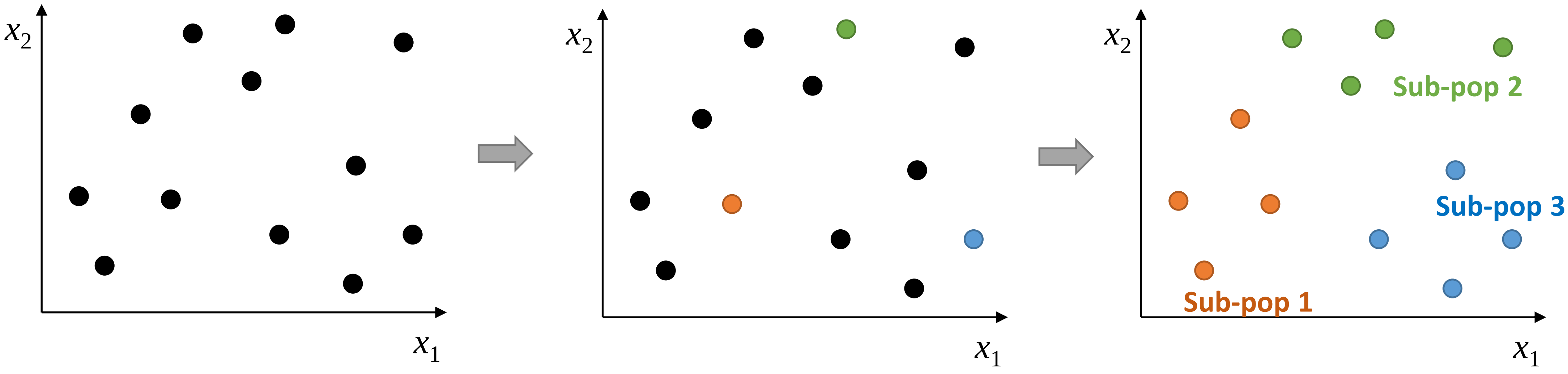
- Evolve by mL-SHADE

  with two intensity polynomial mutation



Multiple populations

- An effective approach to enhance the diversity and the search ability for DE to solve optimization problems.

