Tutorial on

DIFFERENTIAL EVOLUTION: Recent Advances

P. N. Suganthan
School of Electrical and Electronic Engineering
Nanyang Technological University, Singapore
epnsugan@ntu.edu.sg
http://www.ntu.edu.sg/home/epnsugan

http://www.sigevo.org/gecco-2013/

GECCO'13 Companion, July 6–10, 2013, Amsterdam, The Netherlands.
ACM 978-1-4503-1964-5/13/07.

NANYANG TECHNOLOGICAL UNIVERSITY

Overview

I. Introduction

II. Some DE Variants for Single Objective Optimization

III. Multimodal Optimization

IV. Multiobjective Optimization

V. Large Scale Optimization

VI. Dynamic Optimization

VII. Constrained Optimization

NANYANG TECHNOLOGICAL UNIVERSITY

Benchmark Test Functions

Resources available from
http://www.ntu.edu.sg/home/epnsugan
(limited to our own work & CEC Competitions)

Ensemble Methods for Evolutionary Algorithms

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

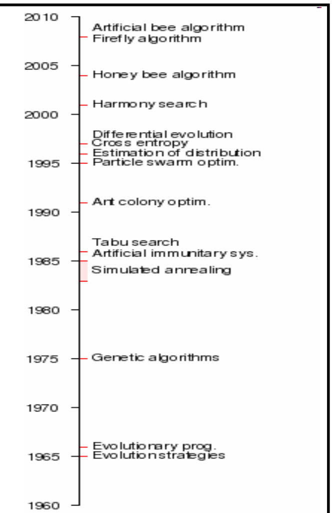
S. Das and P. N. Suganthan, **"**Differential Evolution: A Survey of the State-of-the-Art", *IEEE Trans. on Evolutionary Computation*, 15(1):4 – 31, Feb. 2011.

NANYANG TECHNOLOGICAL UNIVERSITY

Meta-heuristics

A metaheuristic is a heuristic method for solving a very general class of computational problems by combining user-given black-box procedures — usually heuristics themselves — in the hope of obtaining a more efficient or more robust procedure. The name combines the Greek prefix "meta" ("beyond", here in the sense of "higher level") and "heuristic" (from ευρισκειν, heuriskein, "to find").

This Figure is simplified from:
http://en.wikipedia.org/wiki/Metaheuristic



NANYANG TECHNOLOGICAL UNIVERSITY

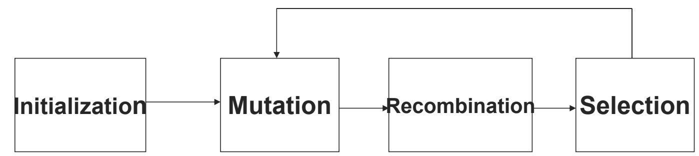
Differential Evolution

- **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 GA and PSO on a 34-function test suite (Vesterstrom & Thomsen, 2004)**
- **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.**

5

NANYANG TECHNOLOGICAL UNIVERSITY

- **DE can be regarded as an Evolutionary Algorithm**
- **This Class also includes GA, Evolutionary Programming and Evolutionary Strategies**

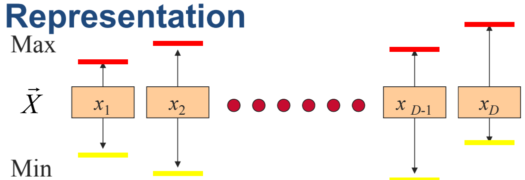


Basic steps of an Evolutionary Algorithm

6

NANYANG TECHNOLOGICAL UNIVERSITY

Representation



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

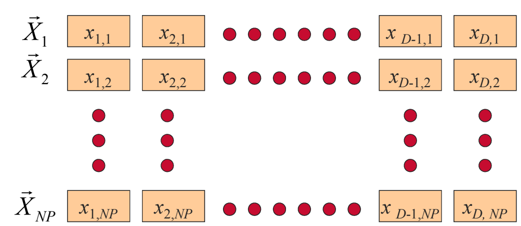
May wish to constrain the values taken in each domain above and below.

7

NANYANG TECHNOLOGICAL UNIVERSITY

Maintain Population - *NP*

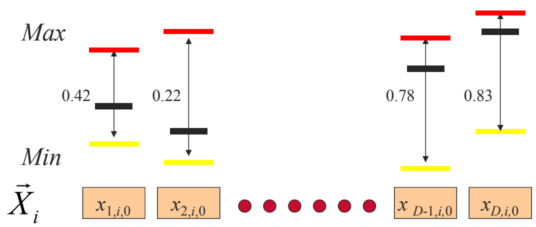
We will maintain a population of size NP



8

NANYANG TECHNOLOGICAL UNIVERSITY

Initialization → Mutation → Recombination → Selection



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

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

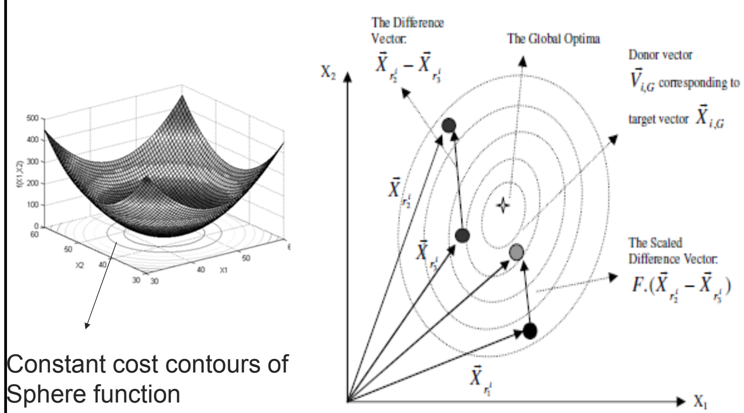
Initialization → Mutation → Recombination → Selection

- 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, 1+)

Example of formation of donor vector over two-dimensional constant cost contours



Constant cost contours of Sphere function

Initialization → Mutation → Recombination → Selection

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*.

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

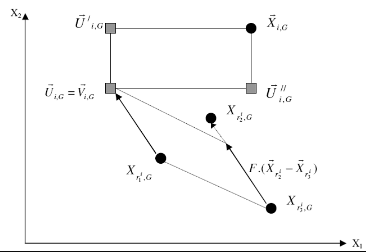
An Illustration of Binomial Crossover in 2D Space

Three possible trial vectors:

i) $\vec{U}_{i,G} = \vec{V}_{i,G}$ such that both the components of $\vec{U}_{i,G}$ are inherited from $\vec{V}_{i,G}$.

ii) $\vec{U}^{/}_{i,G}$, in which the first component ($j = 1$) comes from $\vec{V}_{i,G}$ and the second one ($j = 2$) from $\vec{X}_{i,G}$.

iii) $\vec{U}^{//}_{i,G}$, in which the first component ($j = 1$) comes from $\vec{X}_{i,G}$ and the second one ($j = 2$) from $\vec{V}_{i,G}$.



