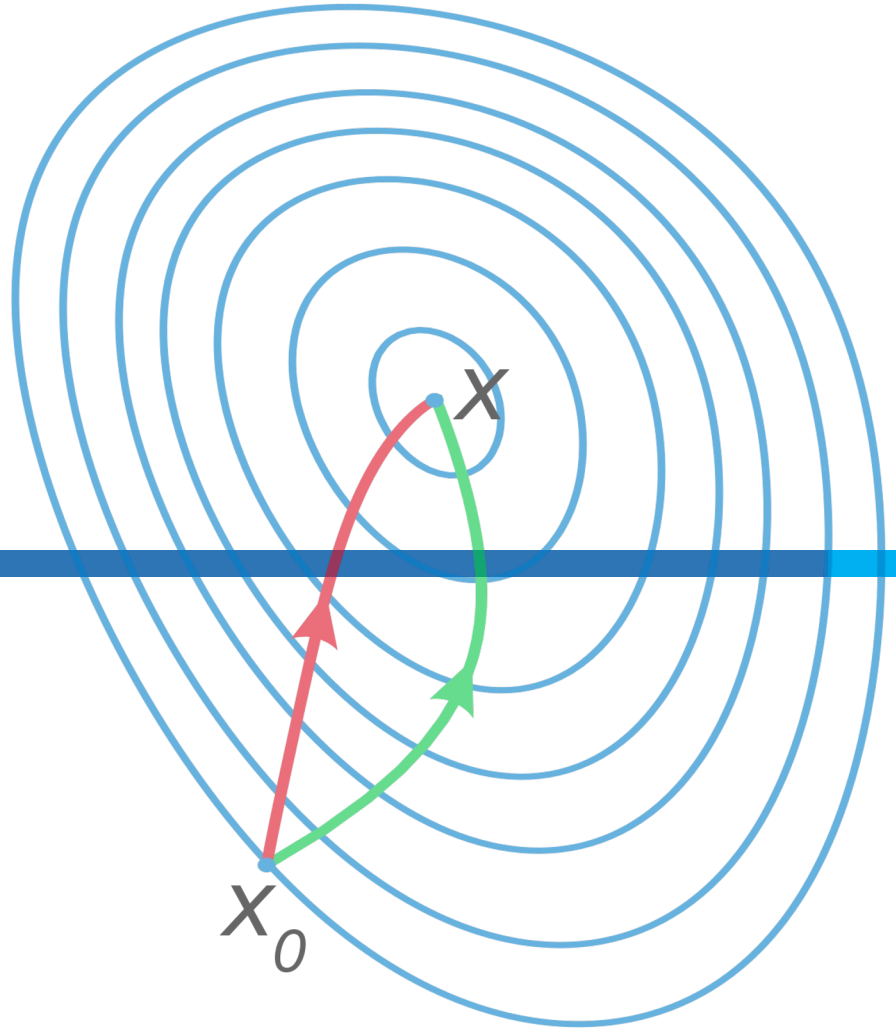
最优化方法



第八周

计算机学院
余皓然

2024/4/15

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）

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

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

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

多目标优化

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

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！

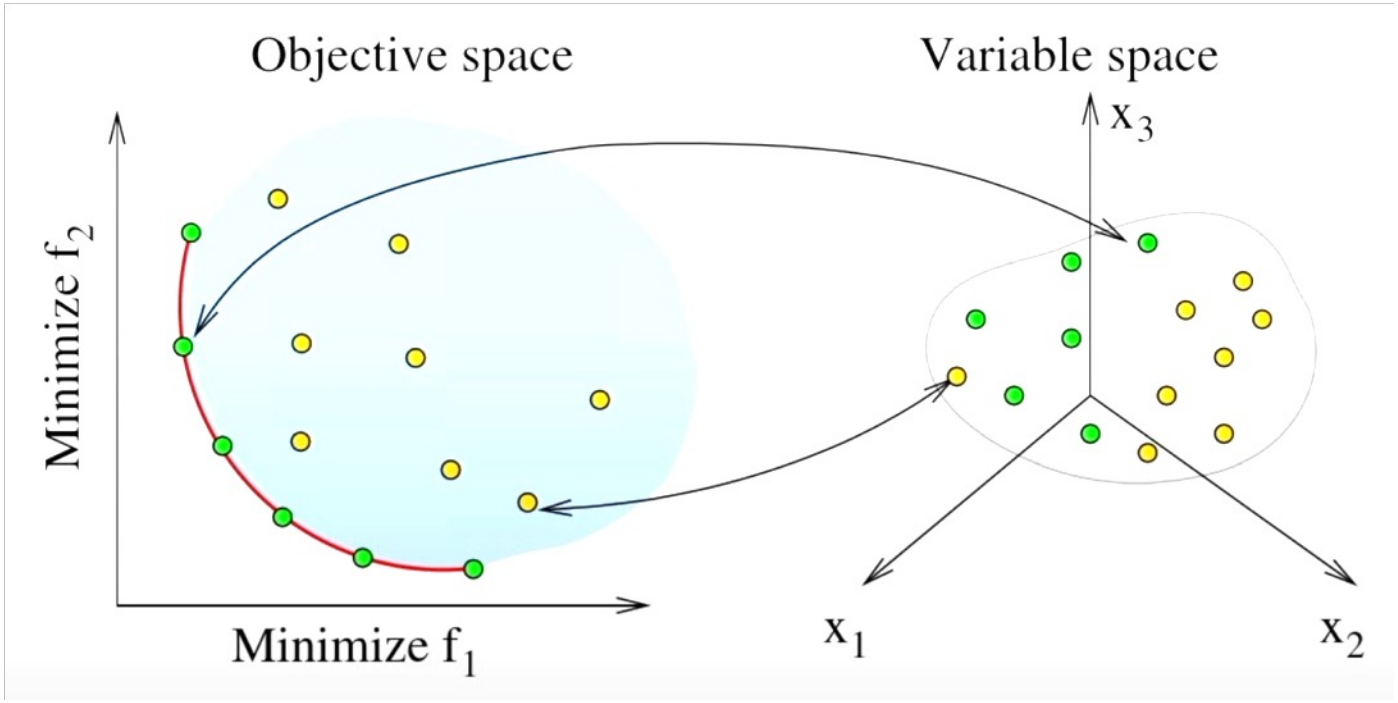
https://www.bilibili.com/video/BV16D4y1D7e2

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

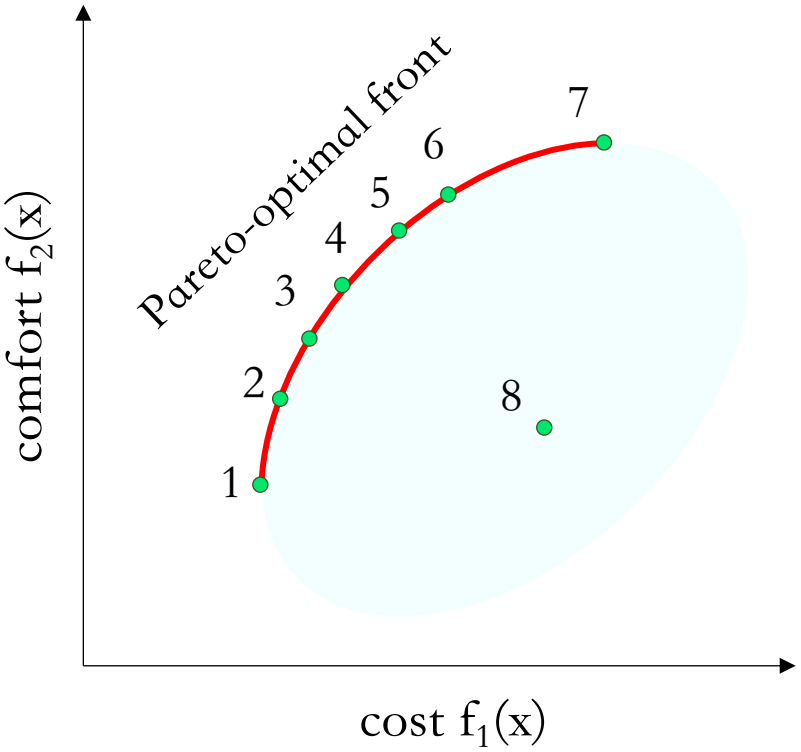
支配与帕累托最优

区分目标函数值空间和解空间



内容取自Deepak Sharma (IIT Guwahati) 课程 Evolutionary Computation for Single and Multi-Objective Optimization

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）

⇔ 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

“A和B之间无脑选A”

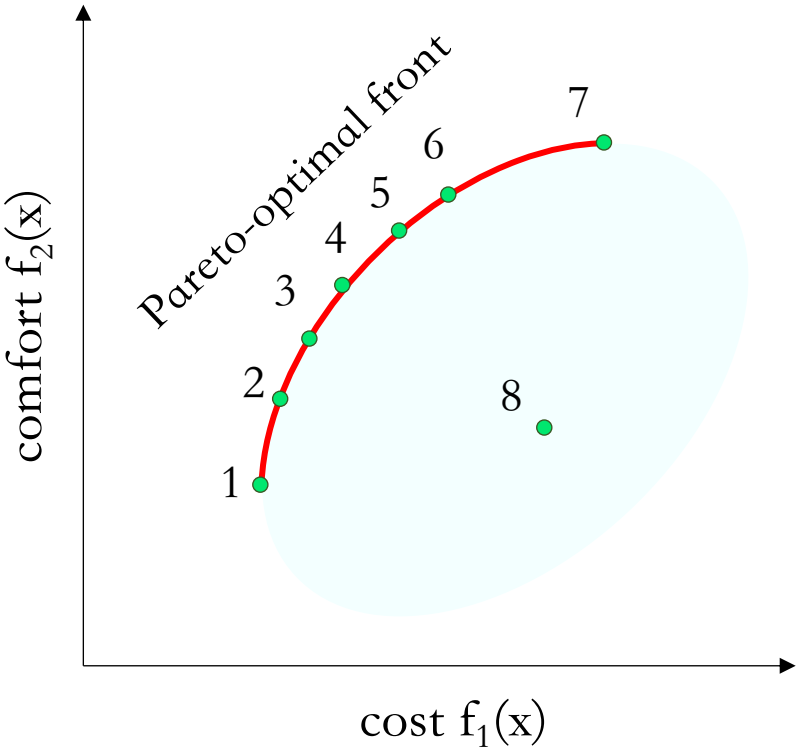
花费 3
舒适 3

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



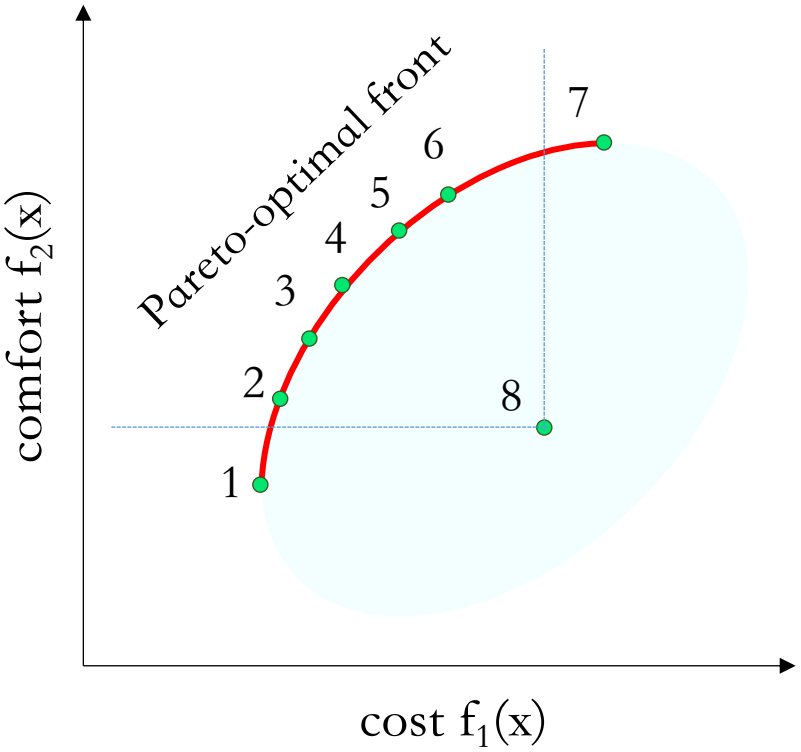
支配（Dominate）：

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

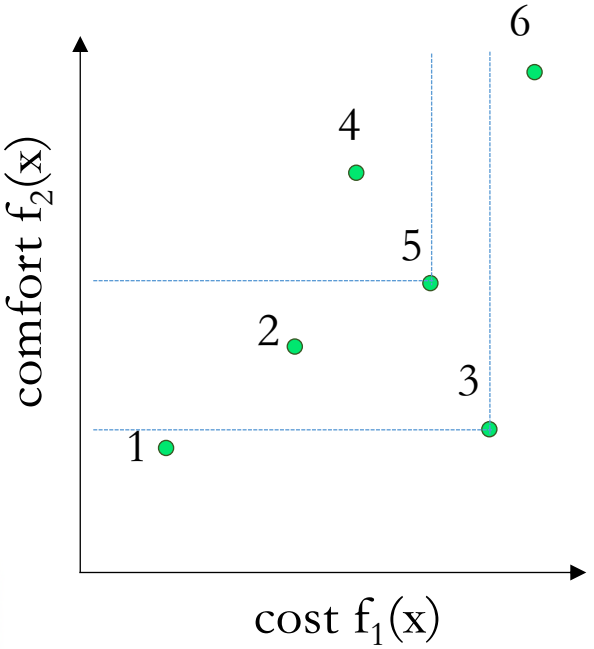


8号解被2、3、4、5、6号解支配

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。



集合 P = {1, 2, 3, 4, 5, 6}

它的非支配集合 = {1, 2, 4, 6}

内容取自Deepak Sharma (IIT Guwahati) 课程 Evolutionary Computation for Single and Multi-Objective Optimization

