Numerical Optimization by Nature Inspired Algorithms

Dr. P. N. Suganthan
School of EEE, NTU, Singapore

Some Software Resources Available from:
http://www.ntu.edu.sg/home/epnsugan

ICHSA 2015
Korea University, Seoul, 19th – 21st Aug 2015

Overview

I. Introduction to Real Variable Optimization & DE

II. Future of Real Parameter Optimization

III. Single Objective Optimization

IV. Multimodal Optimization

But, first a little publicity ….

Benchmark Test Functions

Resources available from

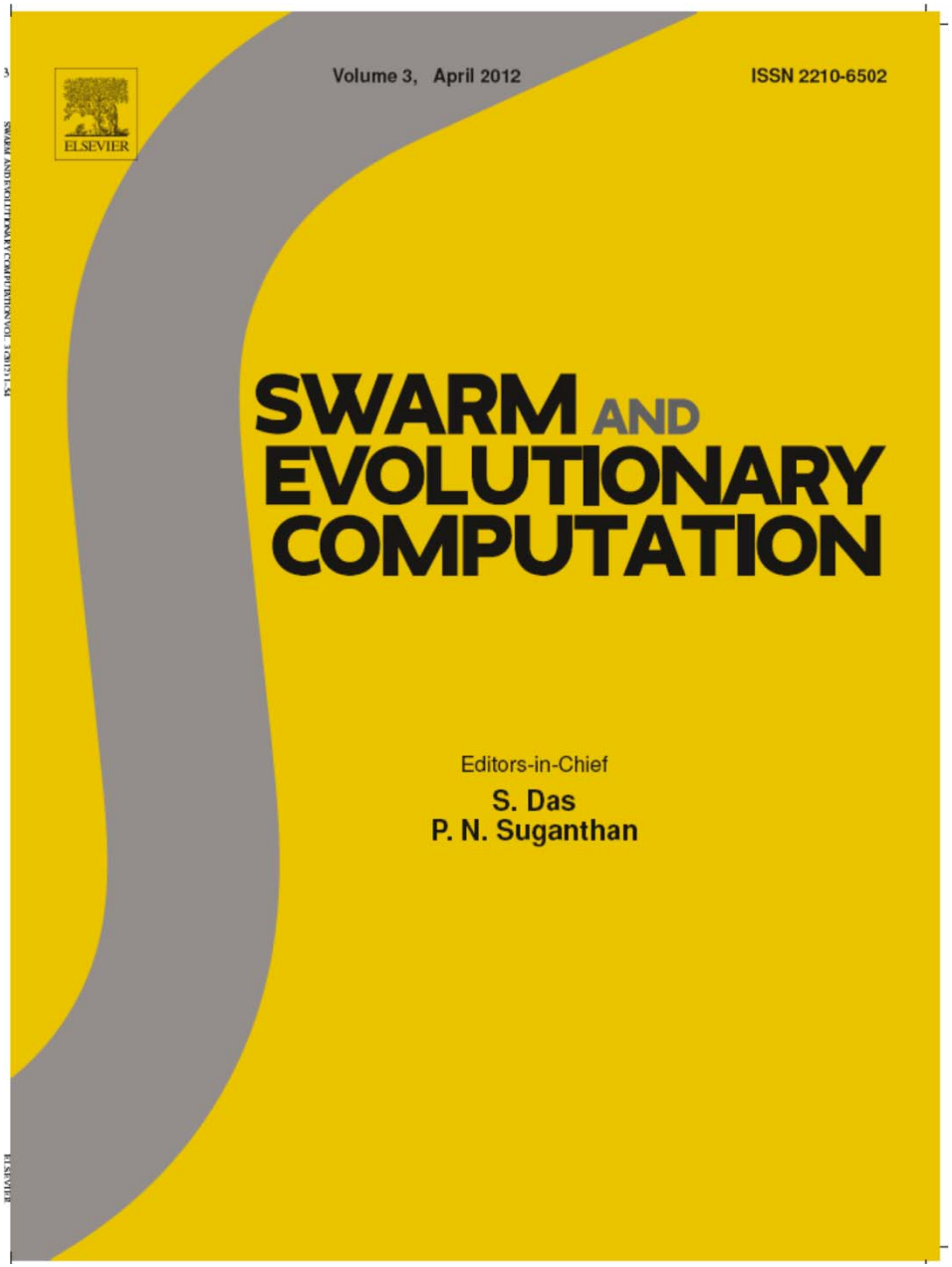
http://www.ntu.edu.sg/home/epnsugan

(limited to our own work & CEC Competitions)

Ensemble / Adaptive Methods for Evolutionary Algorithms:

http://www.ntu.edu.sg/home/epnsugan/index_files/EEAs-EOAs.htm

Please consider submitting to SWEVO journal dedicated to the EC-SI fields. SCI Indexed from Vol. 1, Issue 1.



Overview

I. Introduction to Real Variable Optimization & DE

II. Future of Real Parameter Optimization

III. Single Objective Optimization

IV. Multimodal Optimization

General Thoughts: NFL (No Free Lunch Theorem)

- Glamorous Name for Commonsense?
  - Over a large set of problems, it is impossible to find a single best algorithm
  - DE with Cr=0.90 & Cr=0.91 are two different algorithms → Infinite algos.
  - **Practical Relevance:** Is it common for a practicing engineer to solve several practical problems at the same time?
  - **Academic Relevance:** Very High

Other NFL Like Commonsense Scenarios

Panacea: A medicine to cure all diseases, *Amrita*: the nectar of immortal perfect life …

Silver bullet: in politics … (**you can search these on internet**)

Jack of all trades, but master of none

If you have a hammer all problems look like nails

General Thoughts: Convergence

- What is exactly convergence in the context of EAs & SAs ?
  - The whole population reaching a single point (within a tolerance)
  - Single point search methods & convergence …
- In the context of real world problem solving, are we going to reject a good solution because the population hasn't converged ?

- Good to have all population members converging to the global solution **OR** good to have high diversity even after finding the global optimum ? **(Fixed Computational budget Scenario)**

What we do not want to have:

For example, in the context of PSO, we do not want to have chaotic oscillations

$$c_1 + c_2 > 4.1+$$

General Thoughts: Algorithmic Parameters

- Good to have many algorithmic parameters / operators ?
- Good to be robust against parameter / operator variations ?
- What are Reviewers' preferences ?

Or good to have several parameters that can be adaptively tuned on the fly to achieve top performance on diverse problems?

Why we ALL should think this way? A few good reasons…

CEC 2015 Competitions: "Learning-Based Optimization"

Similar Literature: Thomas Stützle, Holger Hoos, …

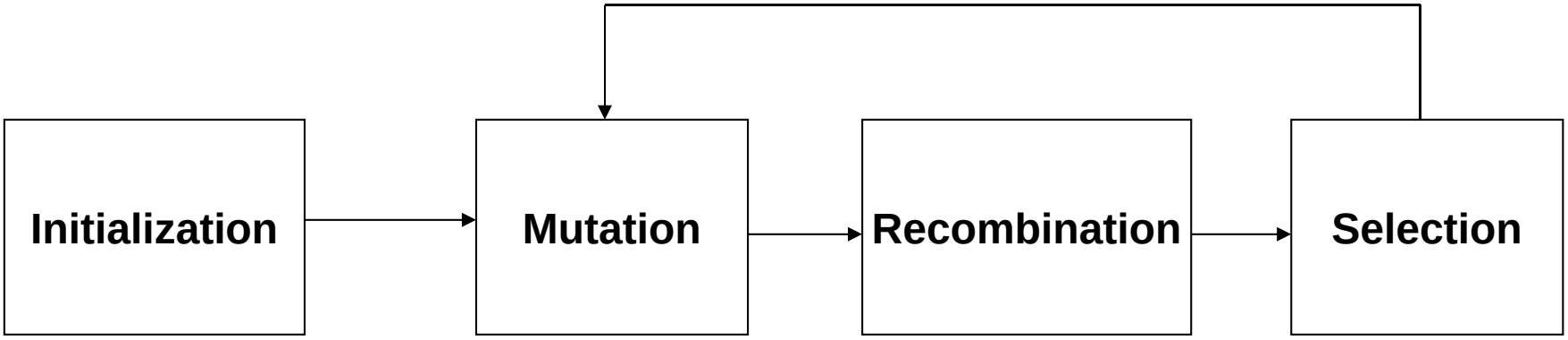
General Thoughts: Nature Inspired Methods

- Good to mimic closely natural phenomena?
- Honey bees solve only one problem (gathering honey). Can this (ABC or BCO) be the best approach for solving all problems?
- NFL & Nature inspired methods.
- Nature inspired methods & crossover.

Differential Evolution

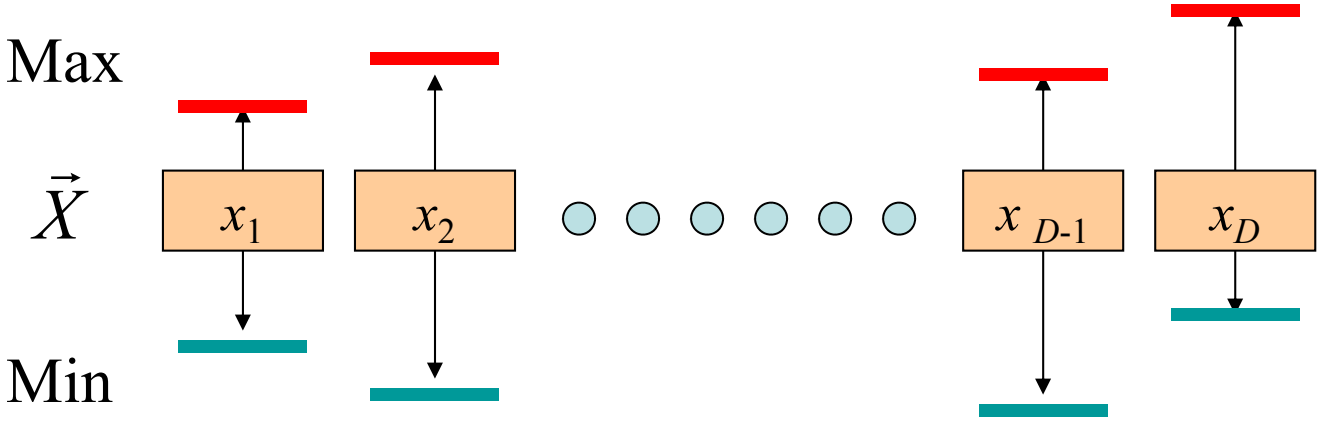
- **A stochastic population-based algorithm for continuous function optimization (Storn and Price, 1995)**
- **Finished 3rd at the First International Contest on Evolutionary Computation, Nagoya, 1996 (*icsi.berkley.edu/~storn*)**
- **Outperformed several variants of GA and PSO over a wide variety of numerical benchmarks over past several years.**
- **Continually exhibited remarkable performance in competitions on different kinds of optimization problems like dynamic, multi-objective, constrained, and multi-modal problems held under IEEE congress on Evolutionary Computation (CEC) conference series.**
- **Very easy to implement in any standard programming language.**
- **Very few control parameters (typically three for a standard DE) and their effects on the performance have been well studied.**
- **Complexity is very low as compared to some of the most competitive continuous optimizers like CMA-ES.**

- DE is an Evolutionary Algorithm
- This Class also includes GA, Evolutionary Programming and Evolutionary Strategies



Basic steps of an Evolutionary Algorithm

Representation

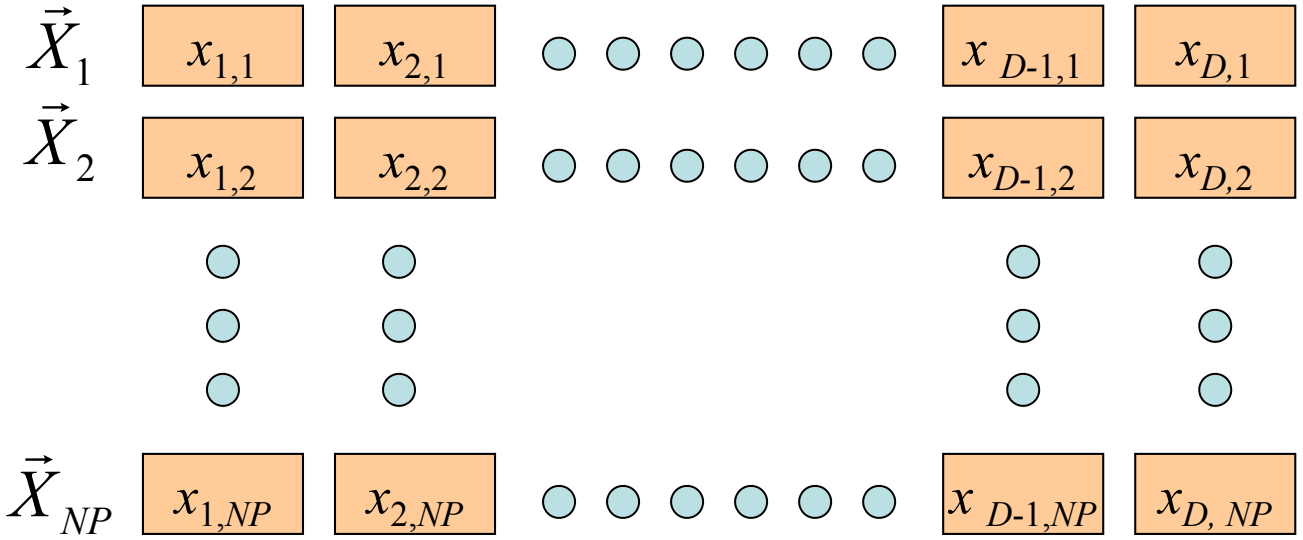


Solutions are represented as vectors of size D with each value taken from some domain.

