# PlatEMO

## 进化多目标优化平台

用户手册 3.3

生物智能与知识发现（BIMK）研究所

2021年8月14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：

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 提供一系列的元启发式算法用于求解各类优化问题。为此，用户需要定义优化问题、选择求解算法并设置参数。PlatEMO 提供以下三种调用方式：

1) 带参数调用主函数：

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

可以利用指定的算法来求解指定的测试问题并设置参数，求解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：

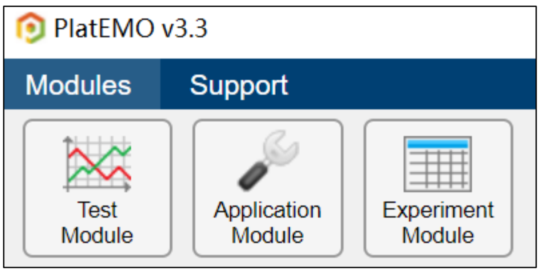
```
f1 = @(x,d)sum(x*d);
f2 = @(x,d)1-sum(x*d);
platemo('objFcn',f1,'conFcn',f2,'algorithm',@GA,…);
```

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();
```

可以弹出一个带有三个模块的图形界面，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）。



# 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 | 算法类 |
| 'problem' | 函数句柄或单元数组 | 不定 | 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 |
| 'save' | 整数 | 0 | 保存的种群数 |
| 'outputFcn' | 函数句柄 | @ALGORITHM.Output | 每代开始前调用的函数 |

- `'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。
- `'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,…` 指定了该算法中的参数值。
- `'N'`表示算法的种群大小，它通常等于最终种群中解的个数。
- `'M'`表示问题的目标数，它仅对一些多目标测试问题生效。
- `'D'`表示问题的变量数，它仅对一些测试问题生效。
- `'maxFE'`表示算法可使用的最大评价次数，它通常等于种群大小乘以迭代次数。
- `'save'`表示保存的种群数，该值大于零时结果将被保存在文件中，该值等

于零时结果将被显示在窗口中（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。

例如，以下代码利用遗传算法求球面函数的最小值，其中种群大小为 50 且最终种群会被显示在窗口中：

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

以下代码利用 NSGA-II 来求解 5 目标、40 变量的 DTLZ2 问题，其中最大评价次数为 20000 且最终种群会被保存在文件中：

```
platemo('algorithm',@NSGAII,'problem',@DTLZ2,'M',5,'D',40,'
maxFE',20000,'save',10);
```

以下代码利用基于 Tchebycheff 方法的 MOEA/D 求解 ZDT1 问题十次，其中每次的结果会被保存在独立的文件中：

```
for i = 1 : 10
    platemo('algorithm',{@MOEAD,2},'problem',@ZDT1,'save',5);
end
```

注意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 |
|---|---|---|---|
| 'encoding' | 字符串 | 'real' | 问题的编码方式 |
| 'objFcn' | 函数句柄或单元数组 | @(x,d)sum(x) | 问题的目标函数；所有目标函数均为最小化问题 |
| 'conFcn' | 函数句柄或单元数组 | @(x,d)0 | 问题的约束；当且仅当约束违反值小于等于零时，约束被满足 |
| 'lower' | 行向量 | 0 | 变量的下界 |
| 'upper' | 行向量 | 1 | 变量的上界 |
| 'initFcn' | 函数句柄 | [] | 种群初始化函数 |
| 'decFcn' | 函数句柄 | [] | 无效解修复函数 |
| 'parameter' | 单元数组 | {} | 问题的数据 |

- 'encoding'表示问题的编码方式，它的值可以是'real'（实数或整数变量）、'binary'（二进制变量）或'permutation'（序列变量）。算法针对不同的编码方式可能使用不同的算子来产生子代。
- '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）或一个单元数组（多目标）。每个目标函数必须有两个输入和一个输出，其中第一个输入是一个决策向量、第二个输入是由'parameter'指定的数据(当'parameter'未被指定时，该输入可以省略)、输出是目标值。所有目标函数均为最小化问题。
- 'conFcn'表示问题的约束，它的值可以是一个函数句柄（单约束）或一个单元数组（多约束）。每个约束函数必须有两个输入和一个输出，其中第一个输入是一个决策向量、第二个输入是由'parameter'指定的数据（当'parameter'未被指定时，该输入可以省略）、输出是约束违反值。当且仅当约束违反值小于等于零时，约束被满足。
- 'lower'表示决策变量的下界，它仅在'encoding'的值为'real'时生效。
- 'upper'表示决策变量的上界，它仅在'encoding'的值为'real'时生效。
- 'initFcn'表示种群初始化函数，它的值必须是一个函数句柄。该函数必须有两个输入和一个输出，其中第一个输入是种群大小、第二个输入是由'parameter'指定的数据（当'parameter'未被指定时，该输入可以省略）、输出是种群的决策向量构成的矩阵。该函数通常在算法开始时被调用。
- 'decFcn'表示无效解修复函数，它的值必须是一个函数句柄。该函数必须有两个输入和一个输出，其中第一个输入是一个决策向量、第二个输入是由'parameter'指定的数据（当'parameter'未被指定时，该输入可以省略）、输出是修复后的决策向量。该函数会在计算目标函数前被调用。
- 'parameter'表示问题的数据，它作为函数'objFcn'、'conFcn'、'initFcn'和'decFcn'的第二个输入参数。

例如，以下代码利用差分进化算法求一个 10 变量的单峰函数的最小值：

```
platemo('objFcn',@(x)sum(x.^2),'lower',zeros(1,10)-10,
'upper',zeros(1,10)+10,'algorithm',@DE);
```

以下代码利用默认的算法求一个带旋转的 10 变量单峰函数的最小值，其中旋转矩阵通过'parameter'来指定：

```
platemo('objFcn',@(x,d)sum((x*d).^2),'lower',zeros(1,10)-
```

```
10,'upper',zeros(1,10)+10,'parameter',rand(10));
```

以下代码利用 NSGA-II 求一个带约束的 2 目标、20 变量的优化问题的最小值，其中种群大小为 50：

```
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},'conFcn',g1,'lower',zeros(1,20),'upper',ones(1,20),'algorithm',@NSGAII,'N',50);
```

## 3. 获取运行结果

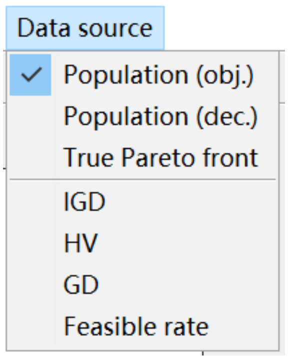
算法运行结束后得到的种群可以被显示在窗口中、保存在文件中或作为函数返回值。若按以下方式调用主函数：

```
[Dec,Obj,Con] = platemo(…);
```

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 的值为零时（默认情况），得到的种群会被显示在窗口中，用户可以利用窗口中的 Data source 菜单来选择要显示的内容。




当 Value 的值大于零时，得到的种群会被保存在命名为 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。每个文件存储一个单元数组 result 和一个结构体 metric。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。以上操作均由默认的输出函数@ALGORITHM.Output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。

```
result =
  6×2 cell array
    {[ 1650]}    {1×50 SOLUTION}
    {[ 3300]}    {1×50 SOLUTION}
    {[ 5000]}    {1×50 SOLUTION}
    {[ 6650]}    {1×50 SOLUTION}
    {[ 8300]}    {1×50 SOLUTION}
    {[10000]}    {1×50 SOLUTION}
```

```
metric =
  struct with fields:

    runtime: 0.3317
        IGD: [6×1 double]
```

此外，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metric中。若需要手动计算指标值，用户需获取问题的最优值并调用指标函数，例如