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。

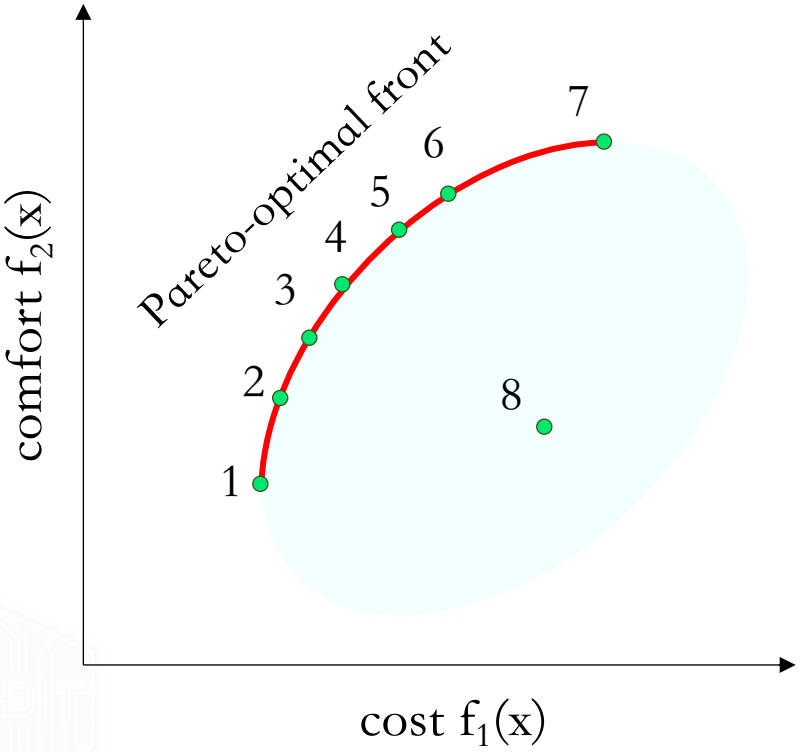
帕累托最优：

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

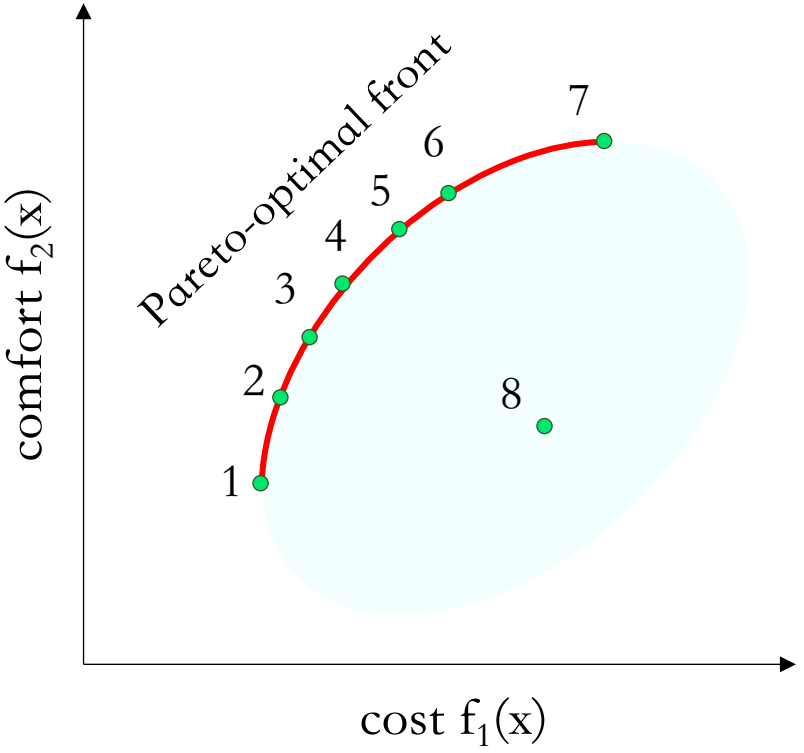
—— 非支配集合

—— 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

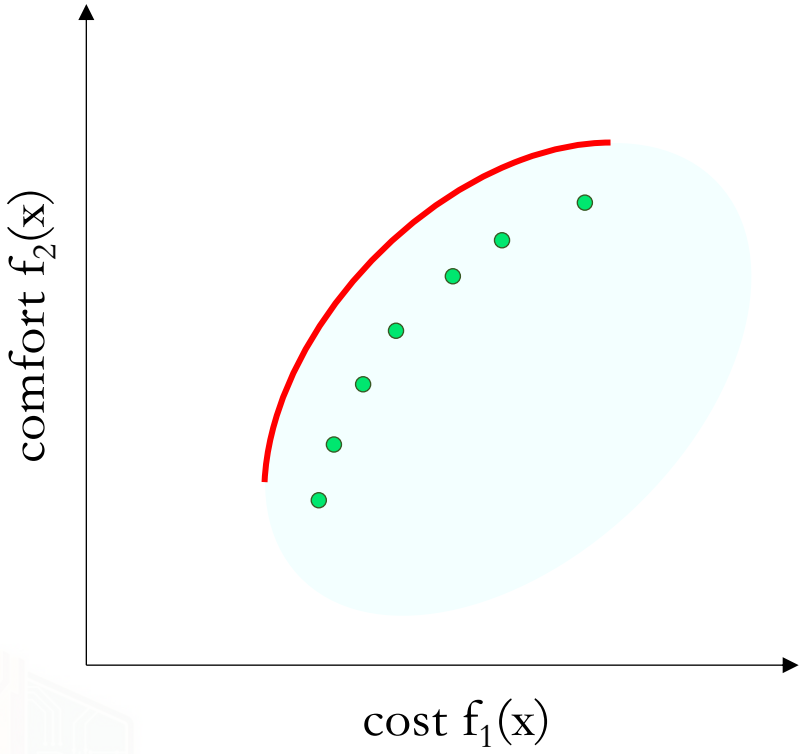


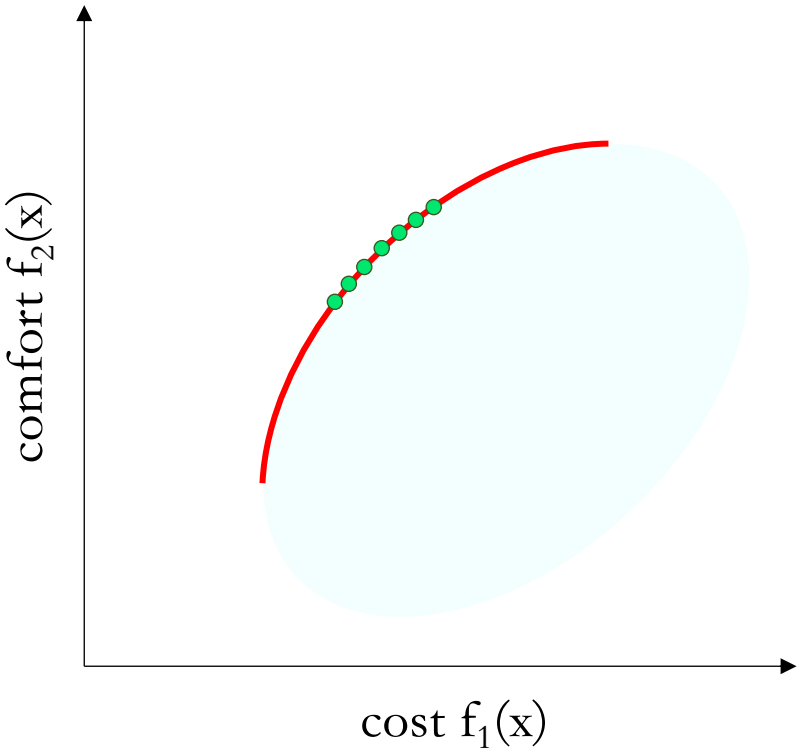
给定一个多目标优化问题，我们需要找到若干帕累托最优解，如1～7号解

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





内容取自Deepak Sharma (IIT Guwahati) 课程 Evolutionary Computation for Single and Multi-Objective Optimization

多目标优化：经典算法

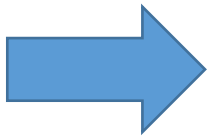
Part1 如何从复杂度的角度衡量算法好坏？
Part2 针对一些有特定形式的最优化问题，如何找到最优解？
Part3 针对较复杂的最优化问题（如NP难问题），如何找到较好解？
Part4 针对较复杂的多目标最优化问题，如何找到较好解？
多目标优化问题、NSGA-II算法

加权求和法

加权求和法（weighted sum method）

多目标优化

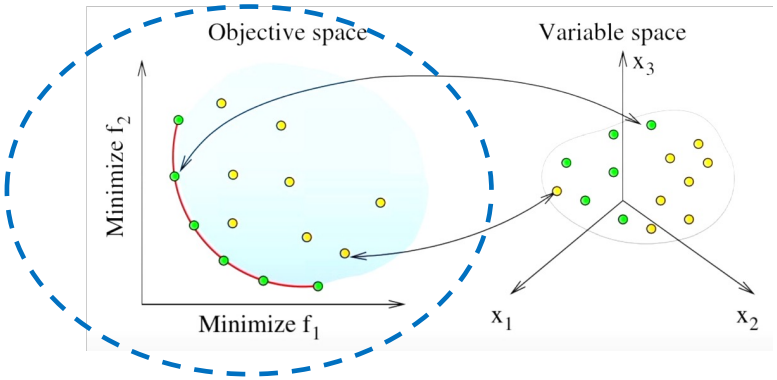
$$
\begin{aligned}
&\min \quad f_1(\boldsymbol{x})\\
&\min \quad f_2(\boldsymbol{x})\\
&\vdots\\
&\min \quad f_M(\boldsymbol{x})\\
&\text{s.t.} \quad g_i(\boldsymbol{x}) \le 0, i = 1, \dots, m,\\
&\qquad h_i(\boldsymbol{x}) = 0, j = 1, \dots, p,\\
&\text{var.} \quad \boldsymbol{x} \in \mathcal{D}.
\end{aligned}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



$$
\begin{aligned}
&\min \textstyle\sum_{m=1}^{M} w_m f_m(\boldsymbol{x})\\
&\text{s.t.} \quad g_i(\boldsymbol{x}) \le 0, i = 1, \dots, m,\\
&\qquad h_i(\boldsymbol{x}) = 0, j = 1, \dots, p,\\
&\text{var.} \quad \boldsymbol{x} \in \mathcal{D}.
\end{aligned}
$$