May wish to constrain the values taken in each domain above and below.

Maintain Population - *NP*

We will maintain a population of size *NP*



The population size *NP*

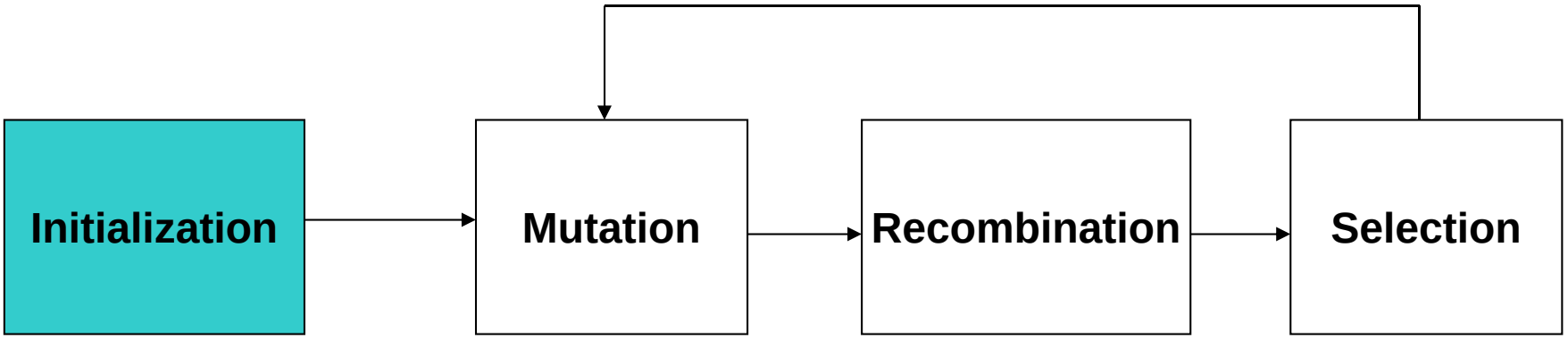
1) The influence of *NP* on the performance of DE is yet to be extensively studied and fully understood.

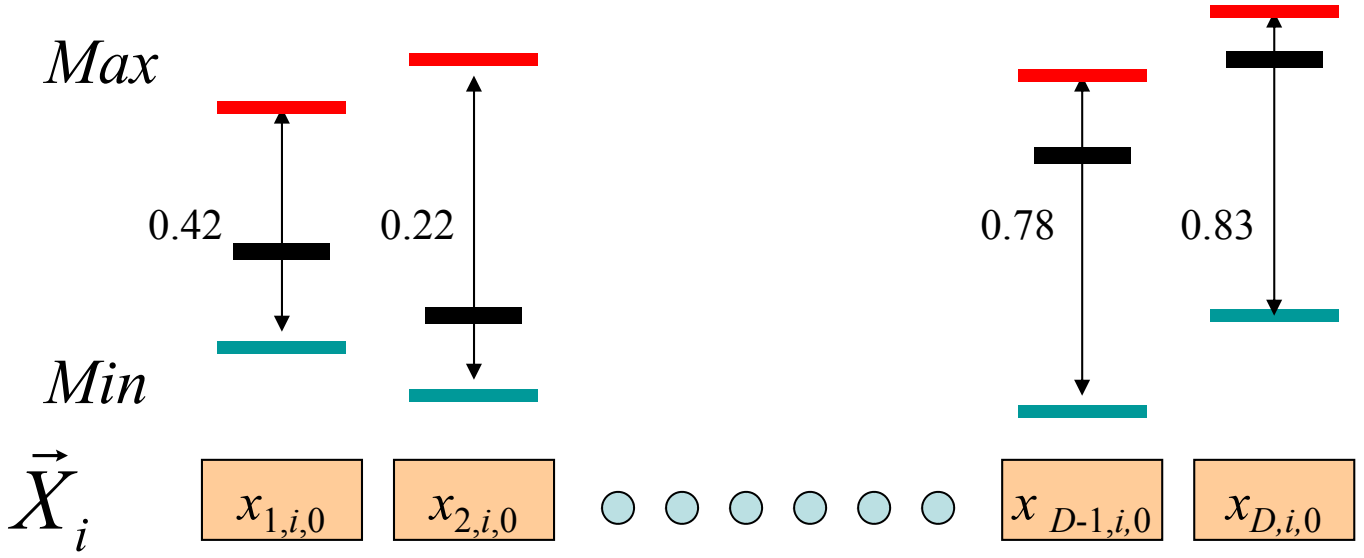
2) Storn and Price have indicated that a reasonable value for *NP* could be chosen between 5*D* and 10*D* (*D* being the dimensionality of the problem).

3) Brest and Maučec presented a method for gradually reducing population size of DE. The method improves the efficiency and robustness of the algorithm and can be applied to any variant of DE.

4) But, recently, all best performing DE variants used populations ~50-100 for dimensions from 50D to 1000D for the following scalability Special Issue:

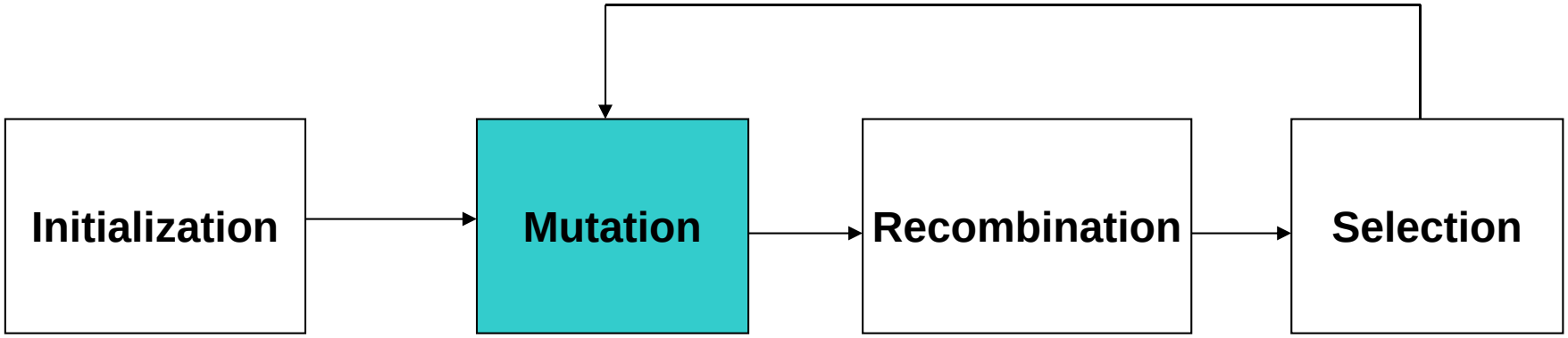
F. Herrera M. Lozano D. Molina, "Test Suite for the Special Issue of Soft Computing on Scalability of Evolutionary Algorithms and other Metaheuristics for Large Scale Continuous Optimization Problems". Available: http://sci2s.ugr.es/eamhco/CFP.php.





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

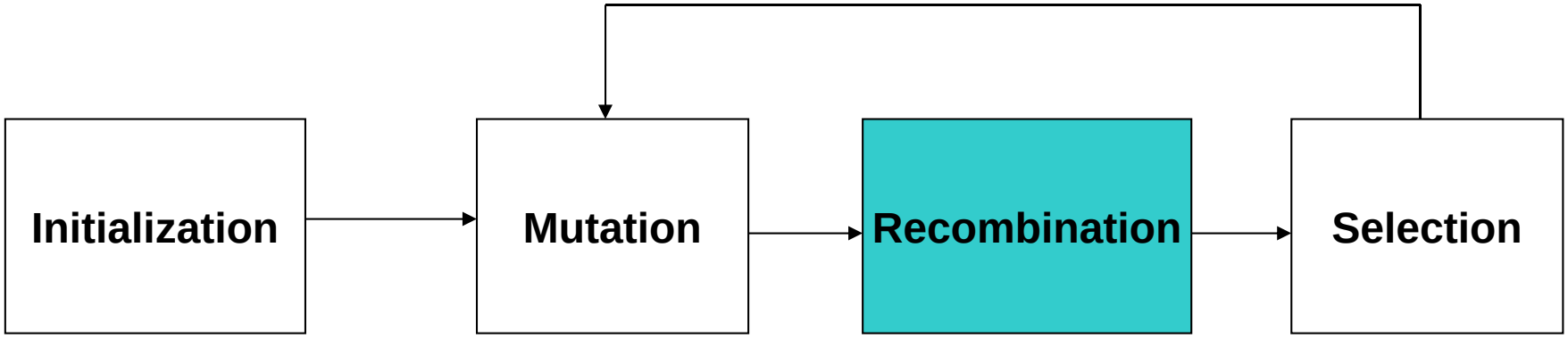
Different $rand_{i,j}[0,1]$ values are instantiated for each i and j.



