# SUPPLEMENTARY FILE

**Algorithm 1** Nearest-better clustering [1]

Sort the whole population $P$ in increasing order based on fitness values
Calculate the distance between each pair
$T = [\,]$ (Empty tree)
**for** each $\mathbf{x}_i \in P$ **do**
    Find the nearest better neighbor $\mathbf{x}_{i,nb}$, and create an edge between $\mathbf{x}_i$ and $\mathbf{x}_{i,nb}$
    Add the edge into the tree $T$
**end for**
Calculate the mean distance $\mu_{dist}$ of all edges in $T$
**for** each edge$\in T$ **do**
    **if** $dist(edge) > \phi \cdot \mu_{dist}$ **then**
        Cut off the edge and record the seed.
    **end if**
**end for**

## 1 Harmonic average distance

The HAD method [2] assesses the crowding of solutions in decision space. For a population $P$ with $N$ solutions, the HAD value for a normalized solution $\mathbf{x}$ is computed as $\mathrm{HAD}(\mathbf{x}') = (N-1)/\sum_{\mathbf{y}'\in P,\mathbf{y}'\neq\mathbf{x}'} \frac{1}{\|\mathbf{x}'-\mathbf{y}'\|}$, where $\|\mathbf{x}'-\mathbf{y}'\|$ is the Euclidean distance between normalized solutions $\mathbf{x}'$ and $\mathbf{y}'$, with normalization defined as $\mathbf{x}'_i = \frac{\mathbf{x}_i - x_{\min,i}}{x_{\max,i} - x_{\min,i}}$ for $i = 1,\ldots,n$. The HAD value reflects crowding based on distances to other normalized solutions, where smaller HAD values indicate higher crowding in decision space.

## 2 Penalty strategies

The penalty strategy is widely used for handling nonlinear equation systems (NESs) [5, 6]. In NESs, this strategy primarily defines penalty regions around previously found roots to encourage the algorithm to explore new areas. This approach helps the algorithm discover new roots while conserving computational resources.

### 2.1 Pourjafari and Mojallali's penalty strategy [5]

The objective is to minimize the penalized function $P_f(\mathbf{x})$ as follows:

$$\text{minimize } P_f(\mathbf{x}) = (f(\mathbf{x}) + \omega) \prod_{n=1}^{N} |\coth(\alpha\delta_n)|,\ \delta_n = ||\mathbf{x} - \mathbf{x}_i^*||$$

In this formulation, $f(\mathbf{x})$ is the original objective function, $\omega$ is a small positive constant, $N$ denotes the number of roots found, and $\alpha \geq 1$ is a coefficient that controls the penalty radius. In this equation, $\delta_n$ represents the distance between $\mathbf{x}$ and the $i$-th root $\mathbf{x}_i^*$.

### 2.2 Hirsch et al.'s penalty strategy [6]

Another penalty function is proposed in [6] as follows:

$$\text{minimize } P_f(\mathbf{x}) = f(\mathbf{x}) + \zeta \sum_{n=1}^{N} e^{-\delta_n} \chi_\gamma(\delta_n)$$

**Algorithm 2** Niche center identification [3]

**Require:** Population
**Ensure:** Niche centers (`nicheCenter`), Niches (`niche`)
Calculate $K$ as $K = K_{init}$ for first half evolutionary procedure, and $K = 2 * K_{init}$ for other half;
Create an empty `scoreBoard`;
Calculate distances between all individuals;
**for** each individual $x_i$ **do**
  Find the nearest $K$ individuals to form set $S$;
  **for** each $x_k$ in $S$ **do**
    **if** $x_i$ is better than $x_k$ **then**
      `scoreBoard`$(x_k) \leftarrow 1$;
    **else**
      `scoreBoard`$(x_i) \leftarrow 1$;
    **end if**
  **end for**
**end for**
`nicheCenter` $\leftarrow \emptyset$;
**for** each individual $x_i$ **do**
  **if** `scoreBoard`$(x_i) = 0$ **then**
    Add $x_i$ to `nicheCenter`;
  **end if**
**end for**
Assign individuals not in `nicheCenter` to the nearest niche center to form `niche`;
**for** each `niche`$(i)$ **do**
  **if** size of `niche`$(i) < 5$ **then**
    Merge `niche`$(i)$ into the nearest niche;
    Remove `niche`$(i)$;
  **end if**
**end for**

where

$$\chi_\gamma(\delta_n) = \begin{cases} 1, & \text{if } \delta_n \leq \gamma \\ 0, & \text{otherwise} \end{cases}$$

Here, $\zeta$ is a large constant to adjust the penalty scale. This equation represents the indicator function, with $\gamma$ being a small constant controlling the penalty radius.

## 3 Affinity Propagation Clustering

The message-passing process in Affinity Propagation Clustering (APC) [7] determines the suitability of an individual $\mathbf{x}_k$ as an exemplar for another individual $\mathbf{x}_i$, and the appropriateness of $\mathbf{x}_i$ selecting $\mathbf{x}_k$ as its exemplar. This process uses two types of messages: *responsibility* ($r(i,k)$) and *availability* ($a(i,k)$).

(i) **Responsibility** ($r(i,k)$): Indicates how well $\mathbf{x}_k$ can serve as the exemplar for $\mathbf{x}_i$. It is calculated as:

$$r(i,k) = s(i,k) - \max_{k' \neq k} \left( a(i,k') + s(i,k') \right), \tag{1}$$

where $s(i,k) = -\|\mathbf{x}_i - \mathbf{x}_k\|^2$ represents the similarity, defined as the negative squared Euclidean distance.

(ii) **Availability** ($a(i,k)$): Reflects the appropriateness of $\mathbf{x}_i$ selecting $\mathbf{x}_k$ as its exemplar. It is computed as:

$$a(i,k) = \min \left( 0, r(k,k) + \sum_{i' \neq i,k} \max(0, r(i',k)) \right). \tag{2}$$

**Algorithm 3** INSDE($P$, $A_{os}$, $d_{min}$, $f_{min}$, $t_{max}$, $FE$) [4]

Sort all individuals of $P$ to form $P_{\text{sort}}$ from the best to the worst based on their fitness values;
**for** $i = 1$ to $N_p$ **do**
  Find the most similar $m$ individuals of the first individual from $P_{\text{sort}}$ to form $subpop_i$;
  Remove the processed members from the current $P_{\text{sort}}$;
**end for**
**for** $i = 1$ to $N_p$ **do**
  **for** $j = 1$ to $|subpop_i|$ **do**
    Produce an offspring $u_{i,j}$ using the mutation and crossover of DE within $subpop_i$;
    Evaluate $u_{i,j}$ and $FE = FE + 1$;
  **end for**
**end for**
$k = 1$;
**for** $i = 1$ to $N_p$ **do**
  **for** $j = 1$ to $|subpop_i|$ **do**
    **if** $f(u_{i,j}) > f(y_{i,j})$ **then**
      $T_k = 0$;
      $y_{i,j} \leftarrow u_{i,j}$;
    **else**
      $T_k = T_k + 1$;
    **end if**
    $k = k + 1$;
  **end for**
**end for**
Merge each $subpop_i$ into $P$;
**for** $i = 1$ to $N_p$ **do**
  Calculate the Euclidean distances of the $i$-th individual to other individuals in $P$, and store them in $D$;
  **for** $j = 1$ to $N_p$ **do**
    **if** $i \neq j$ & $D(j) < d_{\text{min}}$ & $f(x_i) > f(x_j)$ & $f(x_i) - f(x_j) < f_{\text{min}}$ **then**
      Remove the $j$-th individual from $P$;
    **end if**
  **end for**
**end for**
**for** $i = 1$ to $|P|$ **do**
  **if** $T_i \geq$ maxT **then**
    Call Archive_update($x_i$, $A_{os}$, 0.01) as shown in Algorithm 1;
    Remove the $i$-th individual from $P$;
  **end if**
**end for**

Initially, all availabilities $a(i,k)$ are set to 0. During each iteration, responsibilities and availabilities are updated iteratively. To ensure stability, a damping factor $\lambda$ is used in the updates:

$$r(i,k) = \lambda \cdot r(i,k)_{\text{last}} + (1-\lambda) \cdot r(i,k), \tag{3}$$

$$a(i,k) = \lambda \cdot a(i,k)_{\text{last}} + (1-\lambda) \cdot a(i,k). \tag{4}$$

At the end of each iteration, the exemplar for $\mathbf{x}_i$ is determined as the $\mathbf{x}_k$ that maximizes:

$$a(i,k) + r(i,k). \tag{5}$$

The message-passing process terminates when either a maximum number of iterations is reached or the exemplars remain unchanged for a certain number of iterations. This iterative procedure automatically forms clusters.

