

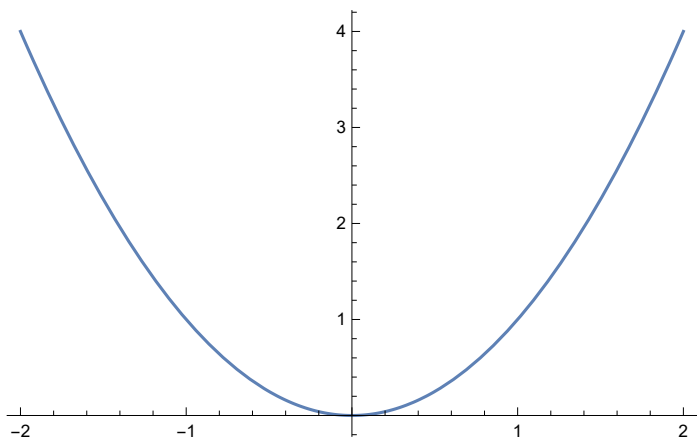
# 粒子群优化算法

## 函数图像

In[\*]:= **Plot**[ $x^2$ , {x, -2, 2}]

[绘图](#)

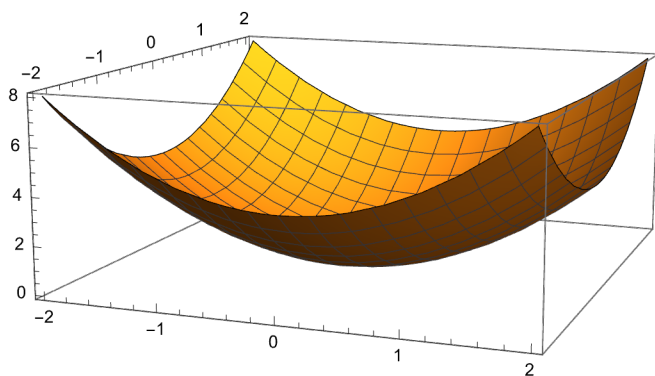
Out[\*]=



In[\*]:= **Plot3D**[ $x^2 + y^2$ , {x, -2, 2}, {y, -2, 2}]

[绘制三维图形](#)

Out[\*]=

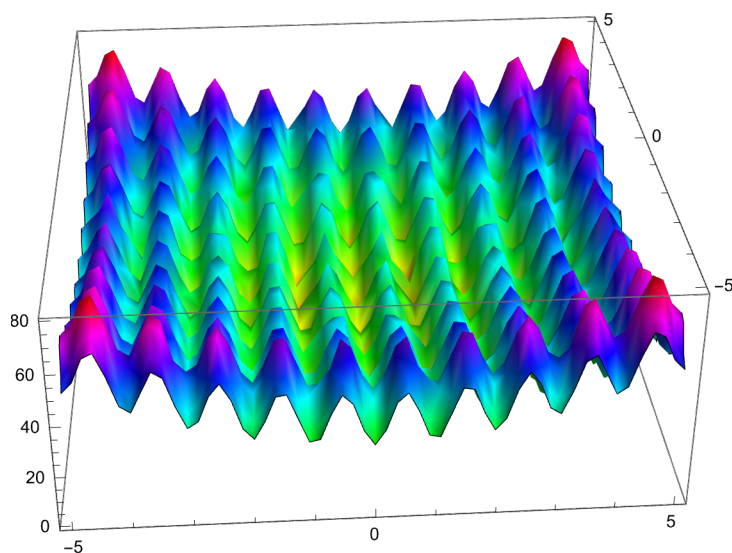


```

In[ ]:= a = 10;
Plot3D[20 + Sum[x^2 - a Cos[2 π x], {x, {x1, x2}}],
|绘制三维图形 |求和 |余弦
{x1, -5, 5}, {x2, -5, 5}, ColorFunction → Hue, Mesh → None]
|颜色函数 |色相 |网格 |无