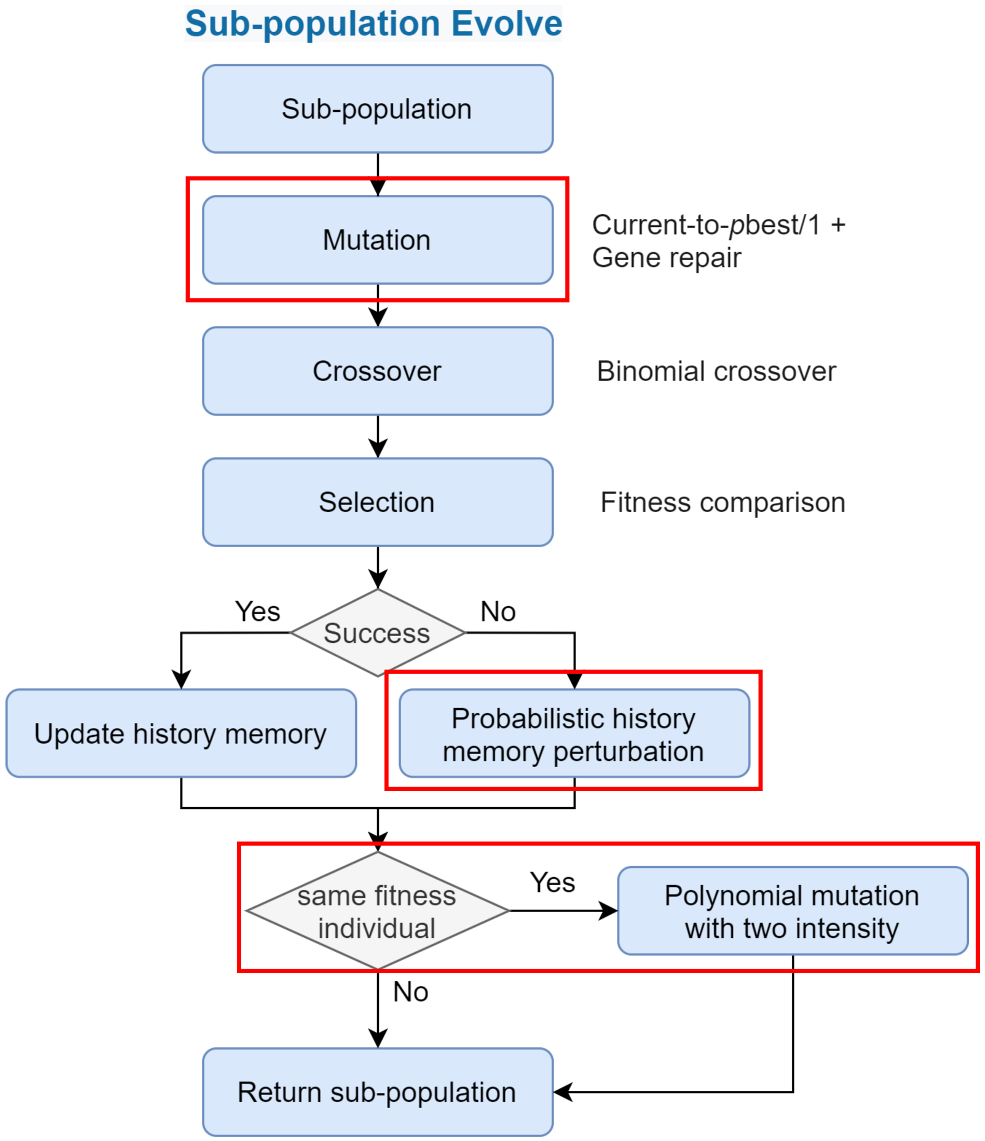


W. Gao, G. G. Yen and S. Liu, “A cluster-based differential evolution with self-adaptive strategy for multimodal optimization,” *IEEE Transactions on Cybernetics*, vol. 44, no. 8, pp. 1314–1327, Aug. 2014

mpmL-SHADE

Algorithm overview

- Based on mL-SHADE
- Polynomial mutation with two intensity
- Dynamic parameter control
  - Probabilistic history memory perturbation
  - Linear scale parameter increment (LSPI) mechanism



mpmL-SHADE

Polynomial mutation with two intensity

- When multiple individuals have the same fitness value in a generation, we mutate the individual to increase the diversity of population.

- It is controlled by two values of the parameter η.
  - Exploitation, while η value is large.
  - Exploration, while η value is small.

```
while fitness x_i == fitness x_j do
    if randu(0.0, 1.0) ≤ nfe/MAX_NFE then
        x_j = PolynomialMutation(x_j,1.0/D, big η)
    else
        x_j = PolynomialMutation(x_j,1.0/D, small η)
    nfe = nfe + 1
end while
```

M. Hamdan, "The distribution index in polynomial mutation for evolutionary multiobjective optimisation algorithms: An experimental study", In: Proceedings of International Conference on Electronics Computer Technology, 2012.

mpmL-SHADE

Probabilistic history memory perturbation

- Remove the fix parameter N^{stuck} in mL-SHADE.

if $stuck \geq N^{stuck}$ **then**

Memory Perturbation

end if

→

if $randu(0.0, 1.0) \leq nfe^{ns} / nfe)$ **then**

Memory Perturbation

end if

nfe^{ns}: The number of fitness evaluations since last improvement.

mpmL-SHADE

Linear scale parameter increment (LSPI)

Observation:

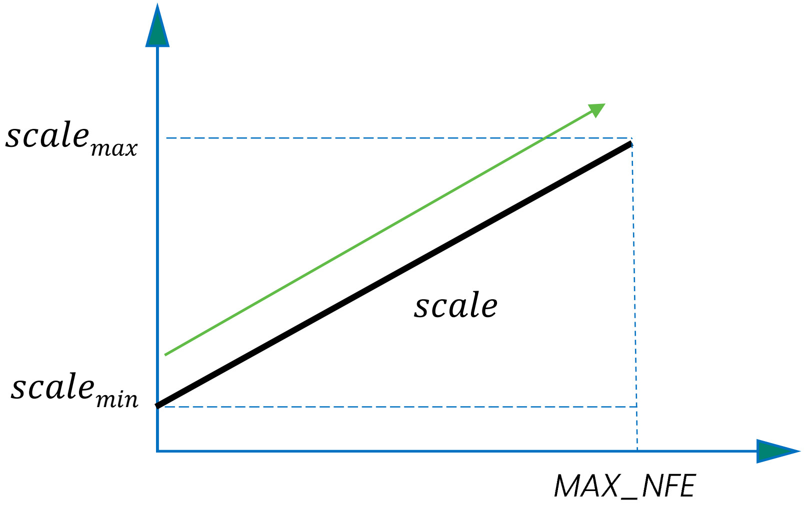
In the late phase of evolution, M_F value in the memory are all getting smaller.

Solution:

- LSPI increase the scale of **Cauchy distribution** for scale factor *F*.
- It provides Cauchy distribution more chance to generate bigger *F* value for exploration.

$$\boldsymbol{scale} = scale_{min} + \frac{(scale_{max} - scale_{min}) \times nfe}{MAX\_NFE}$$

$$F_i = randc(M_{F,r}, \boldsymbol{scale})$$



Experiments and Results

Benchmark functions

- CEC2020 Single Objective Bound Constrained Optimization Competition
  - Ten test functions
  - Four different dimensions: D = 5, 10, 15, 20
  - Search ranges of decision variables: [−100, 100]
  - Maximum number of fitness function evaluations:

Problem Dimension	*MAX_NFE*
$D = 5$	50,000
$D = 10$	1,000,000
$D = 15$	3,000,000
$D = 20$	10,000,000

C. T. Yue, K. V. Price, P. N. Suganthan, J. J. Liang, M. Z. Ali, B. Y. Qu, N. H. Awad, and Partha P Biswas , "Problem definitions and evaluation criteria for the CEC 2020 special session and competition on single objective bound constrained numerical optimization," Technical Report, NanyangTechnological University, Singapore, November 2019.

Experiments and Results

Parameter setting

Parameter	Meaning	Value
N^{init}	size of the initial population	$18D \cdot C$
N^{min}	minimal sub-population size	4
H	size of the success history memory	6
r^{arc}	archive size $\lvert\boldsymbol{A}\rvert$ = round($r^{arc} \cdot N^{subpop}$)	2.6
p	required in cur-to-*p*best/1 mutation	0.11
$[M_c^F]^0, [M_c^{CR}]^0$	initial values of c^{th} sub-population's F/CR memory	0.5
C	the number of sub-populations	$0.2 \cdot D$
$scale_{min}$	required in LSPI	0.1
$scale_{max}$	required in LSPI	0.2
p_m, η	parameters of polynomial mutation	$1/D$, 5 or 20

L-SHADE

mpmL-SHADE

Experiments and Results

Analysis of the number of sub-populations

We compared the six variants by the evaluation criteria in the CEC2020 competition.