## 4 Locality-sensitive hashing

Locality-sensitive hashing (LSH) is an efficient clustering technique [8, 9] that ensures individuals close in the original data space remain close after being mapped onto a projected line via hashing. LSH uses a "*bucket*" concept, where the projected line is divided into segments of length $r$, with each segment acting as a bucket. Individuals within the same bucket are grouped into the same cluster.

The computational steps of LSH in the context of the DE algorithm are detailed in Algorithm 4. Figure 1 of the original work illustrates the process, showing six individuals distributed in a 2D Euclidean space. These individuals are mapped onto three projected lines and divided into three equal-sized buckets, forming three clusters.

**Algorithm 4** Locality-sensitive hashing (LSH) [10]

**Require:** Population of individuals, segment length $r$, projected dimensions
**Ensure:** Clusters of individuals
1: Initialize the population of individuals in a 2D space
2: Define the segment length $r$ for buckets
3: Map each individual onto $k$ projected lines via hashing
4: **for** each individual $x_i$ in the population **do**
5: Compute the hash function to project $x_i$ onto the lines
6: **for** each projected line **do**
7: Assign $x_i$ to a bucket based on the segment $r$
8: **end for**
9: **end for**
10: Group individuals within the same bucket into a cluster
11: Return the formed clusters

## 5 Parameter Adaptation in MHDE Based on JADE Mechanism

The parameter adaptation mechanism in MHDE is also a modified version of JADE [11], incorporating a novel weighted approach based on individual contributions. In MHDE, successful parameter values ($F$ and $CR$) are updated based on their contributions to fitness improvements, ensuring that parameters with higher contributions provide stronger guidance. The crossover probability $CR_c^t$ and scale factor $F_c^t$ are calculated as follows:

### 5.1 Crossover Probability ($CR$)

$$R_{i,m,t}^c = \max\left(\frac{f(\vec{\mathbf{x}}_{i,t}^c) - f(\vec{\mathbf{u}}_{i,t}^c)}{f_{\max}^t - f(\vec{\mathbf{x}}_{i,t}^c) + \varepsilon} \cdot \frac{1}{f_{\max}^t - f_{\min}^t + \varepsilon}, 0\right), \tag{6}$$

$$w\text{mean}_{m,t}^c = \frac{\sum_{i=1}^{|SCR_{m,t}^c|} \text{scr}_{i,m,t}^c \cdot R_{i,m,t}^c}{\sum_{i=1}^{|SCR_{m,t}^c|} R_{i,m,t}^c}, \tag{7}$$

$$\mu_{CR_{m,t}^c} = (1-w) \cdot \mu_{CR_{m,t-1}^c} + w \cdot w\text{mean}_{m,t}^c, \tag{8}$$

$$CR_c^t = \text{randn}(\mu_{CR_{m,t}^c}, 0.1). \tag{9}$$

5.2 Scale Factor ($F$)

$$R^c_{i,m,t} = \max\left(\frac{f(\vec{\mathbf{x}}^c_{i,t}) - f(\vec{\mathbf{u}}^c_{i,t})}{f^t_{\max} - f(\vec{\mathbf{x}}^c_{i,t}) + \varepsilon} \cdot \frac{1}{f^t_{\max} - f^t_{\min} + \varepsilon}, 0\right), \tag{10}$$

$$w\text{meanL}^c_{m,t} = \frac{\sum_{i=1}^{|SF^c_{m,t}|} (sf^c_{i,m,t})^2 \cdot R^c_{i,m,t}}{\sum_{i=1}^{|SF^c_{m,t}|} sf^c_{i,m,t} \cdot R^c_{i,m,t}}, \tag{11}$$

$$\mu_{F^c_{m,t}} = (1-w) \cdot \mu_{F^c_{m,t-1}} + w \cdot w\text{meanL}^c_{m,t}, \tag{12}$$

$$F^t_c = \text{randc}(\mu_{F^c_{m,t}}, 0.1). \tag{13}$$

5.3 Global Updates

The global parameters $\mu_{ACR_{m,t}}$ and $\mu_{AF_{m,t}}$ are updated using information from all clusters:

$$\mu_{CR^c_{m,t}} = \mu_{ACR_{m,t}}, \quad m = 1,2,\ldots,N_{C,t}, \tag{14}$$

$$\mu_{F^c_{m,t}} = \mu_{AF_{m,t}}, \quad m = 1,2,\ldots,N_{C,t}. \tag{15}$$

This mechanism ensures efficient parameter adaptation while balancing contributions across all clusters.

## 6 Fitness and distance-based local search

FDLS [12] incorporates two local search mechanisms: fitness-based and distance-based local search.

**1) Fitness-based local search:** Individuals with better fitness values are more likely near global optima, while those with worse fitness values might be trapped in local optima. To balance efficiency, superior individuals are assigned a higher probability of local search to refine solutions, while inferior ones receive lower probabilities to save fitness evaluations (FEs). The fitness-based local search probability $P_{Fi}$ for an individual $x_i$ is defined as:

$$P_{Fi} = \frac{f_i - f_{\text{worst}} + \phi}{f_{\text{best}} - f_{\text{worst}} + \phi} \tag{16}$$

Here, $f_i$ is the fitness of $x_i$, $f_{\text{best}}$ and $f_{\text{worst}}$ are the best and worst fitness values in the population, and $\phi$ is a small positive constant (set as $10^{-4}$) to handle the case where $f_{\text{best}} = f_{\text{worst}}$.

**2) Distance-based local search:** To encourage exploration, individuals distant from the best solution have a higher probability of local search, while those in similar regions have lower probabilities to avoid redundancy. The distance-based local search probability $P_{Di}$ for $x_i$ is given by:

$$P_{Di} = \frac{d_i - d_{\min} + \phi}{d_{\max} - d_{\min} + \phi} \tag{17}$$

Here, $d_i$ is the distance between $x_i$ and the best individual, $d_{\min}$ and $d_{\max}$ are the minimum and maximum distances within the population, and $\phi$ is set as $10^{-4}$.

**FDLS Logic:** The procedure of FDLS is illustrated in Algorithm 5. The combined probabilities $P_{Fi}$ and $P_{Di}$ ensure efficient and diverse local search. Local search is prioritized for individuals with both high $P_{Fi}$ (indicating promising fitness) and $P_{Di}$ (indicating different regions). This prevents redundant searches around similar areas or wasted efforts on local optima.

**Algorithm 5** FDLS [12]

Determine the standard deviation $\sigma = 10^{-1-\frac{10/D+3}{MaxFE}}$ in Gaussian distribution .
Sample two points around $x_{\text{best}}$ using $Gaussian(x, \sigma)$.
Evaluate the two points and select the better one as $x'_{\text{best}}$.
Increment FE by 2.
**if** $x'_{\text{best}}$ is better than $x_{\text{best}}$ **then**
    Replace $x_{\text{best}}$ with $x'_{\text{best}}$.
**end if**
Calculate $P_{Fi}$ for each individual using Eq. (16).
Calculate $P_{Di}$ for each individual using Eq. (17).
**for** each individual $x_i$ **do**
    **if** rand $< P_{Fi}$ **then**
        **if** rand $< P_{Di}$ **then**
            Sample two points around $x_i$ using $Gaussian(x, \sigma)$.
            Evaluate the two points and select the better one as $x'_i$.
            Increment FE by 2.
            **if** $x'_i$ is better than $x_i$ **then**
                Replace $x_i$ with $x'_i$.
            **end if**
        **end if**
    **end if**
**end for**

This approach ensures that local searches are directed towards promising and diverse areas, improving search efficiency while avoiding redundancy or wasted evaluations.