13

NANYANG TECHNOLOGICAL UNIVERSITY

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)) ;
```

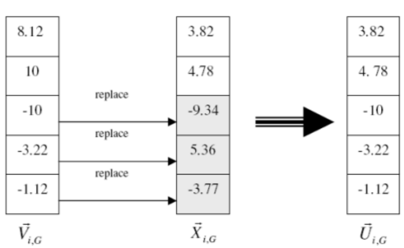
14

NANYANG TECHNOLOGICAL UNIVERSITY

Example: Let us consider the following pair of donor and target vectors

$$\vec{X}_{i,G} = \begin{bmatrix} 3.82 \\ 4.78 \\ -9.34 \\ 5.36 \\ -3.77 \end{bmatrix} \qquad \vec{V}_{i,G} = \begin{bmatrix} 8.12 \\ 10 \\ -10 \\ -3.22 \\ -1.12 \end{bmatrix}$$

Suppose *n* = 2 and *L* = 3 for this specific example. Then the exponential crossover process can be shown as:



15

NANYANG TECHNOLOGICAL UNIVERSITY

Scalable EXP Xover

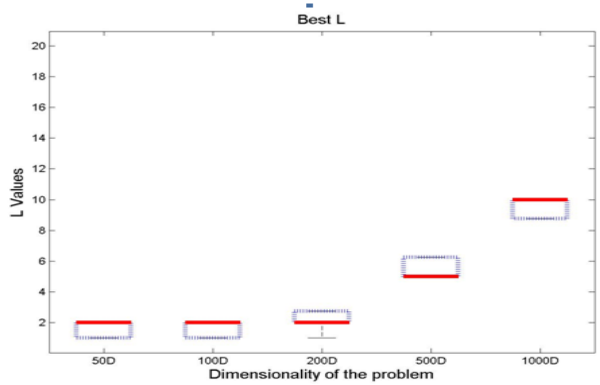
- Original EXP does not scale with dimensionality of the problem.
- *Cr* is directly related to the number of dimensions copied from the mutant vector for BIN Xover.
- Our investigations suggest that a good value for *L* is 1% of the dimensionality of the problem (based on 19 problems used for the Soft Computing Journal's Large Scale Optimization Problems.

S. Z. Zhao, P N Suganthan, "Empirical Investigations into the Exponential Crossover of Differential Evolution." *Swarm and Evolutionary Computation,* Vol. 9, April 2013, pp. 27–36.

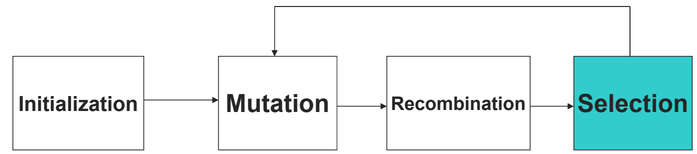
16

NANYANG TECHNOLOGICAL UNIVERSITY

Scalable EXP Xover



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.



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

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

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)

The Crossover Rate *Cr*:

1) The parameter *Cr* controls how many parameters in expectation, are changed in a population member.
2) Low value of *Cr*, a small number of parameters are changed in each generation and the stepwise movement tends to be orthogonal to the current coordinate axes.
3) High values of *Cr* (near 1) cause most of the directions of the mutant vector to be inherited prohibiting the generation of axis orthogonal steps.

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 a DE algorithm.

4) But, recently, all best performing DE algorithms 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.

21 NANYANG TECHNOLOGICAL UNIVERSITY

Prominent Real-world Applications of DE (From DE-Survey- 2011)

Sub areas and details	Types of DE applied and references
Electrical Power Systems	
Economic dispatch Optimal power flow Power system planning, generation expansion planning Capacitor placement Distribution systems' network reconfiguration Power filter, power system stabilizer	Chaotic DE [S31], Hybrid DE with acceleration and migration [S87], DE/rand/1/bin [S88], hybrid DE with improved constraint handling [S89], variable scaling hybrid DE [S90] DE/target-to-best/1/bin [S91], Cooperative Co-evolutionary DE [S92], DE/rand/1/bin with non-dominated sorting [S93], conventional DE/rand/1/bin [S94, S96], DE with Random Localization (DERL) [S95]. Modified DE with fitness sharing [S97], conventional DE/rand/1/bin [S98], comparison of 10 DE strategies of Storn and Price [S99], Robust Searching Hybird DE (RSHDE) [S100] Hybrid of Ant System and DE [S49] Hybrid DE with variable scale factor [S101], mixed integer hybrid DE [S185]. Hybrid DE with acceleration and migration operators [S102], DE/target-to-best/1/bin [S103], hybrid of DE with ant systems [S104]
Electromagnetism, Propagation, and Microwave Engineering	
Capacitive voltage divider Electromagnetic inverse scattering Design of circular waveguide mode converters Parameter estimation and property analysis for electromagnetic devices, materials, and machines electromagnetic imaging Antenna array design	Multi-Objective DE (MODE) and NSDE (DE with Non-dominated Sorting) [S105] DE/rand/1/bin [S106], conventional DE with individuals in groups (GDES) [S107], Dynamic DE [77] DE/rand/1/bin [S108] DE/rand/1/bin [S109 – S111, S113], DE/target-to-best/1/bin [S112] conventional DE/rand/1/bin [S114, S115], DE/best/1/bin [S116] multi-member DE (see [93] for details) [S117], hybrid real/integer-coded DE [S118], DE/rand/1/bin [S119, S120], modified DE with refreshing distribution operator and fittest individual refinement operator [S121], DE/best/1/bin [S122], MOEA/D-DE [68,69]
Control Systems and Robotics	
System identification Optimal control problems Controller design and tuning Aircraft control nonlinear system control	Conventional DE/rand/1/bin [S123 – S126] DE/rand/1/bin and DE/best/2/bin [S127], modified DE with adjustable control weight gradient methods [S128]. Self adaptive DE [S129], DE/rand/1 with arithmetic crossover [S130], DE/rand/1/bin with random scale factor and time-varying *Cr* [S131]. Hybrid DE with downhill simplex local optimization [55]. Hybrid DE with convex mutation [15].

22 NANYANG TECHNOLOGICAL UNIVERSITY

Sub areas and details	Types of DE applied and references
Bioinformatics	
Gene regulatory networks Micro-array data analysis Protein folding Bioprocess optimization	DE with adaptive local search (see [22] for details) [63], hybrid of DE and PSO [S137] Multi-objective DE-variants (MODE, DEMO) [S138] DE/rand/1/bin [S139] DE/rand/1/bin [S140]
Chemical Engineering	
Chemical process synthesis and design Phase equilibrium and phase study Parameter estimation of chemical process	Modified DE with single array updating [S141, 7], 10 DE-variants of Storn and Price (see [74,75]) compared in [S142, S144], multi-objective DE [S143], hybrid DE with migration and acceleration operators [S145]. DE/rand/1/bin [S146]. Hybrid DE with geometric mean mutation [S147], DE/target-to-best/1/exp [S148].
Pattern Recognition and Image Processing	
Data clustering Pixel clustering and region-based image segmentation Feature extraction Image registration and enhancement Image Watermarking	DE/rand/1/bin [S149], DE with random scale factor and time-varying crossover rate [20], DE with neighborhood-based mutation [S150] Modified DE with local and global best mutation [S151], DE with random scale factor and time-varying crossover rate [S152]. DE/rand/1/bin [S153] DE/rand/1/bin [S154], DE with chaotic local search [S155] DE/rand/1/bin and DE/target-to-best/1/bin [S156]
Artificial neural networks (ANN)	
Training of feed-forward ANNs Training of wavelet neural networks (WNNs) Training of B-Spline neural networks	DE/rand/1/bin [S157, S160], generalization-based DE (GDE) [S158], DE/target-to-best/1/bin [S159] DE/rand/1/bin [S161] DE with chaotic sequence-based adjustment of scale factor *F* [S162]
Signal Processing	
estimation Digital filter design Fractional order signal processing	Dynamic DE (DyDE) [S163] DE/rand/1/bin [S164, S165], DE with random scale factor [S166] DE/rand/1/bin [S167]

23 NANYANG TECHNOLOGICAL UNIVERSITY

Overview

I. Introduction

II. Some DE Variants for Single Objective Optimization

III. Multimodal Optimization in DE

IV. Multiobjective Optimization

V. Large Scale Optimization

VI. Dynamic Optimization

VII. Constrained Optimization

24 NANYANG TECHNOLOGICAL UNIVERSITY

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.

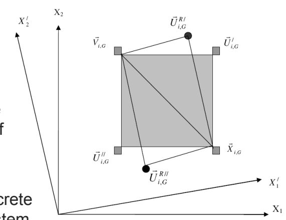
General form: $\vec{W}_{i,G} = \vec{X}_{r_1,G} + k_i.(\vec{X}_{r_1,G} - \vec{X}_{r_2,G})$

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.

The 'jDE' Algorithm (Brest et al., 2006)

- **Control parameters *F* and *Cr* are coded into the individual and adjusted them by introducing two new parameters τ_1 and τ_2**
- **The new control parameters for the next generation are computed as follows:**

$$F_{i,G+1} = F_l + rand_1 * F_u \text{ if } rand_2 < \tau_1$$
$$= F_{i,G} \text{ else.}$$
$$Cr_{i,G+1} = rand_3 \text{ if } rand_4 < \tau_2$$
$$= Cr_{i,G} \text{ else,}$$

$\tau_1 = \tau_2 = 0.1 \quad F_l = 0.1,$

The new *F* takes a value from [0.1, 0.9] while the new *Cr* takes a value from [0, 1].

J. Brest, S. Greiner, B. Bošković, M. Mernik, and V. Žumer, "Self-adapting Control parameters in differential evolution: a comparative study on numerical benchmark problems," *IEEE Trans. on Evolutionary Computation*, Vol. 10, Issue 6, pp. 646 – 657, 2006

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*).

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.**

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.

Opposition-based DE (Rahnamayan *et al.*, 2008)

- **Three stage modification to original DE framework based on the concept of *Opposite Numbers* :**

  Let x be a real number defined in the closed interval [a, b]. Then the opposite number of x may be defined as:

$$\breve{x} = a + b - x$$

ODE Steps:

1) Opposition based Population Initialization: Fittest NP individuals are chosen as the starting population from a combination of NP randomly generated population members and their opposite members.

