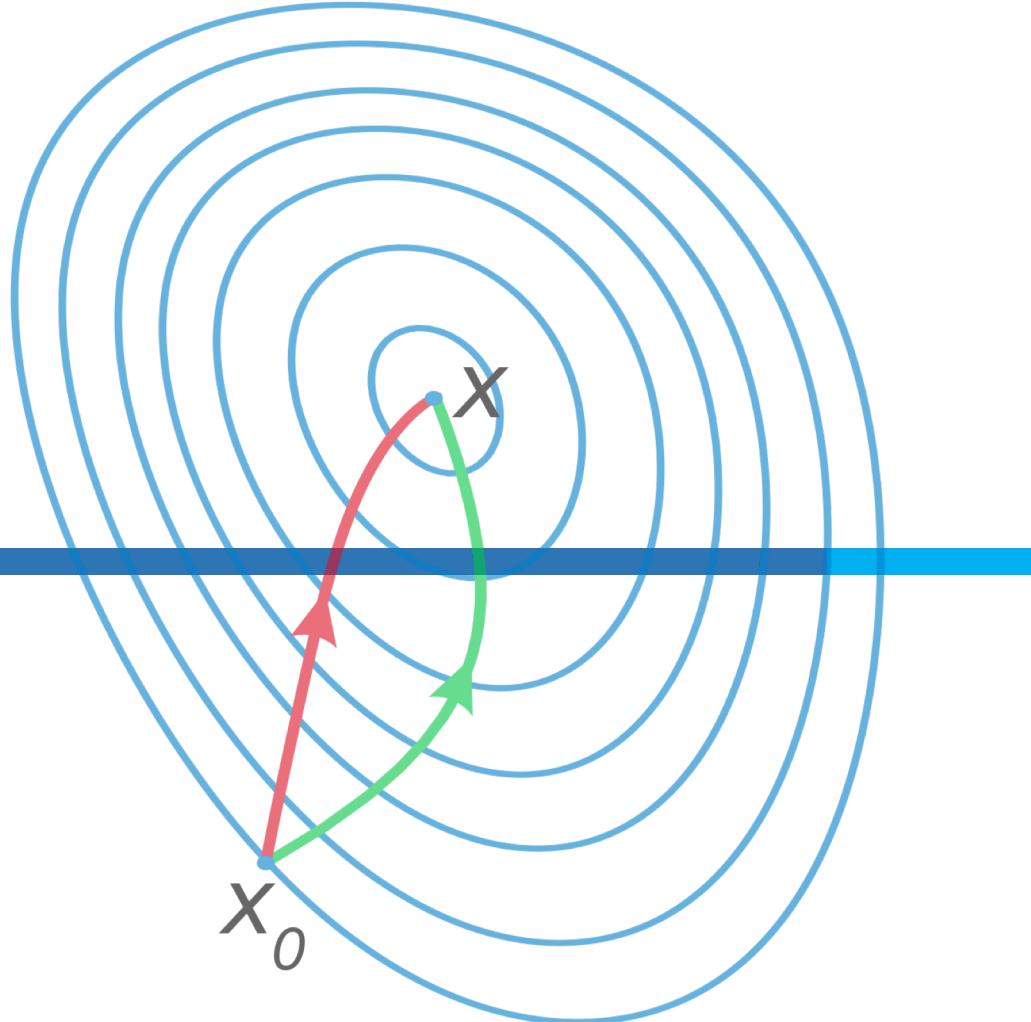


# 最优化方法

## 第八讲

计算机学院  
余皓然

2023/5/4



# 多目标优化问题

Part1 如何从复杂度的角度衡量算法好坏？

Part2 针对一些有特定形式的最优化问题，如何找到最优解？

Part3 针对较复杂的最优化问题（如NP难问题），如何找到较好解？

Part4 针对较复杂的多目标最优化问题，如何找到较好解？

多目标优化问题、NSGA-II算法

# 单目标优化

---

之前涉及的优化问题都仅有一个目标函数：

- 旅行商问题（路径总长度）
- **0-1背包问题**（装进背包物品总价值）
- 公共设施选址问题（各处到最近设施距离之和）
- 聚类问题（数据到类中心距离平方和）

$$\begin{aligned} \min \quad & f(\boldsymbol{x}) \\ \text{s.t.} \quad & g_i(\boldsymbol{x}) \leq 0, i = 1, \dots, m, \\ & h_i(\boldsymbol{x}) = 0, j = 1, \dots, p, \\ \text{var.} \quad & \boldsymbol{x} \in \mathcal{D}. \end{aligned}$$

# 多目标优化

---

多目标优化问题（multi-objective optimization）



选择餐馆：  
**最大化用餐体验；**  
**最小化花费**

# 多目标优化

## 多目标优化问题（multi-objective optimization）



买火车票/机票：  
最大化舒适程度；  
最小化花费

# 多目标优化

---

## 多目标优化问题（multi-objective optimization）

5	100074301	自然语言理解初步	32	2	考查	计算机学院
6	100074302	机器学习初步	32	2	考查	计算机学院
6	100074303	智能计算工程实践	40	2.5	考查	计算机学院
5	100074304	计算机新技术专题	32	2	考查	计算机学院
7	100074335	创新创业实践	64	2	考查	计算机学院
4	100074402	密码学基础	40	2.5	考查	计算机学院
6	100074403	网络与信息安全	32	2	考查	计算机学院
5	100074406	安全协议设计与分析	32	2	考查	计算机学院
4	100074407	知识工程	32	2	考查	计算机学院
4	100074408	网络与通信	32	2	考查	计算机学院
5	100074501	人工智能基础	40	2.5	考查	计算机学院
5	100074503	数字图像处理	40	2.5	考查	计算机学院
4	100074506	现代人机交互	32	2	考查	计算机学院
4	100074601	计算机图形学	40	2.5	考查	计算机学院

选课：

最大化分数；

最大化与个人发展匹配程度；

最大化听课体验

# 单目标优化与多目标优化

---

## 单目标优化

$$\begin{aligned} \min \quad & f(\boldsymbol{x}) \\ \text{s.t.} \quad & g_i(\boldsymbol{x}) \leq 0, i = 1, \dots, m, \\ & h_i(\boldsymbol{x}) = 0, j = 1, \dots, p, \\ \text{var.} \quad & \boldsymbol{x} \in \mathcal{D}. \end{aligned}$$

## 多目标优化

$$\begin{aligned} \min \quad & f_1(\boldsymbol{x}) \\ \min \quad & f_2(\boldsymbol{x}) \\ & \vdots \\ \min \quad & f_M(\boldsymbol{x}) \\ \text{s.t.} \quad & g_i(\boldsymbol{x}) \leq 0, i = 1, \dots, m, \\ & h_i(\boldsymbol{x}) = 0, j = 1, \dots, p, \\ \text{var.} \quad & \boldsymbol{x} \in \mathcal{D}. \end{aligned}$$

# 多目标优化

---



相关概念



经典多目标优化算法



基于遗传算法的多目标优化算法（NSGA-II）

# 多目标进化算法的应用

---

R. Shen, et al., “Generating Behavior-Diverse Game AIs with Evolutionary Multi-Objective Deep Reinforcement Learning,” IJCAI 2020.



进化强化学习生成风格多样AI，总有一款适合你！

<https://www.bilibili.com/video/BV16D4y1D7e2>

# 多目标优化：概念

Part1 如何从复杂度的角度衡量算法好坏？

Part2 针对一些有特定形式的最优化问题，如何找到最优解？

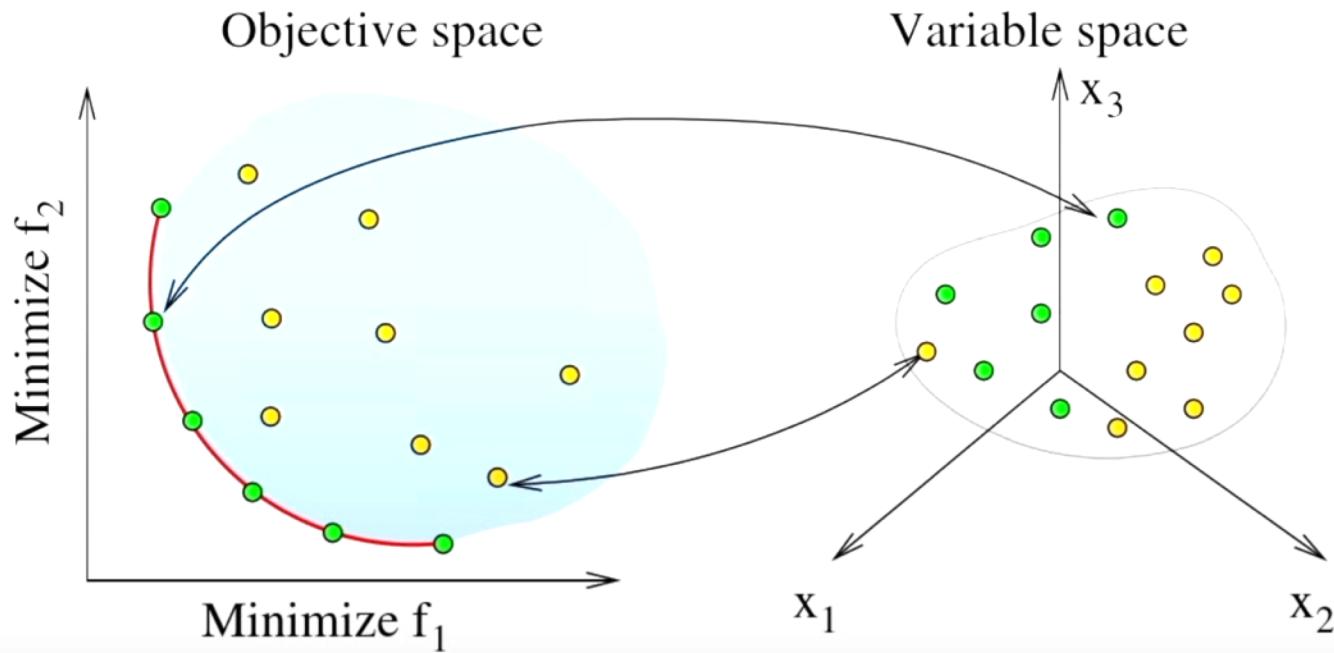
Part3 针对较复杂的最优化问题（如NP难问题），如何找到较好解？

Part4 针对较复杂的多目标最优化问题，如何找到较好解？

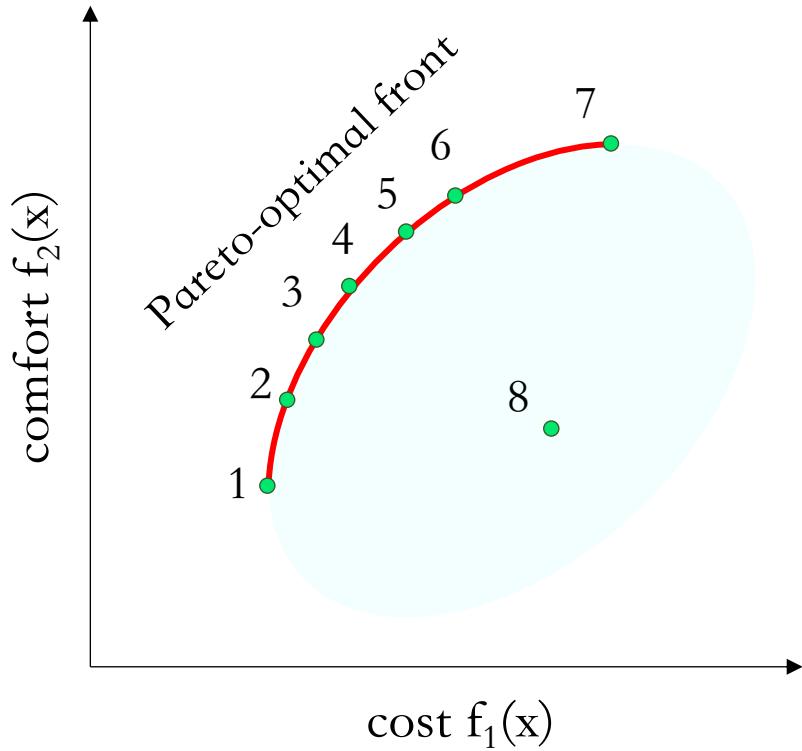
多目标优化问题、NSGA-II算法

# 支配与帕累托最优

区分目标函数值空间和解空间



# 支配与帕累托最优



假设有八款车可以选择

左图显示每款车（每个解）对应的  
舒适度和花费（目标函数值）

蓝色区域为所有可行解对应的目标  
函数值范围

注意：图中横纵坐标都是目标函数

# 支配与帕累托最优

支配 (Dominate) :