```
pro = DTLZ2();
IGD(result{end},pro.optimum);
```

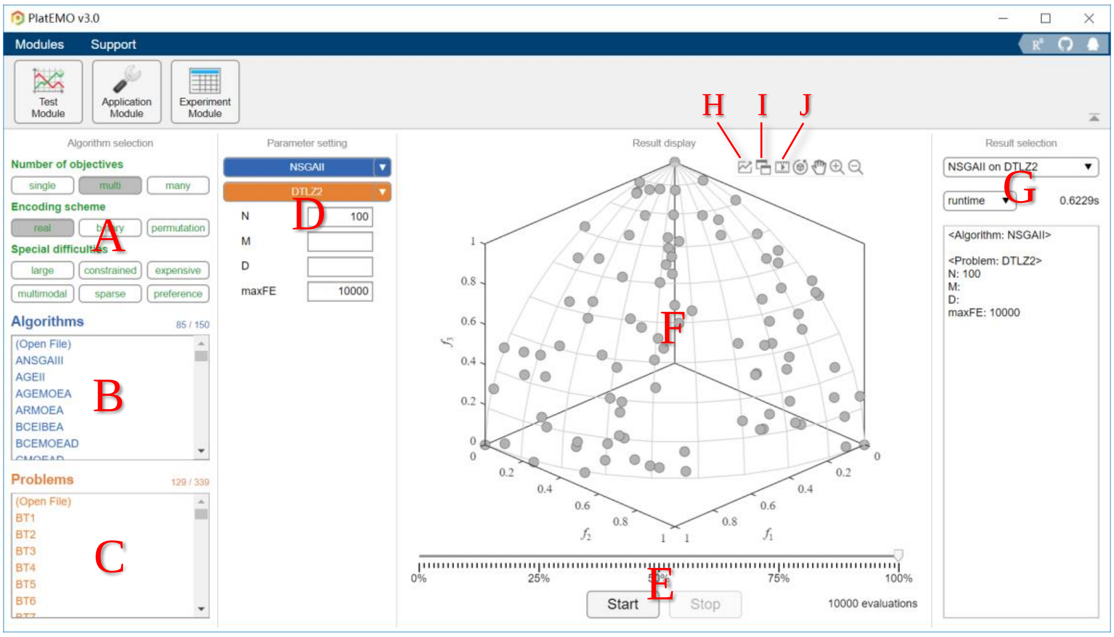
# 三 通过图形界面使用 PlatEMO

## 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。

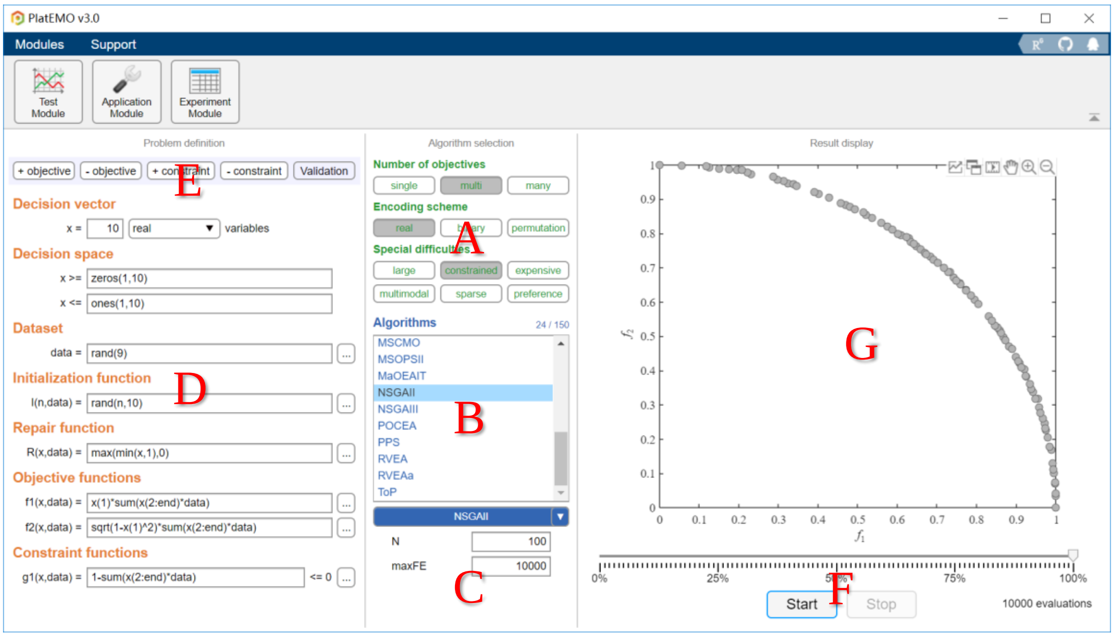



在该模块中，用户首先需要在区域 A 中选择问题类型（参阅算法和问题的标签章节），在区域 B 中选择一个算法，在区域 C 中选择一个测试问题，并在区域 D 中设定相关参数。之后，用户可以在区域 E 中控制算法的运行，在区域 F 中观察算法运行的实时结果，并在区域 G 中调取历史运行结果。

按钮 H 用于选择要显示的内容，按钮 I 用于将当前显示内容显示在一个新窗口中并存储至工作空间，按钮 J 用于将算法运行的动态过程保存为一个 20 帧的 GIF 图像。

## 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。



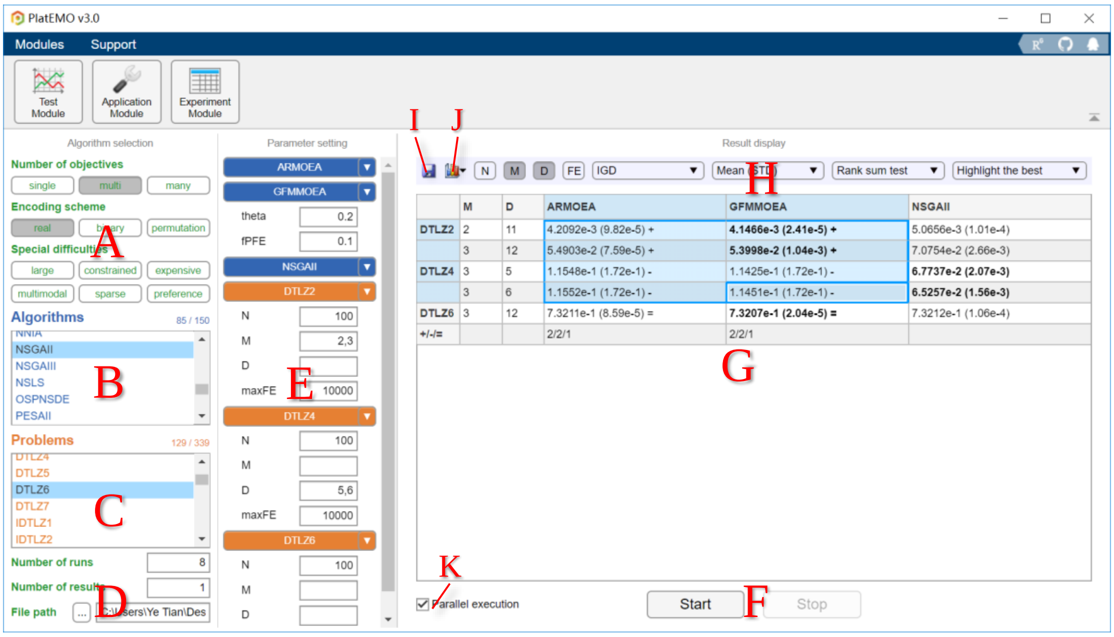

在该模块中，用户首先需要在区域 D 中定义问题，定义方式与求解自定义问题相同，其中

- Decision vector 对应 'encoding'
- Decision space 对应 'lower' 和 'upper'
- Dataset 对应 'parameter'
- Initialization function 对应 'initFcn'
- Repair function 对应 'decFcn'
- Objective functions 对应 'objFcn'
- Constraint functions 对应 'conFcn'

在简单情况下，用户可以仅定义 Decision vector、Decision space、Objective functions 和 Constraint functions。同时，用户可以在区域 E 中更改目标数量、约束数量和验证问题定义的合法性。之后，区域 A 中的问题类型可以被自动确定，用户需要在区域 B 中选择一个算法并在区域 C 中设定相关参数。最后，用户可以在区域 F 中控制算法的运行，并在区域 G 中观察算法运行的实时结果。

## 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。




在该模块中，用户首先需要在区域 A 中选择问题类型（参阅算法和问题的标签章节），在区域 B 中选择一个或多个算法，在区域 C 中选择一个或多个测试问题，在区域 D 中设定实验参数，并在区域 E 中设定相关参数；这里的目标数 M 和变量数 D 可以是向量。之后，用户可以在区域 F 中控制实验的运行，并在区域 G 中观察实验运行的实时结果。

需要在表格中显示的统计信息可以在区域 H 中选择。按钮 I 用于将当前表格保存为 Excel、TeX、TXT 或 MAT 文件，按钮 J 用于将所选单元格中的结果显示在一个新窗口中，按钮 K 用于选择实验在单处理器上运行（串行）或多处理器上运行（并行）。

所有结果将会被以 MAT 文件的形式保存在区域 D 中指定的文件夹中。如果存在同名的结果文件，该文件将会被读取以代替算法运行。

## 4. 算法和问题的标签

每个算法或测试问题需要被添加上标签，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
% <single> <real> <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：问题含有两或三个目标函数 |
| <many> | 超多目标优化：问题含有三个以上目标函数 |
| <real> | 连续优化：决策变量为实数或整数 |
| <binary> | 二进制优化：决策变量为二进制数 |
| <permutation> | 序列优化：决策变量构成一个序列 |
| <large> | 大规模优化：问题含有 100 个以上的决策变量 |
| <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 |
| <preference> | 偏好优化：仅需寻找前沿面上指定区域的最优解 |
| <none> | 空标签 |

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的问题；若标签集合为<single><real><constrained>，表示该算法只能求解带约束的问题；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。

每个算法和测试问题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。当用户在图形界面的区域 A 中选择问题类型后，所有可求解该类型问题的算法将会出现在图形界面的区域 B 中，且所有符合该类型的测试问题将会出现在图形界面的区域 C 中。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。

# 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 |
| 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 |
| 方法 | 是否可重定义 | 描述 |
| ALGORITHM | 不可 | 设定由用户指定的属性值 |
| Solve | 不可 | 通过调用 alg.Solve(pro) 来利用算法 alg 求解问题 pro |
| main | 必须 | 算法的主体部分 |
| NotTerminated | 不可 | main() 中每次迭代前调用的函数 |
| ParameterSet | 不可 | 根据 parameter 设定算法参数 |

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
% <single><real/binary/permutation><large/none><constrained/none>
% Genetic algorithm
% proC ---  1 --- Probability of crossover
% disC --- 20 --- Distribution index of crossover
% proM ---  1 --- Expectation of the number of mutated variables
% disM --- 20 --- Distribution index of mutation

%------------------------------- Reference --------------------------------
% J. H. Holland, Adaptation in Natural and Artificial Systems,
% MIT Press, 1992.
%--------------------------------------------------------------------------

```

```
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                  P1 = TournamentSelection(2,Pro.N,FitnessSingle(P));
20                  O = OperatorGA(P(P1),{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
```

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*算法和问题的标签*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评价次数是否超过最大值；若超过则终止算法；

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 | 算法每次迭代前调用的函数 |
| ALGORITHM. ParameterSet | 根据用户的输入设定算法参数 |
| PROBLEM. Initialization | 初始化一个种群 |
| CrowdingDistance | 计算解的拥挤距离（用于多目标优化） |
| FitnessSingle | 计算解的适应度（用于单目标优化） |
| NDSort | 非支配排序 |
| OperatorDE | 差分进化算子 |
| OperatorFEP | 进化规划算子 |
| OperatorGA | 遗传算子 |
| OperatorGAhalf | 遗传算子（仅产生前一半的子代） |
| OperatorPSO | 粒子群优化算子 |
| RouletteWheel Selection | 轮盘赌选择 |
| Tournament Selection | 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 | SOLUTION() | 当前运行中已消耗的评价次数 |
| encoding | Setting() | 问题的编码方式 |
| lower | Setting() | 决策变量的下界 |
| upper | Setting() | 决策变量的上界 |
| 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 |
| Setting | 必须 | 设定默认的属性值 |
| Initialization | 可以 | 初始化一个种群 |
| CalDec | 可以 | 修复种群中的无效解 |
| CalObj | 必须 | 计算种群中解的目标值；所有目标函数均为最小化问题 |
| CalCon | 可以 | 计算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 |
| GetOptimum | 可以 | 产生问题的最优值并保存在 optimum 中 |
| GetPF | 可以 | 产生问题的前沿面并保存在 PF 中 |
| DrawDec | 可以 | 显示一个种群的决策向量 |
| DrawObj | 可以 | 显示一个种群的目标向量 |
| Current | 不可 | 用来设定或获取当前 PROBLEM 对象的静态方法 |
| ParameterSet | 不可 | 根据 parameter 设定问题参数 |

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
% <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)
            obj.M = 1;
            if isempty(obj.D); obj.D = 30; end
            obj.lower = zeros(1,obj.D) - 100;
            obj.upper = zeros(1,obj.D) + 100;
            obj.encoding = 'real';
        end
        function PopObj = CalObj(obj,PopDec)
            PopObj = sum(PopDec.^2,2);
```

```
21            end
22        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（参阅算法和问题的标签章节）；

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：　设置问题的编码方式；

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 = SOLUTION(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：

```
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 MW1.m 添加了一个约束：