Score 1 : based on sums of normalized best error values (50%)

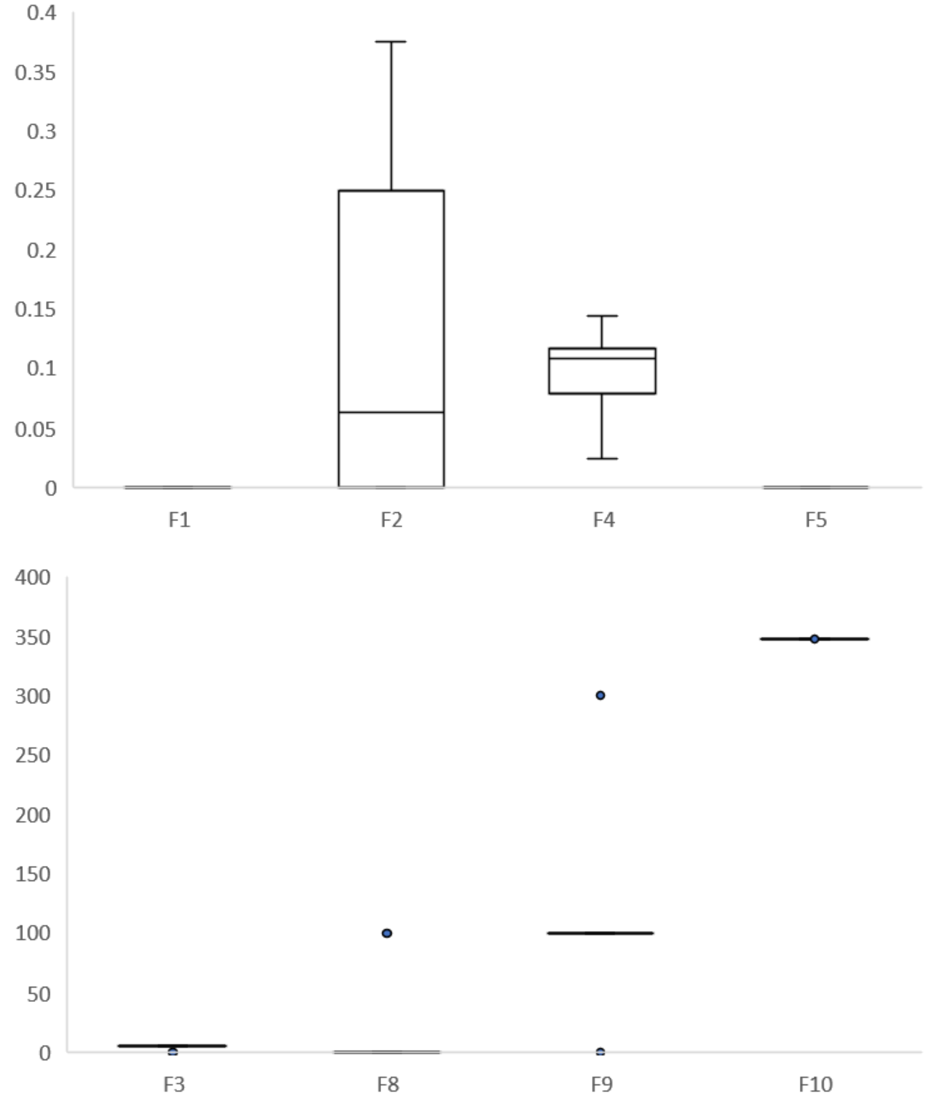
Score 2: based on sums of weighted rank mean error values (50%)

Number of Sub-populations	**0.2·*D***	0.3·*D*	0.4·*D*	0.5·*D*	0.6·*D*	0.7·*D*
Score1	50	48.2995	44.1225	41.7733	45.9315	39.7564
Score2	49.0814	47.9487	50	47.1033	44.6301	43.7939
Total Score	99.0814	96.2482	94.1225	88.8766	90.5615	83.5503

Experiments and Results

Error value of mpmL-SHADE (D = 5)