- For each vector select three other parameter vectors randomly.
- Add the weighted difference of two of the parameter vectors to the third to form a donor vector (most commonly seen form of DE-mutation):

$$\vec{V}_{i,G} = \vec{X}_{r_1^i,G} + F \cdot (\vec{X}_{r_2^i,G} - \vec{X}_{r_3^i,G}).$$

- The scaling factor *F* is a constant from (0, 2)
- Self-referential Mutation



Binomial (Uniform) Crossover:

Components of the donor vector enter into the trial offspring vector in the following way:

Let j_{rand} be a randomly chosen integer between 1,...,D.

$$u_{j,i,G} = \begin{cases} v_{j,i,G}, & \text{if } (rand_{i,j}[0,1) \le Cr \text{ or } j = j_{rand}) \\ x_{j,i,G}, & \text{otherwise,} \end{cases}$$

Exponential (two-point modulo) Crossover:

First choose integers *n* (as starting point) and *L* (number of components the donor actually contributes to the offspring) from the interval [1,*D*]

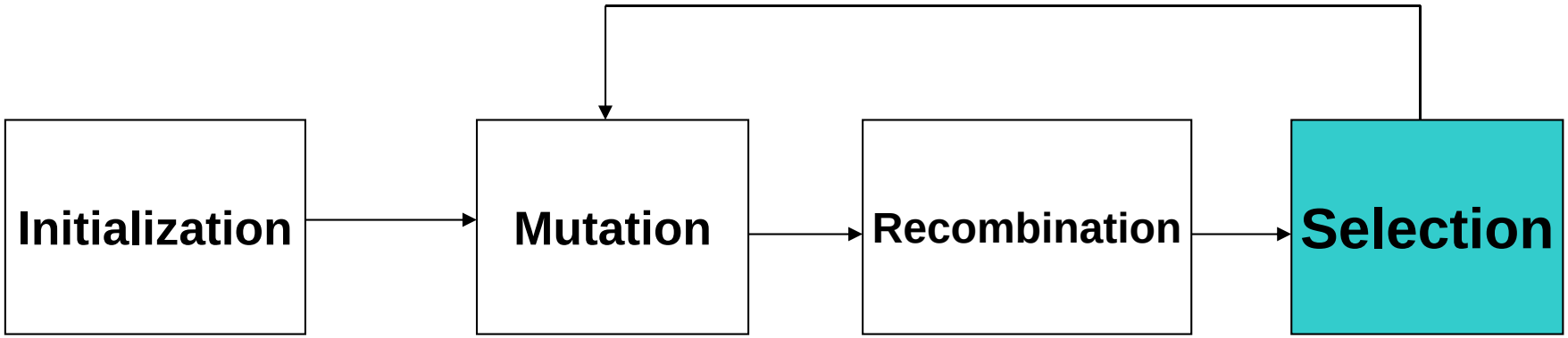
$$u_{j,i,G} = \begin{cases} v_{j,i,G}, & \text{for } j = \langle n \rangle_D, \langle n+1 \rangle_D, \ldots, \langle n+L-1 \rangle_D \\ x_{j,i,G}, & \text{for all other } j \in [1, D], \end{cases}$$

where the angular brackets $\langle \ \rangle_D$ denote a modulo function with modulus *D*.

Pseudo-code for choosing *L:*

```
L = 0;
DO
{
    L = L+1;
} WHILE (((rand[0,1) ≤ Cr) AND (L < D)) ;
```

Exploits linkages among neighboring decision variables. If benchmarks have this feature, it performs well. Similarly, for real-world problems with neighboring linkages.



- **"Survival of the fitter" principle in selection: The trial offspring vector is compared with the target (parent) vector and the one with a better fitness is admitted to the next generation population.**

$$\vec{X}_{i,G+1} = \vec{U}_{i,G}, \text{ if } f(\vec{U}_{i,G}) \le f(\vec{X}_{i,G})$$

$$= \vec{X}_{i,G}, \text{ if } f(\vec{U}_{i,G}) > f(\vec{X}_{i,G})$$

- **Importance of parent-mutant crossover & parent-offspring competition-based selection**

Five most frequently used DE mutation schemes

"DE/rand/1": $\vec{V}_i(t) = \vec{X}_{r_1^i}(t) + F \cdot (\vec{X}_{r_2^i}(t) - \vec{X}_{r_3^i}(t)).$

"DE/best/1": $\vec{V}_i(t) = \vec{X}_{best}(t) + F.(\vec{X}_{r_1^i}(t) - \vec{X}_{r_2^i}(t)).$

"DE/target-to-best/1": $\vec{V}_i(t) = \vec{X}_i(t) + F.(\vec{X}_{best}(t) - \vec{X}_i(t)) + F.(\vec{X}_{r_1^i}(t) - \vec{X}_{r_2^i}(t)),$

"DE/best/2": $\vec{V}_i(t) = \vec{X}_{best}(t) + F.(\vec{X}_{r_1^i}(t) - \vec{X}_{r_2^i}(t)) + F.(\vec{X}_{r_3^i}(t) - \vec{X}_{r_4^i}(t)).$

"DE/rand/2": $\vec{V}_i(t) = \vec{X}_{r_1^i}(t) + F_1.(\vec{X}_{r_2^i}(t) - \vec{X}_{r_3^i}(t)) + F_2.(\vec{X}_{r_4^i}(t) - \vec{X}_{r_5^i}(t)).$

The general convention used for naming the various mutation strategies is DE/x/y/z, where DE stands for Differential Evolution, x represents a string denoting the vector to be perturbed, y is the number of difference vectors considered for perturbation of x, and z stands for the type of crossover being used (exp: exponential; bin: binomial)

Overview

I. Introduction to Real Variable Optimization & DE

II. Future of Real Parameter Optimization

III. Single Objective Optimization

IV. Multimodal Optimization

DE with Arithmetic Crossover

1) In *continuous* or *arithmetic* recombination, the individual components of the trial vector are expressed as a linear combination of the components from mutant/donor vector and the target vector.

General form: $\vec{W}_{i,G} = \vec{X}_{r_1,G} + k_i.(\vec{X}_{r_1,G} - \vec{X}_{r_2,G})$

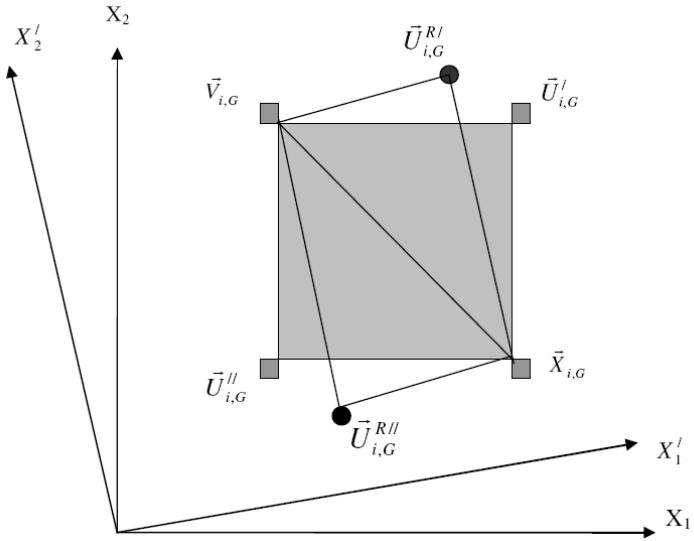
2) 'DE/current-to-rand/1' replaces the binomial crossover operator with the rotationally invariant arithmetic line recombination operator to generate the trial vector by a linear arithmetic recombination of target and donor vectors:

$$\vec{U}_{i,G} = \vec{X}_{i,G} + k_i.(\vec{V}_{i,G} - \vec{X}_{i,G})$$

which further simplifies to: $\vec{U}_{i,G} = \vec{X}_{i,G} + k_i.(\vec{X}_{r_1,G} - \vec{X}_{i,G}) + F'.(\vec{X}_{r_2,G} - \vec{X}_{r_3,G})$

Change of the trial vectors generated through the discrete and random intermediate recombination due to rotation of the coordinate system.

$\vec{U}^{R/}_{i,G}$ and $\vec{U}^{R//}_{i,G}$ indicate the new trial vectors due to discrete recombination in rotated coordinate system.



K. Price, R. Storn, and J. Lampinen, *Differential Evolution - A Practical Approach to Global Optimization*, Springer, Berlin, 2005.

Importance of Population Topologies

- In population based algorithms, population members exchange information between them.
- Single population topology permits all members to exchange information among themselves – the most commonly used.
- Other population topologies have restrictions on information exchange between members – the oldest is island model
- Restrictions on information exchange can slow down the propagation of information from the best member in the population to other members (i.e. single objective global optimization)
- Hence, this approach
  - slows down movement of other members towards the best member(s)
  - Enhances the exploration of the search space
  - Beneficial when solving multi-modal problems

As global version of the PSO converges fast, many topologies were Introduced to slow down PSO …

PSO with Neighborhood Operator

Presumed to be the oldest paper to consider distance based neighborhoods for real-parameter optimization.

Lbest is selected from the members that are closer (w.r.t. Euclidean distance) to the member being updated.

Initially only a few members are within the neighborhood (small distance threshold) and finally all members are in the n'hood.

Island model and other static/dynamic neighborhoods did not make use of Euclidean distances, instead just the indexes of population members.

Our recent works are extensively making use of distance based neighborhoods to solve many classes of problems.

P. N. Suganthan, "Particle swarm optimizer with neighborhood operator," in *Proc. Congr. Evol. Comput.*, Washington, DC, pp.1958–1962, **1999**.

Two Subpopulations with Heterogeneous Ensembles & Topologies

- Proposed for balancing exploration and exploitation capabilities
- Population is divided into exploration / exploitation sub-poplns
  - Exploration Subpopulation group uses exploration oriented ensemble of parameters and operators
  - Exploitation Subpopulation group uses exploitation oriented ensemble of parameters and operators.

- Topology allows information exchange only from explorative subpopulation to exploitation sub-population. Hence, diversity of exploration popln not affected even if exploitation popln converges.

- The need for memetic algorithms in real parameter optimization: Memetic algorithms were developed because we were not able to have an EA or SI to be able to perform both exploitation and exploration simultaneously. This 2-popln topology allows with heterogeneous information exchange.

Two Subpopulations with Heterogeneous Ensembles & Topologies

- Sa.EPSDE realization (for single objective Global):

N. Lynn, R Mallipeddi, P. N. Suganthan, “Differential Evolution with Two Subpopulations," LNCS 8947, SEMCCO 2014.

- 2 Subpopulations CLPSO (for single objective Global)

N. Lynn, P. N. Suganthan, “Comprehensive Learning Particle Swarm Optimization with Heterogeneous Population Topologies for Enhanced Exploration and Exploitation,” *Swarm and Evolutionary Computation*, 2015.

- Neighborhood-Based Niching-DE: Distance based neighborhood forms local topologies while within each n’hood, we employ exploration-exploitation ensemble of parameters and operators.

S. Hui, P N Suganthan, “Ensemble and Arithmetic Recombination-Based Speciation Differential Evolution for Multimodal Optimization,” IEEE T. Cybernetics, Online since Mar 2015. 10.1109/TCYB.2015.2394466

Ensemble Methods

- Ensemble methods are commonly used for pattern recognition (PR), forecasting, and prediction, e.g. multiple predictors.

- Not commonly used in Evolutionary algorithms ...

There are two advantages in EA (compared to PR):

1. In PR, we have no idea if a predicted value is correct or not. In EA, we can look at the objective values and make some conclusions.

2. Sharing of function evaluation among ensembles possible.

Adaptations

- Self-adaptation: parameters and operators are evolved by coding them together with solution vector
- Separate adaptation based on performance: operators and parameter values yielding improved solutions are rewarded.
- 2nd approach is more successful and frequently used with population-based numerical optimizers.

Overview

I. Introduction to Real Variable Optimization & DE

II. Future of Real Parameter Optimization

III. Single Objective Optimization

IV. Multimodal Optimization

Self-Adaptive DE (SaDE) (Qin *et al*., 2009)

- Includes both **control parameter adaptation** and **strategy adaptation**

Strategy Adaptation:

Four effective trial vector generation strategies: DE/rand/1/bin, DE/rand-to-best/2/bin, DE/rand/2/bin and DE/current-to-rand/1 are chosen to constitute a strategy candidate pool.

For each target vector in the current population, one trial vector generation strategy is selected from the candidate pool according to the probability learned from its success rate in generating improved solutions (that can survive to the next generation) within a certain number of previous generations, called the *Learning Period* (*LP*).

A. K. Qin, V. L. Huang, and P. N. Suganthan, Differential evolution algorithm with strategy adaptation for global numerical optimization", *IEEE Trans. on Evolutionary Computation,* 13(2):398-417, April, 2009.

SaDE (Contd..)

Control Parameter Adaptation:

1) **_NP_ is left as a user defined parameter.**

2) **A set of _F_ values are randomly sampled from normal distribution _N_(0.5, 0.3) and applied to each target vector in the current population.**

3) **_CR_ obeys a normal distribution with mean value CR_m and standard deviation _Std_ =0.1, denoted by $N(CR_m, Std)$ where CR_m is initialized as 0.5.**

4) **SaDE gradually adjusts the range of _CR_ values for a given problem according to previous _CR_ values that have generated trial vectors successfully entering the next generation.**

JADE (Zhang and Sanderson, 2009)

1) Uses DE/current-to-*p*best strategy as a less greedy generalization of the DE/current-to-best/ strategy. Instead of only adopting the best individual in the DE/current-to-best/1 strategy, the current-to-pbest/1 strategy utilizes the information of other good solutions.