解A支配解B (A dominates B)

“A和B之间无脑选A”

↔ 解B被解A支配 (B is dominated by A)

↔ 解A的所有目标函数取值都不差于解B，且解A的至少一个目标函数取值优于解B



花费 3  
舒适 3



花费 100  
舒适 100

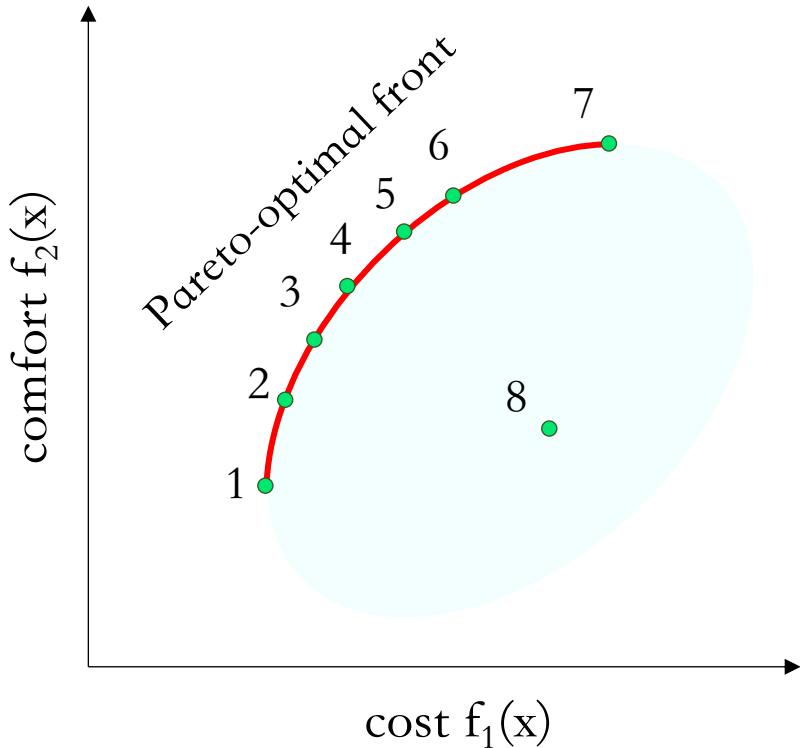


花费 120  
舒适 100



花费 500  
舒适 500

# 支配与帕累托最优



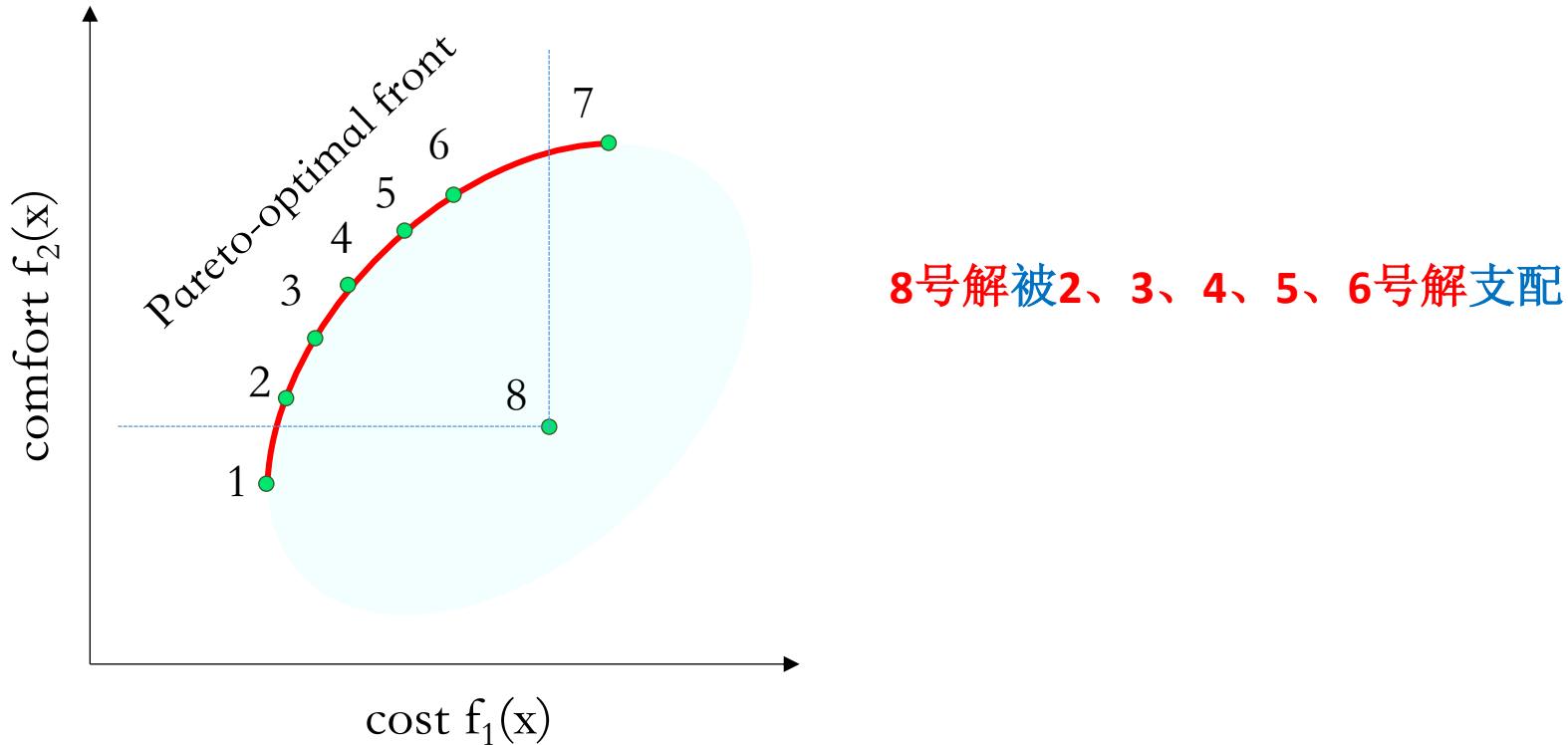
支配 (Dominate) :

解A支配解B (A dominates B)

↔ 解B被解A支配 (B is dominated by A)

↔ 解A的所有目标函数取值都不差于解B，且解A的至少一个目标函数取值优于解B

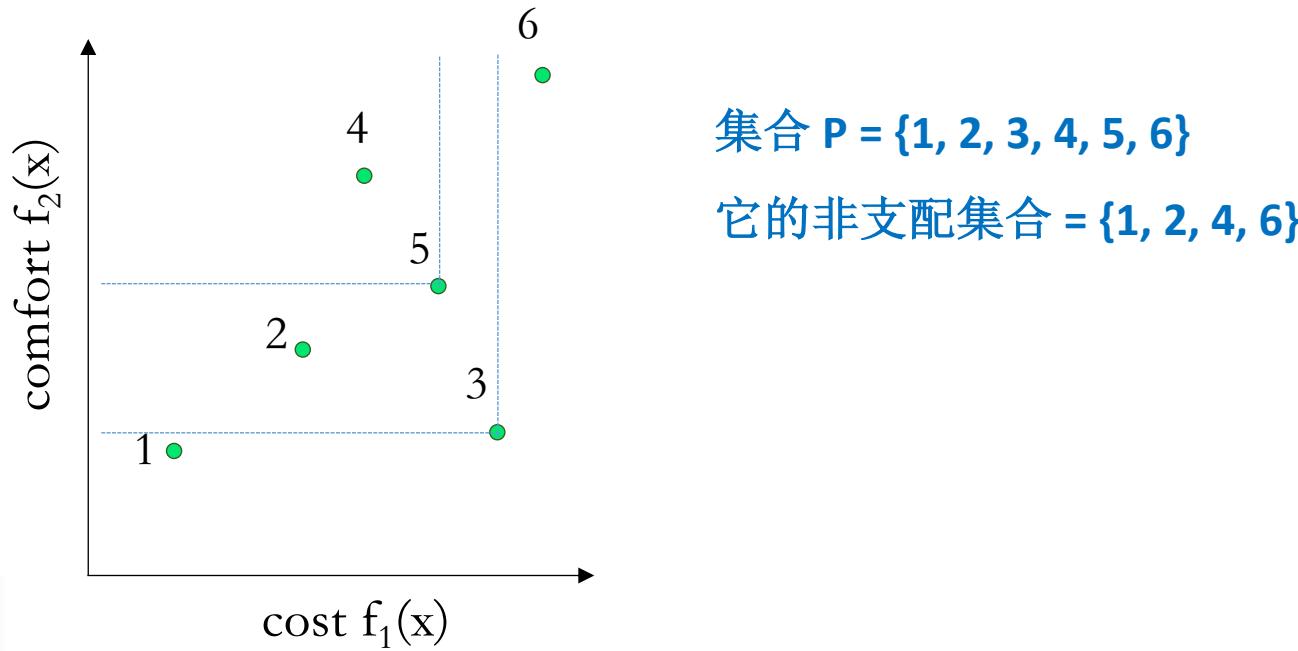
# 支配与帕累托最优



# 支配与帕累托最优

非支配集合：

对于一个集合  $P$ ，它的非支配集合是  $P$  的子集、包含所有不被任何集合  $P$  中其它解支配的解。



# 支配与帕累托最优

---

非支配集合：

对于一个集合 $P$ ，它的非支配集合是 $P$ 的子集、包含所有不被任何集合 $P$ 中其它解支配的解。

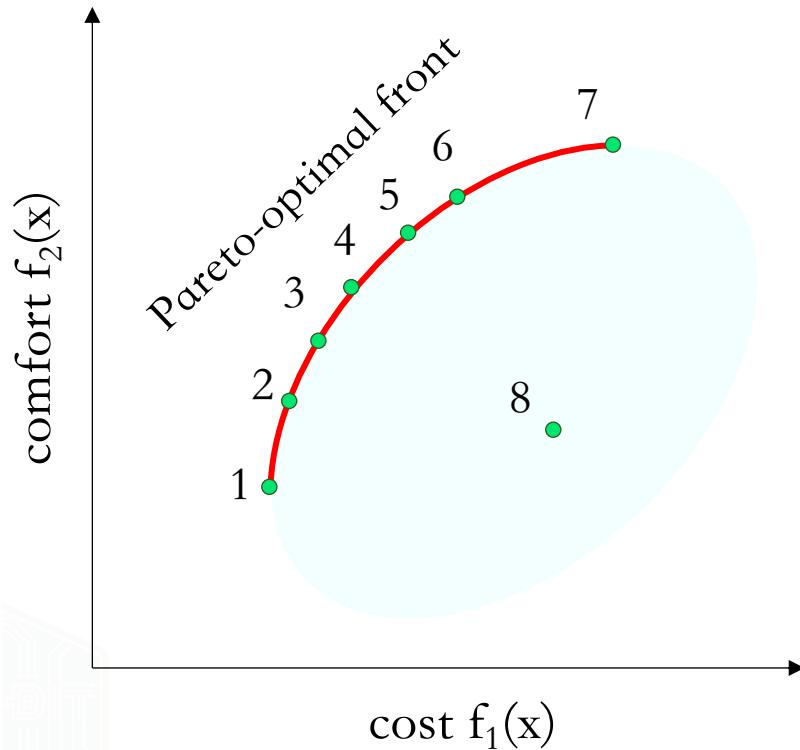
帕累托最优：

可行域的非支配集合即定义为帕累托最优集合（Pareto-optimal set）

# 支配与帕累托最优

帕累托最优:

可行域的非支配集合即定义为帕累托最优集合（Pareto-optimal set）



1~7号解都属于该问题的帕累托最优集合

所有帕累托最优解对应的目标函数值即构成帕累托最优前沿（Pareto-optimal front）

# 支配与帕累托最优

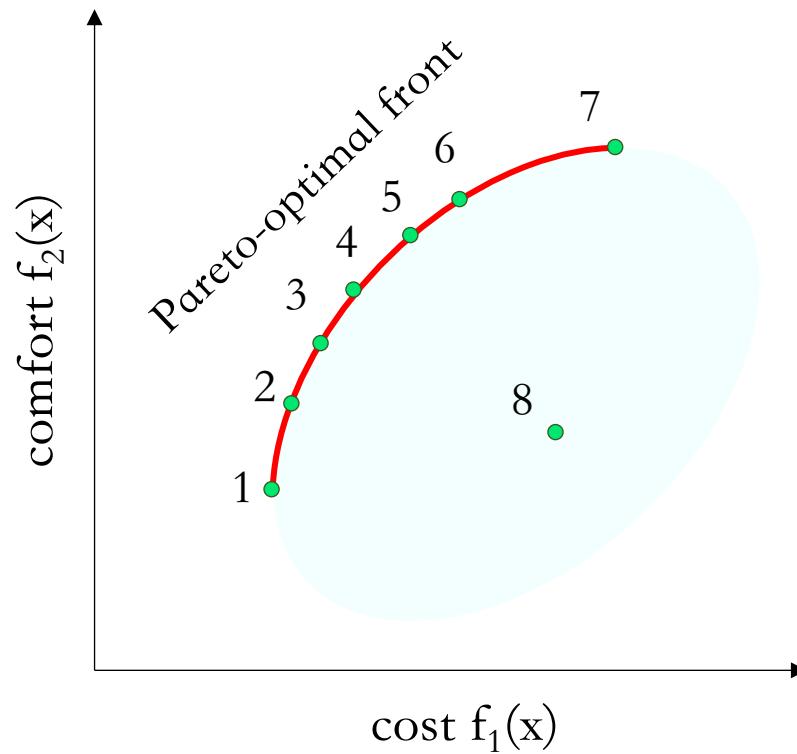
---

主要概念：

- 支配、被支配
- 非支配集合
- 帕累托最优解、帕累托最优集合
- 帕累托最优前沿

# 多目标优化的目的

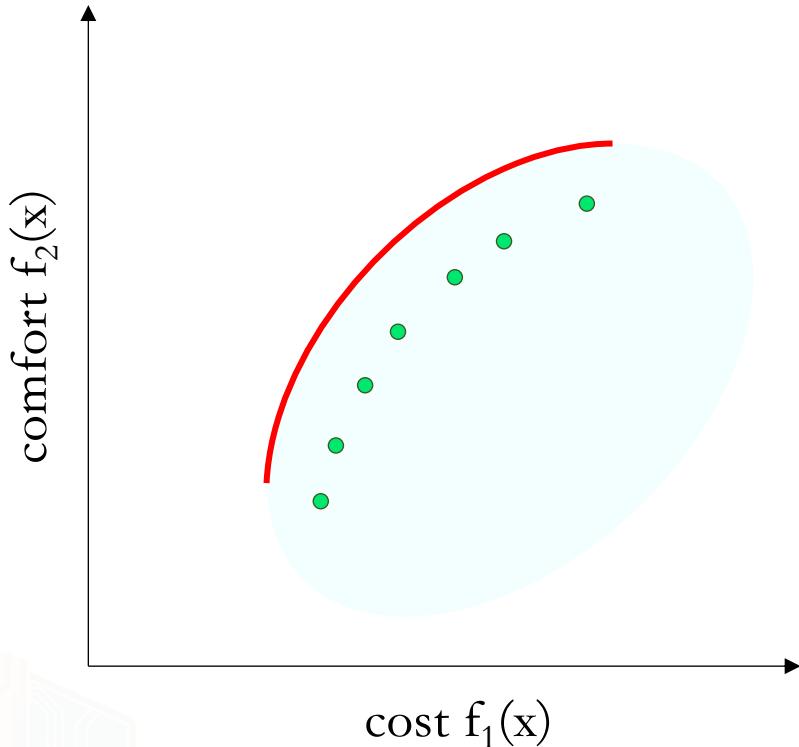
目的：找到帕累托最优集合



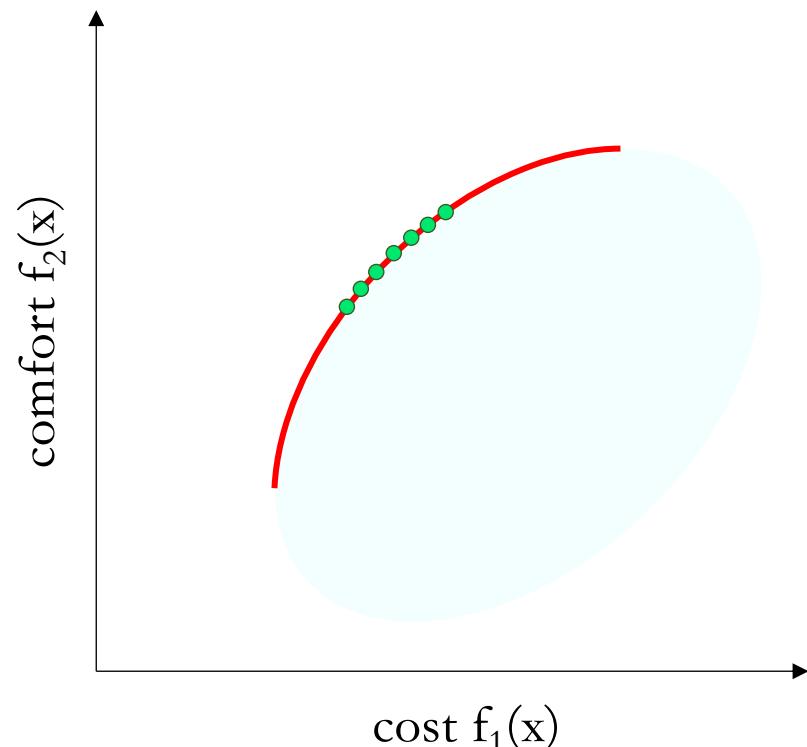
给定一个多目标优化问题，我们需要找到若干帕累托最优解，如1~7号解

# 多目标优化的目的

**关键1：**算法找到的解的目标函数值  
要趋近于帕累托最优前沿



**关键2：**算法找到的解需要尽可能分散



# 多目标优化：经典算法

Part1 如何从复杂度的角度衡量算法好坏？

Part2 针对一些有特定形式的最优化问题，如何找到最优解？

Part3 针对较复杂的最优化问题（如NP难问题），如何找到较好解？

Part4 针对较复杂的多目标最优化问题，如何找到较好解？

多目标优化问题、NSGA-II算法

# 加权求和法

加权求和法 (weighted sum method)

多目标优化

$$\min f_1(\mathbf{x})$$

$$\min f_2(\mathbf{x})$$

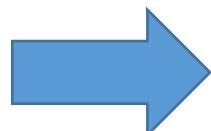
:

$$\min f_M(\mathbf{x})$$

$$\text{s.t. } g_i(\mathbf{x}) \leq 0, i = 1, \dots, m,$$

$$h_i(\mathbf{x}) = 0, j = 1, \dots, p,$$

$$\text{var. } \mathbf{x} \in \mathcal{D}.$$



$$\min \sum_{m=1}^M w_m f_m(\mathbf{x})$$

$$\text{s.t. } g_i(\mathbf{x}) \leq 0, i = 1, \dots, m,$$

$$h_i(\mathbf{x}) = 0, j = 1, \dots, p,$$

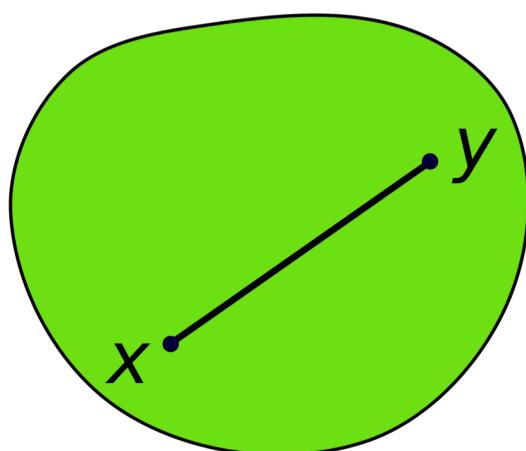
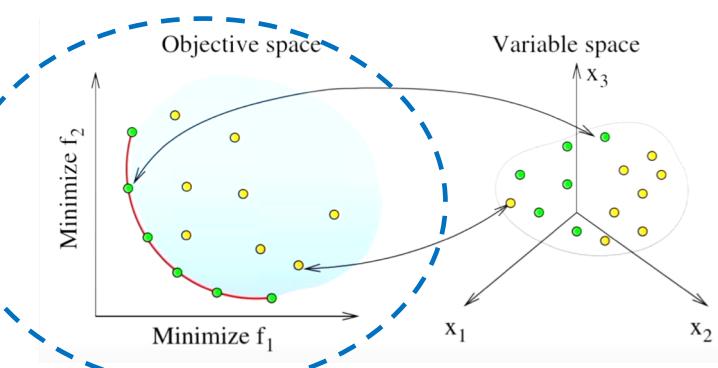
$$\text{var. } \mathbf{x} \in \mathcal{D}.$$

人为调整w向量，得到不同的解

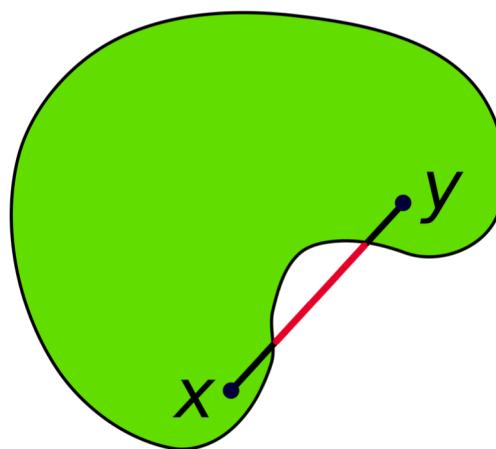
# 加权求和法

讨论两种情况：

- (1) 所有可行解对应的目标函数值的区域是凸集 (convex set)
- (2) 所有可行解对应的目标函数值的区域是非凸集 (non-convex set)



