Particle Swarm Optimization & Differential Evolution

Presenter: Assoc. Prof. P. N. Suganathan
School of Electrical and Electronic Engineering
Nanyang Technological University, Singapore.

Some Software Resources Available from:
http://www.ntu.edu.sg/home/epnsugan
Outline of the Presentation

I. Benchmark Test Functions
II. Real Parameter Particle swarm optimization (PSO)
   ■ Basic PSO, its variants, Comprehensive learning PSO (CLPSO), Dynamic multi-swarm PSO (DMS-PSO)
III. Real Parameter Differential evolution (DE)
   ■ DE, its variants, Self-adaptive differential evolution
IV. Constrained optimization
V. Multi-objective PSO / DE
VI. Multimodal optimization (niching)
VII. Binary / Discrete PSO & DE
□ Dynamic, Robust optimization – excluded.
I - Benchmark Test Functions

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

From Prof Xin Yao’s group http://www.cs.bham.ac.uk/research/projects/ecb/ Includes diverse problems.
Why do we require benchmark problems?

- Why we need test functions?
  - To evaluate a novel optimization algorithm’s property on different types of landscapes
  - Compare different optimization algorithms

- Types of benchmarks
  - Bound constrained problems (real, binary, discrete, mixed)
  - Constrained problems
  - Single / Multi-objective problems
  - Static / Dynamic optimization problems
  - Multimodal problems
  - Various combinations of the above
Shortcomings in Bound constrained Benchmarks

- Some properties of benchmark functions may make them unrealistic or may be exploited by some algorithms:
  - Global optimum having the same parameter values for different variables / dimensions
  - Global optimum at the origin
  - Global optimum lying in the center of the search range
  - Global optimum on the bound
  - Local optima lying along the coordinate axes
  - no linkage among the variables / dimensions or the same linkages over the whole search range
  - Repetitive landscape structure over the entire space

Do real-world problems possess these properties?
Liang et. al 2006c (Natural Computation) has more details.
How to Solve?

- Shift the global optimum to a random position to make the global optimum to have different parameter values for different dimensions.

- Rotate the functions as below:

\[ F(x) = f(R \times x) \]

where \( R \) is an orthogonal rotation matrix.

- Use different classes of benchmark functions, different rotation matrices to compose a single test problem.

- These **Composition Functions** mix different properties of different basic test functions together to destroy repetitive structures.
Novel Composition Test Functions

- Compose the standard benchmark functions to construct a more challenging function with a randomly located global optimum and several randomly located deep local optima with different linkage properties over the search space.

- Gaussian functions are used to combine these benchmark functions and to blur individual functions’ structures mainly around the transition regions.

Novel Composition Test Functions

\( F(x) \): new composition function.
\( f_i(x) \): \( i \)th basic function used to construct the composition function.
\( n \): number of basic functions. The bigger \( n \) is, the more complex \( F(x) \) is.
\( D \): dimension.

\([X_{\text{min}}, X_{\text{max}}]^D : F(x) \)'s search range

\([x_{\text{min}_i}, x_{\text{max}_i}]^D : f_i(x) \)'s search range

\( M_i \): orthogonal rotation matrix for each \( f_i(x) \)
\( o_i \): new shifted optimum position for each \( f_i(x) \)
\( o_{\text{old}_i} \): old optimum position for each \( f_i(x) \)

\[
F(x) = \sum_{i=1}^{n} \left( w_i \ast \left[ f_i' \left( (x - o_i + o_{\text{old}_i}) / \lambda_i \ast M_i \right) + bias_i \right] \right) + f\_bias
\]
Novel Composition Test Functions

\( w_i \): weight value for each \( f_i(x) \), calculated as below:

\[
\begin{align*}
  w_i &= \exp \left( -\frac{\sum_{k=1}^{D} (x_k - \sigma_k + \sigma_k^{old})^2}{2D\sigma_i^2} \right), \\
  w_i &= \begin{cases} 
    w_i & \text{if } w_i = \max(w_i) \\
    w_i \ast (1 - \max(w_i) \ast 10) & \text{if } w_i \neq \max(w_i)
  \end{cases}
\end{align*}
\]

then normalize the weight \( w_i = w_i / \sum_{i=1}^{n} w_i \)

\( \sigma_i \): used to control each \( f_i(x) \)'s coverage range, a small \( \sigma_i \) gives a narrow range for \( f_i(x) \).

\( \lambda_i \): used to stretch or compress the function, \( \lambda_i > 1 \) means stretch, \( \lambda_i < 1 \) means compress.

usually set \( \lambda_i = \sigma_i \ast \frac{X_{\max} - X_{\min}}{x_{\max} - x_{\min}} \)
Novel Composition Test Functions

\( a_i \) define the global and local optima’s position, 
\( bias \) define which optimum is global optimum.

If \( f_i(x) \) are different functions, different functions have different properties and height, 
In order to get a better mixture, we estimate the biggest function value \( f_{\text{max}} \), 
then normalize each basic function to similar height as below:

\[ f_i'(x) = C \times \frac{f_i(x)}{|f_{\text{max}}|}, \text{C is a predefined constant.} \]

These composition functions can also be used as multimodal functions.
A couple of Examples

- Many composition functions are available from our homepage

**Composition Function 1 (F1):**
Made of Sphere Functions

**Composition Function 2 (F2):**
Made of Griewank’s Functions

Similar analysis is needed for other benchmarks such as the multi-objective, constrained, etc.
II - Particle Swarm Optimizer

- Emulates flocking behavior of birds, animals, insects, fish, etc. to solve optimization problems
- Each solution in the landscape is a particle
- All particles have fitness values and velocities
- The standard PSO does not have mutation, crossover, selection, etc.
Particle Swarm Optimizer

- Two versions of PSO
  - Global version (**May not be used alone to solve multimodal problems**): Learning from the personal best (pbest) and the best position achieved by the whole population (gbest)
    \[
    V_i^d \leftarrow c_1 \times \text{rand1}_i^d \times (\text{pbest}_i^d - X_i^d) + c_2 \times \text{rand2}_i^d \times (\text{gbest}_d^d - X_i^d)
    \]
    \[
    X_i^d \leftarrow X_i^d + V_i^d \\
    \]  
    \(i\) – particle counter & \(d\) – dimension counter

  - Local Version: Learning from the pbest and the best position achieved in the particle's neighborhood population (lbest)
    \[
    V_i^d \leftarrow c_1 \times \text{rand1}_i^d \times (\text{pbest}_i^d - X_i^d) + c_2 \times \text{rand2}_i^d \times (\text{lbest}_k^d - X_i^d)
    \]
    \[
    X_i^d \leftarrow X_i^d + V_i^d \\
    \]

- The random numbers (rand1 & rand2) should be generated for each dimension of each particle in every iteration. lbest to be defined w. r. t. a neighborhood.
General parameters in PSO

- $c_1$ and $c_2$ denote the acceleration constants usually set to ~2.
- $\text{rand}_1^d$ and $\text{rand}_2^d$ are two uniform random numbers within the range $[0,1]$
- $x_i = (x_i^1, x_i^2, ..., x_i^D)$ represents the position of the $i^{th}$ particle
- $v_i = (v_i^1, v_i^2, ..., v_i^D)$ represents the position changing rate (velocity) of the $i^{th}$ particle
- $\text{pbest}_i = (pbest_i^1, pbest_i^2, ..., pbest_i^D)$ represents the best previous position (the position giving the best objective function value) of the $i^{th}$ particle
- $\text{gbest} = (gbest^1, gbest^2, ..., gbest^D)$ represents the best previous position of the whole swarm
- $\text{lbest}_i = (lbest_i^1, lbest_i^2, ..., lbest_i^D)$ represents the best previous position achieved by those particles within the neighborhood of the $i^{th}$ particle
PSO variants

- Modifying the Parameters
  - Inertia weight $\omega$ (Shi & Eberhart, 1998; Shi & Eberhart, 2001; Eberhart & Shi, 2001, …)
  - Constriction coefficient (Clerc, 1999; Clerc & Kennedy, 2002)
  - Time varying acceleration coefficients (Ratnaweera et al. 2004)
  - Linearly decreasing $V_{\text{max}}$ (Fan & Shi, 2001)
  - Tribes (Clerc 2006)

- Using Topologies
  - Extensive experimental studies (Kennedy, 1999; Kennedy & Mendes, 2002, …)
  - Dynamic neighborhood (Suganthan, 1999; Hu and Eberhart, 2002; Peram et al. 2003)
  - Combine the global version and local version together (Parsopoulos and Vrahatis, 2004) named as the unified PSO or UPSO.
  - Fully informed PSO or FIPS (Mendes & Kennedy 2004) and so on …
PSO variants and Applications

- Hybrid PSO Algorithms
  - PSO + selection operator (Angeline, 1998)
  - PSO + crossover operator (Lovbjerg, 2001)
  - PSO + mutation operator (Lovbjerg & Krink, 2002; Blackwell & Bentley, 2002; … …)
  - PSO + dimension-wise search (Bergh & Engelbrecht, 2004)
  - …

- Various Optimization Scenarios & Applications
  - Binary Optimization (Kennedy & Eberhart, 1997; Agrafiotis et al. 2002;)
  - Constrained Optimization (Parsopoulos et al. 2002; Hu & Eberhart, 2002; … )
  - Multi-objective Optimization (Ray et al. 2002; Coello et al. 2002/04; … )
  - Dynamic Tracking (Eberhart & Shi 2001; … )
  - Yagi-Uda antenna (Baskar et al 2005b), Photonic FBG design (Baskar et al 2005a), FBG sensor network design (Liang et al June 2006)
PSO with Momentum / Constriction

- In PSO with momentum [SE98], a momentum term \( \omega \) is introduced to the original equation:

\[
v_i^d \leftarrow \omega v_i^d + c_1 rand_1 (pbest_i^d - x_i^d) + c_2 rand_2 (gbest^d - x_i^d)
\]

\[
x_i^d \leftarrow x_i^d + v_i^d
\]

\( \omega \) is usually reduced from 0.9 to 0.4

- PSO with constriction factor [CK02]:

\[
v_i^d \leftarrow \chi [v_i^d + c_1 rand_1 (pbest_i^d - x_i^d) + c_2 rand_2 (gbest^d - x_i^d)]
\]

\[
x_i^d \leftarrow x_i^d + v_i^d
\]

Constriction factor \( \chi \) can be set to 0.7298.
PSO Variants by Kennedy et. al

- In fully informed particle swarm (FIPS) [KM06, MKN04], each particle’s velocity is adjusted based on contributions from pbest of all its neighbors.

- Bare bones PSO [K03]: PSO without the velocity term, i.e. with the social & cognitive terms only.

- Essential Particle swarm [K06]: The velocity is expressed as direction defined by the particle’s position at time $t$ and time $(t-1)$, i.e. the persistence and social influence.

- Essential Particle Swarm is another realization of the FIPS.
Comprehensive learning PSO (CLPSO)

- CLPSO learning strategy:
  \[
  v_i^d \leftarrow w \times v_i^d + c \times \text{rand}_i^d \times (pbest_{f_i(d)}^d - x_i^d)
  \]
  \[
  x_i^d \leftarrow x_i^d + v_i^d
  \]

- \(f_i = [f_i(1), f_i(2), \ldots, f_i(D)]\) denotes a set of particle indices with respect to each dimension of the particle \(i\). \(f_i(d)\) represents a comprehensive exemplar with each dimension composed of the value from the corresponding dimension of the pbest of particle \(pbest_i\). These indices take the value \(i\) itself with the probability \(Pc_i\), referred to as the learning probability, which takes different values with respect to different particles.

- For each dimension of particle \(i\), we generate a random number. If this random number is larger than \(Pc_i\), the corresponding dimension of particle \(i\) will learn from its own pbest, otherwise it will learn from the pbest of another randomly chosen particle.
Tournament selection with size 2 is used to choose the index \( f_i(d) \).

To ensure that a particle learns from good exemplars and to minimize the time wasted on poor directions, we allow each particle to learn from the exemplars until such particle stop to improve for a certain number of generations, called the refreshing gap \( m \) (7 generations).

After that, we re-assign \( f_i = [f_i(1), f_i(2), ..., f_i(D)] \) for each particle \( i \).

The detailed description and algorithmic implementation can be found in [LQSB06]. Matlab codes including CLPSO and several state-of-the-art PSO variants are available for academic use.
Three major differences between CLPSO and the conventional PSO are highlighted:

- Instead of using particle’s `pbest` and `gbest` as the exemplars, all particles’ `pbests` can be used to guide a particle’s flying direction.

- Instead of learning from the same exemplar for all dimensions, different dimensions of a particle may learn from different exemplars within certain generations. In other words, at one iteration, each dimension of a particle may learn from the corresponding dimension of different particle’s `pbest`.

- Instead of learning from two exemplars (pbest and gbest) in every generation, each dimension of a particle in CLPSO learns from just one comprehensive exemplar within certain generations.

Experimental results [LQSB06] over a suite of 16 numerical test functions have demonstrated the promising performance of the CLPSO to solve the multi-modal optimization problems in comparison with 8 state-of-the-art PSO variants.
CLPSO with Probability Adaptation

- Adaptive Self-Learning Strategy

- Assume $P_c$ normally distributed in a range with mean($P_c$) and a standard deviation of 0.1.

- Initially, mean($P_c$) is set at 0.5 and different $P_c$ values conforming to this normal distribution are generated for each individual in the current population.

- During every generation, the $P_c$ values associated with the particles which find new $p_{best}$ are recorded.

- The mean of normal distribution of $P_c$ is recalculated according to all the recorded $P_c$ values corresponding to successful movements during the last several generations.

- As a result, the proper $P_c$ value range for the current problem can be learned to suit the particular problem.
Dynamic multi-swarm PSO (DMS-PSO)

- DMS-PSO is constructed based on the local version of PSO with a novel neighborhood topology.