Denoting $\vec{X}^{p}_{best,G}$ as a randomly chosen vector from the top 100*p*% individuals of the current population,

DE/current-to-*p*best/1 without external archive: $\vec{V}_{i,G} = \vec{X}_{i,G} + F_i \cdot (\vec{X}^{p}_{best,G} - \vec{X}_{i,G}) + F_i \cdot (\vec{X}_{r_1^i,G} - \vec{X}_{r_2^i,G})$

2) JADE can optionally make use of an external archive (A), which stores the recently explored inferior solutions. In case of DE/current-to-pbest/1 with archive, $\vec{X}_{i,G}$ $\vec{X}^{p}_{best,G}$, and $\vec{X}_{r_1^i,G}$ are selected from the current population P, but $\vec{X}_{r_2^i,G}$ is selected from $P \cup A$

J. Zhang, and A. C. Sanderson, "JADE: Adaptive differential evolution with optional external archive", *IEEE Transactions on Evolutionary Computation*, Vol. 13, Issue 5, Page(s): 945-958, Oct. 2009.

JADE (Contd..)

3) JADE adapts the control parameters of DE in the following manner:

A) *Cr* for each individual and at each generation is randomly generated from a normal distribution $N(\mu_{Cr},0.1)$ and then truncated to [0, 1].

The mean of normal distribution is updated as: $\mu_{Cr} = (1-c).\mu_{Cr} + c.mean_A(S_{Cr})$

where S_{Cr} be the set of all successful crossover probabilities Cr_i s at generation *G*

B) Similarly for each individual and at each generation F_i is randomly generated from a Cauchy distribution $C(\mu_F,0.1)$ with location parameter μ_F and scale parameter 0.1.

F_i is truncated if F_i > 1 or regenerated if F_i <= 0

The location parameter of the Cauchy distribution is updated as: $\mu_F = (1-c).\mu_F + c.mean_L(S_F)$

where S_F is the set of all successful scale factors at generation G and $mean_L$ is the Lehmer mean:

$$mean_L(S_F) = \frac{\sum_{F \in S_F} F^2}{\sum_{F \in S_F} F}$$

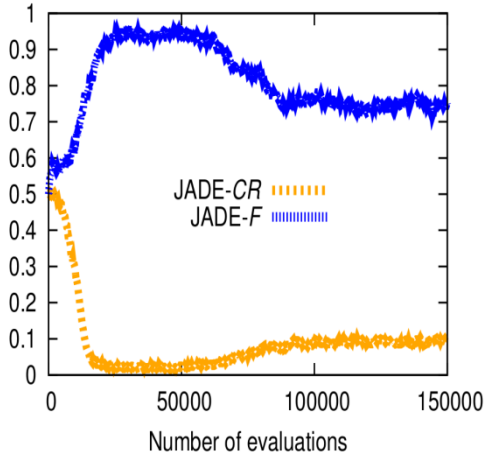
JADE usually performs best with 1/*c* chosen from [5, 20] and *p* from [5%, 20%]

Success-History based Adaptive DE (SHADE)

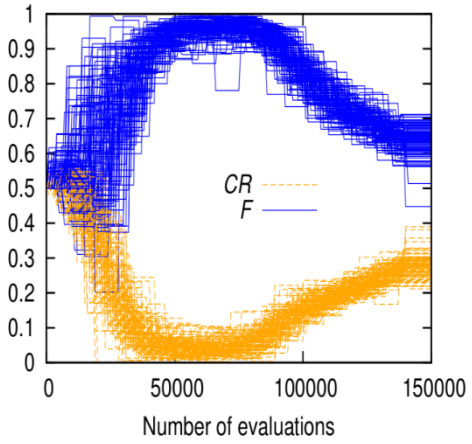
- An improved version of JADE
- Uses a success-history based adaptation
  - Based on a historical memory of successful parameter settings that were previously found during the run
  - A historical memory M_{CR} , M_F are used, instead of adaptive parameter u_{cr} , u_F
  - This improves the robustness of JADE

Fig. Their adaptation behaviors on Rastrigin (30 dimensions)

JADE uses a single pair u_{cr}, u_F

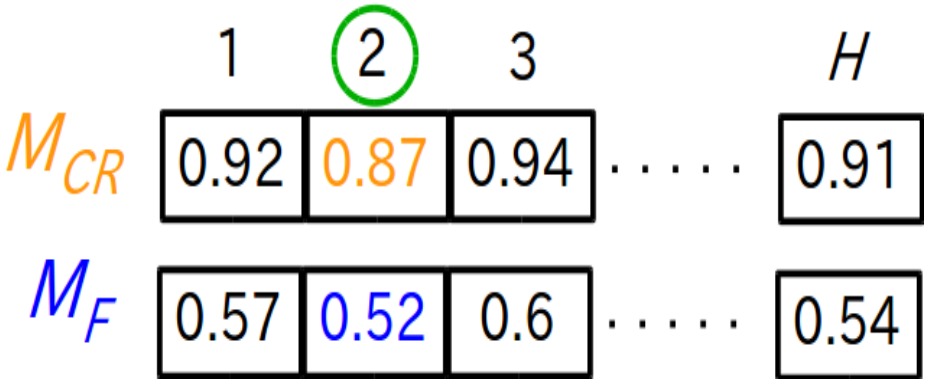


SHADE maintains a diverse set of parameters
in a historical memory M_{CR} , M_F



SHADE

- The weighted Lehmer mean (in CEC'14 ver.) values of S_{CR} and S_F , which are successful parameters for each generation, are stored in a historical memory M_{CR} and M_F

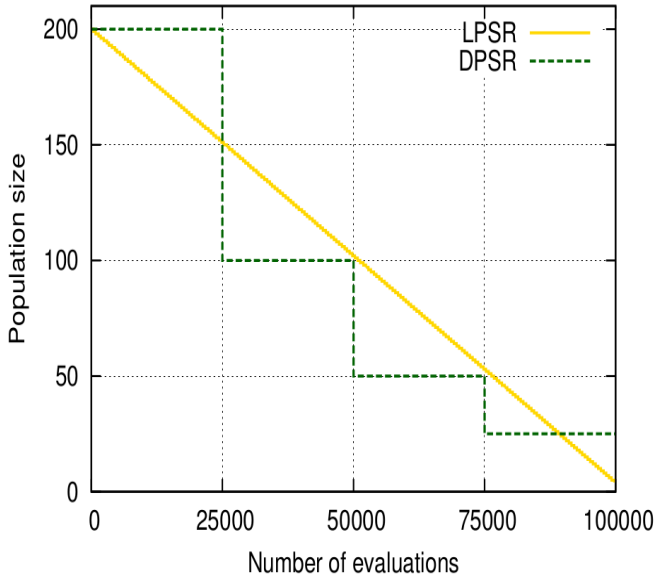


- CR_i and F_i are generated by selecting an index r_i randomly from [1, memory size H]
- Example: if selected index $r_i = 2$
  - CR_i = NormalRand(0.87, 0.1)
  - F_i = CauchyRand(0.52, 0.1)

L-SHADE: SHADE with Linear Population Size Reduction

- Deterministic Population Size Reduction (DPSR) [Brest 08]
  - reduces the population by half at predetermined intervals
  - The frequency of the population reduction has to be tuned to match the initial population size as well as the dimensionality of the problem…
- Simple Variable Population Sizing (SVPS) [Laredo 09]
  - is a more general framework in which the shape of the population size reduction schedule is determined according to two control parameters
  - Due to its general versatility, tuning the two control parameters is very hard…
- Linear Population Size Reduction (LPSR) [Tanabe CEC 2014]
  - is a special case of SVPS which reduces the population linearly, and requires only initial population sizes
  - L-SHADE is an extended SHADE with LPSR

Fig. Comparison of population resizing schedule between LPSR and DPSR (# of reduction = 4)



L-SHADE's C++ and Matlab/Octave code can be downloaded from Ryoji Tanabe's site (https://sites.google.com/site/tanaberyoji/)

Ensemble of Parameters and Mutation and Crossover Strategies in DE (EPSDE)

- Motivation
  - Empirical guidelines
  - Adaptation/self-adaptation (different variants)
  - Optimization problems (Ex: uni-modal & multimodal)
  - Fixed single mutation strategy & parameters – may not be the best always
- Implementation
  - Contains a pool of mutation strategies & parameter values
  - Compete to produce successful offspring population.
  - Candidate pools must be restrictive to avoid unfavorable influences
  - The pools should be diverse

R. Mallipeddi, P. N. Suganthan, Q. K. Pan and M. F. Tasgetiren, "Differential Evolution Algorithm with ensemble of parameters and mutation strategies," *Applied Soft Computing*, 11(2):1679–1696, March 2011.

Adaptive EPSDE

- Selection of pool of mutation strategies
  1. strategies without crossover (DE/current-to-rand/1/bin)
  2. strategies with crossover
     1. individuals of mutant vector randomly selected (DE/rand/1/bin)
     2. rely on the best found so far (DE/best/2/bin)
- Selection of pool of parameters

$F = \{0.4, 0.5, 0.6, 0.7, 0.8, 0.9\}$ $Cr = \{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9\}$

- Initial population randomly assigned with a mutation strategy & parameters
- Success rate of each parameter or operator is recorded and future usage is proportional to each one's success rate over a few recent past generations.

Overview

I. Introduction to Real Variable Optimization & DE

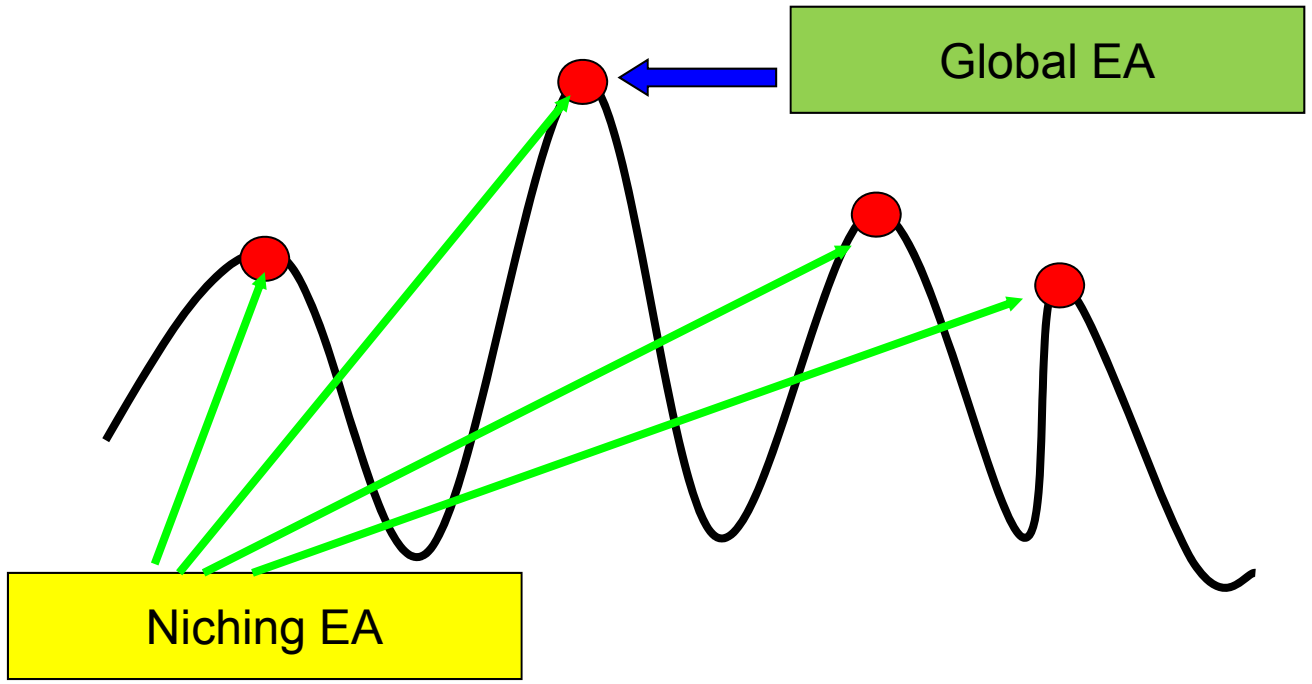
II. Future of Real Parameter Optimization

III. Single Objective Optimization

IV. Multimodal Optimization

Niching and Multimodal Optimization with DE

- Traditional evolutionary algorithms with elitist selection are suitable to locate a single optimum of **functions**.
- Real problem may require the identification of optima along with several optima.
- For this purpose, **niching methods** extend the simple evolutionary algorithms by ***promoting the formation of subpopulations in the neighborhood of the local optimal solutions.***



Multi-modal Optimization Methods

- Some existing Niching Techniques
  - Sharing
  - Clearing
  - Crowding
  - Restricted Tournament Selection
  - Clustering
  - Species Based
  - Adaptive Neighborhood Topology based DE

B-Y Qu, P N Suganthan, J J Liang, "Differential Evolution with Neighborhood Mutation for Multimodal Optimization," *IEEE Trans on Evolutionary Computation,* Doi: 10.1109/TEVC.2011.2161873, 2012.