2) **Opposition Based Generation Jumping:** In this stage, after each iteration, instead of generating new population by evolutionary process, the opposite population is calculated with a predetermined probability *Jr* () and the *NP* fittest individuals may be selected from the current population and the corresponding opposite population.

29

NANYANG TECHNOLOGICAL UNIVERSITY

Rahnamayan, H. R. Tizhoosh, and M. M. A. Salama, "Opposition-based differential evolution", *IEEE Trans. on Evolutionary Computation*, Vol. 12, No. 1, pp. 64-79, 2008.

ODE (Contd.)

3) Opposition Based Best Individual Jumping: In this phase, at first a difference-offspring of the best individual in the current population is created as:

$$\vec{X}_{new\_best,G} = \vec{X}_{best,G} + F'.(\vec{X}_{r_1,G} - \vec{X}_{r_2,G})$$

where r_1 and r_2 are mutually different random integer indices selected from {1, 2, ..., *NP*} and *F'* is a real constant. Next the opposite of offspring is generated as $\vec{X}_{opp\_newbestG}$ Finally the current best member is replaced by the fittest member of the set $\{\vec{X}_{best,G}, \vec{X}_{new\_best,G}, \vec{X}_{opp\_newbest,G}\}$

30

NANYANG TECHNOLOGICAL UNIVERSITY

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 $P \cup A$

31

NANYANG TECHNOLOGICAL UNIVERSITY

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:

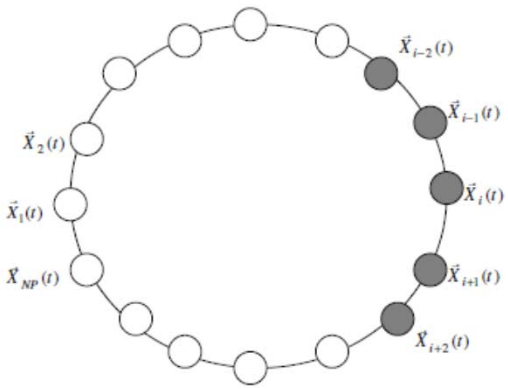
$$mean_L(S_F) = \frac{\sum_{F \in S_F} F^2}{\sum_{F \in S_F} F}$$

32

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

NANYANG TECHNOLOGICAL UNIVERSITY

Differential Evolution with Neighborhood-based Mutation



Swagatam Das, Amit Konar, Uday K. Chakraborty, and Ajith Abraham, "Differential evolution with a neighborhood based mutation operator: a comparative study", *IEEE Transactions on Evolutionary Computing*, Vol 13, No. 3, June 2009.

33

Local Mutation Model:

$$\vec{L}_i(t) = \vec{X}_i(t) + \alpha \cdot (\vec{X}_{n\_best_i}(t) - \vec{X}_i(t)) + \beta \cdot (\vec{X}_p(t) - \vec{X}_q(t))$$

Global Mutation Model:

$$\vec{g}_i(t) = \vec{X}_i(t) + \alpha \cdot (\vec{X}_{g\_best}(t) - \vec{X}_i(t)) + \beta \cdot (\vec{X}_{r_1}(t) - \vec{X}_{r_2}(t))$$

Combined Model for Donor Vector generation:

$$\vec{V}_i(t) = w.\vec{g}_i(t) + (1-w).\vec{L}_i(t)$$

The weight factor *w* may be adjusted during the run or self-adapted through the evolutional learning process.

34

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.

35

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
- Trial vector better than target vector - retain setting
- Trial vector worse than target vector - re-initialize setting
- Increased probability of offspring production by better combination

36

EPSDE

- 14 problems – (10D, 30D and 50D)
- 30 runs (100000, 300000 and 500000FEs for 10D, 30D and 50D)
- *NP* = 50 (all algorithms)

EPSDE	*D*	SaDE	jDE	ADE	SDE	JADE	Total
inferior	10	1	1	0	0	1	3
	30	1	1	0	1	1	4
	50	1	1	0	1	2	5
equal	10	12	9	9	6	9	45
	30	6	6	8	0	6	26
	50	3	4	9	1	5	22
better	10	1	4	5	8	4	**22**
	30	7	7	6	13	7	**40**
	50	10	9	5	12	7	**43**

- Scalability

37 NANYANG TECHNOLOGICAL UNIVERSITY

EPSDE -SEMCCO 2010

- Crossover: Binomial and Exponential
- DE/rand/1/bin replaced with JADE mutation
- Test Problems: 25 benchmark problems of CEC 2005 (10D & 30D)
- Runs : 25
- Statistical *t*-test
- Comparative Results with JADE (Jingqiao's & Arthur's DE, i.e. JADE):
  - EPSDE better, similar and worst in 13, 8 and 4 in 10D
  - EPSDE better, similar and worst in 13, 9 and 3 in 30D

R. Mallipeddi and P. N. Suganthan, "Differential Evolution Algorithm with Ensemble of Parameters and Mutation and Crossover Strategies", Swarm Evolutionary and Memetic Computing Conference, pp. 71-78, LNCS, Vol. 6466, Chennai, India 2010.

38 NANYANG TECHNOLOGICAL UNIVERSITY

The MDE_*p*BX Algorithm

- In MDE_*p*BX, we propose a new mutation strategy, a fitness-induced parent selection scheme for the binomial crossover of DE, and a simple but effective scheme of adapting two of its most important control parameters.
- First, a less greedy and more explorative variant of DE/current-to-best/1 is used. We call it DE/current-to-gr_best/1.
- DE/current-to-gr_best/1 utilizes the best member of a dynamic group of *q*% population members to perturb the target vector.
- This overcomes the limitations of fast but less reliable convergence performance of DE/current-to-best/1.

Sk. Minhazul Islam, S. Das, S. Ghosh, S. Roy, and P. N. Suganthan, "An Adaptive Differential Evolution Algorithm with Novel Mutation and Crossover Strategies for Global Numerical Optimization", *IEEE Trans. on* SMC-B, Vol. 42, No. 2, pp. 482-500, 2012.

39 NANYANG TECHNOLOGICAL UNIVERSITY

Algorithmic Components of MDE_*p*BX

- Second, we modify the conventional binomial crossover of DE by introducing a fitness-induced bias in the selection of parents from the current generation.
- The exploitative crossover scheme is referred to as "*p*-best crossover".
- Here a mutant vector is allowed to exchange its components through binomial crossover with a randomly selected member from the *p* top-ranked individuals of the current generation instead of exchanging with the parent of the same index
- Third, we suggest simple schemes to update the values of *F* and *Cr* in each generation, guided by the knowledge of their successful values that were able to generate better offspring in the last generation.

40 NANYANG TECHNOLOGICAL UNIVERSITY

The DCMA-EA Algorithm

- Hybridization aims at combining the operators and methodologies from different Evolutionary Computation paradigms to form a single algorithm that enjoys a statistically superior performance.
- DCMA-EA is an hybridization of CMA-ES and DE algorithm that aims at improving the performance of the CMA-ES algorithm on complicated landscapes, such as noisy and hybrid composition functions.
- In the proposed hybridization, our aim is to incorporate the difference-vector based mutation scheme of DE into CMA-ES as these difference vectors have the ability to adjust to the natural scaling of the problem.
- Further, in order to enhance the diversity among the population members as well as increase the convergence speed, the selection and crossover operators of DE have also been embedded.

S. Ghosh, S. Das, S. Roy, Sk. Minhazul Islam, and P. N. Suganthan "A Differential Covariance Matrix Adaptation Evolutionary Algorithm for Real Parameter Optimization", *Information Sciences,* Vol. 182, No. 1, pp 199-219 Jan. 2012.

A Special Issue on DE appeared in: Feb. 2011

IEEE TRANSACTIONS ON
EVOLUTIONARY COMPUTATION
A PUBLICATION OF THE IEEE COMPUTATIONAL INTELLIGENCE SOCIETY
www.ieee-cis.org/pubs/tec

Guest Editors:
Swagatam Das, Jadavpur University, India
P. N. Suganthan, Nanyang Technological University,Singapore.
Carlos A. Coello Coello, Col. San Pedro Zacatenco, México.

This tutorial is partly based on the survey article published in this special issue:

S. Das and P. N. Suganthan, "Differential evolution – a survey of the state-of-the-art", IEEE TEVC, Vol. 15, No. 1, feb. 2011.

Introduction to a few DE-variants from the special issue of IEEE TEVC:

1) cDE (Mininno et al., "Compact differential evolution", IEEE TEVC, Vol 15, No. 1):

- An interesting Estimation of Distribution Algorithm (EDA) which, analogously to other compact Evolutionary Algorithms (cEAs), does not store and process the entire population and all of its individuals, but adopts instead a static representation of the population to perform the optimization process.

- However, unlike other cEAs, cDE encodes the actual survivor selection mechanism of the original DE algorithm, as well as its original search logic (i.e., its crossover and mutation operators).

- cDE uses a fairly limited amount of memory, which makes it appropriate for hardware implementations.

CoDE (Wang *et al.*, "Differential evolution with composite trial vector generation strategies and control parameters", IEEE TEVC, Vol. 15, No. 1):

-adopts three different trial vector generation strategies (rand/1/bin, rand/2/bin, and current-to-rand/1) and three mechanisms to control DE's parameter settings. The parametric choices were:

1) F = 1.0, Cr = 0.1 - meant for dealing with separable problems.
2) F = 1.0, Cr = 0.9 – meant for maintaining population diversity.
3) F = 0.8, Cr = 0.2 – meant for exploiting the search space and achieving better convergence characteristics.

-Such strategies offer different advantages and, somehow, complement one another.

DE with Proximity-based Mutation Operators (Epitropakis *et al.*, "Enhancing differential evolution with proximity-based mutation operators", IEEE TEVC, Vol. 15, No. 1):

-is based on framework that incorporates information of neighboring individuals to guide the search towards the global optimum in a more efficient manner.

-The main idea is to adopt a stochastic selection mechanism in which the probability of selecting an individual to become a parent is inversely proportional to its distance from the individual undergoing mutation.