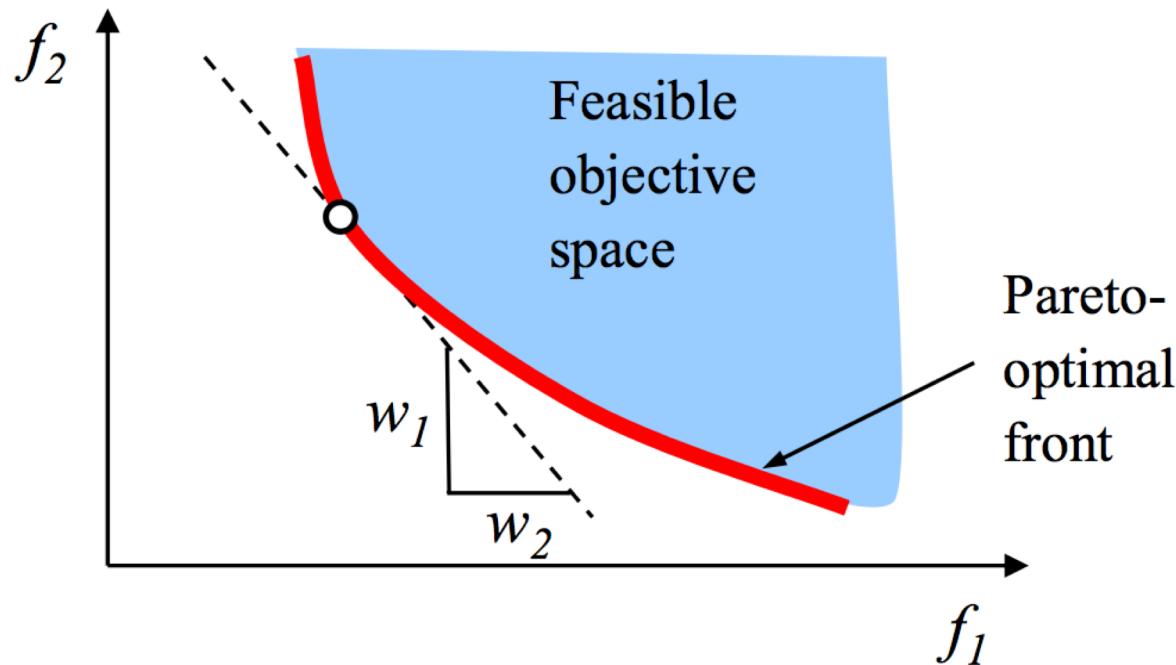
凸集



非凸集

# 加权求和法

情况一：所有可行解对应的目标函数值的区域是凸集（convex set）



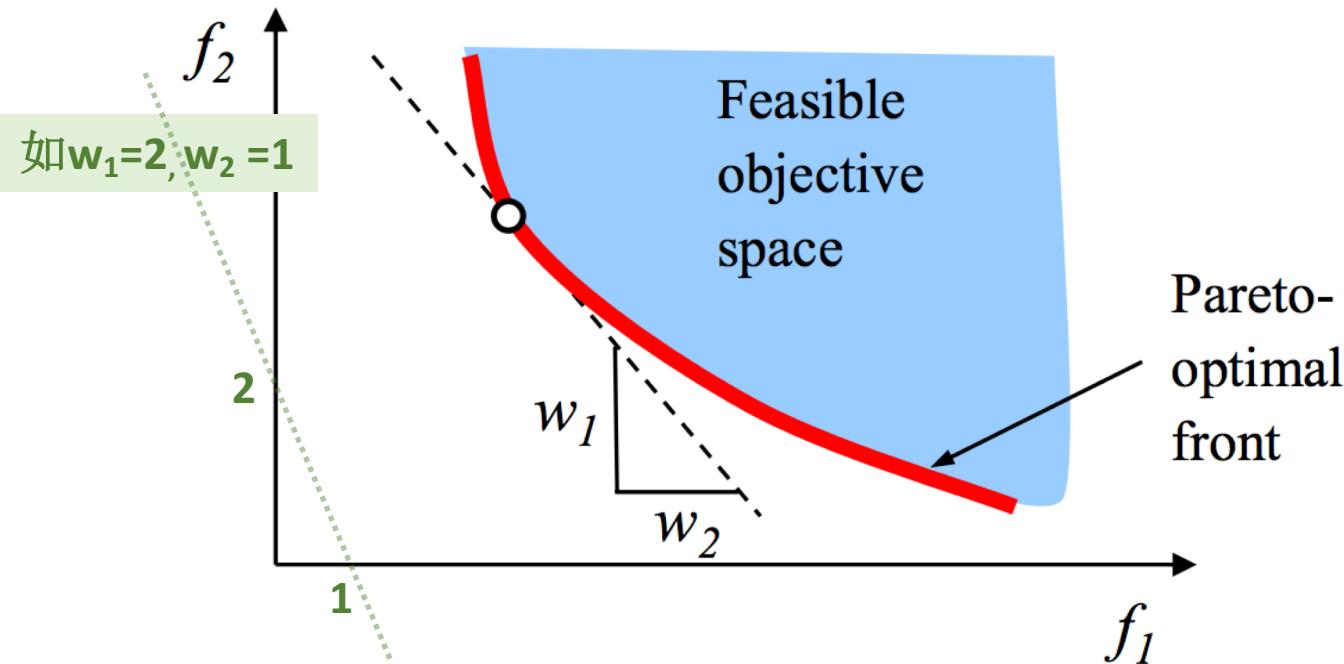
转化后的目标函数  $w_1 f_1 + w_2 f_2$

当确定( $w_1, w_2$ )后，通过加权求和法求到的最优解的( $f_1, f_2$ )位置在哪儿？

画“等值线”

# 加权求和法

情况一：所有可行解对应的目标函数值的区域是凸集（convex set）



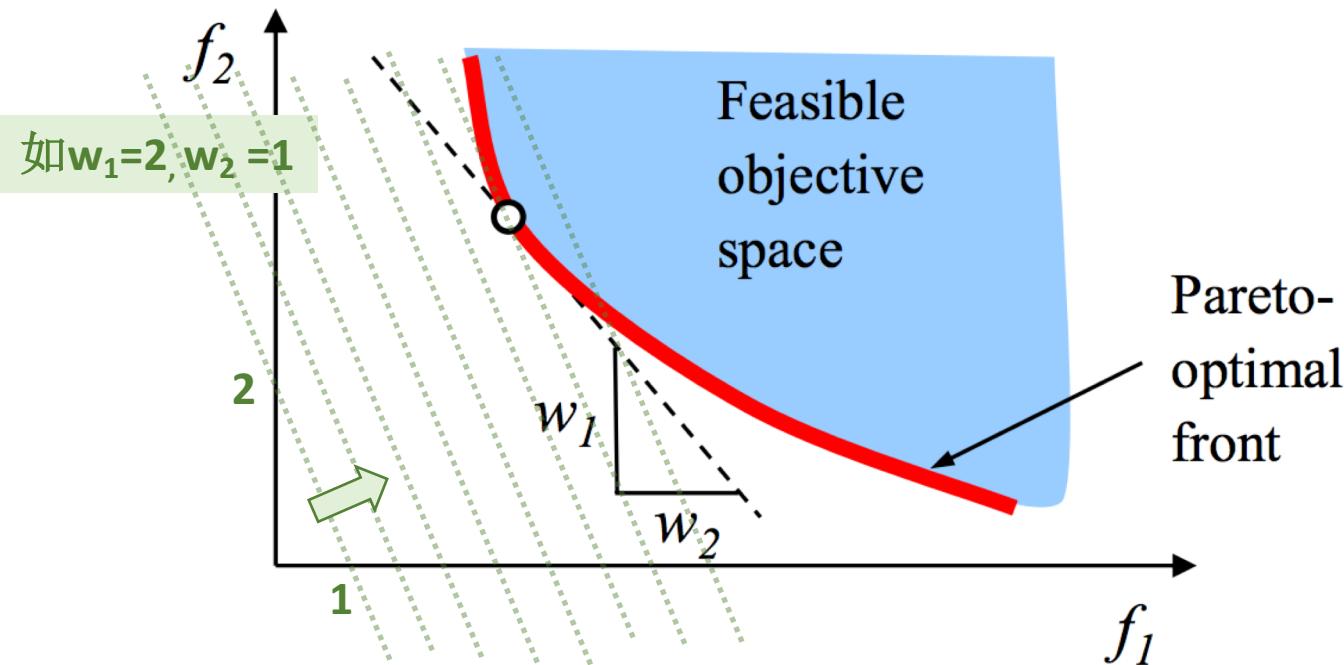
转化后的目标函数  $w_1f_1+w_2f_2$

当确定( $w_1, w_2$ )后，通过加权求和法求到的最优解的( $f_1, f_2$ )位置在哪儿？

画“等值线”

# 加权求和法

情况一：所有可行解对应的目标函数值的区域是凸集（convex set）



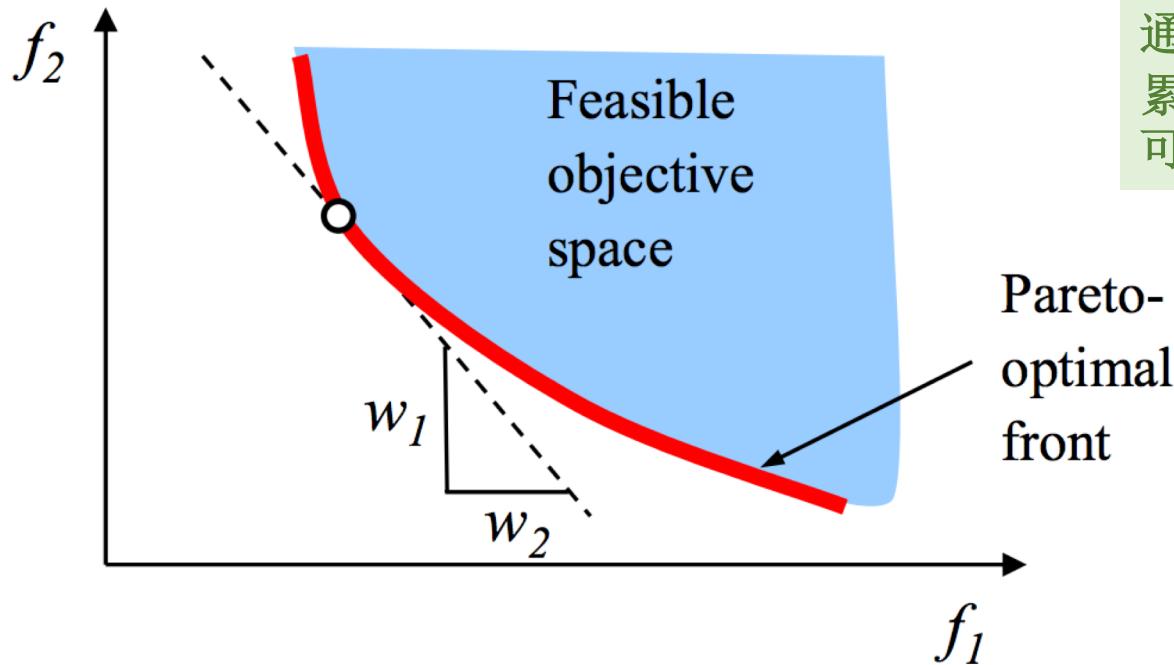
转化后的目标函数  $w_1f_1+w_2f_2$

当确定( $w_1, w_2$ )后，通过加权求和法求到的最优解的( $f_1, f_2$ )位置在哪儿？

画“等值线”

# 加权求和法

情况一：所有可行解对应的目标函数值的区域是凸集（convex set）



通过调整( $w_1, w_2$ )可以得到帕累托前沿上所有的点对应的可行解

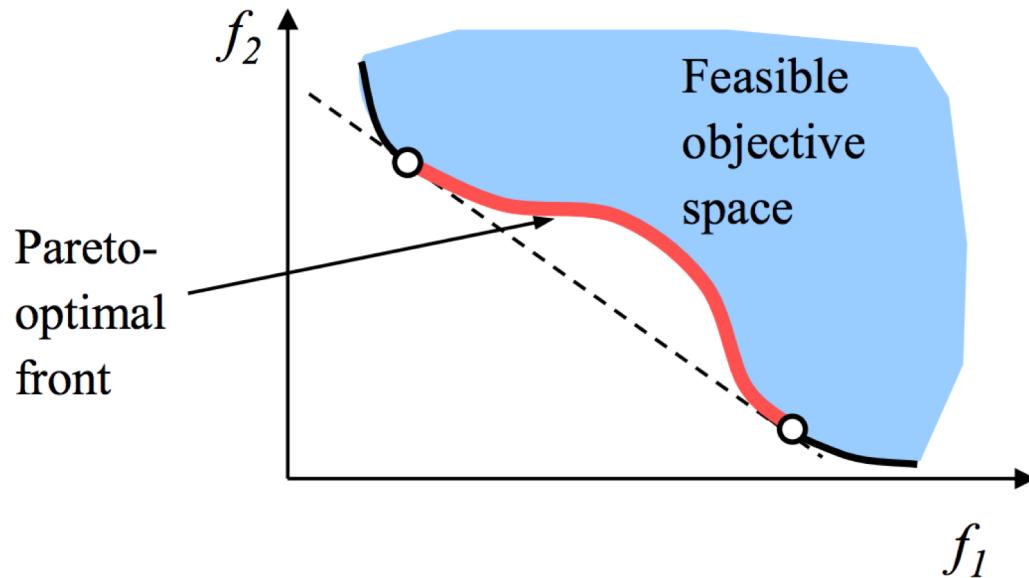
转化后的目标函数  $w_1f_1 + w_2f_2$

当确定( $w_1, w_2$ )后，通过加权求和法求到的最优解的( $f_1, f_2$ )位置在哪儿？

画“等值线”