Adaptive Neighborhood Mutation Based DE

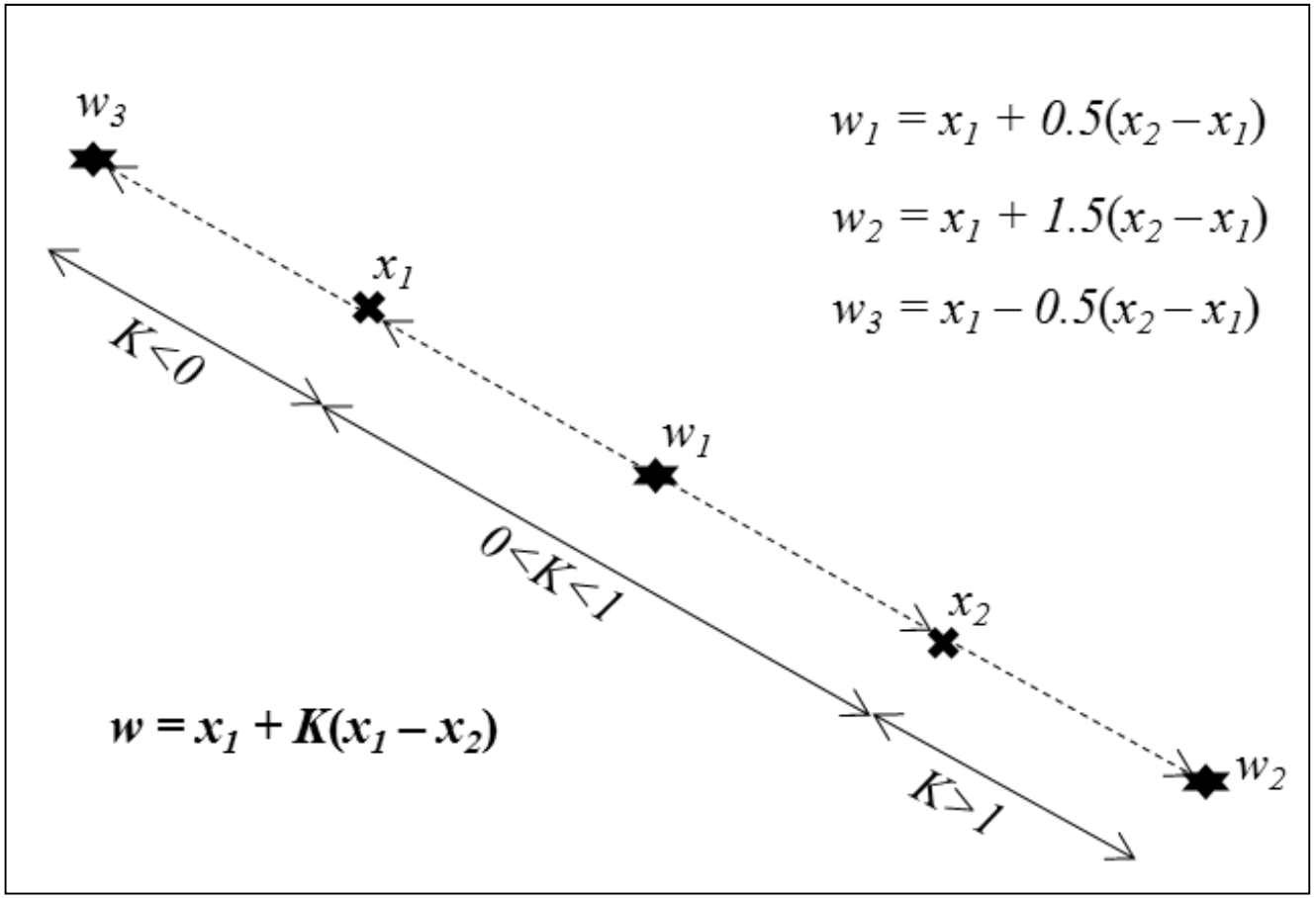
STEPS OF GENERATING OFFSPRING USING NEIGHBORHOOD MUTATION

Input	A population of solutions of current generation (current parents)
Step 1	For $i = 1$ to NP (population size)
	1.1 Calculate the Euclidean distances between individual i and other members in the population.
	1.2 Select m smallest Euclidean distance members to individual i and form a subpopulation ($subpop$) using these m members.
	1.3 Produce an offspring u_i using DE equations within $subpop_i$, i.e., pick r_1, r_2, r_3 from the subpopulation.
	2.3 Reset offspring u_i within the bounds if any of the dimensions exceed the bounds.
	2.4 Evaluate offspring u_i using the fitness function.
	Endfor
Step 2	Selection NP fitter solutions for next generation according to the strategies of different niching algorithm.
Output	A population of solutions for next generation

Compared with about 15 other algorithms on about 27 benchmark problems including IEEE TEC articles published in 2010-2012 period.

B-Y Qu, P N Suganthan, J J Liang, "Differential Evolution with Neighborhood Mutation for Multimodal Optimization," *IEEE Trans on Evolutionary Computation,* Doi: 10.1109/TEVC.2011.2161873, Oct. 2012.

Arithmetic Recombination

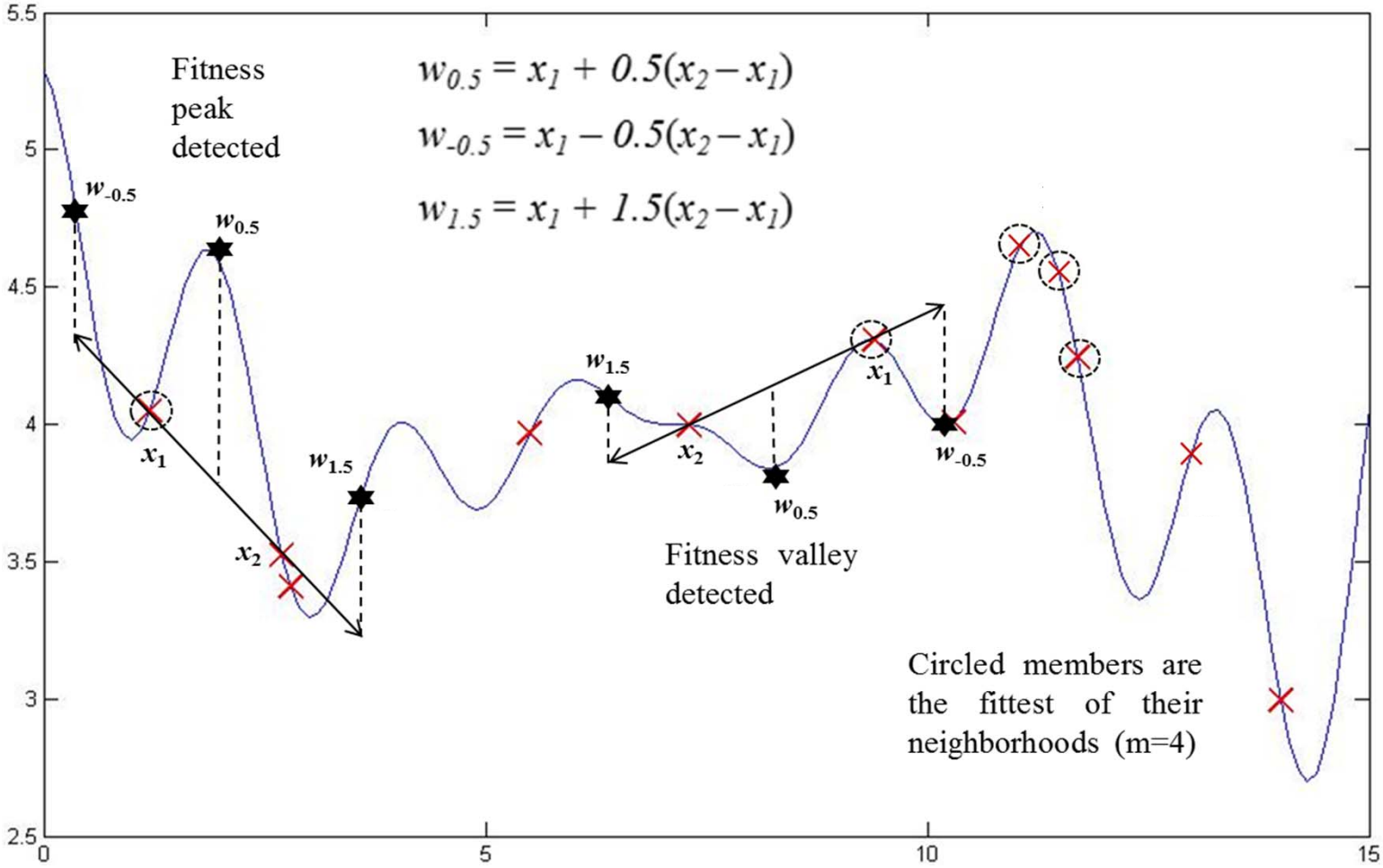


Search region covered by line Arithmetic Recombination for *K= [-0.5, 0.5, 1.5]*

Neighborhood Arithmetic Recombination-based Speciation DE (based on DoI: . 10.1109/TCYB.2015.2394466)

- Classical niching and clustering methods highly sensitive to parameter settings and initial population distribution in addition to niche size
  - e.g Speciation radius $R_{species}$, Fitness-sharing radius $R_{sharing,}$ Crowding factor, CF
- Difficult to separate the initial population into niches in uneven and rugged regions,
  - i.e when niching radius contains more than one peak.
- A guaranteed way to identify separate niches: detecting fitness valleys and peaks
- Arithmetic Recombination interpolates and extrapolates between neighbors from niche centers (local fittest member)
- Self-adaptive generalization across different fitness terrains
- Reduction of multiple niching parameters to only neighborhood popln size, *m*

Interpolating and Extrapolating members using Arithmetic Recombination



X – popln member ; Star: solutions generated Arith. Recomb.

Neighborhood Arithmetic Recombination-based Speciation DE

Step 1: Initialize a randomly distributed population of size N within the range of $[\mathbf{X}^{Lower}, \mathbf{X}^{Upper}]$ in D dimensions

Step 2: Compute Euclidean distance for all members

$$\mathbf{dist}_{ij}=\sqrt{\sum(x_{i,j})^2}, j=1, \ldots D, i=1,2\ldots N$$

Step 3: Sort all individuals in descending order of their fitness values. This is the speciation pool.