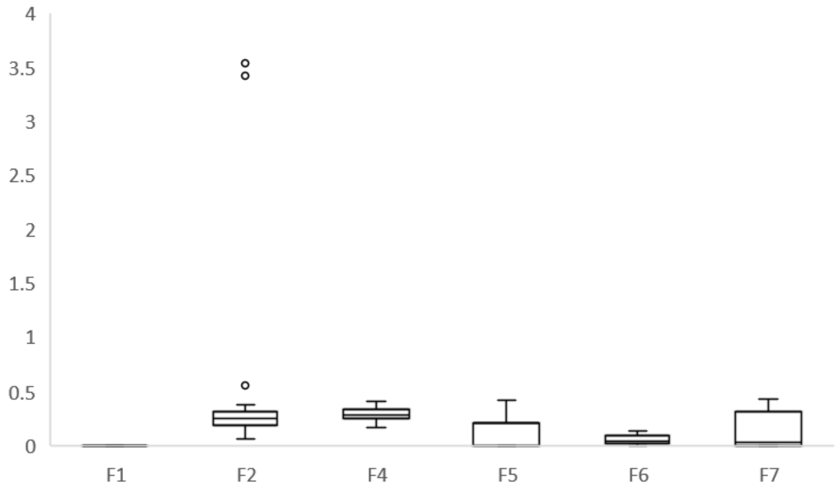
Func.	Best	Worst	Median	Mean	Std
f_1	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
f_2	0.0000E+00	3.7470E–01	6.2947E–02	1.1661E–01	1.3654E–01
f_3	0.0000E+00	5.4146E+00	5.1482E+00	4.7000E+00	1.4661E+00
f_4	2.3494E–02	1.4360E–01	1.0851E–01	9.6084E–02	3.0934E–02
f_5	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
f_8	0.0000E+00	1.0000E+02	0.0000E+00	3.3333E+00	1.7951E+01
f_9	0.0000E+00	3.0000E+02	1.0000E+02	1.0333E+02	4.0689E+01
f_{10}	3.0000E+02	3.4737E+02	3.4737E+02	3.4105E+02	1.6102E+01

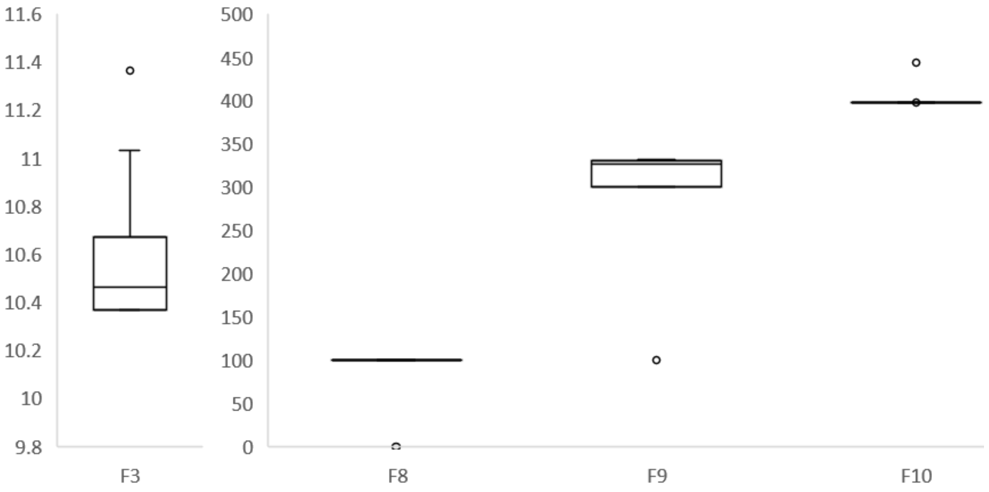


Experiments and Results

Error value of mpmL-SHADE (D = 10)

