进化多目标优化平台

用户手册 4.14

生物智能与知识发现（BIMK）研究所

2026 年 1 月 30 日

非常感谢使用由安徽大学生物智能与知识发现（BIMK）研究所开发的进化多目标优化平台 PlatEMO。本平台是一个开源免费的代码库，仅供教学与科研使用，不得用于商业用途。本平台中的代码基于作者对论文的理解编写而成，作者不对用户因使用代码产生的任何后果负责。包含利用本平台产生的数据的论文应在正文中声明对 PlatEMO 的使用，并引用以下参考文献之一：

[1] Ye Tian, Weijian Zhu, Xingyi Zhang, and Yaochu Jin, “A practical tutorial on solving optimization problems via PlatEMO,” Neurocomputing, 2023, 518: 190-205.

[2] Ye Tian, Ran Cheng, Xingyi Zhang, and Yaochu Jin, “PlatEMO: A MATLAB platform for evolutionary multi-objective optimization [educational forum],” IEEE Computational Intelligence Magazine, 2017, 12(4): 73-87.

如有任何意见或建议，欢迎联系 field910921@gmail.com（田野）。如想将您的代码添加进 PlatEMO 中并公开，也欢迎联系 field910921@gmail.com。您可以在 GitHub 上获取 PlatEMO 的最新版本。

目　录

一 快速入门

软件要求：MATLAB R2018a 或以上（不使用 PlatEMO 图形界面）或
MATLAB R2020b 或以上（使用 PlatEMO 图形界面）及
并行计算工具箱 和
统计与机器学习工具箱

PlatEMO 是一个用于求解优化问题的开源平台，它的输入是一个优化问题，输出是在该优化问题上得到的最优解。一个优化问题满足以下定义：

$$
\begin{aligned}
\min_{\mathbf{x}} \quad & \mathbf{f}(\mathbf{x}) = \left(f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_M(\mathbf{x})\right) \\
\text{s.t.} \quad & \mathbf{x} = (x_1, x_2, \ldots, x_D) \in \Omega \\
& g_1(\mathbf{x}), g_2(\mathbf{x}), \ldots, g_K(\mathbf{x}) \leq 0
\end{aligned}
$$

其中 $\mathbf{x}$ 表示该问题的一个解或决策向量，它由 D 个决策变量 x_i 组成，其中每个决策变量可能被限制为实数、整数或二进制数等。Ω 表示该问题的搜索空间，它由下界 $l_1, l_2, \ldots, l_D$ 和上界 $u_1, u_2, \ldots, u_D$ 构成，即任意决策变量始终满足 $l_i \leq x_i \leq u_i$ 。$f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_M(\mathbf{x})$ 表示该解的 M 个目标函数值，$g_1(\mathbf{x}), g_2(\mathbf{x}), \ldots, g_K(\mathbf{x})$ 表示该解的 K 个约束违反值。

为了定义一个优化问题，用户至少需要输入以下内容：

- 每个决策变量的编码方式（实数、整数或二进制数等）；
- 决策变量的下界 $l_1, l_2, \ldots, l_D$ 和上界 $u_1, u_2, \ldots, u_D$；
- 至少一个目标函数 $f_1(\mathbf{x})$。

为了更精准地定义问题，用户还能输入以下内容：

- 多个目标函数 $f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_M(\mathbf{x})$；
- 多个约束函数 $g_1(\mathbf{x}), g_2(\mathbf{x}), \ldots, g_K(\mathbf{x})$；
- 解的初始化函数；
- 无效解的修复函数；
- 解的评价函数；
- 目标和约束的梯度函数；

· 各函数计算中使用到的数据（一个任意类型的常量）。

以上函数均指的是代码函数而非数学函数，即它需要有符合规定的输入和输出，但不需要有显式的数学表达式。此外，用户还能定义与优化算法相关的内容，通过选择合适的算法和参数设置以提升优化效果。

在 MATLAB 中，用户可以用以下三种方式运行主函数文件 platemo.m：

1) 带参数调用主函数：

```
platemo('problem',@SOP_F1,'algorithm',@GA);
```

可以利用指定的算法来求解指定的测试问题并设置参数，优化结果可以被显示在窗口中、保存在文件中或作为函数返回值（参阅求解测试问题章节）。

2) 带参数调用主函数：

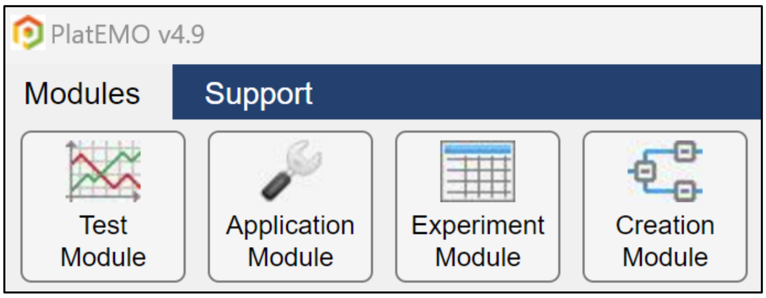
```
f1 = @(x)sum(x);
g1 = @(x)1-sum(x);
platemo('objFcn',f1,'conFcn',g1,'algorithm',@GA);
```

可以利用指定的算法来求解自定义的问题（参阅求解自定义问题章节）。

3) 不带参数调用主函数：

```
platemo();
```

可以弹出一个带有四个模块的图形界面，其中测试模块用于可视化地研究单个算法在单个问题上的性能（参阅测试模块章节），应用模块用于求解自定义问题（参阅应用模块章节），实验模块用于统计分析多个算法在多个问题上的性能（参阅实验模块章节），创造模块用于零代码创建全新的 NeuroEA 算法（参阅创造模块章节）。



二 通过命令行使用 PlatEMO

1. 求解测试问题

用户可以以如下形式带参数调用主函数 platemo()来求解测试问题：

```
platemo('Name1',Value1,'Name2',Value2,'Name3',Value3);
```

其中所有可接受的参数列举如下：

参数名	数据类型	默认值	描述
'algorithm'	函数句柄或单元数组	不定	要运行的算法类
'problem'	函数句柄或单元数组	不定	要求解的问题类
'N'	正整数	100	种群大小
'M'	正整数	不定	问题的目标数
'D'	正整数	不定	问题的变量数
'maxFE'	正整数	10000	最大评价次数
'maxRuntime'	正数	inf	最大运行时间
'save'	整数	-10	保存的种群数
'run'	正整数	[]	当前运行的编号
'metName'	字符串或单元数组	{}	要计算的指标名称
'outputFcn'	函数句柄	@DefaultOutput	每代开始前调用的函数 输入一：ALGORITHM 对象 输入二：PROBLEM 对象 输出：无

- 'algorithm'表示待运行的算法，它的值可以是一个算法类的句柄，例如@GA。它的值还可以是形如{@GA,p1,p2,…}的单元数组，其中 p1,p2,… 指定了该算法中的参数值。例如以下代码用算法@GA 求解默认问题，并设置了该算法中的参数值：

```
platemo('algorithm',{@GA,1,30,1,30});
```

- 'problem'表示待求解的测试问题，它的值可以是一个问题类的句柄，例

如@SOP_F1。它的值还可以是形如{@SOP_F1,p1,p2,…}的单元数组，其中 p1,p2,… 指定了该问题中的参数值。例如以下代码用默认算法求解问题@WFG1，并设置了该问题中的参数值：

```
platemo('problem',{@WFG1,20});
```

- 'N'表示算法使用的种群的大小，它通常等于最终输出的解的个数。例如以下代码用算法@GA 求解问题@SOP_F1，并设置种群大小为 50：

```
platemo('algorithm',@GA,'problem',@SOP_F1,'N',50);
```

- 'M'表示问题的目标个数，它仅对一些多目标测试问题生效。例如以下代码用算法@NSGAII 求解具有 5 个目标的@DTLZ2 问题：

```
platemo('algorithm',@NSGAII,'problem',@DTLZ2,'M',5);
```

- 'D'表示问题的变量个数，它仅对一些测试问题生效。例如以下代码用算法@GA 求解具有 100 个变量的@SOP_F1 问题：

```
platemo('algorithm',@GA,'problem',@SOP_F1,'D',100);
```

- 'maxFE'表示算法可用的最大评价次数，它通常等于种群大小乘以迭代次数。例如以下代码设置算法@GA 的最大评价次数为 20000：

```
platemo('algorithm',@GA,'problem',@SOP_F1,'maxFE',20000);
```

- 'maxRuntime'表示算法可用的最大运行时间，单位为秒。当'maxRuntime'等于默认值 inf 时，算法将在'maxFE'次评价次数后停止；否则，算法将在'maxRuntime'秒后停止。例如以下代码设置算法@GA 的最大运行时间为 10 秒：

```
platemo('algorithm',@GA,'problem',@SOP_F1,'maxRuntime',10);
```

- 'save'表示保存的种群数，该值大于零时优化结果将被保存在文件中，该值小于零时优化结果将被显示在窗口中（参阅获取运行结果章节）。
- 'run'表示当前运行的编号，它附加在保存文件名的末尾，使相同算法在相同问题上的多次运行结果对应的文件名不同（参阅获取运行结果章节）。
- 'metName'表示要计算的指标名称，它可以是一个字符串（单个指标）或一个单元数组（多个指标）。保存的种群会被计算指定的指标值，并保存在文件或显示在窗口中（参阅获取运行结果章节）。
- 'outputFcn'表示算法每代开始前调用的函数。该函数必须有两个输入和

零个输出，其中第一个输入是当前的 ALGORITHM 对象、第二个输入是当前的 PROBLEM 对象。默认的'outputFcn'会根据'save'的值来保存或显示优化结果。

注意以上每个参数均有一个默认值，用户可以在调用时省略任意参数。

2. 求解自定义问题

当不指定参数'problem'时，用户可以通过指定以下参数来自定义问题：

参数名	数据类型	默认值	描述
'objFcn'	函数句柄、矩阵或单元数组	{}	问题的目标函数；所有目标函数均被最小化 输入：一个决策向量 输出：目标值（标量）
'encoding'	标量或行向量	1	每个变量的编码方式
'lower'	标量或行向量	0	每个变量的下界
'upper'	标量或行向量	1	每个变量的上界
'conFcn'	函数句柄、矩阵或单元数组	{}	问题的约束函数；当且仅当约束违反值小于等于零时，该约束被满足 输入：一个决策向量 输出：约束违反值（标量）
'decFcn'	函数句柄	{}	无效解修复函数 输入：一个决策向量 输出：修复后的决策向量
'evalFcn'	函数句柄	{}	解的评价函数 输入：一个决策向量 输出一：修复后的决策向量 输出二：所有目标值（向量） 输出三：所有约束违反值（向量）
'initFcn'	函数句柄	{}	种群初始化函数 输入：种群大小 输出：种群的决策向量构成的矩阵
'gradFcn'	函数句柄	{}	目标和约束的梯度函数 输入：一个决策向量 输出一：目标雅可比矩阵 输出二：约束雅可比矩阵
'data'	任意	{}	问题的数据
'once'	逻辑	0	是否支持同时评价多个解

- 'objFcn'表示问题的目标函数，它的值可以是一个函数句柄（单目标）、矩阵（自动拟合出函数）或一个单元数组（多目标）。每个目标函数必须有一个输入和一个输出，其中输入是一个决策向量、输出是目标值。所有目标函数均被最小化。例如以下代码利用默认算法求解一个含有六个实数变量的双目标优化问题：

```
f1 = @(x)x(1)+sum(x(2:end));
f2 = @(x)sqrt(1-x(1)^2)+sum(x(2:end));
platemo('objFcn',{f1,f2},'D',6);
```

  其中第一个目标为 $x_1+\sum_{i=2}^{D}x_i$、第二个目标为 $\sqrt{1-x_1^2}+\sum_{i=2}^{D}x_i$。若一个目标函数是矩阵，则高斯过程回归会利用该矩阵自动拟合出一个函数，其中矩阵的每行表示一个样本、每列表示一个变量（除最后一列）或函数值（最后一列）。例如以下代码求解相同的问题，但目标函数是根据矩阵自动拟合出来的：

```
x = rand(50,6);
y1 = x(:,1)+sum(x(:,2:end),2);
y2 = sqrt(1-x(:,1).^2)+sum(x(:,2:end),2);
platemo('objFcn',{[x,y1],[x,y2]},'D',6);
```

- 'encoding'表示每个变量的编码方式，它的值可以是一个标量或行向量，且每维的值可以为 1（实数）、2（整数）、3（标签）、4（二进制数）或 5（序列编号）。算法针对不同的编码方式可能使用不同的算子来产生解。例如以下代码指定三个实数变量、两个整数变量以及一个二进制变量：

```
f1 = @(x)x(1)+sum(x(2:end));
f2 = @(x)sqrt(1-x(1)^2)+sum(x(2:end));
platemo('objFcn',{f1,f2},'encoding',[1,1,1,2,2,4]);
```

  问题的变量数 D 将根据'encoding'的长度自动确定。

- 'lower'和'upper'分别表示每个变量的下界和上界，它们的值可以是标量或行向量，且每维的值必须为实数。'lower'和'upper'的长度必须与'encoding'相同。例如以下代码指定搜索空间为 $[0,1]\times[0,9]^5$：

```
f1 = @(x)x(1)+sum(x(2:end));
f2 = @(x)sqrt(1-x(1)^2)+sum(x(2:end));
platemo('objFcn',{f1,f2},'encoding',[1,1,1,2,2,4],...
'lower',0,'upper',[1,9,9,9,9,9]);
```

- 'conFcn'表示问题的约束函数，它的值可以是一个函数句柄（单约束）、矩阵（自动拟合出函数）或一个单元数组（多约束）。每个约束函数必须有一个输入和一个输出，其中输入是一个决策向量、输出是约束违反值。当且仅当约束违反值小于等于零时，该约束被满足。例如以下代码利用默认算法求解一个双目标优化问题：

```
f1 = @(x)x(1)+sum(x(2:end));
f2 = @(x)sqrt(1-x(1)^2)+sum(x(2:end));
g1 = @(x)1-sum(x(2:end));
platemo('objFcn',{f1,f2},'encoding',[1,1,1,2,2,4],...
'conFcn',g1,'lower',0,'upper',[1,9,9,9,9,9]);
```

  并添加约束函数 $\sum_{i=2}^{6} x_i \geq 1$。注意，等式约束必须转换为不等式约束来处理，详细方法可参阅该论文的 3.2 节。若一个约束函数是矩阵，则高斯过程回归会利用该矩阵自动拟合出一个函数，其中矩阵的每行表示一个样本、每列表示一个变量（除最后一列）或函数值（最后一列）。例如以下代码求解相同的问题，但约束函数是根据矩阵自动拟合出来的：

```
f1 = @(x)x(1)+sum(x(2:end));
f2 = @(x)sqrt(1-x(1)^2)+sum(x(2:end));
x = rand(50,6);
y = 1-sum(x(:,2:end),2);
platemo('objFcn',{f1,f2},'encoding',[1,1,1,2,2,4],...
'conFcn',[x,y],'lower',0,'upper',[1,9,9,9,9,9]);
```

- 'decFcn'表示问题的无效解修复函数，它的值必须是一个函数句柄。该函数必须有一个输入和一个输出，其中输入是一个决策向量、输出是修复后的决策向量。默认的'decFcn'将所有解的范围限定在'lower'和'upper'之间，而以下代码定义了一个新的'decFcn'限制 x_1 为 0.1 的倍数：

```
f1 = @(x)x(1)+sum(x(2:end));
f2 = @(x)sqrt(1-x(1)^2)+sum(x(2:end));
g1 = @(x)1-sum(x(2:end));
h = @(x)[round(x(1)/0.1)*0.1,x(2:end)];
platemo('objFcn',{f1,f2},'encoding',[1,1,1,2,2,4],...
'conFcn',g1,'decFcn',h,'lower',0,'upper',[1,9,9,9,9,9]);
```

- 'evalFcn'表示解的评价函数，它的值必须是一个函数句柄。该函数必须有一个输入和三个输出，其中输入是一个决策向量、第一个输出是修复后的决策向量、第二个输出是目标值向量、第三个输出是约束违反值向量。默认

的'evalFcn'通过依次调用'decFcn'、'objFcn'和'conFcn'来评价解，而以下代码定义了一个新的'evalFcn'来同时进行解的修复、目标计算和约束计算：

```
function [x,f,g] = Eval(x)
    x = [round(x(1)/0.1)*0.1,x(2:end)];
    x = max(0,min([1,9,9,9,9,9],x));
    f(1)  = x(1)+sum(x(2:end));
    f(2)  = sqrt(1-x(1)^2)+sum(x(2:end));
    g = 1-sum(x(2:end));
end
```

接着，以下代码通过仅指定评价函数定义了相同的问题：

```
platemo('evalFcn',@Eval,'encoding',[1,1,1,2,2,4],...
'lower',0,'upper',[1,9,9,9,9,9]);
```

- 'initFcn'表示种群初始化函数，它的值必须是一个函数句柄。该函数必须有一个输入和一个输出，其中输入是种群大小、输出是种群的决策向量构成的矩阵。默认的'initFcn'在整个搜索空间内随机产生初始解，而以下代码定义了一个新的'initFcn'以加速收敛：

```
q = @(N)rand(N,6);
platemo('evalFcn',@Eval,'encoding',[1,1,1,2,2,4],...
'initFcn',q,'lower',0,'upper',[1,9,9,9,9,9]);
```

- 'gradFcn'表示目标和约束的梯度函数，它的值必须是一个函数句柄。该函数必须有一个输入和两个输出，其中输入是一个决策向量、第一个输出是目标雅可比矩阵、第二个输出是约束雅可比矩阵。默认的梯度函数通过有限差分来估计梯度，而以下代码定义了一个新的'gradFcn'以加速收敛：

```
function [oGrad,cGrad] = Grad(x)
    oGrad = [0,x(2:end);0,x(2:end)];
    cGrad = [0,x(2:end)-1/5];
end
```

接着，以下代码通过指定梯度函数来更好地求解问题：

```
platemo('evalFcn',@Eval,'encoding',[1,1,1,2,2,4],...
'gradFcn',@Grad,'lower',0,'upper',[1,9,9,9,9,9]);
```

注意仅有少量算法会使用梯度函数。

- 'data'表示问题的数据，它可以是任意类型的常量。当指定'data'后，以

上所有函数必须增加一个输入参数来接收'data'。例如以下代码求解一个旋转的单目标优化问题：

```
d = rand(RandStream('mlfg6331_64','Seed',28),10)*2-1;
[d,~] = qr(d);
f1 = @(x,d)sum((x*d-0.5).^2);
platemo('objFcn',f1,'encoding',ones(1,10),'data',d);
```

- 'once'表示是否可以同时评价多个解，它是默认值为零的逻辑变量。当指定'once'的值为 1 后，'evalFcn'、'decFcn'、'objFcn'和'conFcn'的输入可以为多个决策向量，即同时评价多个解。在函数中使用矩阵运算或并行计算来支持同时评价多个解，可以显著提升求解效率。例如以下代码将目标函数改写为矩阵运算：

```
d = rand(RandStream('mlfg6331_64','Seed',28),10)*2-1;
[d,~] = qr(d);
f1 = @(x,d)sum((x*d-0.5).^2,2);
platemo('objFcn',f1,'encoding',ones(1,10),'data',d,'once',1);
```

除以上定义问题的方式之外，用户还能创建一个自定义问题对象并创建算法对象予以求解。例如以下代码利用算法@GA 和算法@DE 求解相同的问题：

```
d = rand(RandStream('mlfg6331_64','Seed',28),10)*2-1;
[d,~] = qr(d);
f1 = @(x,d)sum((x*d-0.5).^2);
PRO = UserProblem('objFcn',f1,'encoding',ones(1,10),'data',d);
ALG1 = GA();
ALG2 = DE();
ALG1.Solve(PRO);
ALG2.Solve(PRO);
```

3. 获取运行结果

算法运行结束后得到的种群可以被显示在窗口中、保存在文件中或作为函数返回值。若按以下方式调用主函数：

```
[Dec,Obj,Con] = platemo('Name1',Value1,'Name2',Value2);
```

则最终种群会被返回，其中 Dec 表示种群的决策向量构成的矩阵、Obj 表示种群的目标值构成的矩阵、Con 表示种群的约束违反值构成的矩阵。若按以下方式调用主函数：

```
platemo('save',Value);
```

则当 Value 的值为负整数时（默认情况），得到的种群会被显示在窗口中，用户可以在窗口中的 Data source 菜单选择要显示的内容。当 Value 的值为正整数时，得到的种群会被保存在名为 PlatEMO\Data\alg\alg_pro_M_D_run.mat 的 MAT 文件中，其中 alg 表示算法名、pro 表示问题名、M 表示目标数、D 表示变量数、run 是一个自动确定的正整数以保证不和已有文件重名。同时，可按以下方式主动指定 run 的值：

```
parfor i = 1 : 100
    platemo('save',Value,'run',i);
end
```