Step 4: Set niches number S=1

WHILE sorted population is not empty, execute steps 4.1 to 4.6:

1. Identify the fittest member (*lbest*) from speciation pool and remove it as the specie seed for specie number *S*.
2. Extract *m* nearest neighbors to the *lbest* from speciation pool for arithmetic recombination.
3. Detect the presence of fitness valleys and better fitness between *lbest* and the neighbors in the specie using $0<K<1$.
    a) Reject neighbors that are separated by fitness valleys from the *lbest* of the current specie. Rejected neighbors are returned to the speciation pool.
    b) Check for existence of better fitness between *lbest* and neighbors and store these as new members of the same species
4. Explore regions beyond *lbest* and the remaining neighbors using $K>1$ and $K<0$. Solutions with $K>1$ are included into the common speciation pool while $K<0$ are included in the current specie *S*. These solutions will be subjected to steps 4.3 & 4.4 in the next iteration.
5. Check the number of members within specie. If the population size exceeds that of specified, remove the excess members in order of weakest fitness. On the other hand if insufficient members are in the same species, randomly initialize members within Euclidean distance of the furthest specie member (or nearest non-specie neighbor) from the specie seed.
6. Increment *S* until all members are classified into species.

END WHILE

Step 5: Perform Ensemble-DE operations within every specie separately.

Step 6: Repeat Steps 2 to 5 until a termination criterion.

S. Hui, P N Suganthan, "Ensemble and Arithmetic Recombination-Based Speciation Differential Evolution for Multimodal Optimization," IEEE T. Cybernetics, Online. 10.1109/TCYB.2015.2394466

Ensemble Parameters of EARSDE

- Neighborhood size, $m = 6$
- Scaling factor $F_i \in [0.3, 0.5, 0.9]$.
- Crossover probability $C_i \in [0.1, 0.5]$.
- Binomial Crossover

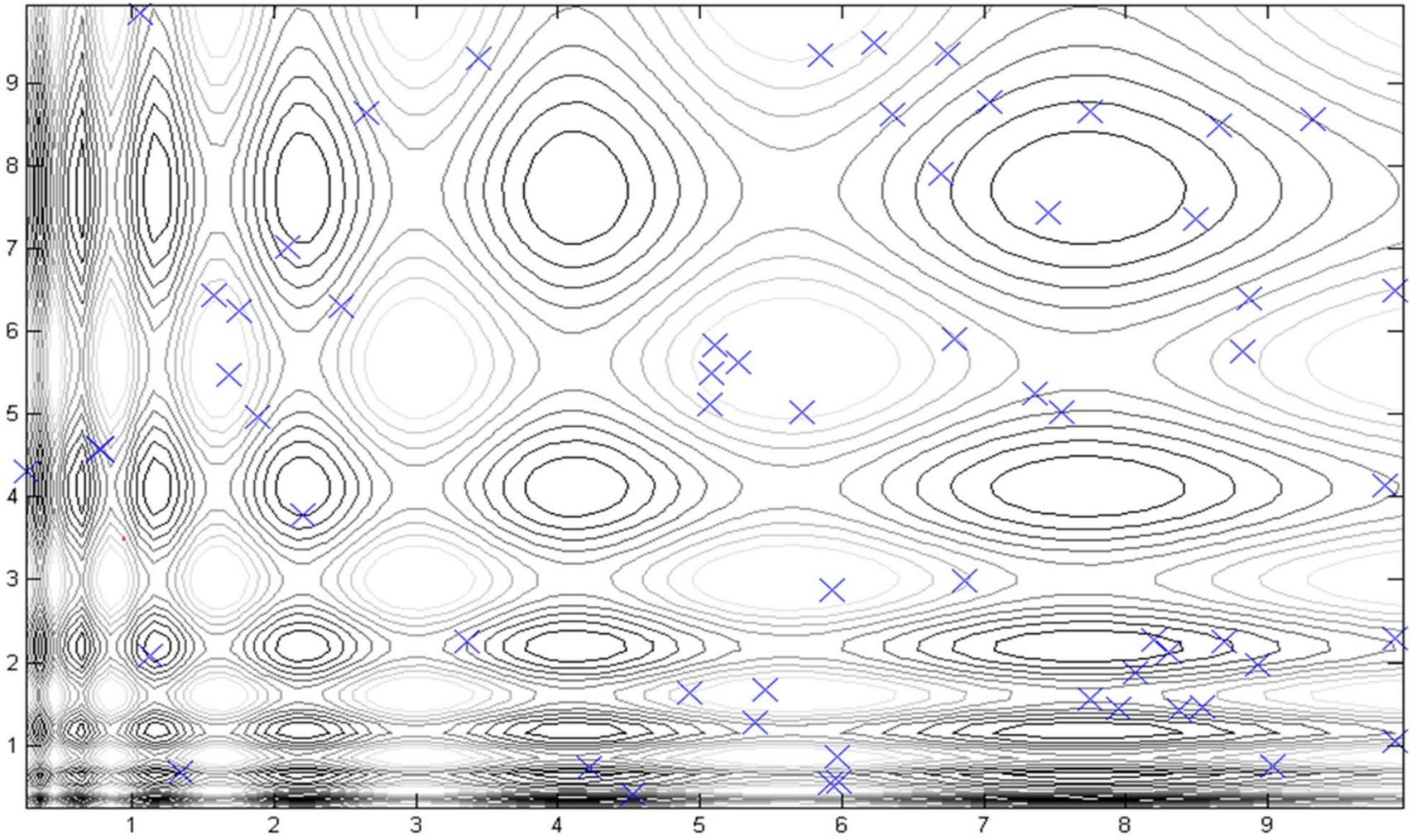
$$U_{i,j} = \begin{cases} v_{i,j} \; if \; (rand_j[0,1) \le Cr) \; or \; (j = j_{rand}) \\ x_{i,j} \qquad\qquad otherwise \end{cases}$$

- Mutation Strategies
  1. DE/rand/1

     $\boldsymbol{v_i^G} = \boldsymbol{x}_{rand1,i}{}^G + F(\boldsymbol{x}_{rand2,i}{}^G - \boldsymbol{x}_{rand3,i}{}^G)$
  2. DE/ best/1

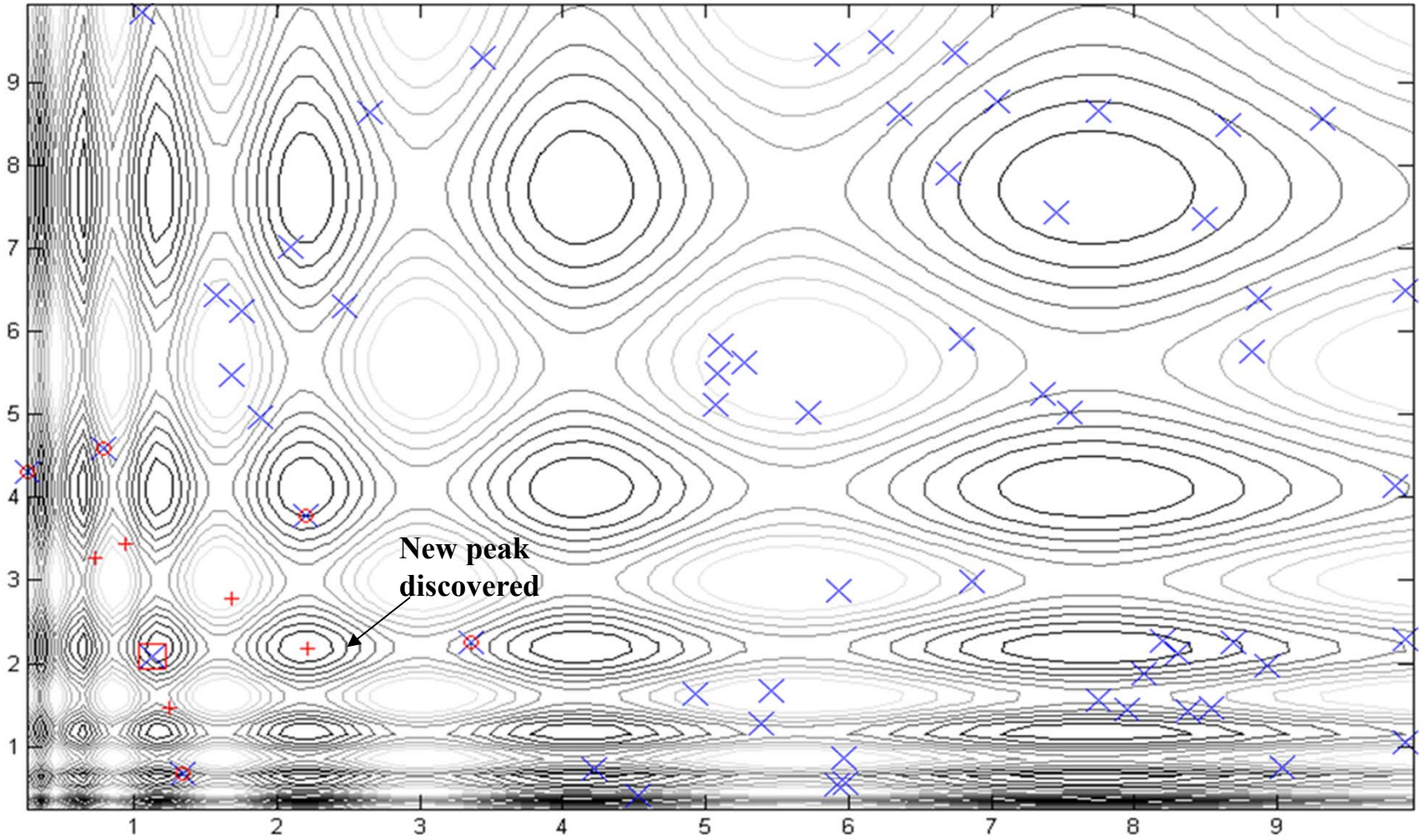
     $\boldsymbol{v}_i{}^G = \boldsymbol{x}_{best}{}^G + F(\boldsymbol{x}_{rand1,i}{}^G - \boldsymbol{x}_{rand2,i}{}^G)$
- Arithmetic Recombination scaling factor, $K = [-0.5, 0.5, 1.5]$ with additive variable, $\pm\Delta$
- $\Delta \in [0, 0.1]$

Speciation process in EARSDE in 2D Vincent problem



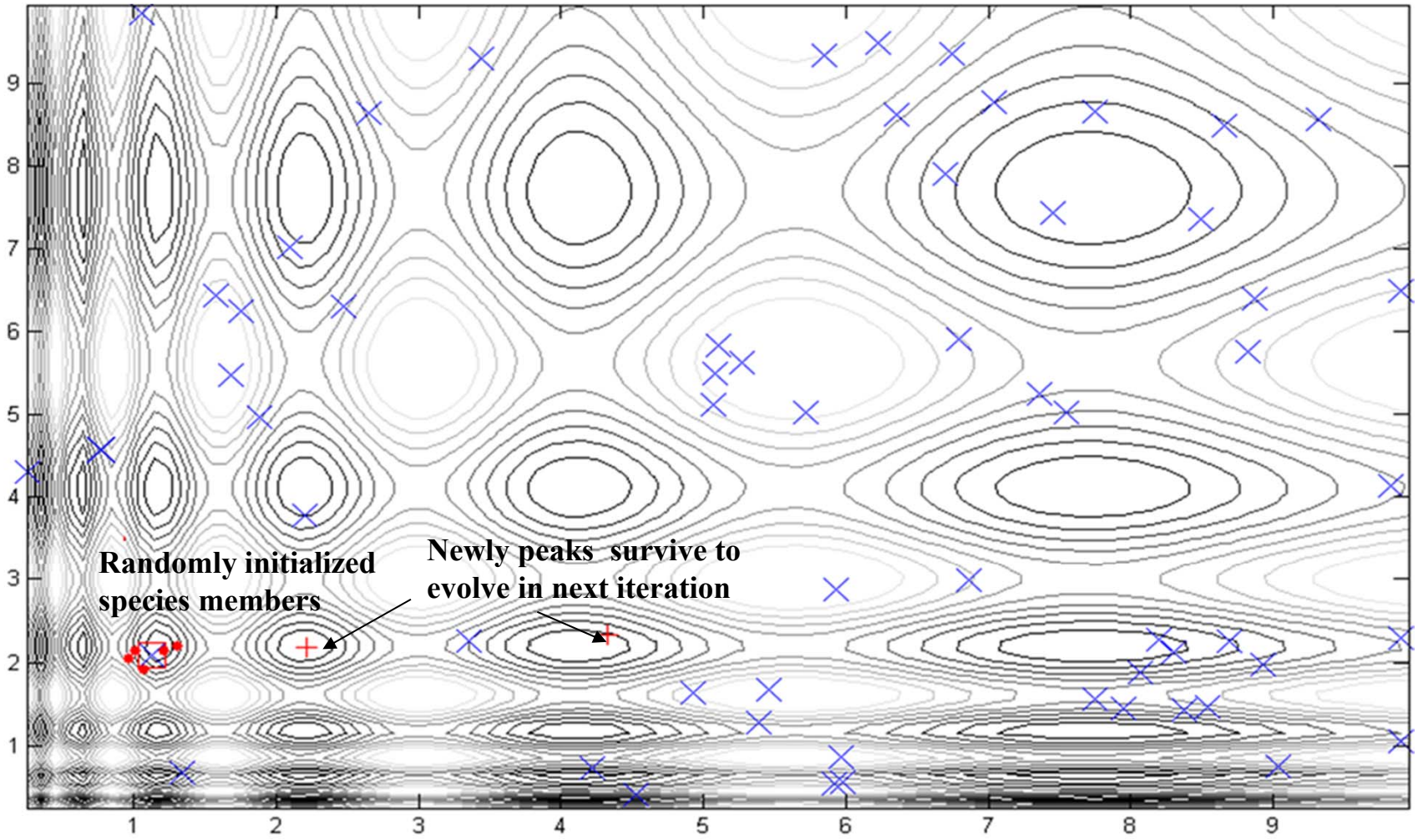
- Initial population - 'X', initial pop-size = 60, neighborhood size = 6
- Fittest regions are represented by the darkest contours.
- Fittest member is first selected as the species seed for AR operations