- Two major characteristics of the novel neighborhood topology:
  - Small sized swarms
  - Randomized re-grouping scheme

- The population is divided into several sub-swarms randomly.
- Each sub-swarm utilizes its own particles to search for better solutions and converge to some suboptimal solution.
- The whole population is re-grouped into new sub-swarms periodically. New sub-swarms continue the search procedure.
- This process continues until a termination criterion is satisfied.
DMS-PSO

DMS-PSO learning strategy

Each particle $i$ has an associated vector $\mathbf{Pc}_i$. After every $R$ generations, an indicator vector $\text{keepid}_i$ will be updated according to $\mathbf{Pc}_i$: if $\text{rand}_i^d$ is larger than or equal to $\mathbf{Pc}_i(d)$, $\text{keepid}_i(d)$ is set to 1 and the $d^{th}$ dimension of particle $i$ will be set as the value of its own $\text{pbest}_i(d)$, otherwise $\text{keepid}_i(d)$ is set to 0, and the $d^{th}$ dimension of particle $i$ will learn from its $\text{lbest}_i(d)$, and its own $\text{pbest}_i(d)$, as the PSO with constriction coefficients:

If $\text{keep } _i id_i^d = 0$

$$v_i^d \leftarrow 0.729 \times v_i^d + 1.49445 \times \text{rand}_1^d \times (\text{pbest}_i^d - x_i^d) + 1.49445 \times \text{rand}_2^d \times (\text{lbest}_i^d - x_i^d)$$

$$v_i^d = \min(v_{\text{max}}^d, \max(-v_{\text{max}}^d, v_i^d))$$

$$x_i^d \leftarrow x_i^d + v_i^d$$

Otherwise

Somewhat similar to DE & CPSO

$$x_i^d \leftarrow \text{pbest}_i^d$$
Parameter adaptation scheme

Assume $Pc_i$ is normally distributed with mean $mean_Pc$ and standard deviation 0.1.

Initially, $mean_Pc$ is set to 0.5 and a set of $Pc_i$ vectors with respect to each particle $i$ in the current population are generated according to such normal distribution.

At each generation, the $Pc_i$ values associated with those particles that find new pbests are recorded.

When sub-swarms are regrouped, $mean_Pc$ is recalculated according to all the recorded successful $Pc_i$ values. The recorded successful $Pc_i$ values will be cleared when $mean_Pc$ is recalculated.

As a result, a proper $Pc_i$ distribution with respect to the given problem can be evolved.
DMS-PSO with local search

- Although we can achieve larger diversity using DMS-PSO, the convergence rate may slow down. In order to alleviate this problem, a local search procedure is incorporated:
  - Every $L$ generations, pbests of five randomly chosen particles will be used as the starting points and the Quasi-Newton method is applied to conduct the local search with maximum function evaluations $L_{FEs}$.
  - At the end of the DMS-PSO search, particles in each sub-swarm are grouped into a whole swarm to perform the global PSO. The best solution achieved so far is refined using the Quasi-Newton method every $L$ generations with the $5 \times L_{FEs}$ as the maximum search step.
  - If local search results in improvements, the nearest pbest is replaced.
III - Outline of Presentation on DE

- Motivation for Differential Evolution (DE)
- Classical DE
- DE Variants
- Self-adaptive DE (SaDE)
Motivation for DE

- DE, proposed by Price and Storn in 1995 [PS95], was motivated by the attempts to use Genetic Annealing [P94] to solve the Chebychev polynomial fitting problem.

- Genetic annealing is a population-based, combinatorial optimization algorithm that implements a thermodynamic annealing criterion via thresholds. Although successfully applied to solve many combinatorial tasks, genetic annealing could not solve the Chebychev problem satisfactorily.

- Price modified genetic annealing by using floating-point encoding instead of bit-string one, arithmetic operations instead of logical ones, population-driven differential mutation instead of bit-inversion mutation and removed the annealing criterion. Storn suggested creating separate parent and children populations. Eventually, Chebychev problem can be solved effectively.

- DE is closely related to many other multi-point derivative free search methods [PSL05] such as evolutionary strategies, genetic algorithms, Nelder and Mead direct search and controlled random search.
DE at a glance

- **Characteristics**
  - Population-based stochastic direct search
  - Self-referential mutation
  - Simple but powerful
  - Reliable, robust and efficient
  - Easy parallelization
  - Floating-point encoding

- **Basic components**
  - Initialization
  - Trial vector generation
    - Mutation
    - Recombination
  - Replacement
**Initialization**

A population $P_{x,0}$ of $Np$ $D$-dimensional parameter vectors $x_{i,0} = [x_{1,i,0}, \ldots, x_{D,i,0}]$, $i=1,\ldots,Np$ is randomly generated within the prescribed lower and upper bound $b_L = [b_{1,L}, \ldots, b_{D,L}]$ and $b_U = [b_{1,U}, \ldots, b_{D,U}]$.

**Example:** the initial value (at generation $g=0$) of the $j^{th}$ parameter of the $i^{th}$ vector is generated by: $x_{j,i,0} = \text{rand}[0,1] \cdot (b_{j,U} - b_{j,L}) + b_{j,L}$, $j=1,\ldots,D$, $i=1,\ldots,Np$

**Trial vector generation**

At the $g^{th}$ generation, a trial population $P_{u,g}$ consisting of $Np$ $D$-dimensional trial vectors $v_{i,g} = [v_{1,i,g}, \ldots, v_{D,i,g}]$ is generated via mutation and recombination operations applied to the current population $P_{x,g}$.

**Differential mutation:** with respect to each vector $x_{i,g}$ in the current population, called target vector, a mutant vector $v_{i,g}$ is generated by adding a scaled, randomly sampled, vector difference to a basis vector randomly selected from the current population.
Insight into classical DE (DE/rand/1/bin)

Example: at the $g$th generation, the $i$th mutant vector $v_{i,g}$ with respect to $i$th target vector $x_{i,g}$ in the current population is generated by $v_{i,g} = x_{r0,g} + F \cdot (x_{r1,g} - x_{r2,g})$, $i \neq r0 \neq r1 \neq r2$, mutation scale factor $F \in (0,1+)$

Discrete recombination: with respect to each target vector $x_{i,g}$ in the current population, a trial vector $u_{i,g}$ is generated by crossing the target vector $x_{i,g}$ with the corresponding mutant vector $v_{i,g}$ under a pre-specified crossover rate $Cr \in [0,1]$

Example: at the $g$th generation, the $i$th trial vector $u_{i,g}$ with respect to $i$th target vector $x_{i,g}$ in the current population is generated by:

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

Replacement

If the trial vector $u_{i,g}$ has equal or better objective function value than that of its corresponding target vector $x_{i,g}$, it replaces the target vector in the $(g+1)$th generation; otherwise the target vector remains in the $(g+1)$th generation
Illustration of classical DE

Illustration of classic DE
Illustration of classical DE

Illustration of classical DE
Illustration of classical DE

Four operating vectors in 2D continuous space
Illustration of classical DE

Trial vector after Mutation
Illustration of classical DE

Trial vector after Crossover
Replacement of target vector by the trial vector
Differential vector distribution

A population of 5 vectors
20 generated difference vectors

Most important characteristics of DE: self-referential mutation!
ES: fixed probability distribution function with adaptive step-size
DE: adaptive distribution of difference vectors with fixed step-size
DE variants

Modification of different components of DE can result in many DE variants:

**Initialization**
Uniform distribution and Gaussian distribution

**Trial vector generation**
- Choices in base vector selection
  - Random selection without replacement: \( r_0 = \text{ceil}(\text{rand}_i[0,1] \cdot Np) \)
  - Permutation selection: \( r_0 = \text{permute}[i] \)
  - Random offset selection: \( r_0 = (i + r_g) \% Np \) (e.g. \( r_g = 2 \))
  - Biased selection: global best, local best or tournament
DE variants

- Differential mutation
  - One difference vector: \( F \cdot (x_{r1} - x_{r2}) \)
  - Two difference vector: \( F \cdot (x_{r1} - x_{r2}) + F \cdot (x_{r3} - x_{r4}) \)
  - Mutation scale factor \( F \)
    - Crucial role: balance exploration and exploitation
    - Dimension dependence?: \( jitter, \text{ if yes (rotation variant) and dither, if no (rotation invariant)} \)
    - Randomization: different distributions of \( F \)

DE/rand/1: \( V_{i,G} = X_{r1,G} + F \cdot (X_{r2,G} - X_{r3,G}) \)

DE/best/1: \( V_{i,G} = X_{best,G} + F \cdot (X_{r1,G} - X_{r2,G}) \)

DE/current-to-best/1: \( V_{i,G} = X_{i,G} + F \cdot (X_{best,G} - X_{i,G}) + F \cdot (X_{r1,G} - X_{r2,G}) \)

DE/rand/2: \( V_{i,G} = X_{r1,G} + F \cdot (X_{r2,G} - X_{r3,G} + X_{r4,G} - X_{r5,G}) \)

DE/best/2: \( V_{i,G} = X_{best,G} + F \cdot (X_{r1,G} - X_{r2,G} + X_{r3,G} - X_{r4,G}) \)
DE variants

- Recombination
  - Discrete recombination (crossover) (rotation variant)
    - One point and multi-point
    - Exponential (somewhat comparable to two-point)
    - Binominal (uniform)
  - Arithmetic recombination
    - Line recombination (rotation invariant, vector operation)
    - Intermediate recombination (rotation variant, dimension-wise)
    - Extended intermediate recombination (rotation variant)
Motivation for self-adaptation in DE

The performance of DE on different problems depends on:

- Population size
- **Strategy** and the associated **parameter setting** to generate trial vectors
- Replacement scheme

*It is hard to choose a unique combination to successfully solve any problem at hand*

- Population size usually depends on the problem scale and complexity
- During evolution, **different strategies coupled with specific parameter settings** may be effective for different search stages.
- Replacement schemes influence the population diversity
- Trial and error scheme may be a waste of computational time & resources

**Automatically adapt the configuration in DE so as to generate effective trial vectors during evolution**
Related works

**Practical guideline** [SP95], [SP97], [CDG99], [BO04], [PSL05], [GMK02]: for example, \( N_p \in [5D, 10D] \); Initial choice of \( F = 0.5 \) and \( CR = 0.1/0.9 \); Increase \( NP \) and/or \( F \) if premature convergence happens. **Conflicting conclusions with respect to different test functions.**

**Fuzzy adaptive DE** [LL02]: use fuzzy logical controllers whose inputs incorporate the relative function values and individuals of successive generations to adapt the mutation and crossover parameters.

**Self-adaptive Pareto DE** [A02]: encode crossover rate in each individual, which is simultaneously evolved with other parameters. Mutation scale factor is generated for each variable according to Gaussian distribution \( N(0,1) \).

**Zaharie** [Z02]: theoretically study the DE behavior so as to adapt the control parameters of DE according to the evolution of population diversity.

**Self-adaptive DE (1)** [OSE05]: encode mutation scale factor in each individual, which is simultaneously evolved with other parameters. Crossover rate is generated for each variable according to Gaussian distribution \( N(0.5, 0.15) \).

**DE with self-adaptive population** [T06]: population size, mutation scale factor and crossover rate are all encoded into each individual.
Self-Adapting Control Parameters in DE

- **[BGBMZ06]** jDE algorithm encodes mutation scale factor $F$ and crossover rate $CR$ in each individual.

- New values for $F$ & $CR$ are assigned to each individual from a set of values and the assignment is performed randomly with respect to pre-specified parameter values.

- jDE2 algorithm [BBG06] introduces re-initialization of poorly performing individuals to the jDE algorithm.
Self-adaptive DE (SaDE)

DE with strategy and parameter self-adaptation [QS05, HQS06]

Strategy adaptation: select one strategy from a pool of candidate strategies with the probability proportional to its previous successful rate to generate effective trial vectors during a certain learning period

Steps:
1. Initialize selection probability \( p_i = \frac{1}{\text{num}_st}, i = 1, \ldots, \text{num}_st \) for each strategy
2. According to the current probabilities, we employ stochastic universal selection to assign one strategy to each target vector in the current population
3. For each strategy, define vectors \( ns_i \) and \( nf_i \), \( i = 1, \ldots, \text{num}_st \) to store the number of trial vectors successfully entering the next generation or discarded by applying such strategy, respectively, within a specified number of generations, called “learning period (LP)”
4. Once the current number of generations is over LP, the first element of \( ns_i \) and \( nf_i \) with respect to the earliest generation will be removed and the behavior in current generation will update \( ns_i \) and \( nf_i \)
Self-adaptive DE (SaDE)

Parameter adaptation

Mutation scale factor ($F$): for each target vector in the current population, we randomly generate $F$ value according to a normal distribution $N(0.5,0.3)$. Therefore, 99% $F$ values fall within the range of $[-0.4,1.4]$.

Crossover rate ($CR_j$): when applying strategy $j$ with respect to a target vector, the corresponding $CR_j$ value is generated according to an assumed distribution, and those $CR_j$ values that have generated trial vectors successfully entering the next generation are recorded and updated every LP generations so as to update the parameters of the $CR_j$ distribution. We hereby assume that each $CR_j$, $j=1,…,num_st$ is normally distributed with its mean and standard deviation initialized to 0.5 and 0.1, respectively.