```
function PopCon = CalCon(obj,X)
    PopObj = obj.CalObj(X);
    l = sqrt(2)*PopObj(:,2) - sqrt(2)*PopObj(:,1);
    PopCon = sum(PopObj,2) - 1 - 0.5*sin(2*pi*l).^8;
end
```

利用 all(PopCon<=0,2)来确定每个解是否满足所有约束。注意等式约束需要被松弛为以上形式的不等式约束来处理。用户可以重定义方法 GetOptimum()来指定问题的最优值。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。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 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。以上方法的详细调用方式参阅代码中的注释。

## 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 | 用户 | 解的决策向量 |
| obj | SOLUTION() | 解的目标值 |
| con | SOLUTION() | 解的约束违反值 |
| add | adds() | 解的额外属性值（例如速度） |
| **方法** | **描述** | |
| SOLUTION | 接受一个或多个解的决策向量并计算相应的目标值与约束违反值；PROBLEM.FE 的值会自动加上生成的 SOLUTION 对象数目 | |
| decs | 获取多个解的决策向量构成的矩阵 | |
| objs | 获取多个解的目标值构成的矩阵 | |
| cons | 获取多个解的约束违反值构成的矩阵 | |

| | |
|---|---|
| adds | 获取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） |

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：

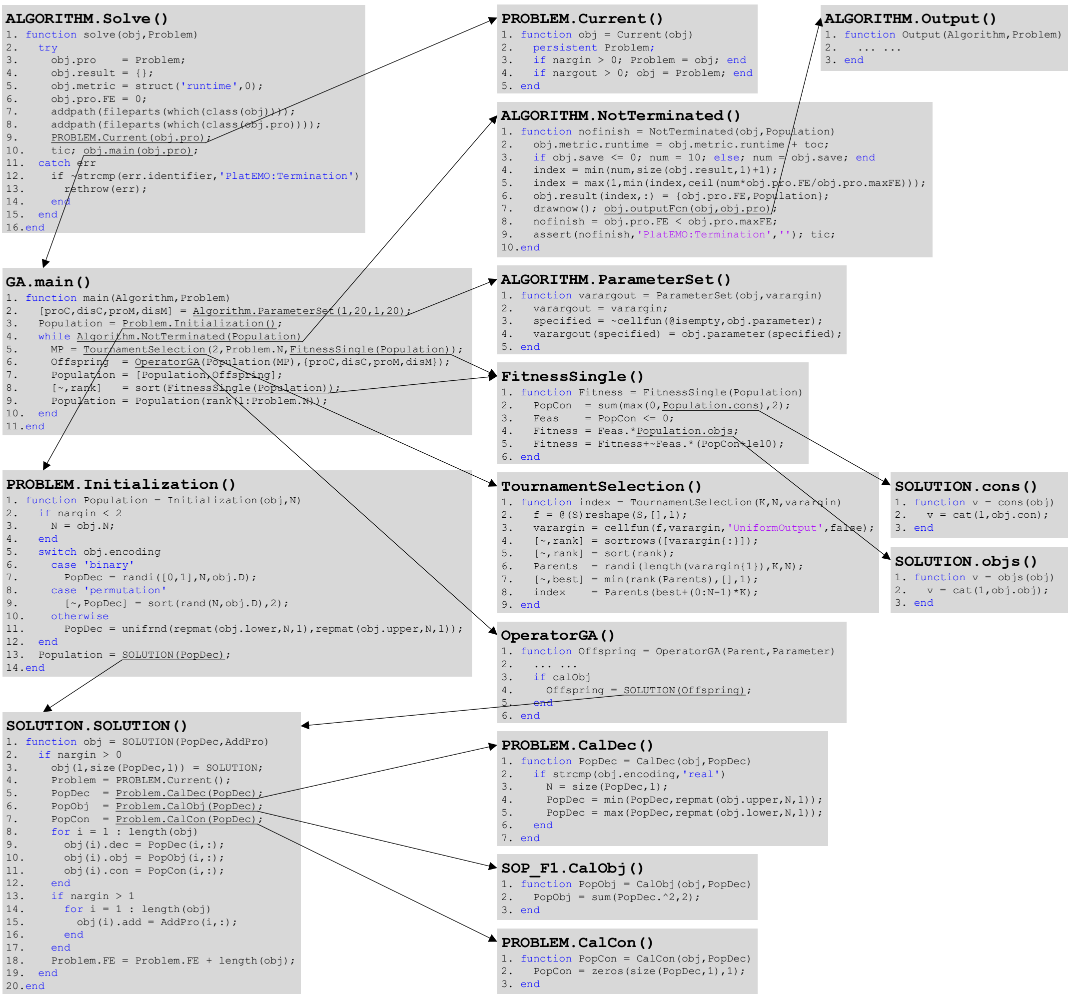
```
Population = SOLUTION(rand(10,5));
BestObjs = Population.best.objs
```

# 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：

```
1  function score = IGD(Population,optimum)
2  % <min>
3  % Inverted generational distance
4  
5  %------------------------------- Reference --------------------------------
6  % C. A. Coello Coello and N. C. Cortes, Solving multiobjective
7  % optimization problem using an artificial immune system, Genetic
8  % Programming and Evolvable Machines, 2005, 6(2): 163-190.
9  %--------------------------------------------------------------------------
10 
11     PopObj = Population.best.objs;
12     if size(PopObj,2) ~= size(optimum,2)
13       score = nan;
14     else
15       score = mean(min(pdist2(optimum,PopObj),[],2));
16     end
17 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，其中`<min>`表示指标值越小越好、`<max>`表示指标值越大越好；

第 3 行：　指标的全称；

第 5-9 行：　指标的参考文献；

第 11 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12-13行：若种群不存在可行解则返回 `nan`；

第14-15行：否则返回可行且非支配的解的指标值。

# 五 算法列表

| | 算法缩写 | 算法全称 | single | multi | many | real | binary | permutation | large | constrained | expensive | multimodal | sparse | preference |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | ABC | Artificial bee colony algorithm | √ | | | √ | | | √ | √ | | | | |
| 2 | AB-SAEA | Adaptive Bayesian based surrogate-assisted evolutionary algorithm | | √ | √ | √ | | | | | √ | | | |
| 3 | ACO | Ant colony optimization | √ | | | | | √ | √ | | | | | |
| 4 | AGE-II | Approximation-guided evolutionary multi-objective algorithm II | | √ | | √ | √ | √ | | | | | | |
| 5 | AGE-MOEA | Adaptive geometry estimation-based many-objective evolutionary algorithm | | √ | √ | √ | √ | √ | | √ | | | | |
| 6 | A-NSGA-III | Adaptive NSGA-III | | √ | √ | √ | √ | √ | | √ | | | | |
| 7 | AR-MOEA | Adaptive reference points based multi-objective evolutionary algorithm | | √ | √ | √ | √ | √ | | √ | | | | |
| 8 | BCE-IBEA | Bi-criterion evolution based IBEA | | √ | √ | √ | √ | √ | | | | | | |
| 9 | BCE-MOEA/D | Bi-criterion evolution based MOEA/D | | √ | √ | √ | √ | √ | | | | | | |
| 10 | BFGS | A quasi-Newton method proposed by Broyden, Fletcher, Goldfarb, and Shanno | √ | | | √ | | | √ | | | | | |
| 11 | BiGE | Bi-goal evolution | | | √ | √ | √ | √ | | | | | | |
| 12 | BSPGA | Binary space partition tree based genetic algorithm | √ | | | | √ | | √ | √ | | | | |
| 13 | CA-MOEA | Clustering based adaptive multi-objective evolutionary algorithm | | √ | | √ | √ | √ | | | | | | |
| 14 | CCGDE3 | Cooperative coevolution GDE3 | | √ | | √ | | | √ | | | | | |
| 15 | CCMO | Coevolutionary constrained multi-objective optimization framework | | √ | | √ | √ | √ | | √ | | | | |
| 16 | c-DPEA | Constrained dual-population evolutionary algorithm | | √ | | √ | √ | √ | | √ | | | | |
| 17 | CMA-ES | Covariance matrix adaptation evolution strategy | √ | | | √ | | | √ | √ | | | | |
| 18 | C-MOEA/D | Constraint-MOEA/D | | √ | √ | √ | √ | √ | | √ | | | | |
| 19 | CMOEA-MS | Constrained multiobjective evolutionary algorithm with multiple stages | | √ | | √ | √ | √ | | √ | | | | |
| 20 | CMOPSO | Competitive mechanism based multi-objective particle swarm optimizer | | √ | | √ | | | | | | | | |
| 21 | CPS-MOEA | Classification and Pareto domination based multi-objective evolutionary | | √ | | √ | | | | | √ | | | |
| 22 | CSEA | Classification based surrogate-assisted evolutionary algorithm | | √ | √ | √ | | | | | √ | | | |
| 23 | CSO | Competitive swarm optimizer | √ | | | √ | | | √ | √ | | | | |
| 24 | C-TAEA | Two-archive evolutionary algorithm for constrained MOPs | | √ | √ | √ | √ | √ | | √ | | | | |
| 25 | DAEA | Duplication analysis based evolutionary algorithm | | √ | | | √ | | | | | | | |
| 26 | DCNSGA-III | Dynamic constrained NSGA-III | | √ | √ | √ | √ | √ | | √ | | | | |
| 27 | DE | Differential evolution | √ | | | √ | | | √ | √ | | | | |
| 28 | DEA-GNG | Decomposition based evolutionary algorithm guided by growing neural gas | | √ | √ | √ | √ | √ | | | | | | |

| | 算法缩写 | 算法全称 | single | multi | many | real | binary | permutation | large | constrained | expensive | multimodal | sparse | preference |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 29 | DGEA | Direction guided evolutionary algorithm | | √ | √ | √ | | | √ | | | | | |
| 30 | DMOEA-eC | Decomposition-based multi-objective evolutionary algorithm with the e-constraint framework | | √ | | √ | √ | √ | | | | | | |
| 31 | dMOPSO | MOPSO based on decomposition | | √ | | √ | | | | | | | | |
| 32 | DN-NSGA-II | Decision space based niching NSGA-II | | √ | | √ | | | | | | √ | | |
| 33 | DWU | Dominance-weighted uniformity multi-objective evolutionary algorithm | | √ | | √ | √ | √ | | | | | | |
| 34 | EAG-MOEA/D | External archive guided MOEA/D | | √ | | √ | √ | √ | | | | | | |
| 35 | EDN-ARMOEA | Efficient dropout neural network based AR-MOEA | | √ | √ | √ | | | | | √ | | | |
| 36 | EFR-RR | Ensemble fitness ranking with a ranking restriction scheme | | √ | √ | √ | √ | √ | | | | | | |
| 37 | EGO | Efficient global optimization | √ | | | √ | | | | | √ | | | |
| 38 | EIM-EGO | Expected improvement matrix based efficient global optimization | | √ | | √ | | | | | √ | | | |
| 39 | e-MOEA | Epsilon multi-objective evolutionary algorithm | | √ | √ | √ | √ | √ | | | | | | |
| 40 | EMyO/C | Evolutionary many-objective optimization algorithm with clustering-based | | √ | √ | √ | | | | | | | | |
| 41 | ENS-MOEA/D | Ensemble of different neighborhood sizes based MOEA/D | | √ | √ | √ | | | | | | | | |
| 42 | FEP | Fast evolutionary programming | √ | | | √ | | | √ | √ | | | | |
| 43 | FRCG | Fletcher-Reeves conjugate gradient | √ | | | √ | | | √ | | | | | |
| 44 | FROFI | Feasibility rule with the incorporation of objective function information | √ | | | √ | | | √ | √ | | | | |
| 45 | GA | Genetic algorithm | √ | | | √ | √ | √ | √ | √ | | | | |
| 46 | GDE3 | Generalized differential evolution 3 | | √ | | √ | | | | √ | | | | |
| 47 | GFM-MOEA | Generic front modeling based multi-objective evolutionary algorithm | | √ | √ | √ | √ | √ | | | | | | |
| 48 | GLMO | Grouped and linked mutation operator algorithm | | √ | | √ | | | √ | | | | | |
| 49 | g-NSGA-II | g-dominance based NSGA-II | | √ | | √ | √ | √ | | | | | | √ |
| 50 | GrEA | Grid-based evolutionary algorithm | | | √ | √ | √ | √ | | | | | | |
| 51 | HeE-MOEA | Multiobjective evolutionary algorithm with heterogeneous ensemble based infill criterion | | √ | | √ | | | | | √ | | | |
| 52 | hpaEA | Hyperplane assisted evolutionary algorithm | | √ | √ | √ | √ | √ | | | | | | |
| 53 | HypE | Hypervolume estimation algorithm | | √ | √ | √ | √ | √ | | | | | | |
| 54 | IBEA | Indicator-based evolutionary algorithm | | √ | √ | √ | √ | √ | | | | | | |
| 55 | ICMA | Indicator based constrained multi-objective algorithm | | √ | | √ | | | | √ | | | | |
| 56 | I-DBEA | Improved decomposition-based evolutionary algorithm | | √ | √ | √ | √ | √ | | √ | | | | |
| 57 | IM-MOEA | Inverse modeling based multiobjective evolutionary algorithm | | √ | | √ | | | √ | | | | | |
| 58 | IMODE | Improved multi-operator differential evolution | √ | | | √ | | | √ | √ | | | | |
| 59 | I-SIBEA | Interactive simple indicator-based evolutionary algorithm | | √ | | √ | √ | √ | | | | | | √ |
| 60 | KnEA | Knee point driven evolutionary algorithm | | | √ | √ | √ | √ | | √ | | | | |
| 61 | K-RVEA | Surrogate-assisted RVEA | | √ | √ | √ | | | | | √ | | | |
| 62 | KTA2 | Kriging-assisted Two_Arch2 | | √ | √ | √ | | | | | √ | | | |