-This will favor the search in the vicinity of the mutated individual, which should promote a proper exploitation of such a neighborhood, without sacrificing the exploration capabilities of the mutation operator.

-The authors incorporate the proposed framework to several DE variants, finding that in most cases its use significantly improves the performance of the algorithm (when there is no improvement, there is no significant degradation in performance either).

Overview

I. Introduction

II. Some DE variants for Single Objective Optimization

III. Multimodal Optimization in DE

IV. Multiobjective Optimization

V. Large Scale Optimization

VI. Dynamic Optimization

VII. Constrained Optimization

Multi-modal Optimization

- Aim: To find multiple global and local optima

  (Resonance points of an Electrical Circuit)

- Evolutionary Algorithms vs. Classical Optimization Methods
- EAs converge to the global or a sub-optimal point
- Prevent convergence to a single solution and maintain multiple solutions – Niching (each desired solution is called a niche)

Multi-modal Optimization Methods

- Some existing Niching Techniques
  - Sharing
  - Clearing
  - Crowding
  - Restricted Tournament Selection
  - Clustering
  - Species Based
  - Neighborhood based DE (very competitive)

Multi-modal Optimization - Sharing

- Sharing
  - First among Niching Techniques
  - Proposed by Holland – Improved by Goldberg & Richardson
  - Population divided into subgroups based on similarity of individuals (σ - threshold of dissimilarity or niche radius)
  - Information sharing with other individuals in the same niche (Fitness sharing)

$$f_i' = \frac{f_i}{m}$$

  m is niche count

  - Complexity – $O(NP)$

  NP – population size

49

NANYANG TECHNOLOGICAL UNIVERSITY

Multi-modal Optimization - Clearing

- Clearing
  - Retain the best members while eliminating the worst individuals of each niche
  - Complexity – $O(cNP)$

  NP – population size, c – number of subpopulations

  - Advantages
    - Lower Complexity
    - Significant reduction in genetic drift due to selection noise
    - Population can be much smaller

50

NANYANG TECHNOLOGICAL UNIVERSITY

Multi-modal Optimization - Crowding

- Crowding
  - Proposed by De Jong
  - Newly generated individuals replace similar individuals
  - Similarity determined by distance metric
  - 2 parents randomly selected and produce 2 offspring by Mutation and crossover
  - Offspring replace nearest or similar parent if they are of greater fitness
  - Complexity $O(NP)$

51

NANYANG TECHNOLOGICAL UNIVERSITY

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 recent IEEE TEC articles.

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.

52

NANYANG TECHNOLOGICAL UNIVERSITY

Multi-modal Optimization - RTS

- Restricted Tournament Selection (RTS)
  - Proposed by Harick
  - Similar to Crowding
  - Corresponding to each offspring randomly choose *w* individuals from the population

    w – window size
  - From *w*, pick the nearest or similar individual to the offspring
  - Restricts competition with dissimilar individuals
  - Complexity – $O(NP * w)$
  - DE with ensemble of Restricted Tournament Selection

B. Y. Qu and P. N. Suganthan, "Novel multimodal Problems and Differential Evolution with Ensemble of Restricted Tournament Selection", IEEE Congress on Evolutionary Computation, pp. 1-7, Barcelona, Spain, July 2010.

Multi-modal Optimization - Clustering

- Clustering
  - Proposed by Yin
  - Clustering helps to form niches – K-mean algorithm used
  - Reduces time complexity
  - Complexity – $O(Nc*NP)$

    NP – population size, *Nc* – number of clusters

Multi-modal Optimization

- Species based
  - Separating population into several species based on similarity
  - Similar to sharing – except no change in fitness

    (σ - species distance)
- Ensemble of Niching Algorithms (ENA)
  - Population divided into niches using various niching methods
  - Same selection and survival criteria used

E. L. Yu, P. N. Suganthan, "Ensemble of niching algorithms", *Information Sciences*, Vol. 180, No. 15, pp. 2815-2833, Aug. 2010.

V. Ensemble of niching algorithms

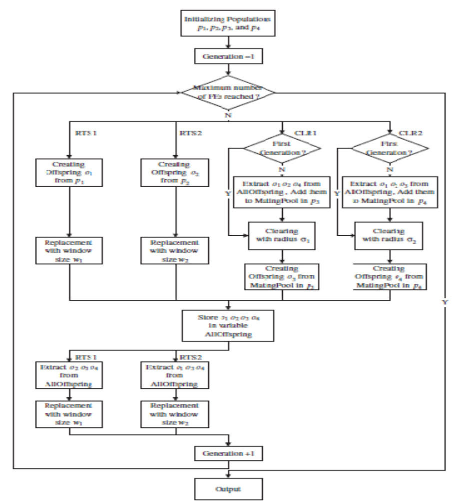


Fig. 3. Flowchart of the ensemble of niching algorithm.

Overview

I. Introduction

II. Some DE variants for Single Objective Optimization

III. Multimodal Optimization in DE

IV. Multiobjective Optimization

V. Large Scale Optimization

VI. Dynamic Optimization

VII. Constrained Optimization

Non-Domination Sorting

- Multi-objective optimization
  Minimize: $F(x) = (f_1(x), f_2(x), ..., f_m(x))^T$
  Subject to: $x_i^L \le x_i \le x_i^U$
- Non-domination sorting
  The solution $\boldsymbol{x_1}$ is no worse than $\boldsymbol{x_2}$ in all objectives.
  The solution $\boldsymbol{x_1}$ is strictly better than $\boldsymbol{x_2}$ in at least one objective.
- Most current MOEAs are based on non-domination sorting. The standard DE algorithm (i.e. xover & mutation) is used with different selection mechanisms.
- Recently, decomposition method was proposed which decompose an MOP into a large number of single objective problems.

Techbycheff Approach in MOEA/D

In this approach, the scalar optimization problems are in the form:

$$\text{minimize} \quad g^{te}(x|\lambda, z^*) = \max_{1 \le i \le m}\{\lambda_i |f_i(x) - z_i^*|\}$$
$$\text{subject to} \quad x \in \Omega$$

where m is the number of objectives. λ is a set of well-spread weight vectors ($\lambda^1 \ldots \lambda^N$) and z_i^* is the reference/best fitness of each objective. The problem of approximation of the PF is decomposed into N scalar optimization subproblems. Solve these N subproblems one by one. The distribution of final solutions could be uniform if g( ) and λ are properly chosen.

Neighborhood relationship in MOEA/D

- **These sub-problems are related to each other.**
  If λ^i and λ^j are close, $g(x,\lambda^i)$ and $g(x,\lambda^j)$ are neighbors. Neighboring problems should have similar solutions.
- **N agents are used for solving these N subproblems.**

X_1	X_2	…	X^N
Min $g(x,\lambda^1)$	Min $g(x,\lambda^2)$	…	Min $g(x,\lambda^N)$

- **During the search, neighboring agents can help each other.**

MOEA/D framework

- Agent *i* records x^i, the best solution found so far for this particular subproblem.
- At each generation, each agent *i* does the following:
- ✓ Select several neighbors and obtain their best solutions.
- ✓ Apply genetic operators (mutation & crossover in MOEA/D-DE) on these selected solutions and generate a new solution y'.
- ✓ Apply single objective local search on y' to optimize its objective $g(x,\lambda^i)$ and obtain y.
- ✓ Replace x^i by y if $g(y,\lambda^i)<g(x^i,\lambda^i)$.
- ✓ If not replaced, let one of its neighbors replace their best solutions by y if y is better than their current best solutions (measured by their individual objectives).

61 NANYANG TECHNOLOGICAL UNIVERSITY

ENS-MOECA/D

- For effective performance of MOEA/D, neighborhood size (NS) parameter has to be tuned.
- A large NS promotes collaboration among dissimilar subproblems, which advances the information exchange among the diverse subproblems, and thus it speeds up the convergence of the whole population; while a small NS encourages combination of similar solutions and is good for local search in a local neighborhood, which maintains the diversity for the whole population.
- However, in some cases, a large NS can also benefit on diversity recovery; while a small NS is also able to facilitate the convergence.
- For instance, during the evolution, some subproblems may get trapped in a locally optimal regions. In order to force those subproblems escape from the premature convergence, a large NS is required for the exploration. On the other hand, if the global optima area is already found, a small NS will be favorable for local exploitation.

62 NANYANG TECHNOLOGICAL UNIVERSITY

ENS-MOECA/D

Neighborhood sizes (NS) are a crucial control parameter in MOEA/D.

An ensemble of different neighborhood sizes (NSs) with online self-adaptation is proposed (ENS-MOEA/D) to overcome the difficulties such as

1) tuning the numerical values of NS for different problems;

2) specifications of appropriate NS over different evolution stages when solving a single problem.

S. Z. Zhao, P. N. Suganthan, Q. Zhang, "Decomposition Based Multiobjective Evolutionary Algorithm with an Ensemble of Neighborhood Sizes", *IEEE Trans. on Evolutionary Computation,* **DOI:** 10.1109/TEVC.2011.2166159, 2012.

63 NANYANG TECHNOLOGICAL UNIVERSITY

ENS-MOECA/D

In ENS-MOEA/D, *K* fixed neighborhood sizes (NSs) are used as a pool of candidates. During the evolution, a neighborhood size will be chosen for each subproblem from the pool based on the candidates' previous performances of generating improved solutions. In ENS-MOEA/D, the certain fixed number of previous generations used to store the success probability is defined as the Learning Period (LP). At the generation G>LP-1 , the probability of choosing the kth (*k = 1, 2, …, K*) NS is updated by:

$$p_{k,G} = \frac{R_{k,G}}{\sum_{k=1}^{K} R_{k,G}}$$

$$R_{k,G} = \frac{\sum_{g=G-LP}^{G-1} FEs\_success_{k,g}}{\sum_{g=G-LP}^{G-1} FEs_{k,g}} + \varepsilon, \quad (k=1,2,..,K;\ G>LP)$$

64 NANYANG TECHNOLOGICAL UNIVERSITY

ENS-MOECA/D Experimental results

- ENS-MOEA/D is tested on the 10 unconstrained test instances in CEC 2009 MOEA Competition which includes two and three objective problems (Latest benchmark on MO problems).
- The IGD performance measure is used as in the CEC 2009 MOEA competition.
- The four different NSs for the two-objective problems are 30, 60, 90 and 120, where NS=60 is the original parameter setting in the MOEA/D in the NSs for the three-objective problems are 60, 80, 100, 120 and 140, where 100 is the original parameter setting for NS in the MOEA/D.

65

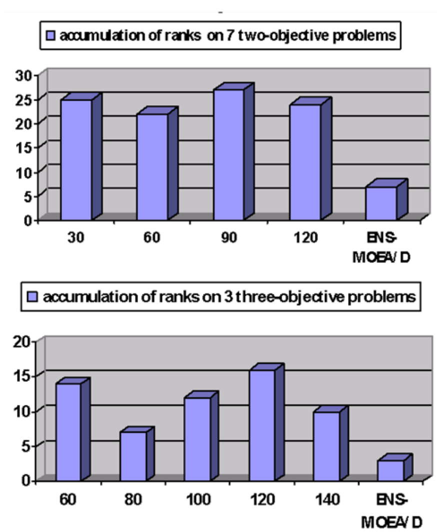
NANYANG TECHNOLOGICAL UNIVERSITY

ENS-MOECA/D Experimental results

- We conducted a parameter sensitivity investigation of *LP* for ENS-MOEA/D using four different values (10, 25, 50 and 75) on the 10 benchmark instances. By observing the mean of IGD values over 25 runs we can conclude that the *LP* is not so sensitive to most of the benchmark functions, and it is set as *LP*=50.
- The mean of IGD values over 25 runs among all the variants of MOEA/D with different fixed *NS* and ENS-MOEA/D are ranked. Smaller ranks, better performance.

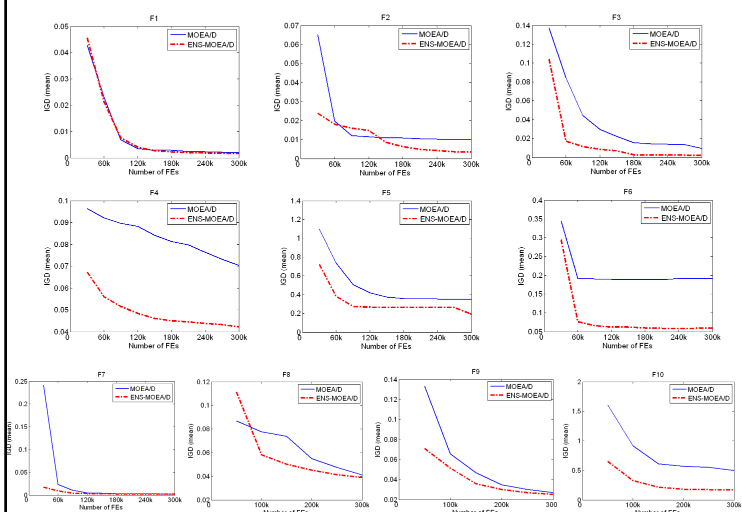
66

NANYANG TECHNOLOGICAL UNIVERSITY



67

NANYANG TECHNOLOGICAL UNIVERSITY



UNIVERSITY

MODE based summation of normalized objective value and diversified selection

Step 1	For $m = 1$ to M (*number_of_objectives*)
Step 2	Find the maximum and minimum objective values of the m^{th} objective and calculate the range of the m^{th} objective.
Step 3	Normalize the m^{th} objective values of all members using the equation below: $$f_m^{'}(x) = \frac{f_m(x) - f_{\min}}{f_{\max} - f_{\min}}$$ where $f_m^{'}$ is the normalized m^{th} objective value.
Step 4	Endfor
Step 5	For $i = 1$ to NP (*population_size*)
Step 6	Sum all normalized objective values of the member to obtain a single value.
Step 7	Endfor

B. Y. Qu, P. N. Suganthan, "Multi-Objective Evolutionary Algorithms based on the Summation of Normalized Objectives and Diversified Selection", *Information Sciences,*180(17):3170-3181.

B. Y. Qu and P. N. Suganthan, "Multi-objective differential evolution based on the summation of normalized objectives and improved selection method," SDE-2011 *IEEE Symposium on Differential Evolution*, Paris, France, April 2011

NANYANG TECHNOLOGICAL UNIVERSITY

Diversified selection

TABLE III. THE STEPS OF REMOVING BAD INDIVIDUALS

Step 1	Indentify the reference point using equation 1.
Step 2	Find the closest individual in population to the reference point and set it as reference individual.
Step 3	Remove all the individuals that dominated by the reference individual.

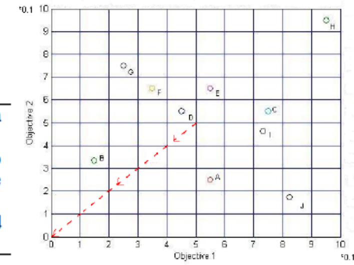


Fig. 2 Pre-selection for two objectives

Step 1	For $m = 1$ to M (*number_of_objectives*)
Step 2	(i) Evenly divide the range of the objective space into 100 bins.
Step 3	(ii) Scan P percentage of the 100 bins (*i.e.* from bin 1 to P, P may be chosen as 80 or 90).
Step 4	(iii) For each scanned bin (if this bin is empty, otherwise just continue to next bin), the solution with the smallest summation of normalized objective values will be chosen to enter preferential set.
Step 5	End For
Step 6	Accumulate the solutions excluded from the preferential set and store them in backup set.

Use solutions in the preferential set to generate offspring.

70

NANYANG TECHNOLOGICAL UNIVERSITY

Overview

I. Introduction

II. Some DE variants for Single Objective Optimization

III. Multimodal Optimization in DE

IV. Multiobjective Optimization

V. Large Scale Optimization

VI. Dynamic Optimization

VII. Constrained Optimization

71

NANYANG TECHNOLOGICAL UNIVERSITY

Outline

- **Large Scale Optimization (Based on Soft Computing special issue)**
- **LSO algorithms**
  - Self-adaptive Differential Evolution with Modified Multi-Trajectory Search (SaDE-MMTS)
  - MOS-based Dynamic Memetic Differential Evolution (MOS) (multiple offspring sampling)
  - Self-adaptive Differential Evolution Algorithm using Population Size Reduction and Three Strategies (jDElscop)
- **Experiments Results**

NANYANG TECHNOLOGICAL UNIVERSITY

72

Large Scale Optimization

- Optimization algorithms perform differently when solving different optimization problems due to their distinct characteristics. Most optimization algorithms lose their efficacy when solving high dimensional problems. Two main difficulties are:
  - The high demand on exploration capabilities of the optimization methods. When the solution space of a problem increases exponentially with increasing dimensions, more efficient search strategies are required to explore all promising regions within a given time budget.
  - The complexity of a problem characteristics may increase with increasing dimensionality, e.g. unimodality in lower dimensions may become multi-modality in higher dimensions for some problems (e.g. Rosenbrock's)

Large Scale Optimization

- Due to these reasons, a successful search strategy in lower dimensions may no longer be capable of finding good solutions in higher dimension.
- Three LSO algorithms based on DE with the best performance are presented – MOS, jDElscop and SaDE-MMTS
- From the special issue of the Soft Computing Journal on Scalability of Evolutionary Algorithms and other Meta-heuristics for Large Scale Continuous Optimization Problems.

SaDE-MMTS – Two levels of self-adaptation

❑SaDE benefits from the self-adaptation of trial vector generation strategies and control parameter adaptation schemes by learning from their previous experiences to suit different characteristic of the problems and different search requirements of evolution phases.

❑ Every generation, a selection among the JADE mutation strategy with two basic crossover operators (binomial crossover and exponential crossover) as well as no crossover option is also adaptively determined for each DE population member based on the previous search experiences.

SaDE-MMTS – Two levels of self-adaptation

Low Level Self-adaptation in MMTS:

❑An adaptation approach is proposed to adaptively determine the initial step size parameter used in the MMTS. In each MMTS phase, the average of all mutual dimension-wise distances between current population members (AveDis) is calculated, one of the five linearly reducing factors (LRF) from 1 to 0.1, 5 to 0.1, 10 to 0.1, 20 to 0.1 and 40 to 0.1 is selected based on the performance, and this LRF is applied to scale AveDis over the evolution.

SaDE-MMTS

High Level Self-adaptation between SaDE & MMTS:

- ❑The MMTS is used periodically for a certain number of function evaluations along with SaDE, which is determined by an adaptation way.
- ❑At the beginning of optimization procedure, the SaDE and the MMTS are firstly conducted sequentially within one search cycle by using the same number of function evaluations. Then the success rates of both SaDE and MMTS are calculated. Subsequently, function evaluations are assigned to SaDE and MMTS in each search cycle proportional to the success rates of both search methods.

MOS -- High-level Relay Hybrid (HRH)

Algorithm 2 HRH MOS Algorithm

1: Create initial overall population of candidate solutions P_0
2: Uniformly distribute participation among the n used techniques $\rightarrow \forall j \; \Pi_0^{(j)} = \frac{FEs_0}{n}$. Each technique produces a subset of individuals according to its participation $(\Pi_0^{(j)})$
3: Evaluate initial population P_0
4: **while** number of steps not exceeded **do**
5: Update Quality of $T^{(j)}$ computed as the average quality of all the individuals created by technique $T^{(j)}$ in the previous step
6: Update participation ratios from Quality values computed in Step 5 $\rightarrow \forall j \; \Pi_{i+1}^{(j)} = PF(Q_i^{(j)})$
7: Update FEs allocated for each technique at this step: $\rightarrow \forall j \; FEs_i^{(j)} = \Pi_{i+1}^{(j)} \cdot FEs_i$
8: **for** every available technique $T^{(j)}$ **do**
9: **while** $FEs_i^{(j)}$ not exceeded **do**
10: Evolve
11: **end while**
12: **end for**
13: **end while**

jDElscop

{*pop* ... population}
{*imin* ... index of currently best individual}
{$\mathbf{x}_i$ ... i-th individual of population}
{*MAX_FEs* ... maximum number of function evaluations}
{*s* ... one of strategy (*jDEbin*, *jDEexp*, *jDEbest*)}
Initialization() {Generate uniformly distributed random population}
for ($it = 0$; $it < MAX\_FEs$; $it++$) **do**