The selection probability $p_i$ is updated by $\frac{\sum ns_i}{(\sum ns_i + \sum nf_i)}$. Go to 2$^{nd}$ step.
Instantiations

- In CEC’05, we use 2 strategies:

  DE/rand/1/bin: \[ \mathbf{v}_{i,G} = \mathbf{x}_{r_1,G} + F \cdot (\mathbf{x}_{r_2,G} - \mathbf{x}_{r_3,G}) \]

  DE/current-to-best/2/bin: \[ \mathbf{v}_{i,G} = \mathbf{x}_{i,G} + F \cdot (\mathbf{x}_{\text{best}_G} - \mathbf{x}_{i,G}) + F \cdot (\mathbf{x}_{r_1,G} - \mathbf{x}_{r_2,G} + \mathbf{x}_{r_3,G} - \mathbf{x}_{r_4,G}) \]

  LP = 50

- In CEC’06, we employ 4 strategies:

  DE/rand/1/bin: \[ \mathbf{v}_{i,G} = \mathbf{x}_{r_1,G} + F \cdot (\mathbf{x}_{r_2,G} - \mathbf{x}_{r_3,G}) \]

  DE/rand/2/bin: \[ \mathbf{v}_{i,G} = \mathbf{x}_{r_1,G} + F \cdot (\mathbf{x}_{r_2,G} - \mathbf{x}_{r_3,G} + \mathbf{x}_{r_4,G} - \mathbf{x}_{r_5,G}) \]

  DE/current-to-best/2/bin: \[ \mathbf{v}_{i,G} = \mathbf{x}_{i,G} + F \cdot (\mathbf{x}_{\text{best}_G} - \mathbf{x}_{i,G}) + F \cdot (\mathbf{x}_{r_1,G} - \mathbf{x}_{r_2,G} + \mathbf{x}_{r_3,G} - \mathbf{x}_{r_4,G}) \]

  DE/current-to-rand/1: \[ \mathbf{v}_{i,G} = \mathbf{x}_{i,G} + F \cdot (\mathbf{x}_{r_1,G} - \mathbf{x}_{i,G}) + F \cdot (\mathbf{x}_{r_2,G} - \mathbf{x}_{r_3,G}) \]

  LP = 50
Local search enhancement

To improve convergence speed, we apply a local search procedure every 500 generations:

- To apply local search, we choose \( n = 0.05 \cdot N_p \) individuals, which include the individual having the best objective function value and the \( n-1 \) individuals randomly selected from the top 50% individuals in the current population.

- We perform the local search by applying the Quasi-Newton method to the selected \( n \) individuals.
Overview of DE research trends

DE Applications

- Digital Filter Design
- Multiprocessor synthesis
- Neural network learning
- Diffraction grating design
- Crystallographic characterization
- Beam weight optimization in radiotherapy
- Heat transfer parameter estimation in a trickle bed reactor
- Electricity market simulation
- Scenario-Integrated Optimization of Dynamic Systems
- Optimal Design of Shell-and-Tube Heat Exchangers
- Optimization of an Alkylation's Reaction
- Optimization of Thermal Cracker Operation
- Optimization of Non-Linear Chemical Processes
- Optimum planning of cropping patterns
- Optimization of Water Pumping System
- Optimal Design of Gas Transmission Network
- Differential Evolution for Multi-Objective Optimization
- Bioinformatics
IV - Constrained Optimization

- Optimization of constrained problems is an important area in the optimization field.
- In general, the constrained problems can be transformed into the following form:

\[
\begin{align*}
\text{Minimize} & \quad f(\mathbf{x}), \ \mathbf{x} = [x_1, x_2, \ldots, x_D] \\
\text{subjected to:} & \\
& g_i(\mathbf{x}) \leq 0, \ i = 1, \ldots, q \\
& h_j(\mathbf{x}) = 0, \ j = q + 1, \ldots, m
\end{align*}
\]

\( q \) is the number of inequality constraints and \( m-q \) is the number of equality constraints.
For convenience, the equality constraints can be transformed into inequality form:

\[ |h_j(x)| - \varepsilon \leq 0 \]

where \( \varepsilon \) is the allowed tolerance.

Then, the constrained problems can be expressed as

Minimize \( f(x), x = [x_1, x_2, \ldots, x_D] \)

subjected to \( G_j(x) \leq 0, j = 1, \ldots, m, \)

\[ G_{1,\ldots,q}(x) = g_{1,\ldots,q}(x), G_{q+1,\ldots,m}(x) = |h_{q+1,\ldots,m}(x)| - \varepsilon \]

If we denote with \( F \) the feasible region and \( S \) the whole search space, \( x \in F \) if \( x \in S \) and all constraints are satisfied. In this case, \( x \) is a feasible solution.
Constraint-Handling (CH) Techniques

- Penalty Functions:
  - Static Penalties (Homaifar et al., 1994; …)
  - Dynamic Penalty (Joines & Houck, 1994; Michalewicz & Attia, 1994; …)
  - Adaptive Penalty (Eiben et al., 1998; Coello, 1999; Tessema & Gary Yen 2006, Smith & Tate 1993; …)
  - …

- Superiority of feasible solutions
  - Start with a population of feasible individuals (Michalewicz, 1992; Hu & Eberhart, 2002; …)
  - Feasible favored comparing criterion (Ray, 2002; Takahama & Sakai, 2005; …)
  - Specially designed operators (Michalewicz, 1992; …)
  - …
Constraint-Handling (CH) Techniques

- Separation of objective and constraints
  - Stochastic Ranking (Runarsson & Yao, TEC, Sept 2000)
  - Co-evolution methods (Coello, 2000a)
  - Multi-objective optimization techniques (Coello, 2000b; Mezura-Montes & Coello, 2002;… )
  - Feasible solution search followed by optimization of objective (Venkatraman & Gary Yen, 2005)
  - …

- While most CH techniques are modular (i.e. we can pick one CH technique and one search method independently), there are also CH techniques embedded as an integral part of the EA.
Novel Constraint-Handling Mechanism

Suppose that there are \( m \) constraints, the population is divided into \( n \) sub-swarms with \( sn \) members in each sub-swarm and the population size is \( ps \) (\( ps=n*sn \)). \( n \) is a positive integer and ‘\( n=m \)’ is not required.

The objective and constraints are assigned to the sub-swarms adaptively according to the difficulties of the constraints.

By this way, it is expected to have population of feasible individuals with high fitness values.
DMS-PSO’s Constraint-Handling Mechanism

How to assign the objective and constraints to each sub-swarm?

Define

\[ a > b = \begin{cases} 
1 & \text{if } a > b \\
0 & \text{if } a \leq b 
\end{cases} \]

\[ p_i = \frac{\sum_{j=1}^{ps} (g_i(x_j) > 0)}{ps}, \quad i = 1, 2, \ldots, m \]

Thus

\[ fp = 1 - p \quad p = [p_1, p_2, \ldots, p_m] \]

\[ fp + \sum_{i=1}^{m} (p_i / m) = 1 \]
DMS-PSO’s Constraint-Handling Mechanism

For each sub-swarm,

- Using roulette selection according to $fp$ and $p_i/m$ to assign the objective function or a single constraint as its target.

  If sub-swarm $i$ is assigned to improve constraint $j$, set $obj(i)=j$ and if sub-swarm $i$ is assigned to improve the objective function, set $obj(i)=0$.

- Assigning swarm member for this sub-swarm: Sort the unassigned particles according to $obj(i)$, and assign the best and $sn-1$ worst particles to sub-swarm $i$. 
DMS-PSO’s Comparison Criteria

1. If $\text{obj}(i) = \text{obj}(j) = k$ (particle $i$ and $j$ handling the same constraint $k$), particle $i$ wins if

   \[
   G_k(x_i) < G_k(x_j) \quad \text{with} \quad G_k(x_j) > 0
   \]

   or

   \[
   V(x_i) < V(x_j) \quad \text{and} \quad G_k(x_i), G_k(x_j) \leq 0
   \]

   or

   \[
   f(x_i) < f(x_j) \quad \text{and} \quad V(x_i) = V(x_j)
   \]

2. If $\text{obj}(i) = \text{obj}(j) = 0$ (particle $i$ and $j$ handling $f(x)$) or $\text{obj}(i) \neq \text{obj}(j)$ ($i$ and $j$ handling different objectives), particle $i$ wins if

   \[
   V(x_i) < V(x_j)
   \]

   or

   \[
   f(x_i) < f(x_j) \quad \text{and} \quad V(x_i) = V(x_j)
   \]

   \[
   V(x) = \sum_{i=1}^{m} (\text{weight}_i \cdot G_i(x) \cdot (G_i(x) \geq 0))
   \]

   \[
   \text{weight}_i = \frac{1/G_i \max}{\sum_{i=1}^{m} (1/G_i \max)}, \quad i = 1, 2, \ldots, m
   \]
DMS-PSO for Constrained Optimization

- Step 1: Initialization -
  Initialize \( ps \) particles (position \( X \) and velocity \( V \)), calculate \( f(X), G_j(X) \) \((j=1,2...,m)\) for each particle.

- Step 2: Divide the population into sub-swarms and assign \( obj \) for each sub-swarm using the novel constraint-handling mechanism, calculate the mean value of \( Pc \) (except in the first generation, \( \text{mean}(Pc)=0.5 \)), calculate \( Pc \) for each particle. Then empty \( Pc \).

- Step 3: Update the particles according to their objectives; update \( pbest \) and \( gbest \) of each particle according to the same comparison criteria, record the \( Pc \) value if \( pbest \) is updated.
Step 5: Local Search-
Every $L$ generations, randomly choose 5 particles’ $p_{\text{best}}$ and start local search with Sequential Quadratic Programming (SQP) method using these solutions as start points ($\text{fmincon}(\ldots,\ldots,\ldots)$ function in Matlab is employed). The maximum fitness evaluations for each local search is $L_{\text{FEs}}$.

Step 6: If $F_{\text{Es}} \leq 0.7 \times \text{Max}_{\text{FEs}}$, go to Step 3. Otherwise go to Step 7.

Step 7: Merge the sub-swarms into one swarm and continue PSO (Global Single Swarm). Every $L$ generations, start local search using $g_{\text{best}}$ as start points using $5 \times L_{\text{FEs}}$ as the Max FEs. Stop search if $F_{\text{Es}} \geq \text{Max}_{\text{FEs}}$
SaDE for Constrained Optimization

- Strategy Adaptation
  - Probabilistically select one out of several available learning strategies to apply for each individual in the current population

\[
\begin{align*}
\text{DE/Rand/1:} & \quad V_{i,G} = X_{r_1,G} + F \cdot (X_{r_2,G} - X_{r_3,G}) \\
\text{DE/Current to best/2:} & \quad V_{i,G} = X_{i,G} + F \cdot (X_{\text{best},G} - X_{i,G}) + F \cdot (X_{r_1,G} - X_{r_2,G} + X_{r_3,G} - X_{r_4,G}) \\
\text{DE/Rand/2:} & \quad V_{i,G} = X_{r_1,G} + F \cdot (X_{r_2,G} - X_{r_3,G}) + F \cdot (X_{r_4,G} - X_{r_5,G}) \\
\text{DE/Current-to-rand/1:} & \quad U_{i,G} = X_{r_1,G} + K \cdot (X_{r_3,G} - X_{i,G}) + F \cdot (X_{r_1,G} - X_{r_2,G})
\end{align*}
\]
Initial probabilities \( p_1 = p_2 = p_3 = p_4 = 0.25 \)

According to the probability, we apply Stochastic Universal Selection to select the strategy for each individual in the current population.

\( n_{si} \) (\( n_{fi} \)), \( i = 1, 2, 3, 4 \): the accumulated number of trial vectors, successfully entering (discarded) the next generation while generated by each strategy.

\( n_{si} \) and \( n_{fi} \) are accumulated within a specified number of generations, called the “learning period (LP)”. The probability \( p_i \) is updated as:

\[
p_i = \frac{n_{si}}{n_{si} + n_{fi}}
\]
Self-adaptive Differential Evolution

Parameters adaptation

\( F: \) different random values \( \text{normrnd}(0.5, 0.3) \) in the range \((0, 2]\)  for different individuals

\( CR: \) accumulating the previous learning experience within a certain generational interval so as to dynamically adapt the value of \( CR \) to a suitable range

\[
\begin{align*}
\text{IF REM (G, LP)=0} \\
CRm &= \text{mean(CRpool)} \\
\text{END IF}
\end{align*}
\]

\[
\begin{align*}
\text{IF REM (G, 5)=0} \\
\text{FOR } i = 1 \text{ to } NP \\
CRi &= \text{normrand}(CRm, 0.1) \\
\text{END FOR} \\
\text{END IF}
\end{align*}
\]
Extend SaDE to Handle Constraints

Selection procedure

The trial vector $U_{i,G}$ is compared to its corresponding target vector $X_{i,G}$ in the current population considering both the fitness value and constraints.

$U_{i,G}$ will replace $X_{i,G}$ if any of the following conditions is true

1. $U_{i,G}$ is feasible, $X_{i,G}$ is not.
2. $U_{i,G}$ and $X_{i,G}$ are both feasible, and $U_{i,G}$ has smaller or equal fitness value (for minimization problem) than $X_{i,G}$.
3. $U_{i,G}$ and $X_{i,G}$ are both infeasible, but $U_{i,G}$ has a smaller overall constrain violation.
Local Search

- To speed up the convergence, we apply a local search procedure once every 500 generations

- $n=5\%$ of $NP$ individuals
  
  $DE_{gbest}$ + randomly selected $n-1$ individuals from the best 50% individuals in the current population

- We employ the Sequential Quadratic Programming (SQP) method as the local search method.
Many real-world problems involve multiple, conflicting objectives

- Robotics and control engineering
- Transport engineering
- Scheduling
- Finance
- Bioinformatics
- Pattern recognition
- PID design

Non-dominated solutions: In a set of solutions $P$, the non-dominated set of solutions $P'$ are those that are not dominated by any member of the set $P$.

- Pareto-optimality: When the set $P$ is the entire search space, the resulting $P'$ is called the Pareto-optimal set.

Non-dominated solutions: In a set of solutions $P$, the non-dominated set of solutions $P'$ are those that are not dominated by any member of the set $P$.