则 run 的值会被指定为 1 到 100。在并行多次运行时，主动指定 run 的值可以避免文件编号混乱或缺失。

每个保存的数据文件存储一个单元数组 result 和一个结构体 metric，其中 result 保存得到的种群、metric 保存指标值。算法的整个优化过程被等分为 Value 块，其中 result 的第一列存储每块最后一代时所消耗的评价次数、result 的第二列存储每块最后一代时的种群、metric 存储所有种群的指标值。

```
result =
  6×2 cell array
    {[ 1600]}    {1×100 SOLUTION}
    {[ 3300]}    {1×100 SOLUTION}
    {[ 5000]}    {1×100 SOLUTION}
    {[ 6600]}    {1×100 SOLUTION}
    {[ 8300]}    {1×100 SOLUTION}
    {[10000]}    {1×100 SOLUTION}
```

```
metric =
  struct with fields:

    runtime: 0.2267
        IGD: [6×1 double]
         HV: [6×1 double]
```

可以通过参数'metName'来指定要计算的指标，例如以下代码用算法@NSGAII 求解@DTLZ2 问题，并计算 IGD 和 HV 指标值保存在文件中：

```
platemo('algorithm',@NSGAII,'problem',@DTLZ2,...
'save',6,'metName',{'IGD','HV'});
```

其中'IGD'和'HV'为要计算的指标名（参阅指标函数章节）。特别地，IGD 和 HV 是多目标优化中最常用的性能指标，它们的适用范围和参考点定义方法参阅该论文的 5.3 节。以上操作均由默认的输出函数@DefaultOutput 实现，用户可以通过指定'outputFcn'的值为其它函数来实现自定义的结果展示或保存方式。此外，可按以下方式计算单个种群的指标值：

```
% 在执行以下代码之前需先载入 result
pro = DTLZ2();
pro.CalMetric('IGD',result{end});
```

同时，图形界面的实验模块可以自动计算种群的指标值并存储到文件中。

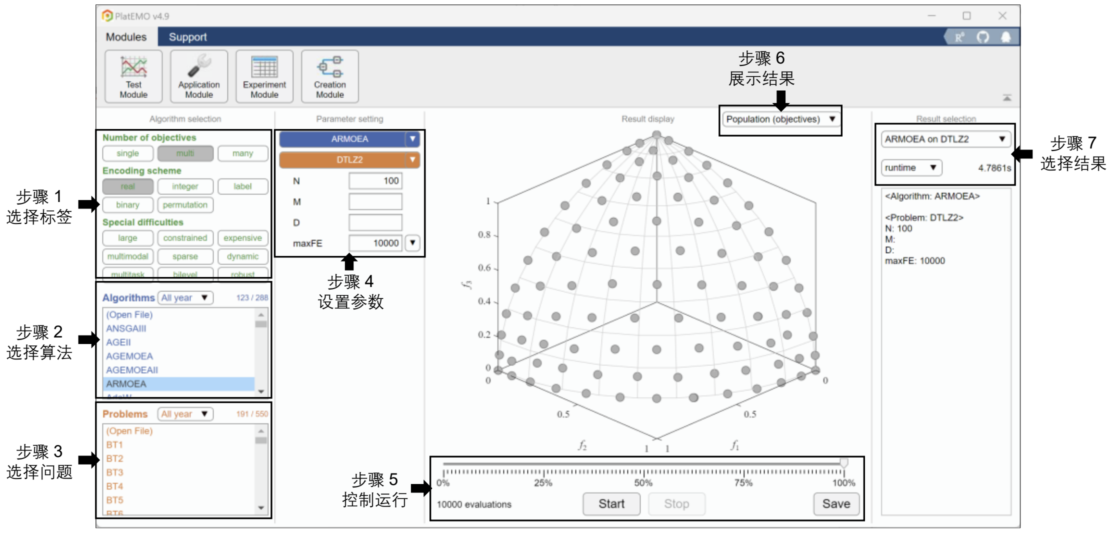
三　通过图形界面使用 PlatEMO

1. 测试模块

用户可以通过无参数调用主函数 `platemo()` 来使用 PlatEMO 的图形界面：

```
platemo();
```

图形界面的测试模块会被首先显示，它用于可视化地研究单个算法在单个问题上的性能。

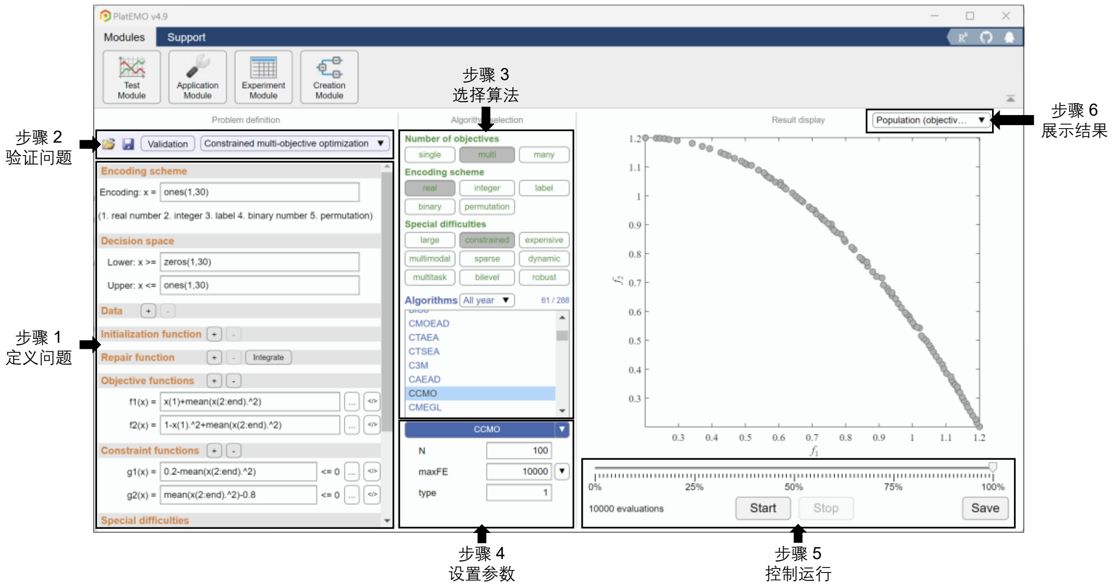


在该模块中，用户能用以下步骤研究单个算法在单个问题上的性能：

- 步骤 1：选择多个标签确定问题类型（参阅算法、问题和指标的标签章节）。
- 步骤 2：在列表中选择一个算法。
- 步骤 3：在列表中选择一个问题。
- 步骤 4：设置算法和问题的参数。不同算法和问题可能有不同的参数，在参数上悬停可查看具体说明。
- 步骤 5：开始、暂停、停止或回退算法的运行；保存当前结果到文件。当前结果可被保存为一个 N 行 $D+M+K$ 列的矩阵，N 表示解的个数，D 表示决策变量个数，M 表示目标个数，K 表示约束个数。
- 步骤 6：选择要显示的数据，例如当前种群的目标值、变量值和各指标值。
- 步骤 7：选择要显示的历史运行结果。

2. 应用模块

用户可以通过图形界面中的菜单切换至应用模块，它用于求解自定义问题。

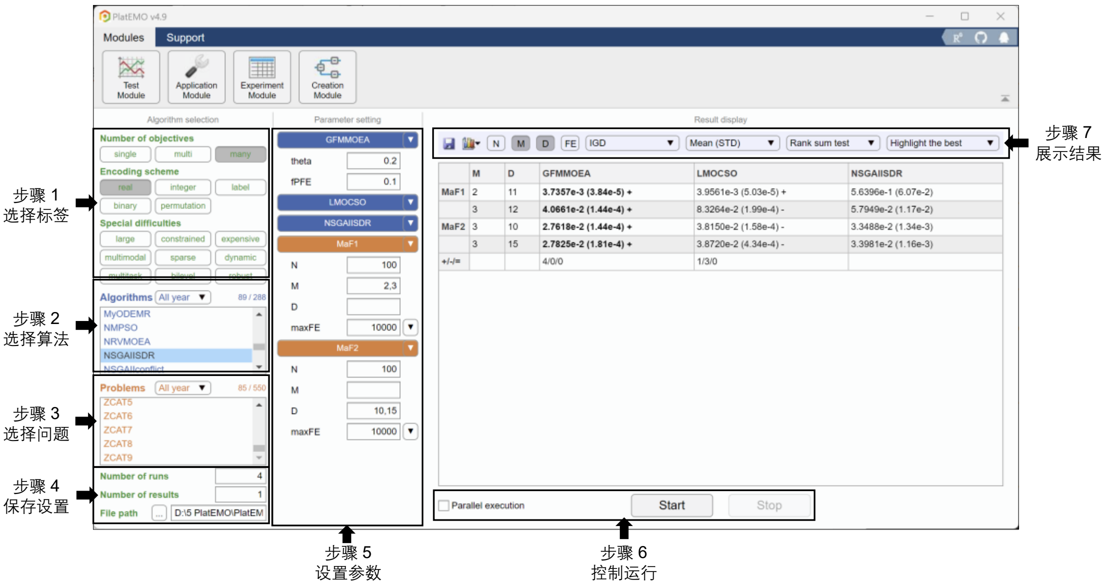


在该模块中，用户能用以下步骤求解自定义问题：

- 步骤 1：定义一个问题，定义的内容与*求解自定义问题*相同，其中 Encoding scheme 对应`'encoding'`，Decision space 对应`'lower'`和`'upper'`，Data 对应`'data'`，Initialization function 对应`'initFcn'`，Repair function 对应`'decFcn'`，Objective functions 对应`'objFcn'`，Constraint functions 对应`'conFcn'`，Evaluation function 对应`'evalFcn'`。
- 步骤 2：保存或载入问题；检测问题定义的合法性；选择一个问题模板。保存的问题可在其它模块中打开并求解。
- 步骤 3：在列表中选择一个算法。标签会根据问题定义自动确定（参阅*算法、问题和指标的标签*章节）。
- 步骤 4：设置算法的参数。不同算法可能有不同的参数，在参数上悬停可查看具体说明。
- 步骤 5：开始、暂停、停止或回退算法的运行；保存当前结果到文件。当前结果可被保存为一个 N 行 $D+M+K$ 列的矩阵，N 表示解的个数，D 表示决策变量个数，M 表示目标个数，K 表示约束个数。
- 步骤 6：选择要显示的数据，例如种群的目标值、变量值和各指标值。

3. 实验模块

用户可以通过图形界面中的菜单切换至实验模块，它用于统计分析多个算法在多个问题上的性能。该模块中所有优化结果将被保存至 MAT 文件（参见获取运行结果章节），如文件存在则会直接读取而不运行算法。

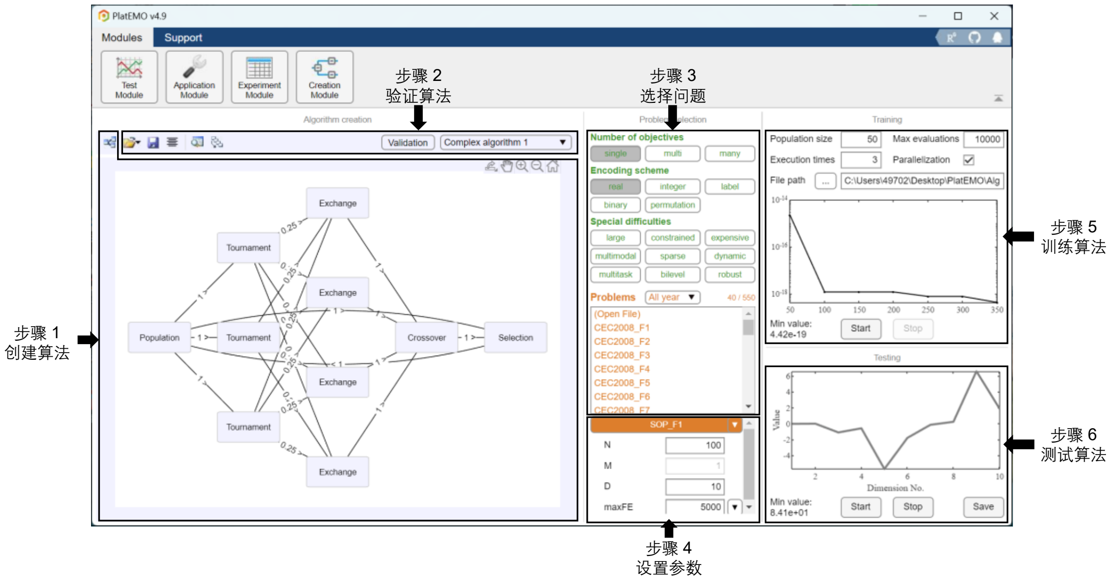


在该模块中，用户能用以下步骤比较多个算法在多个问题上的性能：

- 步骤 1：选择多个标签确定问题类型（参阅算法、问题和指标的标签章节）。
- 步骤 2：在列表中选择多个算法。
- 步骤 3：在列表中选择多个问题。
- 步骤 4：设置实验重复次数、每次保存的种群个数及保存的文件路径（参阅获取运行结果章节）。
- 步骤 5：设置算法和问题的参数。不同算法和问题可能有不同的参数，在参数上悬停可查看具体说明。此处问题的参数可以设置为向量，这使得同一个问题可以产生多个不同的测试实例。
- 步骤 6：开始或停止实验的运行；选择串行（单 CPU）或并行（多 CPU）运行实验。
- 步骤 7：选择要显示的指标值；选择要执行的统计分析；保存表格到文件；将选中的多个单元格的数据显示在图窗中。

4. 创造模块

用户可以通过图形界面中的菜单切换至创造模块，它用于创建全新的 NeuroEA 算法，并在指定问题上训练它。关于 NeuroEA 算法的细节可参阅该论文，以无界面的方式创建 NeuroEA 算法的方法可参阅创建 NeuroEA 算法章节。



在该模块中，用户能用以下步骤创建并训练算法：

- 步骤 1：通过点击按钮来添加模块，通过点击两个模块来添加连接，通过拖动模块和连接来改变布局。模块包含种群模块、算子模块和选择模块，每个模块有一些预设的超参数和一些待训练的参数；连接表示模块间解的传递方向和比例。一个算法视为一个以模块为节点、以连接为边的有权有向循环图，其中第一个节点必须为种群模块、算法至少包含一个算子模块节点、所有节点必须有前驱和后继节点、所有节点必须互相可达、所有环中必须包含至少一个种群模块节点。
- 步骤 2：保存或载入算法或模块；生成算法代码；改变显示样式；自动排列模块；检测算法的合法性；选择一个算法模板。算法训练完成后，可生成算法代码并在其它模块使用。
- 步骤 3：选择多个标签确定问题类型（参阅算法、问题和指标的标签章节）；在列表中选择一个问题。
- 步骤 4：设置问题的参数。不同问题可能有不同的参数，在参数上悬停可查看具体说明。
- 步骤 5：在选择的问题上训练算法中所有模块的参数。这个过程可能较慢，

较大的模块数目、问题变量数目、种群大小和评价次数可能耗费数天。

- 步骤 6：在选择的问题上测试训练后的算法的性能。

5. 算法、问题和指标的标签

每个算法、测试问题和指标需要被添加上标签，这些标签以注释的形式添加在主函数代码的第二行。例如在 PSO.m 代码的开头部分：

```
classdef PSO < ALGORITHM
% <1995> <single> <real/integer> <large/none> <constrained/none>
```

通过多个标签指定了该算法可求解的问题类型。所有的标签列举如下：

标签	描述
<single>	单目标优化：问题含有一个目标函数
<multi>	多目标优化：问题含有两个或三个目标函数
<many>	超多目标优化：问题含有四个或更多目标函数
<real>	连续优化：决策变量为实数
<integer>	整数优化：决策变量为整数
<label>	标签优化：决策变量为标签
<binary>	二进制优化：决策变量为二进制数
<permutation>	序列优化：决策变量构成一个排列
<large>	大规模优化：问题含有 100 或更多的决策变量
<constrained>	约束优化：问题含有至少一个约束
<expensive>	昂贵优化：目标函数的计算非常耗时，即最大评价次数非常小
<multimodal>	多模优化：存在多个目标值接近但决策向量差异很大的最优解，它们都需要被找到
<sparse>	稀疏优化：最优解中大部分的决策变量均为零
<dynamic>	动态优化：目标函数和约束函数随时间变化
<multitask>	多任务优化：同时优化多个问题，每个问题可能含有多个目标函数和约束函数
<bilevel>	双层优化：旨在寻找上层问题的可行且最优的解，一个解对于上层问题是可行的当且仅当它是下层问题的最优解
<robust>	鲁棒优化：目标函数和约束函数受噪声影响，旨在寻找受噪声影响尽可能小且尽可能优的解
<none>	空标签
<min>	（仅用于指标）该指标值越小表示性能越好
<max>	（仅用于指标）该指标值越大表示性能越好

每个算法可能含有多个标签集合，这些集合的笛卡尔积构成该算法可求解的所有的问题类型。例如当标签集合为`<single> <real> <constrained/none>`时，表示该算法可求解带或不带约束的单目标连续优化问题；若标签集合为`<single> <real>`，表示该算法只能求解无约束问题；若标签集合为`<single> <real> <constrained>`，表示该算法只能求解有约束问题；若标签集合为`<single> <real/binary>`，表示该算法可以求解连续或二进制优化问题。

每个算法、测试问题和指标都需要被添加至少一个标签，否则它将不会在图形界面的列表中出现。当用户在图形界面中选择多个标签后，仅有符合该标签组合的算法、测试问题和指标才会被显示以供选择。标签过滤的具体原理可参阅这里。PlatEMO 中所有算法和测试问题的标签分别参阅算法列表和问题列表章节。

除此之外，每个算法和测试问题可以被添加一个年份标签如`<2024>`，这使得图形界面的列表中的算法和测试问题可以按年份过滤。

四　扩展 PlatEMO

1. 算法类

每个算法需要被定义为 ALGORITHM 类的子类并保存在 PlatEMO\Algorithms 文件夹中。算法类包含的属性与方法如下：

属性	赋值方式	描述
parameter	用户	算法的参数
save	用户	每次运行中保存的种群数
run	用户	当前运行的编号
metName	用户	要计算的指标名称
outputFcn	用户	在 NotTerminated() 中调用的函数
pro	Solve()	当前运行中求解的问题对象
result	NotTerminated()	当前运行中保存的种群
metric	NotTerminated()	当前保存的种群的指标值
starttime	NotTerminated()	用于记录当前运行用时
方法	是否可重定义	描述
ALGORITHM	不可	设定由用户指定的属性值 输入：形如'Name',Value 的参数设置 输出：ALGORITHM 对象
Solve	不可	利用算法求解一个问题 输入：PROBLEM 对象 输出：无
main	必须	算法的主体部分 输入：PROBLEM 对象 输出：无
NotTerminated	不可	main() 中每次迭代前调用的函数 输入：SOLUTION 对象数组，即种群 输出：是否达到终止条件（逻辑变量）
ParameterSet	不可	根据 parameter 设定算法参数 输入：默认的参数设置 输出：用户指定的参数设置

每个算法需要继承 ALGORITHM 类并重定义方法 main()。例如 GA.m 的代码为：

```
classdef GA < ALGORITHM
% <1992><single><real/integer/label/binary/permutation><large/none><constrained/none>
% Genetic algorithm
```

```
4  % proC ---  1 --- Probability of crossover
5  % disC --- 20 --- Distribution index of crossover
6  % proM ---  1 --- Expectation of the number of mutated variables
7  % disM --- 20 --- Distribution index of mutation
8
9  %------------------------ Reference -----------------------
10 % J. H. Holland, Adaptation in Natural and Artificial
11 % Systems, MIT Press, 1992.
12 %----------------------------------------------------------
13
14     methods
15         function main(Alg,Pro)
16             [proC,disC,proM,disM] = Alg.ParameterSet(1,20,1,20);
17             P = Pro.Initialization();
18             while Alg.NotTerminated(P)
19                 Q = TournamentSelection(2,Pro.N,FitnessSingle(P));
20                 O = OperatorGA(P(Q),{proC,disC,proM,disM});
21                 P = [P,O];
22                 [~,rank] = sort(FitnessSingle(P));
23                 P = P(rank(1:Pro.N));
24             end
25         end
26     end
27 end
```