人为调整w向量，得到不同的解

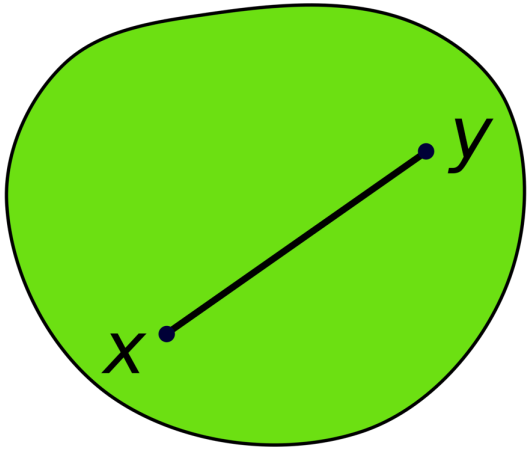
加权求和法



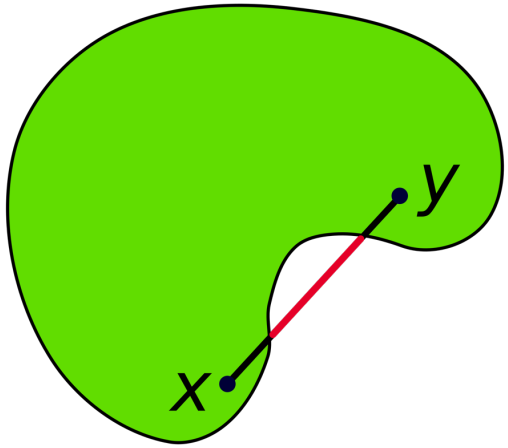
讨论两种情况：

（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）



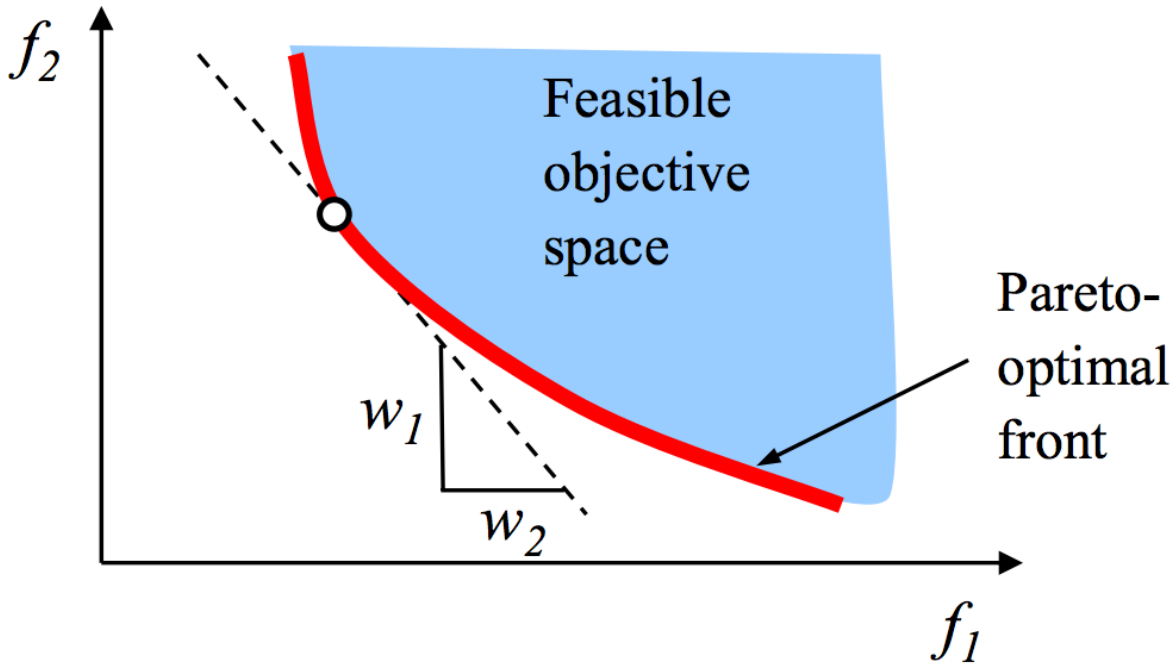
凸集



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）

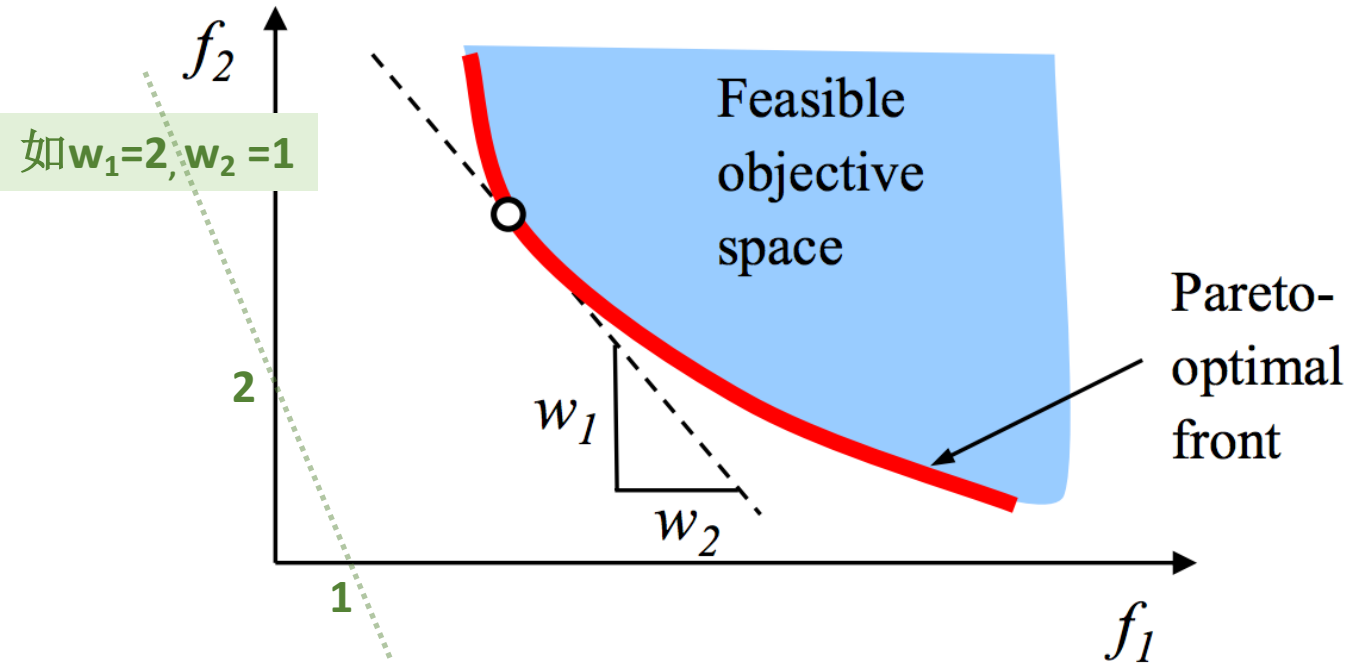


转化后的目标函数 $w_1f_1+w_2f_2$

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）

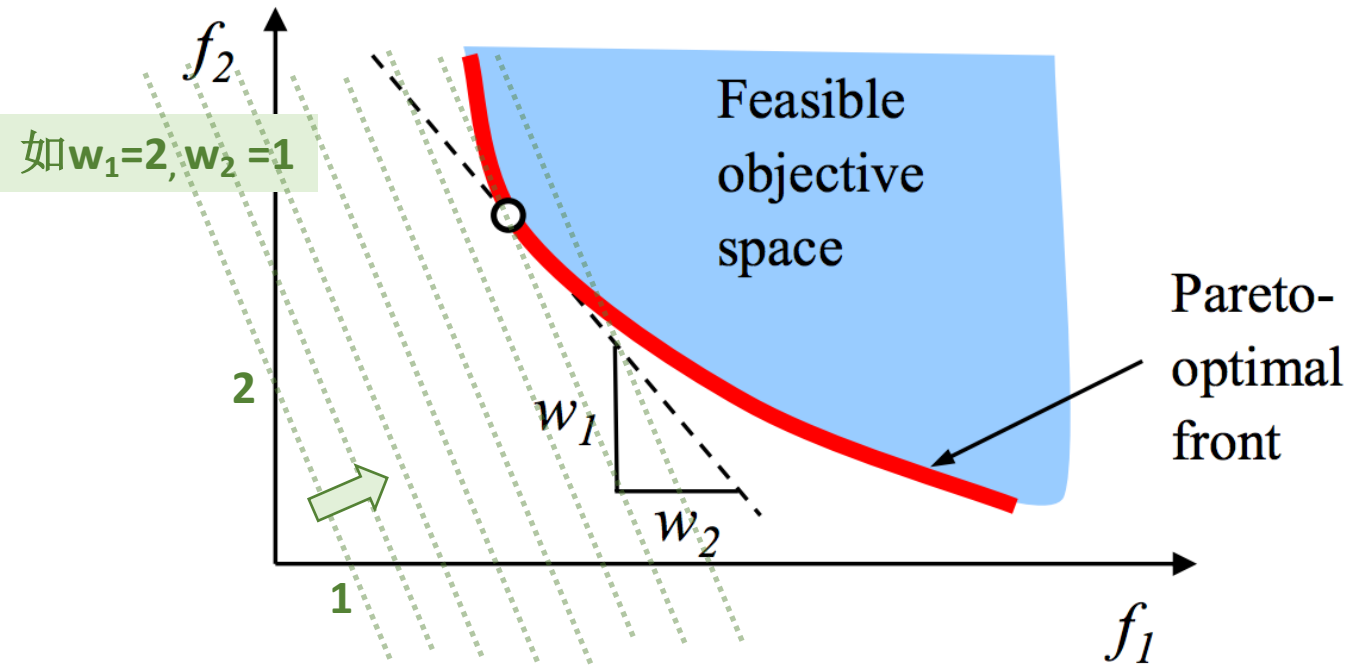


转化后的目标函数 $w_1f_1+w_2f_2$

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）

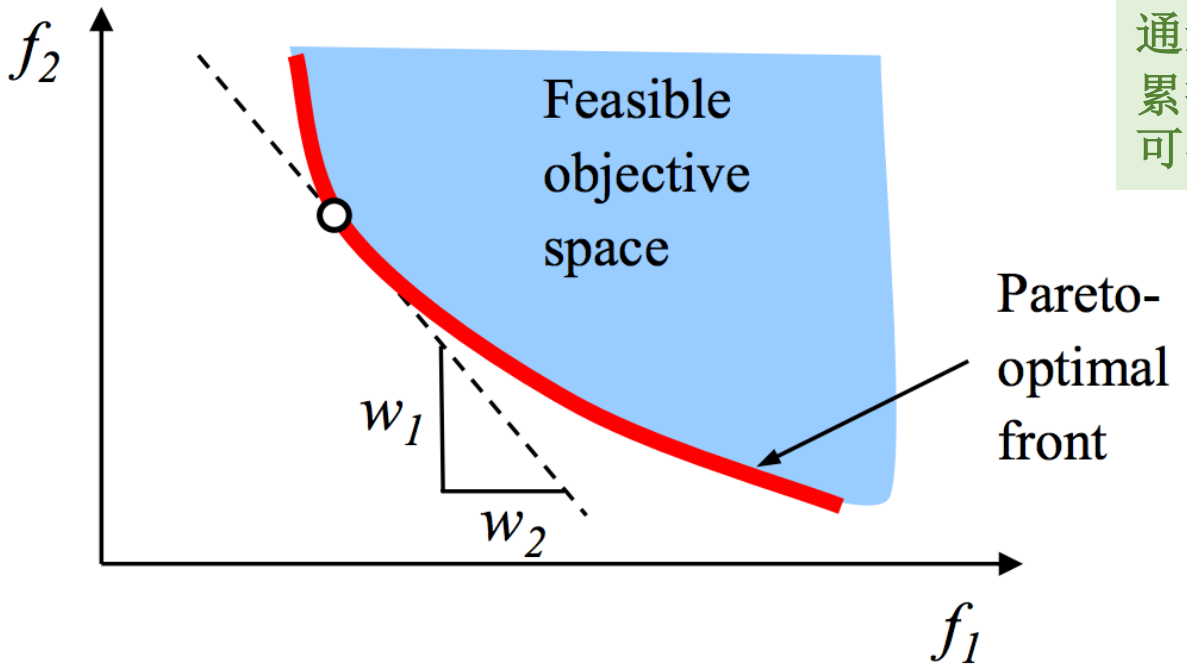


转化后的目标函数 $w_1f_1+w_2f_2$

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）



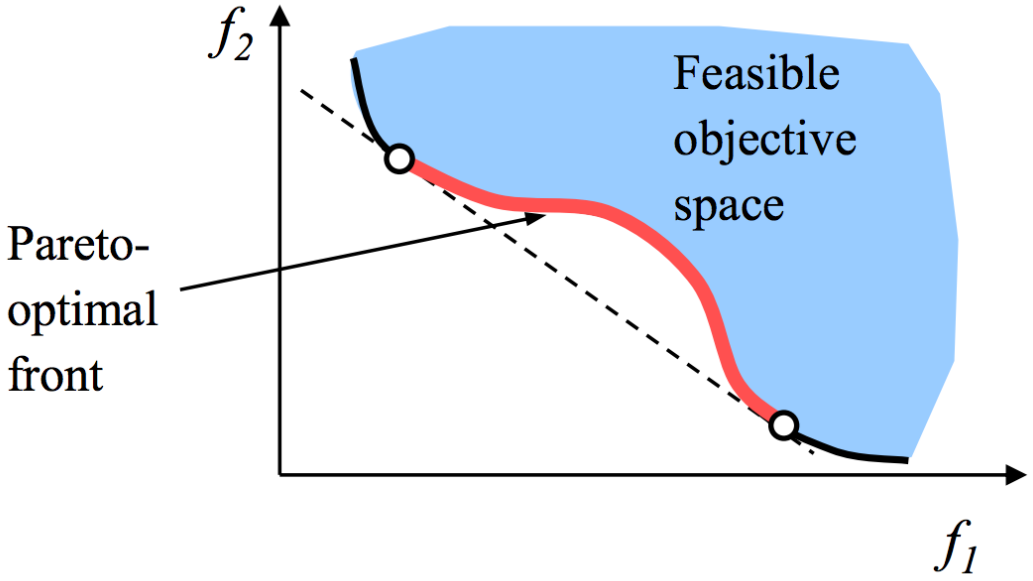
通过调整(w_1,w_2)可以得到帕累托前沿上所有的点对应的可行解

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

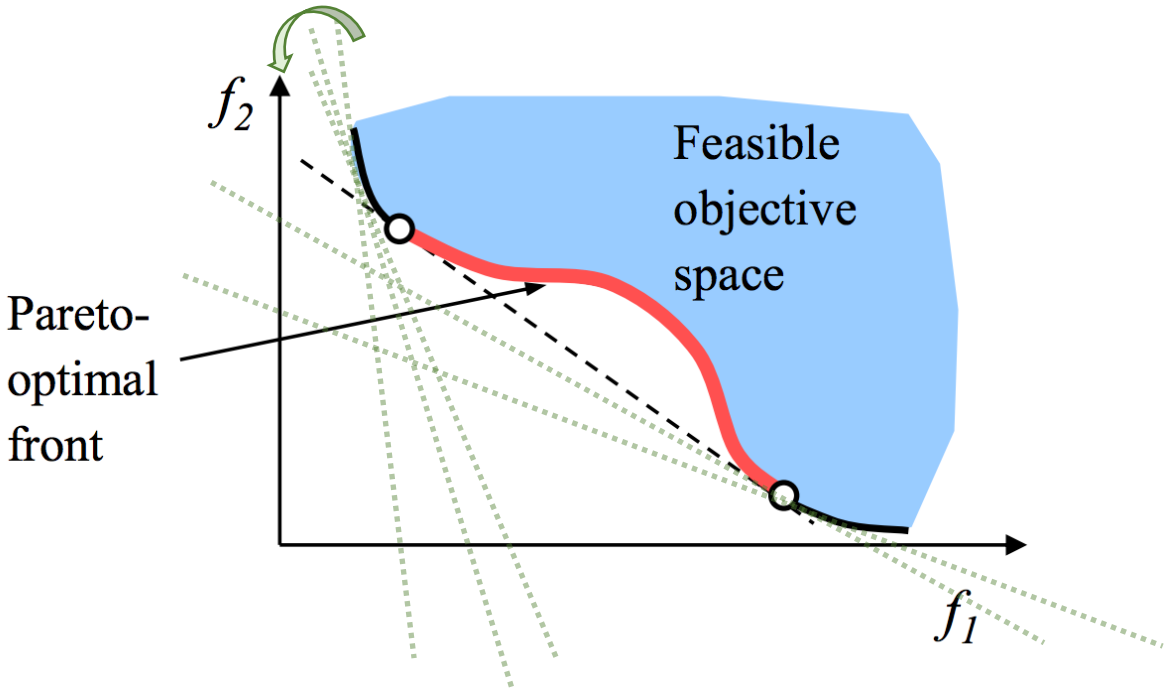
情况二：所有可行解对应的目标函数值的区域是非凸集（non-convex set）