Speciation process in EARSDE in 2D Vincent problem



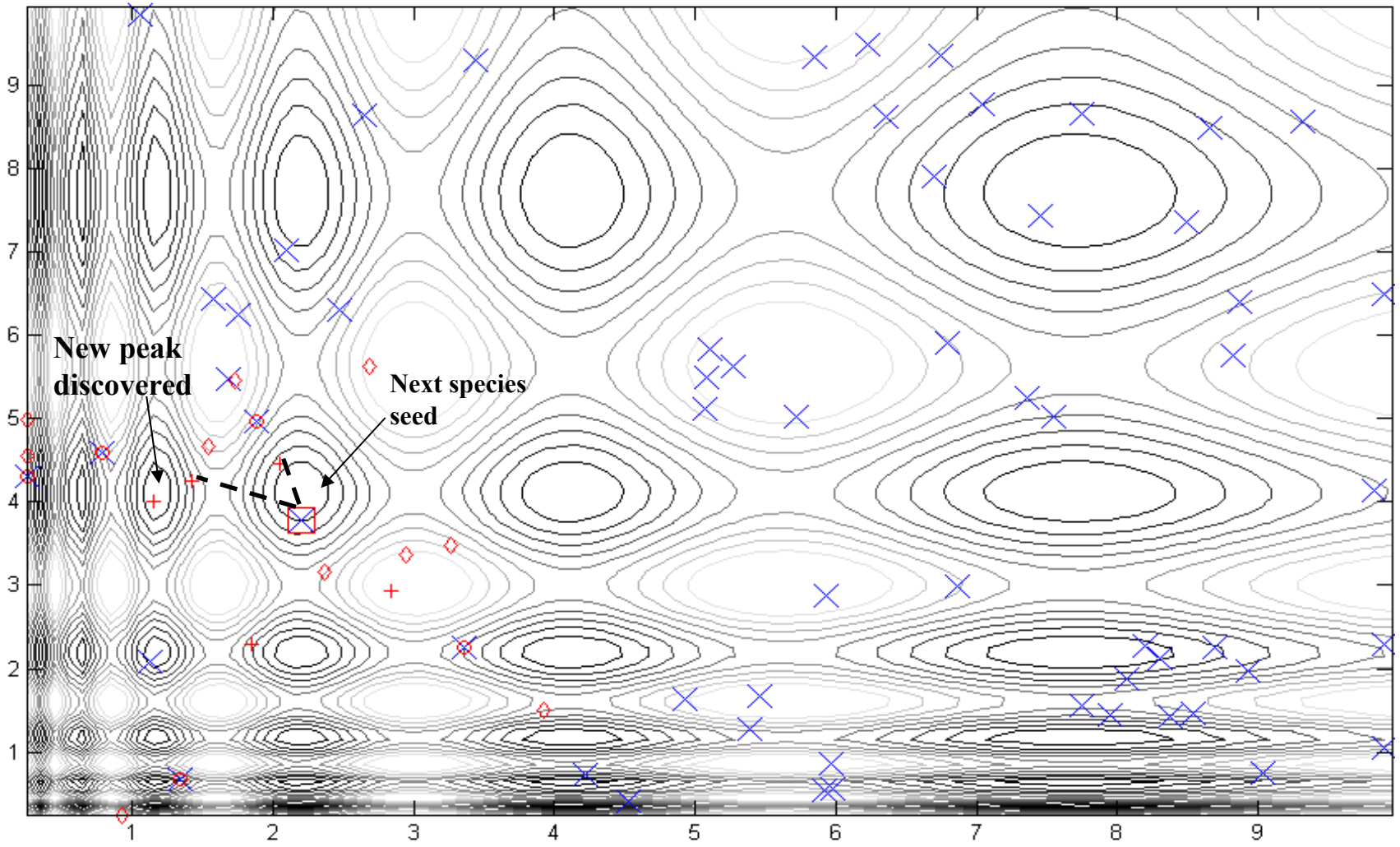
- species seed is highlighted with a red square '□'
- 5 nearest neighbors are highlighted with red circle 'o'
- *AR* (*K*=0.5) applied to check for any fitness valleys.
- Midpoints represented with red plus signs '+'

Speciation process in EARSDE in 2D Vincent problem



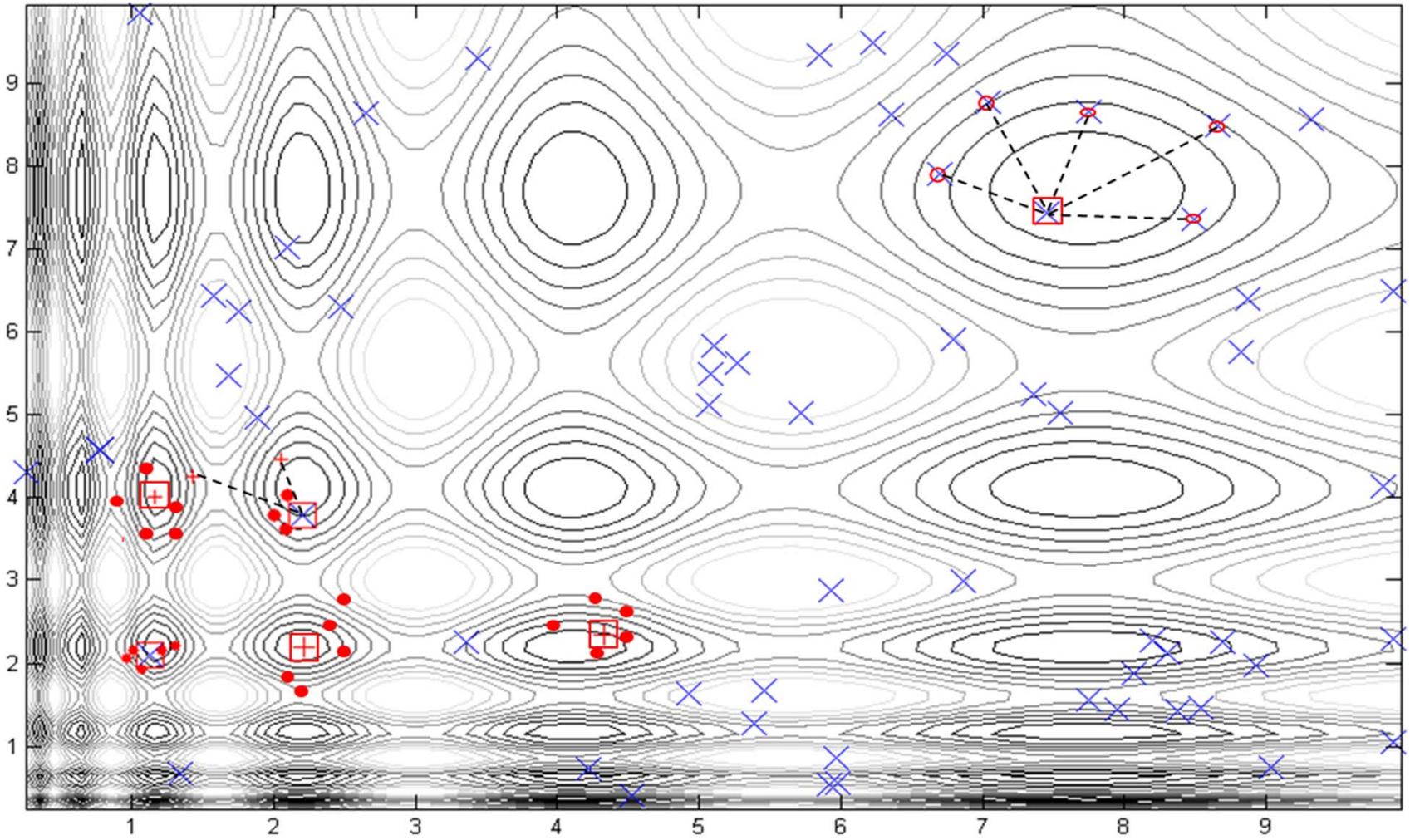
- Existing neighbors of the species seed could not be grouped into the same species due to fitness valleys or doubled peaks
- Random initialization executed around original species seed within 0.5 of the distance to the nearest neighbor separated by fitness valley.
- New random members are represented by the red circles '●'

Speciation process in EARSDE in 2D Vincent problem



- A neighbor rejected by the previous species now selected for speciation. Repeat process
- Dotted black lines to indicate association to species

Speciation process in EARSDE in 2D Vincent problem



- New peaks (highlighted by a red square '□' with red plus signs '+') would only enter speciation process in the next iteration
- Random members populated around new peaks.