  $i = it$ **mod** NP {mod ... modulo operation}
  if ($rand(0,1) < 0.1$ **and** ($it > \frac{MAX\_FEs}{2}$)) **then**
    $s = 1$; {*jDEbest* strategy}
  else
    if ($i < \frac{NP}{2}$) **then**
      $s = 2$; {*jDEbin* strategy}
    else
      $s = 3$; {*jDEexp* strategy}
    end if
  end if
  {Perform one iteration of the jDE using strategy s on $\mathbf{x}_i^{(G)}$}
  {The jDE applies mutation and crossover to generate trial vector $\mathbf{u}^{(G)}$}
  $\mathbf{u}^{(G)}$ =jDE(i, pop, s)
  {Fitness evaluation and selection}
  if $(f(\mathbf{u}^{(G)}) \leq f(\mathbf{x}_i^{(G)}))$ **then**
    $\mathbf{x}_i^{(G+1)} = \mathbf{u}^{(G)}$
  end if
end for

jDElscop

{*pop* ... population of *NP* individuals}
{$\mathbf{c}$ ... vector of individuals' fitness values}
{k_1, k_2, m ... temporary variables}
for ($i = 0$; $i < \frac{NP}{4}$; $i++$) **do**
  $k_2 = \frac{NP}{4} + i$
  if $\mathbf{c}[i] < \mathbf{c}[k_2]$ **then**
    $pop[i] = pop[k_2]$
    $\mathbf{c}[i] = \mathbf{c}[k_2]$
  end if
end for
for ($i = 0$; $i < \frac{NP}{4}$; $i++$) **do**
  $k_1 = \frac{NP}{2} + i$
  $k_2 = \frac{3NP}{4} + i$
  $m = \frac{NP}{4} + i$
  if $\mathbf{c}[k_1] < \mathbf{c}[k_2]$ **then**
    $pop[m] = pop[k_1]$
    $\mathbf{c}[m] = \mathbf{c}[k_1]$
  else
    $pop[m] = pop[k_2]$
    $\mathbf{c}[m] = \mathbf{c}[k_2]$
  end if
end for
$NP = \lceil \frac{NP}{2} \rceil$

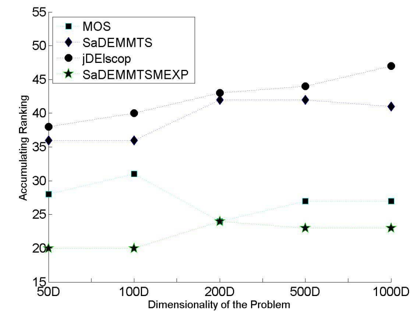
Experiments Results

- Algorithms were tested on 19 benchmark functions prepared for a Special Issue on Scalability of Evolutionary Algorithms and other Metaheuristics for Large Scale Continuous Optimization Problems. (http://sci2s.ugr.es/eamhco/CFP.php)
- The benchmark functions are scalable. The dimensions of functions were 50, 100, 200, 500, and 1000, respectively, and 25 runs of an algorithm were needed for each function.
- The optimal solution results, f (x), were known for all benchmark functions.

81 NANYANG TECHNOLOGICAL UNIVERSITY

SWEVO S-EXP

➤ Comparisons based on summation of ranks among four methods. SaDE-MMTS-S-EXP performs the best overall.



S. Z. Zhao, P N Suganthan, "Empirical Investigations into the Exponential Crossover of Differential Evolution." *Swarm and Evolutionary Computation, Vol. 9, April 2013, pp. 27–36.*

82 NANYANG TECHNOLOGICAL UNIVERSITY

Conclusions on LSO

- It is worth to mention here, in DE variants, the *Exponential Crossover* shows the superior performance over *Binomial* Crossover on the LSO problems with non-separable property. In contrast, Bin-xover is performing better than Exp-xover on separable and low dimensional problems.
- Rule of thumbs for population size?
- Local search methods are significant in all the three LSO algorithms mentioned above. One or more appropriate LS can be investigated and included into the existing algorithms.
- jDElscop does not use a distinct local search, instead it uses the DE mutation operator with the best solution as the base vector. This is better than introducing distinct LS methods.

83 NANYANG TECHNOLOGICAL UNIVERSITY

Overview

I. Introduction

II. Some DE variants for Single Objective Optimization

III. Multimodal Optimization in DE

IV. Multiobjective Optimization

V. Large Scale Optimization

VI. Dynamic Optimization

VII. Constrained Optimization

84 NANYANG TECHNOLOGICAL UNIVERSITY

Dynamic Optimization Problems (DOPs)

- Many real-life applications are transient in time. E.g. stock market, satellite array beam-forming, adaptive filter coefficients.
- Fitness evaluation in Dynamic Optimization Problems (DOPs) is subject to time

$$F = f(\bar{x}, t), \bar{x} = [x_1, x_2, \dots x_D]$$

- Population-based optimization looses population diversity as popln converges to global solution(s), but in DOPs, the true optima may shift after convergence.

85 NANYANG TECHNOLOGICAL UNIVERSITY

Dynamic Optimization Problems (DOPs)

Techniques for Handling Dynamic objectives in Evolutionary Algorithms (EAs)

1. Generate Diversity after change – Re-evaluate some members for change after every generation, and reinitialize whole population upon change detection.
2. Maintain Diversity throughout search – Apply different updating strategies for individuals in population, e.g. quantum/Brownian individuals
3. Memory-based approaches – Archive past population information for future re-insertion (Useful for oscillating optima)
4. Multiple-population approaches – Conduct multiple concurrent searches using multiple sub-populations thereby maintaining diversity.

86 NANYANG TECHNOLOGICAL UNIVERSITY

Dynamism Handling

1. Multi-population approaches –
   a) Multiple sub-populations
   b) Favored population: more individuals allocated to most promising sub-population (i.e. fittest local optima)
   c) Exclusion principle (exclude overlapping search by more than one sub-population)
2. Maintaining Diversity throughout run –
   a) Quantum Individuals
   b) Brownian Individuals
   c) Entropic Individuals
3. Memory-based approaches –
   a) Aged Individuals or stagnated individuals to be reinitialized
   b) Memory can be used to copy past good solutions in the memory back into the evolving population.

87 NANYANG TECHNOLOGICAL UNIVERSITY

Ensemble Differential Evolution with Dynamic Subpopulations and Adaptive Clearing (EDESAC)

- Motivation
  - Diversity maintenance to prevent stagnation
  - Self-adaptation strategy for unpredictable dynamic change
  - Memory technique – good for recurrent /periodic changes
  - Greedy mutation strategy for exploitation
- Implementation
  - Multi-population approach using Ensemble of mutation, crossover strategies & parameter values (EPSDE)
  - Gbestpop with greedy tournament mutation strategy grows over fixed FEs; shift from exploration to exploitation
  - Adaptive clearing of past good solutions and local optima in archive

S. Hui, P. N. Suganthan, “Ensemble Differential Evolution with Dynamic Subpopulations and Adaptive Clearing for solving Dynamic Optimization Problems”, *IEEE Congress on Evolutionary Computation*, Brisbane, Australia, June 2012.

88 NANYANG TECHNOLOGICAL UNIVERSITY

Ensemble Differential Evolution with Dynamic Subpopulations and Adaptive Clearing (EDESAC)

1) Total pop size (N) = 80, sub-pop size (L_size) = 10
2) Sub-pop EPSDE Mutation Strategies:
   a) DE/current-to-pbest/2
   $$v_i^G = x_i^G + (1 - F_i)(x_{pbest}^G - x_i^G) + F_i\,(x_{rand1,i}^G - x_{rand2,i}^G)$$
   b) DE/rand/2
   $$v_i^G = x_{rand1,i}^G + F_i\,(x_{rand2,i}^G - x_{rand3,i}^G + x_{rand4,i}^G - x_{rand5,i}^G)$$
   F_i =[0.3, 0.7, 0.9], C_i =[0.1, 0.5, 0.9], CR methods = [*Bin*, *Exp*]
3) *Gbestpop* chosen at 30000 FEs (max FEs before change = 100000)
   - Greedy tournament mutation strategy:
     $$v_i^G = x_{tourn1,i}^G + F_i\,(x_{tourn2,i}^G - x_{tourn3,i}^G)$$
     where $f(x_{tourn1}) > f(x_{tourn2}) > f(x_{tourn3})$ in maximization problem
   - ***for k=1 to 3, (r1, r2*** = random index without replacement)
     $$x_{tourn\ k} \begin{cases} x_{r1\ k}, & \text{if } f(x_{r1\ k}) > f(x_{r2\ k}) \\ = x_{r2\ k}, & \text{otherwise} \end{cases}$$
   - *Gbestpop* size incremented every 3000 Fes
   - Worst performing sub-pop eliminated when total pop size ≥ $N + L\_size$

89

NANYANG TECHNOLOGICAL UNIVERSITY

Ensemble Differential Evolution with Dynamic Subpopulations and Adaptive Clearing (EDESAC)

❖ Aging applied on local best in sub-pop to identify local optima

if $|f(u_i^G) - f(x_i^G)| < 0.1$ or $\| u_i^G - x_i^G \| < 0.01$
  $age_i = max(age_i, 20)$
else $age_i = min(age_i, 5)$ *end*
if $i = lbest$ & $age_i > 30$ & $i \neq gbest$
  $x_i^G \rightarrow archive$
  $f(x_i^G) \rightarrow archive\_fit$
end

❖ Adaptive clearing radius (CL_R) performed on archive

Rank archive $=[a_1, a_2 \ldots, a_P]$, *such that* $f(a_i) < f(a_{i+1})$, *for* $i=1,2\ldots P-1$
if $P<5$ & $rand()<0.1$, $CL\_R=0.5$
else if $5<P<20$ & $rand()<0.1$, $CL\_R=0.1$
else $CL\_R=1$
for $i=1$ *to* $P-1$
  for $k=2$ *to* P
    if $\| a_i - a_k \| < CL\_R$, *remove* a_k *from archive*
  end
end

❖ Archive used for reinitialization when change detected

90

NANYANG TECHNOLOGICAL UNIVERSITY

Dynamic DE (DynDE)

➢ Multi-population DE approach to DOPs
  - Locate multiple prospective solutions on different peaks
  - Ensure sub-populations do not converge onto same local peak (Exclusion)
  - Maintain local population diversity with special updating individuals: Quantum, Browning and Entropy Individuals

➢ Special Updating Scheme
  1. Quantum Individuals – generated within quantum cloud of radius r_{cloud}
     a) Generate an individual normally at random, $x_i \sim N[0,1], 1 \le i \le d$
     b) Compute distance to origin, $dist = \sqrt{\sum_{i=1}^{i=d} x_i^2}$
     c) Update the individual around global optima $\bar{g}_b$,
     $$x' = \bar{g}_b + \frac{r x_i}{dist}, r = U[0, r_{cloud}]$$

91

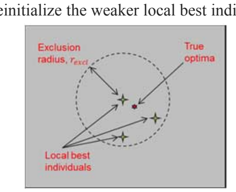