## 7 Broyden-Fletcher-Goldfarb-Shanno

The BFGS algorithm [13] is an iterative gradient-based optimization method. In black-box problems, where gradients are unavailable, numerical estimation is used. A simple hill descent is performed during the line search. The pseudocode of BFGS is summarized in Algorithm 7, where scalars are italicized, vectors are bold, and matrices are bold uppercase. Note that Algorithm 7 addresses minimization, whereas multimodal problems are typically maximization tasks.

## 8 Reinforcement learning

Reinforcement learning (RL) is a machine learning method that enables agents to learn the mapping between states and behaviors by actively interacting with an environment, aiming to maximize long-term cumulative rewards [14]. RL is distinct from supervised and unsupervised learning as it relies on learning from interactions with the environment. In RL, agents strive to maximize cumulative rewards through optimal decision-making. RL finds applications in intelligent control, robotics, analysis, and prediction.

RL comprises five key components: state space $S_s$, reward function $R_f$, state transfer function $S_t$, action $A_t$, and discount factor $D_f$. These components form a Markov Decision Process (MDP), which serves as the foundation for agent decision-making. The agent selects actions based on the current state, as illustrated in Fig. 2 of the main file, learning through trial and error to maximize rewards. The agent's performance is evaluated based on rewards obtained through interactions with the environment. Using the state change and policy table, the agent determines its next action, constructing a policy $\pi$ that combines state and action. The agent continuously updates this policy to maximize desired rewards, as defined by [15]:

$$Q^{\pi}(s,a) = E_{\pi}\left[\sum_{t=0}^{\infty} D_f^t R(s_t, a_t) | s_0 = s, a_0 = a\right]. \tag{18}$$

**Algorithm 6** Information matching mechanism [3]

**Input:** $sto$, $stagList$, $\mu F$ List, $\mu CR$ List, $nicheCenter$, $niche$
**Output:** $record$
**if** $sto$ is empty **then**
  Add all niche centers to $sto$
  Initialize the stagnation generation, $\mu F$, and $\mu CR$ for all niches to 0, $\mu F_{init}$, and $\mu CR_{init}$
**else**
  Calculate distances between niche centers and historical niche centers
  **for** each $nicheCenter(i)$ **do**
    Find the minimum distance $mindis$
    Find the corresponding historical niche centers $nearCent$ in $sto$
    **if** $mindis \leq d$ **then**
      Map $nicheCenter(i)$ to $nearCent$ in $record$
      **if** $nicheCenter(i)$ is better than $nearCent$ **then**
        $nearCent \leftarrow nicheCenter(i)$
        $stag_0 \leftarrow 0$
      **else**
        **if** $stag_0 \geq T$ **then**
          Add $nearCent$ to the archive, then remove it and its historical experience
          Remove $niche(i)$ from the current population
        **end if**
      **end if**
    **else**
      Add $nicheCenter(i)$ to $sto$
      Add a 0 to $stagList$
      Add a $\mu F_{init}$ to $\mu F$ List
      Add a $\mu CR_{init}$ to $\mu CR$ List
    **end if**
  **end for**
**end if**
All stag in $stagList$ are increased by 1

RL methods are categorized as model-based RL and model-free RL, depending on their use of environment models [14]. Model-based RL constructs complete environment models, offering precise decision-making but facing challenges in obtaining accurate MDPs for many problems. In contrast, model-free RL does not rely on learning environment models, making use of artificial neural networks (ANNs) for policy representation. Model-free RL has gained prominence due to its lower learning complexity and effective application in recent years.

## 9 Radial Basis Function (RBF)

The radial basis function (RBF) was first introduced by Hardy [16] in 1971 and has been widely used to interpolate scattered data points. RBFs are capable of approximating almost any nonlinear function [17]. In this work, RBF is employed as a surrogate model to predict the original objective function.

Suppose the function $f(\mathbf{x})$ to be approximated is defined over a given training dataset Data $= \{\mathbf{x}_i \mid i = 1,\ldots,N\}$. The fitted function $\tilde{f}(\mathbf{x})$ can be expressed as:

$$\tilde{f}(\mathbf{x}) = \sum_{i=1}^{N} w_i \phi(\|\mathbf{x} - \mathbf{x}_i\|), \tag{19}$$

where $w_i$ is the $i$th element of the weight coefficient vector $\mathbf{W} = (w_1, w_2, \ldots, w_N)^\top$, and $\phi(\cdot)$ represents the basis function. Various forms of $\phi(\cdot)$ can be used, such as the Gaussian, multiquadric, and cubic func-

**Algorithm 7** BFGS Algorithm [13]

**Require:** An initial guess $\mathbf{x}_0$
Generate an identity matrix $\mathbf{B}$ whose size matches the length of $\mathbf{x}_0$
$k \leftarrow 0$
**while** BFGS stop criterion is not met **do**
  **Identify the direction of the gradient $\mathbf{p}_k$**
  Find the direction vector $\mathbf{p}_k$ by solving the system of linear equations $\mathbf{B}_k\mathbf{p}_k = -\nabla f(\mathbf{x}_k)$
  **Calculate the step size $\mathbf{s}_k$**
  Perform a line search to find $\alpha_k$ in the direction $\mathbf{p}_k$, i.e., $\alpha_k = \arg\min f(\mathbf{x}_k + \alpha\mathbf{p}_k)$
  Compute $\mathbf{s}_k \leftarrow \alpha_k\mathbf{p}_k$
  **Create the new solution $\mathbf{x}_{k+1}$**
  $\mathbf{x}_{k+1} \leftarrow \mathbf{x}_k + \mathbf{s}_k$
  **Update B**
  $\mathbf{y}_k \leftarrow \nabla f(\mathbf{x}_{k+1}) - \nabla f(\mathbf{x}_k)$
  $\mathbf{B}_{k+1} \leftarrow \mathbf{B}_k + \frac{\mathbf{y}_k\mathbf{y}_k^\top}{\mathbf{y}_k^\top\mathbf{s}_k} - \frac{\mathbf{B}_k\mathbf{s}_k\mathbf{s}_k^\top\mathbf{B}_k^\top}{\mathbf{s}_k^\top\mathbf{B}_k\mathbf{s}_k}$
  $k \leftarrow k+1$
**end while**
**return** An improved candidate solution $\mathbf{x}_k$

tions. A cubic basis function $\phi(l) = l^3$ is explained here, as suggested by Wild and Shoemaker [18], who demonstrated its superior performance compared to other basis functions.

Given the output vector $\mathbf{F} = (f(\mathbf{x}_1), f(\mathbf{x}_2), \ldots, f(\mathbf{x}_N))^\top$ corresponding to the input vector $\mathbf{x} = (\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N)^\top$, the weights $\mathbf{W}$ must satisfy the following condition:

$$\tilde{f}(\mathbf{x}_i) = f(\mathbf{x}_i), \quad i = 1, 2, \ldots, N. \tag{20}$$

The basis function matrix $\Phi$ is defined as:

$$\Phi = \begin{bmatrix} \phi(\|\mathbf{x}_1 - \mathbf{x}_1\|) & \cdots & \phi(\|\mathbf{x}_1 - \mathbf{x}_N\|) \\ \vdots & \ddots & \vdots \\ \phi(\|\mathbf{x}_N - \mathbf{x}_1\|) & \cdots & \phi(\|\mathbf{x}_N - \mathbf{x}_N\|) \end{bmatrix}. \tag{21}$$

Using this matrix, Equation (1) can be rewritten as:

$$\Phi\mathbf{W} = \mathbf{F}. \tag{22}$$

The solution for $\mathbf{W}$ is given by:

$$\mathbf{W} = \Phi^\dagger\mathbf{F}, \tag{23}$$

where $\Phi^\dagger$ is the Moore–Penrose pseudoinverse of $\Phi$. This system is solvable when $\Phi$ is nonsingular, and conditions to guarantee nonsingularity have been provided by Micchelli [19].