| | 算法缩写 | 算法全称 | single | multi | many | real | binary | permutation | large | constrained | expensive | multimodal | sparse | preference |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 63 | LCSA | Linear combination-based search algorithm | | √ | √ | √ | | | √ | | | | | |
| 64 | LMEA | Evolutionary algorithm for large-scale many-objective optimization | | √ | √ | √ | | | √ | | | | | |
| 65 | LMOCSO | Large-scale multi-objective competitive swarm optimization algorithm | | √ | √ | √ | | | √ | √ | | | | |
| 66 | LSMOF | Large-scale multi-objective optimization framework with NSGA-II | | √ | | √ | | | √ | | | | | |
| 67 | MaOEA-CSS | Many-objective evolutionary algorithms based on coordinated selection | | √ | √ | √ | √ | √ | | | | | | |
| 68 | MaOEA-DDFC | Many-objective evolutionary algorithm based on directional diversity and favorable convergence | | √ | √ | √ | √ | √ | | | | | | |
| 69 | MaOEA/IGD | IGD based many-objective evolutionary algorithm | | | √ | √ | √ | √ | | | | | | |
| 70 | MaOEA/IT | Many-objective evolutionary algorithms based on an independent two-stage | | √ | √ | √ | | | | √ | | | | |
| 71 | MaOEA-R&D | Many-objective evolutionary algorithm based on objective space reduction | | | √ | √ | √ | √ | | | | | | |
| 72 | MMOPSO | MOPSO with multiple search strategies | | √ | | √ | | | | | | | | |
| 73 | MO_Ring_ PSO_SCD | Multiobjective PSO using ring topology and special crowding distance | | √ | | √ | | | | | | √ | | |
| 74 | MOCell | Cellular genetic algorithm | | √ | | √ | √ | √ | | √ | | | | |
| 75 | MO-CMA | Multi-objective covariance matrix adaptation evolution strategy | | √ | | √ | | | | | | | | |
| 76 | MOEA/D | Multiobjective evolutionary algorithm based on decomposition | | √ | √ | √ | √ | √ | | | | | | |
| 77 | MOEA/D-AWA | MOEA/D with covariance matrix adaptation evolution strategy | | √ | √ | √ | | | | | | | | |
| 78 | MOEA/D-CMA | MOEA/D with covariance matrix adaptation evolution strategy | | √ | √ | √ | | | | | | | | |
| 79 | MOEA/DD | Many-objective evolutionary algorithm based on dominance and decomposition | | √ | √ | √ | √ | √ | | √ | | | | |
| 80 | MOEA/D-DAE | MOEA/D with detect-and-escape strategy | | √ | | √ | √ | √ | | √ | | | | |
| 81 | MOEA/D-DE | MOEA/D based on differential evolution | | √ | √ | √ | | | | | | | | |
| 82 | MOEA/D-DRA | MOEA/D with dynamical resource allocation | | √ | √ | √ | | | | | | | | |
| 83 | MOEA/D-DU | MOEA/D with a distance based updating strategy | | √ | √ | √ | √ | √ | | | | | | |
| 84 | MOEA/D-EGO | MOEA/D with efficient global optimization | | √ | | √ | | | | | √ | | | |
| 85 | MOEA/D-FRRMAB | MOEA/D with fitness-rate-rank-based multiarmed bandit | | √ | √ | √ | | | | | | | | |
| 86 | MOEA/D-M2M | MOEA/D based on MOP to MOP | | √ | | √ | | | | | | | | |
| 87 | MOEA/D-MRDL | MOEA/D with maximum relative diversity loss | | √ | | √ | | | | | | | | |
| 88 | MOEA/D-PaS | MOEA/D with Pareto adaptive scalarizing approximation | | √ | √ | √ | | | | | | | | |
| 89 | MOEA/D-STM | MOEA/D with stable matching | | √ | √ | √ | | | | | | | | |
| 90 | MOEA/D-URAW | MOEA/D with uniform randomly adaptive weights | | √ | √ | √ | √ | √ | | | | | | |
| 91 | MOEA/DVA | Multi-objective evolutionary algorithm based on decision variable | | √ | | √ | | | √ | | | | | |

| | 算法缩写 | 算法全称 | single | multi | many | real | binary | permutation | large | constrained | expensive | multimodal | sparse | preference |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 92 | MOEA/IGD-NS | Multi-objective evolutionary algorithm based on an enhanced IGD | | √ | | √ | √ | √ | | | | | | |
| 93 | MOEA-PC | Multiobjective evolutionary algorithm based on polar coordinates | | √ | | √ | | | | | | | | |
| 94 | MOEA/PSL | Multi-objective evolutionary algorithm based on Pareto optimal subspace | | √ | | √ | √ | | √ | √ | | | √ | |
| 95 | MOMBI-II | Many objective metaheuristic based on the R2 indicator II | | √ | √ | √ | √ | √ | | | | | | |
| 96 | MOPSO | Multi-objective particle swarm optimization | | √ | | √ | | | | | | | | |
| 97 | MOPSO-CD | MOPSO with crowding distance | | √ | | √ | | | | | | | | |
| 98 | M-PAES | Memetic algorithm with Pareto archived evolution strategy | | √ | | √ | | | | | | | | |
| 99 | MP-MMEA | Multi-population multi-modal multi-objective evolutionary algorithm | | √ | | √ | | | √ | | | √ | √ | |
| 100 | MPSO/D | Multi-objective particle swarm optimization algorithm based on decomposition | | √ | √ | √ | | | | | | | | |
| 101 | MSCMO | Multi-stage constrained multi-objective evolutionary algorithm | | √ | | √ | √ | √ | | √ | | | | |
| 102 | MSEA | Multi-stage multi-objective evolutionary algorithm | | √ | | √ | √ | √ | | | | | | |
| 103 | MSOPS-II | Multiple single objective Pareto sampling II | | √ | √ | √ | | | | √ | | | | |
| 104 | MTS | Multiple trajectory search | | √ | | √ | | | | | | | | |
| 105 | MultiObjective EGO | Multi-objective efficient global optimization | | √ | | √ | | | | √ | √ | | | |
| 106 | MyO-DEMR | Many-objective differential evolution with mutation restriction | | √ | √ | √ | | | | | | | | |
| 107 | NMPSO | Novel multi-objective particle swarm optimization | | √ | √ | √ | | | | | | | | |
| 108 | NNIA | Nondominated neighbor immune algorithm | | √ | | √ | √ | √ | | | | | | |
| 109 | NSGA-II | Nondominated sorting genetic algorithm II | | √ | | √ | √ | √ | | √ | | | | |
| 110 | NSGA-II+ARSBX | NSGA-II with adaptive rotation based simulated binary crossover | | √ | | √ | | | | √ | | | | |
| 111 | NSGA-II-conflict | NSGA-II with conflict-based partitioning strategy | | | √ | √ | √ | √ | | | | | | |
| 112 | NSGA-III | Nondominated sorting genetic algorithm III | | √ | √ | √ | √ | √ | | √ | | | | |
| 113 | NSGA-II/SDR | NSGA-II with strengthened dominance relation | | | √ | √ | √ | √ | | | | | | |
| 114 | NSLS | Multiobjective optimization framework based on nondominated sorting and local search | | √ | | √ | | | | | | | | |
| 115 | OFA | Optimal foraging algorithm | √ | | | √ | | | √ | √ | | | | |
| 116 | one-by-one EA | Many-objective evolutionary algorithm using a one-by-one selection | | √ | √ | √ | √ | √ | | | | | | |
| 117 | OSP-NSDE | Non-dominated sorting differential evolution with prediction in the objective space | | √ | | √ | | | | | | | | |
| 118 | ParEGO | Efficient global optimization for Pareto optimization | | √ | | √ | | | | | √ | | | |
| 119 | PeEA | Pareto front shape estimation based evolutionary algorithm | | √ | √ | √ | √ | √ | | | | | | |
| 120 | PESA-II | Pareto envelope-based selection algorithm II | | √ | | √ | √ | √ | | | | | | |
| 121 | PICEA-g | Preference-inspired coevolutionary algorithm with goals | | √ | √ | √ | √ | √ | | | | | | |
| 122 | PM-MOEA | Pattern mining based multi-objective evolutionary algorithm | | √ | | √ | | | √ | √ | | | | |
| 123 | POCEA | Paired offspring generation based constrained evolutionary algorithm | | √ | | √ | | | √ | √ | | | | |

| | 算法缩写 | 算法全称 | single | multi | many | real | binary | permutation | large | constrained | expensive | multimodal | sparse | preference |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 124 | PPS | Push and pull search algorithm | | √ | √ | √ | | | | √ | | | | |
| 125 | PREA | Promising-region based EMO algorithm | | √ | √ | √ | √ | √ | | | | | | |
| 126 | PSO | Particle swarm optimization | √ | | | √ | | | √ | √ | | | | |
| 127 | RM-MEDA | Regularity model-based multiobjective estimation of distribution | | √ | | √ | | | | | | | | |
| 128 | r-NSGA-II | r-dominance based NSGA-II | | √ | | √ | √ | √ | | | | | | √ |
| 129 | RPD-NSGA-II | Reference point dominance-based NSGA-II | | √ | √ | √ | √ | √ | | | | | | |
| 130 | RPEA | Reference points-based evolutionary algorithm | | | √ | √ | √ | √ | | | | | | |
| 131 | RSEA | Radial space division based evolutionary algorithm | | √ | √ | √ | √ | √ | | | | | | |
| 132 | RVEA | Reference vector guided evolutionary algorithm | | √ | √ | √ | √ | √ | | √ | | | | |
| 133 | RVEAa | RVEA embedded with the reference vector regeneration strategy | | | √ | √ | √ | √ | | | | | | |
| 134 | RVEA-iGNG | RVEA based on improved growing neural gas | | √ | √ | √ | √ | √ | | | | | | |
| 135 | S3-CMA-ES | Scalable small subpopulations based covariance matrix adaptation | | √ | √ | √ | | | √ | | | | | |
| 136 | SA | Simulated annealing | √ | | | √ | | | √ | √ | | | | |
| 137 | SACC-EAM-II | Surrogate-assisted cooperative co-evolutionary algorithm of Minamo | √ | | | √ | | | | | √ | | | |
| 138 | SACOSO | Surrogate-assisted cooperative swarm optimization | √ | | | √ | | | √ | | √ | | | |
| 139 | SADE-Sammon | Sammon mapping assisted differential evolution | √ | | | √ | | | | | √ | | | |
| 140 | SAMSO | Multiswarm-assisted expensive optimization | √ | | | √ | | | √ | | √ | | | |
| 141 | S-CDAS | Self-controlling dominance area of solutions | | | √ | √ | √ | √ | | | | | | |
| 142 | SHADE | Success-history based adaptive differential evolution | √ | | | √ | | | √ | √ | | | | |
| 143 | SIBEA | Simple indicator-based evolutionary algorithm | | √ | | √ | √ | √ | | | | | | |
| 144 | SIBEA-kEMOSS | SIBEA with minimum objective subset of size k with minimum error | | | √ | √ | √ | √ | | | | | | |
| 145 | SMEA | Self-organizing multiobjective evolutionary algorithm | | √ | | √ | | | | | | | | |
| 146 | SMPSO | Speed-constrained multi-objective particle swarm optimization | | √ | | √ | | | | | | | | |
| 147 | SMS-EGO | S metric selection based efficient global optimization | | √ | | √ | | | | | √ | | | |
| 148 | SMS-EMOA | S metric selection based evolutionary multiobjective optimization | | √ | | √ | √ | √ | | | | | | |
| 149 | SparseEA | Evolutionary algorithm for sparse multi-objective optimization problems | | √ | | √ | √ | | √ | √ | | | √ | |
| 150 | SPEA2 | Strength Pareto evolutionary algorithm 2 | | √ | | √ | √ | √ | | | | | | |
| 151 | SPEA2+SDE | SPEA2 with shift-based density estimation | | | √ | √ | √ | √ | | | | | | |
| 152 | SPEA/R | Strength Pareto evolutionary algorithm based on reference direction | | √ | √ | √ | √ | √ | | | | | | |
| 153 | SQP | Sequential quadratic programming | √ | | | √ | | | √ | √ | | | | |
| 154 | SRA | Stochastic ranking algorithm | | | √ | √ | √ | √ | | | | | | |
| 155 | t-DEA | theta-dominance based evolutionary algorithm | | √ | √ | √ | √ | √ | | | | | | |
| 156 | TiGE-2 | Tri-Goal Evolution Framework for CMaOPs | | | √ | √ | √ | √ | | √ | | | | |
| 157 | ToP | Two-phase framework with NSGA-II | | √ | | √ | | | | √ | | | | |
| 158 | TriMOEA- | Multi-modal MOEA using two-archive and | | √ | | √ | | | | | | √ | | |