R. Mendes, A. S. Mohais, "DynDE: a Differential Evolution for Dynamic Optimization Problems", Proc. CEC 2005.

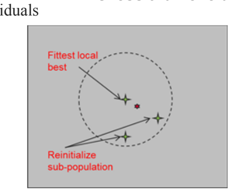
NANYANG TECHNOLOGICAL UNIVERSITY

Dynamic DE (DynDE)

2. Brownian Individuals – generated about the global best with a normally distributed random displacement with standard deviation of σ
   $$x' = \bar{g}_b + \bar{N}(0, \sigma)$$
3. Entropic Individuals – similar to Brownian except entropy is added to previous generation of entropic individuals instead of global best
   $$x' = \bar{x} + \bar{N}(0, \sigma)$$

➢ Exclusion - prevent local populations from converging on same peak
  a) Compute displacement between the best individuals of all sub-populations
  b) If Euclidean distance between 2 or more local best **is less than exclusion radius** reinitialize the weaker local best individuals





92

NANYANG TECHNOLOGICAL UNIVERSITY

Dynamic DE (DynDE)

- Favored Population – eliminate functional evaluation of sub-optimal populations
  a) Perform normal evolution on all populations to identify peaks
  b) Halt evolution on weaker populations, even migrating some individuals from weaker populations to stronger populations to enhance search for global optima
  c) Return to normal operation when change detected
- Algorithm flow
  I. Initialize multiple populations, identify at random special updating individuals (Quantum/Brownian/Entropic)
  II. With the exception of special individuals, evolve all individuals using normal DE strategies
  III. Update special individuals around their local best
  IV. Perform Exclusion on overlapping populations
  V. Over generations, migrate individuals to better populations
  VI. Check for environmental change by re-evaluating best individual(s), if change detected, restore all population sizes
  VII. Repeat Step II to VI until stop criterion

M. C. du Plessis and A. P. Engelbrecht, "Improved Differential Evolution for Dynamic Optimization Problems", *Proc. of the 2008 Congress on Evolutionary Computation*, pp. 229 – 234, 2008.

NANYANG TECHNOLOGICAL UNIVERSITY

Self-Adaptive DE (jDE)

- Multi-population approach with self-adapting parameter control
  - Aging of all individuals except global optima to escape local optima
  - Archiving of past global best individuals for future re-initialization
- Self-adaptive control parameters *F* and *CR*

$$F_i^{G+1} = \begin{cases} F_i^G, U(0,1) \geq \tau_1 \\ F_L + U(0,1).F_H, otherwise \end{cases}, F_L \leq F_i \leq F_H$$

$$CR_i^{G+1} = \begin{cases} CR_i^G, U(0,1) \geq \tau_2 \\ U(0,1), otherwise \end{cases}$$

where$\tau_1, \tau_2$are probabilities for controlling *F* and *CR*
Also to maintain a degree of diversity, the following must be satisfied

$$2F^2 - \frac{2}{NP} + \frac{CR}{NP} = 0$$

[1] J. Brest, A. Zamuda, B. Boskovic, M. S. Maucec and V. Zumer, "Dynamic Optimization using Self-Adaptive Differential Evolution", *Proc. of the 2009 Congress on Evolutionary Computation*, pp. 415 – 422, 2009.
[2] D. Zaharie, "Critical Values for the Control Parameters of Differential Evolution Algorithms", Proc of Mendel 2002, 8th Int. Conf. on Soft Computing, 2002, pp. 62- 67.

NANYANG TECHNOLOGICAL UNIVERSITY

Self-Adaptive DE (jDE)

- Aging strategy of individuals

$$\begin{gathered} \boldsymbol{if}\ x_i = g_{best}, \\ age_i = 0 \\ \boldsymbol{else\ if}\ (x_i = l_{best})\&(age_i > 30)\&(rand < 0.1), \\ reinitialize\ subpop\ that\ contains\ x_i \end{gathered}$$

- Overlapping search (Exclusion) – reinitialize sub-pop whose local best is too close to another local best
- Maintain local diversity – reinitialize individuals that are too close to local best.
- Age of individual is incremented every generation. Fitness and diversity control.

$$\begin{aligned} &\boldsymbol{if}\ \|x_i - x_j\| < 0.01\ \boldsymbol{OR}\ \|f(x_i) - f(x_j)\| < 0.1, \\ &age_i = min\{age_i, 20\} \\ &\boldsymbol{else}\ age_i = min\{age_i, 5\} \end{aligned}$$

- Archiving of past global best after every change detection. Whenever re-initialization occurs, individuals have chances to be either taken from archive or be randomly generated

NANYANG TECHNOLOGICAL UNIVERSITY

Overview

I. Introduction

II. Some DE variants for Single Objective Optimization

III. Multimodal Optimization in DE

IV. Multiobjective Optimization

V. Large Scale Optimization

VI. Dynamic Optimization

VII. Constrained Optimization

NANYANG TECHNOLOGICAL UNIVERSITY

Constraint Handling in DE

- Constraints reduce feasible region – complicates search process
- Evolutionary algorithms (EAs) perform unconstrained search
- EAs require additional mechanisms to handle constraints
- Minimize: $f(X) \quad X=(x_1,x_2,\ldots,x_D)$ and $X \in S$

subject to

$$g_i(X) \le 0, \qquad i = 1,\ldots, p$$
$$h_j(X) = 0, \qquad j = p+1,\ldots, m$$

where S search space, m is the total number of constraints, p is no. of inequality constraints.

97

NANYANG TECHNOLOGICAL UNIVERSITY

Constraint Handling in DE

- Equality constraints $\longrightarrow$ inequality and combined with other inequality constraints as

$$G_i(X) = \begin{cases} \max\{g_i(X),0\} & i = 1,\ldots p. \\ \max\{|h_i(X)| - \delta, 0\} & i = p+1,\ldots,m \end{cases}$$

δ - tolerance parameter for equality constraints

- Overall constraint violation - weighted mean of all the constraints $G_i(X)$

$$\upsilon(X) = \frac{\sum_{i=1}^{m} w_i(G_i(X))}{\sum_{i=1}^{m} w_i}$$

$w_i(=1/G_{\max_i})$ - weight parameter $\quad G_{\max_i}$ - maximum violation of $G_i(X)$

- The objective transforms to minimize $f(X)$ such that $\upsilon(X) = 0$

98

NANYANG TECHNOLOGICAL UNIVERSITY

Constraint Handling in DE

- Handling infeasible solutions
  - Discard infeasible solutions (some potential information may be lost)
  - Exploit the information present in infeasible solutions
- Constraint Handling Techniques are grouped
  - preserving feasibility of solutions
  - penalty functions
  - make a separation between feasible and infeasible solutions
  - hybrid methods
  - multi-objective approach (include non-domination sorting)

99

NANYANG TECHNOLOGICAL UNIVERSITY

Constraint Handling in DE

- Superiority of Feasible (SF)

  Among X_i and X_j, X_i is regarded superior to X_j if :

  - Both infeasible & $\upsilon(X_i) < \upsilon(X_j)$

    push infeasible solutions to feasible region

  - Both feasible & $f(X_i) < f(X_j)$ (minimization problems)

    improves overall solution

  - X_i - feasible & X_j – infeasible

100

NANYANG TECHNOLOGICAL UNIVERSITY

Constraint Handling in DE

- Self-adaptive Penalty

$$F(X)=d(X)+p(X)$$

$$d(X)=\begin{cases}\upsilon(X), & \text{if } r_f=0\\ \sqrt{f''(X)^2+\upsilon(X)^2}, & \text{otherwise}\end{cases} \qquad p(X)=(1-r_f)M(X)+r_f N(X)$$

$$f''(X)=\frac{f(X)-f_{\min}}{f_{\max}-f_{\min}} \qquad f_{\min}, f_{\max} \text{ - min. \& max. of } f(X) \text{ irrespective of feasibility} \qquad r_f=\frac{\#\text{ feasible individuals}}{\text{population size}}$$

$$M(X)=\begin{cases}0, & \text{if } r_f=0\\ \upsilon(X), & \text{otherwise}\end{cases} \qquad N(X)=\begin{cases}0, & \text{if } X \text{ is feasible}\\ f''(X), & \text{if } X \text{ is infeasible}\end{cases}$$

- Amount of penalties - controlled by # of feasible individuals present
- Few feasible – high penalty added to infeasible individuals with high constraint violation.
- More feasible – high penalty added to feasible individuals with high objective values
- Switch from requiring feasible solutions to feasible & better solutions

Constraint Handling in DE

- Epsilon Constraint (EC)
  - Relaxation of constraints is controlled by ε parameter
  - High quality solutions for problems with equality constraints

$$\varepsilon(0)=\upsilon(X_\theta)$$

X_θ - top θ th individual in initial population (sorted w. r. t. υ)

$$\varepsilon(k)=\begin{cases}\varepsilon(0)\left(1-\frac{k}{T_c}\right)^{cp}, & 0<k<T_c\\ 0, & k\ge T_c\end{cases}$$

  - The recommended parameter settings are

$$T_c\in[0.1T_{\max}, 0.8T_{\max}] \qquad cp\in[2,10]$$

Constraint Handling in DE

- Stochastic Ranking (SR)
  - Balances between objective and υ stochastically (low computational cost)

Basic form of SR

If (no constraint violation or rand < p_f)

Rank based on the objective value only

else

Rank based on the constraint violation only

end

ECHT - DE

- No free lunch theorem (NFL)
- Each constrained problem is unique

(feasible /search space, multi-modality and nature of constraint functions)

- Evolutionary algorithms are stochastic in nature.

(same problem & algorithm – diff. constraint handling methods - evolution paths can be diff.)

- Diff. stages– Diff. constraint handling methods effective

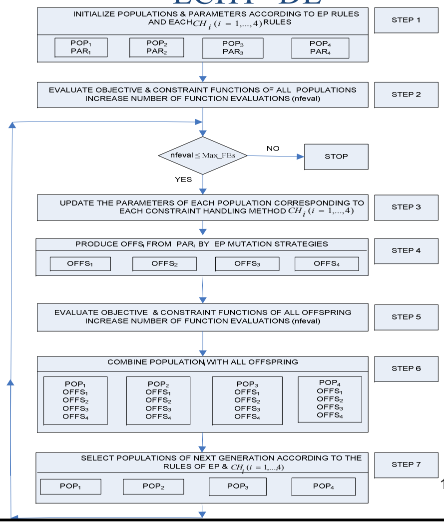
(feasible/ search space, multi-modality, nature of constraints, chosen EA)

- To solve a particular problem - numerous trial-and-error runs

(suitable constraint handling technique and to fine tune associated parameters)