Multi-modal CEC 2010Benchmark Problems

Test Function Set 1		Parameter setup for peer algo.	
Function Name	Peaks/Dim/ ε_{peak}/MaxFEs	Radius (r)	Pop. Size (NP)
E1-F1. Two-peak trap	1/1/ 0.05/10000	0.5	50
E1-F2. Central two-peak trap	1/1/ 0.05/10000	0.5	50
E1-F3. Five-uneven-peak trap	2/1/ 0.05/10000	0.5	50
E1-F4. Equal maxima	5/1/ 0.000001/10000	0.01	50
E1-F5. Decreasing maxima	1/1/ 0.000001/10000	0.01	50
E1-F6. Uneven maxima	5/1/ 0.000001/10000	0.01	50
E1-F7. Uneven decreasing maxima	1/1/ 0.000001/10000	0.01	50
E1-F8. Himmelblau's function	4/2/ 0.00005/10000	0.5	50
E1-F9. Six-hump camel back	2/2/ 0.000001/10000	0.5	50
E1-F10. Shekel's foxholes	1/2/ 0.00001/10000	0.5	50
E1-F11. 2-D Inverted Shubert	18/2/ 0.05/100000	0.5	250
E1-F12. 1-D Inverted Vincent	6/1/ 0.0001/20000	0.2	100
E1-F13. 2-D Inverted Vincent	36/2/ 0.001/200000	0.2	500
E1-F14. 3-D Inverted Vincent	216/3/ 0.001/400000	0.2	1000

Test Function Set 2 (Max FES = 300000, ε_{peak}=0.5)	
Function Name	Peaks/Dim/radius/Pop
E1-F15: Composite Function 1	8/10/1/600
E1-F16: Composite Function 2	6/10/1/600
E1-F17: Composite Function 3	6/10/1/600
E1-F18: Composite Function 4	6/10/1/600
E1-F19: Composite Function 5	6/10/1/600
E1-F20: Composite Function 6	6/10/1/600
E1-F21: Composite Function 7	6/10/1/600
E1-F22: Composite Function 8	6/10/1/600
E1-F23: Composite Function 9	6/10/1/600
E1-F24: Composite Function 10	6/10/1/600
E1-F25: Composite Function 11	8/10/1/600
E1-F26: Composite Function 12	8/10/1/600
E1-F27: Composite Function 13	10/10/1/600
E1-F28: Composite Function 14	10/10/1/600
E1-F29: Composite Function 15	10/10/1/600

Peak ratio (average peaks found/total), and (*t*-test/ Wilcoxon test) for Benchmark 1 in 25 independent runs. Best results in bold

Function	EARSDE	SDE	CDE	ShrDE	NSDE	NCDE	NShrDE	FERPSO	PNPCDE	SPSO	r2PSO	r3PSO
E1-F1	**1/1 (N.A)**	1/1 (0/0)	1/1 (0/0)	1/1 (0/0)	1/1 (0/0)	1/1 (0/0)	1/1 (0/0)	0.72/1 (1/1)	1/1 (0/0)	0.48/1 (1/1)	0.76/1 (1/1)	0.84/1 (1/1)
E1-F2	**1/1 (N.A)**	1/1 (0/0)	1/1 (0/0)	1/1 (0/0)	1/1 (0/0)	1/1 (0/0)	1/1 (0/0)	1/1 (0/0)	1/1 (0/0)	0.44/1 (1/1)	0.88/1 (1/1)	0.96/1 (1/0)
E1-F3	**2/2 (N.A)**	1.96/2 (0/0)	2/2 (0/0)	2/2 (0/0)	2/2 (0/0)	2/2 (0/0)	2/2 (0/0)	0.8/2 (1/1)	2/2 (0/0)	0.24/2 (1/1)	0.48/2 (1/1)	0.6/2 (1/1)
E1-F4	**5/5 (N.A)**	4.72/5 (1/1)	3.84/5 (1/1)	3.28/5 (1/1)	5/5 (0/0)	5/5 (0/0)	5/5 (0/0)	4.84/5 (1/1)	5/5 (0/0)	4.88/5 (0/0)	4.92/5 (0/0)	4.88/5 (0/0)
E1-F5	**1/1 (N.A)**	1/1 (0/0)	0.72/1 (1/1)	0.44/1 (1/1)	1/1 (0/0)	1/1 (0/0)	1/1 (0/0)	1/1 (0/0)	1/1 (0/0)	1/1 (0/0)	1/1 (0/0)	1/1 (0/0)
E1-F6	**5/5 (N.A)**	4.6/5 (1/1)	3.96/5 (1/1)	3.28/5 (1/1)	5/5 (0/0)	5/5 (0/0)	5/5 (0/0)	5/5 (0/0)	5/5 (0/0)	4.92/5 (0/0)	4.88/5 (0/0)	4.72/5 (0/0)
E1-F7	**1/1 (N.A)**	1/1 (0/0)	0.6/1 (1/1)	0.4/1 (1/1)	1/1 (0/0)	1/1 (0/0)	1/1 (0/0)	1/1 (0/0)	1/1 (0/0)	1/1 (0/0)	1/1 (0/0)	1/1 (0/0)
E1-F8	**4/4 (N.A)**	3.72/4 (1/1)	0.32/4 (1/1)	0.16/4 (1/1)	4/4 (0/0)	4/4 (0/0)	3.92/4 (0/0)	3.68/4 (1/1)	4/4 (0/0)	0.84/4 (1/1)	2.92/4 (1/1)	2.76/4 (1/1)
E1-F9	**2/2 (N.A)**	2/2 (0/0)	0.04/2 (1/1)	0.04/2 (1/1)	2/2 (0/0)	2/2 (0/0)	2/2 (0/0)	1.96/2 (0/0)	2/2 (0/0)	0.08/2 (1/1)	1.44/2 (1/1)	1.56/2 (1/1)
E1-F10	**1/1 (N.A)**	0.32/1 (1/1)	0.52/1 (1/1)	0.96/1 (0/0)	1/1 (0/0)	1/1 (0/0)	0.96/1 (0/0)	1/1 (0/0)	1/1 (0/0)	0.56/1 (1/1)	0.88/1 (1/1)	0.76/1 (1/1)
E1-F11	**18/18 (N.A)**	12.4/18 (1/1)	17.7/18 (0/0)	16.56/18 (1/1)	18/18 (0/0)	18/18 (0/0)	18/18 (0/0)	17.4/18 (1/1)	18/18 (0/0)	8.52/18 (1/1)	15. 2/18 (1/1)	15.6/18 (1/1)
E1-F12	**6/6 (N.A)**	4.88/6 (1/1)	5.56/6 (1/1)	5.6/6 (1/1)	5.84/6 (0/0)	5.8/6 (1/1)	5.88/6 (0/0)	5.36/6 (1/1)	6/6 (0/0)	5.6/6 (1/1)	5.52/6 (1/1)	5.16/6 (1/1)
E1-F13	**36/36 (N.A)**	22.8/36 (1/1)	33.8/36 (1/1)	35.92/36 (0/0)	30.6/36 (1/1)	35.9/36 (0/0)	35.96/36 (0/0)	23.6/36 (1/1)	35.92/36 (0/0)	25.7/36 (1/1)	21.8/36 (1/1)	22.2/36 (1/1)
E1-F14	**211/216 (N.A)**	50.6/216 (1/1)	152/216 (1/1)	197.88/216 (1/1)	84.28/216 (1/1)	179/216 (1/1)	198.96/216 (1/1)	68.6/216 (1/1)	204/216 (0/1)	70.1/216 (1/1)	40.6/216 (1/1)	45.4/216 (1/1)

Average number of peaks found for Benchmark 2 for 25 independent runs with ranking besides. (*t*-test/Wilcoxon test) below (1=significance).

Function	EARSDE	ARSDE	SDE	CDE	ShrDE	NSDE	NCDE	NShrDE	FERPSO	PNPCDE	SPSO	r2PSO	r3PSO
E1-F15	**6.98(1)** **(N.A.)**	5.96(4) (1/1)	1.79(7) (1/1)	0(11) (1/1)	0(11) (1/1)	6.7(2) (1/1)	5.18(5) (1/1)	3.7(6) (1/1)	1.08(8) (1/1)	6.48(3) (1/1)	0(11) (1/1)	0(11) (1/1)	0(11) (1/1)
E1-F16	**6(1)** **(N.A.)**	4.88(2) (1/1)	1.2(8.5) (1/1)	1.2(8.5) (1/1)	1.1(10) (1/1)	4(4) (1/1)	3.6(5) (1/1)	2.8(6) (1/1)	2(7) (1/1)	4.2(3) (1/1)	0(12) (1/1)	0(12) (1/1)	0(12) (1/1)
E1-F17	**6(1)** **(N.A.)**	4.44(5) (1/1)	1.5(8) (1/1)	0.7(10) (1/1)	1.11(9) (1/1)	6(1) (0/0)	5.8(4) (1/1)	4(6) (1/1)	2.5(7) (1/1)	6(1) (0/0)	0(12) (1/1)	0(12) (1/1)	0(12) (1/1)
E1-F18	**6(1)** **(N.A.)**	6(1) (0/0)	0(10) (1/1)	0(10) (1/1)	0(10) (1/1)	5.4(4) (1/1)	4.8(5) (1/1)	4.5(6) (1/1)	0(10) (1/1)	5.44(3) (1/1)	0(10) (1/1)	0(10) (1/1)	0(10) (1/1)
E1-F19	**6(1)** **(N.A.)**	4.56(5) (1/1)	1.3(8.5) (1/1)	1.1(10) (1/1)	1.3(8.5) (1/1)	5.9(2) (1/1)	5.2(3) (1/1)	3.6(6) (1/1)	2(7) (1/1)	5.16(4) (1/1)	0(12) (1/1)	0(12) (1/1)	0(12) (1/1)
E1-F20	**5.14(1)** **(N.A.)**	3.88(2) (1/1)	1.4(7) (1/1)	0(11) (1/1)	0(11) (1/1)	3(4.5) (1/1)	3(4.5) (1/1)	3(4.5) (1/1)	1.2(8) (1/1)	3 (4.5) (1/1)	0(11) (1/1)	0(11) (1/1)	0(11) (1/1)
E1-F21	**4.98(1)** **(N.A.)**	3.2(3) (1/1)	0(10.5) (1/1)	0(10.5) (1/1)	0(10.5) (1/1)	1.9(4) (1/1)	1.8(5) (1/1)	1(6) (1/1)	0.5(7) (1/1)	3.32(2) (1/1)	0(10.5) (1/1)	0(10.5) (1/1)	0(10.5) (1/1)
E1-F22	**4.72(1)** **(N.A.)**	3.36(2) (1/1)	1.4(8) (1/1)	0(11) (1/1)	0(11) (1/1)	3(4.5) (1/1)	3(4.5) (1/1)	3(4.5) (1/1)	1.5(7) (1/1)	3(4.5) (1/1)	0(11) (1/1)	0(11) (1/1)	0(11) (1/1)
E1-F23	**4.38(1)** **(N.A.)**	3.04(3) (1/1)	1.8(7) (1/1)	0(11) (1/1)	0(11) (1/1)	3(5) (1/1)	3(5) (1/1)	3(5) (1/1)	1.5(8) (1/1)	3.92(2) (1/1)	0(11) (1/1)	0(11) (1/1)	0(11) (1/1)
E1-F24	**4.02(1)** **(N.A.)**	3.6(2) (1/1)	1.1(6.5) (1/1)	0(11) (1/1)	0(11) (1/1)	2(4) (1/1)	1.3(5) (1/1)	1(8) (1/1)	1.1(6.5) (1/1)	2.2(3) (1/1)	0(11) (1/1)	0(11) (1/1)	0(11) (1/1)
E1-F25	**4.12(1)** **(N.A.)**	2.82(4) (1/1)	1(7) (1/1)	0(10.5) (1/1)	0(10.5) (1/1)	4(2) (1/1)	2.8(5) (1/1)	2.2(6) (1/1)	0(10.5) (1/1)	3.2(3) (1/1)	0(10.5) (1/1)	0(10.5) (1/1)	0(10.5) (1/1)
E1-F26	**6.24(1)** **(N.A.)**	5.72(2) (1/1)	1.48(8) (1/1)	0(11) (1/1)	0(11) (1/1)	2.9(4) (1/1)	2.5(5) (1/1)	2(6) (1/1)	1.6(7) (1/1)	3.76(3) (1/1)	0(11) (1/1)	0(11) (1/1)	0(11) (1/1)
E1-F27	**5.97(1)** **(N.A.)**	3.36(4) (1/1)	0.8(7) (1/1)	0(11) (1/1)	0(11) (1/1)	3.8(3) (1/1)	2.3(5) (1/1)	1(6) (1/1)	0.3(8) (1/1)	3.88(2) (1/1)	0(11) (1/1)	0(11) (1/1)	0(11) (1/1)
E1-F28	**6.56(1)** **(N.A.)**	4.04(2) (1/1)	1.96(3) (1/1)	0(11) (1/1)	0(11) (1/1)	1(6.5) (1/1)	1(6.5) (1/1)	1(6.5) (1/1)	1(6.5) (1/1)	1.04(4) (1/1)	0(11) (1/1)	0(11) (1/1)	0(11) (1/1)
E1-F29	**4.46(1)** **(N.A.)**	3.32(5) (1/1)	1.6(7) (1/1)	0(11) (1/1)	0(11) (1/1)	4(3) (1/1)	3.8(4) (1/1)	2.4(6) (1/1)	1.2(8) (1/1)	4.32(2) (1/1)	0(11) (1/1)	0(11) (1/1)	0(11) (1/1)
Total rank	**15**	46	113	148.5	157.5	54	71	88.5	115.5	44	166	166	166

THANK YOU

Q & A