## 10 Practical problems

### 10.1 Map-based problem

The map-based problem is a distance minimization task based on a practice map [20]. It involves three disconnected Pareto-optimal regions and four objectives, each related to a different type of facility: - $f_1(x)$: Distance to the nearest elementary school (red circle) - $f_2(x)$: Distance to the nearest junior high school (blue circle) - $f_3(x)$: Distance to the nearest convenience store (pink circle) - $f_4(x)$: Distance to the nearest railway station (black circle)

The decision space is $[0,100] \times [0,100]$, and the problem is defined as:

**Algorithm 8** CrowdingDE

**for** each Individual ∈ Population **do**
  Individual ← *InitializeRandomPositions* - Rosetta Stage 1()
  **for** $s \in 2:4$ **do**
    **for** each Individual ∈ Population **do**
      Rosetta Stage $s$ applied to all individuals in DE generation 1
    **end for**
    **for** $g \in 2 : MAX\_GEN$ **do**
      **for** each Individual $x' \in$ Population **do**
        $x'_1, x'_2, x'_3 \leftarrow$ GetRandomIndividual(Population)
        // $x'_1, x'_2, x'_3$ must be distinct from each other and $x'$
        R ← GetRandom(1, n)
        **for** each $i \in 1 : n$ **do**
          Compute individual's potentially new position
          $y = \{y_1, ..., y_n\}$ (trial vector)
          $r_i \leftarrow$ GetRandom(0, 1)
          **if** $((i = R)$ or $(r_i < CR))$ **then**
            $y_i = x'_1 i + F(x'_2 i - x'_3 i)$
          **else**
            $y_i = x'_i$
          **end if**
          $y' \leftarrow$ Rosetta Stage $s(y)$
          **if** $f(y') \leq f(x')$ **then**
            $x' \leftarrow y'$
          **end if**
        **end for**
      **end for**
    **end for**
  **end for**
**end for**

$$\mathrm{Min} F(x) = (f_1(x), f_2(x), f_3(x), f_4(x))^T \tag{9}$$

where $x = (x_1, x_2)$ is a 2-D decision vector.

### 10.2 Two-impulse orbit transfer problem

The optimal two-impulse orbit transfer problem [21] is a typical space trajectory design problem with two objectives: 1) Minimize the propellant consumption, and 2) Minimize the transfer time. This problem is multimodal due to the presence of multiple local optima and is solved using the Lambert problem method [22].

The first objective is to minimize the total transfer time, given by:

$$f_1 = t_w + t_f \tag{10}$$

where $t_w$ is the waiting time on the initial orbit and $t_f$ is the transfer time.

The second objective is to minimize the total transfer cost, which is the sum of the initial and final impulses:

$$f_2 = |v_I - v_{I_t}| + |v_F - v_{F_t}| \tag{11}$$

where $v_I$ and $v_F$ are the velocities on the initial and final orbits, and $v_{I_t}$ and $v_{F_t}$ are the velocities at the transfer orbit's initial and final positions.

**Algorithm 9** SharingDE

**for** each Individual ∈ Population **do**
  Individual ← *InitializeRandomPositions* - Rosetta Stage 1()
  **for** $s \in 2:4$ **do**
    **for** each Individual ∈ Population **do**
      Rosetta Stage $s$ applied to all individuals in DE generation 1
    **end for**
    **for** $g \in 2 : GEN\_MAX$ **do**
      **for** each Individual $x' \in$ Population **do**
        $x'_1, x'_2, x'_3 \leftarrow$ GetRandomIndividual(Population)
        // $x'_1, x'_2, x'_3$ must be distinct from each other and $x'$
        R ← GetRandom(1, n)
        **for** each $i \in 1 : n$ **do**
          Compute individual's potentially new position
          $y = [y_1, ..., y_n]$ (trial vector)
          $r_i \leftarrow$ GetRandom(0, 1)
          **if** $((i = R)$ or $(r_i < CR))$ **then**
            $y_i = x'_1 i + F(x'_2 i - x'_3 i)$
          **else**
            $y_i = x'_i$
          **end if**
          $y' \leftarrow$ Rosetta Stage $s(y)$
          **if** $f(y') \leq f(x')$ **then**
            $x' \leftarrow y'$
          **end if**
        **end for**
      **end for**
    **end for**
  **end for**
**end for**
Doubled Population ← Add($x'$)
**for** each Individual $z \in$ Doubled Population **do**
  $f_{shared}(z) \leftarrow$ Shared Fitness($f_{original}(z)$)
  $f_{shared}$ is calculated for each $z$ of the doubled population
**end for**
Sorted Doubled Population ← Sort(Doubled Population)
Population ← Select the best half part(Sorted Doubled Population)
The worst half part of the enlarged population is removed to define the new population in the next generation.
Elitism preserves the overall best solution found so far. If the best individual is removed during this selection step, it is replaced.

A test example is the two-impulse transfer from a low Earth orbit (LEO) to a high eccentricity Molniya-like orbit, with the following mathematical formulation:

$$\text{Min} F(t) = (f_1(t), f_2(t))^T, \quad t = (t_w, t_f)$$

where $t_w \in [0, 10.8]$ and $t_f \in [0.03, 10.8]$.

**Algorithm 10** ARA and LM Strategies

**for** each individual $X_i$ in the population **do**
  **if** $cg_i \geq mcg$ **then**
    $R_i = R_i/2$;
    $cg_i = 0$;
    $ht_i = ht_i + 1$;
  **end if**
**end for**
**Lifetime Mechanism**
**for** each individual $X_i$ in the population **do**
  **if** $ht_i \geq mht$ **then**
    $rank = 1$;
    **for** each other individual $X_j$ **do**
      **if** $f(X_i) < f(X_j)$ **then**
        $rank = rank + 1$;
      **end if**
    **end for**
    **if** $rank \leq N \times at$ **then**
      Add $X_i$ to the archive;
    **end if**
    Re-initialize $X_i$;
    $R_i = U - L$, $cg_i = 0$, $ht_i = 0$;
  **end if**
**end for**

Table S1: Experimental results of various DE variants on CEC 2013 MMOPs at $\varepsilon = 1E-03$ accuracy level.