- Pareto-optimality: When the set $P$ is the entire search space, the resulting $P'$ is called the Pareto-optimal set.
Multi-Objective Optimization

Mathematically, we can use the following formula to express the multi-objective optimization problems (MOP):

Minimize \( y = f(x) = (f_1(x), f_2(x), K, f_m(x)) \) \( x \in [X_{\text{min}}, X_{\text{max}}] \)

subject to \( g_j(x) \leq 0, j = 1, ..., z \)
\( h_k(x) = 0, k = q + 1, ..., m \)

The objective of multi-objective optimization is to find a set of solutions which can represent the Pareto-optimal set well, thus there are two goals for the optimization:

1) Convergence to the Pareto-optimal set
2) Diversity of solutions in the Pareto-optimal set
Representative MOEAs [CJKO01][D01][ZLT01][CVL02][AJG05][TKL05]

- Non-elitist MOEAs
  - Weight based GA (WBGA)
  - Multiple objective GA (MOGA)
  - Niched Pareto GA
  - Non-dominated sorting GA (NSGA)

- Elitist MOEAs
  - Distance-based Pareto GA (DPGA)
  - Strength Pareto GA (SPEA), SPEA-II
  - Non-dominated GA-II (NSGA-II)
  - Pareto-archived ES (PAES),
  - Pareto envelope-based selection algorithm (PESA), PESA-II
  - Multi-objective messy GA (MOMGA)
Mainly used techniques:

- External archive for the non-dominated solution set.
- How to update pbest and gbest (or lbest)
  - Execute non-domination comparison with pbest or gbest
  - Execute non-domination comparison among all particles’ pbests and their offspring in the entire population
- How to choose gbest (or lbest)
  - Choose gbest (or lbest) from the recorded non-dominated solutions
  - Choose good local guides
- How to keep diversity
  - Crowding distance sorting
  - Subpopulation
From CLPSO and DMS-PSO to MOPSOs

- Combine an external archive which is used to record the non-dominated solutions found so far.

- Use Non-dominated Sorting and Crowding distance Sorting which have been used in NSGAII (Deb et al., 2002) to sort the members in the external archive.

- Choose exemplars (CLPSO) or lbest (DMS-PSO) from the non-dominated solutions recorded in the external archive.

- Experiments show that MO-CLPSO and DMS-MO-PSO are both capable of converging to the true Pareto optimal front and maintaining a good diversity along the Pareto front.
Selection of \textit{pbest}, \textit{gbest} in MO-CLPSO [HSL06]

\section*{Selection of \textit{pbest}}

\begin{align*}
\text{if } X_i \text{ dominates } pbest_i, \quad & pbest_i = X_i \\
\text{if } pbest_i \text{ and } X_i \text{ are non-dominated with each other,} & \\
\quad \text{if } rand < 0.5, \quad & pbest_i = X_i
\end{align*}

\section*{Selection of \textit{gbest} in MOCLPSO}

- all the non-dominated solutions are good individuals
- Randomly choose a particle from the non-dominated solutions.
- Other alternatives would be to form grids in the objective space and to select representatives from each cells, or to select more from less crowded cells, etc.
MO-CLPSO Algorithm

1) Initialize
Randomly initialize particle positions, Initialize particle velocities
Evaluate the fitness values of particles, initialize the external archive.

2) Optimize
WHILE stopping criterion is not satisfied
   DO
      For \( i = 1 \) to \( NP \)
      Select \( gbest \) from external archive
      Assign each dimension to learn from \( gbest, pbest \) of this particle and \( pbests \) of other particles,
MO-CLPSO Algorithm (cont.)

Update particle velocity

\[
\begin{align*}
\text{if } a_i(d) &= 1 \\
V_i(d) &= \omega_k \ast V_i(d) + \text{rand()} \ast (\text{gbest}(d) - X_i(d)) \\
\text{if } b_i(d) &= 1 \\
V_i(d) &= \omega_k \ast V_i(d) + \text{rand()} \ast (\text{pbest}_{f_i(d)}(d) - X_i(d)) \\
\text{else} \\
V_i(d) &= \omega_k \ast V_i(d) + \text{rand()} \ast (\text{pbest}_i(d) - X_i(d))
\end{align*}
\]

Update particle position

\[ X_i(d) = X_i(d) + V_i(d) \]

Evaluate the fitness values of particle

Update \text{pbest} if current position is better than \text{pbest}

End For

Update the external archive

Increment the generation count

END WHILE
The solutions in the external archive are sorted based on one randomly chosen objective and then partitioned into $n$ groups where $n$ is the number of sub-swarms.

Each sub-swarm randomly selects one representative as the gbest from each partition of the external archive.
MOSaDE

- MOSaDE is an extension of SaDE to optimize problems with multiple objectives.
- Similar to SaDE, the MOSaDE algorithm automatically adapts the trial vector generation strategies and their associated parameters according to their previous experience of generating promising or inferior individuals.
- However, when extending the single-objective algorithm to multi-objective domain, the evaluation criteria of promising or inferior individuals must be changed.
MOSaDE

We use the following:

Individual A is better than individual B, if

(1) individual A dominates B, or

(2) individual A and individual B are non-dominated with each other, but A is less crowded than individual B.

Therefore, in case that the trial vector is better than the target vector according to this criterion, we will record the associated parameter and strategy.
The strategies incorporated into our proposed MOSaDE algorithm are ‘rand/1/bin’ and ‘best/2/bin’.

The $X_{best}$ in ‘best/2/bin’ is randomly selected from external archive.

**MOSaDE Algorithm**

**Step 1.** Randomly initialize a population of $NP$ individuals. Initialize strategy probability ($p_k$, $k=1,\ldots,K$, $K$ is the no. of available strategies), the median value of $CR(CRmk)$ for each strategy, learning period ($LP=50$).

**Step 2.** Evaluate the individuals in the population, and fill the external archive with these individuals.
MOSaDE Algorithm

Step 3. Repeat

(1) Calculate strategy probability $p_k$: the percentage of the success rate of trial vectors generated by each strategy during the learning period.

(2) Assign trial vector generation strategy and parameters to each target vector $X_i$
   (a) Use stochastic universal sampling to select one strategy $k$ for each target vector $X_i$
   (b) Assign control parameters $F$ and $CR$

   $F$: Generate the $F$ values under $\text{Normrnd}(0.3, 0.1)$

   $CR$: After the first $LP$ generation, calculate $CRm_k$ according to the recorded $CR$ values. Generate the $CR$ values under $\text{Normrnd}(CRm_k, 0.1)$

(3) Generate a new population where each trial vector is generated according to associated trial vector generation strategy $k$ and parameter $F$ and $CR$ in (2).

(4) Selection:
FOR $i=1:NP$

(a) Evaluate the trial vector $U_i^k$, and compare with the target vector $X_{n(i)}$ nearest to the trial vector in the solution space.

   IF $X_{n(i)}$ dominates $U_i^k$, discard $U_i^k$.

ELSE

   IF $U_i^k$ dominates $X_{n(i)}$, replace $X_{n(i)}$ with $U_i^k$;

   IF non-dominated with each other, choose less crowded one to be the new target vector;

   $U_i^k$ will enter the external archive if (i) $U_i^k$ dominates some individual(s) of the archive (the dominated individuals in the archive are deleted); or (ii) $U_i^k$ is nondominated with archived individuals

   END IF

(b) If trial vector is better than $X_{n(i)}$, record the associated parameter $CR$ and flag strategy $k$ as successful strategy. Otherwise, flag strategy $k$ as failed strategy.

(c) When the external archive exceeds the maximum specified size, we select the less crowded individuals based on harmonic average distance to keep the archive size.

END FOR
Local search

- Use local search to further improve solutions found by the MOSaDE algorithm.
- Employ the Quasi-Newton method as the local search method, considering only one objective randomly selected each time.
- The local search procedure is applied once every 200 generations, on 10 individuals randomly selected among the non-dominated solutions that were not applied local search previously.
Fitness sharing [GR-87] modifies the search landscape by reducing the fitness of individuals in densely-populated regions. A sharing radius $\sigma_s$ is used to determine whether two individuals share the same niche.

Reducing an individual’s fitness is controlled by two operations, a similarity function and a sharing function. The shared fitness $f_i'$ is given by the formula as below:

$$f_i' = f_i / \sum_{j=1}^{N} sh(d_{ij})$$

with

$$sh(d_{ij}) = \begin{cases} 
1 - \left( \frac{d_{ij}}{\sigma_s} \right)^\alpha & \text{if } d_{ij} < \sigma_s \\
0 & \text{otherwise} 
\end{cases}$$

where $f_i$ denotes the original fitness of the individual $i$, $N$ the population size, and $d_{ij}$ the distance between the individual $i$ and the individual $j$. $\alpha$ is a constant parameter which regulates the shape of the sharing function $sh$ (typically $\alpha=1$).

The effect of this scheme is to encourage search in unexplored regions. The weakness of this method lies in the fact that it requires a priori knowledge about the distance between the peaks in the search space.
**k-means clustering based**

- **K-means clustering algorithm** is used to divide the population into niches [YG-93]. The fitness is calculated based on the distance $d_{ic}$ between the individual and its niche centroid.

- The final fitness of an individual is calculated by the relation:

$$F_i = \frac{f_i}{n_c (1 - (d_{ic}/2d_{max})^\alpha)}$$

  $n_c$ is the number of individuals in the niche containing individual $i$, $d_{max}$ is the maximum distance allowed between an individual and its niche centroid, and $\alpha$ is a constant.

- The formation of the niches is based on the adaptive $K$-mean algorithm. The algorithm begins with a fixed number ($k$) of seed points taken as the best $k$ individuals.

- Using a minimum allowable distance $d_{min}$ between niche centroids, a few clusters are formed from the seed points.

- The remaining population members are then added to these existing clusters or are used to form new clusters based on $d_{min}$ and $d_{max}$. These computations are performed in each generation.
Deterministic crowding (DC)

- Deterministic crowding [M-95] is an extension of a technique first used by De Jong to help promote diverse populations [D-75]. After crossover and mutation, the offspring then replace their closest parent if it has a better fitness.

- Calculate the distances between $p_1$ and $c_1$, $p_2$ and $c_2$, $p_1$ and $c_2$, $p_2$ and $c_1$, and name them $d_1$, $d_2$, $d_3$, $d_4$ respectively.
  
  - If $d_1+d_2 \leq d_3+d_4$, then
  - If the fitness of $c_1$ is higher than the fitness of $p_1$, replace $p_1$ with $c_1$;
  - If the fitness of $c_2$ is higher than the fitness of $p_2$, replace $p_2$ with $c_2$.
  - Else
  - If the fitness of $c_2$ is higher than the fitness of $p_1$, replace $p_1$ with $c_2$;
  - If the fitness of $c_1$ is higher than the fitness of $p_2$, replace $p_2$ with $c_1$.

- Deterministic crowding uses a distance measure to determine similarity between individuals. As, DC does not require the use of a similarity radius, this relaxes the requirement of a priori domain knowledge and makes DC more suitable for difficult problems than fitness sharing. DC is an elitist niching method. This means that once a peak is discovered, it is never lost from the population.
RTS [H-94] adapts tournament selection for multimodal optimization. It initially selects two elements from the population to undergo crossover and mutation. Then a random sample of \( w \) individuals is taken from the population to be compared with each offspring created, and the most similar (or the closest) individual is chosen to compete with the offspring. If the offspring wins, it is allowed to enter the population.
Clearing

- Clearing [P-96] is best described as a variant of the sharing technique.

- Instead of sharing resources between all individuals of a same niche as in the fitness sharing scheme, clearing attributes them only to the best few members of the niche and removes the inferior individuals. The remaining individuals form the mating pool and generate offspring.

```java
void Clearing(double Sigma, int Kappa) {
    int i, j, nbWinners;
    SortFitness(P);
    for (i=0 ; i<n ; i++) {
        if (Fitness(P[i]) > 0) {
            nbWinners = 1;
            for (j = i+1 ; j< n-1 ; j++) {
                if (Fitness(P[j])>0 && Distance (P[I], P[j])<Sigma) {
                    if (nbWinners < Kappa) {
                        nbWinners++;
                    }
                    else {
                        Fitness (P[j]) = 0.0;
                    }
                }
            }
        }
    }
}
```

the algorithm of Clearing
Kennedy proposed the niched PSO using K-means clustering [K-00].

The gbest / pbest / lbest were replaced by cluster centers or the best particle of each cluster to obtain several variants.

Clustering-based variants performed better than the original PSO.
Deflection, Stretching, Repulsion based Niched PSO

- Parsopoulos [PV04b] et al. made use of deflection, stretching, repulsion, etc. to locate as many optima as possible.

- These techniques transform the objective function to make previously obtained local optima to have high function values (or low fitness).
(1) Initialize the main particle swarm.
(2) Train the main swarm particles using one iteration of the *cognition only* model.
(3) Update the fitness of each main swarm particle.
(4) For each subswarm:
   (a) Train subswarm particles using one iteration of the GCPSO algorithm.
   (b) Update each particle’s fitness.
   (c) Update swarm radius
(5) If possible, merge subswarms.
(6) Allow subswarms to absorb any particles from the main swarm that moved into it.
(7) Search the main swarm for any particle that meets the partitioning criteria. If any is found, create a new subswarm with this particle and its closest neighbor.
(8) Repeat from 2 until stopping criteria are met.

Fig. 1. NichePSO algorithm.
Binary PSO (K&E97)

Sigmoid function

- Force the real values between 0 and 1

Velocity is updated with traditional equation

Sigmoid function is used to squash them to be within [0,1]

\[ s(v_{ij}) = \frac{1}{1 + \exp(-v_{ij})} \]

\[ X_{ij} = 1 \text{ if } r \leq s(v_{ij}) \]

\[ X_{ij} = 0 \text{ if } r > s(v_{ij}) \]

\[ r = \text{uniform random number} \]
Angle Modulated PSO / DE [PFE05, PEF06]

\[ g(x) = \sin(2\pi(x - a) \times b \times \cos(A)) + d \]

where \[ A = 2\pi \times c(x - a) \]

• \(a, b, c\) and \(d\) are real valued variables to be optimized by the PSO or DE.

• If there are 10 binary variables, \(x\) takes 10 different values, for example, from 1 to 10.