各行代码的功能如下：

第 1 行： 继承 ALGORITHM 类；

第 2 行： 为算法添加标签（参阅算法、问题和指标的标签章节）；

第 3 行： 算法的全称；

第 4-7 行： 参数名 --- 默认值 --- 参数描述，将会显示在图形界面的参数设置列表中；

第 9-12 行： 算法的参考文献；

第 15 行： 重定义算法主体流程的方法；

第 16 行： 获取用户指定的参数设置，其中 1,20,1,20 分别表示参数 proC, disC,proM,disM 的默认值。

第 17 行： 调用 PROBLEM 类的方法获得一个初始种群；

第 18 行： 保存当前种群并检查是否达到终止条件；若达到终止条件则通过抛出错误强行终止算法；

第 19 行： 调用公共函数实现基于二元联赛的交配池选择；

第 20 行： 调用公共函数产生子代种群；

第 21 行： 将父子代种群合并；

第 22 行： 调用公共函数计算种群中解的适应度，并依此对解进行排序；

第 23 行： 保留适应度较好的一半解进入下一代。

在以上代码中，函数 `ParameterSet()`和 `NotTerminated()`是 `ALGORITHM` 类的方法，函数 `Initialization()`是 `PROBLEM` 类的方法，而函数 `TournamentSelection()`、`FitnessSingle()`和 `OperatorGA()`是在 `PlatEMO\Algorithms\Utility functions` 文件夹中的公共函数。所有可被算法调用的方法及公共函数列举如下，详细的调用方式参阅代码中的注释。此外，函数中用于提升算法效率的技术参阅这里。

函数名	描述
ALGORITHM.NotTerminated	算法每代前调用的函数，用于保存当前种群及判断是否终止
ALGORITHM.ParameterSet	根据用户的输入设定算法参数
PROBLEM.Initialization	初始化一个种群
PROBLEM.Evaluation	评价一个种群并产生 SOLUTION 对象数组
CrowdingDistance	计算解的拥挤距离（仅用于多目标优化）
FitnessSingle	计算解的适应度（仅用于单目标优化）
NDSort	非支配排序（仅用于多目标优化）
OperatorDE	差分进化算子
OperatorFEP	进化规划算子
OperatorGA	遗传算子
OperatorGAhalf	遗传算子（仅返回前一半的子代）
OperatorPSO	粒子群优化算子
RouletteWheelSelection	轮盘赌选择
TournamentSelection	联赛选择
UniformPoint	产生均匀分布的参考点

2. 问题类

每个问题需要被定义为 `PROBLEM` 类的子类并保存在 `PlatEMO\ Problems` 文件夹中。问题类包含的属性与方法如下：

属性	赋值方式	描述
N	用户	求解该问题的算法的种群大小
M	用户和 Setting()	问题的目标数
D	用户和 Setting()	问题的变量数
maxFE	用户	求解该问题可使用的最大评价次数
FE	Evaluation()	当前运行中已消耗的评价次数
maxRuntime	用户	求解该问题可使用的最大运行时间（秒）
encoding	Setting()	每个变量的编码方式
lower	Setting()	每个变量的下界
upper	Setting()	每个变量的上界
optimum	GetOptimum()	问题的最优值，例如目标函数的最小值（单目标优化）和前沿面上一组均匀参考点（多目标优化）
PF	GetPF()	问题的前沿面，例如 1 维曲线（双目标优化）、2 维曲面（三目标优化）和可行区域（约束优化）
parameter	用户	问题的参数
方法	是否可重定义	描述
PROBLEM	不可	设定由用户指定的属性值 输入：形如'Name',Value 的参数设置 输出：PROBLEM 对象
Setting	必须	设定默认的属性值 输入：无 输出：无
Initialization	可以	初始化一个种群 输入：种群大小 输出：SOLUTION 对象数组，即种群
Evaluation	可以	评价一个种群并产生解对象 输入：种群的决策向量构成的矩阵 输出：SOLUTION 对象数组，即种群
CalDec	可以	修复一个种群中的无效解 输入：种群的决策向量构成的矩阵 输出：修复后的决策向量构成的矩阵
CalObj	必须	计算一个种群中解的目标值；所有目标函数均被最小化 输入：种群的决策向量构成的矩阵 输出：种群的目标值构成的矩阵
CalCon	可以	计算一个种群中解的约束违反值；当且仅当约束

		违反值小于等于零时，约束被满足 输入：种群的决策向量构成的矩阵 输出：种群的约束违反值构成的矩阵
CalGrad	可以	计算一个解在所有目标和约束上的梯度 输入：一个决策向量 输出一：目标雅可比矩阵 输出二：约束雅可比矩阵
GetOptimum	可以	产生问题的最优值并保存在 optimum 中 输入：最优值的个数 输出：最优值集合（矩阵）
GetPF	可以	产生问题的前沿面并保存在 PF 中 输入：无 输出：用于绘制前沿面的数据（矩阵或单元数组）
CalMetric	可以	计算种群的指标值 输入一：指标名 输入二：SOLUTION 对象数组，即种群 输出：指标值（标量）
DrawDec	可以	显示一个种群的决策向量 输入：SOLUTION 对象数组，即种群 输出：无
DrawObj	可以	显示一个种群的目标向量 输入：SOLUTION 对象数组，即种群 输出：无
ParameterSet	不可	根据 parameter 设定问题参数 输入：默认的参数设置 输出：用户指定的参数设置

每个算法需要继承 PROBLEM 类并重定义方法 Setting()和 CalObj()。例如 SOP_F1.m 的代码为：

```
classdef SOP_F1 < PROBLEM
% <1999><single><real><expensive/none>
% Sphere function

%------------------------- Reference ------------------------
% X. Yao, Y. Liu, and G. Lin, Evolutionary programming made
% faster, IEEE Transactions on Evolutionary Computation,
% 1999, 3(2): 82-102.
%------------------------------------------------------------

    methods
        function Setting(obj)
```

```
13                 obj.M = 1;
14                 if isempty(obj.D); obj.D = 30; end
15                 obj.lower = zeros(1,obj.D) - 100;
16                 obj.upper = zeros(1,obj.D) + 100;
17                 obj.encoding = ones(1,obj.D);
18             end
19             function PopObj = CalObj(obj,PopDec)
20                 PopObj = sum(PopDec.^2,2);
21             end
22         end
23     end
```

各行代码的功能如下：

第 1 行：　继承 PROBLEM 类；

第 2 行：　为问题添加标签（参阅算法、问题和指标的标签章节）；

第 3 行：　问题的全称；

第 5-9 行：问题的参考文献；

第 12 行：　重定义设定默认属性值的方法；

第 13 行：　设置问题的目标数；

第 14 行：　设置问题的变量数（若未被用户指定）；

第15-16行：设置决策变量的上下界；

第 17 行：　设置决策变量的编码方式；

第 19 行：　重定义计算目标函数的方法；

第 20 行：　计算种群中解的目标值。

除以上代码外，默认的方法 Initialization()用于随机初始化一个种群，用户可以重定义该方法来指定特殊的种群初始化策略。例如 Sparse_NN.m 将初始化的种群中随机一半的决策变量置零：

```
function Population = Initialization(obj,N)
    if nargin < 2; N = obj.N; end
    PopDec = (rand(N,obj.D)-0.5)*2.*randi([0 1],N,obj.D);
    Population = obj.Evaluation(PopDec);
end
```

默认的方法 CalDec()将大于上界的决策变量设为上界值、将小于下界的决策变量设为下界值，用户可以重定义该方法来指定特殊的解修复策略。例如 MOKP.m 修复了超过背包容量限制的解，使得该问题无需添加约束函数：

```
function PopDec = CalDec(obj,PopDec)
    C = sum(obj.W,2)/2;
    [~,rank] = sort(max(obj.P./obj.W));
    for i = 1 : size(PopDec,1)
        while any(obj.W*PopDec(i,:)'>C)
            k = find(PopDec(i,rank),1);
            PopDec(i,rank(k)) = 0;
        end
    end
end
```

默认的方法 `CalCon()` 返回零作为解的约束违反值（即解都是满足约束的），用户可以重定义该方法来指定问题的约束。例如 `CF4.m` 添加了一个约束：

```
function PopCon = CalCon(obj,X)
    t = X(:,2)-sin(6*pi*X(:,1)+2*pi/size(X,2))-0.5*X(:,1)+0.25;
    PopCon = -t./(1+exp(4*abs(t)));
end
```

利用 `all(PopCon<=0,2)` 可确定每个解是否满足所有约束。注意等式约束必须转换为不等式约束来处理，详细方法可参阅该论文的 3.2 节。默认的方法 `Evaluation()` 通过依次调用 `CalDec()`、`CalObj()` 和 `CalCon()` 来实例化 `SOLUTION` 对象，同时增加已消耗的评价次数 `FE` 的值。用户可以重定义该方法在一个函数内完成种群的修复、目标计算和约束计算工作，此时 `CalDec()`、`CalObj()` 和 `CalCon()` 将不会被调用。例如 `MW2.m` 同时计算了种群的目标值与约束违反值：

```
function Population = Evaluation(obj,varargin)
    X = varargin{1};
    X=max(min(X,repmat(obj.upper,size(X,1),1)),repmat(obj.lower,size(X,1),1));
    z=1-exp(-10*(X(:,obj.M:end)-(repmat(obj.M:obj.D,size(X,1),1)-1)/obj.D).^2);
    g = 1+sum((1.5+(0.1/obj.D)*z.^2-1.5*cos(2*pi*z)),2);
    PopObj(:,1) = X(:,1);
    PopObj(:,2) = g.*(1-PopObj(:,1)./g);
    L = sqrt(2)*PopObj(:,2)-sqrt(2)*PopObj(:,1);
    PopCon = sum(PopObj,2)-1-0.5*sin(3*pi*l).^8;
    Population = SOLUTION(X,PopObj,PopCon,varargin{2:end});
    obj.FE = obj.FE+length(Population);
end
```

默认的方法 `CalGrad()` 通过有限差分来估计目标函数和约束函数的梯度，用户可以重定义该方法以更准确地计算梯度。用户可以重定义方法 `GetOptimum()`

来指定问题的最优值，最优值被用于指标值的计算。例如 SOP_F8.m 指定了目标函数的最小值：

```
function R = GetOptimum(obj,N)
    R = -418.9829*obj.D;
end
```

DTLZ2.m 生成了一组前沿面上均匀分布的参考点：

```
function R = GetOptimum(obj,N)
    R = UniformPoint(N,obj.M);
    R = R./repmat(sqrt(sum(R.^2,2)),1,obj.M);
end
```

在不同形状前沿面上的采点方法参阅这里。用户可以重定义方法 GetPF()来指定多目标优化问题的前沿面或可行区域，它们被用于 DrawObj()的可视化中。例如 DTLZ2.m 生成了 2 维和 3 维的前沿面数据：

```
function R = GetPF(obj)
    if obj.M == 2
        R = obj.GetOptimum(100);
    elseif obj.M == 3
        a = linspace(0,pi/2,10)';
        R = {sin(a)*cos(a'),sin(a)*sin(a'),cos(a)*ones(size(a'))};
    else
        R = [];
    end
end
```

MW1.m 生成了可行区域的数据：

```
function R = GetPF(obj)
    [x,y]= meshgrid(linspace(0,1,400),linspace(0,1.5,400));
    z    = nan(size(x));
    fes  = x+y-1-0.5*sin(2*pi*(sqrt(2)*y-sqrt(2)*x)).^8 <= 0;
    z(fes&0.85*x+y>=1) = 0;
    R    = {x,y,z};
end
```

默认的方法 CalMetric()将一个种群与问题的最优值 optimum 传入指标函数中进行计算，用户可以重定义该方法来将不同的变量传入指标函数中。例如 SMMOP1.m 在计算 IGDX 指标时传入问题的最优解集而非前沿面上的参考点：

```
function score = CalMetric(obj,metName,Population)
```

```
    switch metName
        case 'IGDX'
            score = feval(metName,Population,obj.POS);
        otherwise
            score = feval(metName,Population,obj.optimum);
    end
end
```

默认的方法 DrawDec()显示种群的决策向量（用于图形界面中），用户可以重定义该方法来指定特殊的显示方式。例如 TSP.m 显示了种群中最优解的路径：

```
function DrawDec(obj,P)
    [~,best] = min(P.objs);
    Draw(obj.R(P(best).dec([1:end,1]),:),'-k','LineWidth',1.5);
    Draw(obj.R);
end
```

默认的方法 DrawObj()显示种群的目标向量（用于图形界面中），用户可以重定义该方法来指定特殊的显示方式。例如 Sparse_CD.m 添加了坐标轴的标签：

```
function DrawObj(obj,P)
    Draw(P.objs,{'Kernel k-means','Ratio cut',[]});
end
```

其中 Draw()用于显示数据，它位于 PlatEMO\GUI 文件夹中。

3. 个体类

一个 SOLUTION 类的对象表示一个个体（即一个解），一组 SOLUTION 类的对象表示一个种群。个体类包含的属性与方法如下：

属性	赋值方式	描述
dec	PROBLEM.Evaluation()	解的决策向量
obj	PROBLEM.Evaluation()	解的目标值
con	PROBLEM.Evaluation()	解的约束违反值
add	PROBLEM.Evaluation()	解的额外属性值（例如速度）
方法	描述	
SOLUTION	生成 SOLUTION 对象数组	

	输入一：多个解的决策向量构成的矩阵 输入二：多个解的目标值构成的矩阵 输入三：多个解的约束违反值构成的矩阵 输入四：多个解的额外属性值构成的矩阵 输出：SOLUTION 对象数组
decs	获取多个解的决策向量 输入：无 输出：多个解的决策向量构成的矩阵
objs	获取多个解的目标值 输入：无 输出：多个解的目标值构成的矩阵
cons	获取多个解的约束违反值 输入：无 输出：多个解的约束违反值构成的矩阵
adds	设置并获取多个解的额外属性值 输入：默认的额外属性值 输出：多个解的额外属性值构成的矩阵
best	获取种群中可行且最好的解（单目标优化）或可行且非支配的解（多目标优化） 输入：无 输出：种群中可行且最好的 SOLUTION 对象子数组

例如，以下代码产生一个具有十个解的种群，并获取其中最好的解的目标值矩阵：

```
Population = SOLUTION(rand(10,5),rand(10,1),zeros(10,1));
BestObjs   = Population.best.objs
```

注意应只在 PROBLEM 类的方法 Evaluation()内调用 SOLUTION()。

4. 一次完整的运行过程

以下代码利用遗传算法求球面函数的最小值：

```
Alg = GA();
Pro = SOP_F1();
Alg.Solve(Pro);
```

其中代码 Alg.Solve(Pro)执行时所涉及的函数调用过程如下图所示。

5. 指标函数

每个性能指标需要被定义为一个函数并保存在 PlatEMO\Metrics 文件夹中。例如 IGD.m 的代码为:

```
function score = IGD(Population,optimum)
% <min> <multi/many> <real/integer/label/binary/permutation> <large/none> <constrained/none> <expensive/none> <multimodal/none> <sparse/none> <dynamic/none> <robust/none>
% Inverted generational distance

%------------------------- Reference --------------------------
% C. A. Coello Coello and N. C. Cortes, Solving
% multiobjective optimization problem using an artificial
% immune system, Genetic Programming and Evolvable
```

```
9  % Machines, 2005, 6(2): 163-190.
10 %------------------------------------------------------------
11 
12     PopObj = Population.best.objs;
13     if size(PopObj,2) ~= size(optimum,2)
14         score = nan;
15     else
16         score = mean(min(pdist2(optimum,PopObj),[],2));
17     end
18 end
```

各行代码的功能如下：

第 1 行：　函数声明，其中第一个输入为一个种群（即一个 SOLUTION 对象数组）、第二个输入为问题的最优值（即问题的 optimum 属性）、输出为种群的指标值；

第 2 行：　为指标添加标签（参阅算法、问题和指标的标签章节）；注意标签 <min>或<max>必须为第一个标签；

第 3 行：　指标的全称；

第 5-10 行：指标的参考文献；

第 12 行：　获取种群中最好的解（可行且非支配的解）的目标值矩阵；

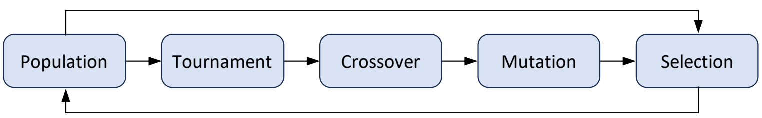
第13-14行：若种群不存在可行解则返回 nan；

第15-16行：否则返回可行且非支配的解的指标值。

6. 创建 NeuroEA 算法

NeuroEA 提供了一种创建新算法的灵活框架，它通过有权有向循环图来定义算法。图中每个节点表示一个种群处理模块如交叉、变异和选择，图中每条边决定了解在节点之间的传递方向和比例。每个节点包含许多可以自动训练的参数，一个充分训练的 NeuroEA 算法可以在训练问题上具有突出的性能。

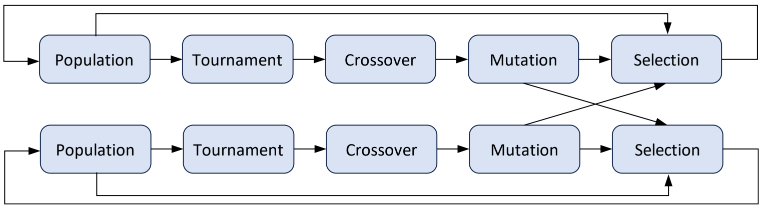
一个 NeuroEA 算法通过一个 BLOCK 对象数组和一个邻接矩阵来表示。例如一个具有如下形式的 NeuroEA 算法



可以使用如下代码创建和运行：

```
addpath('Algorithms\NeuroEA');
Blocks = [Block_Population
          Block_Tournament(200,10)
          Block_Crossover(2,5)
          Block_Mutation(5)
          Block_Selection(100)];
Graph = [0 1 0 0 1
         0 0 1 0 0
         0 0 0 1 0
         0 0 0 0 1
         1 0 0 0 0];
platemo('algorithm',{@NeuroEA,Blocks,Graph},'problem',@SOP_F1);
```

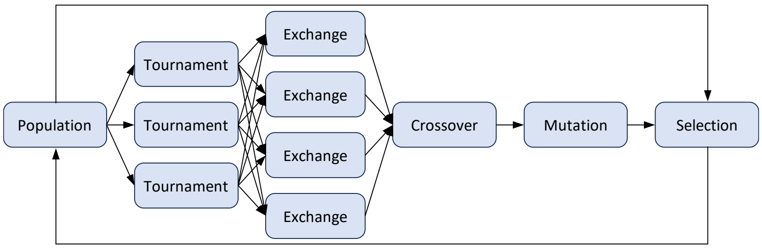
一个具有如下形式的多种群 NeuroEA 算法



可以使用如下代码创建和运行：

```
addpath('Algorithms\NeuroEA');
Blocks = [Block_Population
          Block_Tournament(100,10)
          Block_Crossover(2,5)
          Block_Mutation(5)
          Block_Selection(100)
          Block_Population
          Block_Tournament(100,10)
          Block_Crossover(2,5)
          Block_Mutation(5)
          Block_Selection(100)];
Graph = [0 1 0 0 1 0 0 0 0 0
         0 0 1 0 0 0 0 0 0 0
         0 0 0 1 0 0 0 0 0 0
         0 0 0 0 1 0 0 0 0 1
         1 0 0 0 0 0 0 0 0 0
         0 0 0 0 0 0 1 0 0 1
         0 0 0 0 0 0 0 1 0 0
         0 0 0 0 0 0 0 0 1 0
         0 0 0 0 0 0 0 0 0 1
         0 0 0 0 0 1 0 0 0 0];
platemo('algorithm',{@NeuroEA,Blocks,Graph},'problem',@SOP_F1);
```

一个具有如下形式的复杂 NeuroEA 算法



可以使用如下代码创建和运行：

```
addpath('Algorithms\NeuroEA');
Blocks = [Block_Population
          Block_Tournament(200,10)
          Block_Tournament(200,10)
          Block_Tournament(200,10)
          Block_Exchange(3)
          Block_Exchange(3)
          Block_Exchange(3)
          Block_Exchange(3)
          Block_Crossover(2,5)
          Block_Mutation(5)
          Block_Selection(100)];
Graph = [0  1  1  1  0  0  0  0  0  0  1
         0  0  0  0  1/4  1/4  1/4  1/4  0  0  0
         0  0  0  0  1/4  1/4  1/4  1/4  0  0  0
         0  0  0  0  1/4  1/4  1/4  1/4  0  0  0
         0  0  0  0  0  0  0  0  1  0  0
         0  0  0  0  0  0  0  0  1  0  0
         0  0  0  0  0  0  0  0  1  0  0
         0  0  0  0  0  0  0  0  1  0  0
         0  0  0  0  0  0  0  0  0  1  0
         0  0  0  0  0  0  0  0  0  0  1
         1  0  0  0  0  0  0  0  0  0  0];
platemo('algorithm',{@NeuroEA,Blocks,Graph},'problem',@SOP_F1);
```

NeuroEA 算法中的每个模块是模块类的一个实例。一个模块类需要被定义为 BLOCK 类的子类并保存在 PlatEMO\Algorithms\NeuroEA 文件夹中。模块类包含的属性与方法如下：