| Func | NetCDE | ANDE | DSDE | LBPDE | PMODE | PNPCDE | NCIDE | ESPDE | HANDE | SA-DQN-DE |
|---|---|---|---|---|---|---|---|---|---|---|
| | PR ( SR ) | PR ( SR ) | PR ( SR ) | PR ( SR ) | PR ( SR ) | PR ( SR ) | PR ( SR ) | PR ( SR ) | PR ( SR ) | PR ( SR ) |
| F1 | 1 ( 1 ) | 1 ( 1 ) | 1 ( 1 ) | 1 ( 1 ) | 1 ( 1 ) | 1 ( 1 ) | 1 ( 1 ) | 1 ( 1 ) | 1 ( 1 ) | 1 ( 1 ) |
| F2 | 1 ( 1 ) | 1 ( 1 ) | 1 ( 1 ) | 1 ( 1 ) | 1 ( 1 ) | 1 ( 1 ) | 1 ( 1 ) | 1 ( 1 ) | 1 ( 1 ) | 1 ( 1 ) |
| F3 | 1 ( 1 ) | 1 ( 1 ) | 1 ( 1 ) | 1 ( 1 ) | 1 ( 1 ) | 1 ( 1 ) | 1 ( 1 ) | 1 ( 1 ) | 1 ( 1 ) | 1 ( 1 ) |
| F4 | 1 ( 1 ) | 1 ( 1 ) | 1 ( 1 ) | 1 ( 1 ) | 1 ( 1 ) | 1 ( 1 ) | 1 ( 1 ) | 1 ( 1 ) | 1 ( 1 ) | 1 ( 1 ) |
| F5 | 1 ( 1 ) | 1 ( 1 ) | 1 ( 1 ) | 1 ( 1 ) | 1 ( 1 ) | 1 ( 1 ) | 1 ( 1 ) | 1 ( 1 ) | 1 ( 1 ) | 1 ( 1 ) |
| F6 | 1 ( 1 ) | 1 ( 1 ) | 1 ( 1 ) | 1 ( 1 ) | 1 ( 1 ) | 1 ( 0.28 ) | 1 ( 1 ) | 1 ( 1 ) | 1 ( 1 ) | 1 ( 1 ) |
| F7 | 0.97 ( 0.37 ) | 0.94 ( 0.18 ) | 0.90 ( 0 ) | 0.93 ( 0 ) | 0.67 ( 0 ) | 0.89 ( 0 ) | 0.85 ( 0 ) | 0.96 ( 0.36 ) | 1 ( 1 ) | 1 ( 1 ) |
| F8 | 1 ( 1 ) | 0.95 ( 0.08 ) | 0.68 ( 0 ) | 0.83 ( 0 ) | 0.62 ( 0 ) | 0 ( 0 ) | 0.88 ( 0 ) | 0.91 ( 0 ) | 1.00 ( 0.67 ) | 1 ( 1 ) |
| F9 | 0.52 ( 0 ) | 0.52 ( 0 ) | 0.39 ( 0 ) | 0.62 ( 0 ) | 0.32 ( 0 ) | 0.47 ( 0 ) | 0.59 ( 0 ) | 0.73 ( 0 ) | 0.88 ( 0 ) | 0.93 ( 0 ) |
| F10 | 1 ( 1 ) | 1 ( 1 ) | 1 ( 1 ) | 1 ( 1 ) | 1 ( 1 ) | 1 ( 1 ) | 1 ( 1 ) | 1 ( 1 ) | 1 ( 0 ) | 1 ( 1 ) |
| F11 | 0.98 ( 0.90 ) | 1 ( 1 ) | 1 ( 1 ) | 0.78 ( 0 ) | 1 ( 1 ) | 0.67 ( 0 ) | 1 ( 1 ) | 1 ( 1 ) | 1 ( 1 ) | 1 ( 1 ) |
| F12 | 0.93 ( 0.53 ) | 1 ( 1 ) | 1 ( 1 ) | 0.82 ( 0.14 ) | 1 ( 1 ) | 0.02 ( 0 ) | 1 ( 1 ) | 0.98 ( 0.88 ) | 1 ( 1 ) | 1 ( 1 ) |
| F13 | 0.67 ( 0 ) | 0.77 ( 0.08 ) | 0.92 ( 0.55 ) | 0.76 ( 0 ) | 0.95 ( 0.72 ) | 0.64 ( 0 ) | 1 ( 1 ) | 0.81 ( 0.12 ) | 1 ( 1 ) | 1 ( 1 ) |
| F14 | 0.67 ( 0 ) | 0.67 ( 0 ) | 0.67 ( 0 ) | 0.67 ( 0 ) | 0.80 ( 0 ) | 0.59 ( 0 ) | 0.78 ( 0 ) | 0.73 ( 0 ) | 0.78 ( 0 ) | 0.88 ( 0.30 ) |
| F15 | 0.66 ( 0 ) | 0.65 ( 0 ) | 0.68 ( 0 ) | 0.75 ( 0 ) | 0.75 ( 0 ) | 0.15 ( 0 ) | 0.71 ( 0 ) | 0.74 ( 0 ) | 0.75 ( 0 ) | 0.79 ( 0 ) |
| F16 | 0.67 ( 0 ) | 0.67 ( 0 ) | 0.67 ( 0 ) | 0.67 ( 0 ) | 0.67 ( 0 ) | 0.67 ( 0 ) | 0.67 ( 0 ) | 0.67 ( 0 ) | 0.67 ( 0 ) | 0.67 ( 0 ) |
| F17 | 0.50 ( 0 ) | 0.40 ( 0 ) | 0.38 ( 0 ) | 0.63 ( 0 ) | 0.41 ( 0 ) | 0 ( 0 ) | 0.68 ( 0 ) | 0.70 ( 0 ) | 0.67 ( 0 ) | 0.69 ( 0 ) |
| F18 | 0.67 ( 0 ) | 0.65 ( 0 ) | 0.67 ( 0 ) | 0.75 ( 0 ) | 0.50 ( 0 ) | 0.16 ( 0 ) | 0.67 ( 0 ) | 0.66 ( 0 ) | 0.67 ( 0 ) | 0.67 ( 0 ) |
| F19 | 0.48 ( 0 ) | 0.36 ( 0 ) | 0.40 ( 0 ) | 0.69 ( 0 ) | 0.25 ( 0 ) | 0.00 ( 0 ) | 0.61 ( 0 ) | 0.46 ( 0 ) | 0.50 ( 0 ) | 0.54 ( 0 ) |
| F20 | 0.39 ( 0 ) | 0.25 ( 0 ) | 0.31 ( 0 ) | 0.38 ( 0 ) | 0.25 ( 0 ) | 0.00 ( 0 ) | 0.50 ( 0 ) | 0.27 ( 0 ) | 0.48 ( 0 ) | 0.45 ( 0 ) |
| $AVR_{PR}$ | 0.80 ( 0.49 ) | 0.79 ( 0.4666 ) | 0.78 ( 0.48 ) | 0.81 ( 0.36 ) | 0.76 ( 0.49 ) | 0.55 ( 0.31 ) | 0.85 ( 0.5 ) | 0.83 ( 0.47 ) | 0.87 ( 0.53 ) | 0.88 ( 0.62 ) |
| $+/\approx/-$ | 10/10/0 | 10/10/0 | 9/11/0 | 10/8/2 | 10/10/0 | 13/7/0 | 6/12/2 | 10/9/1 | 6/13/1 | |

**Algorithm 11** ELM

**if** there is/are new elite solution(s) added to the archive **then**
  Initialize its/their $\sigma$ and $sc$ to $\sigma_{ini}$ and 0, respectively;
  Cluster the solutions in the archive by mean-shift clustering;
**end if**
**for** each cluster **do**
  Its best solution refers to a solution $A_i$ in the archive;
  **if** $\sigma_i \geq \sigma_{termin}$ **then**
    Generate two elite learning solutions $ELS_{i,1}$ and $ELS_{i,2}$ and evaluate their fitness values;
    **if** $f(ELS_{i,1}) > f(A_i)$ **then**
      Replace $A_i$ with $ELS_{i,1}$;
    **end if**
    **if** $f(ELS_{i,2}) > f(A_i)$ **then**
      Replace $A_i$ with $ELS_{i,2}$;
    **else**
      $sc_i = sc_i + 2$;
      **if** $sc_i \geq dt$ **then**
        $\sigma_i = \sigma_i/10$;
        $sc_i = 0$;
      **end if**
    **end if**
  **else**
    **if** $f(A_{best}) > f(A_i)$ **then**
      $\sigma_i = \sigma_{ini}$; Restart a new elite learning process for $A_i$
      **Go to Step 8**;
    **end if**
  **end if**
**end for**

**Algorithm 12** Original DE

Randomly initialize the population with $N$ individuals;
**while** termination criterion is not met **do**
  **for** each individual $X_i$ **do**
    $V_i = X_{r1} + F \times (X_{r2} - X_{r3})$,
    where $r1, r2, r3 \in \{1, 2, \ldots, N\}, r1 \neq r2 \neq r3 \neq i$;
    $rnbri = \text{Random}(1, D)$;
    **for** each dimension $d$ **do**
      $U_i^d = \begin{cases} V_i^d, & \text{if Random}(0,1) \leq CR \text{ or } d == rnbri \\ X_i^d, & \text{otherwise} \end{cases}$
    **end for**
    **if** $f(U_i) \geq f(X_i)$ **then**
      $X_i = U_i$;
    **end if**
  **end for**
**end while**

Table S2: PR values of various DE variants on CEC 2013 MMOPs at $\varepsilon = 1E-04$ accuracy level.