```

Out[ ]:=

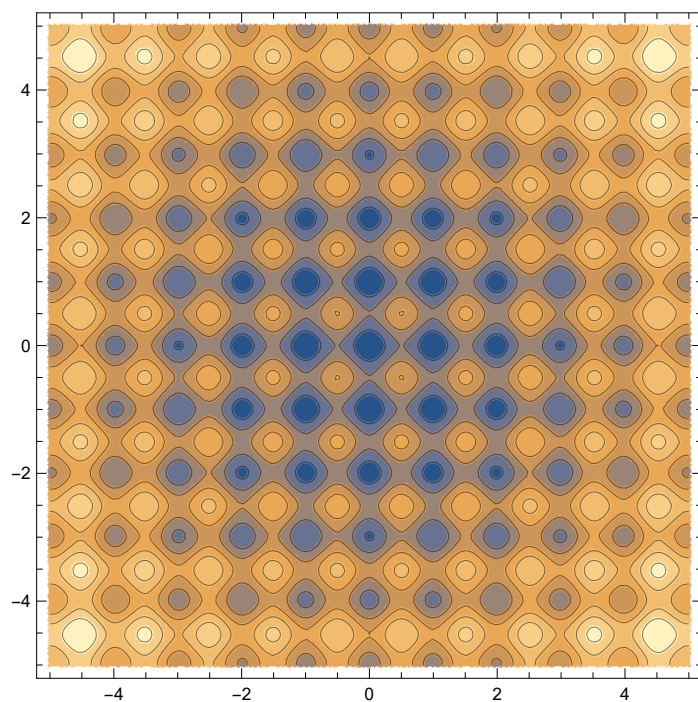


```

In[ ]:= ContourPlot[20 + Sum[x^2 - a Cos[2 π x], {x, {x1, x2}}], {x1, -5, 5}, {x2, -5, 5}]
|绘制等高线 |求和 |余弦

```

Out[ ]:=



```

In[*]:= FindMinimum[Sum[x2 - 10 Cos[2 π x], {x, {x1, x2}}],
  |求极小值和... |求和 |余弦
  {{x1, 3}, {x2, 3}}, WorkingPrecision -> 30]
  |工作精度

Out[*]=
{-2.09079751702597872937274781811,
 {x1 -> 2.98485570103948124686274356531, x2 -> 2.98485570103948124686274356531}}

In[*]:= Minimize[Sum[x2 - 10 Cos[2 π x], {x, {x1, x2}}], {x1, x2}]
|最小点值 |求和 |余弦

Out[*]=
Minimize[x12 + x22 - 10 Cos[2 π x1] - 10 Cos[2 π x2], {x1, x2}]

```

## PSO实现

### 目标函数(适应度)

```

In[*]:= fitness[xn_] := Sum[x2 - 10 Cos[2 π x], {x, xn}]
  |求和 |余弦

```

### 测试fitness

## 初始化参数

```

In[ ]:= init[] := (ω = 0.5;
    c1 = 2;
    c2 = 2;
    r1 = RandomReal[];
    伪随机实数
    r2 = RandomReal[];
    伪随机实数
    α = 1;

    dimension = 2;
    num = 50;
    vMax = Table[1, dimension];
    表格
    xMax = Table[5, dimension];
    表格
    x = RandomReal[{-5, 5}, {num, dimension}];
    伪随机实数
    v = RandomReal[{-1, 1}, {num, dimension}];
    伪随机实数
    px = x;
    pxF = fitness[#] & /@ px;

    pgF = Min[pxF];
    最小值
    pg = x[[PositionIndex[pxF][pgF][1]]];
    位置索引
    iterNum = 25;)

```

## 更新粒子

```

In[ ]:= pso[] := (
    ax = {x};
    Do[v = ω v + c1 r1 (px - x) + c2 r2 (Table[pg, num] - x);
    Do循环 表格
    x = x + α v;
    xF = fitness[#] & /@ x;
    AppendTo[ax, x];
    附加
    Do[
    Do循环
    If[xF[[r]] < pxF[[r]],
    如果
    px[[r]] = x[[r]]; pxF[[r]] = xF[[r]];
    If[xF[[r]] < pgF,
    如果
    pg = x[[r]]; pgF = xF[[r]]],
    {r, num}],
    iterNum)