属性	赋值方式	描述
parameter	ParameterSet()	模块的参数
lower	构造函数	每个参数的下界
upper	构造函数	每个参数的上界

output	Main()	模块当前的输出种群
nextOut	Gather()	下个输出的解在种群中的编号
trainTime	用户	已训练次数
方法	是否可重定义	描述
构造函数	必须	设置由用户指定的属性值 输入：超参数的设定值 输出：BLOCK 对象
Main	必须	模块的主体部分 输入一：PROBLEM 对象 输入二：该模块的所有前驱模块 输入三：从每个前驱模块获取的解的比例 输出：无
ParameterAssign	可以	根据模块的参数确定模块的属性值 输入：无 输出：无
ParameterSet	不可	设置多个模块的参数 输入：多个模块的参数构成的向量 输出：无
parameters	不可	获取多个模块的参数 输入：无 输出：多个模块的参数构成的向量
lowers	不可	获取多个模块的参数的下界 输入：无 输出：所有参数的下界构成的向量
uppers	不可	获取多个模块的参数的上界 输入：无 输出：所有参数的上界构成的向量
Gather	不可	从所有前驱模块获取解 输入一：PROBLEM 对象 输入二：该模块的所有前驱模块 输入三：从每个前驱模块获取的解的比例 输入四：获取 SOLUTION 对象还是决策变量 输入五：值 k，获取的解的数目必须是 k 的倍数 输出：SOLUTION 对象数组或决策变量矩阵
Validity	不可	检查 NeuroEA 算法的合法性 输入：邻接矩阵 输出：无

每个模块类需要继承 BLOCK 类并重定义构造函数和方法 Main()。例如 Block_Mutation.m 的代码为

```
1  classdef Block_Mutation < BLOCK
2  % Unified mutation for real variables
3  % nSets --- 5 --- Number of parameter sets
4  
5      properties
6          nSets;
7          Weight;
8          Fit;
9          nDec = 1;
10     end
11     methods
12         function obj = Block_Mutation(nSets)
13             obj.nSets = nSets;
14             obj.lower = repmat([0 1e-20],1,nSets);
15             obj.upper = repmat([1 5],1,nSets);
16             obj.parameter = unifrnd(obj.lower,ones(1,2*nSets));
17             obj.ParameterAssign();
18         end
19         function ParameterAssign(obj)
20             obj.Weight = reshape(obj.parameter,[],obj.nSets)';
21             obj.Weight(:,end) = obj.Weight(:,end)./obj.nDec;
22             obj.Weight = [obj.Weight;0,max(0,1-sum(obj.Weight(:,end)))];
23             obj.Fit = cumsum(obj.Weight(:,end));
24             obj.Fit = obj.Fit./max(obj.Fit);
25         end
26         function Main(obj,Problem,Precursors,Ratio)
27             ParentDec = obj.Gather(Problem,Precursors,Ratio,2,1);
28             if size(ParentDec,2) ~= obj.nDec
29                 obj.nDec = size(ParentDec,2);
30                 obj.ParameterAssign();
31             end
32             r = ParaSampling(size(ParentDec),obj.Weight(:,1),obj.Fit);
33             obj.output = ParentDec + repmat(Problem.upper-...
                          Problem.lower,size(ParentDec,1),1).*r;
34         end
35     end
36 end
```

各行代码的功能如下:

第 1 行:　继承 BLOCK 类;

第 2 行:　模块的全称;

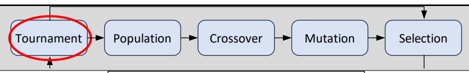
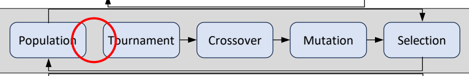
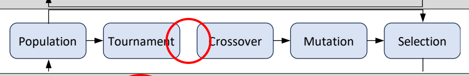
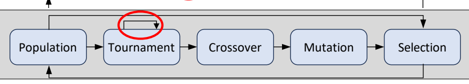
第 3 行:　超参数名 --- 默认值 --- 超参数描述，将会显示在图形界面的模块

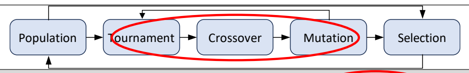
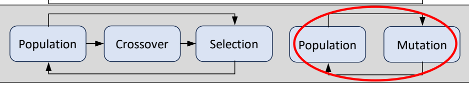
设置面板中；
第5-10行： 模块的特有属性；
第12行： 重定义构造函数；
第13行： 设置一个特有属性的值；
第14-15行：设置模块参数的上下界；
第16行： 随机产生模块参数；
第17行： 调用方法 `ParameterAssign()` 来根据模块参数确定模块的属性值；
第19-25行：重定义属性值设置的方法；该方法会在方法 `ParameterSet()` 内被自动调用；
第26行： 重定义模块主体流程的方法；
第27行： 从所有前驱模块获取解的决策变量矩阵；
第28-33行：通过变异生成新解的决策变量矩阵；该方法会在 `NeuroEA.m` 内被自动调用。

平台当前提供了七种用于创建 NeuroEA 算法的模块(即 `BLOCK` 类的子类)，包括：

模块	描述
`Block_Population`	仅用于存储解，没有任何处理过程
`Block_Crossover`	基于多个解的交叉产生一个新解
`Block_Mutation`	变异一个解
`Block_Exchange`	基于多个解的交换产生一个新解
`Block_Kopt`	翻转一个解的部分变量的顺序，主要用于序列优化
`Block_Tournament`	联赛选择，主要用于交配池选择
`Block_Selection`	保留部分解，主要用于环境选择

这些模块及其超参数的详细介绍可参阅该论文。这些模块可以被任意连接以创建 NeuroEA 算法，但需要避免以下非法情形：

非法情形	示例
第一个模块不是种群模块	
图中不包含任何算子模块	
一个模块没有前驱模块	
一个模块没有后继模块	
一个模块有自环	

一个循环中没有种群模块	
图不是强连通图	

通过调用方法 Validity()可检查 NeuroEA 算法的合法性。例如以下代码检查了一个由 Blocks 和 Graph 定义的 NeuroEA 算法的合法性，且输出由 Validity()抛出的错误 err 中存储的非法模块信息：

```
try
    Block.Validity(Graph);
catch err
    switch err.identifier
        case 'BLOCK:NoPopulation'

        case 'BLOCK:NoOperator'

        case 'BLOCK:NoInput'
            str2num(err.cause{1}.message)
        case 'BLOCK:NoOutput'
            str2num(err.cause{1}.message)
        case 'BLOCK:SelfLoop'
            str2num(err.cause{1}.message)
        case 'BLOCK:InfLoop'
            str2num(err.cause{1}.message)
        case 'BLOCK:Isolation'
            str2num(err.cause{1}.message)
    end
end
```

BLOCK 类的构造函数的输入是模块的超参数，它们决定了模块的一些特有属性的值。此外，属性 parameter 存储了模块的参数，它们也决定了一些特有属性的值且可以被训练以显著提升算法的性能。训练过程可以在创造模块中或使用以下代码实现：

```
addpath('Algorithms\NeuroEA');
Blocks = [Block_Population
            Block_Tournament(200,10)
            Block_Crossover(2,5)
            Block_Mutation(5)
            Block_Selection(100)];
Graph = [0  1  0  0  1
           0  0  1  0  0
```

```
            0  0  0  1  0
            0  0  0  0  1
            1  0  0  0  0];
function y = Fcn(x,data)
    data{1}.ParameterSet(x);
    for i = 1 : 3
        [~,obj] = platemo('algorithm',...
                          {@NeuroEA,data{1},data{2}},...
                          'problem',@SOP_F1,...
                          'outputFcn',@(~,~)[]);
        s(i) = min(obj);
    end
    y = mean(s);
end
platemo('algorithm',@GA,'objFcn',@Fcn,...
        'lower',Blocks.lowers,...
        'upper',Blocks.uppers,...
        'D',length(Blocks.lowers),...
        'data',{Blocks,Graph},'N',30,'save',1);
```

以上代码将 NeuroEA 算法的训练视为一个优化问题并利用`@GA`求解，其中决策变量是所有模块的参数，目标函数 `Fcn()`定义为该 NeuroEA 算法在`@SOP_F1`上三次优化性能的平均值。

五　算法列表

	算法缩写	算法全称	single	multi	many	real	integer	label	binary	permutation	large	constrained	expensive	multimodal	sparse	dynamic	multitask	bilevel	robust
1	ABC	Artificial bee colony algorithm	√			√	√				√	√							
2	AB-SAEA	Adaptive Bayesian based surrogate-assisted evolutionary algorithm		√	√	√	√						√						
3	AC-MMEA	Adaptive merging and coordinated offspring generation based multi-modal multi-objective evolutionary algorithm		√		√	√				√			√	√				
4	ACO	Ant colony optimization	√							√	√								
5	Adam	Adaptive moment estimation	√			√					√								
6	AdaW	Evolutionary algorithm with adaptive weights		√	√	√	√	√	√	√									
7	ADSAPSO	Adaptive dropout based surrogate-assisted particle swarm optimization		√	√	√	√						√						
8	AE-NSGA-II	Autoencoding NSGA-II		√		√	√	√	√	√		√				√			
9	AESSPSO	Adaptive exploration state-space particle swarm optimization	√			√	√				√	√							
10	AFSEA	Adjoint feature-selection-based evolutionary algorithm		√		√	√		√		√	√			√				
11	AGE-II	Approximation-guided evolutionary multi-objective algorithm II		√		√	√	√	√	√									
12	AGE-MOEA	Adaptive geometry estimation-based many-objective evolutionary algorithm		√	√	√	√	√	√	√		√							
13	AGE-MOEA-II	Adaptive geometry estimation-based many-objective evolutionary algorithm II		√	√	√	√	√	√	√		√							
14	AGSEA	Automated guiding vector selection-based evolutionary algorithm		√		√	√		√		√	√			√				
15	AMG-PSL	Adaptive multi-granular Pareto-optimal subspace learning		√		√	√		√		√	√			√				
16	A-NSGA-III	Adaptive NSGA-III		√	√	√	√	√	√	√		√							
17	APSEA	Adaptive population sizing based evolutionary algorithm		√		√	√	√	√	√		√							
18	AR-MOEA	Adaptive reference points based multi-objective evolutionary algorithm		√	√	√	√	√	√	√		√							
19	AutoV	Automated design of variation operators	√	√		√	√				√	√							
20	AVG-SAEA	Adaptive variable grouping based surrogate-assisted evolutionary algorithm		√		√	√				√		√						
21	BCE-IBEA	Bi-criterion evolution based IBEA		√	√	√	√	√	√	√									
22	BCE-MOEA/D	Bi-criterion evolution based MOEA/D		√	√	√	√	√	√	√									
23	BFGS	A quasi-Newton method proposed by Broyden, Fletcher, Goldfarb, and Shanno	√			√					√								
24	BiCo	Bidirectional coevolution constrained multiobjective evolutionary algorithm		√		√	√	√	√	√		√							

	算法缩写	算法全称	single	multi	many	real	integer	label	binary	permutation	large	constrained	expensive	multimodal	sparse	dynamic	multitask	bilevel	robust
25	BiGE	Bi-goal evolution			√	√	√	√	√	√									
26	BLEAQII	Bilevel evolutionary algorithm based on quadratic approximations II		√		√						√						√	
27	BL-SAEA	Bi-level surrogate modelling based evolutionary algorithm		√		√						√						√	
28	BSPGA	Binary space partition tree based genetic algorithm	√						√		√	√							
29	C3M	Constraint, multiobjective, multi-stage, multi-constraint evolutionary algorithm		√		√	√	√	√	√		√							
30	CAEAD	Dual-population evolutionary algorithm based on alternative evolution and degeneration		√		√	√	√	√	√		√							
31	CA-MOEA	Clustering based adaptive multi-objective evolutionary algorithm		√		√	√	√	√	√									
32	CCGDE3	Cooperative coevolution GDE3		√		√	√				√								
33	CCMO	Coevolutionary constrained multi-objective optimization framework		√		√	√	√	√	√		√							
34	c-DPEA	Constrained dual-population evolutionary algorithm		√		√	√	√	√	√		√							
35	CGLP	Correlation-guided layered prediction		√		√	√	√	√	√						√			
36	CLIA	Evolutionary algorithm with cascade clustering and reference point incremental learning		√	√	√	√	√	√	√									
37	CMaDPPs	Constrained many-objective optimization with determinantal point processes		√	√	√	√	√	√	√		√							
38	CMA-ES	Covariance matrix adaptation evolution strategy	√			√	√				√	√							
39	CMDEIPCM	Constrained multiobjective differential evolution algorithm with an infeasible proportion control mechanism		√		√	√				√	√							
40	CMEGL	Constrained evolutionary multitasking with global and local auxiliary tasks		√		√	√	√	√	√		√							
41	CMME	Constrained many-objective evolutionary algorithm with enhanced mating and environmental selections		√		√	√	√	√	√		√							
42	CMMO	Coevolutionary multi-modal multi-objective optimization framework		√		√	√	√	√	√				√					
43	CMOBR	Constrained multiobjective optimization via both constraint and objective relaxations		√	√	√	√					√							
44	CMOCSO	Competitive and cooperative swarm optimization constrained multi-objective optimization algorithm		√		√					√	√							
45	CMODE-FTR	Constrained multiobjective differential evolution based on the fusion of two rankings		√		√	√					√							
46	CMODRL	Constrained multiobjective optimization via deep reinforcement learning		√		√	√	√	√	√		√							
47	CMOEA-CD	Constraint-Pareto dominance and diversity enhancement strategy based CMOEA		√	√	√	√	√	√	√		√							
48	C-MOEA/D	Constraint-MOEA/D		√	√	√	√	√	√	√		√							
49	CMOEA-MS	Constrained multiobjective evolutionary algorithm with multiple stages		√		√	√	√	√	√		√							
50	CMOEA-MSG	Multi-stage constrained multi-objective evolutionary algorithm		√		√	√					√							
51	CMOEBOD	Constrained multiobjective evolutionary Bayesian optimization based on decomposition		√	√	√						√	√						

	算法缩写	算法全称	single	multi	many	real	integer	label	binary	permutation	large	constrained	expensive	multimodal	sparse	dynamic	multitask	bilevel	robust
52	CMOEMT	Constrained multi-objective optimization based on evolutionary multitasking optimization		√		√						√							
53	CMOES	Constrained multi-objective optimization based on even search		√		√	√	√	√	√		√							
54	CMOPSO	Competitive mechanism based multi-objective particle swarm optimizer		√		√	√												
55	CMOQLMT	Constrained multi-objective optimization based on Q-learning and multitasking		√		√						√							
56	CMOSMA	Constrained multi-objective evolutionary algorithm with self-organizing map		√	√	√	√					√							
57	CNSDE/DVC	Constrained nondominated sorting differential evolution based on decision variable classification		√		√	√												√
58	CoMMEA	Coevolutionary multimodal multi-objective evolutionary algorithm		√		√	√	√	√	√				√					
59	CPS-MOEA	Classification and Pareto domination based multi-objective evolutionary		√		√	√						√						
60	CSEA	Classification based surrogate-assisted evolutionary algorithm		√	√	√							√						
61	CSEMT	Constraints separation based evolutionary multitasking		√		√	√	√	√	√		√							
62	CSO	Competitive swarm optimizer	√			√	√				√	√							
63	C-TAEA	Two-archive evolutionary algorithm for constrained MOPs		√	√	√	√	√	√	√		√							
64	C-TSEA	Constrained two-stage evolutionary algorithm		√	√	√	√	√	√	√		√							
65	DAEA	Duplication analysis based evolutionary algorithm		√					√										
66	DBEMTO	Double-balanced evolutionary multi-task optimization		√		√	√	√	√	√		√							
67	DCNSGA-III	Dynamic constrained NSGA-III		√	√	√	√	√	√	√		√							
68	DE	Differential evolution	√			√	√				√	√							
69	DEA-GNG	Decomposition based evolutionary algorithm guided by growing neural gas		√	√	√	√	√	√	√									
70	DGEA	Direction guided evolutionary algorithm		√	√	√	√				√								
71	DirHV-EI	Expected direction-based hypervolume improvement		√	√	√	√						√						
72	DISK	Distribution-based Kriging-assisted evolutionary algorithm		√	√	√	√						√						
73	DISKplus	Distribution-based Kriging-assisted constrained evolutionary algorithm		√	√	√	√					√	√						
74	DKCA	Dynamic knowledge-guided coevolutionary algorithm		√		√			√		√	√			√				
75	DM-MOEA	Dual model based multi-objective evolutionary algorithm		√		√	√		√		√	√			√	√			
76	DMOEA-eC	Decomposition-based multi-objective evolutionary algorithm with the e-constraint framework		√		√	√	√	√	√									
77	dMOPSO	MOPSO based on decomposition		√		√	√												
78	DN-NSGA-II	Decision space based niching NSGA-II		√		√	√							√					
79	DNSGA-II	Dynamic NSGA-II		√		√	√	√	√	√						√			

	算法缩写	算法全称	single	multi	many	real	integer	label	binary	permutation	large	constrained	expensive	multimodal	sparse	dynamic	multitask	bilevel	robust
80	DOA	Dandelion optimization algorithm	√			√	√				√	√							
81	DPCPRA	Dual-population with dynamic constraint processing and resource allocating		√		√	√	√	√	√		√							
82	DP-PPS	Tri-population based push and pull search		√		√						√							
83	DPVAPS	Dual-population with variable auxiliary population size		√		√	√				√	√							
84	DRLOS-EMCMO	EMCMO with deep reinforcement learning-assisted operator selection		√		√	√	√	√	√		√							
85	DRL-SAEA	Deep reinforcement learning-based expensive constrained evolutionary algorithm		√		√						√	√						
86	DSPCMDE	Dynamic selection preference-assisted constrained multiobjective differential evolution		√		√	√					√							
87	DSSEA	Dynamic subspace search-based evolutionary algorithm		√	√	√	√				√	√							
88	DVCEA	Decision variables classification-based evolutionary algorithm		√	√	√	√				√	√							
89	DWU	Dominance-weighted uniformity multi-objective evolutionary algorithm		√		√	√	√	√	√									
90	EAG-MOEA/D	External archive guided MOEA/D		√		√	√	√	√	√									
91	ECPO	Electric charged particles optimization	√			√	√				√	√							
92	EDN-ARMOEA	Efficient dropout neural network based AR-MOEA		√	√	√	√						√						
93	EFR-RR	Ensemble fitness ranking with a ranking restriction scheme		√	√	√	√	√	√	√									
94	EGO	Efficient global optimization	√			√	√						√						
95	EIM-EGO	Expected improvement matrix based efficient global optimization		√		√	√						√						
96	EMCMMS	Evolutionary multitasking with a cooperative multistep mutation strategy		√		√	√	√	√	√		√							
97	EMCMO	Evolutionary multitasking-based constrained multiobjective optimization		√		√	√	√	√	√		√							
98	EMMOEA	Expensive multi-/many-objective evolutionary algorithm		√		√	√						√						
99	e-MOEA	Epsilon multi-objective evolutionary algorithm		√	√	√	√	√	√	√									
100	EMOSKT	Evolutionary multi-objective optimization with sparsity knowledge transfer		√		√			√		√	√			√		√		
101	EM-SAEA	Ensemble-based surrogate model-assisted evolutionary algorithm		√	√	√						√	√						
102	EMyO/C	Evolutionary many-objective optimization algorithm with clustering-based		√	√	√	√												
103	ENS-MOEA/D	Ensemble of different neighborhood sizes based MOEA/D		√	√	√	√												
104	ESBCEO	Bayesian co-evolutionary optimization based entropy search		√		√							√						
105	FDV	Fuzzy decision variable framework with various internal optimizers		√	√	√	√				√								
106	FEP	Fast evolutionary programming	√			√	√				√	√							