• For every solution of “\(a, b, c\) and \(d\)” binary bits are generated by \(\text{sign}(g(x))\) operation (as \(x\) runs from 1 to 10 in the case 10 bit problem.)
VIII - Benchmarking Evolutionary Algorithms

- CEC05 comparison results (Single obj. + bound const.)
- CEC06 comparison results (Single obj + general const.)
- Experimental Results on MOPSOs
- CEC07 comparison results on MOEAs

- CEC benchmarking resources available from http://www.ntu.edu.sg/home/epnsugan/
CEC’05 Comparison Results

- Algorithms involved in the comparison:
  - **BLX-GL50** (Garcia-Martinez & Lozano, 2005): Hybrid Real-Coded Genetic Algorithms with Female and Male Differentiation
  - **BLX-MA** (Molina et al., 2005): Adaptive Local Search Parameters for Real-Coded Memetic Algorithms
  - **CoEVO** (Posik, 2005): Mutation Step Co-evolution
  - **DE** (Ronkkonen et al., 2005): Differential Evolution
  - **DMS-L-PSO**: Dynamic Multi-Swarm Particle Swarm Optimizer with Local Search
  - **EDA** (Yuan & Gallagher, 2005): Estimation of Distribution Algorithm
  - **G-CMA-ES** (Auger & Hansen, 2005): A restart Covariance Matrix Adaptation Evolution Strategy with increasing population size
  - **K-PCX** (Sinha et al., 2005): A Population-based, Steady-State real-parameter optimization algorithm with parent-centric recombination operator, a polynomial mutation operator and a niched -selection operation.
  - **L-CMA-ES** (Auger & Hansen, 2005): A restart local search Covariance Matrix Adaptation Evolution Strategy
  - **L-SaDE** (Qin & Suganthan, 2005): Self-adaptive Differential Evolution algorithm with Local Search
  - **SPC-PNX** (Ballester et al., 2005): A steady-state real-parameter GA with PNX crossover operator
CEC’05 Comparison Results

- Problems: 25 minimization problems (Suganthan et al. 2005)
- Dimensions: \( D=10, 30 \)
- Runs / problem: 25
- Max_FES: \( 10000*D \) (Max_FES_10D= 100000; for 30D=300000; for 50D=500000)
- Initialization: Uniform random initialization within the search space, except for problems 7 and 25, for which initialization ranges are specified. The same initializations are used for the comparison pairs (problems 1, 2, 3 & 4, problems 9 & 10, problems 15, 16 & 17, problems 18, 19 & 20, problems 21, 22 & 23, problems 24 & 25).
- Global Optimum: All problems, except 7 and 25, have the global optimum within the given bounds and there is no need to perform search outside of the given bounds for these problems. 7 & 25 are exceptions without a search range and with the global optimum outside of the specified initialization ranges.
CEC’05 Comparison Results

- **Termination**: Terminate before reaching Max_FES if the error in the function value is $10^{-8}$ or less.
- **Ter_Err**: $10^{-8}$ (termination error value)
- **Successful Run**: A run during which the algorithm achieves the fixed accuracy level within the Max_FES for the particular dimension.
- **Success Rate** = (# of successful runs) / total runs
- **Success Performance** = mean (FEs for successful runs)*(# of total runs) / (# of successful runs)
### CEC’05 Comparison Results

#### Success Rates of the 11 algorithms for 10-D

<table>
<thead>
<tr>
<th>Algorithms</th>
<th>Func 1</th>
<th>Func 2</th>
<th>Func 3</th>
<th>Func 4</th>
<th>Func 5</th>
<th>Func 6</th>
<th>Func 7</th>
<th>Func 9</th>
<th>Func 10</th>
<th>Func 11</th>
<th>Func 12</th>
<th>Func 15</th>
</tr>
</thead>
<tbody>
<tr>
<td>BLX-GL50</td>
<td>100%</td>
<td>100%</td>
<td>0%</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td>36%</td>
<td>12%</td>
<td>0%</td>
<td>0%</td>
<td>52%</td>
<td>0%</td>
</tr>
<tr>
<td>BLX-MA</td>
<td>100%</td>
<td>100%</td>
<td>0%</td>
<td>0.96</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>72%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>20%</td>
</tr>
<tr>
<td>CoEVO</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
</tr>
<tr>
<td>DE</td>
<td>100%</td>
<td>100%</td>
<td>80%</td>
<td>100%</td>
<td>100%</td>
<td>96%</td>
<td>6%</td>
<td>44%</td>
<td>0%</td>
<td>48%</td>
<td>76%</td>
<td>4%</td>
</tr>
<tr>
<td>DMS-L-PSO</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td>16%</td>
<td>100%</td>
<td>0%</td>
<td>0%</td>
<td>80%</td>
<td>84%</td>
</tr>
<tr>
<td>EDA</td>
<td>100%</td>
<td>100%</td>
<td>92%</td>
<td>100%</td>
<td>100%</td>
<td>88%</td>
<td>4%</td>
<td>0%</td>
<td>0%</td>
<td>12%</td>
<td>40%</td>
<td>0%</td>
</tr>
<tr>
<td>G-CMA-ES</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td>76%</td>
<td>92%</td>
<td>24%</td>
<td>88%</td>
<td>0%</td>
</tr>
<tr>
<td>K-PCX</td>
<td>100%</td>
<td>100%</td>
<td>0%</td>
<td>84%</td>
<td>0%</td>
<td>40%</td>
<td>20%</td>
<td>96%</td>
<td>88%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
</tr>
<tr>
<td>L-CMA-ES</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td>28%</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>48%</td>
<td>0%</td>
</tr>
<tr>
<td>L-SaDE</td>
<td>100%</td>
<td>100%</td>
<td>64%</td>
<td>96%</td>
<td>0%</td>
<td>100%</td>
<td>24%</td>
<td>100%</td>
<td>0%</td>
<td>0%</td>
<td>100%</td>
<td>92%</td>
</tr>
<tr>
<td>SPC-PNX</td>
<td>100%</td>
<td>100%</td>
<td>0%</td>
<td>100%</td>
<td>100%</td>
<td>0%</td>
<td>4%</td>
<td>4%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
</tr>
</tbody>
</table>

*In the comparison, only the problems in which at least one algorithm succeeded once are considered.*
CEC’05 Comparison Results

First three algorithms:
1. G-CMA-ES
2. DE
3. DMS-L-PSO

Empirical distribution over all successful functions for 10-D (SP here means the Success Performance for each problem. \( SP = \text{mean (FEs for successful runs)} \times (\text{# of total runs}) / (\text{# of successful runs}). SP_{best} \) is the minimal FES of all algorithms for each problem.)
**CEC’05 Comparison Results**

Success Rates of the 11 algorithms for 30-D

<table>
<thead>
<tr>
<th>Algorithms</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
</tr>
</thead>
<tbody>
<tr>
<td>BLX-GL50</td>
<td>100%</td>
<td>100%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>100%</td>
<td>100%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
</tr>
<tr>
<td>BLX-MA</td>
<td>100%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>36%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
</tr>
<tr>
<td>CoEVO</td>
<td>12%</td>
<td>32%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>44%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
</tr>
<tr>
<td>DE</td>
<td>100%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>88%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
</tr>
<tr>
<td>DMS-L-PSO</td>
<td>100%</td>
<td>100%</td>
<td>88%</td>
<td>0%</td>
<td>0%</td>
<td>96%</td>
<td>96%</td>
<td>100%</td>
<td>0%</td>
<td>0%</td>
<td>20%</td>
</tr>
<tr>
<td>EDA</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td>0%</td>
<td>0%</td>
<td>100%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
</tr>
<tr>
<td>G-CMA-ES</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td>40%</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td>36%</td>
<td>12%</td>
<td>4%</td>
<td>32%</td>
</tr>
<tr>
<td>K-PCX</td>
<td>100%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>44%</td>
<td>72%</td>
<td>56%</td>
<td>0%</td>
<td>0%</td>
</tr>
<tr>
<td>L-CMA-ES</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td>0%</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
</tr>
<tr>
<td>L-SaDE</td>
<td>100%</td>
<td>96%</td>
<td>0%</td>
<td>52%</td>
<td>0%</td>
<td>0%</td>
<td>80%</td>
<td>100%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
</tr>
<tr>
<td>SPC-PNX</td>
<td>100%</td>
<td>88%</td>
<td>0%</td>
<td>76%</td>
<td>0%</td>
<td>4%</td>
<td>64%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
</tr>
</tbody>
</table>

*In the comparison, only the problems which at least one algorithm succeeded once are considered.*
CEC’05 Comparison Results

Empirical distribution over all successful functions for 30-D (SP here means the Success Performance for each problem. \( SP = \text{mean} (\text{FEs for successful runs}) \times (\text{# of total runs}) / (\text{# of successful runs}) \). \( SP_{best} \) is the minimal FES of all algorithms for each problem.)

First three algorithms:
1. G-CMA-ES
2. DMS-L-PSO
3. L-CMA-ES
CEC’06 Comparison Results

- Algorithms

- DE (Zielinski & Laur, 2006): Differential Evolution
- DMS-C-PSO (Liang & Suganthan, 2006): Dynamic Multi-Swarm Particle Swarm Optimizer with the New Constraint-Handling Mechanism
- ε - DE [TS06] Constrained Differential Evolution with Gradient-Based Mutation and Feasible Elites
- GDE (Kukkonen & Lampinen, 2006): Generalized Differential Evolution
- MPDE (Tasgetiren & Suganthan, 2006): Multi-Populated DE Algorithm
- PESO+ (Munoz-Żavala et al, 2006): Particle Evolutionary Swarm Optimization Plus
CEC’06 Comparison Results

- Problems: 24 minimization problems with constraints (Liang, 2006b)
- Runs / problem: 25 (total runs)
- Max_FES: 500,000
- Feasible Rate = (# of feasible runs) / total runs
- Success Rate = (# of successful runs) / total runs
- Success Performance = mean (FEs for successful runs) * (# of total runs) / (# of successful runs)

The above three quantities are computed for each problem separately.

- Feasible Run: A run during which at least one feasible solution is found in Max_FES.
- Successful Run: A run during which the algorithm finds a feasible solution \( x \) satisfying \( f(x) - f(x^*) \leq 0.0001 \)
# CEC’06 Comparison Results

## Algorithms’ Parameters

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>DE</td>
<td>NP, F, CR</td>
</tr>
<tr>
<td>DMS-PSO</td>
<td>$\omega, c_1, c_2, \text{Vmax}, n, ns, R, L, L_{FES}$</td>
</tr>
<tr>
<td>ε DE</td>
<td>N, F, CR, Tc, Tmax, cp, Pg, Rg, Ne</td>
</tr>
<tr>
<td>GDE</td>
<td>NP, F, CR</td>
</tr>
<tr>
<td>jDE-2</td>
<td>NP, F, CR, k, l</td>
</tr>
<tr>
<td>MDE</td>
<td>$\mu, \text{CR}, \text{Max_Gen}, \lambda, F_\alpha, F_\beta$</td>
</tr>
<tr>
<td>MPDE</td>
<td>F, CR, np1, np2</td>
</tr>
<tr>
<td>PCX</td>
<td>N, $\lambda$, r (a different N is used for g02),</td>
</tr>
<tr>
<td>PESO+</td>
<td>$\omega, c_1, c_2, n$, not sensitive to $\omega$, c1, c2</td>
</tr>
<tr>
<td>SaDE</td>
<td>NP, LP, LS_gap</td>
</tr>
</tbody>
</table>

-100-
# CEC’06 Comparison Results

## Success Rate for Problems 1-11

<table>
<thead>
<tr>
<th>Algorithms</th>
<th>Func</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
</tr>
</thead>
<tbody>
<tr>
<td>DE</td>
<td>81.64%</td>
<td>100%</td>
<td>84%</td>
<td>0%</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
</tr>
<tr>
<td>DMS-C-PSO</td>
<td>95.27%</td>
<td>100%</td>
<td>96%</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
</tr>
<tr>
<td>e-DE</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
</tr>
<tr>
<td>GDE</td>
<td>80.91%</td>
<td>100%</td>
<td>72%</td>
<td>4%</td>
<td>100%</td>
<td>92%</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
</tr>
<tr>
<td>jDE-2</td>
<td>83.64%</td>
<td>100%</td>
<td>92%</td>
<td>0%</td>
<td>100%</td>
<td>68%</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td>96%</td>
</tr>
<tr>
<td>MDE</td>
<td>91.64%</td>
<td>100%</td>
<td>16%</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
</tr>
<tr>
<td>MPDE</td>
<td>91.64%</td>
<td>100%</td>
<td>92%</td>
<td>84%</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td>96%</td>
</tr>
<tr>
<td>PCX</td>
<td>98.36%</td>
<td>100%</td>
<td>64%</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
</tr>
<tr>
<td>PESO+</td>
<td>70.91%</td>
<td>100%</td>
<td>56%</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td>96%</td>
<td>100%</td>
<td>100%</td>
<td>16%</td>
<td>100%</td>
<td>100%</td>
</tr>
<tr>
<td>SaDE</td>
<td>91.09%</td>
<td>100%</td>
<td>84%</td>
<td>96%</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
</tr>
</tbody>
</table>
### CEC’06 Comparison Results