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）

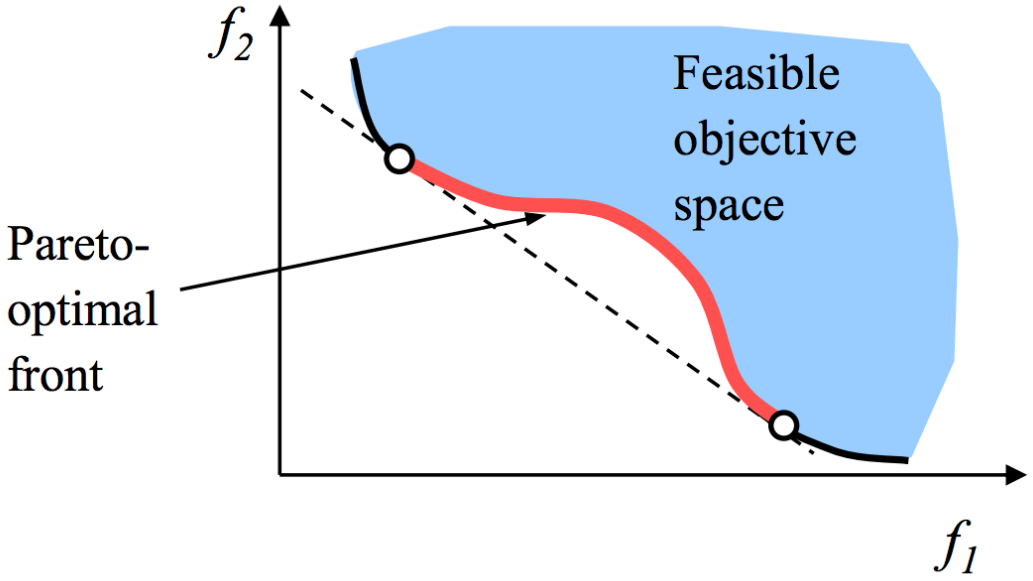


无论如何设置w_1和w_2，标红的帕累托最优解无法取到

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

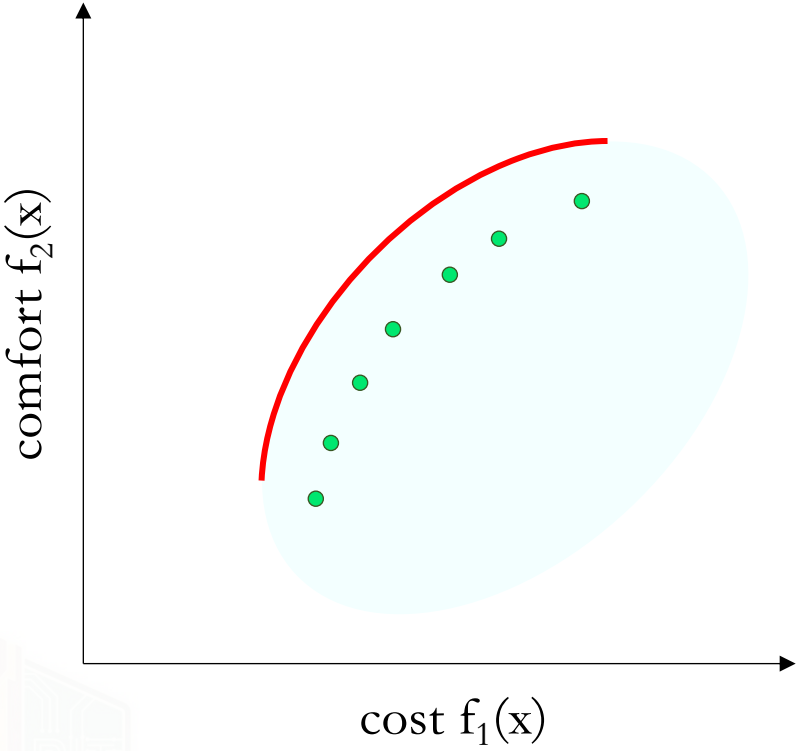
需要解决的问题：如何对解进行筛选？

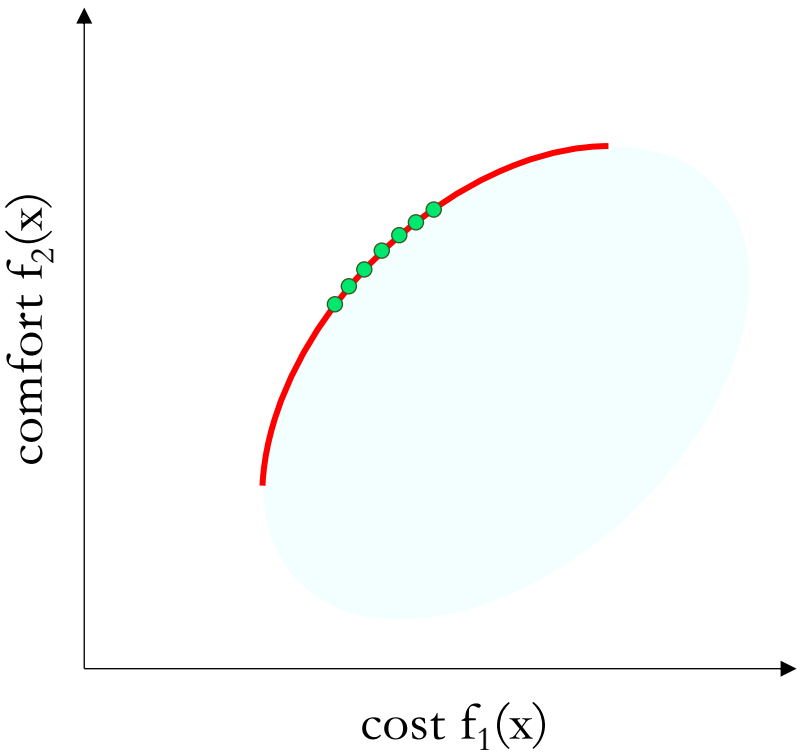
智能优化算法

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

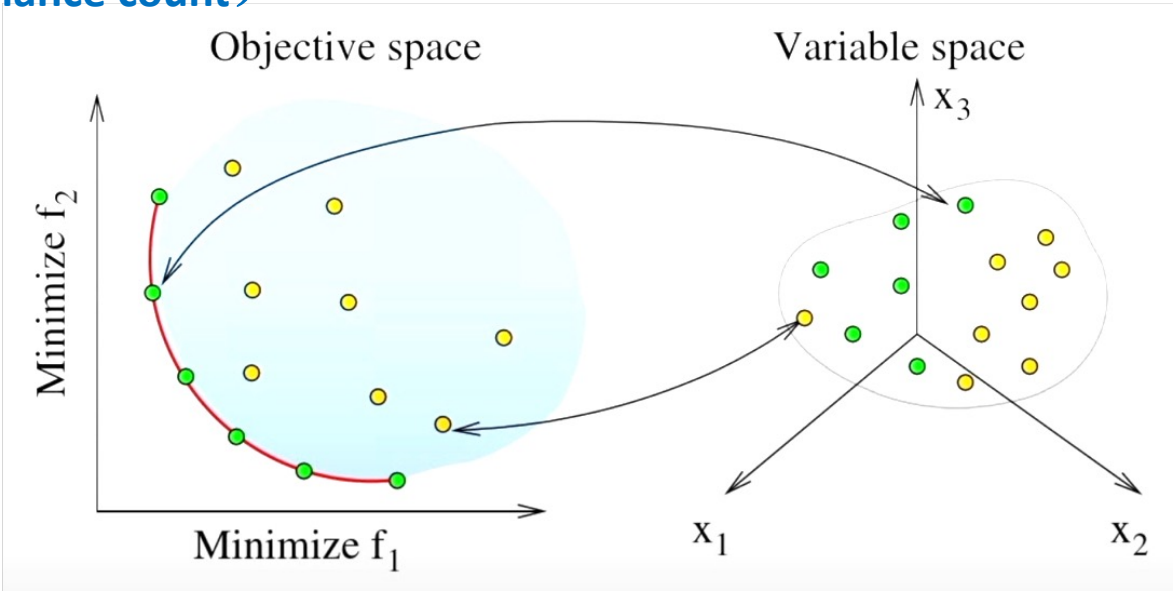
如何衡量哪些解的目标函数值更靠近帕累托最优前沿？

有多种方法：

（1）支配深度（Dominance depth）

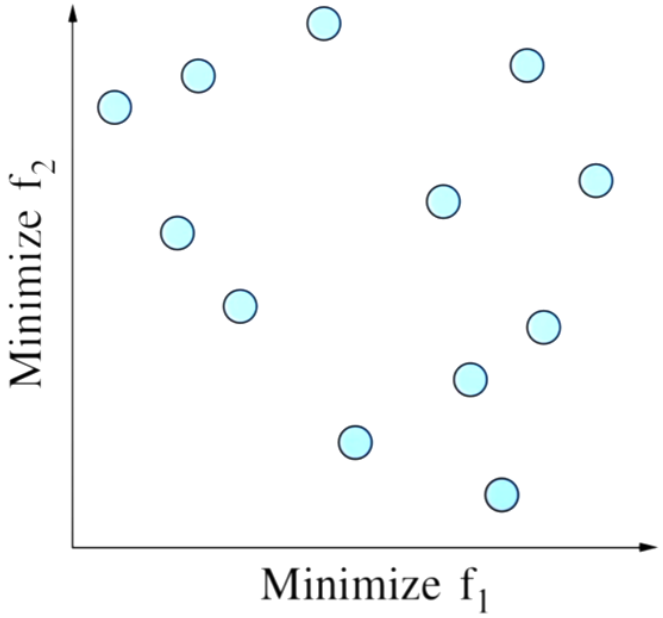
（2）支配次序（Dominance rank）

（3）支配数目（Dominance count）



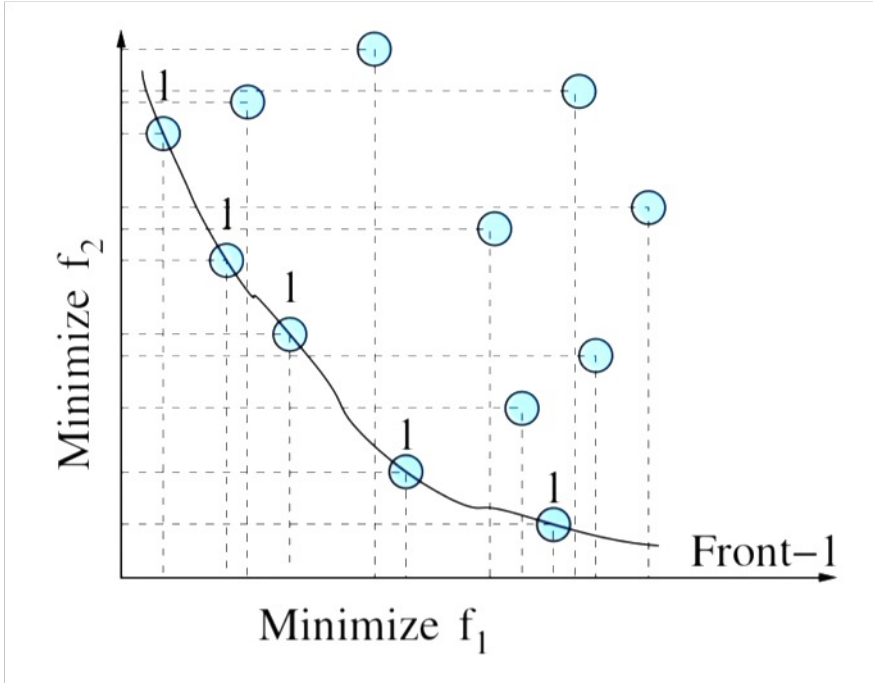
解的筛选之一：排序

利用支配深度（Dominance depth）对解进行排序

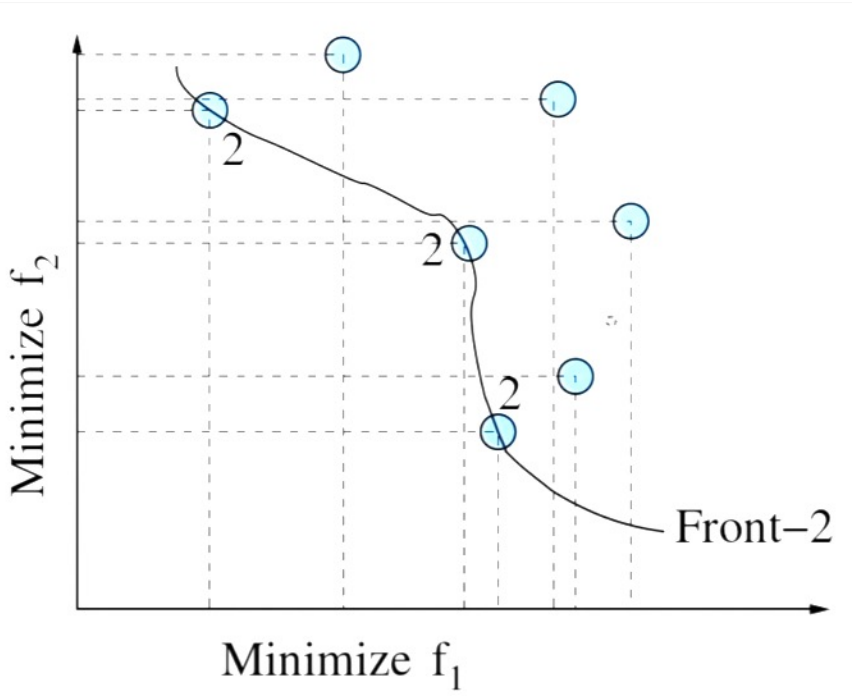


内容取自Deepak Sharma (IIT Guwahati) 课程 Evolutionary Computation for Single and Multi-Objective Optimization