| | 算法缩写 | 算法全称 | single | multi | many | real | binary | permutation | large | constrained | expensive | multimodal | sparse | preference |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | TA&R | recombination strategies | | | | | | | | | | | | |
| 159 | Two_Arch2 | Two-archive algorithm 2 | | √ | √ | √ | √ | √ | | | | | | |
| 160 | VaEA | Vector angle based evolutionary algorithm | | √ | √ | √ | √ | √ | | | | | | |
| 161 | WOF | Weighted optimization framework | | √ | | √ | | | √ | | | | | |
| 162 | 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 | binary | permutation | large | constrained | expensive | multimodal | sparse | preference |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | BT1 | Benchmark MOP with bias feature | | √ | | √ | | | √ | | | | | |
| 2 | BT2 | Benchmark MOP with bias feature | | √ | | √ | | | √ | | | | | |
| 3 | BT3 | Benchmark MOP with bias feature | | √ | | √ | | | √ | | | | | |
| 4 | BT4 | Benchmark MOP with bias feature | | √ | | √ | | | √ | | | | | |
| 5 | BT5 | Benchmark MOP with bias feature | | √ | | √ | | | √ | | | | | |
| 6 | BT6 | Benchmark MOP with bias feature | | √ | | √ | | | √ | | | | | |
| 7 | BT7 | Benchmark MOP with bias feature | | √ | | √ | | | √ | | | | | |
| 8 | BT8 | Benchmark MOP with bias feature | | √ | | √ | | | √ | | | | | |
| 9 | BT9 | Benchmark MOP with bias feature | | √ | | √ | | | √ | | | | | |
| 10 | CEC2008_F1 | Shifted sphere function | √ | | | √ | | | √ | | √ | | | |
| 11 | CEC2008_F2 | Shifted Schwefel's function | √ | | | √ | | | √ | | √ | | | |
| 12 | CEC2008_F3 | Shifted Rosenbrock's function | √ | | | √ | | | √ | | √ | | | |
| 13 | CEC2008_F4 | Shifted Rastrign's function | √ | | | √ | | | √ | | √ | | | |
| 14 | CEC2008_F5 | Shifted Griewank's function | √ | | | √ | | | √ | | √ | | | |
| 15 | CEC2008_F6 | Shifted Ackley's function | √ | | | √ | | | √ | | √ | | | |
| 16 | CEC2008_F7 | FastFractal 'DoubleDip' function | √ | | | √ | | | √ | | √ | | | |
| 17 | CEC2010_F1 | CEC'2010 constrained optimization benchmark problem | √ | | | √ | | | | √ | | | | |
| 18 | CEC2010_F2 | CEC'2010 constrained optimization benchmark problem | √ | | | √ | | | | √ | | | | |
| 19 | CEC2010_F3 | CEC'2010 constrained optimization benchmark problem | √ | | | √ | | | | √ | | | | |
| 20 | CEC2010_F4 | CEC'2010 constrained optimization benchmark problem | √ | | | √ | | | | √ | | | | |
| 21 | CEC2010_F5 | CEC'2010 constrained optimization benchmark problem | √ | | | √ | | | | √ | | | | |
| 22 | CEC2010_F6 | CEC'2010 constrained optimization benchmark problem | √ | | | √ | | | | √ | | | | |
| 23 | CEC2010_F7 | CEC'2010 constrained optimization benchmark problem | √ | | | √ | | | | √ | | | | |
| 24 | CEC2010_F8 | CEC'2010 constrained optimization benchmark problem | √ | | | √ | | | | √ | | | | |
| 25 | CEC2010_F9 | CEC'2010 constrained optimization benchmark problem | √ | | | √ | | | | √ | | | | |
| 26 | CEC2010_F10 | CEC'2010 constrained optimization benchmark problem | √ | | | √ | | | | √ | | | | |
| 27 | CEC2010_F11 | CEC'2010 constrained optimization benchmark problem | √ | | | √ | | | | √ | | | | |
| 28 | CEC2010_F12 | CEC'2010 constrained optimization benchmark problem | √ | | | √ | | | | √ | | | | |
| 29 | CEC2010_F13 | CEC'2010 constrained optimization benchmark problem | √ | | | √ | | | | √ | | | | |
| 30 | CEC2010_F14 | CEC'2010 constrained optimization benchmark problem | √ | | | √ | | | | √ | | | | |
| 31 | CEC2010_F15 | CEC'2010 constrained optimization benchmark problem | √ | | | √ | | | | √ | | | | |
| 32 | CEC2010_F16 | CEC'2010 constrained optimization benchmark problem | √ | | | √ | | | | √ | | | | |
| 33 | CEC2010_F17 | CEC'2010 constrained optimization benchmark problem | √ | | | √ | | | | √ | | | | |

| | 问题缩写 | 问题全称 | single | multi | many | real | binary | permutation | large | constrained | expensive | multimodal | sparse | preference |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 34 | CEC2010_F18 | CEC'2010 constrained optimization benchmark problem | √ | | | √ | | | | √ | | | | |
| 35 | CEC2013_F1 | Shifted elliptic function | √ | | | √ | | | √ | | | | | |
| 36 | CEC2013_F2 | Shifted Rastrigin's function | √ | | | √ | | | √ | | | | | |
| 37 | CEC2013_F3 | Shifted Ackley's function | √ | | | √ | | | √ | | | | | |
| 38 | CEC2013_F4 | 7-nonseparable, 1-separable shifted and rotated elliptic function | √ | | | √ | | | √ | | | | | |
| 39 | CEC2013_F5 | 7-nonseparable, 1-separable shifted and rotated Rastrigin's function | √ | | | √ | | | √ | | | | | |
| 40 | CEC2013_F6 | 7-nonseparable, 1-separable shifted and rotated Ackley's function | √ | | | √ | | | √ | | | | | |
| 41 | CEC2013_F7 | 7-nonseparable, 1-separable shifted and rotated Schwefel's function | √ | | | √ | | | √ | | | | | |
| 42 | CEC2013_F8 | 20-nonseparable shifted and rotated elliptic function | √ | | | √ | | | √ | | | | | |
| 43 | CEC2013_F9 | 20-nonseparable shifted and rotated Rastrigin's function | √ | | | √ | | | √ | | | | | |
| 44 | CEC2013_F10 | 20-nonseparable shifted and rotated Rastrigin's function | √ | | | √ | | | √ | | | | | |
| 45 | CEC2013_F11 | 20-nonseparable shifted and rotated Schwefel's function | √ | | | √ | | | √ | | | | | |
| 46 | CEC2013_F12 | Shifted Rosenbrock's function | √ | | | √ | | | √ | | | | | |
| 47 | CEC2013_F13 | Shifted Schwefel's function with conforming overlapping subcomponents | √ | | | √ | | | √ | | | | | |
| 48 | CEC2013_F14 | Shifted Schwefel's function with conflicting overlapping subcomponents | √ | | | √ | | | √ | | | | | |
| 49 | CEC2013_F15 | Shifted Schwefel's function | √ | | | √ | | | √ | | | | | |
| 50 | CEC2017_F1 | CEC'2017 constrained optimization benchmark problem | √ | | | √ | | | | √ | | | | |
| 51 | CEC2017_F2 | CEC'2017 constrained optimization benchmark problem | √ | | | √ | | | | √ | | | | |
| 52 | CEC2017_F3 | CEC'2017 constrained optimization benchmark problem | √ | | | √ | | | | √ | | | | |
| 53 | CEC2017_F4 | CEC'2017 constrained optimization benchmark problem | √ | | | √ | | | | √ | | | | |
| 54 | CEC2017_F5 | CEC'2017 constrained optimization benchmark problem | √ | | | √ | | | | √ | | | | |
| 55 | CEC2017_F6 | CEC'2017 constrained optimization benchmark problem | √ | | | √ | | | | √ | | | | |
| 56 | CEC2017_F7 | CEC'2017 constrained optimization benchmark problem | √ | | | √ | | | | √ | | | | |
| 57 | CEC2017_F8 | CEC'2017 constrained optimization benchmark problem | √ | | | √ | | | | √ | | | | |
| 58 | CEC2017_F9 | CEC'2017 constrained optimization benchmark problem | √ | | | √ | | | | √ | | | | |
| 59 | CEC2017_F10 | CEC'2017 constrained optimization benchmark problem | √ | | | √ | | | | √ | | | | |
| 60 | CEC2017_F11 | CEC'2017 constrained optimization benchmark problem | √ | | | √ | | | | √ | | | | |
| 61 | CEC2017_F12 | CEC'2017 constrained optimization benchmark problem | √ | | | √ | | | | √ | | | | |
| 62 | CEC2017_F13 | CEC'2017 constrained optimization benchmark problem | √ | | | √ | | | | √ | | | | |
| 63 | CEC2017_F14 | CEC'2017 constrained optimization benchmark problem | √ | | | √ | | | | √ | | | | |
| 64 | CEC2017_F15 | CEC'2017 constrained optimization benchmark problem | √ | | | √ | | | | √ | | | | |
| 65 | CEC2017_F16 | CEC'2017 constrained optimization benchmark problem | √ | | | √ | | | | √ | | | | |
| 66 | CEC2017_F17 | CEC'2017 constrained optimization benchmark problem | √ | | | √ | | | | √ | | | | |
| 67 | CEC2017_F18 | CEC'2017 constrained optimization benchmark problem | √ | | | √ | | | | √ | | | | |
| 68 | CEC2017_F19 | CEC'2017 constrained optimization benchmark problem | √ | | | √ | | | | √ | | | | |
| 69 | CEC2017_F20 | CEC'2017 constrained optimization benchmark problem | √ | | | √ | | | | √ | | | | |