Success Rate for Problems 12-19, 21, 23, 24

<table>
<thead>
<tr>
<th>Algorithms</th>
<th>Func 12</th>
<th>Func 13</th>
<th>Func 14</th>
<th>Func 15</th>
<th>Func 16</th>
<th>Func 17</th>
<th>Func 18</th>
<th>Func 19</th>
<th>Func 21</th>
<th>Func 23</th>
<th>Func 24</th>
</tr>
</thead>
<tbody>
<tr>
<td>DE</td>
<td>100%</td>
<td>32%</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td>20%</td>
<td>100%</td>
<td>100%</td>
<td>60%</td>
<td>0%</td>
<td>100%</td>
</tr>
<tr>
<td>DMS-C-PSO</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td>0%</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
</tr>
<tr>
<td>ε-DE</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
</tr>
<tr>
<td>GDE</td>
<td>100%</td>
<td>40%</td>
<td>96%</td>
<td>96%</td>
<td>100%</td>
<td>16%</td>
<td>76%</td>
<td>88%</td>
<td>60%</td>
<td>40%</td>
<td>100%</td>
</tr>
<tr>
<td>jDE-2</td>
<td>100%</td>
<td>0%</td>
<td>100%</td>
<td>96%</td>
<td>100%</td>
<td>4%</td>
<td>100%</td>
<td>100%</td>
<td>92%</td>
<td>92%</td>
<td>100%</td>
</tr>
<tr>
<td>MDE</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td>0%</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
</tr>
<tr>
<td>MPDE</td>
<td>100%</td>
<td>48%</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td>28%</td>
<td>100%</td>
<td>100%</td>
<td>68%</td>
<td>100%</td>
<td>100%</td>
</tr>
<tr>
<td>PCX</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
</tr>
<tr>
<td>PESO+</td>
<td>100%</td>
<td>100%</td>
<td>0%</td>
<td>100%</td>
<td>100%</td>
<td>0%</td>
<td>92%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>100%</td>
</tr>
<tr>
<td>SaDE</td>
<td>100%</td>
<td>100%</td>
<td>80%</td>
<td>100%</td>
<td>100%</td>
<td>4%</td>
<td>92%</td>
<td>100%</td>
<td>60%</td>
<td>8%</td>
<td>100%</td>
</tr>
</tbody>
</table>
CEC’06 Comparison Results

Empirical distribution over all functions (SP here means the Success Performance for each problem. SP = mean (FEs for successful runs) * (# of total runs) / (# of successful runs). SP_{best} is the minimal FES of all algorithms for each problem.)
Results of MOCLPSO on ZDT1

<table>
<thead>
<tr>
<th>Test Problem (ZDT1)</th>
<th>Converge Metric</th>
<th>Diversity Metric Δ</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>MOCLPSO</td>
<td>MOPSO</td>
</tr>
<tr>
<td>Best</td>
<td>0.001945</td>
<td>0.063836</td>
</tr>
<tr>
<td>Worst</td>
<td>0.002455</td>
<td>0.124011</td>
</tr>
<tr>
<td>Average</td>
<td><strong>0.002235</strong></td>
<td>0.081128</td>
</tr>
<tr>
<td>Median</td>
<td>0.002232</td>
<td>0.072365</td>
</tr>
<tr>
<td>Variance</td>
<td>2.69E-08</td>
<td>4.30E-04</td>
</tr>
</tbody>
</table>

NSGA-II [D01]; MOPSO [CL04]; MOCLPSO [HSL06]
Results of MOCLPSO on ZDT3

<table>
<thead>
<tr>
<th>Test Problem (ZDT3)</th>
<th>Converge Metric</th>
<th>Diversity Metric Δ</th>
</tr>
</thead>
<tbody>
<tr>
<td>Best</td>
<td>MOCLPSO</td>
<td>0.503779</td>
</tr>
<tr>
<td>Worst</td>
<td>MOPSO</td>
<td>0.646290</td>
</tr>
<tr>
<td>Average</td>
<td>NSGA-II</td>
<td>0.554944</td>
</tr>
<tr>
<td>Median</td>
<td>MOCLPSO</td>
<td>0.549698</td>
</tr>
<tr>
<td>Variance</td>
<td>MOPSO</td>
<td>0.607451</td>
</tr>
<tr>
<td></td>
<td>NSGA-II</td>
<td>0.795274</td>
</tr>
</tbody>
</table>

Converge Metric:
- MOCLPSO
- MOPSO
- NSGA-II

Diversity Metric Δ:
- MOCLPSO
- MOPSO
- NSGA-II
Results of MOCLPSO on ZDT6

<table>
<thead>
<tr>
<th>Test Problem (ZDT6)</th>
<th>Converge Metric $\psi$</th>
<th>Diversity Metric $\Delta$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Best</td>
<td>MOCLPSO: 0.002950</td>
<td>MOCLPSO: 0.237585</td>
</tr>
<tr>
<td></td>
<td>MOPSO: 0.527778</td>
<td>MOPSO: 0.788246</td>
</tr>
<tr>
<td></td>
<td>NSGA-II: 2.498893</td>
<td>NSGA-II: 0.934297</td>
</tr>
<tr>
<td>Worst</td>
<td>MOCLPSO: 0.015873</td>
<td>MOCLPSO: 0.964462</td>
</tr>
<tr>
<td></td>
<td>MOPSO: 1.667019</td>
<td>MOPSO: 1.016125</td>
</tr>
<tr>
<td></td>
<td>NSGA-II: 3.707603</td>
<td>NSGA-II: 0.983176</td>
</tr>
<tr>
<td>Average</td>
<td>MOCLPSO: 0.006283</td>
<td>MOCLPSO: 0.486495</td>
</tr>
<tr>
<td></td>
<td>MOPSO: 1.029723</td>
<td>MOPSO: 0.927657</td>
</tr>
<tr>
<td></td>
<td>NSGA-II: 3.073925</td>
<td>NSGA-II: 0.954907</td>
</tr>
<tr>
<td>Median</td>
<td>MOCLPSO: 0.004871</td>
<td>MOCLPSO: 0.346380</td>
</tr>
<tr>
<td></td>
<td>MOPSO: 0.944908</td>
<td>MOPSO: 0.929350</td>
</tr>
<tr>
<td></td>
<td>NSGA-II: 3.023464</td>
<td>NSGA-II: 0.954590</td>
</tr>
<tr>
<td>Variance</td>
<td>MOCLPSO: 1.77E-05</td>
<td>MOCLPSO: 6.75E-02</td>
</tr>
<tr>
<td></td>
<td>MOPSO: 1.90E-01</td>
<td>MOPSO: 5.30E-03</td>
</tr>
<tr>
<td></td>
<td>NSGA-II: 1.40E-01</td>
<td>NSGA-II: 3.00E-04</td>
</tr>
</tbody>
</table>

NSGA-II [D01]; MOPSO [CL04]; MOCLPSO [HSL06]
Results of MO-DMS-PSO

Convergence Metric ($\gamma$) comparison of the four algorithms

<table>
<thead>
<tr>
<th>Algorithms</th>
<th>SCH</th>
<th>FON</th>
<th>KUR</th>
<th>ZDT1</th>
<th>ZDT2</th>
<th>ZDT3</th>
<th>ZDT4</th>
<th>ZDT6</th>
</tr>
</thead>
<tbody>
<tr>
<td>NSGA-II</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>mean</td>
<td>0.0043</td>
<td>0.0021</td>
<td>0.0324</td>
<td>0.0674</td>
<td>0.1897</td>
<td>0.6211</td>
<td>5.1219</td>
<td>3.1209</td>
</tr>
<tr>
<td>std</td>
<td>0.0004</td>
<td>0.0002</td>
<td>0.1074</td>
<td>0.0246</td>
<td>0.0615</td>
<td>0.0329</td>
<td>2.2526</td>
<td>0.3413</td>
</tr>
<tr>
<td>PAES</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>mean</td>
<td>0.0045</td>
<td>0.0360</td>
<td>1.0955</td>
<td><strong>0.0006</strong></td>
<td><strong>0.0005</strong></td>
<td>0.0745</td>
<td>3.5097</td>
<td>7.5964</td>
</tr>
<tr>
<td>std</td>
<td>0.0004</td>
<td>0.1315</td>
<td>2.1724</td>
<td>0.0003</td>
<td>0.0003</td>
<td>0.0034</td>
<td>1.0967</td>
<td>0.8102</td>
</tr>
<tr>
<td>MOPSO</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>mean</td>
<td>0.0044</td>
<td>0.0013</td>
<td>0.0252</td>
<td>0.0189</td>
<td>0.0162</td>
<td>0.0267</td>
<td>5.6413</td>
<td>0.7501</td>
</tr>
<tr>
<td>std</td>
<td>0.0004</td>
<td>0.0001</td>
<td>0.0041</td>
<td>0.0032</td>
<td>0.0099</td>
<td>0.0069</td>
<td>2.7814</td>
<td>0.4208</td>
</tr>
<tr>
<td>DMS</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>mean</td>
<td><strong>0.0044</strong></td>
<td><strong>0.0012</strong></td>
<td><strong>0.0162</strong></td>
<td>0.0018</td>
<td>0.0016</td>
<td><strong>0.0018</strong></td>
<td>0.0018</td>
<td>0.0039</td>
</tr>
<tr>
<td>std</td>
<td>0.0003</td>
<td>0.0001</td>
<td>0.0015</td>
<td>0.0003</td>
<td>0.0004</td>
<td>0.0004</td>
<td>0.0008</td>
<td>0.0012</td>
</tr>
</tbody>
</table>

$h$ = 0 1 1 1 1 1 1 1 1 1

NSGA-II [D01]; MOPSO [CL04]; PAES [KC00]
### Results of MO-DMS-PSO

#### Diversity Metric (Δ) comparison of the four algorithms

<table>
<thead>
<tr>
<th>Algorithms</th>
<th>SCH</th>
<th>FON</th>
<th>KUR</th>
<th>ZDT1</th>
<th>ZDT2</th>
<th>ZDT3</th>
<th>ZDT4</th>
<th>ZDT6</th>
</tr>
</thead>
<tbody>
<tr>
<td>NSGA-II</td>
<td>0.2823</td>
<td>0.4470</td>
<td>0.7680</td>
<td>0.5401</td>
<td>0.9482</td>
<td>0.7958</td>
<td>0.9445</td>
<td>0.9639</td>
</tr>
<tr>
<td></td>
<td>0.0269</td>
<td>0.0335</td>
<td>0.0533</td>
<td>0.0454</td>
<td>0.1344</td>
<td>0.0148</td>
<td>0.0590</td>
<td>0.0212</td>
</tr>
<tr>
<td>PAES</td>
<td>0.7416</td>
<td>0.5533</td>
<td>0.6896</td>
<td>1.1197</td>
<td>1.1539</td>
<td>0.9707</td>
<td>1.0108</td>
<td>0.8900</td>
</tr>
<tr>
<td></td>
<td>0.0427</td>
<td>0.1276</td>
<td>0.0840</td>
<td>0.1561</td>
<td>0.1849</td>
<td>0.0331</td>
<td>0.1671</td>
<td>0.0815</td>
</tr>
<tr>
<td>MOPSO</td>
<td>0.7618</td>
<td>0.5861</td>
<td>0.7849</td>
<td>0.5980</td>
<td>0.6810</td>
<td>0.7203</td>
<td>0.9720</td>
<td>1.0054</td>
</tr>
<tr>
<td></td>
<td>0.0517</td>
<td>0.0413</td>
<td>0.0961</td>
<td>0.0470</td>
<td>0.2006</td>
<td>0.0322</td>
<td>0.0310</td>
<td>0.0741</td>
</tr>
<tr>
<td>DMS</td>
<td>0.1723</td>
<td>0.1371</td>
<td>0.2665</td>
<td>0.1615</td>
<td>0.1835</td>
<td>0.5015</td>
<td>0.1647</td>
<td>0.1573</td>
</tr>
<tr>
<td></td>
<td>0.0110</td>
<td>0.0142</td>
<td>0.0125</td>
<td>0.0109</td>
<td>0.1547</td>
<td>0.0155</td>
<td>0.0124</td>
<td>0.0129</td>
</tr>
</tbody>
</table>

| h         | 1     | 1     | 1     | 1      | 1      | 1      | 1      | 1      |

*Note: Values represent the mean or standard deviation (std) for each algorithm across different test cases.*
Results of MO-DMS-PSO

(a) NSGAII

(b) PAES

(c) MOPSO

(d) DMS-MOPSO

Fig. 6-14 Pareto fronts generated by the four algorithms on ZDT6
CEC07 comparison results on MOEAs: Evaluation Criteria

- Quantitative performance measurements, R indicator and Hypervolume difference to a reference set is used as a measure for the expected number of function evaluations to reach a target Pareto front.

- Invariance is a non-empirical statement on the ability to generalize performance results. Invariance guarantees identical performance on a class of functions. Possible invariances are invariance against translation, scaling, or even order preserving transformations of the objective function value invariance against angle preserving (rigid) transformations of the search space (translation, rotation)

- Parameters Settings
  - how many parameters of the algorithm need to be adjusted to the object function?
  - how many different settings were tested?
  - how many different settings were finally used?
References to Algorithms in CEC07 papers

☐ NSGAII_SBX: Sharma, Kumar et al.
☐ NSGAII_PCX: Kumar et al.
☐ GDE3: Kukkonen and Lampinen
☐ DEMOwSA: Zamuda et al.
☐ MOSaDE: Huang et al.
☐ MO_DE: Zielinski and Laur
☐ MO_PSO: Zielinski and Laur
☐ MTS: Tseng and Chen
CEC07 Function Sets

- Three subsets
  - 2-objective functions
  - 3-objective functions
  - 5-objective functions

- Comparison: Rank of the mean of the metric values from 25 runs
### M=2, Rank(R indicator)

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>NSGA2_SBX</td>
<td>1.71</td>
<td>2</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>NSGA2_PCX</td>
<td>5.43</td>
<td>4</td>
<td>1</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>1</td>
<td>8</td>
</tr>
<tr>
<td>GDE3</td>
<td>4.14</td>
<td>1</td>
<td>7</td>
<td>4</td>
<td>4</td>
<td>5</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>DEMOwSA</td>
<td>6.43</td>
<td>3</td>
<td>8</td>
<td>6</td>
<td>7</td>
<td>7</td>
<td>7</td>
<td>7</td>
</tr>
<tr>
<td>MOSaDE</td>
<td>4.14</td>
<td>8</td>
<td>4</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>5</td>
<td>3</td>
</tr>
<tr>
<td>MO_DE</td>
<td>5.71</td>
<td>5</td>
<td>6</td>
<td>5</td>
<td>5</td>
<td>6</td>
<td>8</td>
<td>5</td>
</tr>
<tr>
<td>MO_PSO</td>
<td>5.00</td>
<td>7</td>
<td>2</td>
<td>7</td>
<td>6</td>
<td>4</td>
<td>3</td>
<td>6</td>
</tr>
<tr>
<td>MTS</td>
<td>3.43</td>
<td>6</td>
<td>5</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>6</td>
<td>2</td>
</tr>
</tbody>
</table>