| Func | NetCDE | ANDE | DSDE | LBPDE | PMODE | PNPCDE | NCIDE | OADE | ESPDE | HANDE | TNDE | ADEA | DIDE | SA-DQN-DE |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| F1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| F2 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| F3 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| F4 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| F5 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| F6 | 1 | 1 | 1 | 1 | 1 | 0.537 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| F7 | 0.947 | 0.933 | 0.891 | 0.889 | 0.672 | 0.847 | 0.852 | 0.886 | 0.963 | 1 | 1 | 0.933 | 0.921 | 1 |
| F8 | 0.999 | 0.944 | 0.655 | 0.575 | 0.616 | 0 | 0.872 | 0.505 | 0.88 | 0.994 | 1 | 0.914 | 0.692 | 0.998 |
| F9 | 0.511 | 0.512 | 0.363 | 0.476 | 0.324 | 0.474 | 0.562 | 0.339 | 0.729 | 0.843 | 0.765 | 0.572 | 0.571 | 0.927 |
| F10 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| F11 | 0.984 | 1 | 1 | 0.674 | 1 | 0.667 | 1 | 0.935 | 1 | 1 | 1 | 1 | 1 | 1 |
| F12 | 0.904 | 1 | 1 | 0.75 | 1 | 0.002 | 1 | 0.757 | 0.93 | 0.979 | 0.998 | 1 | 1 | 0.992 |
| F13 | 0.667 | 0.686 | 0.912 | 0.667 | 0.953 | 0.461 | 1 | 0.674 | 0.793 | 1 | 0.696 | 0.986 | 0.987 | 1 |
| F14 | 0.667 | 0.667 | 0.667 | 0.667 | 0.8 | 0.258 | 0.781 | 0.667 | 0.727 | 0.778 | 0.68 | 0.851 | 0.773 | 0.878 |
| F15 | 0.63 | 0.632 | 0.635 | 0.654 | 0.75 | 0.015 | 0.706 | 0.65 | 0.73 | 0.75 | 0.73 | 0.752 | 0.748 | 0.75 |
| F16 | 0.667 | 0.667 | 0.667 | 0.667 | 0.667 | 0 | 0.667 | 0.667 | 0.667 | 0.667 | 0.667 | 0.667 | 0.667 | 0.667 |
| F17 | 0.48 | 0.397 | 0.375 | 0.532 | 0.405 | 0 | 0.679 | 0.475 | 0.685 | 0.667 | 0.574 | 0.593 | 0.593 | 0.683 |
| F18 | 0.667 | ac 0.654 | 0.667 | 0.667 | 0.5 | 0.15 | 0.667 | 0.638 | 0.66 | 0.667 | 0.647 | 0.667 | 0.667 | 0.667 |
| F19 | 0.461 | 0.363 | 0.395 | 0.475 | 0.25 | 0 | 0.61 | 0.425 | 0.445 | 0.5 | 0.505 | 0.544 | 0.543 | 0.5 |
| F20 | 0.38 | 0.248 | 0.314 | 0.275 | 0.245 | 0 | 0.5 | 0.323 | 0.265 | 0.479 | 0.336 | 0.45 | 0.355 | 0.4 |
| $AVR_{PR}$ | 0.7982 | 0.78515 | 0.77705 | 0.7484 | 0.7591 | 0.47055 | 0.8448 | 0.74705 | 0.8237 | 0.8662 | 0.8299 | 0.84645 | 0.82585 | 0.8731 |
| $+/\approx/-$ | 10/9/1 | 10/9/1 | 9/10/1 | 11/9/0 | 9/10/1 | 14/6/0 | 6/11/3 | 12/8/0 | 10/9/1 | 5/14/1 | 7/10/3 | 6/10/4 | 8/10/2 | |

Table S3: SR values of various DE variants on CEC 2013 MMOPs at $\varepsilon = 1E-04$ accuracy level.

| | NetCDE | ANDE | DSDE | LBPDE | PMODE | PNPCDE | NCIDE | OADE | ESPDE | HANDE | TNDE | ADEA | DIDE | SA-DQN-DE |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| F1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| F2 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| F3 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| F4 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| F5 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| F6 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| F7 | 0.255 | 0.176 | 0 | 0 | 0 | 0 | 0 | 0.058 | 0.36 | 1 | 0.98 | 0.623 | 0.04 | 1 |
| F8 | 0.902 | 0.078 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0.667 | 0.98 | 0.471 | 0 | 0.8 |
| F9 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| F10 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| F11 | 0.902 | 1 | 1 | 0 | 1 | 0 | 1 | 0.728 | 1 | 1 | 1 | 1 | 1 | 1 |
| F12 | 0.471 | 1 | 1 | 0 | 1 | 1 | 1 | 0 | 0.56 | 0.833 | 0.98 | 1 | 1 | 0.933 |
| F13 | 0 | 0 | 0.549 | 0 | 0.72 | 0 | 1 | 0 | 0.008 | 1 | 0 | 0.88 | 0.92 | 1 |
| F14 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0.02 | 0.267 |
| F15 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| F16 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| F17 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| F18 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| F19 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| F20 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| $AVR_{SR}$ | 0.4765 | 0.4627 | 0.47745 | 0.35 | 0.486 | 0.3 | 0.5 | 0.3893 | 0.4464 | 0.575 | 0.547 | 0.5487 | 0.499 | 0.6 |

Table S4: Experimental results of various DE variants on CEC 2015 MMOPs at $\varepsilon = 1E-03$ accuracy level.

| Func | PNPCDE | LMCEDA | LMSEDA | LBPADE | FBK-DE | PMODE | NCD-DE | ANDE | NMMSO | WSNADE |
|---|---|---|---|---|---|---|---|---|---|---|
| | PR ( SR ) | PR ( SR ) | PR ( SR ) | PR ( SR ) | PR ( SR ) | PR ( SR ) | PR ( SR ) | PR ( SR ) | PR ( SR ) | PR ( SR ) |
| F1 | 1 ( 1 ) | 1 ( 1 ) | 1 ( 1 ) | 1 ( 1 ) | 1 ( 1 ) | 1 ( 1 ) | 1 ( 1 ) | 1 ( 1 ) | 1 ( 1 ) | 1 ( 1 ) |
| F2 | 1 ( 1 ) | 1 ( 1 ) | 1 ( 1 ) | 1 ( 1 ) | 1 ( 1 ) | 1 ( 1 ) | 1 ( 1 ) | 1 ( 1 ) | 1 ( 1 ) | 1 ( 1 ) |
| F3 | 1 ( 1 ) | 1 ( 1 ) | 1 ( 1 ) | 1 ( 1 ) | 1 ( 1 ) | 1 ( 1 ) | 1 ( 1 ) | 1 ( 1 ) | 1 ( 1 ) | 1 ( 1 ) |
| F4 | 0.88 ( 0.55 ) | 1 ( 1 ) | 1 ( 1 ) | 1 ( 1 ) | 1 ( 1 ) | 1 ( 1 ) | 1 ( 1 ) | 1 ( 1 ) | 1 ( 1 ) | 1 ( 1 ) |
| F5 | 1 ( 1 ) | 1 ( 1 ) | 1 ( 1 ) | 1 ( 1 ) | 1 ( 1 ) | 1 ( 1 ) | 1 ( 1 ) | 1 ( 1 ) | 1 ( 1 ) | 1 ( 1 ) |
| F6 | 1 ( 1 ) | 0.98 ( 0.61 ) | 0.93 ( 0.22 ) | 1 ( 1 ) | 0.80 ( 0.08 ) | 1.00 ( 0.98 ) | 1 ( 1 ) | 1 ( 1 ) | 0.99 ( 0.88 ) | 1 ( 1 ) |
| F7 | 0.95 ( 0.10 ) | 0.77 ( 0 ) | 0.75 ( 0 ) | 0.86 ( 0 ) | 0.67 ( 0 ) | 0.65 ( 0 ) | 0.85 ( 0 ) | 0.94 ( 0.18 ) | 1 ( 1 ) | 0.87 ( 0 ) |
| F8 | 0.52 ( 0 ) | 0.37 ( 0 ) | 0.35 ( 0 ) | 0.80 ( 0 ) | 0.32 ( 0 ) | 0.64 ( 0 ) | 0.89 ( 0 ) | 0.95 ( 0.08 ) | 0.92 ( 0.02 ) | 0.96 ( 0.12 ) |
| F9 | 0.60 ( 0 ) | 0.34 ( 0 ) | 0.29 ( 0 ) | 0.43 ( 0 ) | 0.29 ( 0 ) | 0.32 ( 0 ) | 0.41 ( 0 ) | 0.52 ( 0 ) | 0.98 ( 0.12 ) | 0.50 ( 0 ) |
| F10 | 1 ( 1 ) | 1 ( 1 ) | 1 ( 1 ) | 1 ( 1 ) | 1 ( 1 ) | 1 ( 1 ) | 1 ( 1 ) | 1 ( 1 ) | 1 ( 1 ) | 1 ( 1 ) |
| F11 | 0.56 ( 0 ) | 0.67 ( 0 ) | 0.67 ( 0 ) | 0.72 ( 0 ) | 0.99 ( 0.94 ) | 1 ( 1 ) | 1 ( 1 ) | 1 ( 1 ) | 0.99 ( 0.94 ) | 1 ( 1 ) |
| F12 | 0.03 ( 0 ) | 0.75 ( 0 ) | 0.86 ( 0 ) | 0.76 ( 0 ) | 0.91 ( 0.31 ) | 1 ( 1 ) | 0.92 ( 0.47 ) | 1 ( 1 ) | 1.00 ( 0.96 ) | 0.96 ( 0.69 ) |
| F13 | 0.34 ( 0 ) | 0.67 ( 0 ) | 0.67 ( 0 ) | 0.67 ( 0 ) | 0.99 ( 0.94 ) | 0.91 ( 0.61 ) | 0.87 ( 0.31 ) | 0.77 ( 0.08 ) | 0.98 ( 0.90 ) | 0.96 ( 0.80 ) |
| F14 | 0.33 ( 0 ) | 0.67 ( 0 ) | 0.67 ( 0 ) | 0.67 ( 0 ) | 0.78 ( 0.10 ) | 0.79 ( 0.04 ) | 0.67 ( 0 ) | 0.67 ( 0 ) | 0.72 ( 0.02 ) | 0.74 ( 0 ) |
| F15 | 0.13 ( 0 ) | 0.69 ( 0 ) | 0.62 ( 0 ) | 0.64 ( 0 ) | 0.70 ( 0 ) | 0.74 ( 0 ) | 0.64 ( 0 ) | 0.65 ( 0 ) | 0.64 ( 0 ) | 0.75 ( 0 ) |
| F16 | 0.23 ( 0 ) | 0.67 ( 0 ) | 0.67 ( 0 ) | 0.67 ( 0 ) | 0.68 ( 0 ) | 0.67 ( 0 ) | 0.67 ( 0 ) | 0.67 ( 0 ) | 0.66 ( 0 ) | 0.67 ( 0 ) |
| F17 | 0.13 ( 0 ) | 0.46 ( 0 ) | 0.52 ( 0 ) | 0.50 ( 0 ) | 0.66 ( 0 ) | 0.42 ( 0 ) | 0.50 ( 0 ) | 0.40 ( 0 ) | 0.47 ( 0 ) | 0.68 ( 0 ) |
| F18 | 0.17 ( 0 ) | 0.65 ( 0 ) | 0.66 ( 0 ) | 0.65 ( 0 ) | 0.67 ( 0 ) | 0.53 ( 0 ) | 0.66 ( 0 ) | 0.65 ( 0 ) | 0.65 ( 0 ) | 0.67 ( 0 ) |
| F19 | 0.02 ( 0 ) | 0.46 ( 0 ) | 0.49 ( 0 ) | 0.48 ( 0 ) | 0.45 ( 0 ) | 0.28 ( 0 ) | 0.50 ( 0 ) | 0.36 ( 0 ) | 0.46 ( 0 ) | 0.50 ( 0 ) |
| F20 | 0.13 ( 0 ) | 0.25 ( 0 ) | 0.25 ( 0 ) | 0.26 ( 0 ) | 0.34 ( 0 ) | 0.25 ( 0 ) | 0.26 ( 0 ) | 0.25 ( 0 ) | 0.17 ( 0 ) | 0.28 ( 0 ) |
| *AVR* | 0.55 ( 0.33235 ) | 0.72 ( 0.33 ) | 0.72 ( 0.31 ) | 0.75 ( 0.35 ) | 0.76 ( 0.42 ) | 0.76 ( 0.48 ) | 0.79 ( 0.44 ) | 0.79 ( 0.47 ) | 0.83 ( 0.54 ) | 0.83 ( 0.48 ) |
| $+/\approx/-$ | 12/6/2 | 13/7/0 | 13/7/0 | 12/8/0 | 9/7/4 | 10/7/3 | 11/9/0 | 8/9/3 | 10/6/4 | |