| | 问题缩写 | 问题全称 | single | multi | many | real | binary | permutation | large | constrained | expensive | multimodal | sparse | preference |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 70 | CEC2017_F21 | CEC'2017 constrained optimization benchmark problem | √ | | | √ | | | | √ | | | | |
| 71 | CEC2017_F22 | CEC'2017 constrained optimization benchmark problem | √ | | | √ | | | | √ | | | | |
| 72 | CEC2017_F23 | CEC'2017 constrained optimization benchmark problem | √ | | | √ | | | | √ | | | | |
| 73 | CEC2017_F24 | CEC'2017 constrained optimization benchmark problem | √ | | | √ | | | | √ | | | | |
| 74 | CEC2017_F25 | CEC'2017 constrained optimization benchmark problem | √ | | | √ | | | | √ | | | | |
| 75 | CEC2017_F26 | CEC'2017 constrained optimization benchmark problem | √ | | | √ | | | | √ | | | | |
| 76 | CEC2017_F27 | CEC'2017 constrained optimization benchmark problem | √ | | | √ | | | | √ | | | | |
| 77 | CEC2017_F28 | CEC'2017 constrained optimization benchmark problem | √ | | | √ | | | | √ | | | | |
| 78 | CEC2020_F1 | Bent cigar function | √ | | | √ | | | | | | | | |
| 79 | CEC2020_F2 | Shifted and rotated Schwefel's function | √ | | | √ | | | | | | | | |
| 80 | CEC2020_F3 | Shifted and rotated Lunacek bi-Rastrigin function | √ | | | √ | | | | | | | | |
| 81 | CEC2020_F4 | Expanded Rosenbrock's plus Griewangk's function | √ | | | √ | | | | | | | | |
| 82 | CEC2020_F5 | Hybrid function 1 | √ | | | √ | | | | | | | | |
| 83 | CEC2020_F6 | Hybrid function 2 | √ | | | √ | | | | | | | | |
| 84 | CEC2020_F7 | Hybrid function 3 | √ | | | √ | | | | | | | | |
| 85 | CEC2020_F8 | Composition function 1 | √ | | | √ | | | | | | | | |
| 86 | CEC2020_F9 | Composition function 2 | √ | | | √ | | | | | | | | |
| 87 | CEC2020_F10 | Composition function 3 | √ | | | √ | | | | | | | | |
| 88 | CF1 | Constrained benchmark MOP | | √ | | √ | | | √ | √ | | | | |
| 89 | CF2 | Constrained benchmark MOP | | √ | | √ | | | √ | √ | | | | |
| 90 | CF3 | Constrained benchmark MOP | | √ | | √ | | | √ | √ | | | | |
| 91 | CF4 | Constrained benchmark MOP | | √ | | √ | | | √ | √ | | | | |
| 92 | CF5 | Constrained benchmark MOP | | √ | | √ | | | √ | √ | | | | |
| 93 | CF6 | Constrained benchmark MOP | | √ | | √ | | | √ | √ | | | | |
| 94 | CF7 | Constrained benchmark MOP | | √ | | √ | | | √ | √ | | | | |
| 95 | CF8 | Constrained benchmark MOP | | √ | | √ | | | √ | √ | | | | |
| 96 | CF9 | Constrained benchmark MOP | | √ | | √ | | | √ | √ | | | | |
| 97 | CF10 | Constrained benchmark MOP | | √ | | √ | | | √ | √ | | | | |
| 98 | DAS-CMOP1 | Difficulty-adjustable and scalable constrained benchmark MOP | | √ | | √ | | | √ | √ | | | | |
| 99 | DAS-CMOP2 | Difficulty-adjustable and scalable constrained benchmark MOP | | √ | | √ | | | √ | √ | | | | |
| 100 | DAS-CMOP3 | Difficulty-adjustable and scalable constrained benchmark MOP | | √ | | √ | | | √ | √ | | | | |
| 101 | DAS-CMOP4 | Difficulty-adjustable and scalable constrained benchmark MOP | | √ | | √ | | | √ | √ | | | | |
| 102 | DAS-CMOP5 | Difficulty-adjustable and scalable constrained benchmark MOP | | √ | | √ | | | √ | √ | | | | |
| 103 | DAS-CMOP6 | Difficulty-adjustable and scalable constrained benchmark MOP | | √ | | √ | | | √ | √ | | | | |
| 104 | DAS-CMOP7 | Difficulty-adjustable and scalable constrained benchmark MOP | | √ | | √ | | | √ | √ | | | | |
| 105 | DAS-CMOP8 | Difficulty-adjustable and scalable constrained benchmark MOP | | √ | | √ | | | √ | √ | | | | |
| 106 | DAS-CMOP9 | Difficulty-adjustable and scalable constrained benchmark MOP | | √ | | √ | | | √ | √ | | | | |

| | 问题缩写 | 问题全称 | single | multi | many | real | binary | permutation | large | constrained | expensive | multimodal | sparse | preference |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 107 | DOC1 | Benchmark MOP with constraints in decision and objective spaces | | √ | | √ | | | | √ | | | | |
| 108 | DOC2 | Benchmark MOP with constraints in decision and objective spaces | | √ | | √ | | | | √ | | | | |
| 109 | DOC3 | Benchmark MOP with constraints in decision and objective spaces | | √ | | √ | | | | √ | | | | |
| 110 | DOC4 | Benchmark MOP with constraints in decision and objective spaces | | √ | | √ | | | | √ | | | | |
| 111 | DOC5 | Benchmark MOP with constraints in decision and objective spaces | | √ | | √ | | | | √ | | | | |
| 112 | DOC6 | Benchmark MOP with constraints in decision and objective spaces | | √ | | √ | | | | √ | | | | |
| 113 | DOC7 | Benchmark MOP with constraints in decision and objective spaces | | √ | | √ | | | | √ | | | | |
| 114 | DOC8 | Benchmark MOP with constraints in decision and objective spaces | | √ | | √ | | | | √ | | | | |
| 115 | DOC9 | Benchmark MOP with constraints in decision and objective spaces | | √ | | √ | | | | √ | | | | |
| 116 | DTLZ1 | Benchmark MOP proposed by Deb, Thiele, Laumanns, and Zitzler | | √ | √ | √ | | | √ | | √ | | | |
| 117 | DTLZ2 | Benchmark MOP proposed by Deb, Thiele, Laumanns, and Zitzler | | √ | √ | √ | | | √ | | √ | | | |
| 118 | DTLZ3 | Benchmark MOP proposed by Deb, Thiele, Laumanns, and Zitzler | | √ | √ | √ | | | √ | | √ | | | |
| 119 | DTLZ4 | Benchmark MOP proposed by Deb, Thiele, Laumanns, and Zitzler | | √ | √ | √ | | | √ | | √ | | | |
| 120 | DTLZ5 | Benchmark MOP proposed by Deb, Thiele, Laumanns, and Zitzler | | √ | √ | √ | | | √ | | √ | | | |
| 121 | DTLZ6 | Benchmark MOP proposed by Deb, Thiele, Laumanns, and Zitzler | | √ | √ | √ | | | √ | | √ | | | |
| 122 | DTLZ7 | Benchmark MOP proposed by Deb, Thiele, Laumanns, and Zitzler | | √ | √ | √ | | | √ | | √ | | | |
| 123 | DTLZ8 | Benchmark MOP proposed by Deb, Thiele, Laumanns, and Zitzler | | √ | √ | √ | | | √ | √ | √ | | | |
| 124 | DTLZ9 | Benchmark MOP proposed by Deb, Thiele, Laumanns, and Zitzler | | √ | √ | √ | | | √ | √ | √ | | | |
| 125 | CDTLZ2 | Convex DTLZ2 | | √ | √ | √ | | | √ | | √ | | | |
| 126 | IDTLZ1 | Inverted DTLZ1 | | √ | √ | √ | | | √ | | √ | | | |
| 127 | IDTLZ2 | Inverted DTLZ2 | | √ | √ | √ | | | √ | | √ | | | |
| 128 | SDTLZ1 | Scaled DTLZ1 | | √ | √ | √ | | | √ | | √ | | | |
| 129 | SDTLZ2 | Scaled DTLZ2 | | √ | √ | √ | | | √ | | √ | | | |
| 130 | C1-DTLZ1 | Constrained DTLZ1 | | √ | √ | √ | | | √ | √ | √ | | | |
| 131 | C1-DTLZ3 | Constrained DTLZ3 | | √ | √ | √ | | | √ | √ | √ | | | |
| 132 | C2-DTLZ2 | Constrained DTLZ2 | | √ | √ | √ | | | √ | √ | √ | | | |
| 133 | C3-DTLZ4 | Constrained DTLZ4 | | √ | √ | √ | | | √ | √ | √ | | | |
| 134 | DC1-DTLZ1 | DTLZ1 with constrains in decision space | | √ | √ | √ | | | √ | √ | √ | | | |
| 135 | DC1-DTLZ3 | DTLZ3 with constrains in decision space | | √ | √ | √ | | | √ | √ | √ | | | |
| 136 | DC2-DTLZ1 | DTLZ1 with constrains in decision space | | √ | √ | √ | | | √ | √ | √ | | | |
| 137 | DC2-DTLZ3 | DTLZ3 with constrains in decision space | | √ | √ | √ | | | √ | √ | √ | | | |
| 138 | DC3-DTLZ1 | DTLZ1 with constrains in decision space | | √ | √ | √ | | | √ | √ | √ | | | |
| 139 | DC3-DTLZ3 | DTLZ3 with constrains in decision space | | √ | √ | √ | | | √ | √ | √ | | | |
| 140 | FCP1 | Benchmark constrained MOP proposed by Yuan | | √ | | √ | | | | √ | | | | |
| 141 | FCP2 | Benchmark constrained MOP proposed by Yuan | | √ | | √ | | | | √ | | | | |
| 142 | FCP3 | Benchmark constrained MOP proposed by Yuan | | √ | | √ | | | | √ | | | | |
| 143 | FCP4 | Benchmark constrained MOP proposed by Yuan | | √ | | √ | | | | √ | | | | |