解的筛选之一：排序

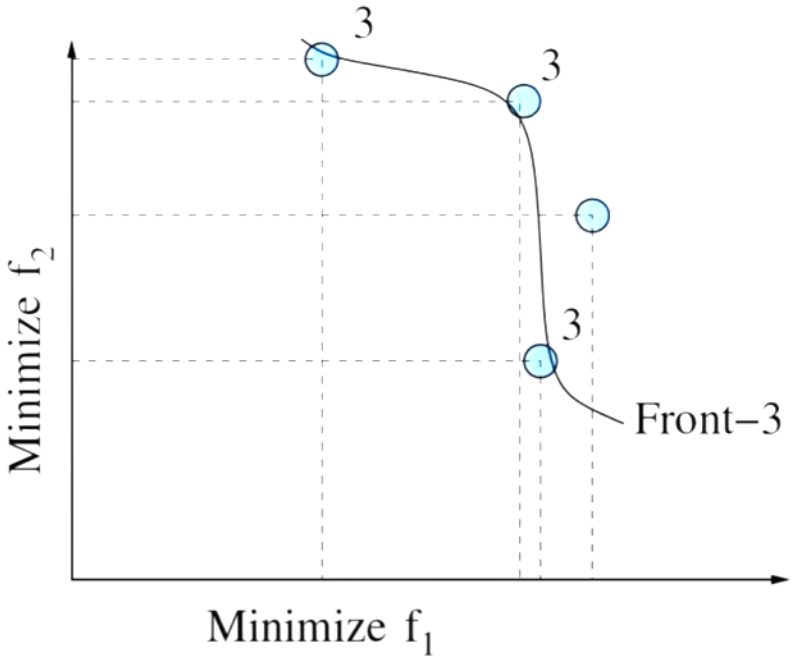


第一步：找到帕累托最优解

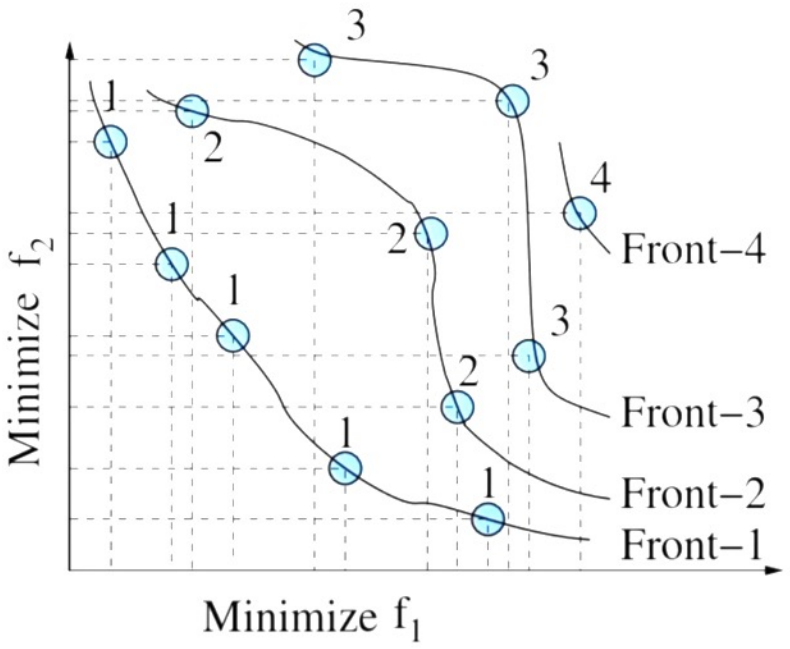


第二步：把这些解标记为**front-1**（第一前沿对应的解）并移除，再在剩余解中找帕累托最优解

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



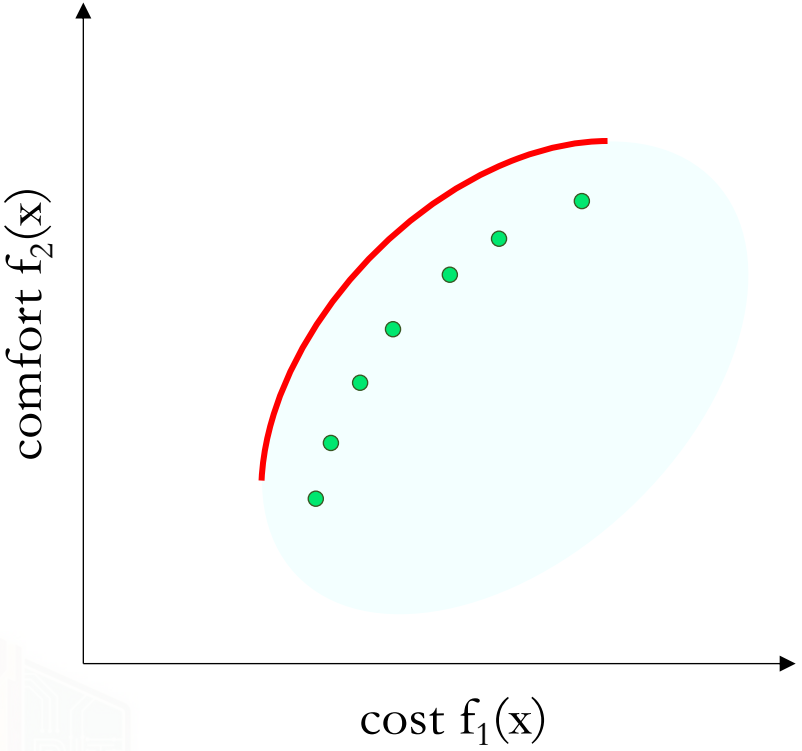
第四步：把这些解标记为front-3并移除，再在剩余解中找帕累托最优解

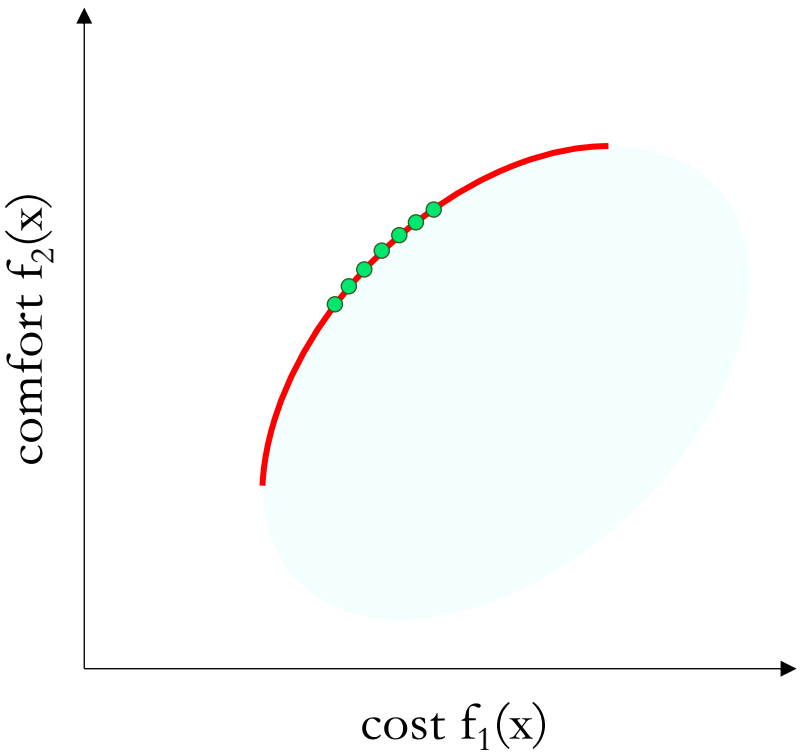
解的筛选之一：排序

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





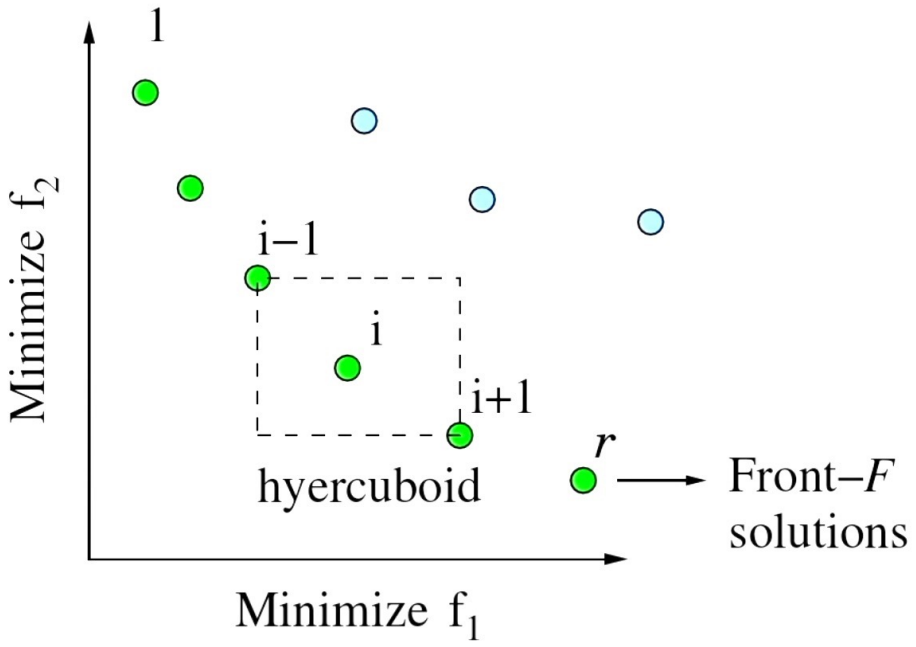
如何衡量哪些解的目标函数值更靠近帕累托最优前沿？ 如何保证筛选的解足够分散？

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**

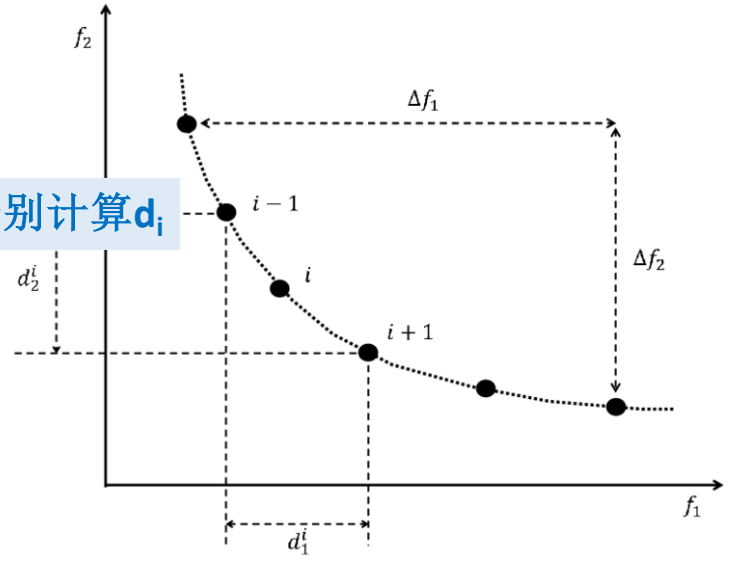
解的筛选之二：多样性

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 $d_i = 0$
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**

为每一条前沿上的点i分别计算d_i



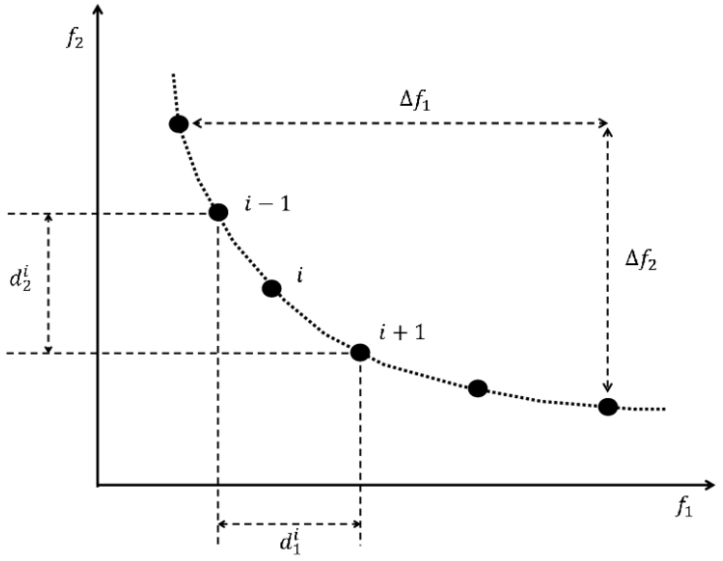
解的筛选之二：多样性

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



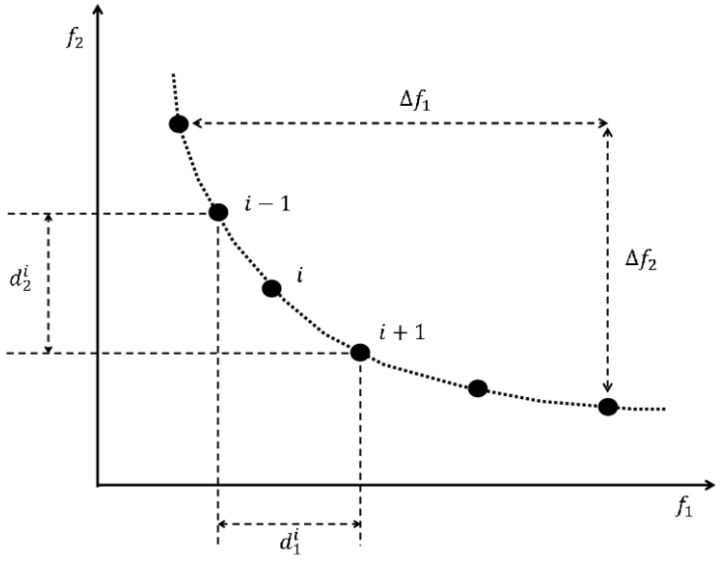
解的筛选之二：多样性

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

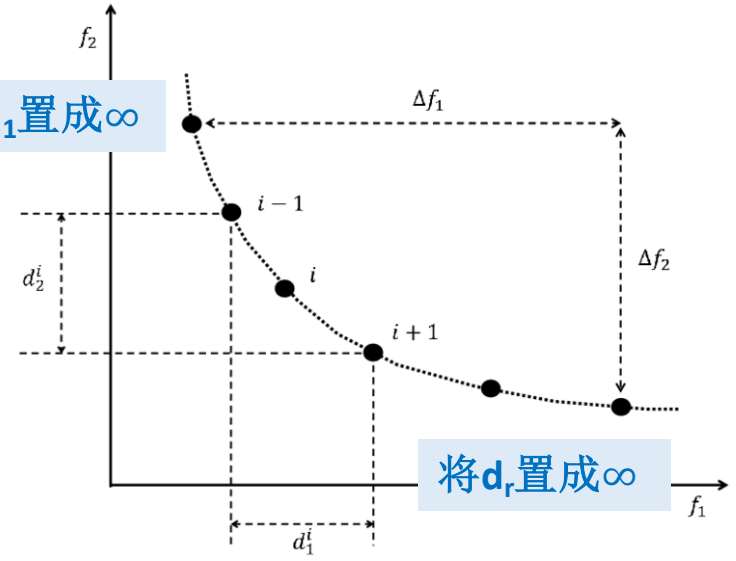
解的筛选之二：多样性

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**

将d_1置成∞



将d_r置成∞

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

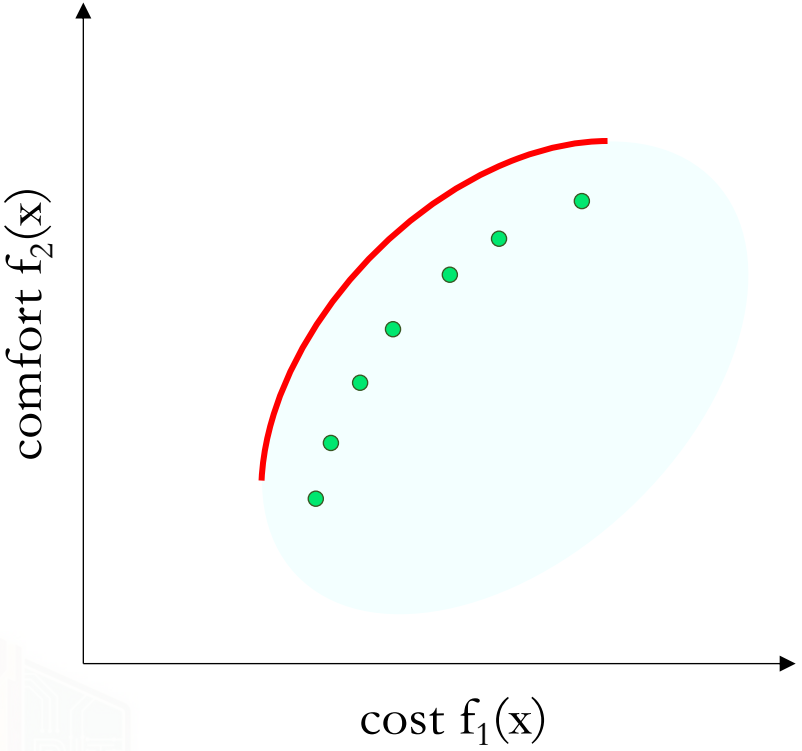
f_1维度　f_2维度

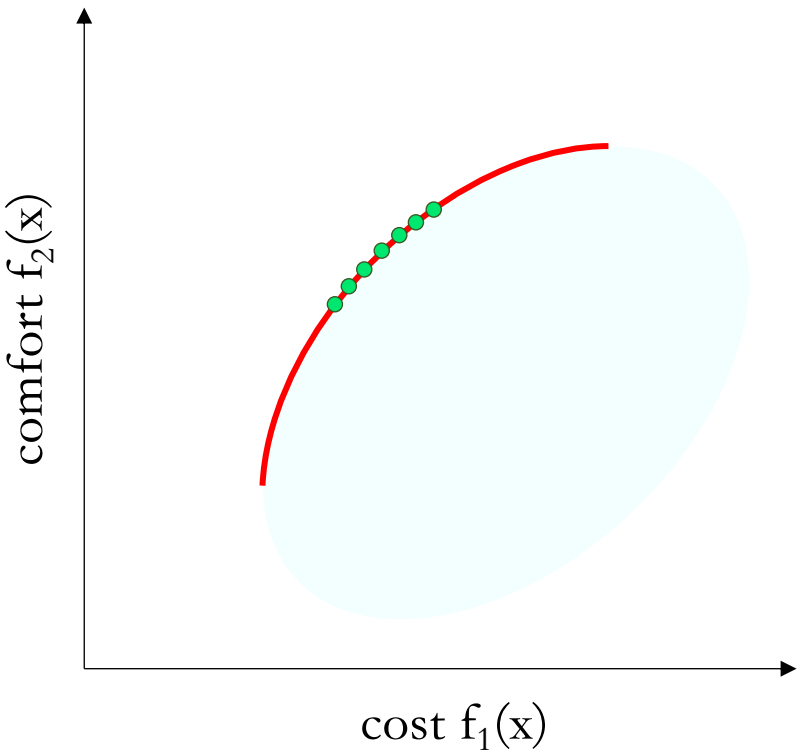
解的筛选之二：多样性

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** …
☆ 99 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 …
☆ 99 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** …
☆ 99 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 …
☆ 99 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 ∞:
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

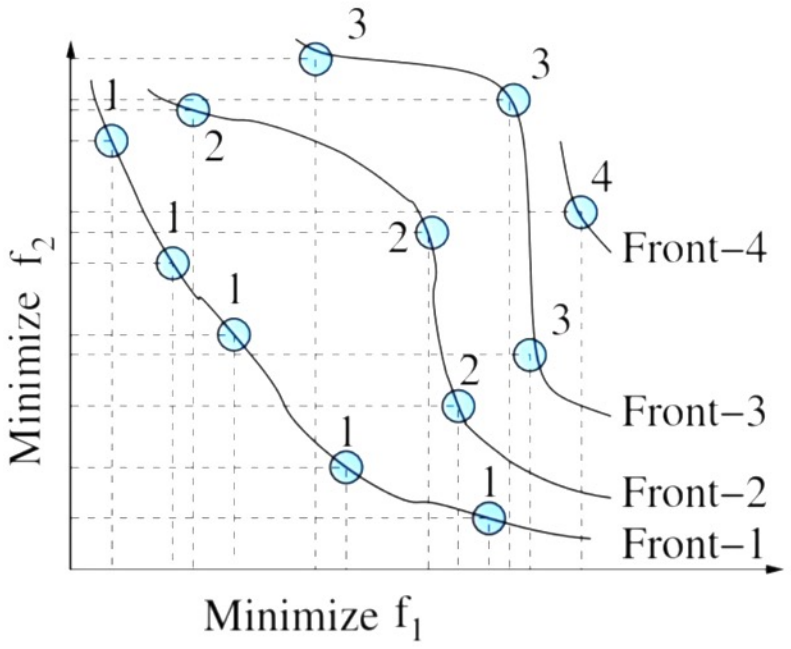
0 初始化当前种群$P(t)$
1 评估当前种群$P(t)$
2 for t=1 to ∞:
3 　根据当前种群$P(t)$，确定父代种群$P_P(t)$
4 　根据父代种群$P_P(t)$，通过基因交叉生成子代种群$P_C(t)$
5 　对子代种群$P_C(t)$进行基因突变
6 　评估子代种群$P_C(t)$
7 　根据当前种群$P(t)$和子代种群$P_C(t)$，筛选出新的当前种群$P(t+1)$
8 　检查循环终止条件，若满足则结束算法