	算法缩写	算法全称	single	multi	many	real	integer	label	binary	permutation	large	constrained	expensive	multimodal	sparse	dynamic	multitask	bilevel	robust
107	FLEA	Fast sampling based evolutionary algorithm		√	√	√					√								
108	FRCG	Fletcher-Reeves conjugate gradient	√			√					√								
109	FRCGM	Fletcher-Reeves conjugate gradient (for multi-objective optimization)		√	√	√					√	√							
110	FROFI	Feasibility rule with the incorporation of objective function information	√			√	√				√	√							
111	GA	Genetic algorithm	√			√	√	√	√	√	√	√							
112	GCNMOEA	Graph convolutional network based multi-objective evolutionary algorithm		√		√	√												
113	GDE3	Generalized differential evolution 3		√		√	√					√							
114	GFM-MOEA	Generic front modeling based multi-objective evolutionary algorithm		√	√	√	√	√	√	√									
115	GLMO	Grouped and linked mutation operator algorithm		√		√	√				√								
116	g-NSGA-II	g-dominance based NSGA-II		√		√	√	√	√	√									
117	GPSO	Gradient based particle swarm optimization algorithm	√			√					√	√							
118	GPSOM	Gradient based particle swarm optimization algorithm (for multi-objective optimization)		√	√	√					√	√							
119	GrEA	Grid-based evolutionary algorithm			√	√	√	√	√	√									
120	GWASF-GA	Global weighting achievement scalarizing function genetic algorithm		√		√	√	√	√	√									
121	GWO	Grey wolf optimizer	√			√	√				√	√							
122	HEA	Hyper-dominance based evolutionary algorithm		√	√	√			√	√									
123	HeE-MOEA	Multiobjective evolutionary algorithm with heterogeneous ensemble based infill criterion		√		√	√						√						
124	HHC-MMEA	Hybrid hierarchical clustering based multi-modal multi-objective evolutionary algorithm		√		√					√			√	√				
125	hpaEA	Hyperplane assisted evolutionary algorithm		√	√	√	√	√	√	√									
126	HREA	Hierarchy ranking based evolutionary algorithm		√		√	√							√					
127	HypE	Hypervolume estimation algorithm		√	√	√	√	√	√	√									
128	IBEA	Indicator-based evolutionary algorithm		√	√	√	√	√	√	√									
129	ICMA	Indicator based constrained multi-objective algorithm		√		√	√					√							
130	I-DBEA	Improved decomposition-based evolutionary algorithm		√	√	√	√	√	√	√		√							
131	IM-C-MOEA/D	Inverse modeling constrained MOEA/D		√		√	√				√	√							
132	IM-MOEA	Inverse modeling based multiobjective evolutionary algorithm		√		√	√				√								
133	IM-MOEA/D	Inverse modeling MOEA/D		√		√	√				√								
134	IMODE	Improved multi-operator differential evolution	√			√	√				√	√							
135	IMTCMO	Improved evolutionary multitasking-based CMOEA		√		√	√	√	√	√		√							
136	IMTCMO_BS	Improved evolutionary multitasking-based CMOEA with bidirectional sampling		√	√	√	√	√	√	√		√							

	算法缩写	算法全称	single	multi	many	real	integer	label	binary	permutation	large	constrained	expensive	multimodal	sparse	dynamic	multitask	bilevel	robust
137	I-SIBEA	Interactive simple indicator-based evolutionary algorithm		√		√	√	√	√	√									
138	Izui	An aggregative gradient based multi-objective optimizer proposed by Izui et al.		√	√	√					√	√							
139	KLEA	Knowledge learning-based evolutionary algorithm		√		√	√		√		√	√			√				
140	KL-NSGA-II	Knowledge learning based NSGA-II		√		√	√	√	√	√		√				√			
141	KMA	Komodo mlipir algorithm	√			√	√				√	√							
142	KnEA	Knee point driven evolutionary algorithm			√	√	√	√	√	√		√							
143	K-RVEA	Surrogate-assisted RVEA		√	√	√	√						√						
144	KTA2	Kriging-assisted Two_Arch2		√	√	√	√						√						
145	KTS	Kriging-assisted evolutionary algorithm with two search modes		√	√		√					√	√						
146	L2SMEA	Linear subspace surrogate modeling assisted evolutionary algorithm	√			√							√						
147	LCMEA	Large-scale constrained multi-objective evolutionary algorithm		√		√					√	√							
148	LCSA	Linear combination-based search algorithm		√	√	√	√				√								
149	LDS-AF	Low-dimensional surrogate aggregation function		√		√	√				√		√						
150	LERD	Large-scale evolutionary algorithm with reformulated decision variable analysis		√	√	√					√								
151	LMEA	Evolutionary algorithm for large-scale many-objective optimization		√	√	√	√				√								
152	LMOCSO	Large-scale multi-objective competitive swarm optimization algorithm		√	√	√	√				√	√							
153	LMOEA-DS	Large-scale evolutionary multi-objective optimization assisted by directed sampling		√		√	√				√								
154	LMPFE	Evolutionary algorithm with local model based Pareto front estimation		√	√	√	√	√	√	√									
155	LRMOEA	Large-scale robust multi-objective evolutionary algorithm		√		√			√		√	√			√				√
156	LSMOF	Large-scale multi-objective optimization framework with NSGA-II		√		√	√				√								
157	MaOEA-CSS	Many-objective evolutionary algorithms based on coordinated selection		√	√	√	√	√	√	√									
158	MaOEA-DDFC	Many-objective evolutionary algorithm based on directional diversity and favorable convergence		√	√	√	√	√	√	√									
159	MaOEA/IGD	IGD based many-objective evolutionary algorithm			√	√	√	√	√	√									
160	MaOEA/IT	Many-objective evolutionary algorithms based on an independent two-stage		√	√	√	√					√							
161	MaOEA-R&D	Many-objective evolutionary algorithm based on objective space reduction			√	√	√	√	√	√									
162	MCCMO	Multi-population coevolutionary constrained multi-objective optimization		√		√	√	√	√	√		√							
163	MCEA/D	Multiple classifiers-assisted evolutionary algorithm based on decomposition		√	√	√	√						√						
164	MFEA	Multifactorial evolutionary algorithm	√			√	√	√	√	√	√						√		

	算法缩写	算法全称	single	multi	many	real	integer	label	binary	permutation	large	constrained	expensive	multimodal	sparse	dynamic	multitask	bilevel	robust
165	MFEA-II	Multifactorial evolutionary algorithm II	√			√	√	√	√	√	√						√		
166	MFFS	Multiform feature selection		√					√										
167	MFO-SPEA2	Multiform optimization framework based on SPEA2		√		√	√	√	√	√		√							
168	MGCEA	Multi-granularity clustering based evolutionary algorithm		√		√			√		√	√			√				
169	MGO	Mountain gazelle optimizer	√			√	√				√	√							
170	MGSAEA	Multigranularity surrogate-assisted constrained evolutionary algorithm		√		√						√	√						
171	MiSACO	Multi surrogate-assisted ant colony optimization	√			√	√						√						
172	MMEAPSL	Multimodal multi-objective evolutionary algorithm assisted by Pareto set learning		√		√	√	√	√	√				√					
173	MMEA-WI	Weighted indicator-based evolutionary algorithm for multimodal multi-objective optimization		√		√	√							√					
174	MMOPSO	MOPSO with multiple search strategies		√		√	√												
175	MO_Ring_PSO_SCD	Multiobjective PSO using ring topology and special crowding distance		√		√	√							√					
176	MOBCA	Multi-objective besiege and conquer algorithm		√		√	√												
177	MOCell	Cellular genetic algorithm		√		√	√	√	√	√		√							
178	MOCGDE	Multi-objective conjugate gradient and differential evolution algorithm		√	√	√					√	√							
179	MO-CMA	Multi-objective covariance matrix adaptation evolution strategy		√		√	√												
180	MOEA/CKF	Multi-objective evolutionary algorithm based on cross-scale knowledge fusion		√		√			√		√	√			√				
181	MOEA/D	Multiobjective evolutionary algorithm based on decomposition		√	√	√	√	√	√	√									
182	MOEA/D-2WA	MOEA/D with two-type weight vector adjustments		√	√	√	√	√	√	√		√							
183	MOEA/D-AWA	MOEA/D with adaptive weight adjustment		√	√	√	√	√	√	√									
184	MOEA/D-CMA	MOEA/D with covariance matrix adaptation evolution strategy		√	√	√	√												
185	MOEA/D-CMT	MOEA/D with competitive multitasking		√		√						√							
186	MOEA/DD	Many-objective evolutionary algorithm based on dominance and decomposition		√	√	√	√	√	√	√		√							
187	MOEA/D-DAE	MOEA/D with detect-and-escape strategy		√		√	√	√	√	√		√							
188	MOEA/D-DCWV	MOEA/D with distribution control of weight vector set		√	√	√	√	√	√	√									
189	MOEA/D-DE	MOEA/D based on differential evolution		√	√	√	√												
190	MOEA/D-DQN	MOEA/D based on deep Q-network		√	√	√	√												
191	MOEA/D-DRA	MOEA/D with dynamical resource allocation		√	√	√	√												
192	MOEA/D-DU	MOEA/D with a distance based updating strategy		√	√	√	√	√	√	√									
193	MOEA/D-DYTS	MOEA/D with dynamic Thompson sampling		√	√	√	√												
194	MOEA/D-EGO	MOEA/D with efficient global optimization		√		√	√						√						
195	MOEA/D-	MOEA/D with fitness-rate-rank-based		√	√	√	√												

	算法缩写	算法全称	single	multi	many	real	integer	label	binary	permutation	large	constrained	expensive	multimodal	sparse	dynamic	multitask	bilevel	robust
	FRRMAB	multiarmed bandit																	
196	MOEA/D-M2M	MOEA/D based on MOP to MOP		√		√	√												
197	MOEA/D-MRDL	MOEA/D with maximum relative diversity loss		√		√	√												
198	MOEA/D-PaS	MOEA/D with Pareto adaptive scalarizing approximation		√	√	√	√												
199	MOEA/D-PFE	MOEA/D with Pareto front estimation		√	√	√	√	√	√	√									
200	MOEA/D-STM	MOEA/D with stable matching		√	√	√	√												
201	MOEA/D-UR	MOEA/D with update when required		√	√	√	√	√	√	√									
202	MOEA/D-URAW	MOEA/D with uniform randomly adaptive weights		√	√	√	√	√	√	√									
203	MOEA/DVA	Multi-objective evolutionary algorithm based on decision variable		√		√	√				√								
204	MOEA/D-VOV	MOEA/D with virtual objective vectors		√	√	√	√	√	√	√									
205	MOEA/IGD-NS	Multi-objective evolutionary algorithm based on an enhanced IGD		√		√	√	√	√	√									
206	MOEA-NZD	Multi-objective evolutionary algorithm with nonzero detection		√	√	√					√	√			√				
207	MOEA-PC	Multiobjective evolutionary algorithm based on polar coordinates		√		√	√												
208	MOEA/PSL	Multi-objective evolutionary algorithm based on Pareto optimal subspace		√		√	√		√		√	√			√				
209	MOEA-RE	Multi-objective evolutionary algorithm with robustness enhancement		√		√	√	√	√	√									√
210	MO-EGS	Multi-objective evolutionary gradient search		√		√					√								
211	MO-L2SMEA	Multi-objective linear subspace surrogate modeling assisted evolutionary algorithm		√		√					√		√						
212	MOMBI-II	Many objective metaheuristic based on the R2 indicator II		√	√	√	√	√	√	√									
213	MO-MFEA	Multi-objective multifactorial evolutionary algorithm		√		√	√	√	√	√		√					√		
214	MO-MFEA-II	Multi-objective multifactorial evolutionary algorithm II		√		√	√	√	√	√		√					√		
215	MOMFEA-SADE	Multi-objective multifactorial evolutionary algorithm with subspace alignment and adaptive differential evolution		√		√	√	√	√	√		√					√		
216	MONAS	Multi-objective neural architecture search		√		√	√	√	√	√				√					
217	MOPSO	Multi-objective particle swarm optimization		√		√	√												
218	MOPSO-CD	MOPSO with crowding distance		√		√	√												
219	MOSD	Multiobjective steepest descent		√		√					√	√							
220	M-PAES	Memetic algorithm with Pareto archived evolution strategy		√		√	√												
221	MP-MMEA	Multi-population multi-modal multi-objective evolutionary algorithm		√		√	√				√			√	√				
222	MPSO/D	Multi-objective particle swarm optimization algorithm based on decomposition		√	√	√	√												

	算法缩写	算法全称	single	multi	many	real	integer	label	binary	permutation	large	constrained	expensive	multimodal	sparse	dynamic	multitask	bilevel	robust
223	MSCEA	Multi-stage constrained multi-objective evolutionary algorithm		√		√	√	√	√	√		√							
224	MSCMO	Multi-stage constrained multi-objective evolutionary algorithm		√		√	√	√	√	√		√							
225	MSEA	Multi-stage multi-objective evolutionary algorithm		√		√	√	√	√	√									
226	MSKEA	Multi-stage knowledge-guided evolutionary algorithm		√		√	√		√		√	√			√				
227	MSOPS-II	Multiple single objective Pareto sampling II		√	√	√	√					√							
228	MTCMO	Multitasking constrained multi-objective optimization		√		√	√	√	√	√		√							
229	MTDE-MKTA	Multitasking differential evolution with multiple knowledge types and transfer adaptation		√		√	√	√	√	√		√					√		
230	MTEA/D-DN	Multiobjective multitask evolutionary algorithm based on decomposition with dual neighborhoods		√		√	√	√	√	√		√					√		
231	MTS	Multiple trajectory search		√		√	√												
232	MultiObjective EGO	Multi-objective efficient global optimization		√		√	√					√	√						
233	MVPA	Most valuable player algorithm	√			√	√				√	√							
234	MyO-DEMR	Many-objective differential evolution with mutation restriction		√	√	√	√												
235	NBLEA	Nested bilevel evolutionary algorithm		√		√						√						√	
236	NelderMead	The Nelder-Mead algorithm	√			√													
237	NMPSO	Novel multi-objective particle swarm optimization		√	√	√	√												
238	NNDREA-MO	Evolutionary algorithm with neural network-based dimensionality reduction (multi-objective)		√					√		√	√			√				
239	NNDREA-SO	Evolutionary algorithm with neural network-based dimensionality reduction (single-objective)	√						√		√	√			√				
240	NNIA	Nondominated neighbor immune algorithm		√		√	√	√	√	√									
241	NRV-MOEA	Adaptive normal reference vector-based multi- and many-objective evolutionary algorithm		√	√	√	√	√	√	√									
242	NSBiDiCo	Non-dominated sorting bidirectional differential coevolution algorithm		√		√	√	√	√	√		√							
243	NSGA-II	Nondominated sorting genetic algorithm II		√		√	√	√	√	√		√							
244	NSGA-II+ARSBX	NSGA-II with adaptive rotation based simulated binary crossover		√		√	√					√							
245	NSGA-II-conflict	NSGA-II with conflict-based partitioning strategy			√	√	√	√	√	√									
246	NSGA-II-DTI	NSGA-II of Deb's type I robust version		√		√	√	√	√	√		√							√
247	NSGA-III	Nondominated sorting genetic algorithm III		√	√	√	√	√	√	√		√							
248	NSGAIII-EHVI	NSGA-III with expected hypervolume improvement		√	√	√							√						
249	NSGA-II/SDR	NSGA-II with strengthened dominance relation			√	√	√	√	√	√									
250	NSLS	Multiobjective optimization framework based on nondominated sorting and local search		√		√	√												
251	NUCEA	Non-uniform clustering based evolutionary algorithm		√		√			√		√	√			√				
252	OFA	Optimal foraging algorithm	√			√	√				√	√							

	算法缩写	算法全称	single	multi	many	real	integer	label	binary	permutation	large	constrained	expensive	multimodal	sparse	dynamic	multitask	bilevel	robust
253	one-by-one EA	Many-objective evolutionary algorithm using a one-by-one selection		√	√	√	√	√	√	√									
254	OSP-NSDE	Non-dominated sorting differential evolution with prediction in the objective space		√		√	√												
255	ParEGO	Efficient global optimization for Pareto optimization		√		√	√						√						
256	PB-NSGA-III	NSGA-III based on Pareto based bi-indicator infill sampling criterion		√	√	√	√						√						
257	PB-RVEA	RVEA based on Pareto based bi-indicator infill sampling criterion		√	√	√	√						√						
258	PC-SAEA	Pairwise comparison based surrogate-assisted evolutionary algorithm		√	√								√						
259	PEA	Pareto-based Kriging-assisted constrained multiobjective evolutionary algorithm		√		√	√					√	√						
260	PEAplus	Pareto-based Kriging-assisted constrained multiobjective evolutionary algorithm plus		√		√	√					√	√						
261	PeEA	Pareto front shape estimation based evolutionary algorithm		√	√	√	√	√	√	√									
262	PESA-II	Pareto envelope-based selection algorithm II		√		√	√	√	√	√									
263	PICEA-g	Preference-inspired coevolutionary algorithm with goals		√	√	√	√	√	√	√									
264	PIEA	Performance indicator-based evolutionary algorithm		√	√	√							√						
265	PIMD	Probability and mapping crowding distance		√	√	√							√						
266	PM-MOEA	Pattern mining based multi-objective evolutionary algorithm		√		√	√		√		√	√			√				
267	POCEA	Paired offspring generation based constrained evolutionary algorithm		√		√	√				√	√							
268	PPS	Push and pull search algorithm		√	√	√	√					√							
269	PRDH	Problem reformulation and duplication handling		√					√										
270	PREA	Promising-region based EMO algorithm		√	√	√	√	√	√	√									
271	PSO	Particle swarm optimization	√			√	√				√	√							
272	REMO	Expensive multiobjective optimization by relation learning and prediction		√	√	√							√						
273	RGA-M1-2	Real-coded genetic algorithm with framework M1-2		√		√						√	√						
274	RGA-M2-2	Real-coded genetic algorithm with framework M2-2		√		√						√	√						
275	RM-MEDA	Regularity model-based multiobjective estimation of distribution		√		√	√												
276	RMOEA/DVA	Robust multi-objective evolutionary algorithm with decision variable assortment		√		√	√												√
277	RMSProp	Root mean square propagation	√			√					√								
278	r-NSGA-II	r-dominance based NSGA-II		√		√	√	√	√	√									
279	RPD-NSGA-II	Reference point dominance-based NSGA-II		√	√	√	√	√	√	√									
280	RPEA	Reference points-based evolutionary algorithm			√	√	√	√	√	√									
281	RSEA	Radial space division based evolutionary algorithm		√	√	√	√	√	√	√									

	算法缩写	算法全称	single	multi	many	real	integer	label	binary	permutation	large	constrained	expensive	multimodal	sparse	dynamic	multitask	bilevel	robust
282	RVEA	Reference vector guided evolutionary algorithm		√	√	√	√	√	√	√		√							
283	RVEAa	RVEA embedded with the reference vector regeneration strategy			√	√	√	√	√	√									
284	RVEA-iGNG	RVEA based on improved growing neural gas		√	√	√	√	√	√	√									
285	S3-CMA-ES	Scalable small subpopulations based covariance matrix adaptation		√	√	√	√				√								
286	SA	Simulated annealing	√			√	√				√	√							
287	SACC-EAM-II	Surrogate-assisted cooperative co-evolutionary algorithm of Minamo	√			√	√						√						
288	SACOSO	Surrogate-assisted cooperative swarm optimization	√			√	√				√		√						
289	SADE-AMSS	Surrogate-assisted differential evolution with adaptive multi-subspace search	√			√	√						√						
290	SADE-ATDSC	Surrogate-assisted differential evolution with adaptation of training data selection criterion	√			√	√						√						
291	SADE-Sammon	Sammon mapping assisted differential evolution	√			√	√						√						
292	SAMOEA-TL2M	Surrogate-assisted multiobjective evolutionary algorithm based on two-level model management		√	√	√	√						√						
293	SAMSO	Multiswarm-assisted expensive optimization	√			√	√				√		√						
294	SAPO	Surrogate-assisted partial optimization	√			√	√					√	√						
295	S-CDAS	Self-controlling dominance area of solutions			√	√	√	√	√	√									
296	SCEA	Sparsity clustering basec evolutionary algorithm		√		√			√		√	√			√				
297	SD	Steepest descent	√			√					√								
298	S-ECSO	Enhanced competitive swarm optimizer for sparse optimization		√		√					√				√				
299	SFADE	Scalarization function approximation based differential evolution algorithm		√	√	√	√						√						
300	SGEA	Steady-state and generational evolutionary algorithm		√		√	√	√	√	√		√				√			
301	SGECF	Sparsity-guided elitism co-evolutionary framework		√		√			√		√	√			√				
302	SHADE	Success-history based adaptive differential evolution	√			√	√				√	√							
303	SIBEA	Simple indicator-based evolutionary algorithm		√		√	√	√	√	√									
304	SIBEA-kEMOSS	SIBEA with minimum objective subset of size k with minimum error			√	√	√	√	√	√									
305	SLMEA	Super-large-scale multi-objective evolutionary algorithm		√		√	√		√		√	√			√				
306	SMEA	Self-organizing multiobjective evolutionary algorithm		√		√	√												
307	SMOA	Supervised multi-objective optimization algorithm		√		√							√						
308	SMPSO	Speed-constrained multi-objective particle swarm optimization		√		√	√												
309	SMS-EGO	S metric selection based efficient global optimization		√		√	√						√						
310	SMS-EMOA	S metric selection based evolutionary multiobjective optimization		√		√	√	√	√	√									
311	S-NSGA-II	Sparse NSGA-II		√		√					√	√			√				