| | 问题缩写 | 问题全称 | single | multi | many | real | binary | permutation | large | constrained | expensive | multimodal | sparse | preference |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 144 | FCP5 | Benchmark constrained MOP proposed by Yuan | | √ | | √ | | | | √ | | | | |
| 145 | IMMOEA_F1 | Benchmark MOP for testing IM-MOEA | | √ | | √ | | | √ | | | | | |
| 146 | IMMOEA_F2 | Benchmark MOP for testing IM-MOEA | | √ | | √ | | | √ | | | | | |
| 147 | IMMOEA_F3 | Benchmark MOP for testing IM-MOEA | | √ | | √ | | | √ | | | | | |
| 148 | IMMOEA_F4 | Benchmark MOP for testing IM-MOEA | | √ | | √ | | | √ | | | | | |
| 149 | IMMOEA_F5 | Benchmark MOP for testing IM-MOEA | | √ | | √ | | | √ | | | | | |
| 150 | IMMOEA_F6 | Benchmark MOP for testing IM-MOEA | | √ | | √ | | | √ | | | | | |
| 151 | IMMOEA_F7 | Benchmark MOP for testing IM-MOEA | | √ | | √ | | | √ | | | | | |
| 152 | IMMOEA_F8 | Benchmark MOP for testing IM-MOEA | | √ | | √ | | | √ | | | | | |
| 153 | IMMOEA_F9 | Benchmark MOP for testing IM-MOEA | | √ | | √ | | | √ | | | | | |
| 154 | IMMOEA_F10 | Benchmark MOP for testing IM-MOEA | | √ | | √ | | | √ | | | | | |
| 155 | IMOP1 | Benchmark MOP with irregular Pareto front | | √ | | √ | | | | | √ | | | |
| 156 | IMOP2 | Benchmark MOP with irregular Pareto front | | √ | | √ | | | | | √ | | | |
| 157 | IMOP3 | Benchmark MOP with irregular Pareto front | | √ | | √ | | | | | √ | | | |
| 158 | IMOP4 | Benchmark MOP with irregular Pareto front | | √ | | √ | | | | | √ | | | |
| 159 | IMOP5 | Benchmark MOP with irregular Pareto front | | √ | | √ | | | | | √ | | | |
| 160 | IMOP6 | Benchmark MOP with irregular Pareto front | | √ | | √ | | | | | √ | | | |
| 161 | IMOP7 | Benchmark MOP with irregular Pareto front | | √ | | √ | | | | | √ | | | |
| 162 | IMOP8 | Benchmark MOP with irregular Pareto front | | √ | | √ | | | | | √ | | | |
| 163 | KP | The knapsack problem | √ | | | | √ | | √ | √ | | | | |
| 164 | LIR-CMOP1 | Constrained benchmark MOP with large infeasible regions | | √ | | √ | | | √ | √ | | | | |
| 165 | LIR-CMOP2 | Constrained benchmark MOP with large infeasible regions | | √ | | √ | | | √ | √ | | | | |
| 166 | LIR-CMOP3 | Constrained benchmark MOP with large infeasible regions | | √ | | √ | | | √ | √ | | | | |
| 167 | LIR-CMOP4 | Constrained benchmark MOP with large infeasible regions | | √ | | √ | | | √ | √ | | | | |
| 168 | LIR-CMOP5 | Constrained benchmark MOP with large infeasible regions | | √ | | √ | | | √ | √ | | | | |
| 169 | LIR-CMOP6 | Constrained benchmark MOP with large infeasible regions | | √ | | √ | | | √ | √ | | | | |
| 170 | LIR-CMOP7 | Constrained benchmark MOP with large infeasible regions | | √ | | √ | | | √ | √ | | | | |
| 171 | LIR-CMOP8 | Constrained benchmark MOP with large infeasible regions | | √ | | √ | | | √ | √ | | | | |
| 172 | LIR-CMOP9 | Constrained benchmark MOP with large infeasible regions | | √ | | √ | | | √ | √ | | | | |
| 173 | LIR-CMOP10 | Constrained benchmark MOP with large infeasible regions | | √ | | √ | | | √ | √ | | | | |
| 174 | LIR-CMOP11 | Constrained benchmark MOP with large infeasible regions | | √ | | √ | | | √ | √ | | | | |
| 175 | LIR-CMOP12 | Constrained benchmark MOP with large infeasible regions | | √ | | √ | | | √ | √ | | | | |
| 176 | LIR-CMOP13 | Constrained benchmark MOP with large infeasible regions | | √ | | √ | | | √ | √ | | | | |
| 177 | LIR-CMOP14 | Constrained benchmark MOP with large infeasible regions | | √ | | √ | | | √ | √ | | | | |
| 178 | LSMOP1 | Large-scale benchmark MOP | | √ | √ | √ | | | √ | | | | | |
| 179 | LSMOP2 | Large-scale benchmark MOP | | √ | √ | √ | | | √ | | | | | |
| 180 | LSMOP3 | Large-scale benchmark MOP | | √ | √ | √ | | | √ | | | | | |

| | 问题缩写 | 问题全称 | single | multi | many | real | binary | permutation | large | constrained | expensive | multimodal | sparse | preference |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 181 | LSMOP4 | Large-scale benchmark MOP | | √ | √ | √ | | | √ | | | | | |
| 182 | LSMOP5 | Large-scale benchmark MOP | | √ | √ | √ | | | √ | | | | | |
| 183 | LSMOP6 | Large-scale benchmark MOP | | √ | √ | √ | | | √ | | | | | |
| 184 | LSMOP7 | Large-scale benchmark MOP | | √ | √ | √ | | | √ | | | | | |
| 185 | LSMOP8 | Large-scale benchmark MOP | | √ | √ | √ | | | √ | | | | | |
| 186 | LSMOP9 | Large-scale benchmark MOP | | √ | √ | √ | | | √ | | | | | |
| 187 | MaF1 | Inverted DTLZ1 | | √ | √ | √ | | | √ | | | | | |
| 188 | MaF2 | DTLZ2BZ | | √ | √ | √ | | | √ | | | | | |
| 189 | MaF3 | Convex DTLZ3 | | √ | √ | √ | | | √ | | | | | |
| 190 | MaF4 | Inverted and scaled DTLZ3 | | √ | √ | √ | | | √ | | | | | |
| 191 | MaF5 | Scaled DTLZ4 | | √ | √ | √ | | | √ | | | | | |
| 192 | MaF6 | DTLZ5IM | | √ | √ | √ | | | √ | | | | | |
| 193 | MaF7 | DTLZ7 | | √ | √ | √ | | | √ | | | | | |
| 194 | MaF8 | MP-DMP | | √ | √ | √ | | | | | | | | |
| 195 | MaF9 | ML-DMP | | √ | √ | √ | | | | | | | | |
| 196 | MaF10 | WFG1 | | √ | √ | √ | | | √ | | | | | |
| 197 | MaF11 | WFG2 | | √ | √ | √ | | | √ | | | | | |
| 198 | MaF12 | WFG9 | | √ | √ | √ | | | √ | | | | | |
| 199 | MaF13 | P7 | | √ | √ | √ | | | √ | | | | | |
| 200 | MaF14 | LSMOP3 | | √ | √ | √ | | | √ | | | | | |
| 201 | MaF15 | Inverted LSMOP8 | | √ | √ | √ | | | √ | | | | | |
| 202 | MLDMP | The multi-line distance minimization problem | | √ | √ | √ | | | | | | | | |
| 203 | MMF1 | Multi-modal multi-objective test function | | √ | | √ | | | | | | √ | | |
| 204 | MMF2 | Multi-modal multi-objective test function | | √ | | √ | | | | | | √ | | |
| 205 | MMF3 | Multi-modal multi-objective test function | | √ | | √ | | | | | | √ | | |
| 206 | MMF4 | Multi-modal multi-objective test function | | √ | | √ | | | | | | √ | | |
| 207 | MMF5 | Multi-modal multi-objective test function | | √ | | √ | | | | | | √ | | |
| 208 | MMF6 | Multi-modal multi-objective test function | | √ | | √ | | | | | | √ | | |
| 209 | MMF7 | Multi-modal multi-objective test function | | √ | | √ | | | | | | √ | | |
| 210 | MMF8 | Multi-modal multi-objective test function | | √ | | √ | | | | | | √ | | |
| 211 | MMMOP1 | Multi-modal multi-objective optimization problem | | √ | √ | √ | | | | | | √ | | |
| 212 | MMMOP2 | Multi-modal multi-objective optimization problem | | √ | √ | √ | | | | | | √ | | |
| 213 | MMMOP3 | Multi-modal multi-objective optimization problem | | √ | √ | √ | | | | | | √ | | |
| 214 | MMMOP4 | Multi-modal multi-objective optimization problem | | √ | √ | √ | | | | | | √ | | |
| 215 | MMMOP5 | Multi-modal multi-objective optimization problem | | √ | √ | √ | | | | | | √ | | |
| 216 | MMMOP6 | Multi-modal multi-objective optimization problem | | √ | √ | √ | | | | | | √ | | |
| 217 | MOEADDE_F1 | Benchmark MOP for testing MOEA/D-DE | | √ | | √ | | | √ | | | | | |

| | 问题缩写 | 问题全称 | single | multi | many | real | binary | permutation | large | constrained | expensive | multimodal | sparse | preference |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 218 | MOEADDE_F2 | Benchmark MOP for testing MOEA/D-DE | | √ | | √ | | | √ | | | | | |
| 219 | MOEADDE_F3 | Benchmark MOP for testing MOEA/D-DE | | √ | | √ | | | √ | | | | | |
| 220 | MOEADDE_F4 | Benchmark MOP for testing MOEA/D-DE | | √ | | √ | | | √ | | | | | |
| 221 | MOEADDE_F5 | Benchmark MOP for testing MOEA/D-DE | | √ | | √ | | | √ | | | | | |
| 222 | MOEADDE_F6 | Benchmark MOP for testing MOEA/D-DE | | √ | | √ | | | √ | | | | | |
| 223 | MOEADDE_F7 | Benchmark MOP for testing MOEA/D-DE | | √ | | √ | | | √ | | | | | |
| 224 | MOEADDE_F8 | Benchmark MOP for testing MOEA/D-DE | | √ | | √ | | | √ | | | | | |
| 225 | MOEADDE_F9 | Benchmark MOP for testing MOEA/D-DE | | √ | | √ | | | √ | | | | | |
| 226 | MOEADM2M_F1 | Benchmark MOP for testing MOEA/D-M2M | | √ | | √ | | | √ | | | | | |
| 227 | MOEADM2M_F2 | Benchmark MOP for testing MOEA/D-M2M | | √ | | √ | | | √ | | | | | |
| 228 | MOEADM2M_F3 | Benchmark MOP for testing MOEA/D-M2M | | √ | | √ | | | √ | | | | | |
| 229 | MOEADM2M_F4 | Benchmark MOP for testing MOEA/D-M2M | | √ | | √ | | | √ | | | | | |
| 230 | MOEADM2M_F5 | Benchmark MOP for testing MOEA/D-M2M | | √ | | √ | | | √ | | | | | |
| 231 | MOEADM2M_F6 | Benchmark MOP for testing MOEA/D-M2M | | √ | | √ | | | √ | | | | | |
| 232 | MOEADM2M_F7 | Benchmark MOP for testing MOEA/D-M2M | | √ | | √ | | | √ | | | | | |
| 233 | MOKP | The multi-objective knapsack problem | | √ | √ | | √ | | √ | | | | | |
| 234 | MONRP | The multi-objective next release problem | | √ | | | √ | | √ | | | | | |
| 235 | MOTSP | The multi-objective traveling salesman problem | | √ | √ | | | √ | √ | | | | | |
| 236 | MPDMP | The multi-point distance minimization problem | | √ | √ | √ | | | | | | | | |
| 237 | mQAP | The multi-objective quadratic assignment problem | | √ | √ | | | √ | √ | | | | | |
| 238 | MW1 | Constrained benchmark MOP proposed by Ma and Wang | | √ | | √ | | | √ | √ | | | | |
| 239 | MW2 | Constrained benchmark MOP proposed by Ma and Wang | | √ | | √ | | | √ | √ | | | | |
| 240 | MW3 | Constrained benchmark MOP proposed by Ma and Wang | | √ | | √ | | | √ | √ | | | | |
| 241 | MW4 | Constrained benchmark MOP proposed by Ma and Wang | | √ | √ | √ | | | √ | √ | | | | |
| 242 | MW5 | Constrained benchmark MOP proposed by Ma and Wang | | √ | | √ | | | √ | √ | | | | |
| 243 | MW6 | Constrained benchmark MOP proposed by Ma and Wang | | √ | | √ | | | √ | √ | | | | |
| 244 | MW7 | Constrained benchmark MOP proposed by Ma and Wang | | √ | | √ | | | √ | √ | | | | |
| 245 | MW8 | Constrained benchmark MOP proposed by Ma and Wang | | √ | √ | √ | | | √ | √ | | | | |
| 246 | MW9 | Constrained benchmark MOP proposed by Ma and Wang | | √ | | √ | | | √ | √ | | | | |
| 247 | MW10 | Constrained benchmark MOP proposed by Ma and Wang | | √ | | √ | | | √ | √ | | | | |
| 248 | MW11 | Constrained benchmark MOP proposed by Ma and Wang | | √ | | √ | | | √ | √ | | | | |
| 249 | MW12 | Constrained benchmark MOP proposed by Ma and Wang | | √ | | √ | | | √ | √ | | | | |
| 250 | MW13 | Constrained benchmark MOP proposed by Ma and Wang | | √ | | √ | | | √ | √ | | | | |
| 251 | MW14 | Constrained benchmark MOP proposed by Ma and Wang | | √ | √ | √ | | | √ | √ | | | | |
| 252 | RMMEDA_F1 | Benchmark MOP for testing RM-MEDA | | √ | | √ | | | √ | | | | | |
| 253 | RMMEDA_F2 | Benchmark MOP for testing RM-MEDA | | √ | | √ | | | √ | | | | | |
| 254 | RMMEDA_F3 | Benchmark MOP for testing RM-MEDA | | √ | | √ | | | √ | | | | | |