**FES=5000**

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>NSGA2_SBX</td>
<td>3.14</td>
<td>2</td>
<td>5</td>
<td>2</td>
<td>2</td>
<td>6</td>
<td>2</td>
<td>5</td>
</tr>
<tr>
<td>NSGA2_PCX</td>
<td>4.86</td>
<td>1</td>
<td>7</td>
<td>6</td>
<td>3</td>
<td>3</td>
<td>4</td>
<td>7</td>
</tr>
<tr>
<td>GDE3</td>
<td>2.43</td>
<td>3</td>
<td>2</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>DEMOwSA</td>
<td>4.71</td>
<td>4</td>
<td>4</td>
<td>3</td>
<td>6</td>
<td>7</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>MOSaDE</td>
<td>3.57</td>
<td>8</td>
<td>6</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>MO_DE</td>
<td>4.14</td>
<td>5</td>
<td>3</td>
<td>5</td>
<td>7</td>
<td>6</td>
<td>5</td>
<td>4</td>
</tr>
<tr>
<td>MO_PSO</td>
<td>7.29</td>
<td>7</td>
<td>1</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>7</td>
<td>8</td>
</tr>
<tr>
<td>MTS</td>
<td>5.86</td>
<td>6</td>
<td>8</td>
<td>7</td>
<td>5</td>
<td>5</td>
<td>8</td>
<td>6</td>
</tr>
</tbody>
</table>

**FES=50000**

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>NSGA2_SBX</td>
<td>3.43</td>
<td>2</td>
<td>5</td>
<td>2</td>
<td>2</td>
<td>6</td>
<td>2</td>
<td>5</td>
</tr>
<tr>
<td>NSGA2_PCX</td>
<td>4.43</td>
<td>1</td>
<td>7</td>
<td>6</td>
<td>3</td>
<td>3</td>
<td>4</td>
<td>7</td>
</tr>
<tr>
<td>GDE3</td>
<td>3.14</td>
<td>3</td>
<td>2</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>DEMOwSA</td>
<td>4.14</td>
<td>4</td>
<td>4</td>
<td>3</td>
<td>6</td>
<td>7</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>MOSaDE</td>
<td>2.71</td>
<td>8</td>
<td>6</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>MO_DE</td>
<td>5.00</td>
<td>5</td>
<td>3</td>
<td>5</td>
<td>7</td>
<td>6</td>
<td>5</td>
<td>4</td>
</tr>
<tr>
<td>MO_PSO</td>
<td>6.71</td>
<td>7</td>
<td>1</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>7</td>
<td>8</td>
</tr>
<tr>
<td>MTS</td>
<td>6.43</td>
<td>6</td>
<td>8</td>
<td>7</td>
<td>5</td>
<td>5</td>
<td>8</td>
<td>6</td>
</tr>
</tbody>
</table>
### M=2, Rank(Hypervolumn)

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>NSGA2_SBX</td>
<td>1.86</td>
<td>2</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>NSGA2(PCX)</td>
<td>5.29</td>
<td>3</td>
<td>1</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>1</td>
<td>8</td>
</tr>
<tr>
<td>GDE3</td>
<td>4.43</td>
<td>1</td>
<td>7</td>
<td>4</td>
<td>4</td>
<td>6</td>
<td>4</td>
<td>5</td>
</tr>
<tr>
<td>DEMOwSA</td>
<td>6.71</td>
<td>4</td>
<td>8</td>
<td>7</td>
<td>7</td>
<td>7</td>
<td>7</td>
<td>7</td>
</tr>
<tr>
<td>MOSaDE</td>
<td>4.29</td>
<td>8</td>
<td>4</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>6</td>
<td>3</td>
</tr>
<tr>
<td>MO_DE</td>
<td>5.43</td>
<td>5</td>
<td>6</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>8</td>
<td>4</td>
</tr>
<tr>
<td>MO_PSO</td>
<td>4.71</td>
<td>6</td>
<td>2</td>
<td>6</td>
<td>6</td>
<td>4</td>
<td>3</td>
<td>6</td>
</tr>
<tr>
<td>MTS</td>
<td>3.29</td>
<td>7</td>
<td>5</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>5</td>
<td>1</td>
</tr>
</tbody>
</table>

**FES=5000**

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>NSGA2_SBX</td>
<td>2.86</td>
<td>2</td>
<td>7</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>4</td>
<td>3</td>
</tr>
<tr>
<td>NSGA2(PCX)</td>
<td>4.43</td>
<td>3</td>
<td>3</td>
<td>6</td>
<td>4</td>
<td>5</td>
<td>3</td>
<td>7</td>
</tr>
<tr>
<td>GDE3</td>
<td>1.71</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>4</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>DEMOwSA</td>
<td>4.43</td>
<td>5</td>
<td>2</td>
<td>3</td>
<td>5</td>
<td>7</td>
<td>5</td>
<td>4</td>
</tr>
<tr>
<td>MOSaDE</td>
<td>5.29</td>
<td>8</td>
<td>6</td>
<td>7</td>
<td>6</td>
<td>3</td>
<td>2</td>
<td>5</td>
</tr>
<tr>
<td>MO_DE</td>
<td>4.57</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>7</td>
<td>6</td>
<td>6</td>
<td>1</td>
</tr>
<tr>
<td>MO_PSO</td>
<td>7.29</td>
<td>7</td>
<td>5</td>
<td>8</td>
<td>8</td>
<td>7</td>
<td>8</td>
<td>8</td>
</tr>
<tr>
<td>MTS</td>
<td>5.43</td>
<td>6</td>
<td>8</td>
<td>5</td>
<td>3</td>
<td>2</td>
<td>8</td>
<td>6</td>
</tr>
</tbody>
</table>

**FES=50000**

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>NSGA2_SBX</td>
<td>3.71</td>
<td>2</td>
<td>6</td>
<td>4</td>
<td>1</td>
<td>2</td>
<td>6</td>
<td>5</td>
</tr>
<tr>
<td>NSGA2(PCX)</td>
<td>4.57</td>
<td>3</td>
<td>7</td>
<td>5</td>
<td>2</td>
<td>4</td>
<td>4</td>
<td>7</td>
</tr>
<tr>
<td>GDE3</td>
<td>2.43</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>4</td>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>DEMOwSA</td>
<td>4.00</td>
<td>6</td>
<td>4</td>
<td>1</td>
<td>5</td>
<td>7</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>MOSaDE</td>
<td>4.43</td>
<td>8</td>
<td>5</td>
<td>8</td>
<td>7</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>MO_DE</td>
<td>4.43</td>
<td>4</td>
<td>3</td>
<td>3</td>
<td>6</td>
<td>6</td>
<td>5</td>
<td>4</td>
</tr>
<tr>
<td>MO_PSO</td>
<td>6.57</td>
<td>7</td>
<td>1</td>
<td>7</td>
<td>8</td>
<td>8</td>
<td>7</td>
<td>8</td>
</tr>
<tr>
<td>MTS</td>
<td>5.86</td>
<td>5</td>
<td>8</td>
<td>6</td>
<td>3</td>
<td>5</td>
<td>8</td>
<td>6</td>
</tr>
</tbody>
</table>
### M=3, Rank(R indicator)

#### FES=5000

<table>
<thead>
<tr>
<th></th>
<th>Total</th>
<th>8, S_DTLZ2</th>
<th>9, R_DTLZ2</th>
<th>10, S_DTLZ3</th>
<th>11, WFG1</th>
<th>12, WFG8</th>
<th>13, WFG9</th>
</tr>
</thead>
<tbody>
<tr>
<td>NSGA2_SBX</td>
<td>1.46</td>
<td>1</td>
<td>4</td>
<td>1</td>
<td>4</td>
<td>7</td>
<td>2</td>
</tr>
<tr>
<td>NSGA2_PCX</td>
<td>1.85</td>
<td>8</td>
<td>5</td>
<td>3</td>
<td>2</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>GDE3</td>
<td>1.92</td>
<td>5</td>
<td>3</td>
<td>5</td>
<td>6</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>DEMOwSA</td>
<td><strong>1.31</strong></td>
<td>4</td>
<td>8</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>MOSaDE</td>
<td>3.08</td>
<td>2</td>
<td>7</td>
<td>7</td>
<td>8</td>
<td>8</td>
<td>8</td>
</tr>
<tr>
<td>MO_DE</td>
<td>2.31</td>
<td>3</td>
<td>6</td>
<td>6</td>
<td>5</td>
<td>4</td>
<td>6</td>
</tr>
<tr>
<td>MO_PSO</td>
<td>2.46</td>
<td>6</td>
<td>1</td>
<td>8</td>
<td>7</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>MTS</td>
<td>2.23</td>
<td>7</td>
<td>2</td>
<td>4</td>
<td>3</td>
<td>6</td>
<td>7</td>
</tr>
</tbody>
</table>

#### FES=50000

<table>
<thead>
<tr>
<th></th>
<th>Total</th>
<th>8, S_DTLZ2</th>
<th>9, R_DTLZ2</th>
<th>10, S_DTLZ3</th>
<th>11, WFG1</th>
<th>12, WFG8</th>
<th>13, WFG9</th>
</tr>
</thead>
<tbody>
<tr>
<td>NSGA2_SBX</td>
<td>1.38</td>
<td>5</td>
<td>2</td>
<td>1</td>
<td>3</td>
<td>6</td>
<td>1</td>
</tr>
<tr>
<td>NSGA2_PCX</td>
<td>2.15</td>
<td>7</td>
<td>3</td>
<td>5</td>
<td>5</td>
<td>3</td>
<td>5</td>
</tr>
<tr>
<td>GDE3</td>
<td><strong>1.00</strong></td>
<td>3</td>
<td>1</td>
<td>2</td>
<td>4</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>DEMOwSA</td>
<td>1.62</td>
<td>2</td>
<td>8</td>
<td>4</td>
<td>2</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>MOSaDE</td>
<td>2.85</td>
<td>1</td>
<td>7</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>8</td>
</tr>
<tr>
<td>MO_DE</td>
<td>2.38</td>
<td>4</td>
<td>6</td>
<td>7</td>
<td>6</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>MO_PSO</td>
<td>2.92</td>
<td>6</td>
<td>5</td>
<td>8</td>
<td>8</td>
<td>5</td>
<td>6</td>
</tr>
<tr>
<td>MTS</td>
<td>2.31</td>
<td>8</td>
<td>4</td>
<td>3</td>
<td>1</td>
<td>7</td>
<td>7</td>
</tr>
</tbody>
</table>

#### FES=500000

<table>
<thead>
<tr>
<th></th>
<th>Total</th>
<th>8, S_DTLZ2</th>
<th>9, R_DTLZ2</th>
<th>10, S_DTLZ3</th>
<th>11, WFG1</th>
<th>12, WFG8</th>
<th>13, WFG9</th>
</tr>
</thead>
<tbody>
<tr>
<td>NSGA2_SBX</td>
<td>1.38</td>
<td>5</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>6</td>
<td>3</td>
</tr>
<tr>
<td>NSGA2_PCX</td>
<td>2.23</td>
<td>7</td>
<td>3</td>
<td>6</td>
<td>6</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>GDE3</td>
<td><strong>1.00</strong></td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>5</td>
</tr>
<tr>
<td>DEMOwSA</td>
<td>2.08</td>
<td>3</td>
<td>8</td>
<td>4</td>
<td>4</td>
<td>2</td>
<td>6</td>
</tr>
<tr>
<td>MOSaDE</td>
<td>2.15</td>
<td>1</td>
<td>5</td>
<td>3</td>
<td>3</td>
<td>8</td>
<td>8</td>
</tr>
<tr>
<td>MO_DE</td>
<td>2.23</td>
<td>4</td>
<td>6</td>
<td>7</td>
<td>7</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>MO_PSO</td>
<td>2.77</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>8</td>
<td>5</td>
<td>2</td>
</tr>
<tr>
<td>MTS</td>
<td><strong>2.77</strong></td>
<td>8</td>
<td>4</td>
<td>5</td>
<td>5</td>
<td>7</td>
<td>7</td>
</tr>
<tr>
<td>----------</td>
<td>-------</td>
<td>------------</td>
<td>------------</td>
<td>-------------</td>
<td>-----------</td>
<td>---------</td>
<td>---------</td>
</tr>
<tr>
<td>NSGA2_SBX</td>
<td>1.31</td>
<td>1</td>
<td>3</td>
<td>2</td>
<td>8</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>NSGA2_PCX</td>
<td>2.15</td>
<td>8</td>
<td>5</td>
<td>5</td>
<td>2</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>GDE3</td>
<td>1.69</td>
<td>4</td>
<td>4</td>
<td>3</td>
<td>5</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>DEMO_wSA</td>
<td>1.54</td>
<td>6</td>
<td>6</td>
<td>4</td>
<td>1</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>MOSaDE</td>
<td>3.23</td>
<td>5</td>
<td>8</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>8</td>
</tr>
<tr>
<td>MO_DE</td>
<td>2.38</td>
<td>2</td>
<td>7</td>
<td>7</td>
<td>4</td>
<td>5</td>
<td>6</td>
</tr>
<tr>
<td>MO_PSO</td>
<td>2.31</td>
<td>3</td>
<td>2</td>
<td>8</td>
<td>6</td>
<td>6</td>
<td>5</td>
</tr>
<tr>
<td>MTS</td>
<td>2.00</td>
<td>7</td>
<td>1</td>
<td>1</td>
<td>3</td>
<td>7</td>
<td>7</td>
</tr>
</tbody>
</table>