	算法缩写	算法全称	single	multi	many	real	integer	label	binary	permutation	large	constrained	expensive	multimodal	sparse	dynamic	multitask	bilevel	robust
312	SparseEA	Evolutionary algorithm for sparse multi-objective optimization problems		√		√	√		√		√	√			√				
313	SparseEA2	Improved SparseEA		√		√	√		√		√	√			√				
314	SPEA2	Strength Pareto evolutionary algorithm 2		√		√	√	√	√	√									
315	SPEA2+SDE	SPEA2 with shift-based density estimation			√	√	√	√	√	√									
316	SPEA/R	Strength Pareto evolutionary algorithm based on reference direction		√	√	√	√	√	√	√									
317	SQP	Sequential quadratic programming	√			√					√	√							
318	SRA	Stochastic ranking algorithm			√	√	√	√	√	√									
319	SSCEA	Subspace segmentation based co-evolutionary algorithm		√	√	√	√												
320	SSDE	Self-organized surrogate-assisted differential evolution		√	√	√	√					√	√						
321	SSIO-RL	Search space independent operator based deep reinforcement learning	√			√	√				√	√							
322	SVR-NSGA-II	Support vector regression based NSGA-II		√		√	√	√	√	√		√				√			
323	t-DEA	theta-dominance based evolutionary algorithm		√	√	√	√	√	√	√									
324	tDEA-CPBI	Theta-dominance based evolutionary algorithm with CPBI		√	√	√	√	√	√	√		√							
325	TEA	Two-phase evolutionary algorithm		√	√	√	√					√	√						
326	TELSO	Two-layer encoding learning swarm optimizer		√		√			√		√	√			√				
327	TiGE-2	Tri-Goal Evolution Framework for CMaOPs			√	√	√	√	√	√		√							
328	ToP	Two-phase framework with NSGA-II		√		√	√					√							
329	TPCMaO	Three-population based constrained many-objective co-evolutionary algorithm			√	√	√	√	√	√		√							
330	TriMOEA-TA&R	Multi-modal MOEA using two-archive and recombination strategies		√		√	√							√					
331	TS-NSGA-II	Two-stage NSGA-II		√	√	√	√	√	√	√									
332	TS-SparseEA	Two-stage SparseEA		√		√			√		√	√			√				
333	TSTI	Two-stage evolutionary algorithm with three indicators		√		√	√	√	√	√		√							
334	Two_Arch2	Two-archive algorithm 2		√	√	√	√	√	√	√									
335	URCMO	Utilizing the relationship between constrained and unconstrained Pareto fronts for constrained multi-objective optimization		√		√	√					√							
336	VaEA	Vector angle based evolutionary algorithm		√	√	√	√	√	√	√									
337	WASF-GA	Weighting achievement scalarizing function genetic algorithm		√		√	√	√	√	√									
338	WOA	Whale optimization algorithm	√			√	√				√	√							
339	WOF	Weighted optimization framework		√		√	√				√								
340	WV-MOEA-P	Weight vector based multi-objective optimization algorithm with preference		√		√	√												

六 问题列表

	问题缩写	问题全称	single	multi	many	real	integer	label	binary	permutation	large	constrained	expensive	multimodal	sparse	dynamic	multitask	bilevel	robust
1	BBOB_F1	Sphere function	√			√							√						
2	BBOB_F2	Ellipsoidal function	√			√							√						
3	BBOB_F3	Rastrigin function	√			√							√						
4	BBOB_F4	Buche-Rastrigin function	√			√							√						
5	BBOB_F5	Linear slope	√			√							√						
6	BBOB_F6	Attractive sector function	√			√							√						
7	BBOB_F7	Step ellipsoidal function	√			√							√						
8	BBOB_F8	Rosenbrock function	√			√							√						
9	BBOB_F9	Rotated Rosenbrock function	√			√							√						
10	BBOB_F10	Rotated ellipsoidal function	√			√							√						
11	BBOB_F11	Discus function	√			√							√						
12	BBOB_F12	Bent cigar function	√			√							√						
13	BBOB_F13	Sharp ridge function	√			√							√						
14	BBOB_F14	Different powers function	√			√							√						
15	BBOB_F15	Rastrigin function	√			√							√						
16	BBOB_F16	Weierstrass function	√			√							√						
17	BBOB_F17	Schaffers F7 function	√			√							√						
18	BBOB_F18	Moderately ill-conditioned Schaffers F7 function	√			√							√						
19	BBOB_F19	Composite Griewank-Rosenbrock function F8F2	√			√							√						
20	BBOB_F20	Schwefel function	√			√							√						
21	BBOB_F21	Gallagher's Gaussian 101-me peaks function	√			√							√						
22	BBOB_F22	Gallagher's Gaussian 21-hi peaks function	√			√							√						
23	BBOB_F23	Katsuura function	√			√							√						
24	BBOB_F24	Lunacek bi-Rastrigin function	√			√							√						
25	BT1	Benchmark MOP with bias feature		√		√					√								
26	BT2	Benchmark MOP with bias feature		√		√					√								
27	BT3	Benchmark MOP with bias feature		√		√					√								
28	BT4	Benchmark MOP with bias feature		√		√					√								
29	BT5	Benchmark MOP with bias feature		√		√					√								
30	BT6	Benchmark MOP with bias feature		√		√					√								
31	BT7	Benchmark MOP with bias feature		√		√					√								
32	BT8	Benchmark MOP with bias feature		√		√					√								
33	BT9	Benchmark MOP with bias feature		√		√					√								

	问题缩写	问题全称	single	multi	many	real	integer	label	binary	permutation	large	constrained	expensive	multimodal	sparse	dynamic	multitask	bilevel	robust
34	C10MOP1	Neural architecture search on CIFAR-10		√		√					√								
35	C10MOP2	Neural architecture search on CIFAR-10		√		√					√								
36	C10MOP3	Neural architecture search on CIFAR-10		√		√					√								
37	C10MOP4	Neural architecture search on CIFAR-10		√		√					√								
38	C10MOP5	Neural architecture search on CIFAR-10		√		√					√								
39	C10MOP6	Neural architecture search on CIFAR-10		√		√					√								
40	C10MOP7	Neural architecture search on CIFAR-10		√		√					√								
41	C10MOP8	Neural architecture search on CIFAR-10		√		√					√								
42	C10MOP9	Neural architecture search on CIFAR-10		√		√					√								
43	CEC2008_F1	Shifted sphere function	√			√					√		√						
44	CEC2008_F2	Shifted Schwefel's function	√			√					√		√						
45	CEC2008_F3	Shifted Rosenbrock's function	√			√					√		√						
46	CEC2008_F4	Shifted Rastrign's function	√			√					√		√						
47	CEC2008_F5	Shifted Griewank's function	√			√					√		√						
48	CEC2008_F6	Shifted Ackley's function	√			√					√		√						
49	CEC2008_F7	FastFractal 'DoubleDip' function	√			√					√		√						
50	CEC2010_F1	CEC'2010 constrained optimization benchmark problem	√			√						√							
51	CEC2010_F2	CEC'2010 constrained optimization benchmark problem	√			√						√							
52	CEC2010_F3	CEC'2010 constrained optimization benchmark problem	√			√						√							
53	CEC2010_F4	CEC'2010 constrained optimization benchmark problem	√			√						√							
54	CEC2010_F5	CEC'2010 constrained optimization benchmark problem	√			√						√							
55	CEC2010_F6	CEC'2010 constrained optimization benchmark problem	√			√						√							
56	CEC2010_F7	CEC'2010 constrained optimization benchmark problem	√			√						√							
57	CEC2010_F8	CEC'2010 constrained optimization benchmark problem	√			√						√							
58	CEC2010_F9	CEC'2010 constrained optimization benchmark problem	√			√						√							
59	CEC2010_F10	CEC'2010 constrained optimization benchmark problem	√			√						√							
60	CEC2010_F11	CEC'2010 constrained optimization benchmark problem	√			√						√							
61	CEC2010_F12	CEC'2010 constrained optimization benchmark problem	√			√						√							
62	CEC2010_F13	CEC'2010 constrained optimization benchmark problem	√			√						√							
63	CEC2010_F14	CEC'2010 constrained optimization benchmark problem	√			√						√							

	问题缩写	问题全称	single	multi	many	real	integer	label	binary	permutation	large	constrained	expensive	multimodal	sparse	dynamic	multitask	bilevel	robust
64	CEC2010_F15	CEC'2010 constrained optimization benchmark problem	√			√						√							
65	CEC2010_F16	CEC'2010 constrained optimization benchmark problem	√			√						√							
66	CEC2010_F17	CEC'2010 constrained optimization benchmark problem	√			√						√							
67	CEC2010_F18	CEC'2010 constrained optimization benchmark problem	√			√						√							
68	CEC2013_F1	Shifted elliptic function	√			√					√								
69	CEC2013_F2	Shifted Rastrigin's function	√			√					√								
70	CEC2013_F3	Shifted Ackley's function	√			√					√								
71	CEC2013_F4	7-nonseparable, 1-separable shifted and rotated elliptic function	√			√					√								
72	CEC2013_F5	7-nonseparable, 1-separable shifted and rotated Rastrigin's function	√			√					√								
73	CEC2013_F6	7-nonseparable, 1-separable shifted and rotated Ackley's function	√			√					√								
74	CEC2013_F7	7-nonseparable, 1-separable shifted and rotated Schwefel's function	√			√					√								
75	CEC2013_F8	20-nonseparable shifted and rotated elliptic function	√			√					√								
76	CEC2013_F9	20-nonseparable shifted and rotated Rastrigin's function	√			√					√								
77	CEC2013_F10	20-nonseparable shifted and rotated Rastrigin's function	√			√					√								
78	CEC2013_F11	20-nonseparable shifted and rotated Schwefel's function	√			√					√								
79	CEC2013_F12	Shifted Rosenbrock's function	√			√					√								
80	CEC2013_F13	Shifted Schwefel's function with conforming overlapping subcomponents	√			√					√								
81	CEC2013_F14	Shifted Schwefel's function with conflicting overlapping subcomponents	√			√					√								
82	CEC2013_F15	Shifted Schwefel's function	√			√					√								
83	CEC2017_F1	CEC'2017 constrained optimization benchmark problem	√			√						√							
84	CEC2017_F2	CEC'2017 constrained optimization benchmark problem	√			√						√							
85	CEC2017_F3	CEC'2017 constrained optimization benchmark problem	√			√						√							
86	CEC2017_F4	CEC'2017 constrained optimization benchmark problem	√			√						√							
87	CEC2017_F5	CEC'2017 constrained optimization benchmark problem	√			√						√							
88	CEC2017_F6	CEC'2017 constrained optimization benchmark problem	√			√						√							
89	CEC2017_F7	CEC'2017 constrained optimization benchmark problem	√			√						√							

	问题缩写	问题全称	single	multi	many	real	integer	label	binary	permutation	large	constrained	expensive	multimodal	sparse	dynamic	multitask	bilevel	robust
90	CEC2017_F8	CEC'2017 constrained optimization benchmark problem	√			√						√							
91	CEC2017_F9	CEC'2017 constrained optimization benchmark problem	√			√						√							
92	CEC2017_F10	CEC'2017 constrained optimization benchmark problem	√			√						√							
93	CEC2017_F11	CEC'2017 constrained optimization benchmark problem	√			√						√							
94	CEC2017_F12	CEC'2017 constrained optimization benchmark problem	√			√						√							
95	CEC2017_F13	CEC'2017 constrained optimization benchmark problem	√			√						√							
96	CEC2017_F14	CEC'2017 constrained optimization benchmark problem	√			√						√							
97	CEC2017_F15	CEC'2017 constrained optimization benchmark problem	√			√						√							
98	CEC2017_F16	CEC'2017 constrained optimization benchmark problem	√			√						√							
99	CEC2017_F17	CEC'2017 constrained optimization benchmark problem	√			√						√							
100	CEC2017_F18	CEC'2017 constrained optimization benchmark problem	√			√						√							
101	CEC2017_F19	CEC'2017 constrained optimization benchmark problem	√			√						√							
102	CEC2017_F20	CEC'2017 constrained optimization benchmark problem	√			√						√							
103	CEC2017_F21	CEC'2017 constrained optimization benchmark problem	√			√						√							
104	CEC2017_F22	CEC'2017 constrained optimization benchmark problem	√			√						√							
105	CEC2017_F23	CEC'2017 constrained optimization benchmark problem	√			√						√							
106	CEC2017_F24	CEC'2017 constrained optimization benchmark problem	√			√						√							
107	CEC2017_F25	CEC'2017 constrained optimization benchmark problem	√			√						√							
108	CEC2017_F26	CEC'2017 constrained optimization benchmark problem	√			√						√							
109	CEC2017_F27	CEC'2017 constrained optimization benchmark problem	√			√						√							
110	CEC2017_F28	CEC'2017 constrained optimization benchmark problem	√			√						√							
111	CEC2020_F1	Bent cigar function	√			√													
112	CEC2020_F2	Shifted and rotated Schwefel's function	√			√													
113	CEC2020_F3	Shifted and rotated Lunacek bi-Rastrigin function	√			√													
114	CEC2020_F4	Expanded Rosenbrock's plus Griewangk's function	√			√													

	问题缩写	问题全称	single	multi	many	real	integer	label	binary	permutation	large	constrained	expensive	multimodal	sparse	dynamic	multitask	bilevel	robust
115	CEC2020_F5	Hybrid function 1	√			√													
116	CEC2020_F6	Hybrid function 2	√			√													
117	CEC2020_F7	Hybrid function 3	√			√													
118	CEC2020_F8	Composition function 1	√			√													
119	CEC2020_F9	Composition function 2	√			√													
120	CEC2020_F10	Composition function 3	√			√													
121	CF1	Constrained benchmark MOP		√		√					√	√							
122	CF2	Constrained benchmark MOP		√		√					√	√							
123	CF3	Constrained benchmark MOP		√		√					√	√							
124	CF4	Constrained benchmark MOP		√		√					√	√							
125	CF5	Constrained benchmark MOP		√		√					√	√							
126	CF6	Constrained benchmark MOP		√		√					√	√							
127	CF7	Constrained benchmark MOP		√		√					√	√							
128	CF8	Constrained benchmark MOP		√		√					√	√							
129	CF9	Constrained benchmark MOP		√		√					√	√							
130	CF10	Constrained benchmark MOP		√		√					√	√							
131	CI_HS	Multitasking problem (Griewank function + Rastrigin function)	√			√					√						√		
132	CI_LS	Multitasking problem (Ackley function + Schwefel function)	√			√					√						√		
133	CI_MS	Multitasking problem (Ackley function + Rastrigin function)	√			√					√						√		
134	CitySegMOP1	Neural architecture search on Cityscape segmentation datasets		√		√					√		√						
135	CitySegMOP2	Neural architecture search on Cityscape segmentation datasets		√		√					√		√						
136	CitySegMOP3	Neural architecture search on Cityscape segmentation datasets		√		√					√		√						
137	CitySegMOP4	Neural architecture search on Cityscape segmentation datasets		√		√					√		√						
138	CitySegMOP5	Neural architecture search on Cityscape segmentation datasets		√		√					√		√						
139	CitySegMOP6	Neural architecture search on Cityscape segmentation datasets		√		√					√		√						
140	CitySegMOP7	Neural architecture search on Cityscape segmentation datasets		√		√					√		√						
141	CitySegMOP8	Neural architecture search on Cityscape segmentation datasets		√		√					√		√						
142	CitySegMOP9	Neural architecture search on Cityscape segmentation datasets		√		√					√		√						
143	CitySegMOP10	Neural architecture search on Cityscape segmentation datasets		√		√					√		√						
144	CitySegMOP11	Neural architecture search on Cityscape segmentation datasets		√		√					√		√						

	问题缩写	问题全称	single	multi	many	real	integer	label	binary	permutation	large	constrained	expensive	multimodal	sparse	dynamic	multitask	bilevel	robust
145	CitySegMOP12	Neural architecture search on Cityscape segmentation datasets		√		√					√		√						
146	CitySegMOP13	Neural architecture search on Cityscape segmentation datasets		√		√					√		√						
147	CitySegMOP14	Neural architecture search on Cityscape segmentation datasets		√		√					√		√						
148	CitySegMOP15	Neural architecture search on Cityscape segmentation datasets		√		√					√		√						
149	Community Detection	The community detection problem with label based encoding	√					√			√		√						
150	DAS-CMOP1	Difficulty-adjustable and scalable constrained benchmark MOP		√		√					√	√							
151	DAS-CMOP2	Difficulty-adjustable and scalable constrained benchmark MOP		√		√					√	√							
152	DAS-CMOP3	Difficulty-adjustable and scalable constrained benchmark MOP		√		√					√	√							
153	DAS-CMOP4	Difficulty-adjustable and scalable constrained benchmark MOP		√		√					√	√							
154	DAS-CMOP5	Difficulty-adjustable and scalable constrained benchmark MOP		√		√					√	√							
155	DAS-CMOP6	Difficulty-adjustable and scalable constrained benchmark MOP		√		√					√	√							
156	DAS-CMOP7	Difficulty-adjustable and scalable constrained benchmark MOP		√		√					√	√							
157	DAS-CMOP8	Difficulty-adjustable and scalable constrained benchmark MOP		√		√					√	√							
158	DAS-CMOP9	Difficulty-adjustable and scalable constrained benchmark MOP		√		√					√	√							
159	DOC1	Benchmark MOP with constraints in decision and objective spaces		√		√						√							
160	DOC2	Benchmark MOP with constraints in decision and objective spaces		√		√						√							
161	DOC3	Benchmark MOP with constraints in decision and objective spaces		√		√						√							
162	DOC4	Benchmark MOP with constraints in decision and objective spaces		√		√						√							
163	DOC5	Benchmark MOP with constraints in decision and objective spaces		√		√						√							
164	DOC6	Benchmark MOP with constraints in decision and objective spaces		√		√						√							
165	DOC7	Benchmark MOP with constraints in decision and objective spaces		√		√						√							
166	DOC8	Benchmark MOP with constraints in decision and objective spaces		√		√						√							
167	DOC9	Benchmark MOP with constraints in decision and objective spaces		√		√						√							
168	DSMOP1	Dynamic sparse multi-objective optimization problem		√	√	√					√				√	√			
169	DSMOP2	Dynamic sparse multi-objective optimization		√	√	√					√				√	√			

	问题缩写	问题全称	single	multi	many	real	integer	label	binary	permutation	large	constrained	expensive	multimodal	sparse	dynamic	multitask	bilevel	robust
		problem																	
170	DSMOP3	Dynamic sparse multi-objective optimization problem		√	√	√					√				√	√			
171	DSMOP4	Dynamic sparse multi-objective optimization problem		√	√	√					√				√	√			
172	DSMOP5	Dynamic sparse multi-objective optimization problem		√	√	√					√				√	√			
173	DSMOP6	Dynamic sparse multi-objective optimization problem		√	√	√					√				√	√			
174	DSMOP7	Dynamic sparse multi-objective optimization problem		√	√	√					√				√	√			
175	DSMOP8	Dynamic sparse multi-objective optimization problem		√	√	√					√				√	√			
176	DSMOP9	Dynamic sparse multi-objective optimization problem		√	√	√					√				√	√			
177	DSMOP10	Dynamic sparse multi-objective optimization problem		√	√	√					√				√	√			
178	DSMOP11	Dynamic sparse multi-objective optimization problem		√	√	√					√				√	√			
179	DSMOP12	Dynamic sparse multi-objective optimization problem		√	√	√					√				√	√			
180	DTLZ1	Benchmark MOP proposed by Deb, Thiele, Laumanns, and Zitzler		√	√	√					√		√						
181	DTLZ2	Benchmark MOP proposed by Deb, Thiele, Laumanns, and Zitzler		√	√	√					√		√						
182	DTLZ3	Benchmark MOP proposed by Deb, Thiele, Laumanns, and Zitzler		√	√	√					√		√						
183	DTLZ4	Benchmark MOP proposed by Deb, Thiele, Laumanns, and Zitzler		√	√	√					√		√						
184	DTLZ5	Benchmark MOP proposed by Deb, Thiele, Laumanns, and Zitzler		√	√	√					√		√						
185	DTLZ6	Benchmark MOP proposed by Deb, Thiele, Laumanns, and Zitzler		√	√	√					√		√						
186	DTLZ7	Benchmark MOP proposed by Deb, Thiele, Laumanns, and Zitzler		√	√	√					√		√						
187	DTLZ8	Benchmark MOP proposed by Deb, Thiele, Laumanns, and Zitzler		√	√	√					√	√	√						
188	DTLZ9	Benchmark MOP proposed by Deb, Thiele, Laumanns, and Zitzler		√	√	√					√	√	√						
189	CDTLZ2	Convex DTLZ2		√	√	√					√		√						
190	IDTLZ1	Inverted DTLZ1		√	√	√					√		√						
191	IDTLZ2	Inverted DTLZ2		√	√	√					√		√						
192	SDTLZ1	Scaled DTLZ1		√	√	√					√		√						
193	SDTLZ2	Scaled DTLZ2		√	√	√					√		√						
194	C1-DTLZ1	Constrained DTLZ1		√	√	√					√	√	√						
195	C1-DTLZ3	Constrained DTLZ3		√	√	√					√	√	√						