# 加权求和法

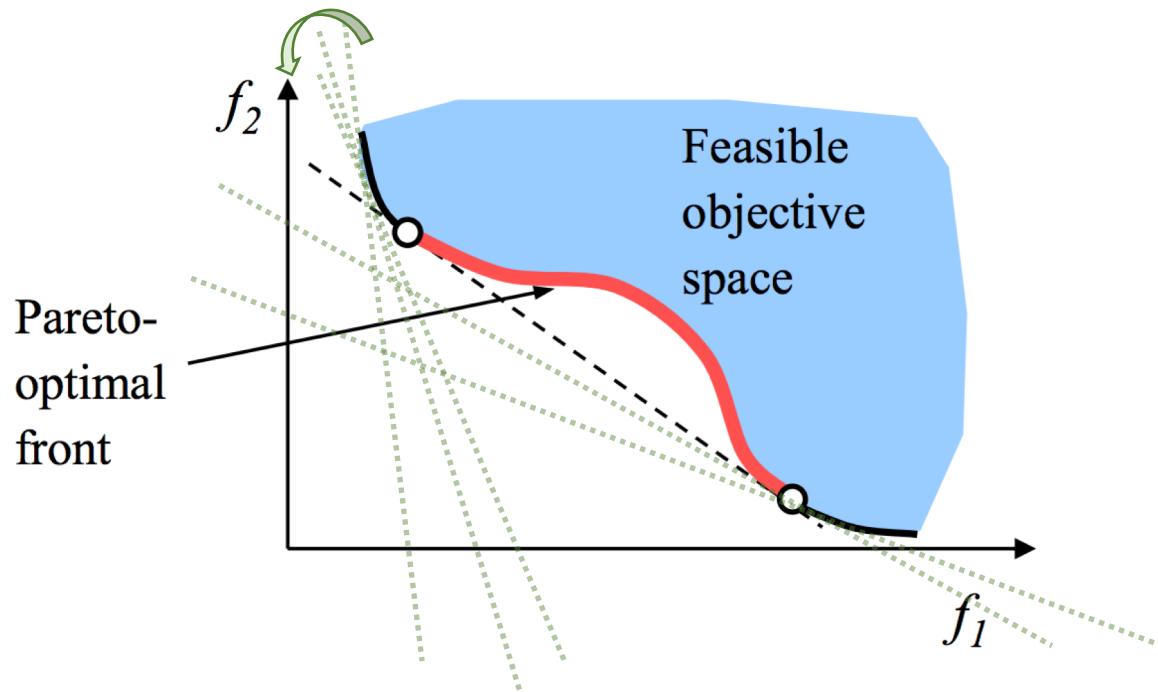
情况二：所有可行解对应的目标函数值的区域是**非凸集**（non-convex set）



无论如何设置 $w_1$ 和 $w_2$ ，标红的帕累托最优解无法取到

# 加权求和法

情况二：所有可行解对应的目标函数值的区域是**非凸集**（non-convex set）

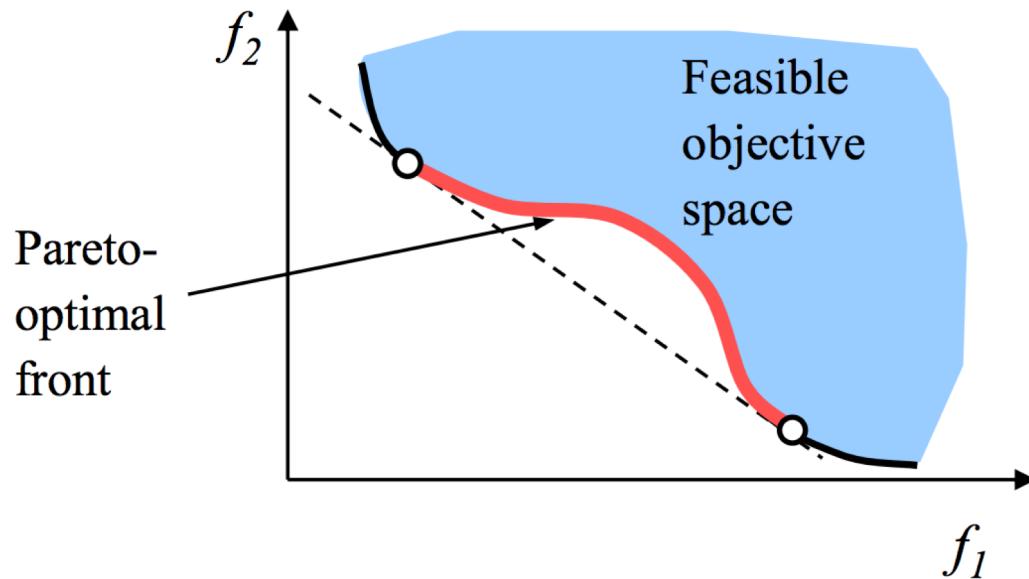


无论如何设置 $w_1$ 和 $w_2$ ，标红的帕累托最优解无法取到

可能漏掉部分帕累托最优解（因为 $w_1f_1+w_2f_2$ 限定了用线性关系比较各个解）

# 加权求和法

情况二：所有可行解对应的目标函数值的区域是**非凸集**（non-convex set）



加权求和法的缺陷：可能无法求得部分帕累托最优解

# 多目标优化：智能优化算法

Part1 如何从复杂度的角度衡量算法好坏？

Part2 针对一些有特定形式的最优化问题，如何找到最优解？

Part3 针对较复杂的最优化问题（如NP难问题），如何找到较好解？

Part4 针对较复杂的多目标最优化问题，如何找到较好解？

多目标优化问题、**NSGA-II算法**

# 智能优化算法

---

遗传算法等许多智能优化算法的基本步骤：

准备若干初始解 ⇒ 利用交叉、突变等产生新解 ⇒ 筛选出下一代的解 ⇒ ...

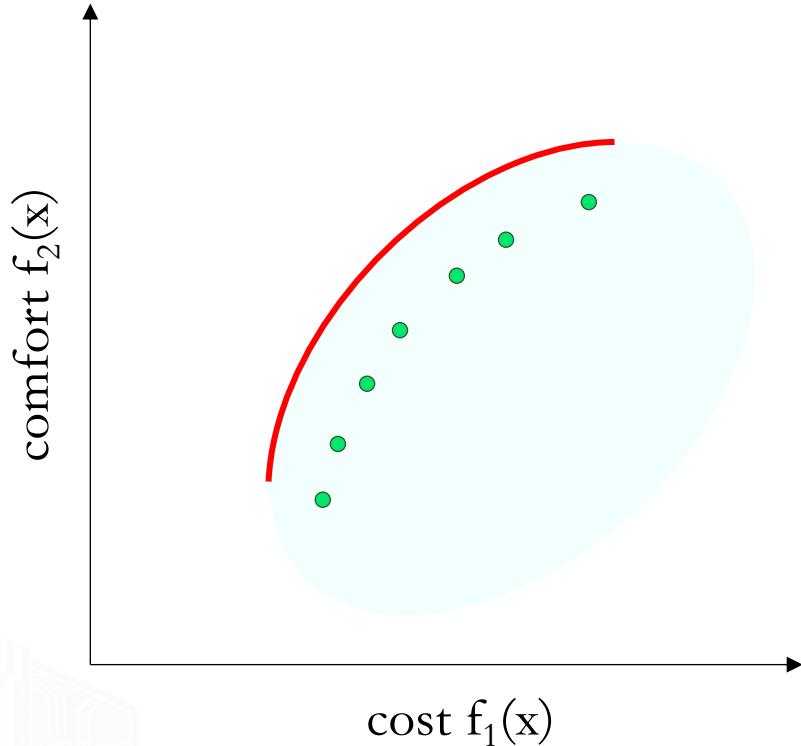
智能优化算法求解多目标优化问题的天然优势：自动生成多个解

需要解决的问题：如何对解进行筛选？

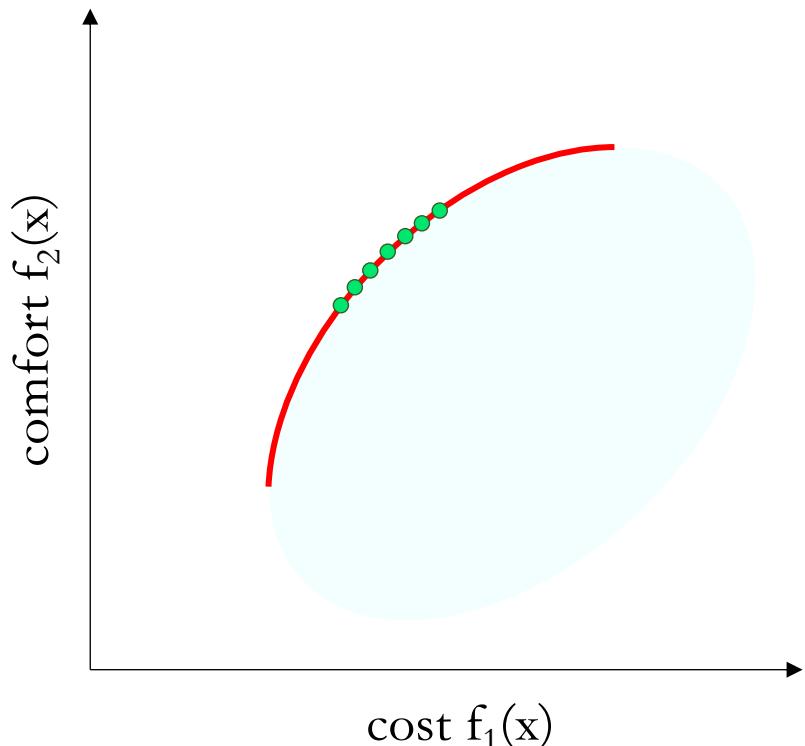
# 智能优化算法

每一次迭代中筛选解的两大原则：

**关键1：** 算法找到的解的目标函数值  
要趋近于帕累托最优前沿



**关键2：** 算法找到的解需要尽可能分散



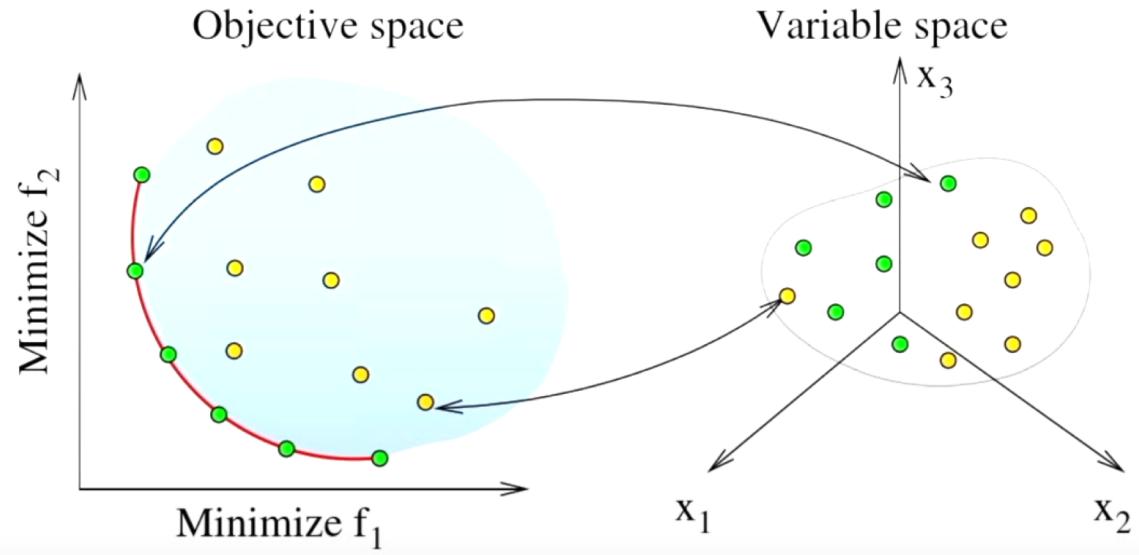
如何衡量哪些解的目标函数值更靠近帕累托最优前沿？ 如何保证筛选的解足够分散？

# 解的筛选之一：排序

如何衡量哪些解的目标函数值更靠近帕累托最优前沿？

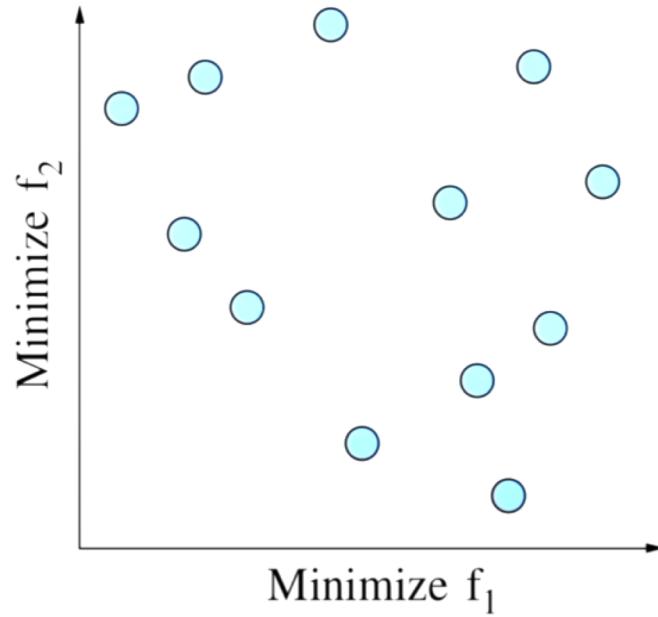
有多种方法：

- (1) 支配深度 (Dominance depth)
- (2) 支配次序 (Dominance rank)
- (3) 支配数目 (Dominance count)