R. Mallipeddi and P. N. Suganthan, Ensemble of Constraint Handling Techniques, *IEEE Transactions on Evolutionary Computation,* Vol. 14, No. 4, pp.561-579, Aug. 2010.

ECHT- DE



105

NANYANG TECHNOLOGICAL UNIVERSITY

ECHT-DE

- Efficient usage of function calls

  (evaluation of objective / constraint functions is computationally expensive)
- Offspring of best suited constraint handling technique survive

  (For a search method and problem during a point in the search process, the offspring population produced by the population of the best suited constraint handling method dominates and enters other populations. In subsequent generations, these superior offspring will become parents in other populations too)
- Performance of ECHT can be improved by selecting diverse and competitive constraint handling methods

  (If the constraint handling methods in ensemble are similar in nature, associated populations may lose diversity and the search ability of ECHT may be deteriorated)

106

NANYANG TECHNOLOGICAL UNIVERSITY

ECHT - DE

- 24 well known benchmark functions (CEC2006)
- Population size (NP)

  single cases: ACEP – 200, DE – 50

  ECHT (each constraint handling method) - 50
- Each algorithm is run 30 times.
- Total function evaluations - 240,000
- ECHT1 uses the same parameters as in single constraint handling method

  $$T_c = 0.5T_{\max}, c_p = 5 \text{ and } p_f = 0.475 \text{ to } 0.025$$
- ECHT2 uses tuned parameters

  $$T_c = 0.8\,T_{\max} \text{ and } c_p = 2$$

107

NANYANG TECHNOLOGICAL UNIVERSITY

ECHT - DE

Algorithm	Rank	Average Rank
SF-DE	40	1.82
SP-DE	72	3.27
EC-DE	25	1.14
SR-DE	80	3.64
ECHT-DE1	25	1.14
ECHT-DE2	22	1.00

- ECHT-DE1 is superior, equal and worst in 35, 57 and 0 cases
- ECHT-DE1 is always better or equal
- To show clear advantage of ECHT - New set of problems

108

NANYANG TECHNOLOGICAL UNIVERSITY

ECHT-DE (Special Session CEC 2010)

Problem/Search Range	Type of Objective	Number of Constraints		Feasibility Region (ρ)	
		E	*I*	10D	30D
C01 $[0,10]^D$	Non Separable	0	2 Non Separable	0.997689	1.000000
C02 $[-5.12,5.12]^D$	Separable	1 Separable	2 Separable	0.000000	0.000000
C03 $[-1000,1000]^D$	Non Separable	1 Non Separable	0	0.000000	0.000000
C04 $[-50,50]^D$	Separable	4 2 Non Separable, 2 Separable	0	0.000000	0.000000
C05 $[-600,600]^D$	Separable	2 Separable	0	0.000000	0.000000
C06 $[-600,600]^D$	Separable	2 Rotated	0	0.000000	0.000000
C07 $[-140,140]^D$	Non Separable	0	1 Separable	0.505123	0.503725
C08 $[-140,140]^D$	Non Separable	0	1 Rotated	0.379512	0.375278
C09 $[-500500]^D$	Non Separable	1 Separable	0	0.000000	0.000000

R. Mallipeddi and P. N. Suganthan, Problem Definitions and Evaluation Criteria for the CEC 2010 Competition on Constrained Real-Parameter Optimization, *Technical Report*, Barcelona, Spain.

ECHT-DE (Special Session CEC 2010)

Problem/Search Range	Type of Objective	Number of Constraints		Feasibility Region (ρ)	
		E	*I*	10D	30D
C10 $[-500,500]^D$	Non Separable	1 Rotated	0	0.000000	0.000000
C11 $[-100,100]^D$	Rotated	1 Non Separable	0	0.000000	0.000000
C12 $[-1000,1000]^D$	Separable	1 Non Separable	1 Separable	0.000000	0.000000
C13 $[-500,500]^D$	Separable	0	3 2 Separable, 1 Non Separable	0.000000	0.000000
C14 $[-1000,1000]^D$	Non Separable	0	3 Separable	0.003112	0.006123
C15 $[-1000,1000]^D$	Non Separable	0	3 Rotated	0.003210	0.006023
C16 $[-10,10]^D$	Non Separable	2 Separable	2 1 Separable, 1 Non Separable	0.000000	0.000000
C17 $[-10,10]^D$	Non Separable	1 Separable	2 Non Separable	0.000000	0.000000
C18 $[-50,50]^D$	Non Separable	1 Separable	1 Separable	0.000010	0.000000

ECHT-DE (Special Session CEC 2010)

- D is the number of decision variables
- $\rho = |F|/|S|$ is the estimated ratio between the feasible region and the search space
- I is the number of inequality constraints
- E is the number of equality constraints
- Runs/problem: 25
- Max_FES : 200000 for 10D and 600000 for 30D
- Feasible Run: A run during which at least one feasible solution is found within Max FES.
- Feasible Rate = (# of feasible runs) / Total runs.
- The above quantity is computed for each problem separately

ECHT-DE (Special Session CEC 2010)

- Ranking is given to each algorithm on every problem based on the following criteria
  1. Algorithms giving 100% feasibility rate are ranked based on mean value of the 25 runs
  2. Algorithms having feasibility rate in the range >0% - <100% are ranked based on feasibility rate.
  3. Algorithms with 0% feasibility rate are ranked based on overall violation (normalized)
- Finally we add all the ranks of a particular algorithm over all problems to get the total rank
- Average Rank = Total rank/36

ECHT-DE (Special Session CEC 2010)

- jDEsoco — Janez Brest, *et al* (An Improved Self-adaptive Differential …)
- DE-VPS — M. Fatih Tasgetiren, *eta al* (An Ensemble of Differential …)
- RGA — Amit Saha, *et al* (Hybrid Gradient Projection Genetic …)
- E-ABC — Efren Mezura Montes, *et al* (Elitist Artificial Bee Colony…)
- εDEg — Tetsuyuki Takahama & Setsuko Sakai (Constrained …)
- DCDE — Zhihui Li, *et al* (Differential Evolution with Dynamic …)
- Co-CLPSO — J. J. Liang, *et al* (Coevolutionary Comprehensive Learning …)
- CDEb6e6rl — Josef Tvrdik & Radka Polakova (Competitive Differential …)
- sp-MODE — Gilberto Reynoso-Meza *et al* (Multiobjective optimization …)
- MTS — Lin-Yu Tseng and Chun Chen (Multiple Trajectory Search …)
- IEMA — Hemanth Kumar Singh, et al (Performance of Infeasibility …)
- ECHT — R. Mallipeddi & P. N. Suganthan (Differential Evolution …)

113

NANYANG TECHNOLOGICAL UNIVERSITY

ECHT-DE (Special Session CEC 2010)

Algorithm	Parameters
jDEsoco	*NP*, $p_0, \tau_1, \tau_2, F_l, F_u, \theta, \beta, c_p, \alpha_1, \alpha_2, G_c$
DE-VPS	*NP, CR, F,* $NFT_0, \lambda, \theta, t_c, c_p$
RGA	*N*, p_c, η_c, p_m, η_m
E-ABC	*SN, S, ε, MR, dec, FEs ratio, cycle limit, Step size variation*
εDEg	$N, F_0, CR_0, T_c, \theta, P_g, R_g, M$
DCDE	*NP, F, CR, P, L, L_FEs*
Co-CLPSO	*w, c,* V_{max}, *ps, R, L, L_FES, T,* P_c
CDEb6e6rl	*NP*, n_0, δ
sp-MODE	$F, Cr, \lvert N_s(k) \rvert, \lvert P(0) \rvert, \alpha_\varepsilon$
MTS	*SSS, Threshold,* M_1, M_2
IEMA	*N, Crossover Probability, Crossover Index, Mutation Probability, Mutation Index, α*
ECHT	*NP, CR, F,* p_f, θ, T_c, c_p

114

NANYANG TECHNOLOGICAL UNIVERSITY

ECHT-DE (Special Session CEC 2010) (10D)

Alg/Prob	C01	C02	C03	C04	C05	C06	C07	C08	C09
jDEsoco	6	12	8	1	9	4	1	2	3
DE-VPS	10	6	10	8	5	9	9	10	5
RGA	8	8	12	7	10	10	10	4	6
E-ABC	9	7	11	10	7	7	11	11	8
εDEg	1	5	1	4	1	1	1	8	1
DCDE	11	4	1	9	1	1	7	9	4
Co-CLPSO	7	3	5	6	1	1	8	1	7
CDEb6e6rl	4	9	6	1	11	11	1	7	11
sp-MODE	1	11	9	12	12	12	1	6	12
MTS	12	10	7	11	6	6	12	12	9
IEMA	5	1	4	5	8	8	5	5	10
ECHT	1	2	1	1	4	5	6	3	2

115

NANYANG TECHNOLOGICAL UNIVERSITY

ECHT-DE (Special Session CEC 2010) (10D)

Alg/Prob	C10	C11	C12	C13	C14	C15	C16	C17	C18
jDEsoco	4	3	4	3	4	7	8	9	9
DE-VPS	5	7	9	5	5	4	1	6	1
RGA	6	8	10	6	7	6	9	7	7
E-ABC	9	11	4	7	11	10	6	8	8
εDEg	1	1	1	1	1	2	7	4	1
DCDE	3	5	8	11	2	1	5	2	5
Co-CLPSO	7	10	3	8	3	3	2	5	6
CDEb6e6rl	11	6	2	1	10	12	11	11	11
sp-MODE	12	12	12	12	9	8	12	12	12
MTS	8	9	11	10	12	11	10	10	10
IEMA	10	2	4	4	6	5	3	1	1
ECHT	2	4	4	9	8	9	4	3	1

116

NANYANG TECHNOLOGICAL UNIVERSITY

ECHT-DE (Special Session CEC 2010) (30D)

Alg/Prob	C01	C02	C03	C04	C05	C06	C07	C08	C09
jDEsoco	5	8	3	3	8	2	1	6	2
DE-VPS	11	6	7	7	4	7	6	9	8
RGA	7	7	11	6	5	8	11	12	7
E-ABC	8	9	10	9	6	6	12	8	9
εDEg	2	3	2	4	1	1	3	2	3
DCDE	10	2	1	8	9	9	5	3	12
Co-CLPSO	9	1	9	5	2	3	8	7	6
CDEb6e6rl	1	10	5	2	11	10	1	1	1
sp-MODE	3	11	8	12	12	12	10	10	11
MTS	12	12	6	10	7	5	7	11	10
IEMA	4	5	12	11	10	11	4	4	5
ECHT	6	4	4	1	3	4	9	5	4

ECHT-DE (Special Session CEC 2010)(30D)

Alg/Prob	C10	C11	C12	C13	C14	C15	C16	C17	C18
jDEsoco	2	3	1	1	4	6	7	9	9
DE-VPS	6	7	9	9	5	4	6	5	4
RGA	7	6	6	8	9	7	9	8	6
E-ABC	10	9	7	5	7	9	8	7	8
εDEg	3	2	10	4	1	2	1	6	7
DCDE	1	5	2	7	3	1	5	4	3
Co-CLPSO	8	10	3	10	5	3	1	3	5
CDEb6e6rl	12	1	8	3	10	11	10	12	11
sp-MODE	11	12	12	12	12	10	12	10	10
MTS	9	8	4	11	11	12	11	11	12
IEMA	5	11	11	2	2	5	4	1	1
ECHT	4	4	5	6	8	8	1	2	1

ECHT-DE (Special Session CEC 2010)

Algorithm	Ranking			
	10D	30D	Overall	Average
jDEsoco	97	80	177	4.92
DE-VPS	115	120	235	6.53
RGA	141	140	281	7.81
E-ABC	155	147	302	8.39
εDEg	42	57	99	2.75
DCDE	89	90	179	4.97
Co-CLPSO	86	98	184	5.11
CDEb6e6rl	136	120	256	7.11
sp-MODE	177	190	367	10.19
MTS	176	169	345	9.58
IEMA	87	108	195	5.42
ECHT	69	79	148	4.11

ECHT-DE (Special Session CEC 2010)

Rank	Algorithm
1st	εDEg
2nd	ECHT
3rd	jDEsoco
4th	DCDE
5th	Co-CLPSO
6th	IEMA
7th	DE-VPS
8th	CDEb6e6rl
9th	RGA
10th	E-ABC
11th	MTS
12th	sp-MODE

THANK YOU

Acknowledgement :- Many of myy research students contributed to these presentations:

Ast Prof M Rammohan, Korea

Dr Qu Boyang

Dr Zhao Shizheng

Mr Sheldon Hui

Ms Yu Ling

In addition, my collaborator Dr S Das also contributed to this presentation.

NANYANG TECHNOLOGICAL UNIVERSITY