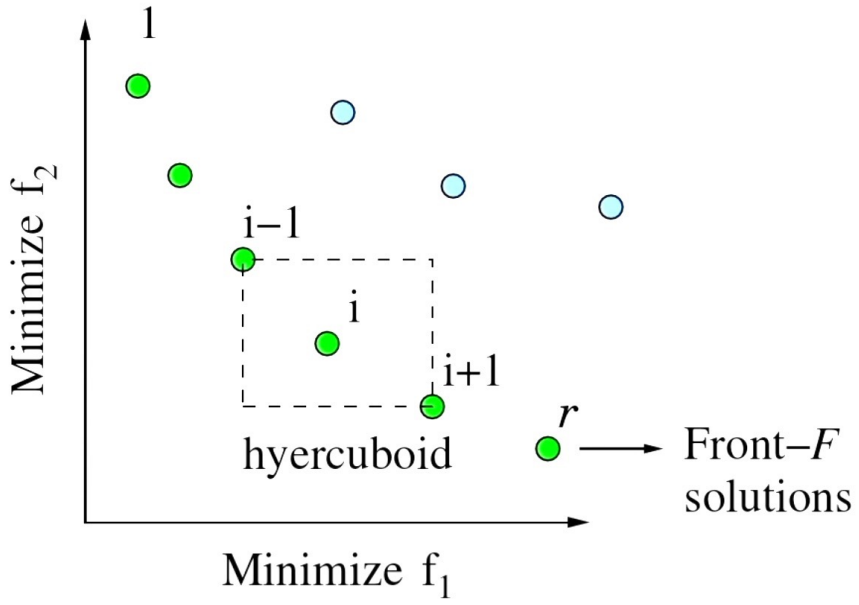
（1）对每个解评估质量（2）对解的筛选

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



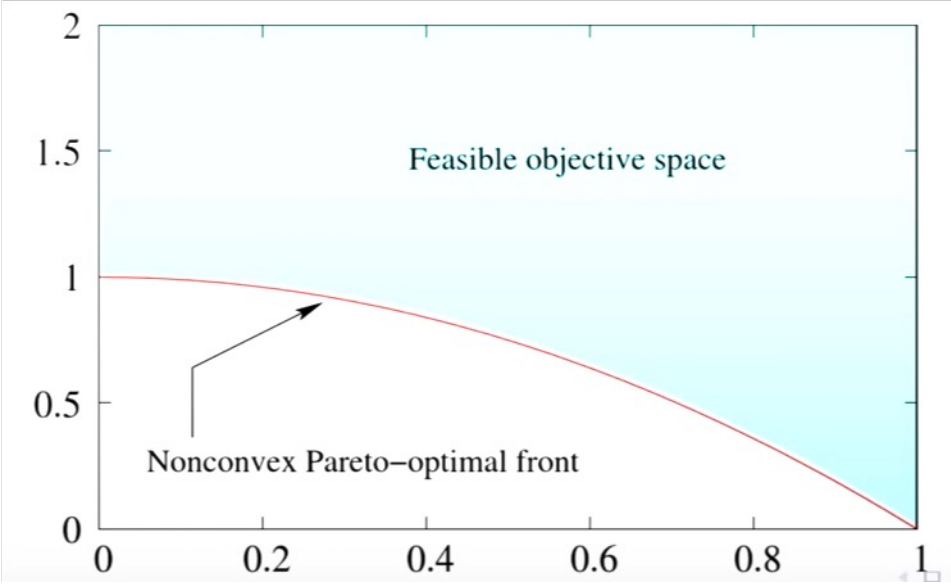


根据这两方面的信息完成对解的筛选

例子

Bi-Objective Optimization

Minimize $f_1(x) = x_1$,
Minimize $f_2(x) = 1 + x_2 - x_1^2$,
bounds $0 \leq x_1 \leq 1$ and $0 \leq x_2 \leq 3$.



加权求和法无法求得
中间的帕累托最优解

内容取自Deepak Sharma (IIT Guwahati) 课程 Evolutionary Computation for Single and Multi-Objective Optimization

例子

- Let the population size is $N = 8$.

得到初始解

Initial population (P)

Index	x_1	x_2
1	0.913	2.181
2	0.599	2.450
3	0.139	1.157
4	0.867	1.505
5	0.885	1.239
6	0.658	2.040
7	0.788	2.166
8	0.342	0.756

例子

评估解质量第一步：计算每个解对应的目标函数取值

- For each solution, calculate f_1 and f_2.
- Let us consider solution 1, $x^{(1)} = (0.913, 2.181)^T$ and $f_1(x) = x_1 = 0.913$ and $f_2(x) = 1 + x_2 - x_1^2 = 2.348$. Similarly, we can calculate both objectives for other solutions.

Solutions with their function values

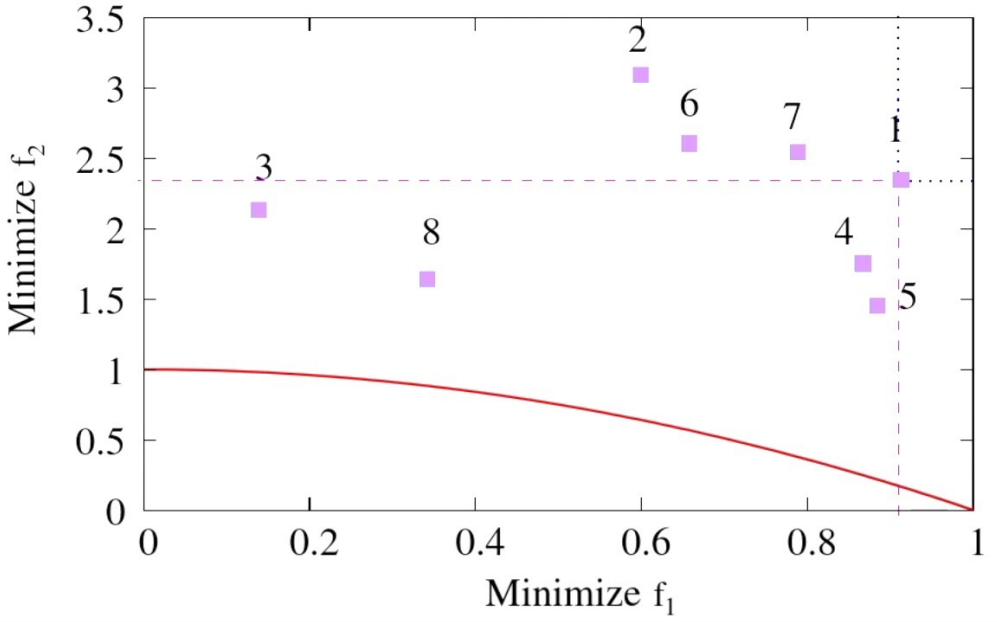
Index	x_1	x_2	f_1	f_2
1	0.913	2.181	0.913	2.348
2	0.599	2.450	0.599	3.092
3	0.139	1.157	0.139	2.138
4	0.867	1.505	0.867	1.753
5	0.885	1.239	0.885	1.455
6	0.658	2.040	0.658	2.607
7	0.788	2.166	0.788	2.545
8	0.342	0.756	0.342	1.639

例子

评估解质量第二步：根据支配深度（Dominance depth）对解进行排序

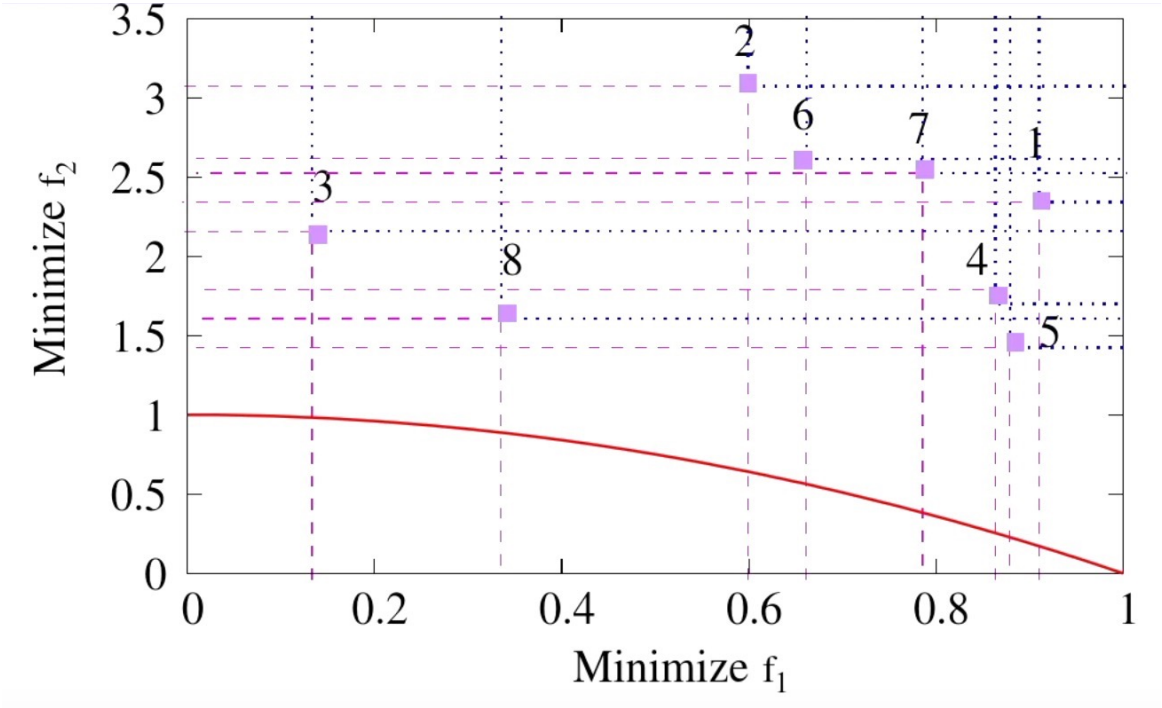
Solutions with their function values

Index	x_1	x_2	f_1	f_2
1	0.913	2.181	0.913	2.348
2	0.599	2.450	0.599	3.092
3	0.139	1.157	0.139	2.138
4	0.867	1.505	0.867	1.753
5	0.885	1.239	0.885	1.455
6	0.658	2.040	0.658	2.607
7	0.788	2.166	0.788	2.545
8	0.342	0.756	0.342	1.639



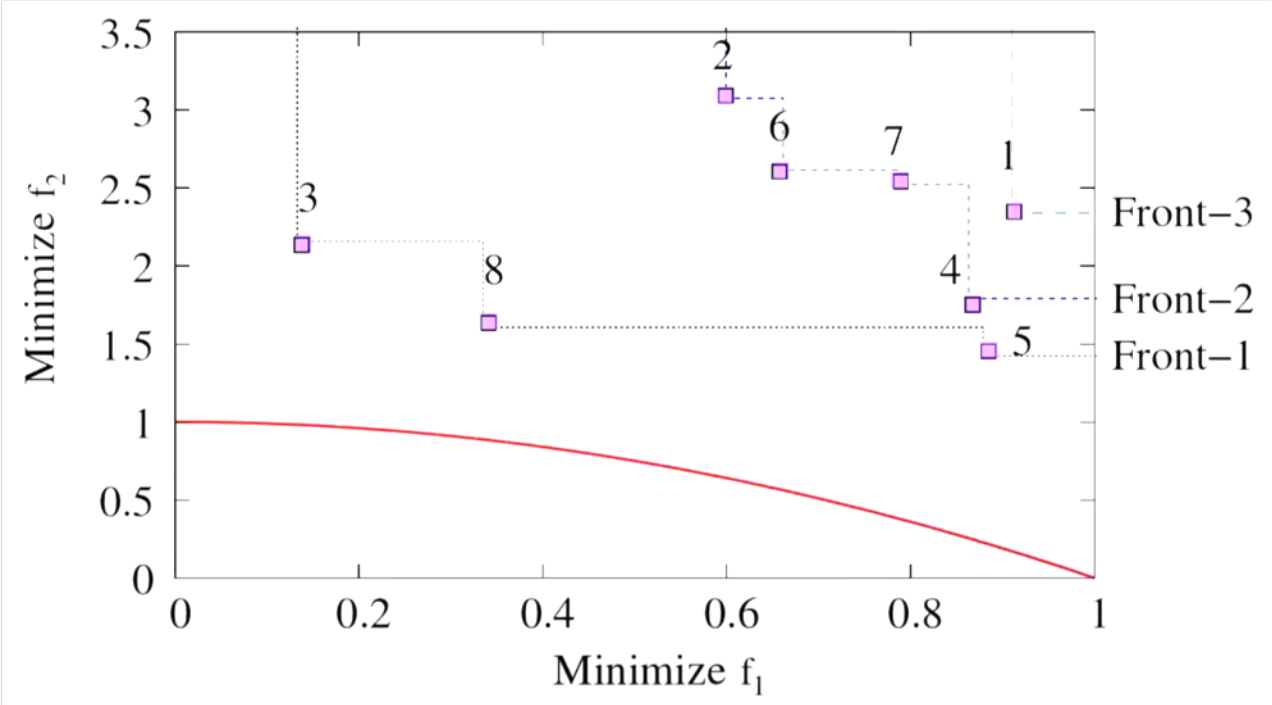
例子

评估解质量第二步：根据支配深度（Dominance depth）对解进行排序



例子

评估解质量第二步：根据支配深度（Dominance depth）对解进行排序

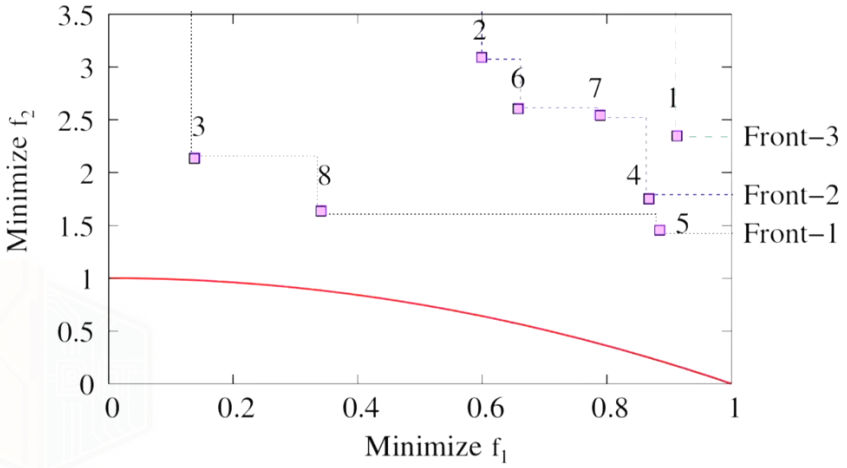


例子

评估解质量第三步：对处在第一前沿、第二前沿等的解分别计算拥挤距离

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**



先看处在front-1的3、8、5号解

- Let us consider f_1 objective.
- f_1 values of these solutions are 0.139, 0.885, 0.342, respectively.
- Sort these solutions based on f_1 value.
- The sorted solutions are $3, 8, 5$. It means that solutions '3' and '5' are extreme solutions.
- Assign $d_3 = d_5 = \infty$.
- The remaining solution is '8'. The crowding distance is

$$d_8 = d_8 + \frac{|f_1(5) - f_1(3)|}{f_1^{max} - f_1^{min}} =$$

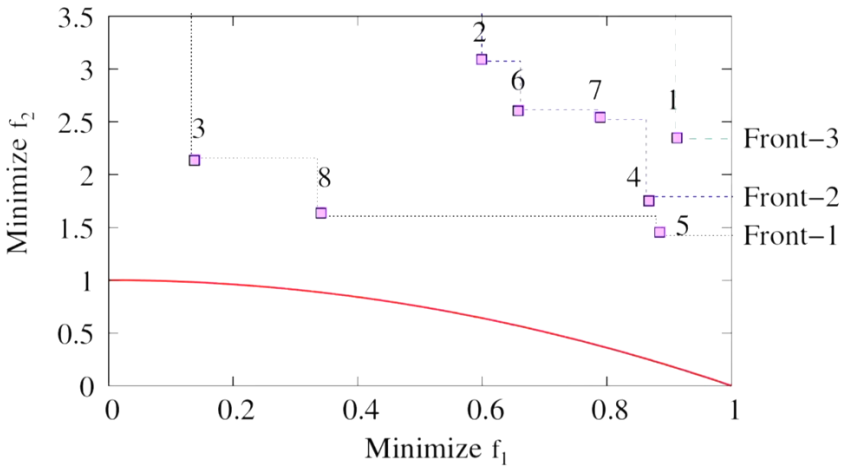
$$d_8 = 0 + \frac{0.885 - 0.139}{0.885 - 0.139} = 1.0$$