Table S5: Experimental results of various DE variants on CEC 2015 MMOPs at $\varepsilon = 1E-04$ accuracy level.

| Func | PNPCDE | LMCEDA | LMSEDA | LBPADE | FBK-DE | PMODE | NCD-DE | ANDE | NMMSO | WSNADE |
|---|---|---|---|---|---|---|---|---|---|---|
| | PR ( SR ) | PR ( SR ) | PR ( SR ) | PR ( SR ) | PR ( SR ) | PR ( SR ) | PR ( SR ) | PR ( SR ) | PR ( SR ) | PR ( SR ) |
| F1 | 1 ( 1 ) | 1 ( 1 ) | 1 ( 1 ) | 1 ( 1 ) | 1 ( 1 ) | 1 ( 1 ) | 1 ( 1 ) | 1 ( 1 ) | 1 ( 1 ) | 1 ( 1 ) |
| F2 | 1 ( 1 ) | 1 ( 1 ) | 1 ( 1 ) | 1 ( 1 ) | 1 ( 1 ) | 1 ( 1 ) | 1 ( 1 ) | 1 ( 1 ) | 1 ( 1 ) | 1 ( 1 ) |
| F3 | 1 ( 1 ) | 1 ( 1 ) | 1 ( 1 ) | 1 ( 1 ) | 1 ( 1 ) | 1 ( 1 ) | 1 ( 1 ) | 1 ( 1 ) | 1 ( 1 ) | 1 ( 1 ) |
| F4 | 0.72 ( 0.12 ) | 1 ( 1 ) | 1 ( 1 ) | 1 ( 1 ) | 1 ( 1 ) | 1 ( 1 ) | 1 ( 1 ) | 1 ( 1 ) | 1 ( 1 ) | 1 ( 1 ) |
| F5 | 1 ( 1 ) | 1 ( 1 ) | 1 ( 1 ) | 1 ( 1 ) | 1 ( 1 ) | 1 ( 1 ) | 1 ( 1 ) | 1 ( 1 ) | 1 ( 1 ) | 1 ( 1 ) |
| F6 | 0.98 ( 0.82 ) | 0.98 ( 0.59 ) | 0.93 ( 0.20 ) | 1 ( 1 ) | 0.80 ( 0.08 ) | 1.00 ( 0.98 ) | 1 ( 1 ) | 1 ( 1 ) | 0.99 ( 0.88 ) | 1 ( 1 ) |
| F7 | 0.93 ( 0.06 ) | 0.72 ( 0 ) | 0.70 ( 0 ) | 0.96 ( 0 ) | 0.67 ( 0 ) | 0.65 ( 0 ) | 0.83 ( 0 ) | 0.93 ( 0.18 ) | 1 ( 1 ) | 0.86 ( 0 ) |
| F8 | 0.38 ( 0 ) | 0.36 ( 0 ) | 0.35 ( 0 ) | 0.59 ( 0 ) | 0.32 ( 0 ) | 0.64 ( 0 ) | 0.86 ( 0 ) | 0.94 ( 0.08 ) | 0.90 ( 0.02 ) | 0.96 ( 0 ) |
| F9 | 0.59 ( 0 ) | 0.29 ( 0 ) | 0.24 ( 0 ) | 0.43 ( 0 ) | 0.29 ( 0 ) | 0.32 ( 0 ) | 0.41 ( 0 ) | 0.51 ( 0 ) | 0.98 ( 0.12 ) | 0.49 ( 0 ) |
| F10 | 1 ( 1 ) | 1 ( 1 ) | 1 ( 1 ) | 1 ( 1 ) | 1 ( 1 ) | 1 ( 1 ) | 1 ( 1 ) | 1 ( 1 ) | 1 ( 1 ) | 1 ( 1 ) |
| F11 | 0.51 ( 0 ) | 0.67 ( 0 ) | 0.67 ( 0 ) | 0.68 ( 0 ) | 0.99 ( 0.94 ) | 1 ( 1 ) | 1 ( 1 ) | 1 ( 1 ) | 0.99 ( 0.94 ) | 1 ( 1 ) |
| F12 | 0.13 ( 0 ) | 0.75 ( 0 ) | 0.81 ( 0.04 ) | 0.75 ( 0 ) | 0.91 ( 0.31 ) | 1 ( 1 ) | 0.91 ( 0.45 ) | 1 ( 1 ) | 0.99 ( 0.94 ) | 0.96 ( 0.57 ) |
| F13 | 0.29 ( 0 ) | 0.67 ( 0 ) | 0.67 ( 0 ) | 0.67 ( 0 ) | 0.99 ( 0.94 ) | 0.91 ( 0.61 ) | 0.86 ( 0.29 ) | 0.69 ( 0 ) | 0.98 ( 0.90 ) | 0.96 ( 0.73 ) |
| F14 | 0.30 ( 0 ) | 0.67 ( 0 ) | 0.67 ( 0 ) | 0.67 ( 0 ) | 0.78 ( 0.10 ) | 0.79 ( 0.04 ) | 0.68 ( 0 ) | 0.67 ( 0 ) | 0.72 ( 0 ) | 0.73 ( 0 ) |
| F15 | 0.13 ( 0 ) | 0.69 ( 0 ) | 0.62 ( 0 ) | 0.64 ( 0 ) | 0.70 ( 0 ) | 0.74 ( 0 ) | 0.64 ( 0 ) | 0.63 ( 0 ) | 0.63 ( 0 ) | 0.74 ( 0 ) |
| F16 | 0.19 ( 0 ) | 0.67 ( 0 ) | 0.67 ( 0 ) | 0.67 ( 0 ) | 0.68 ( 0 ) | 0.67 ( 0 ) | 0.67 ( 0 ) | 0.67 ( 0 ) | 0.33 ( 0 ) | 0.67 ( 0 ) |
| F17 | 0.12 ( 0 ) | 0.49 ( 0 ) | 0.52 ( 0 ) | 0.49 ( 0 ) | 0.66 ( 0 ) | 0.42 ( 0 ) | 0.49 ( 0 ) | 0.40 ( 0 ) | 0.47 ( 0 ) | 0.66 ( 0 ) |
| F18 | 0.17 ( 0 ) | 0.65 ( 0 ) | 0.66 ( 0 ) | 0.66 ( 0 ) | 0.67 ( 0 ) | 0.53 ( 0 ) | 0.66 ( 0 ) | 0.65 ( 0 ) | 0.65 ( 0 ) | 0.67 ( 0 ) |
| F19 | 0 ( 0 ) | 0.46 ( 0 ) | 0.49 ( 0 ) | 0.44 ( 0 ) | 0.45 ( 0 ) | 0.28 ( 0 ) | 0.49 ( 0 ) | 0.36 ( 0 ) | 0.45 ( 0 ) | 0.50 ( 0 ) |
| F20 | 0.13 ( 0 ) | 0.25 ( 0 ) | 0.25 ( 0 ) | 0.25 ( 0 ) | 0.34 ( 0 ) | 0.25 ( 0 ) | 0.26 ( 0 ) | 0.25 ( 0 ) | 0.17 ( 0 ) | 0.27 ( 0 ) |
| *AVR* | 0.53 ( 0.30 ) | 0.72 ( 0.33 ) | 0.71 ( 0.31 ) | 0.74 ( 0.35 ) | 0.76 ( 0.42 ) | 0.76 ( 0.48 ) | 0.79 ( 0.44 ) | 0.79 ( 0.46 ) | 0.81 ( 0.54 ) | 0.82 ( 0.47 ) |
| $+/\approx/-$ | 13/5/2 | 13/7/0 | 13/7/0 | 11/9/0 | 9/7/4 | 9/8/3 | 11/9/0 | 8/9/3 | 10/6/4 | |