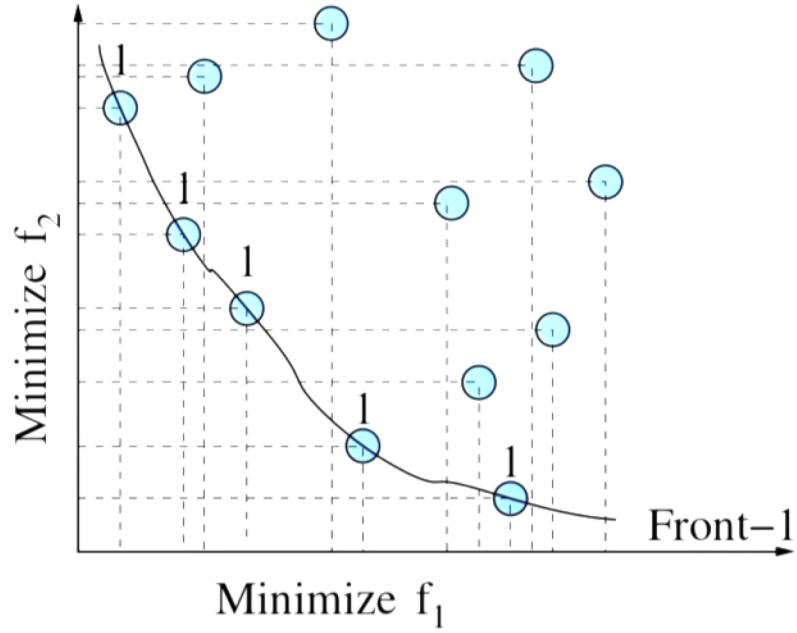


# 解的筛选之一：排序

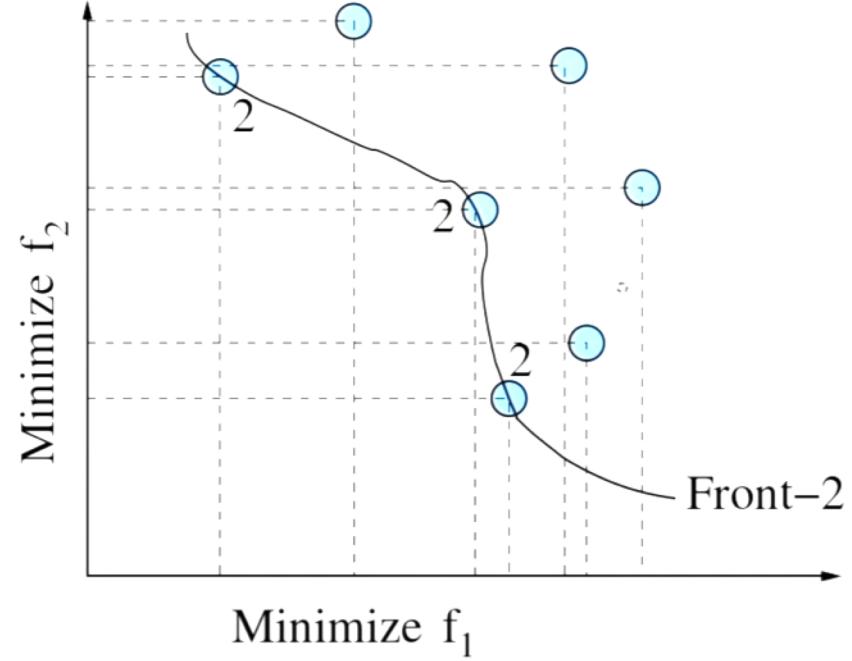
利用支配深度（Dominance depth）对解进行排序



# 解的筛选之一：排序



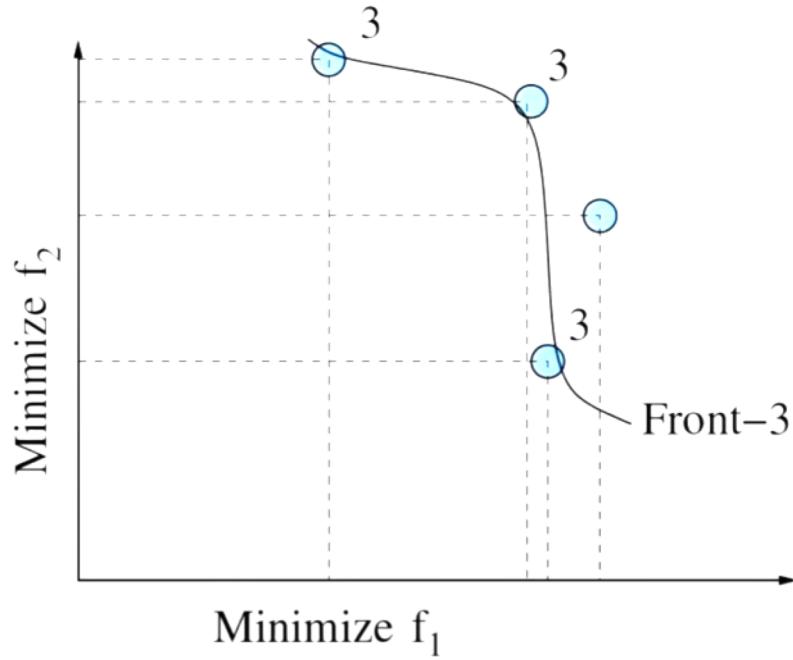
第一步：找到帕累托最优解



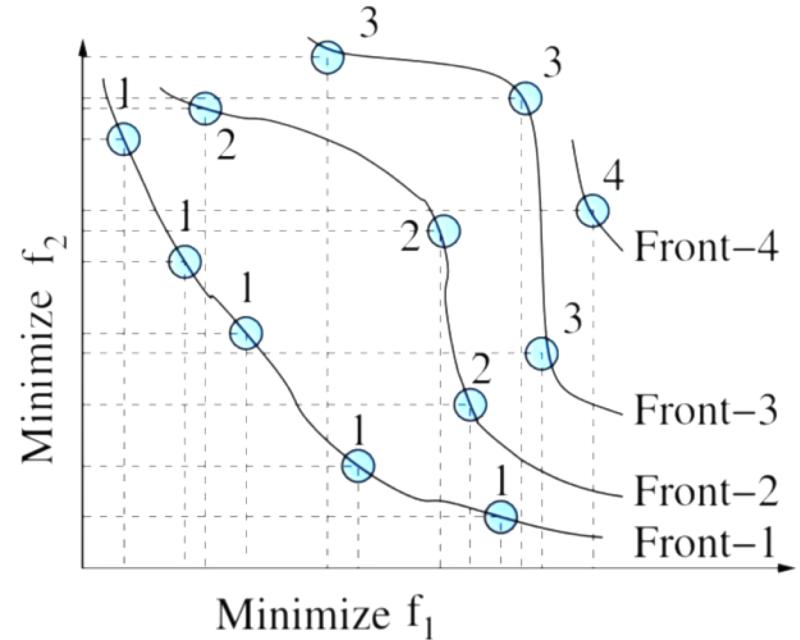
第二步：把这些解标记为front-1

（第一前沿对应的解）并移除，再  
在剩余解中找帕累托最优解

# 解的筛选之一：排序



第三步：把这些解标记为  
front-2并移除，再在剩余解中  
找帕累托最优解

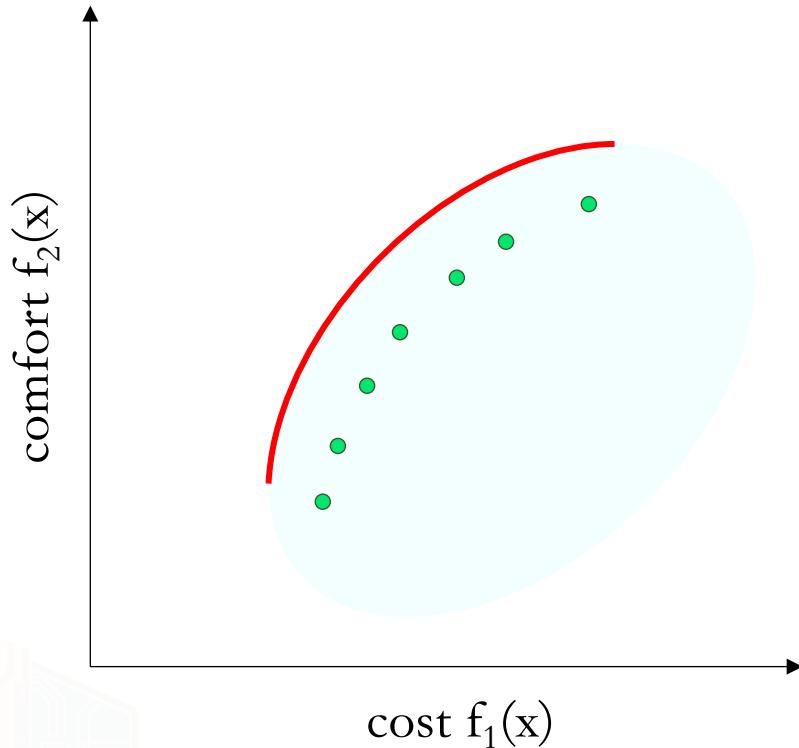


第四步：把这些解标记为  
front-3并移除，再在剩余解中  
找帕累托最优解

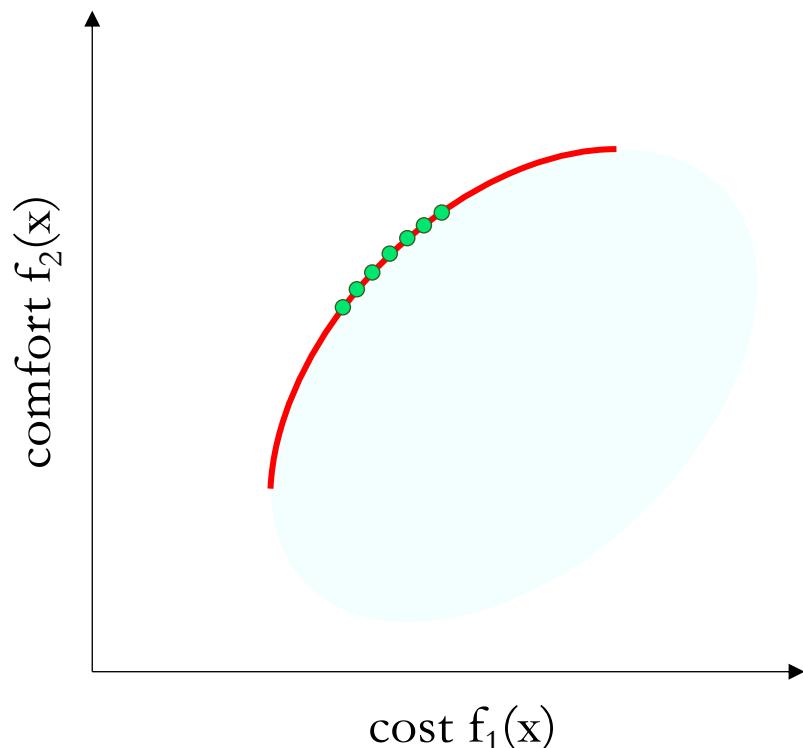
# 解的筛选之一：排序

每一次迭代中筛选解的两大原则：

**关键1：**算法找到的解的目标函数值要趋近于帕累托最优前沿



**关键2：**算法找到的解需要尽可能分散



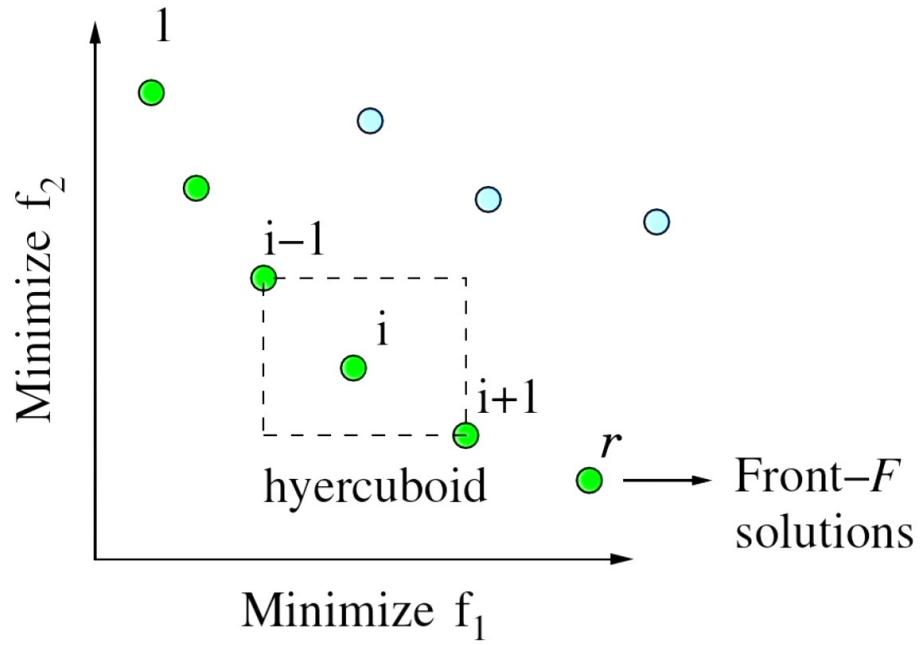
如何衡量哪些解的目标函数值更靠近帕累托最优前沿？ 如何保证筛选的解足够分散？

# 解的筛选之二：多样性

如何保证筛选的解足够分散？

有多种方法：

基于拥挤距离进行筛选



每一条前沿的点之间进行比较

# 解的筛选之二：多样性

---

为每个解计算一个 $d$ 值，该值越大说明该解离其余解越远，越需要保留

---

## Algorithm Crowding distance ( $F$ )

---

```
1:  $r = |F|$ 
2: for each  $i \in F$ , set  $d_i = 0$ 
3: for each objective  $m$  do
4:    $F = \text{sort}(F, m)$ 
5:    $d_1 = d_r = \infty$ 
6:   for  $i = 2$  to  $(r - 1)$  do
7:      $d_i = d_i + \frac{|f_m(i+1) - f_m(i-1)|}{f_m^{\max} - f_m^{\min}}$ 
8:   end for
9: end for
```

---

# 解的筛选之二：多样性

为每个解计算一个 $d$ 值，该值越大说明该解离其余解越远，越需要保留

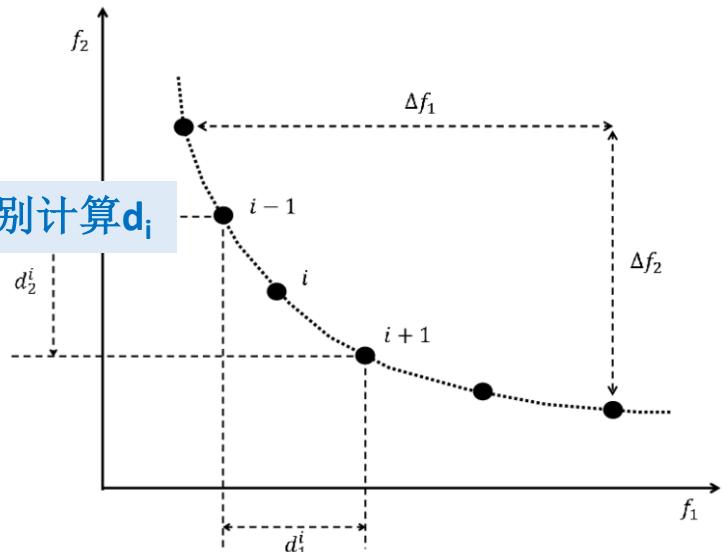
---

## Algorithm Crowding distance ( $F$ )

---

```
1:  $r = |F|$ 
2: for each  $i \in F$ , set  $d_i = 0$  为每一条前沿上的点i分别计算 $d_i$ 
3: for each objective  $m$  do
4:    $F = \text{sort}(F, m)$ 
5:    $d_1 = d_r = \infty$ 
6:   for  $i = 2$  to  $(r - 1)$  do