例子

评估解质量第三步：对处在第一前沿、第二前沿等的解分别计算拥挤距离

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**



先看处在front-1的3、8、5号解

- Let us consider f_2 objective.
- f_2 values of these solutions are 2.138, 1.455, 1.639, respectively.
- Sort these solutions based on f_2 value.
- The sorted solutions are $5, 8, 3$. It means that solutions '3' and '5' are extreme solutions.
- Assign $d_3 = d_5 = \infty$.
- The remaining solution is '8'. The crowding distance is

$$d_8 = d_8 + \frac{|f_2(3) - f_2(5)|}{f_2^{max} - f_2^{min}} =$$

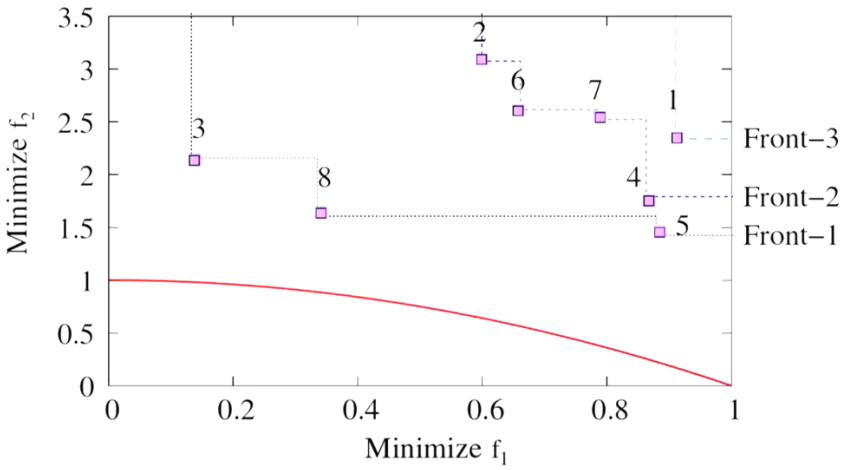
$$d_8 = 1.0 + \frac{2.138 - 1.455}{2.138 - 1.455} = 2.0$$

例子

评估解质量第三步：对处在第一前沿、第二前沿等的解分别计算拥挤距离

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**



再看处在front-2的2、6、7、4号解

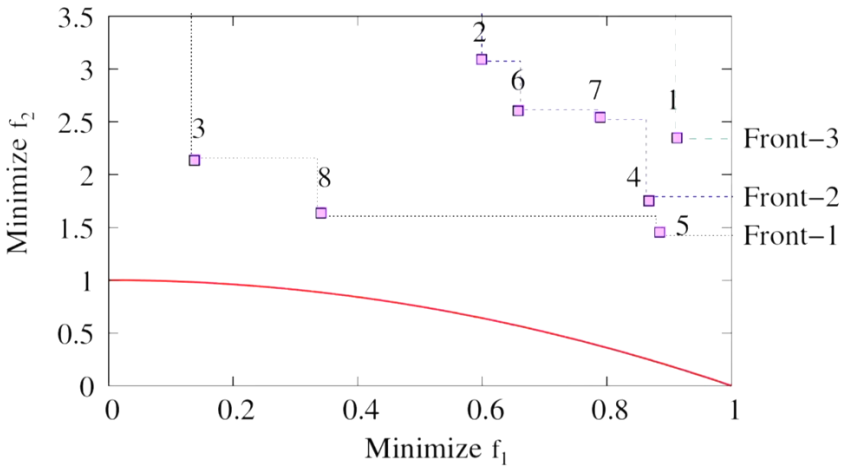
- Let us consider f_1 objective.
- f_1 values of these solutions are 0.599, 0.867, 0.658, 0.788, respectively.
- Sort these solutions based on f_1 value.
- The sorted solutions are $2, 6, 7, 4$. It means that solutions '2' and '4' are extreme solutions.
- Assign $d_2 = d_4 = \infty$.
- For the remaining solutions, the crowding distance is $d_6 = d_6 + \frac{|f_1(7) - f_1(2)|}{f_1^{max} - f_1^{min}} = d_6 = 0 + \frac{0.788 - 0.599}{0.867 - 0.599} = 0.705$
- $d_7 = d_7 + \frac{|f_1(4) - f_1(6)|}{f_1^{max} - f_1^{min}} = d_7 = 0 + \frac{0.867 - 0.658}{0.867 - 0.599} = 0.780$

例子

评估解质量第三步：对处在第一前沿、第二前沿等的解分别计算拥挤距离

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**



再看处在front-2的2、6、7、4号解

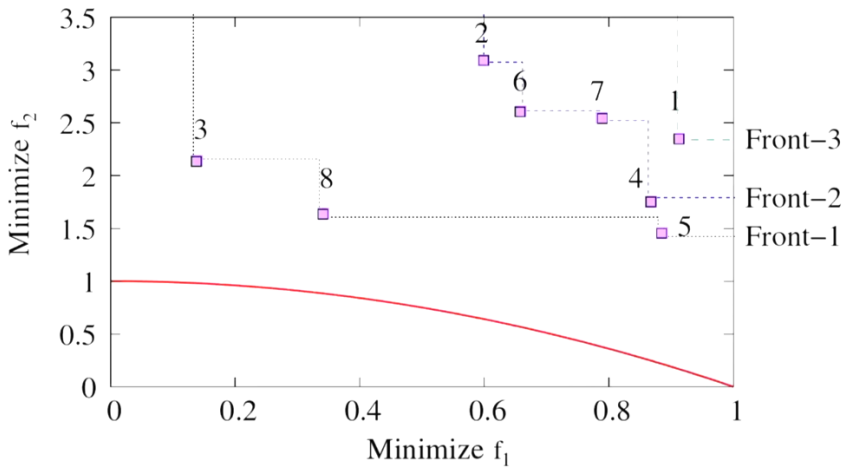
- Let us consider f_2 objective.
- f_2 values of these solutions are 3.092, 1.753, 2.607, 2.545, respectively.
- Sort these solutions based on f_2 value.
- The sorted solutions are $4, 7, 6, 2$. It means that solutions '2' and '4' are extreme solutions.
- Assign $d_2 = d_4 = \infty$.
- For the remaining solutions, the crowding distance is $d_6 = d_6 + \frac{|f_2(2) - f_2(7)|}{f_2^{max} - f_2^{min}} = d_6 = 0.705 + \frac{3.092 - 2.545}{3.092 - 1.753} = 1.113$
- $d_7 = d_7 + \frac{|f_2(6) - f_2(4)|}{f_2^{max} - f_2^{min}} = d_7 = 0.780 + \frac{2.607 - 1.753}{3.092 - 1.753} = 1.418$

例子

评估解质量第三步：对处在第一前沿、第二前沿等的解分别计算拥挤距离

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**

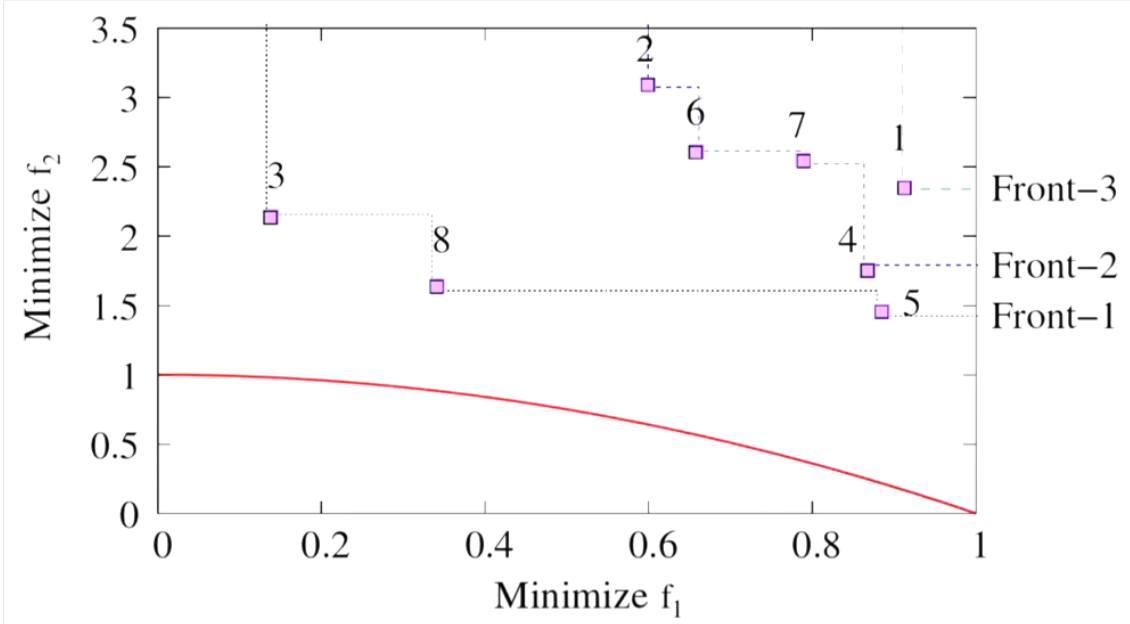


最后看处在front-3的1号解

d_1直接设为∞

例子

完成对所有解的评估



拥挤距离

第几前沿

Solutions with their rank and crowding distance (CD)

Solution	Rank	CD	Solution	Rank	CD	Solution	Rank	CD	Solution	Rank	CD
1	3	∞	2	2	∞	3	1	∞	4	2	∞
5	1	∞	6	2	1.113	7	2	1.418	8	1	2

例子

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 ∞:
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

例子

选择进入配对池的解，设需要选择8个解进入配对池（可重复选择）

Solutions with their rank and crowding distance (CD)

Solution	Rank	CD	Solution	Rank	CD	Solution	Rank	CD	Solution	Rank	CD
1	3	∞	2	2	∞	3	1	∞	4	2	∞
5	1	∞	6	2	1.113	7	2	1.418	8	1	2

采用锦标赛式选择法（**binary tournament selection**）：**8进4淘汰赛**

- Let us perform the first tournament by selecting two solutions randomly.
- Let randomly selected pairs for binary tournament are $\{4, 2\}$, $\{8, 3\}$, $\{5, 1\}$, $\{6, 7\}$.
- Solutions $\{4, 2\}$ are of the same rank and also have the same crowding distance value. Choose any one randomly. We select solution 4.
- Solutions $\{8, 3\}$ are of the same rank but solution 3 is having more crowding distance value than 8. Therefore, solution 3 is selected.
- Between solutions $\{5, 1\}$, solution 5 has better rank and therefore, it is selected.
- Solutions $\{6, 7\}$ are of the same rank but solution 7 is having more crowding distance value than 6. Therefore, solution 7 is selected.

例子

选择进入配对池的解，设需要选择8个解进入配对池（可重复选择）

Solutions with their rank and crowding distance (CD)

Solution	Rank	CD	Solution	Rank	CD	Solution	Rank	CD	Solution	Rank	CD
1	3	∞	2	2	∞	3	1	∞	4	2	∞
5	1	∞	6	2	1.113	7	2	1.418	8	1	2

采用锦标赛式选择法（**binary tournament selection**）：

- Let us perform the second tournament by selecting two solutions randomly.
- Let randomly selected pairs for binary tournament are $\{5, 7\}$, $\{6, 1\}$, $\{8, 2\}$, $\{4, 3\}$.
- Between solutions $\{5, 7\}$, solution 5 has better rank and therefore, it is selected.
- Between solutions $\{6, 1\}$, solution 6 has better rank and therefore, it is selected.
- Between solutions $\{8, 2\}$, solution 8 has better rank and therefore, it is selected.
- Between solutions $\{4, 3\}$, solution 3 has better rank and therefore, it is selected.

例子

选择进入配对池的解，设需要选择8个解进入配对池（可重复选择）

Solutions with their rank and crowding distance (CD)

Solution	Rank	CD	Solution	Rank	CD	Solution	Rank	CD	Solution	Rank	CD
1	3	∞	2	2	∞	3	1	∞	4	2	∞
5	1	∞	6	2	1.113	7	2	1.418	8	1	2

Mating Pool

Old index	New index	x_1	x_2	f_1	f_2
4	1	0.867	1.505	0.867	1.753
3	2	0.139	1.157	0.139	2.138
5	3	0.885	1.239	0.885	1.455
7	4	0.788	2.166	0.788	2.545
5	5	0.885	1.239	0.885	1.455
6	6	0.658	2.040	0.658	2.607
8	7	0.342	0.756	0.342	1.639
3	8	0.139	1.157	0.139	2.138

例子

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 ∞:
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

基因交叉、基因突变与针对单目标优化的实码编码遗传算法中一样

例子

回顾：实码编码遗传算法的基因交叉

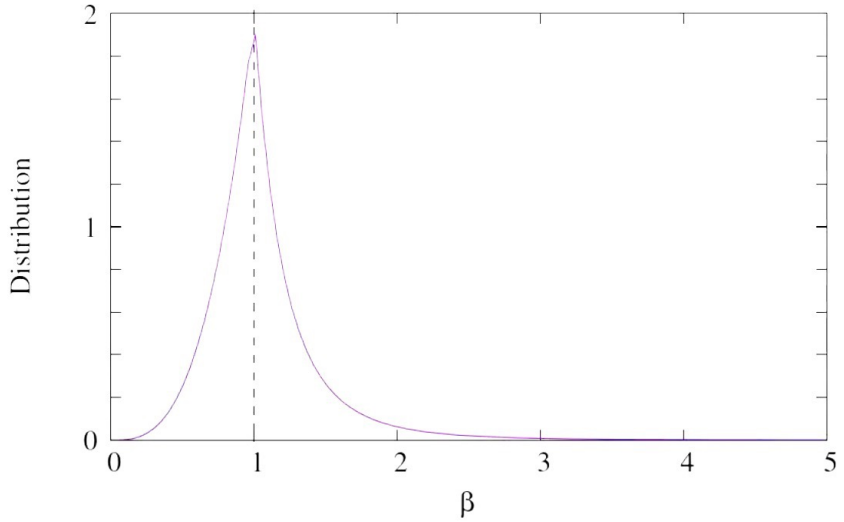
$$o_1 = \bar{x} - \frac{1}{2}\beta(p_2 - p_1)$$
$$o_2 = \bar{x} + \frac{1}{2}\beta(p_2 - p_1)$$

where, $\bar{x} = \frac{1}{2}(p_1 + p_2)$ and $p_2 > p_1$.

- Probability distribution function:

$$p(\beta_i) = \begin{cases} 0.5(\eta_c + 1)\beta_i^{\eta_c}, & \text{if } \beta_i \leq 1 \\ 0.5(\eta_c + 1)\frac{1}{\beta_i^{\eta_c + 2}}, & \text{otherwise.} \end{cases}$$

- η_c is the SBX crossover operator distribution factor that is set by us.