Func.	Best	Worst	Median	Mean	Std
f_1	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
f_2	6.3009E–02	3.6023E+00	2.5191E–01	6.8095E–01	1.1259E+00
f_3	1.0367E+01	1.1365E+01	1.0461E+01	1.0565E+01	2.3384E–01
f_4	1.6905E–01	4.0592E–01	2.7884E–01	2.8404E–01	5.8284E–02
f_5	0.0000E+00	4.1629E–01	2.0814E–01	1.2489E–01	1.1526E–01
f_6	1.5651E–04	1.3554E–01	3.8008E–02	5.0894E–02	4.0899E–02
f_7	7.0608E–06	4.3363E–01	2.7867E–02	1.3732E–01	1.5459E–01
f_8	0.0000E+00	1.0000E+02	1.0000E+02	9.6667E+01	1.7951E+01
f_9	0.0000E+00	3.3186E+02	3.2635E+02	2.7606E+02	9.8182E+01
f_{10}	3.9774E+02	4.4578E+02	3.9801E+02	4.0260E+02	1.3855E+01

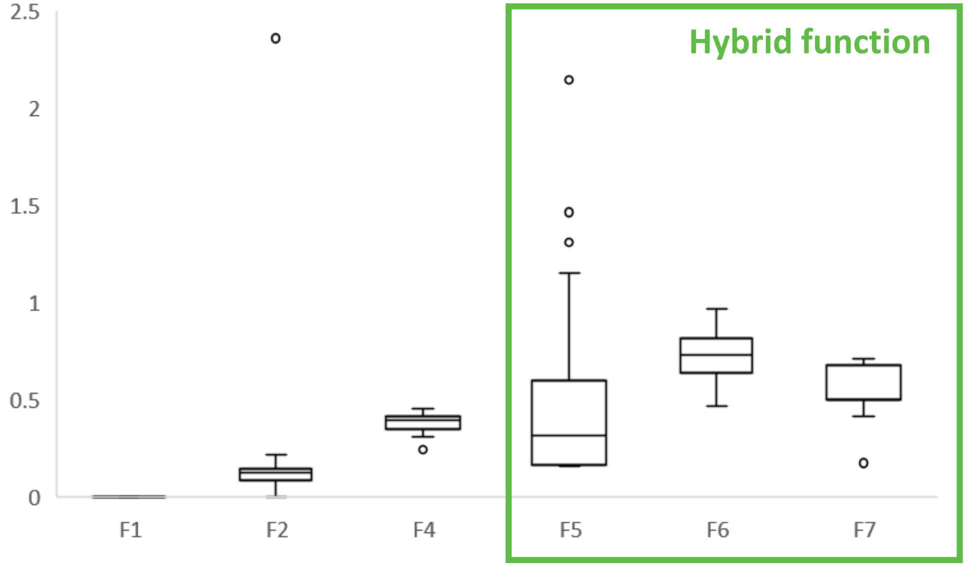


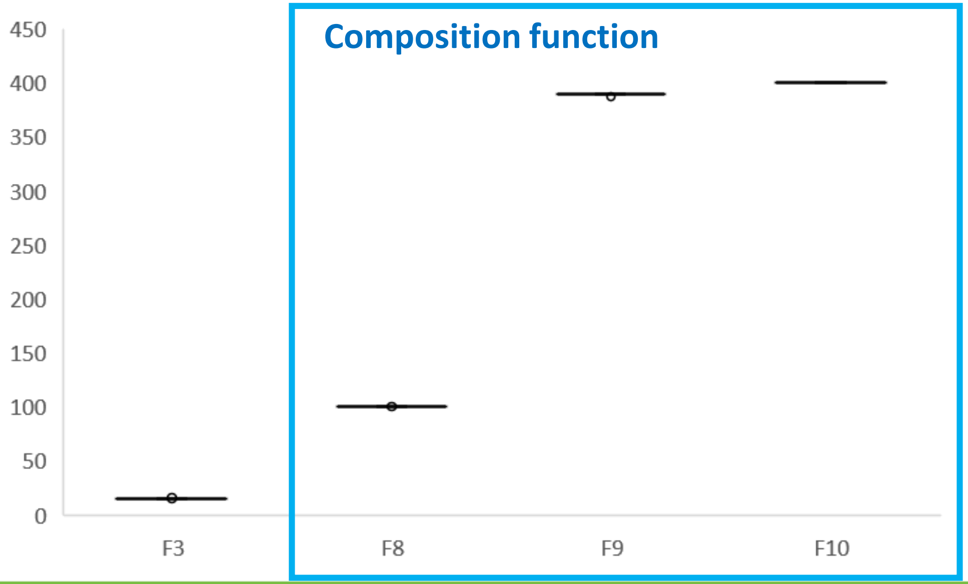


Experiments and Results

Error value of mpmL-SHADE (D = 15)

Func.	Best	Worst	Median	Mean	Std
f_1	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
f_2	0.0000E+00	2.3601E+00	1.2491E–01	1.8306E–01	4.0725E–01
f_3	1.5567E+01	1.5567E+01	1.5567E+01	1.5567E+01	2.0963E–13
f_4	2.4114E–01	4.5490E–01	3.9389E–01	3.8242E–01	4.4318E–02
f_5	1.5612E–01	2.1460E+00	3.1224E–01	5.4042E–01	5.0447E–01
f_6	4.6535E–01	9.6453E–01	7.2995E–01	7.2517E–01	1.2459E–01
f_7	1.7299E–01	7.0814E–01	5.0000E–01	5.4320E–01	1.1498E–01
f_8	1.0000E+02	1.0000E+02	1.0000E+02	1.0000E+02	4.9118E–13
f_9	3.8708E+02	3.8968E+02	3.8968E+02	3.8959E+02	4.6649E–01
f_{10}	4.0000E+02	4.0000E+02	4.0000E+02	4.0000E+02	4.9815E–13