## References


1. Mike Preuss. Niching the cma-es via nearest-better clustering. In *Proceedings of the 12th annual conference companion on Genetic and evolutionary computation*, pages 1711–1718, 2010.

2. V Ling Huang, Ponnuthurai Nagaratnam Suganthan, Alex Kai Qin, and S Baskar. Multiobjective differential evolution with external archive and harmonic distance-based diversity measure. *School of Electrical and Electronic Engineering Nanyang, Technological University Technical Report*, 2005.

3. Shao-Min Liang, Zi-Jia Wang, Yi-Biao Huang, Zhi-Hui Zhan, Sam Kwong, and Jun Zhang. Niche center identification differential evolution for multimodal optimization problems. *Information Sciences*, page 121009, 2024.

4. Kai Wang, Wenyin Gong, Libao Deng, and Ling Wang. Multimodal optimization via dynamically hybrid niching differential evolution. *Knowledge-Based Systems*, 238:107972, 2022.

5. Ebrahim Pourjafari and Hamed Mojallali. Solving nonlinear equations systems with a new approach based on invasive weed optimization algorithm and clustering. *Swarm and Evolutionary Computation*, 4:33–43, 2012.

6. Michael J Hirsch, Panos M Pardalos, and Mauricio GC Resende. Solving systems of nonlinear equations with continuous grasp. *Nonlinear Analysis: Real World Applications*, 10(4):2000–2006, 2009.

7. Zi-Jia Wang, Zhi-Hui Zhan, Ying Lin, Wei-Jie Yu, Hua Wang, Sam Kwong, and Jun Zhang. Automatic niching differential evolution with contour prediction approach for multimodal optimization problems. *IEEE Transactions on Evolutionary Computation*, 24(1):114–128, 2020.

8. Aristides Gionis, Piotr Indyk, Rajeev Motwani, et al. Similarity search in high dimensions via hashing. In *Vldb*, volume 99, pages 518–529, 1999.

9. Yu-Hui Zhang, Yue-Jiao Gong, Hua-Xiang Zhang, Tian-Long Gu, and Jun Zhang. Toward fast niching evolutionary algorithms: A locality sensitive hashing-based approach. *IEEE Transactions on Evolutionary Computation*, 21(3):347–362, 2016.

10. Xianglong Bu, Qingke Zhang, Hao Gao, and Huaxiang Zhang. Multi-strategy differential evolution algorithm based on adaptive hash clustering and its application in wireless sensor networks. *Expert Systems with Applications*, 246:123214, 2024.

11. Jingqiao Zhang and Arthur C. Sanderson. Jade: Adaptive differential evolution with optional external archive. *IEEE Transactions on Evolutionary Computation*, 13(5):945–958, 2009.

12. Zi-Jia Wang, Zhi-Hui Zhan, Yun Li, Sam Kwong, Sang-Woon Jeon, and Jun Zhang. Fitness and distance based local search with adaptive differential evolution for multimodal optimization problems. *IEEE Transactions on Emerging Topics in Computational Intelligence*, 7(3):684–699, 2023.

13. Ferrante Neri and Matthew Todd. A study on six memetic strategies for multimodal optimisation by differential evolution. In *2022 IEEE Congress on Evolutionary Computation (CEC)*, pages 1–8, 2022.

14. Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

15. Yanjie Song, Yutong Wu, Yangyang Guo, Ran Yan, Ponnuthurai Nagaratnam Suganthan, Yue Zhang, Witold Pedrycz, Yingwu Chen, Swagatam Das, Rammohan Mallipeddi, et al. Reinforcement learning-assisted evolutionary algorithm: A survey and research opportunities. *arXiv preprint arXiv:2308.13420*, 2023.

16. Rolland L Hardy. Multiquadric equations of topography and other irregular surfaces. *Journal of geophysical research*, 76(8):1905–1915, 1971.

17. Scott A Sarra and Edward J Kansa. Multiquadric radial basis function approximation methods for the numerical solution of partial differential equations. *Advances in Computational Mechanics*, 2(2):220, 2009.

18. Stefan M Wild and Christine Shoemaker. Global convergence of radial basis function trust-region algorithms for derivative-free optimization. *siam REVIEW*, 55(2):349–371, 2013.

19. Charles A Micchelli. *Interpolation of scattered data: distance matrices and conditionally positive definite functions*. Springer, 1984.

20. Hisao Ishibuchi, Naoya Akedo, and Yusuke Nojima. A many-objective test problem for visually examining diversity maintenance behavior in a decision space. In *Proceedings of the 13th annual conference on Genetic and evolutionary computation*, pages 649–656, 2011.

21. Edmondo A Minisci and Giulio Avanzini. Orbit transfer manoeuvres as a test benchmark for comparison metrics of evolutionary algorithms. In *2009 IEEE Congress on Evolutionary Computation*, pages 350–357. IEEE, 2009.

22. Robert H Gooding. A procedure for the solution of lambert's orbital boundary-value problem. *Celestial Mechanics and Dynamical Astronomy*, 48(2):145–165, 1990.