| | 问题缩写 | 问题全称 | single | multi | many | real | binary | permutation | large | constrained | expensive | multimodal | sparse | preference |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 255 | RMMEDA_F4 | Benchmark MOP for testing RM-MEDA | | √ | | √ | | | √ | | | | | |
| 256 | RMMEDA_F5 | Benchmark MOP for testing RM-MEDA | | √ | | √ | | | √ | | | | | |
| 257 | RMMEDA_F6 | Benchmark MOP for testing RM-MEDA | | √ | | √ | | | √ | | | | | |
| 258 | RMMEDA_F7 | Benchmark MOP for testing RM-MEDA | | √ | | √ | | | √ | | | | | |
| 259 | RMMEDA_F8 | Benchmark MOP for testing RM-MEDA | | √ | | √ | | | √ | | | | | |
| 260 | RMMEDA_F9 | Benchmark MOP for testing RM-MEDA | | √ | | √ | | | √ | | | | | |
| 261 | RMMEDA_F10 | Benchmark MOP for testing RM-MEDA | | √ | | √ | | | √ | | | | | |
| 262 | Sparse_CD | The community detection problem | | √ | | | √ | | √ | | √ | | √ | |
| 263 | Sparse_CN | The critical node detection problem | | √ | | | √ | | √ | | √ | | √ | |
| 264 | Sparse_FS | The feature selection problem | | √ | | | √ | | √ | | √ | | √ | |
| 265 | Sparse_IS | The instance selection problem | | √ | | | √ | | √ | | √ | | √ | |
| 266 | Sparse_KP | The sparse multi-objective knapsack problem | | √ | √ | | √ | | √ | | | | | |
| 267 | Sparse_NN | The neural network training problem | | √ | | √ | | | √ | | √ | | √ | |
| 268 | Sparse_PM | The pattern mining problem | | √ | | | √ | | √ | | √ | | √ | |
| 269 | Sparse_PO | The portfolio optimization problem | | √ | | √ | | | √ | | √ | | √ | |
| 270 | Sparse_SR | The sparse signal reconstruction problem | | √ | | √ | | | √ | | √ | | √ | |
| 271 | SMMOP1 | Sparse multi-modal multi-objective optimization problem | | √ | √ | √ | | | √ | | | √ | √ | |
| 272 | SMMOP2 | Sparse multi-modal multi-objective optimization problem | | √ | √ | √ | | | √ | | | √ | √ | |
| 273 | SMMOP3 | Sparse multi-modal multi-objective optimization problem | | √ | √ | √ | | | √ | | | √ | √ | |
| 274 | SMMOP4 | Sparse multi-modal multi-objective optimization problem | | √ | √ | √ | | | √ | | | √ | √ | |
| 275 | SMMOP5 | Sparse multi-modal multi-objective optimization problem | | √ | √ | √ | | | √ | | | √ | √ | |
| 276 | SMMOP6 | Sparse multi-modal multi-objective optimization problem | | √ | √ | √ | | | √ | | | √ | √ | |
| 277 | SMMOP7 | Sparse multi-modal multi-objective optimization problem | | √ | √ | √ | | | √ | | | √ | √ | |
| 278 | SMMOP8 | Sparse multi-modal multi-objective optimization problem | | √ | √ | √ | | | √ | | | √ | √ | |
| 279 | SMOP1 | Benchmark MOP with sparse Pareto optimal solutions | | √ | √ | √ | | | √ | | √ | | √ | |
| 280 | SMOP2 | Benchmark MOP with sparse Pareto optimal solutions | | √ | √ | √ | | | √ | | √ | | √ | |
| 281 | SMOP3 | Benchmark MOP with sparse Pareto optimal solutions | | √ | √ | √ | | | √ | | √ | | √ | |
| 282 | SMOP4 | Benchmark MOP with sparse Pareto optimal solutions | | √ | √ | √ | | | √ | | √ | | √ | |
| 283 | SMOP5 | Benchmark MOP with sparse Pareto optimal solutions | | √ | √ | √ | | | √ | | √ | | √ | |
| 284 | SMOP6 | Benchmark MOP with sparse Pareto optimal solutions | | √ | √ | √ | | | √ | | √ | | √ | |
| 285 | SMOP7 | Benchmark MOP with sparse Pareto optimal solutions | | √ | √ | √ | | | √ | | √ | | √ | |
| 286 | SMOP8 | Benchmark MOP with sparse Pareto optimal solutions | | √ | √ | √ | | | √ | | √ | | √ | |
| 287 | SOP_F1 | Sphere function | √ | | | √ | | | | | √ | | | |
| 288 | SOP_F2 | Schwefel's function 2.22 | √ | | | √ | | | | | √ | | | |
| 289 | SOP_F3 | Schwefel's function 1.2 | √ | | | √ | | | | | √ | | | |
| 290 | SOP_F4 | Schwefel's function 2.21 | √ | | | √ | | | | | √ | | | |
| 291 | SOP_F5 | Generalized Rosenbrock's function | √ | | | √ | | | | | √ | | | |

| | 问题缩写 | 问题全称 | single | multi | many | real | binary | permutation | large | constrained | expensive | multimodal | sparse | preference |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 292 | SOP_F6 | Step function | √ | | | √ | | | | | √ | | | |
| 293 | SOP_F7 | Quartic function with noise | √ | | | √ | | | | | √ | | | |
| 294 | SOP_F8 | Generalized Schwefel's function 2.26 | √ | | | √ | | | | | √ | | | |
| 295 | SOP_F9 | Generalized Rastrigin's function | √ | | | √ | | | | | √ | | | |
| 296 | SOP_F10 | Ackley's function | √ | | | √ | | | | | √ | | | |
| 297 | SOP_F11 | Generalized Griewank's function | √ | | | √ | | | | | √ | | | |
| 298 | SOP_F12 | Generalized penalized function | √ | | | √ | | | | | √ | | | |
| 299 | SOP_F13 | Generalized penalized function | √ | | | √ | | | | | √ | | | |
| 300 | SOP_F14 | Shekel's foxholes function | √ | | | √ | | | | | √ | | | |
| 301 | SOP_F15 | Kowalik's function | √ | | | √ | | | | | √ | | | |
| 302 | SOP_F16 | Six-hump camel-back function | √ | | | √ | | | | | √ | | | |
| 303 | SOP_F17 | Branin function | √ | | | √ | | | | | √ | | | |
| 304 | SOP_F18 | Goldstein-price function | √ | | | √ | | | | | √ | | | |
| 305 | SOP_F19 | Hartman's family | √ | | | √ | | | | | √ | | | |
| 306 | SOP_F20 | Hartman's family | √ | | | √ | | | | | √ | | | |
| 307 | SOP_F21 | Shekel's family | √ | | | √ | | | | | √ | | | |
| 308 | SOP_F22 | Shekel's family | √ | | | √ | | | | | √ | | | |
| 309 | SOP_F23 | Shekel's family | √ | | | √ | | | | | √ | | | |
| 310 | TREE1 | The time-varying ratio error estimation problem | | √ | | √ | | | √ | √ | √ | | | |
| 311 | TREE2 | The time-varying ratio error estimation problem | | √ | | √ | | | √ | √ | √ | | | |
| 312 | TREE3 | The time-varying ratio error estimation problem | | √ | | √ | | | √ | √ | √ | | | |
| 313 | TREE4 | The time-varying ratio error estimation problem | | √ | | √ | | | √ | √ | √ | | | |
| 314 | TREE5 | The time-varying ratio error estimation problem | | √ | | √ | | | √ | √ | √ | | | |
| 315 | TREE6 | The time-varying ratio error estimation problem | | √ | | √ | | | √ | √ | √ | | | |
| 316 | TSP | The traveling salesman problem | √ | | | | | √ | √ | | | | | |
| 317 | UF1 | Unconstrained benchmark MOP | | √ | | √ | | | √ | | | | | |
| 318 | UF2 | Unconstrained benchmark MOP | | √ | | √ | | | √ | | | | | |
| 319 | UF3 | Unconstrained benchmark MOP | | √ | | √ | | | √ | | | | | |
| 320 | UF4 | Unconstrained benchmark MOP | | √ | | √ | | | √ | | | | | |
| 321 | UF5 | Unconstrained benchmark MOP | | √ | | √ | | | √ | | | | | |
| 322 | UF6 | Unconstrained benchmark MOP | | √ | | √ | | | √ | | | | | |
| 323 | UF7 | Unconstrained benchmark MOP | | √ | | √ | | | √ | | | | | |
| 324 | UF8 | Unconstrained benchmark MOP | | √ | | √ | | | √ | | | | | |
| 325 | UF9 | Unconstrained benchmark MOP | | √ | | √ | | | √ | | | | | |
| 326 | UF10 | Unconstrained benchmark MOP | | √ | | √ | | | √ | | | | | |
| 327 | VNT1 | Benchmark MOP proposed by Viennet | | √ | | √ | | | | | | | | |
| 328 | VNT2 | Benchmark MOP proposed by Viennet | | √ | | √ | | | | | | | | |

| | 问题缩写 | 问题全称 | single | multi | many | real | binary | permutation | large | constrained | expensive | multimodal | sparse | preference |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 329 | VNT3 | Benchmark MOP proposed by Viennet | | √ | | √ | | | | | | | | |
| 330 | VNT4 | Benchmark MOP proposed by Viennet | | √ | | √ | | | | √ | | | | |
| 331 | WFG1 | Benchmark MOP proposed by Walking Fish Group | | √ | √ | √ | | | √ | | √ | | | |
| 332 | WFG2 | Benchmark MOP proposed by Walking Fish Group | | √ | √ | √ | | | √ | | √ | | | |
| 333 | WFG3 | Benchmark MOP proposed by Walking Fish Group | | √ | √ | √ | | | √ | | √ | | | |
| 334 | WFG4 | Benchmark MOP proposed by Walking Fish Group | | √ | √ | √ | | | √ | | √ | | | |
| 335 | WFG5 | Benchmark MOP proposed by Walking Fish Group | | √ | √ | √ | | | √ | | √ | | | |
| 336 | WFG6 | Benchmark MOP proposed by Walking Fish Group | | √ | √ | √ | | | √ | | √ | | | |
| 337 | WFG7 | Benchmark MOP proposed by Walking Fish Group | | √ | √ | √ | | | √ | | √ | | | |
| 338 | WFG8 | Benchmark MOP proposed by Walking Fish Group | | √ | √ | √ | | | √ | | √ | | | |
| 339 | WFG9 | Benchmark MOP proposed by Walking Fish Group | | √ | √ | √ | | | √ | | √ | | | |
| 340 | ZDT1 | Benchmark MOP proposed by Zitzler, Deb, and Thiele | | √ | | √ | | | √ | | √ | | | |
| 341 | ZDT2 | Benchmark MOP proposed by Zitzler, Deb, and Thiele | | √ | | √ | | | √ | | √ | | | |
| 342 | ZDT3 | Benchmark MOP proposed by Zitzler, Deb, and Thiele | | √ | | √ | | | √ | | √ | | | |
| 343 | ZDT4 | Benchmark MOP proposed by Zitzler, Deb, and Thiele | | √ | | √ | | | √ | | √ | | | |
| 344 | ZDT5 | Benchmark MOP proposed by Zitzler, Deb, and Thiele | | √ | | | √ | | √ | | √ | | | |
| 345 | ZDT6 | Benchmark MOP proposed by Zitzler, Deb, and Thiele | | √ | | √ | | | √ | | √ | | | |