例子

基因交叉

Mating Pool

Old index	New index	x_1	x_2	f_1	f_2
4	1	0.867	1.505	0.867	1.753
3	2	0.139	1.157	0.139	2.138
5	3	0.885	1.239	0.885	1.455
7	4	0.788	2.166	0.788	2.545
5	5	0.885	1.239	0.885	1.455
6	6	0.658	2.040	0.658	2.607
8	7	0.342	0.756	0.342	1.639
3	8	0.139	1.157	0.139	2.138

Solutions after crossover

Index	x_1	x_2	f_1	f_2
1	0.620	2.434	0.620	3.050
2	0.118	1.173	0.118	2.159
3	0.885	2.116	0.885	2.332
4	0.913	1.304	0.913	1.471
5	0.885	1.239	0.885	1.455
6	0.788	2.166	0.788	2.545
7	0.342	0.756	0.342	1.639
8	0.139	1.157	0.139	2.138

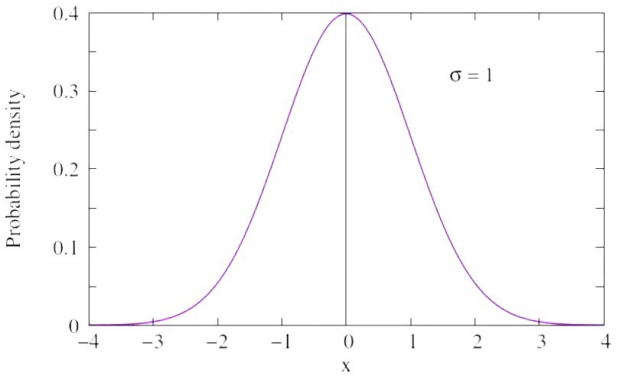
例子

回顾：实码编码遗传算法的基因突变

- Simple and popular method is to use a zero-mean Gaussian probability distribution:

$$y_i^{(1,t)} = x_i^{(1,t)} + N(0, \sigma_i)$$

- σ_i is a fixed user defined parameter. It must be set correctly in a problem.
- Special care must be taken to respect boundary limits on decision variables.



例子

基因突变

Solutions after crossover

Index	x_1	x_2	f_1	f_2
1	0.620	2.434	0.620	3.050
2	0.118	1.173	0.118	2.159
3	0.885	2.116	0.885	2.332
4	0.913	1.304	0.913	1.471
5	0.885	1.239	0.885	1.455
6	0.788	2.166	0.788	2.545
7	0.342	0.756	0.342	1.639
8	0.139	1.157	0.139	2.138

Offspring population

Index	x_1	x_2	f_1	f_2
1	0.620	2.434	0.620	3.050
2	0.165	0.406	0.165	1.379
3	0.885	2.079	0.885	2.295
4	0.985	2.350	0.985	2.380
5	0.826	0.908	0.826	1.226
6	0.788	2.166	0.788	2.545
7	0.343	0.756	0.343	1.639
8	0.121	0.961	0.121	1.946

例子

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 ∞:
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

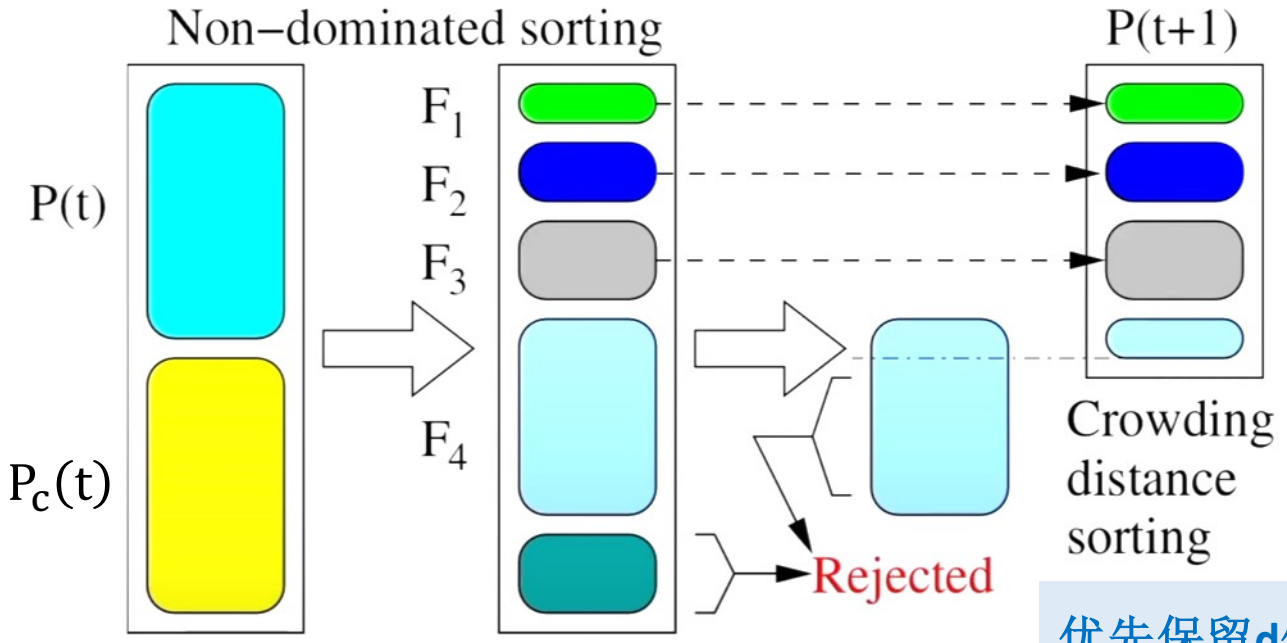
从8+8个解中筛选出下一代的8个解

（需要同时评估16个解的质量：排序、计算d值）

例子

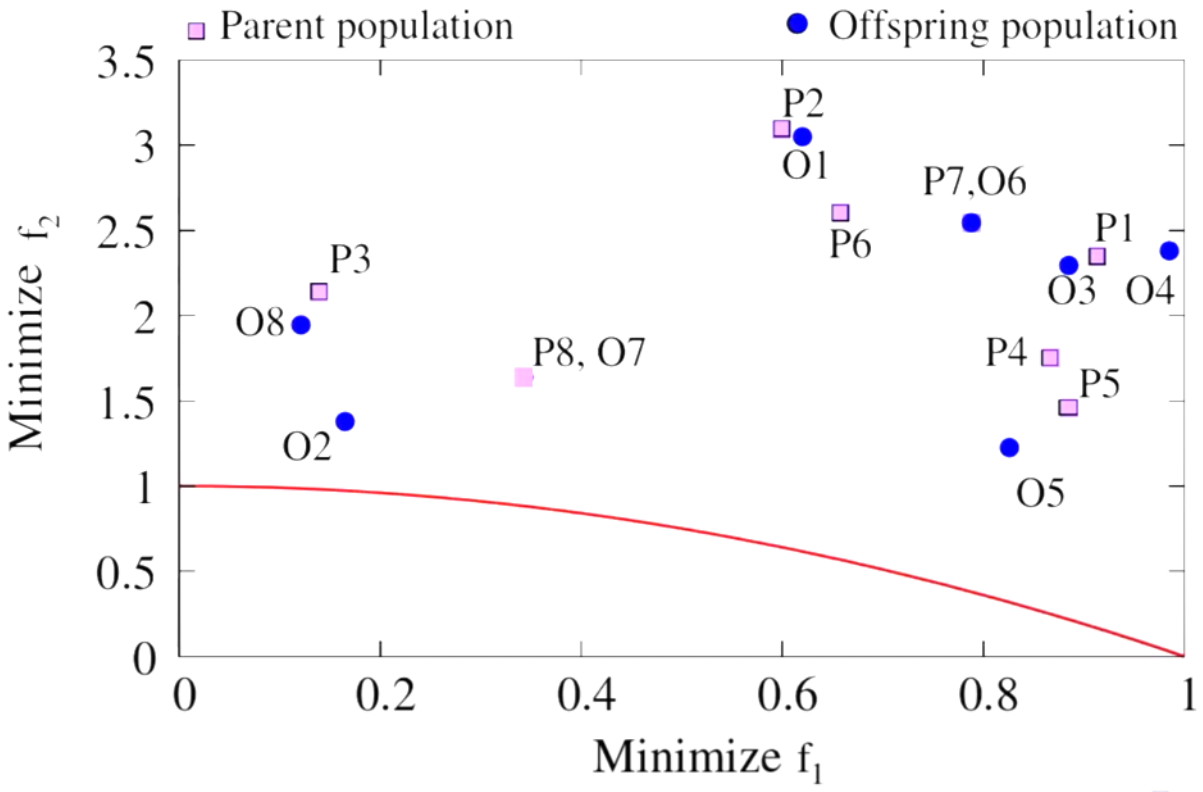
从8+8个解中筛选出下一代的8个解

即按支配深度排序



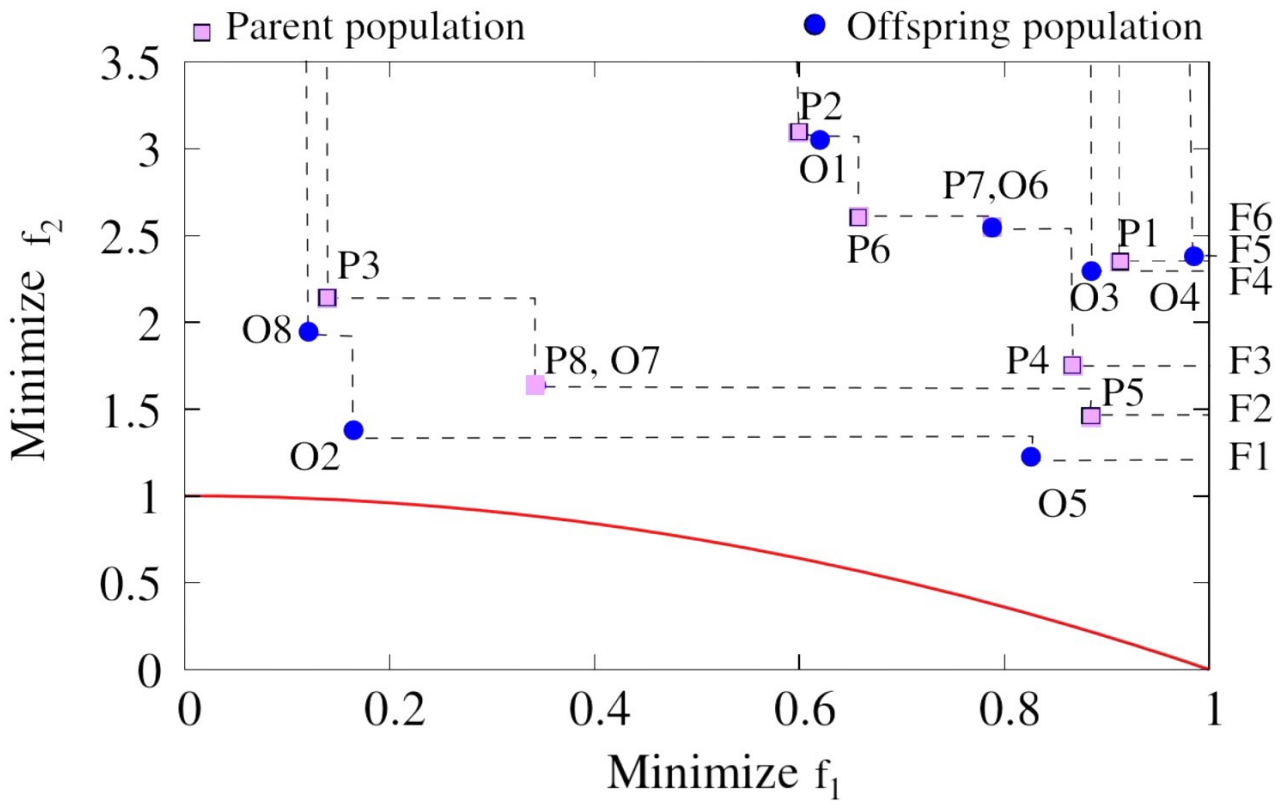
优先保留d值大的解

例子



8个旧解和8个新解的目标函数值在目标函数空间的位置

例子


Parent population
Offspring population
Minimize f_2
Minimize f_1
3.5
3
2.5
2
1.5
1
0.5
0
0
0.2
0.4
0.6
0.8
1
P2
O1
P7,O6
P6
P1
P3
O3
O4
O8
P8, O7
P4
P5
O2
O5
F6
F5
F4
F3
F2
F1

选择哪8个解？

例子

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 ∞:
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

NSGA-II

论文中一个例子的运行结果

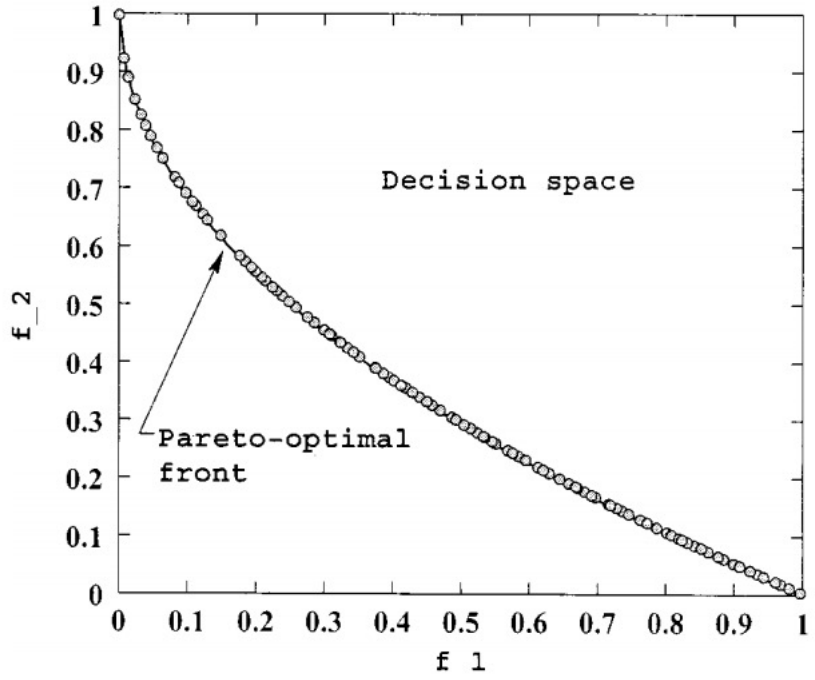


Fig. 12. Obtained nondominated solutions with NSGA-II on problem ZDT4.

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

多目标进化算法的应用

2024 国际大学生数学竞赛华为挑战赛

“基于覆盖仿真模型的无线网络多参数寻优”

赛题三简介：

无线网络信号强度受环境因素（如建筑物遮挡）和网络参数（如天线面板的下倾角）的双重影响。为了保证在动态环境下提供稳定的网络服务，网络运营商必须不断地调整网络参数。通常来说，他们会提供反映网络参数和信号强度之间定量关系的网络仿真模型，使他们能够通过与模型不断交互计算出最佳网络参数配置。然而，模型的计算十分复杂，这导致了模型的交互成本很高。 在本次比赛中，参赛者需要设计出高效、智能的网络优化算法。这些算法可以通过与给定的黑盒仿真模型交互获得给定网络参数对应的适应度值（其表示网络中移动用户平均信号强度）。但由于模型计算的复杂性，算法只能与模型进行有限次的交互。具体来说，我们会为无线网络中的每个天线提供多个候选参数配置。优化算法需要在这些候选项中进行探索，以确定使适应度值最大化的最佳配置。由于巨大的搜索空间和时间限制，暴力搜索对于解决这个问题是不可行

https://www.huawei.com/minisite/imc-challenge/cn/index.html

多目标进化算法的应用

关键方法：多目标进化算法 及 贝叶斯优化

https://www.huawei.com/minisite/imc-challenge/cn/index.html

本讲小结

- 多目标优化经典算法
- 多目标优化进化算法 NSGA-II

计算机学院
余皓然
2024/4/15

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