7:      $d_i = d_i + \frac{|f_m(i+1) - f_m(i-1)|}{f_m^{\max} - f_m^{\min}}$ 
8:   end for
9: end for
```

---



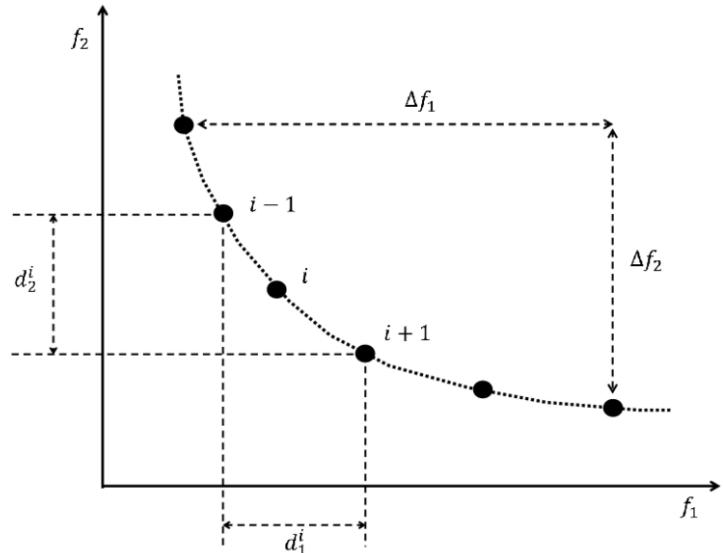
# 解的筛选之二：多样性

为每个解计算一个 $d$ 值，该值越大说明该解离其余解越远，越需要保留

## Algorithm Crowding distance ( $F$ )

```
1:  $r = |F|$ 
2: for each  $i \in F$ , set  $d_i = 0$ 
3: for each objective  $m$  do
4:    $F = \text{sort}(F, m)$ 
5:    $d_1 = d_r = \infty$ 
6:   for  $i = 2$  to  $(r - 1)$  do
7:      $d_i = d_i + \frac{|f_m(i+1) - f_m(i-1)|}{f_m^{\max} - f_m^{\min}}$ 
8:   end for
9: end for
```

右图中有两个目标函数，则 $d_i$ 包含两部分



# 解的筛选之二：多样性

为每个解计算一个 $d$ 值，该值越大说明该解离其余解越远，越需要保留

---

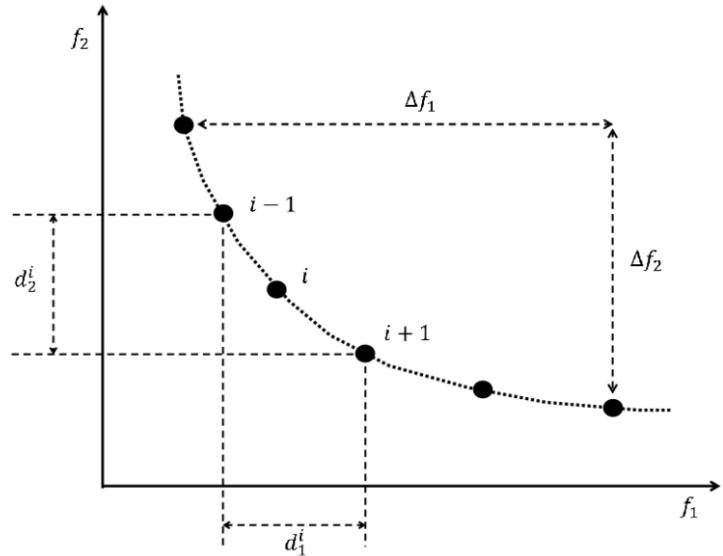
## Algorithm Crowding distance ( $F$ )

---

```
1:  $r = |F|$ 
2: for each  $i \in F$ , set  $d_i = 0$ 
3: for each objective  $m$  do
4:    $F = \text{sort}(F, m)$ 
5:    $d_1 = d_r = \infty$ 
6:   for  $i = 2$  to  $(r - 1)$  do
7:      $d_i = d_i + \frac{|f_m(i+1) - f_m(i-1)|}{f_m^{\max} - f_m^{\min}}$ 
8:   end for
9: end for
```

---

右图中有两个目标函数，则 $d_i$ 包含两部分



$$d_i = \frac{d_1^i}{\Delta f_1} + \frac{d_2^i}{\Delta f_2}$$

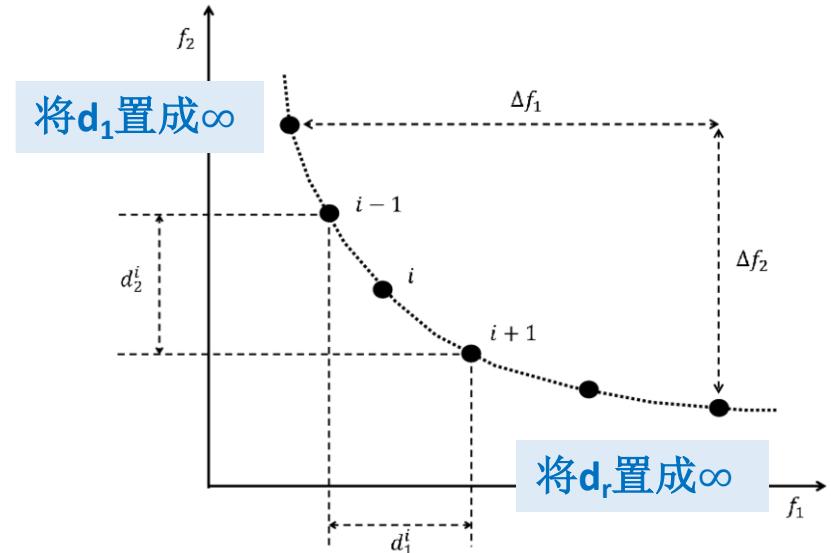
$f_1$ 维度  $f_2$ 维度

# 解的筛选之二：多样性

为每个解计算一个 $d$ 值，该值越大说明该解离其余解越远，越需要保留

## Algorithm Crowding distance ( $F$ )

```
1:  $r = |F|$ 
2: for each  $i \in F$ , set  $d_i = 0$ 
3: for each objective  $m$  do
4:    $F = \text{sort}(F, m)$ 
5:    $d_1 = d_r = \infty$ 
6:   for  $i = 2$  to  $(r - 1)$  do
7:      $d_i = d_i + \frac{|f_m(i+1) - f_m(i-1)|}{f_m^{\max} - f_m^{\min}}$ 
8:   end for
9: end for
```