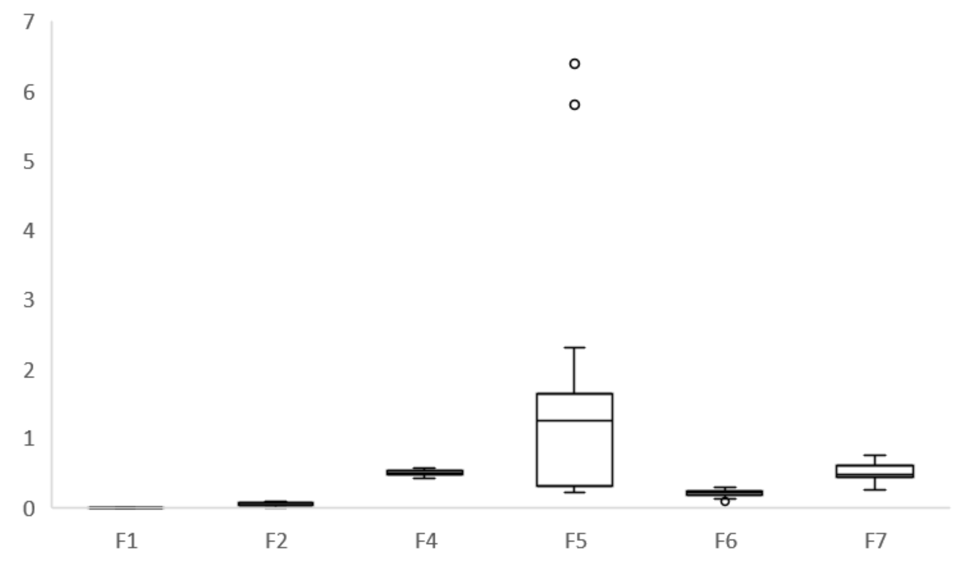


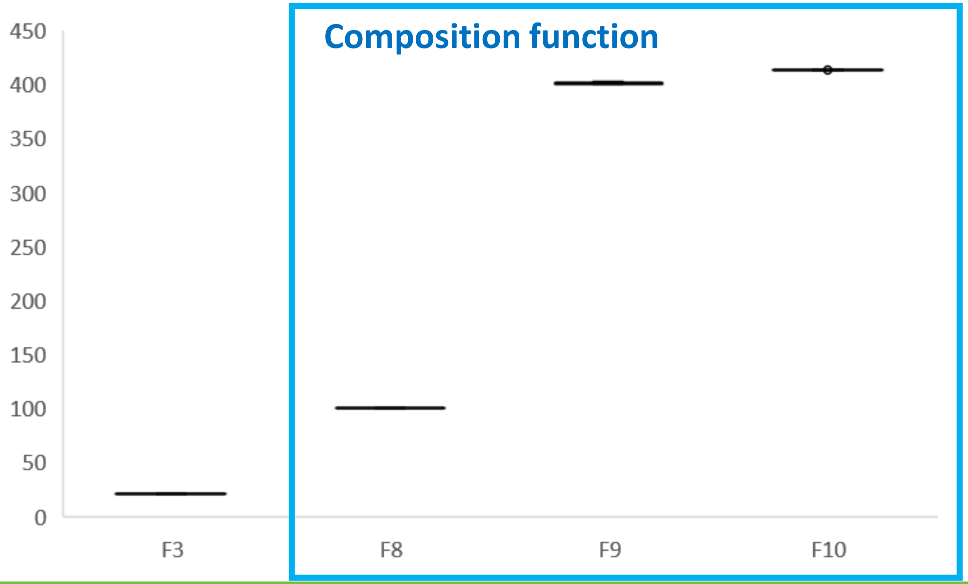


Experiments and Results

Error value of mpmL-SHADE (D = 20)

Func.	Best	Worst	Median	Mean	Std
f_1	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
f_2	0.0000E+00	9.3685E–02	3.1228E–02	3.9673E–02	2.1195E–02
f_3	2.0387E+01	2.0387E+01	2.0387E+01	2.0387E+01	4.6736E–13
f_4	4.1611E–01	5.6583E–01	5.0069E–01	4.9748E–01	4.2334E–02
f_5	2.0817E–01	6.3861E+00	1.2552E+00	1.3801E+00	1.4521E+00
f_6	7.9020E–02	2.9642E–01	2.1061E–01	2.0522E–01	4.7101E–02
f_7	2.5517E–01	7.5000E–01	4.8157E–01	5.0994E–01	1.1874E–01
f_8	1.0000E+02	1.0000E+02	1.0000E+02	1.0000E+02	7.4723E–13
f_9	4.0031E+02	4.0278E+02	4.0133E+02	4.0139E+02	6.6822E–01
f_{10}	4.1366E+02	4.1366E+02	4.1366E+02	4.1366E+02	2.7413E–04





Experiments and Results

Performance Comparison

We did *t*-test at the significance level of 0.05. The symbols +, -, and ≈ mean that the proposed mpmL-SHADE is better than (+), worse than (-) and not statistically different from (≈) the compared algorithm.

mpmL-SHADE vs.	mL-SHADE			L-SHADE		
	+	≈	−	+	≈	−
D = 5	1	7	0	0	8	0
D = 10	5	5	0	7	3	0
D = 15	5	4	1	6	4	0
D = 20	3	6	1	7	2	1
Total	**14**	22	2	**20**	17	1

mpmL-SHADE

Conclusions

- We proposed an adaptive differential evolution algorithm **mpmL-SHADE** for solving real-parameter single objective optimization problems. It has three features:
  - multi-population evolution
  - dynamic control of hyper-parameters
  - mutation with dynamic intensity
- The results showed that the multi-population is well perform on CEC2020 benchmark.
- In the future, we will work on:
  - How to form sub-populations and design the interaction between sub-populations.
  - How to dynamically control the exploration and exploitation for maintaining diversity of populations.

Thanks for your attention!

Meta-heuristics Laboratory,
Department of Computer Science and Information Engineering,
National Taiwan Normal University, Taiwan