**FES=5000**

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>NSGA2_SBX</td>
<td>1.62</td>
<td>6</td>
<td>2</td>
<td>1</td>
<td>8</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>NSGA2_PCX</td>
<td>2.23</td>
<td>7</td>
<td>3</td>
<td>5</td>
<td>4</td>
<td>4</td>
<td>6</td>
</tr>
<tr>
<td>GDE3</td>
<td>0.92</td>
<td>3</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>DEMO_wSA</td>
<td>1.69</td>
<td>2</td>
<td>8</td>
<td>6</td>
<td>1</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>MOSaDE</td>
<td>2.69</td>
<td>1</td>
<td>7</td>
<td>4</td>
<td>7</td>
<td>8</td>
<td>8</td>
</tr>
<tr>
<td>MO_DE</td>
<td>2.46</td>
<td>4</td>
<td>6</td>
<td>7</td>
<td>5</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>MO_PSO</td>
<td>2.62</td>
<td>5</td>
<td>5</td>
<td>8</td>
<td>6</td>
<td>6</td>
<td>4</td>
</tr>
<tr>
<td>MTS</td>
<td>2.38</td>
<td>8</td>
<td>4</td>
<td>3</td>
<td>2</td>
<td>7</td>
<td>7</td>
</tr>
</tbody>
</table>

**FES=50000**

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>NSGA2_SBX</td>
<td>1.77</td>
<td>6</td>
<td>1</td>
<td>2</td>
<td>8</td>
<td>1</td>
<td>5</td>
</tr>
<tr>
<td>NSGA2_PCX</td>
<td>2.31</td>
<td>7</td>
<td>3</td>
<td>5</td>
<td>5</td>
<td>4</td>
<td>6</td>
</tr>
<tr>
<td>GDE3</td>
<td>0.92</td>
<td>2</td>
<td>2</td>
<td>3</td>
<td>1</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>DEMO_wSA</td>
<td>1.92</td>
<td>3</td>
<td>8</td>
<td>4</td>
<td>3</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>MOSaDE</td>
<td>1.92</td>
<td>1</td>
<td>5</td>
<td>1</td>
<td>2</td>
<td>8</td>
<td>8</td>
</tr>
<tr>
<td>MO_DE</td>
<td>2.23</td>
<td>4</td>
<td>6</td>
<td>7</td>
<td>6</td>
<td>5</td>
<td>1</td>
</tr>
<tr>
<td>MO_PSO</td>
<td>2.77</td>
<td>5</td>
<td>7</td>
<td>8</td>
<td>7</td>
<td>6</td>
<td>3</td>
</tr>
<tr>
<td>MTS</td>
<td>2.77</td>
<td>8</td>
<td>4</td>
<td>6</td>
<td>4</td>
<td>7</td>
<td>7</td>
</tr>
</tbody>
</table>

**FES=500000**
### M=5, Rank(R indicator)

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>NSGA2_SBX</td>
<td>0.79</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>8</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>NSGA2_PCX</td>
<td>0.84</td>
<td>4</td>
<td>3</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>GDE3</td>
<td>1.42</td>
<td>5</td>
<td>5</td>
<td>4</td>
<td>5</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>DEMOwSA</td>
<td>1.89</td>
<td>8</td>
<td>8</td>
<td>5</td>
<td>2</td>
<td>6</td>
<td>7</td>
</tr>
<tr>
<td>MOSaDE</td>
<td>1.95</td>
<td>3</td>
<td>7</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>6</td>
</tr>
<tr>
<td>MO_DE</td>
<td>1.74</td>
<td>6</td>
<td>6</td>
<td>7</td>
<td>4</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>MO_PSO</td>
<td>2.11</td>
<td>7</td>
<td>4</td>
<td>8</td>
<td>6</td>
<td>7</td>
<td>8</td>
</tr>
<tr>
<td>MTS</td>
<td>0.63</td>
<td>2</td>
<td>2</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>3</td>
</tr>
</tbody>
</table>

#### FES=5000

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>NSGA2_SBX</td>
<td>0.58</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>6</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>NSGA2_PCX</td>
<td>1.63</td>
<td>8</td>
<td>4</td>
<td>8</td>
<td>5</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>GDE3</td>
<td>0.89</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>2</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>DEMOwSA</td>
<td>1.53</td>
<td>2</td>
<td>8</td>
<td>4</td>
<td>4</td>
<td>6</td>
<td>5</td>
</tr>
<tr>
<td>MOSaDE</td>
<td>2.05</td>
<td>4</td>
<td>6</td>
<td>5</td>
<td>8</td>
<td>8</td>
<td>8</td>
</tr>
<tr>
<td>MO_DE</td>
<td>1.47</td>
<td>6</td>
<td>7</td>
<td>6</td>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>MO_PSO</td>
<td>2.05</td>
<td>7</td>
<td>5</td>
<td>7</td>
<td>7</td>
<td>7</td>
<td>6</td>
</tr>
<tr>
<td>MTS</td>
<td>1.16</td>
<td>5</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>5</td>
<td>7</td>
</tr>
</tbody>
</table>

#### FES=50000

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>NSGA2_SBX</td>
<td>0.47</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>4</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>NSGA2_PCX</td>
<td>1.74</td>
<td>7</td>
<td>5</td>
<td>8</td>
<td>5</td>
<td>6</td>
<td>2</td>
</tr>
<tr>
<td>GDE3</td>
<td>0.84</td>
<td>4</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>DEMOwSA</td>
<td>1.58</td>
<td>2</td>
<td>8</td>
<td>4</td>
<td>7</td>
<td>4</td>
<td>5</td>
</tr>
<tr>
<td>MOSaDE</td>
<td>1.63</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>3</td>
<td>8</td>
<td>8</td>
</tr>
<tr>
<td>MO_DE</td>
<td>1.58</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>MO_PSO</td>
<td>2.16</td>
<td>8</td>
<td>7</td>
<td>7</td>
<td>8</td>
<td>5</td>
<td>6</td>
</tr>
<tr>
<td>MTS</td>
<td>1.37</td>
<td>5</td>
<td>2</td>
<td>3</td>
<td>2</td>
<td>7</td>
<td>7</td>
</tr>
<tr>
<td></td>
<td>Total</td>
<td>8, S_DTLZ2</td>
<td>9, R_DTLZ2</td>
<td>10, S_DTLZ3</td>
<td>11, WFG1</td>
<td>12, WFG8</td>
<td>13, WFG9</td>
</tr>
<tr>
<td>----------------</td>
<td>-------</td>
<td>------------</td>
<td>------------</td>
<td>-------------</td>
<td>----------</td>
<td>----------</td>
<td>----------</td>
</tr>
<tr>
<td>NSGA2_SBX</td>
<td>0.74</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>8</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>NSGA2_PCX</td>
<td>0.95</td>
<td>7</td>
<td>3</td>
<td>2</td>
<td>3</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>GDE3</td>
<td>1.16</td>
<td>2</td>
<td>5</td>
<td>4</td>
<td>5</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>DEMOwSA</td>
<td>1.68</td>
<td>8</td>
<td>8</td>
<td>6</td>
<td>6</td>
<td>4</td>
<td>5</td>
</tr>
<tr>
<td>MOSaDE</td>
<td>1.95</td>
<td>3</td>
<td>7</td>
<td>5</td>
<td>7</td>
<td>8</td>
<td>7</td>
</tr>
<tr>
<td>MO_DE</td>
<td>1.58</td>
<td>4</td>
<td>6</td>
<td>7</td>
<td>4</td>
<td>5</td>
<td>4</td>
</tr>
<tr>
<td>MO_PSO</td>
<td>1.74</td>
<td>5</td>
<td>2</td>
<td>8</td>
<td>6</td>
<td>6</td>
<td>6</td>
</tr>
<tr>
<td>MTS</td>
<td>1.58</td>
<td>6</td>
<td>4</td>
<td>3</td>
<td>2</td>
<td>7</td>
<td>8</td>
</tr>
</tbody>
</table>

**FES=5000**

<table>
<thead>
<tr>
<th></th>
<th>Total</th>
<th>8, S_DTLZ2</th>
<th>9, R_DTLZ2</th>
<th>10, S_DTLZ3</th>
<th>11, WFG1</th>
<th>12, WFG8</th>
<th>13, WFG9</th>
</tr>
</thead>
<tbody>
<tr>
<td>NSGA2_SBX</td>
<td>0.63</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>6</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>NSGA2_PCX</td>
<td>1.63</td>
<td>8</td>
<td>5</td>
<td>8</td>
<td>5</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>GDE3</td>
<td>0.79</td>
<td>3</td>
<td>2</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>DEMOwSA</td>
<td>1.47</td>
<td>4</td>
<td>8</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>MOSaDE</td>
<td>1.84</td>
<td>1</td>
<td>6</td>
<td>5</td>
<td>8</td>
<td>8</td>
<td>7</td>
</tr>
<tr>
<td>MO_DE</td>
<td>1.63</td>
<td>6</td>
<td>7</td>
<td>6</td>
<td>2</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>MO_PSO</td>
<td>1.84</td>
<td>5</td>
<td>4</td>
<td>7</td>
<td>7</td>
<td>6</td>
<td>6</td>
</tr>
<tr>
<td>MTS</td>
<td>1.53</td>
<td>7</td>
<td>3</td>
<td>3</td>
<td>1</td>
<td>7</td>
<td>8</td>
</tr>
</tbody>
</table>

**FES=50000**

<table>
<thead>
<tr>
<th></th>
<th>Total</th>
<th>8, S_DTLZ2</th>
<th>9, R_DTLZ2</th>
<th>10, S_DTLZ3</th>
<th>11, WFG1</th>
<th>12, WFG8</th>
<th>13, WFG9</th>
</tr>
</thead>
<tbody>
<tr>
<td>NSGA2_SBX</td>
<td>0.53</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>3</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>NSGA2_PCX</td>
<td>1.74</td>
<td>6</td>
<td>5</td>
<td>8</td>
<td>5</td>
<td>3</td>
<td>6</td>
</tr>
<tr>
<td>GDE3</td>
<td>0.74</td>
<td>4</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>DEMOwSA</td>
<td>1.37</td>
<td>2</td>
<td>8</td>
<td>3</td>
<td>7</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>MOSaDE</td>
<td>1.37</td>
<td>1</td>
<td>4</td>
<td>4</td>
<td>2</td>
<td>7</td>
<td>8</td>
</tr>
<tr>
<td>MO_DE</td>
<td>1.95</td>
<td>7</td>
<td>7</td>
<td>7</td>
<td>6</td>
<td>6</td>
<td>4</td>
</tr>
<tr>
<td>MO_PSO</td>
<td>1.84</td>
<td>5</td>
<td>6</td>
<td>6</td>
<td>8</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>MTS</td>
<td>1.84</td>
<td>8</td>
<td>3</td>
<td>5</td>
<td>4</td>
<td>8</td>
<td>7</td>
</tr>
</tbody>
</table>
CEC07 Summarized Results - Rank by \( I_{R2} \)
CEC07 Summarized Results - Rank by $I_H$

![Graph showing performance of different algorithms over FES (Function Evaluations).]
Rank ($I_{R2}$ and $I_{H}$) on all test problems

<table>
<thead>
<tr>
<th></th>
<th>FES=5000</th>
<th></th>
<th></th>
<th>FES=50000</th>
<th></th>
<th></th>
<th>FES=500000</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>R</td>
<td>H</td>
<td>Total</td>
<td>R</td>
<td>H</td>
<td>Total</td>
<td>R</td>
<td>H</td>
<td>Total</td>
</tr>
<tr>
<td>NSGA2_SBX</td>
<td>2.42(1)</td>
<td>2.32(1)</td>
<td>1</td>
<td>2.68(2)</td>
<td>2.79(2)</td>
<td>2</td>
<td>2.68(1.5)</td>
<td>3.11(2)</td>
<td>1.75</td>
</tr>
<tr>
<td>NSGA2_PCX</td>
<td>4.11(3)</td>
<td>4.37(4)</td>
<td>3.5</td>
<td>4.89(5.5)</td>
<td>4.79(4)</td>
<td>4.75</td>
<td>4.89(5)</td>
<td>5.00(5)</td>
<td>5</td>
</tr>
<tr>
<td>GDE3</td>
<td>4.26(4)</td>
<td>3.95(2)</td>
<td>3.5</td>
<td>2.47(1)</td>
<td>2.05(1)</td>
<td>1</td>
<td>4.11(3)</td>
<td>4.32(4)</td>
<td>3.5</td>
</tr>
<tr>
<td>DEMOwSA</td>
<td>5.16(5)</td>
<td>5.21(6.5)</td>
<td>5.75</td>
<td>4.37(3)</td>
<td>4.26(3)</td>
<td>3</td>
<td>4.53(4)</td>
<td>4.16(3)</td>
<td>3.5</td>
</tr>
<tr>
<td>MOSaDE</td>
<td>5.58(7)</td>
<td>5.74(8)</td>
<td>7.5</td>
<td>5.32(7)</td>
<td>5.63(7)</td>
<td>7</td>
<td>5.63(8)</td>
<td>6.32(8)</td>
<td>8</td>
</tr>
<tr>
<td>MO_DE</td>
<td>5.42(6)</td>
<td>5.21(6.5)</td>
<td>6.25</td>
<td>4.63(4)</td>
<td>5.00(5)</td>
<td>4.5</td>
<td>4.95(6)</td>
<td>5.11(6)</td>
<td>6</td>
</tr>
<tr>
<td>MO_PSO</td>
<td>5.63(8)</td>
<td>5.05(5)</td>
<td>6.5</td>
<td>6.74(8)</td>
<td>6.32(8)</td>
<td>8</td>
<td>6.53(8)</td>
<td>6.16(8)</td>
<td>8</td>
</tr>
<tr>
<td>MTS</td>
<td>3.42(2)</td>
<td>4.16(3)</td>
<td>2.5</td>
<td>4.89(5.5)</td>
<td>5.16(6)</td>
<td>5.75</td>
<td>6.53(7)</td>
<td>5.89(7)</td>
<td>7</td>
</tr>
</tbody>
</table>
Acknowledgement

- Our research into evolutionary algorithms in Singapore is financially supported by an A*Star (Agency for Science, Technology and Research), Singapore

- Thanks to all past and current researchers working with me for their contributions. These slides show the results of their research efforts.