$$d_i = \frac{d_1^i}{\Delta f_1} + \frac{d_2^i}{\Delta f_2}$$

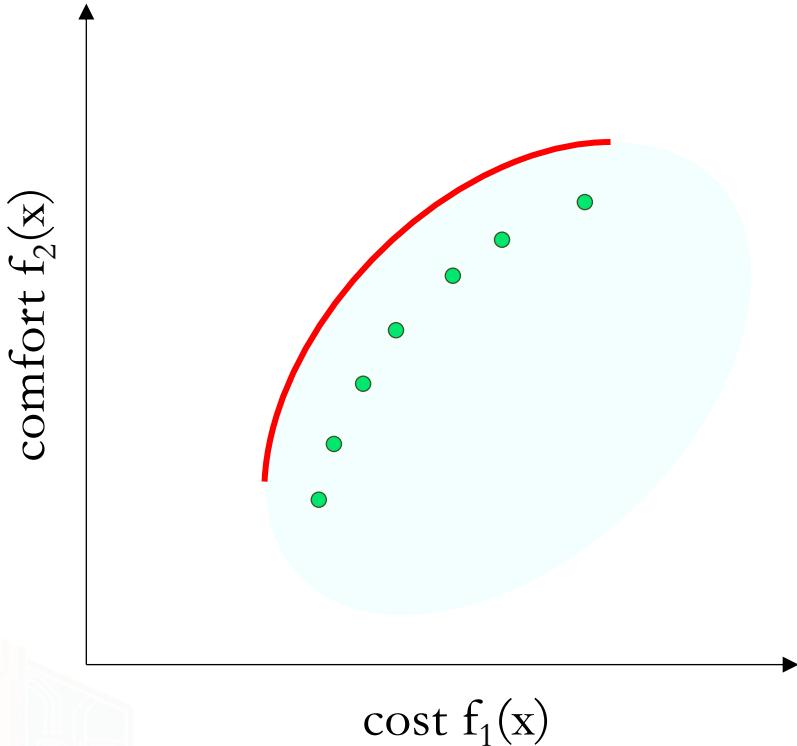
$f_1$ 维度  $f_2$ 维度

# 解的筛选之二：多样性

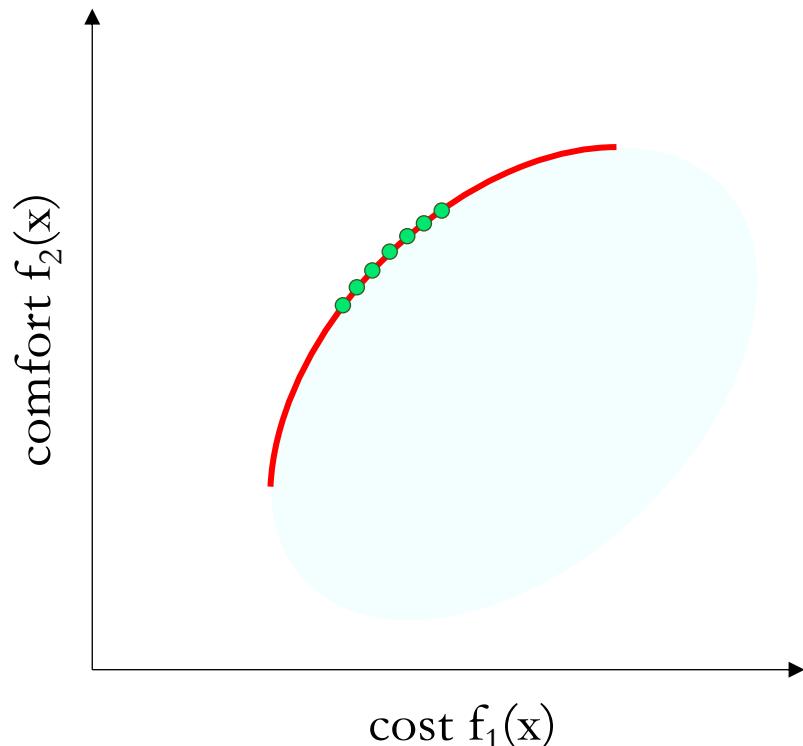
每一次迭代中筛选解的两大原则：

**关键1：**算法找到的解的目标函数值

要趋近于帕累托最优前沿



**关键2：**算法找到的解需要尽可能分散



如何衡量哪些解的目标函数值更靠近帕累托最优前沿？ 如何保证筛选的解足够分散？

# NSGA-II

---

## NSGA-II: Elitist Non-Dominated Sorting Genetic Algorithm

### A fast and elitist multiobjective genetic algorithm: NSGA-II

[K Deb, A Pratap, S Agarwal... - IEEE transactions on ..., 2002 - ieeexplore.ieee.org](#)

... 2) Lack of elitism: Recent results [25], [18] show that elitism can **speed** up the performance of the GA significantly, which also can help preventing the loss of good solutions once they are found ... Page 4. DEB et al.: A **FAST AND ELITIST MULTIOBJECTIVE GA: NSGA-II** ...

☆ 35504 Cited by 35504 Related articles All 42 versions

### A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II

[K Deb, S Agrawal, A Pratap, T Meyarivan - International conference on ..., 2000 - Springer](#)

... Lack of elitism: Recent results [10,7] show clearly that elitism can **speed** up the per ... except that a better book- keeping strategy is performed to make it a **faster** algorithm ... With the properties of a **fast** non-dominated sorting procedure, an **elitist** strategy, and a parameterless approach ...

☆ 5097 Cited by 5097 Related articles All 22 versions

(1) 通过对NSGA改进而提出的

(2) 基于遗传算法

# NSGA-II

给定当前的父代解和子代解，若质量最好的解是父代解，保留其进入下一代  
根据解的非支配性质，对解进行排序（第一前沿、第二前沿...）

## NSGA-II: Elitist Non-Dominated Sorting Genetic Algorithm

### A fast and elitist multiobjective genetic algorithm: NSGA-II

K Deb, A Pratap, S Agarwal... - IEEE transactions on ..., 2002 - ieeexplore.ieee.org

... 2) Lack of elitism: Recent results [25], [18] show that elitism can **speed** up the performance of the GA significantly, which also can help preventing the loss of good solutions once they are found ... Page 4. DEB et al.: A FAST AND ELITIST MULTIOBJECTIVE GA: NSGA-II ...

☆ 29 Cited by 35504 Related articles All 42 versions

### A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II

K Deb, S Agrawal, A Pratap, T Meyarivan - International conference on ..., 2000 - Springer

... Lack of elitism: Recent results [10,7] show clearly that elitism can **speed** up the per ... except that a better book- keeping strategy is performed to make it a **faster** algorithm ... With the properties of a **fast** non-dominated sorting procedure, an **elitist** strategy, and a parameterless approach ...

☆ 29 Cited by 5097 Related articles All 22 versions

(1) 通过对NSGA改进而提出的

(2) 基于遗传算法

# NSGA-II

---

## 回顾针对单目标优化的遗传算法

### 遗传算法

- 0 初始化当前种群 $P(t)$
- 1 评估当前种群 $P(t)$
- 2 for  $t=1$  to  $\infty$ :
- 3 根据当前种群 $P(t)$ , 确定父代种群 $P_P(t)$
- 4 根据父代种群 $P_P(t)$ , 通过基因交叉生成子代种群 $P_C(t)$
- 5 对子代种群 $P_C(t)$ 进行基因突变
- 6 评估子代种群 $P_C(t)$
- 7 根据当前种群 $P(t)$ 和子代种群 $P_C(t)$ , 筛选出新的当前种群 $P(t + 1)$
- 8 检查循环终止条件, 若满足则结束算法

如果用于多目标优化, 哪些步骤需要修改?

# NSGA-II

---

针对多目标优化，需要修改的步骤

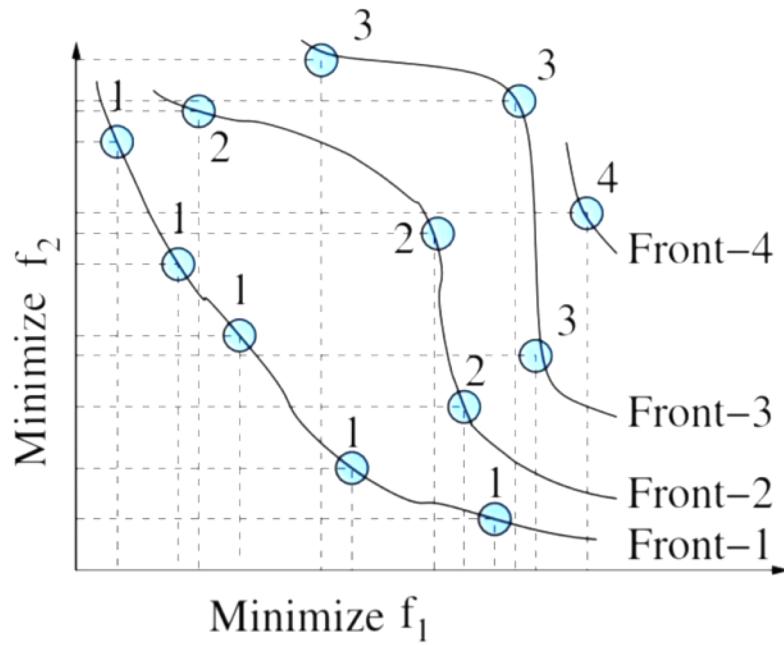
## 遗传算法

- 0 初始化当前种群 $P(t)$
- 1 评估当前种群 $P(t)$
- 2 for  $t=1$  to  $\infty$ :
- 3 根据当前种群 $P(t)$ ，确定父代种群 $P_P(t)$
- 4 根据父代种群 $P_P(t)$ ，通过基因交叉生成子代种群 $P_C(t)$
- 5 对子代种群 $P_C(t)$ 进行基因突变
- 6 评估子代种群 $P_C(t)$
- 7 根据当前种群 $P(t)$ 和子代种群 $P_C(t)$ ，筛选出新的当前种群 $P(t + 1)$
- 8 检查循环终止条件，若满足则结束算法

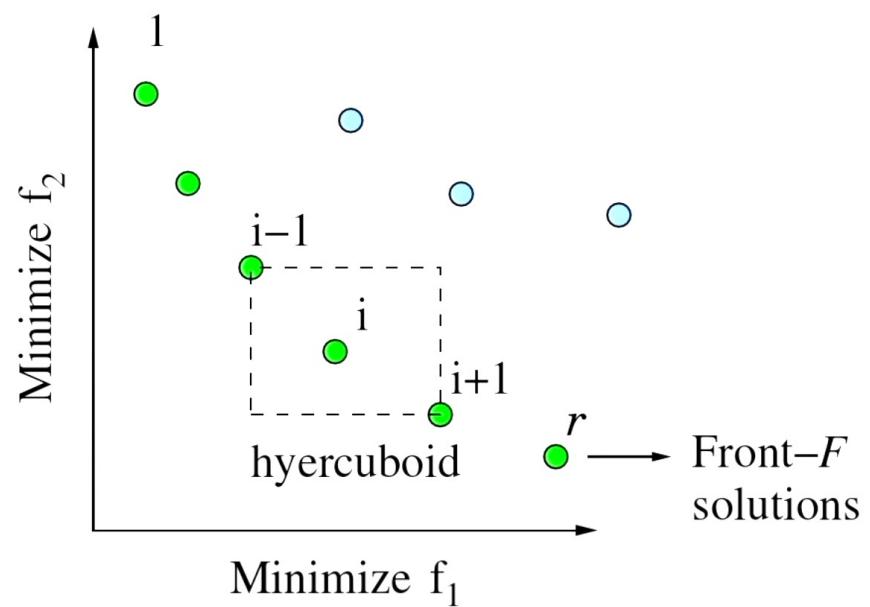
(1) 对每个解评估质量 (2) 对解的筛选

# NSGA-II

对解根据其目标函数离Pareto-optimal front远近排序



对解计算拥挤距离



根据这两方面的信息完成对解的筛选

# NSGA-II

---

论文中一个例子的运行结果

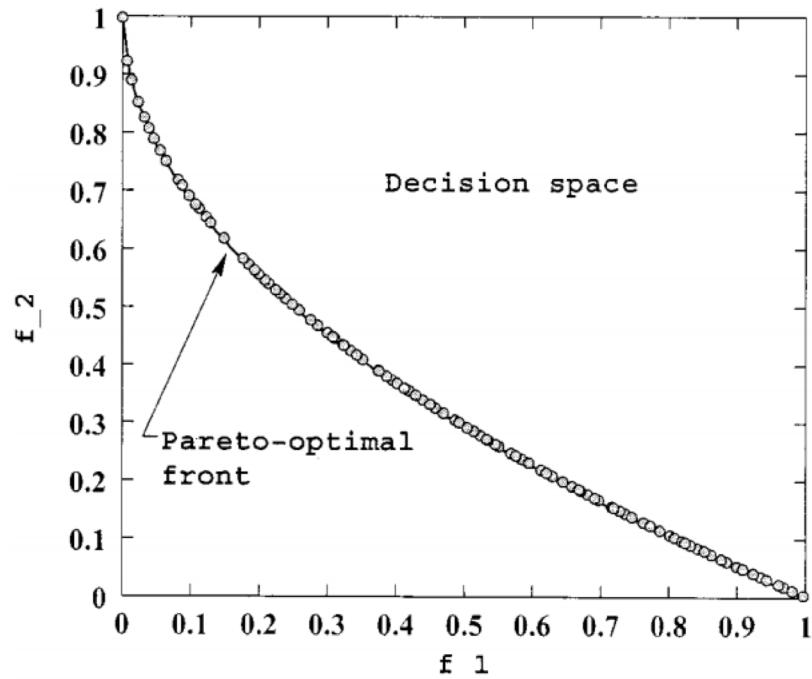


Fig. 12. Obtained nondominated solutions with NSGA-II on problem ZDT4.

# 多目标进化算法的应用

---

R. Shen, et al., “Generating Behavior-Diverse Game AIs with Evolutionary Multi-Objective Deep Reinforcement Learning,” IJCAI 2020.



进化强化学习生成风格多样AI，总有一款适合你！

<https://www.bilibili.com/video/BV16D4y1D7e2>

4分5秒-6分5秒；8分40秒-10分36秒；16分-23分40秒；23分40秒-28分50秒

# 本讲小结

---



多目标优化经典算法



多目标优化进化算法 NSGA-II

# 主要参考资料

Deepak Sharma (IIT Guwahati) <Evolutionary Computation for Single and Multi-Objective Optimization> Slides

网易伏羲 <进化强化学习生成风格多样AI，总有一款适合你！> Video