	问题缩写	问题全称	single	multi	many	real	integer	label	binary	permutation	large	constrained	expensive	multimodal	sparse	dynamic	multitask	bilevel	robust
196	C2-DTLZ2	Constrained DTLZ2		√	√	√					√	√	√						
197	C3-DTLZ4	Constrained DTLZ4		√	√	√					√	√	√						
198	DC1-DTLZ1	DTLZ1 with constrains in decision space		√	√	√					√	√	√						
199	DC1-DTLZ3	DTLZ3 with constrains in decision space		√	√	√					√	√	√						
200	DC2-DTLZ1	DTLZ1 with constrains in decision space		√	√	√					√	√	√						
201	DC2-DTLZ3	DTLZ3 with constrains in decision space		√	√	√					√	√	√						
202	DC3-DTLZ1	DTLZ1 with constrains in decision space		√	√	√					√	√	√						
203	DC3-DTLZ3	DTLZ3 with constrains in decision space		√	√	√					√	√	√						
204	EOPCCV_F1	Expensive optimization problems with continuous and categorical variables	√			√		√					√						
205	EOPCCV_F2	Expensive optimization problems with continuous and categorical variables	√			√		√					√						
206	EOPCCV_F3	Expensive optimization problems with continuous and categorical variables	√			√		√					√						
207	EOPCCV_F4	Expensive optimization problems with continuous and categorical variables	√			√		√					√						
208	EOPCCV_F5	Expensive optimization problems with continuous and categorical variables	√			√		√					√						
209	EOPCCV_F6	Expensive optimization problems with continuous and categorical variables	√			√		√					√						
210	EOPCCV_F7	Expensive optimization problems with continuous and categorical variables	√			√		√					√						
211	EOPCCV_F8	Expensive optimization problems with continuous and categorical variables	√			√		√					√						
212	EOPCCV_F9	Expensive optimization problems with continuous and categorical variables	√			√		√					√						
213	EOPCCV_F10	Expensive optimization problems with continuous and categorical variables	√			√		√					√						
214	EOPCCV_F11	Expensive optimization problems with continuous and categorical variables	√			√		√					√						
215	EOPCCV_F12	Expensive optimization problems with continuous and categorical variables	√			√		√					√						
216	EOPCCV_F13	Expensive optimization problems with continuous and categorical variables	√			√		√					√						
217	EOPCCV_F14	Expensive optimization problems with continuous and categorical variables	√			√		√					√						
218	EOPCCV_F15	Expensive optimization problems with continuous and categorical variables	√			√		√					√						
219	EOPCCV_F16	Expensive optimization problems with continuous and categorical variables	√			√		√					√						
220	EOPCCV_F17	Expensive optimization problems with continuous and categorical variables	√			√		√					√						
221	EOPCCV_F18	Expensive optimization problems with continuous and categorical variables	√			√		√					√						
222	EOPCCV_F19	Expensive optimization problems with continuous and categorical variables	√			√		√					√						

	问题缩写	问题全称	single	multi	many	real	integer	label	binary	permutation	large	constrained	expensive	multimodal	sparse	dynamic	multitask	bilevel	robust
223	EOPCCV_F20	Expensive optimization problems with continuous and categorical variables	√			√		√					√						
224	EOPCCV_F21	Expensive optimization problems with continuous and categorical variables	√			√		√					√						
225	EOPCCV_F22	Expensive optimization problems with continuous and categorical variables	√			√		√					√						
226	EOPCCV_F23	Expensive optimization problems with continuous and categorical variables	√			√		√					√						
227	EOPCCV_F24	Expensive optimization problems with continuous and categorical variables	√			√		√					√						
228	EOPCCV_F25	Expensive optimization problems with continuous and categorical variables	√			√		√					√						
229	EOPCCV_F26	Expensive optimization problems with continuous and categorical variables	√			√		√					√						
230	EOPCCV_F27	Expensive optimization problems with continuous and categorical variables	√			√		√					√						
231	EOPCCV_F28	Expensive optimization problems with continuous and categorical variables	√			√		√					√						
232	EOPCCV_F29	Expensive optimization problems with continuous and categorical variables	√			√		√					√						
233	EOPCCV_F30	Expensive optimization problems with continuous and categorical variables	√			√		√					√						
234	FCP1	Benchmark constrained MOP proposed by Yuan		√		√						√							
235	FCP2	Benchmark constrained MOP proposed by Yuan		√		√						√							
236	FCP3	Benchmark constrained MOP proposed by Yuan		√		√						√							
237	FCP4	Benchmark constrained MOP proposed by Yuan		√		√						√							
238	FCP5	Benchmark constrained MOP proposed by Yuan		√		√						√							
239	FDA1	Benchmark dynamic MOP proposed by Farina, Deb, and Amato		√		√					√					√			
240	FDA2	Benchmark dynamic MOP proposed by Farina, Deb, and Amato		√		√					√					√			
241	FDA3	Benchmark dynamic MOP proposed by Farina, Deb, and Amato		√		√					√					√			
242	FDA4	Benchmark dynamic MOP proposed by Farina, Deb, and Amato		√		√					√					√			
243	FDA5	Benchmark dynamic MOP proposed by Farina, Deb, and Amato		√		√					√					√			
244	GLSMOP1	General large-scale benchmark MOP		√	√	√					√		√						
245	GLSMOP2	General large-scale benchmark MOP		√	√	√					√		√						
246	GLSMOP3	General large-scale benchmark MOP		√	√	√					√		√						
247	GLSMOP4	General large-scale benchmark MOP		√	√	√					√		√						
248	GLSMOP5	General large-scale benchmark MOP		√	√	√					√		√						
249	GLSMOP6	General large-scale benchmark MOP		√	√	√					√		√						
250	GLSMOP7	General large-scale benchmark MOP		√	√	√					√		√						
251	GLSMOP8	General large-scale benchmark MOP		√	√	√					√		√						
252	GLSMOP9	General large-scale benchmark MOP		√	√	√					√		√						

	问题缩写	问题全称	single	multi	many	real	integer	label	binary	permutation	large	constrained	expensive	multimodal	sparse	dynamic	multitask	bilevel	robust
253	IMMOEA_F1	Benchmark MOP for testing IM-MOEA		√		√					√								
254	IMMOEA_F2	Benchmark MOP for testing IM-MOEA		√		√					√								
255	IMMOEA_F3	Benchmark MOP for testing IM-MOEA		√		√					√								
256	IMMOEA_F4	Benchmark MOP for testing IM-MOEA		√		√					√								
257	IMMOEA_F5	Benchmark MOP for testing IM-MOEA		√		√					√								
258	IMMOEA_F6	Benchmark MOP for testing IM-MOEA		√		√					√								
259	IMMOEA_F7	Benchmark MOP for testing IM-MOEA		√		√					√								
260	IMMOEA_F8	Benchmark MOP for testing IM-MOEA		√		√					√								
261	IMMOEA_F9	Benchmark MOP for testing IM-MOEA		√		√					√								
262	IMMOEA_F10	Benchmark MOP for testing IM-MOEA		√		√					√								
263	IMOP1	Benchmark MOP with irregular Pareto front		√		√							√						
264	IMOP2	Benchmark MOP with irregular Pareto front		√		√							√						
265	IMOP3	Benchmark MOP with irregular Pareto front		√		√							√						
266	IMOP4	Benchmark MOP with irregular Pareto front		√		√							√						
267	IMOP5	Benchmark MOP with irregular Pareto front		√		√							√						
268	IMOP6	Benchmark MOP with irregular Pareto front		√		√							√						
269	IMOP7	Benchmark MOP with irregular Pareto front		√		√							√						
270	IMOP8	Benchmark MOP with irregular Pareto front		√		√							√						
271	IN1KMOP1	Neural architecture search on ImageNet 1K		√		√					√		√						
272	IN1KMOP2	Neural architecture search on ImageNet 1K		√		√					√		√						
273	IN1KMOP3	Neural architecture search on ImageNet 1K		√		√					√		√						
274	IN1KMOP4	Neural architecture search on ImageNet 1K		√		√					√		√						
275	IN1KMOP5	Neural architecture search on ImageNet 1K		√		√					√		√						
276	IN1KMOP6	Neural architecture search on ImageNet 1K		√		√					√		√						
277	IN1KMOP7	Neural architecture search on ImageNet 1K		√		√					√		√						
278	IN1KMOP8	Neural architecture search on ImageNet 1K		√		√					√		√						
279	IN1KMOP9	Neural architecture search on ImageNet 1K		√		√					√		√						
280	Instance1	Multitasking multi-objective problem (ZDT4-R + ZDT4-G)		√		√					√						√		
281	Instance2	Multitasking multi-objective problem (ZDT4-RC + ZDT4-A)		√		√					√	√					√		
282	KP	The knapsack problem	√						√		√	√							
283	LIR-CMOP1	Constrained benchmark MOP with large infeasible regions		√		√					√	√							
284	LIR-CMOP2	Constrained benchmark MOP with large infeasible regions		√		√					√	√							
285	LIR-CMOP3	Constrained benchmark MOP with large infeasible regions		√		√					√	√							
286	LIR-CMOP4	Constrained benchmark MOP with large infeasible regions		√		√					√	√							

	问题缩写	问题全称	single	multi	many	real	integer	label	binary	permutation	large	constrained	expensive	multimodal	sparse	dynamic	multitask	bilevel	robust
287	LIR-CMOP5	Constrained benchmark MOP with large infeasible regions		√		√					√	√							
288	LIR-CMOP6	Constrained benchmark MOP with large infeasible regions		√		√					√	√							
289	LIR-CMOP7	Constrained benchmark MOP with large infeasible regions		√		√					√	√							
290	LIR-CMOP8	Constrained benchmark MOP with large infeasible regions		√		√					√	√							
291	LIR-CMOP9	Constrained benchmark MOP with large infeasible regions		√		√					√	√							
292	LIR-CMOP10	Constrained benchmark MOP with large infeasible regions		√		√					√	√							
293	LIR-CMOP11	Constrained benchmark MOP with large infeasible regions		√		√					√	√							
294	LIR-CMOP12	Constrained benchmark MOP with large infeasible regions		√		√					√	√							
295	LIR-CMOP13	Constrained benchmark MOP with large infeasible regions		√		√					√	√							
296	LIR-CMOP14	Constrained benchmark MOP with large infeasible regions		√		√					√	√							
297	LRMOP1	Large-scale robust multi-objective benchmark problem		√	√	√					√		√		√				√
298	LRMOP2	Large-scale robust multi-objective benchmark problem		√	√	√					√		√		√				√
299	LRMOP3	Large-scale robust multi-objective benchmark problem		√	√	√					√		√		√				√
300	LRMOP4	Large-scale robust multi-objective benchmark problem		√	√	√					√		√		√				√
301	LRMOP5	Large-scale robust multi-objective benchmark problem		√	√	√					√		√		√				√
302	LRMOP6	Large-scale robust multi-objective benchmark problem		√	√	√					√		√		√				√
303	LSCM1	Large-scale constrained multiobjective benchmark problem		√		√					√	√							
304	LSCM2	Large-scale constrained multiobjective benchmark problem		√		√					√	√							
305	LSCM3	Large-scale constrained multiobjective benchmark problem		√		√					√	√							
306	LSCM4	Large-scale constrained multiobjective benchmark problem		√		√					√	√							
307	LSCM5	Large-scale constrained multiobjective benchmark problem		√		√					√	√							
308	LSCM6	Large-scale constrained multiobjective benchmark problem		√		√					√	√							
309	LSCM7	Large-scale constrained multiobjective benchmark problem		√		√					√	√							
310	LSCM8	Large-scale constrained multiobjective benchmark problem		√		√					√	√							
311	LSCM9	Large-scale constrained multiobjective		√		√					√	√							

	问题缩写	问题全称	single	multi	many	real	integer	label	binary	permutation	large	constrained	expensive	multimodal	sparse	dynamic	multitask	bilevel	robust
		benchmark problem																	
312	LSCM10	Large-scale constrained multiobjective benchmark problem		√		√					√	√							
313	LSCM11	Large-scale constrained multiobjective benchmark problem		√		√					√	√							
314	LSCM12	Large-scale constrained multiobjective benchmark problem		√		√					√	√							
315	LSMOP1	Large-scale benchmark MOP		√	√	√					√								
316	LSMOP2	Large-scale benchmark MOP		√	√	√					√								
317	LSMOP3	Large-scale benchmark MOP		√	√	√					√								
318	LSMOP4	Large-scale benchmark MOP		√	√	√					√								
319	LSMOP5	Large-scale benchmark MOP		√	√	√					√								
320	LSMOP6	Large-scale benchmark MOP		√	√	√					√								
321	LSMOP7	Large-scale benchmark MOP		√	√	√					√								
322	LSMOP8	Large-scale benchmark MOP		√	√	√					√								
323	LSMOP9	Large-scale benchmark MOP		√	√	√					√								
324	MaF1	Inverted DTLZ1		√	√	√					√								
325	MaF2	DTLZ2BZ		√	√	√					√								
326	MaF3	Convex DTLZ3		√	√	√					√								
327	MaF4	Inverted and scaled DTLZ3		√	√	√					√								
328	MaF5	Scaled DTLZ4		√	√	√					√								
329	MaF6	DTLZ5IM		√	√	√					√								
330	MaF7	DTLZ7		√	√	√					√								
331	MaF8	MP-DMP		√	√	√													
332	MaF9	ML-DMP		√	√	√													
333	MaF10	WFG1		√	√	√					√								
334	MaF11	WFG2		√	√	√					√								
335	MaF12	WFG9		√	√	√					√								
336	MaF13	P7		√	√	√					√								
337	MaF14	LSMOP3		√	√	√					√								
338	MaF15	Inverted LSMOP8		√	√	√					√								
339	MaOPP_binary	Many-objective pathfinding problem based on binary encoding			√				√		√		√						
340	MaOPP_real	Many-objective pathfinding problem based on real encoding			√	√					√		√						
341	Mario	Play with Mario	√				√	√											
342	MaxCut	The max-cut problem	√						√		√								
343	MLDMP	The multi-line distance minimization problem		√	√	√													
344	MMF1	Multi-modal multi-objective test function		√		√								√					
345	MMF2	Multi-modal multi-objective test function		√		√								√					

	问题缩写	问题全称	single	multi	many	real	integer	label	binary	permutation	large	constrained	expensive	multimodal	sparse	dynamic	multitask	bilevel	robust
346	MMF3	Multi-modal multi-objective test function		√		√								√					
347	MMF4	Multi-modal multi-objective test function		√		√								√					
348	MMF5	Multi-modal multi-objective test function		√		√								√					
349	MMF6	Multi-modal multi-objective test function		√		√								√					
350	MMF7	Multi-modal multi-objective test function		√		√								√					
351	MMF8	Multi-modal multi-objective test function		√		√								√					
352	MMMOP1	Multi-modal multi-objective optimization problem		√	√	√								√					
353	MMMOP2	Multi-modal multi-objective optimization problem		√	√	√								√					
354	MMMOP3	Multi-modal multi-objective optimization problem		√	√	√								√					
355	MMMOP4	Multi-modal multi-objective optimization problem		√	√	√								√					
356	MMMOP5	Multi-modal multi-objective optimization problem		√	√	√								√					
357	MMMOP6	Multi-modal multi-objective optimization problem		√	√	√								√					
358	MMOP_HS1	Large-scale sparse multitasking multi-objective optimization problem		√		√					√				√		√		
359	MMOP_HS2	Large-scale sparse multitasking multi-objective optimization problem		√		√					√				√		√		
360	MMOP_LS1	Large-scale sparse multitasking multi-objective optimization problem		√		√					√				√		√		
361	MMOP_LS2	Large-scale sparse multitasking multi-objective optimization problem		√		√					√				√		√		
362	MMOP_MS1	Large-scale sparse multitasking multi-objective optimization problem		√		√					√				√		√		
363	MMOP_MS2	Large-scale sparse multitasking multi-objective optimization problem		√		√					√				√		√		
364	MMOP_NS1	Large-scale sparse multitasking multi-objective optimization problem		√		√					√				√		√		
365	MMOP_NS2	Large-scale sparse multitasking multi-objective optimization problem		√		√					√				√		√		
366	MOEADDE_F1	Benchmark MOP for testing MOEA/D-DE		√		√					√								
367	MOEADDE_F2	Benchmark MOP for testing MOEA/D-DE		√		√					√								
368	MOEADDE_F3	Benchmark MOP for testing MOEA/D-DE		√		√					√								
369	MOEADDE_F4	Benchmark MOP for testing MOEA/D-DE		√		√					√								
370	MOEADDE_F5	Benchmark MOP for testing MOEA/D-DE		√		√					√								
371	MOEADDE_F6	Benchmark MOP for testing MOEA/D-DE		√		√					√								
372	MOEADDE_F7	Benchmark MOP for testing MOEA/D-DE		√		√					√								
373	MOEADDE_F8	Benchmark MOP for testing MOEA/D-DE		√		√					√								
374	MOEADDE_F9	Benchmark MOP for testing MOEA/D-DE		√		√					√								
375	MOEADM2M_F1	Benchmark MOP for testing MOEA/D-M2M		√		√					√								
376	MOEADM2M_F2	Benchmark MOP for testing MOEA/D-M2M		√		√					√								
377	MOEADM2M_F3	Benchmark MOP for testing MOEA/D-M2M		√		√					√								
378	MOEADM2M_F4	Benchmark MOP for testing MOEA/D-M2M		√		√					√								

	问题缩写	问题全称	single	multi	many	real	integer	label	binary	permutation	large	constrained	expensive	multimodal	sparse	dynamic	multitask	bilevel	robust
379	MOEADM2M_F5	Benchmark MOP for testing MOEA/D-M2M		√		√					√								
380	MOEADM2M_F6	Benchmark MOP for testing MOEA/D-M2M		√		√					√								
381	MOEADM2M_F7	Benchmark MOP for testing MOEA/D-M2M		√		√					√								
382	MOKP	The multi-objective knapsack problem		√	√				√		√	√							
383	MONRP	The multi-objective next release problem		√					√		√								
384	MOTSP	The multi-objective traveling salesman problem		√	√					√	√								
385	MPDMP	The multi-point distance minimization problem		√	√	√													
386	mQAP	The multi-objective quadratic assignment problem		√	√					√	√								
387	MW1	Constrained benchmark MOP proposed by Ma and Wang		√		√					√	√							
388	MW2	Constrained benchmark MOP proposed by Ma and Wang		√		√					√	√							
389	MW3	Constrained benchmark MOP proposed by Ma and Wang		√		√					√	√							
390	MW4	Constrained benchmark MOP proposed by Ma and Wang		√	√	√					√	√							
391	MW5	Constrained benchmark MOP proposed by Ma and Wang		√		√					√	√							
392	MW6	Constrained benchmark MOP proposed by Ma and Wang		√		√					√	√							
393	MW7	Constrained benchmark MOP proposed by Ma and Wang		√		√					√	√							
394	MW8	Constrained benchmark MOP proposed by Ma and Wang		√	√	√					√	√							
395	MW9	Constrained benchmark MOP proposed by Ma and Wang		√		√					√	√							
396	MW10	Constrained benchmark MOP proposed by Ma and Wang		√		√					√	√							
397	MW11	Constrained benchmark MOP proposed by Ma and Wang		√		√					√	√							
398	MW12	Constrained benchmark MOP proposed by Ma and Wang		√		√					√	√							
399	MW13	Constrained benchmark MOP proposed by Ma and Wang		√		√					√	√							
400	MW14	Constrained benchmark MOP proposed by Ma and Wang		√	√	√					√	√							
401	NI_HS	Multitasking problem (Rosenbrock function + Rastrigin function)	√			√					√						√		
402	NI_MS	Multitasking problem (Griewank function + Weierstrass function)	√			√					√						√		
403	RMMEDA_F1	Benchmark MOP for testing RM-MEDA		√		√					√								
404	RMMEDA_F2	Benchmark MOP for testing RM-MEDA		√		√					√								
405	RMMEDA_F3	Benchmark MOP for testing RM-MEDA		√		√					√								
406	RMMEDA_F4	Benchmark MOP for testing RM-MEDA		√		√					√								
407	RMMEDA_F5	Benchmark MOP for testing RM-MEDA		√		√					√								