```

## 计算

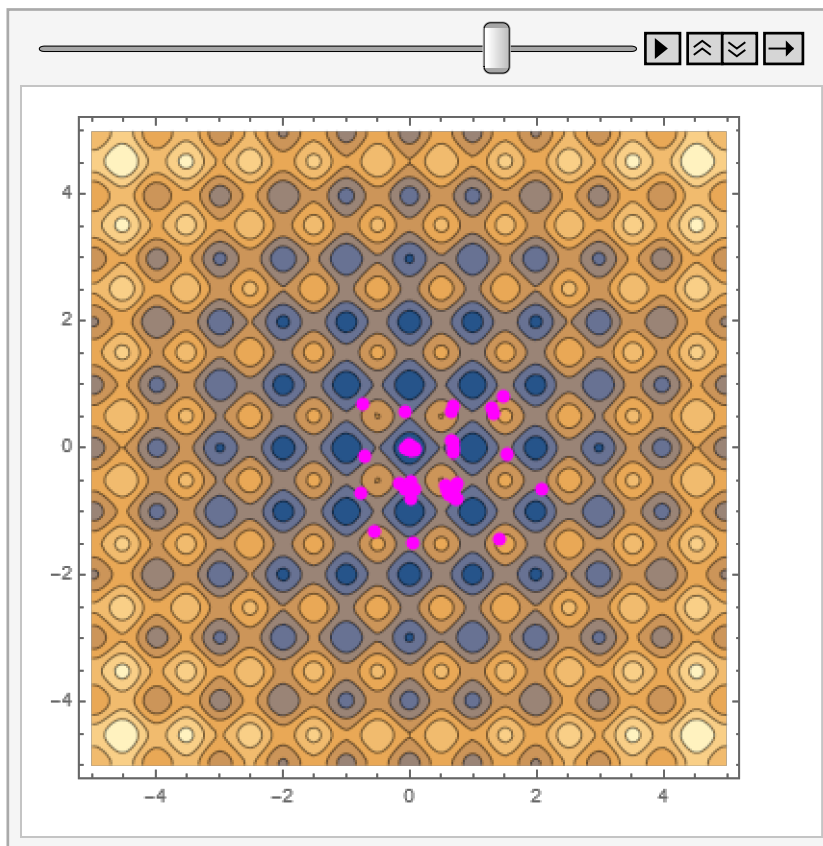
```
In[ ]:= init[]
      pso[]
      {pg, pgF}
```

```
Out[ ]:= {{-0.0000859619, -0.00224058}, -19.999}
```

```
In[ ]:= cp = ContourPlot[fitness[{x, y}], {x, -5, 5}, {y, -5, 5}];
          绘制等高线
lp[pts_] := ListPlot[pts, PlotStyle -> Directive[PointSize[Large], Magenta]];
          绘制点集          绘图样式          指令          点的大小          大          品红色
```

```
Monitor[m = Table[Image@Show[cp, lp[ax[i]]], {i, iterNum}];
          监控          表格          图像          显示
      ListAnimate[m], i]
          列表帧动画
```

```
Out[ ]:=
```



```
In[ ]:= Export["E:\\Mathematica\\pso.gif", m]
          导出          自然常数
```

```
Out[ ]:= E:\\Mathematica\\pso.gif
```

```
In[ ]:= l = {3, 9, 2, 5, 4, 8, 7, 0};  
Max[l]  
|最大值  
First@PositionIndex[l][Max[l]]  
|第一个 |位置索引 |最大值  
Position[l, 9]  
|位置  
Out[ ]:=  
9  
Out[ ]:=  
2  
Out[ ]:=  
{ {2} }
```