	问题缩写	问题全称	single	multi	many	real	integer	label	binary	permutation	large	constrained	expensive	multimodal	sparse	dynamic	multitask	bilevel	robust
408	RMMEDA_F6	Benchmark MOP for testing RM-MEDA		√		√					√								
409	RMMEDA_F7	Benchmark MOP for testing RM-MEDA		√		√					√								
410	RMMEDA_F8	Benchmark MOP for testing RM-MEDA		√		√					√								
411	RMMEDA_F9	Benchmark MOP for testing RM-MEDA		√		√					√								
412	RMMEDA_F10	Benchmark MOP for testing RM-MEDA		√		√					√								
413	RWMOP1	Pressure vessal problem		√		√						√							
414	RWMOP2	Vibrating platform		√		√						√							
415	RWMOP3	Two bar truss design problem		√		√						√							
416	RWMOP4	Weldan beam design problem		√		√						√							
417	RWMOP5	Disc brake design problem		√		√						√							
418	RWMOP6	Speed reducer design problem		√		√						√							
419	RWMOP7	Gear train design problem		√		√						√							
420	RWMOP8	Car side impact design problem		√		√						√							
421	RWMOP9	Four bar plane truss		√		√						√							
422	RWMOP10	Two bar plane truss		√		√						√							
423	RWMOP11	Water resource management problem		√		√						√							
424	RWMOP12	Simply supported I-beam design		√		√						√							
425	RWMOP13	Gear box design		√		√						√							
426	RWMOP14	Multiple-disk clutch brake design problem		√		√						√							
427	RWMOP15	Spring design problem		√		√						√							
428	RWMOP16	Cantilever beam design problem		√		√						√							
429	RWMOP17	Bulk carriers design problem		√		√						√							
430	RWMOP18	Front rail design problem		√		√						√							
431	RWMOP19	Multi-product batch plant		√		√						√							
432	RWMOP20	Hydro-static thrust bearing design problem		√		√						√							
433	RWMOP21	Crash energy management for high-speed train		√		√						√							
434	RWMOP22	Haverly's pooling problem		√		√						√							
435	RWMOP23	Reactor network design		√		√						√							
436	RWMOP24	Heat exchanger network design		√		√						√							
437	RWMOP25	Process synthesis problem		√		√						√							
438	RWMOP26	Process sythesis and design problem		√		√						√							
439	RWMOP27	Process flow sheeting problem		√		√						√							
440	RWMOP28	Two reactor problem		√		√						√							
441	RWMOP29	Process synthesis problem		√		√						√							
442	RWMOP30	Synchronous pptimal pulse-width modulation of 3-level inverters		√		√						√							
443	RWMOP31	Synchronous pptimal pulse-width modulation of 5-level inverters		√		√						√							

	问题缩写	问题全称	single	multi	many	real	integer	label	binary	permutation	large	constrained	expensive	multimodal	sparse	dynamic	multitask	bilevel	robust
444	RWMOP32	Synchronous pptimal pulse-width modulation of 7-level inverters		√		√						√							
445	RWMOP33	Synchronous pptimal pulse-width modulation of 9-level inverters		√		√						√							
446	RWMOP34	Synchronous pptimal pulse-width modulation of 11-level inverters		√		√						√							
447	RWMOP35	Synchronous pptimal pulse-width modulation of 13-level inverters		√		√						√							
448	RWMOP36	Optimal sizing of single phase distributed generation with reactive power support for phase balancing at main transformer/grid and active power loss		√		√						√							
449	RWMOP37	Optimal Sizing of Single Phase Distributed Generation with reactive power support for Phase Balancing at Main Transformer/Grid and reactive Power loss		√		√						√							
450	RWMOP38	Optimal sizing of single phase distributed generation with reactive power support for active and reactive power loss		√		√						√							
451	RWMOP39	Optimal sizing of single phase distributed generation with reactive power support for phase balancing at main transformer/grid and active and reactive power loss		√		√						√							
452	RWMOP40	Optimal power flow for minimizing active and reactive power loss		√		√						√							
453	RWMOP41	Optimal power flow for minimizing voltage deviation, active and reactive power loss		√		√						√							
454	RWMOP42	Optimal power flow for minimizing voltage deviation, and active power loss		√		√						√							
455	RWMOP43	Optimal power flow for minimizing fuel cost, and active power loss		√		√						√							
456	RWMOP44	Optimal power flow for minimizing fuel cost, active and reactive power loss		√		√						√							
457	RWMOP45	Optimal power flow for minimizing fuel cost, voltage deviation, and active power loss		√		√						√							
458	RWMOP46	Optimal power flow for minimizing fuel cost, voltage deviation, active and reactive power loss		√		√						√							
459	RWMOP47	Optimal droop setting for minimizing active and reactive power loss		√		√						√							
460	RWMOP48	Optimal droop setting for minimizing voltage deviation and active power loss		√		√						√							
461	RWMOP49	Optimal droop setting for minimizing voltage deviation, active, and reactive power loss		√		√						√							
462	RWMOP50	Power distribution system planning		√		√						√							
463	SDC1	Scalable high-dimensional decicsion constraint benchamrk		√		√						√							
464	SDC2	Scalable high-dimensional decicsion constraint benchamrk		√		√						√							
465	SDC3	Scalable high-dimensional decicsion constraint benchamrk		√		√						√							
466	SDC4	Scalable high-dimensional decicsion		√		√						√							

	问题缩写	问题全称	single	multi	many	real	integer	label	binary	permutation	large	constrained	expensive	multimodal	sparse	dynamic	multitask	bilevel	robust
		constraint benchamrk																	
467	SDC5	Scalable high-dimensional decicsion constraint benchamrk		√		√						√							
468	SDC6	Scalable high-dimensional decicsion constraint benchamrk		√		√						√							
469	SDC7	Scalable high-dimensional decicsion constraint benchamrk		√		√						√							
470	SDC8	Scalable high-dimensional decicsion constraint benchamrk		√		√						√							
471	SDC9	Scalable high-dimensional decicsion constraint benchamrk		√		√						√							
472	SDC10	Scalable high-dimensional decicsion constraint benchamrk		√		√						√							
473	SDC11	Scalable high-dimensional decicsion constraint benchamrk		√		√						√							
474	SDC12	Scalable high-dimensional decicsion constraint benchamrk		√		√						√							
475	SDC13	Scalable high-dimensional decicsion constraint benchamrk		√		√						√							
476	SDC14	Scalable high-dimensional decicsion constraint benchamrk		√		√						√							
477	SDC15	Scalable high-dimensional decicsion constraint benchamrk		√		√						√							
478	SMD1	Bilevel optimization problems proposed by Sinha, Malo, and Deb		√		√												√	
479	SMD2	Bilevel optimization problems proposed by Sinha, Malo, and Deb		√		√												√	
480	SMD3	Bilevel optimization problems proposed by Sinha, Malo, and Deb		√		√												√	
481	SMD4	Bilevel optimization problems proposed by Sinha, Malo, and Deb		√		√												√	
482	SMD5	Bilevel optimization problems proposed by Sinha, Malo, and Deb		√		√												√	
483	SMD6	Bilevel optimization problems proposed by Sinha, Malo, and Deb		√		√												√	
484	SMD7	Bilevel optimization problems proposed by Sinha, Malo, and Deb		√		√												√	
485	SMD8	Bilevel optimization problems proposed by Sinha, Malo, and Deb		√		√												√	
486	SMD9	Bilevel optimization problems proposed by Sinha, Malo, and Deb		√		√						√						√	
487	SMD10	Bilevel optimization problems proposed by Sinha, Malo, and Deb		√		√						√						√	
488	SMD11	Bilevel optimization problems proposed by Sinha, Malo, and Deb		√		√						√						√	
489	SMD12	Bilevel optimization problems proposed by Sinha, Malo, and Deb		√		√						√						√	
490	SO_ISCSO_2016	International student competition in structural optimization	√				√				√	√							

	问题缩写	问题全称	single	multi	many	real	integer	label	binary	permutation	large	constrained	expensive	multimodal	sparse	dynamic	multitask	bilevel	robust
491	SO_ISCSO_2017	International student competition in structural optimization	√				√				√	√							
492	SO_ISCSO_2018	International student competition in structural optimization	√				√				√	√							
493	SO_ISCSO_2019	International student competition in structural optimization	√				√				√	√							
494	SO_ISCSO_2021	International student competition in structural optimization	√				√				√	√							
495	SO_ISCSO_2022	International student competition in structural optimization	√				√				√	√							
496	Sparse_CD	The community detection problem		√					√		√		√		√				
497	Sparse_CN	The critical node detection problem		√					√		√		√		√				
498	Sparse_FS	The feature selection problem		√					√		√		√		√				
499	Sparse_IS	The instance selection problem		√					√		√		√		√				
500	Sparse_KP	The sparse multi-objective knapsack problem		√	√				√		√								
501	Sparse_NN	The neural network training problem		√		√					√		√		√				
502	Sparse_PM	The pattern mining problem		√					√		√		√		√				
503	Sparse_PO	The portfolio optimization problem		√		√					√		√		√				
504	Sparse_SR	The sparse signal reconstruction problem		√		√					√		√		√				
505	SMMOP1	Sparse multi-modal multi-objective optimization problem		√	√	√					√			√	√				
506	SMMOP2	Sparse multi-modal multi-objective optimization problem		√	√	√					√			√	√				
507	SMMOP3	Sparse multi-modal multi-objective optimization problem		√	√	√					√			√	√				
508	SMMOP4	Sparse multi-modal multi-objective optimization problem		√	√	√					√			√	√				
509	SMMOP5	Sparse multi-modal multi-objective optimization problem		√	√	√					√			√	√				
510	SMMOP6	Sparse multi-modal multi-objective optimization problem		√	√	√					√			√	√				
511	SMMOP7	Sparse multi-modal multi-objective optimization problem		√	√	√					√			√	√				
512	SMMOP8	Sparse multi-modal multi-objective optimization problem		√	√	√					√			√	√				
513	SMOP1	Benchmark MOP with sparse Pareto optimal solutions		√	√	√					√		√		√				
514	SMOP2	Benchmark MOP with sparse Pareto optimal solutions		√	√	√					√		√		√				
515	SMOP3	Benchmark MOP with sparse Pareto optimal solutions		√	√	√					√		√		√				
516	SMOP4	Benchmark MOP with sparse Pareto optimal solutions		√	√	√					√		√		√				
517	SMOP5	Benchmark MOP with sparse Pareto optimal solutions		√	√	√					√		√		√				
518	SMOP6	Benchmark MOP with sparse Pareto optimal		√	√	√					√		√		√				

	问题缩写	问题全称	single	multi	many	real	integer	label	binary	permutation	large	constrained	expensive	multimodal	sparse	dynamic	multitask	bilevel	robust
		solutions																	
519	SMOP7	Benchmark MOP with sparse Pareto optimal solutions		√	√	√					√		√		√				
520	SMOP8	Benchmark MOP with sparse Pareto optimal solutions		√	√	√					√		√		√				
521	SOP_F1	Sphere function	√			√							√						
522	SOP_F2	Schwefel's function 2.22	√			√							√						
523	SOP_F3	Schwefel's function 1.2	√			√							√						
524	SOP_F4	Schwefel's function 2.21	√			√							√						
525	SOP_F5	Generalized Rosenbrock's function	√			√							√						
526	SOP_F6	Step function	√			√							√						
527	SOP_F7	Quartic function with noise	√			√							√						
528	SOP_F8	Generalized Schwefel's function 2.26	√			√							√						
529	SOP_F9	Generalized Rastrigin's function	√			√							√						
530	SOP_F10	Ackley's function	√			√							√						
531	SOP_F11	Generalized Griewank's function	√			√							√						
532	SOP_F12	Generalized penalized function	√			√							√						
533	SOP_F13	Generalized penalized function	√			√							√						
534	SOP_F14	Shekel's foxholes function	√			√							√						
535	SOP_F15	Kowalik's function	√			√							√						
536	SOP_F16	Six-hump camel-back function	√			√							√						
537	SOP_F17	Branin function	√			√							√						
538	SOP_F18	Goldstein-price function	√			√							√						
539	SOP_F19	Hartman's family	√			√							√						
540	SOP_F20	Hartman's family	√			√							√						
541	SOP_F21	Shekel's family	√			√							√						
542	SOP_F22	Shekel's family	√			√							√						
543	SOP_F23	Shekel's family	√			√							√						
544	TP1	Test problem for robust multi-objective optimization		√		√					√								√
545	TP2	Test problem for robust multi-objective optimization		√		√					√								√
546	TP3	Test problem for robust multi-objective optimization		√		√					√								√
547	TP4	Test problem for robust multi-objective optimization		√		√					√								√
548	TP5	Test problem for robust multi-objective optimization		√		√					√								√
549	TP6	Test problem for robust multi-objective optimization		√		√					√								√
550	TP7	Test problem for robust multi-objective optimization		√		√					√								√
551	TP8	Test problem for robust multi-objective optimization		√		√					√								√
552	TP9	Test problem for robust multi-objective optimization		√		√					√								√
553	TP10	Test problem for robust multi-objective optimization		√		√					√	√							√
554	TREE1	The time-varying ratio error estimation problem		√		√					√	√	√						

	问题缩写	问题全称	single	multi	many	real	integer	label	binary	permutation	large	constrained	expensive	multimodal	sparse	dynamic	multitask	bilevel	robust
555	TREE2	The time-varying ratio error estimation problem		√		√					√	√	√						
556	TREE3	The time-varying ratio error estimation problem		√		√					√	√	√						
557	TREE4	The time-varying ratio error estimation problem		√		√					√	√	√						
558	TREE5	The time-varying ratio error estimation problem		√		√					√	√	√						
559	TREE6	The time-varying ratio error estimation problem		√		√					√	√	√						
560	TSP	The traveling salesman problem	√							√	√								
561	UF1	Unconstrained benchmark MOP		√		√					√								
562	UF2	Unconstrained benchmark MOP		√		√					√								
563	UF3	Unconstrained benchmark MOP		√		√					√								
564	UF4	Unconstrained benchmark MOP		√		√					√								
565	UF5	Unconstrained benchmark MOP		√		√					√								
566	UF6	Unconstrained benchmark MOP		√		√					√								
567	UF7	Unconstrained benchmark MOP		√		√					√								
568	UF8	Unconstrained benchmark MOP		√		√					√								
569	UF9	Unconstrained benchmark MOP		√		√					√								
570	UF10	Unconstrained benchmark MOP		√		√					√								
571	VNT1	Benchmark MOP proposed by Viennet		√		√													
572	VNT2	Benchmark MOP proposed by Viennet		√		√													
573	VNT3	Benchmark MOP proposed by Viennet		√		√													
574	VNT4	Benchmark MOP proposed by Viennet		√		√						√							
575	WFG1	Benchmark MOP proposed by Walking Fish Group		√	√	√					√		√						
576	WFG2	Benchmark MOP proposed by Walking Fish Group		√	√	√					√		√						
577	WFG3	Benchmark MOP proposed by Walking Fish Group		√	√	√					√		√						
578	WFG4	Benchmark MOP proposed by Walking Fish Group		√	√	√					√		√						
579	WFG5	Benchmark MOP proposed by Walking Fish Group		√	√	√					√		√						
580	WFG6	Benchmark MOP proposed by Walking Fish Group		√	√	√					√		√						
581	WFG7	Benchmark MOP proposed by Walking Fish Group		√	√	√					√		√						
582	WFG8	Benchmark MOP proposed by Walking Fish Group		√	√	√					√		√						
583	WFG9	Benchmark MOP proposed by Walking Fish Group		√	√	√					√		√						
584	ZCAT1	Benchmark MOP proposed by Zapotecas, Coello, Aguirre, and Tanaka		√	√	√					√		√						
585	ZCAT2	Benchmark MOP proposed by Zapotecas, Coello, Aguirre, and Tanaka		√	√	√					√		√						
586	ZCA3	Benchmark MOP proposed by Zapotecas, Coello, Aguirre, and Tanaka		√	√	√					√		√						
587	ZCA4	Benchmark MOP proposed by Zapotecas, Coello, Aguirre, and Tanaka		√	√	√					√		√						
588	ZCA5	Benchmark MOP proposed by Zapotecas, Coello, Aguirre, and Tanaka		√	√	√					√		√						
589	ZCAT6	Benchmark MOP proposed by Zapotecas,		√	√	√					√		√						

	问题缩写	问题全称	single	multi	many	real	integer	label	binary	permutation	large	constrained	expensive	multimodal	sparse	dynamic	multitask	bilevel	robust
		Coello, Aguirre, and Tanaka																	
590	ZCAT7	Benchmark MOP proposed by Zapotecas, Coello, Aguirre, and Tanaka		√	√	√					√		√						
591	ZCAT8	Benchmark MOP proposed by Zapotecas, Coello, Aguirre, and Tanaka		√	√	√					√		√						
592	ZCAT9	Benchmark MOP proposed by Zapotecas, Coello, Aguirre, and Tanaka		√	√	√					√		√						
593	ZCAT10	Benchmark MOP proposed by Zapotecas, Coello, Aguirre, and Tanaka		√	√	√					√		√						
594	ZCAT11	Benchmark MOP proposed by Zapotecas, Coello, Aguirre, and Tanaka		√	√	√					√		√						
595	ZCAT12	Benchmark MOP proposed by Zapotecas, Coello, Aguirre, and Tanaka		√	√	√					√		√						
596	ZCAT13	Benchmark MOP proposed by Zapotecas, Coello, Aguirre, and Tanaka		√	√	√					√		√						
597	ZCAT14	Benchmark MOP proposed by Zapotecas, Coello, Aguirre, and Tanaka		√	√	√					√		√						
598	ZCAT15	Benchmark MOP proposed by Zapotecas, Coello, Aguirre, and Tanaka		√	√	√					√		√						
599	ZCAT16	Benchmark MOP proposed by Zapotecas, Coello, Aguirre, and Tanaka		√	√	√					√		√						
600	ZCAT17	Benchmark MOP proposed by Zapotecas, Coello, Aguirre, and Tanaka		√	√	√					√		√						
601	ZCAT18	Benchmark MOP proposed by Zapotecas, Coello, Aguirre, and Tanaka		√	√	√					√		√						
602	ZCAT19	Benchmark MOP proposed by Zapotecas, Coello, Aguirre, and Tanaka		√	√	√					√		√						
603	ZCAT20	Benchmark MOP proposed by Zapotecas, Coello, Aguirre, and Tanaka		√	√	√					√		√						
604	ZDT1	Benchmark MOP proposed by Zitzler, Deb, and Thiele		√		√					√		√						
605	ZDT2	Benchmark MOP proposed by Zitzler, Deb, and Thiele		√		√					√		√						
606	ZDT3	Benchmark MOP proposed by Zitzler, Deb, and Thiele		√		√					√		√						
607	ZDT4	Benchmark MOP proposed by Zitzler, Deb, and Thiele		√		√					√		√						
608	ZDT5	Benchmark MOP proposed by Zitzler, Deb, and Thiele		√					√		√		√						
609	ZDT6	Benchmark MOP proposed by Zitzler, Deb, and Thiele		√		√					√		√						
610	ZXH_CF1	Constrained benchmark MOP proposed by Zhou, Xiang, and He		√	√	√					√	√							
611	ZXH_CF2	Constrained benchmark MOP proposed by Zhou, Xiang, and He		√	√	√					√	√							
612	ZXH_CF3	Constrained benchmark MOP proposed by Zhou, Xiang, and He		√	√	√					√	√							
613	ZXH_CF4	Constrained benchmark MOP proposed by Zhou, Xiang, and He		√	√	√					√	√							

	问题缩写	问题全称	single	multi	many	real	integer	label	binary	permutation	large	constrained	expensive	multimodal	sparse	dynamic	multitask	bilevel	robust
614	ZXH_CF5	Constrained benchmark MOP proposed by Zhou, Xiang, and He		√	√	√					√	√							
615	ZXH_CF6	Constrained benchmark MOP proposed by Zhou, Xiang, and He		√	√	√					√	√							
616	ZXH_CF7	Constrained benchmark MOP proposed by Zhou, Xiang, and He		√	√	√					√	√							
617	ZXH_CF8	Constrained benchmark MOP proposed by Zhou, Xiang, and He		√	√	√					√	√							
618	ZXH_CF9	Constrained benchmark MOP proposed by Zhou, Xiang, and He		√	√	√					√	√							
619	ZXH_CF10	Constrained benchmark MOP proposed by Zhou, Xiang, and He		√	√	√					√	√							
620	ZXH_CF11	Constrained benchmark MOP proposed by Zhou, Xiang, and He		√	√	√					√	√							
621	ZXH_CF12	Constrained benchmark MOP proposed by Zhou, Xiang, and He		√	√	√					√	√							
622	ZXH_CF13	Constrained benchmark MOP proposed by Zhou, Xiang, and He		√	√	√					√	√							
623	ZXH_CF14	Constrained benchmark MOP proposed by Zhou, Xiang, and He		√	√	√					√	√							
624	ZXH_CF15	Constrained benchmark MOP proposed by Zhou, Xiang, and He		√	√	√					√	√							
625	ZXH_CF16	Constrained benchmark MOP proposed by Zhou, Xiang, and He		√	√	√					√	√							