9. 多目标优化

堵威

华东理工大学 自动化系

2021.4.29

本章内容

1. 背景知识

2. Pareto最优性

3. 多目标优化的目标

4. 基于Pareto的进化算法

5. 总结

本章内容

1. 背景知识

2. Pareto最优性

3. 多目标优化的目标

4. 基于Pareto的进化算法

5. 总结

背景知识

- 多目标优化

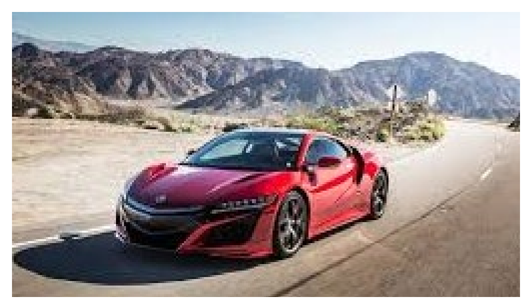
  – 实际优化问题通常包含多个目标，这些目标互相冲突

  1）设计一座桥时，费用最低、强度最大

  2）购买汽车时，舒适度最高、价格最低

  3）设计某种消费品时，盈利最多、市场占有率最高

  4）设计控制系统时，控制器的爬升时间最短、超调最小；对输入敏感性最高、对干扰敏感性最低



背景知识

- 多目标优化
  - 多目标优化（multi-objective optimization, MOO)、多目标优化问题(multi-objective optimization problem, MOP)
  - 多准则优化、多性能优化、向量优化
  - 假设变量**x**为n维，并假定MOP是最小化问题，则问题优化可以写成：

$$\min_{\boldsymbol{x}} f(\boldsymbol{x}) = \min_{\boldsymbol{x}}[f_1(\boldsymbol{x}), f_2(\boldsymbol{x}), \cdots, f_k(\boldsymbol{x})]$$

  - MOP的目标是同时最小化所有k个函数f_i(**x**)
  - 重新定义最优性

本章内容

1. 背景知识

2. Pareto最优性

3. 多目标优化的目标

4. 基于Pareto的进化算法

5. 总结

Pareto最优性

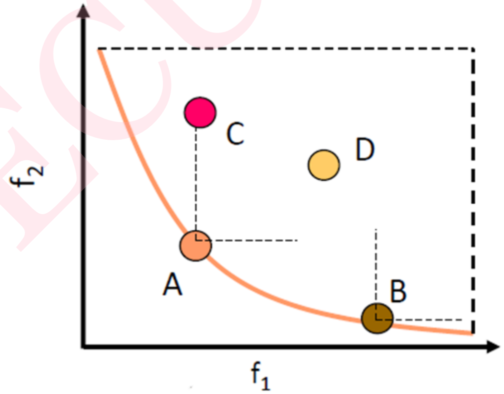
- 基本概念

  - **定义1：支配**

称**x***支配**x**，如果下面的两个条件成立：(1) 对所有$i \in [1,k]$，$f_i(\mathbf{x}^*) \le f_i(\mathbf{x})$；（2）对至少一个$j \in [1,k]$，$f_j(\mathbf{x}^*) < f_j(\mathbf{x})$，即对于所有目标函数值，**x***至少与**x**一样好，并且至少有一个目标函数值比**x**好，记为$\mathbf{x}^* \prec \mathbf{x}$。

  - **定义2：非支配**

称**x***为非支配的，如果不存在能支配它的**x**。



最小化问题：
解A支配解C和解D
解A和解B互不支配

Pareto最优性

- 基本概念

  - **定义3：Pareto最优点**

  Pareto最优点$\mathbf{x}^*$，是不受搜索空间中任一点$\mathbf{x}$支配的点，即$\mathbf{x}$是Pareto最优，则$\nexists x$: $(f_i(\mathbf{x})\leq f_i(\mathbf{x}^*)$对所有$i\in[1,k]$，且$f_j(\mathbf{x})<f_j(\mathbf{x}^*)$对于某个$j\in[1,k])$。

  - **定义4：Pareto最优集**

  所有非支配的$\mathbf{x}^*$的集合，记为P_s。

  - **定义5：Pareto前沿（Pareto front，PF）**

  Pareto最优集的函数$f(\mathbf{x})$的集合，记为P_f。$P_f=\{f(\mathbf{x}^*)\text{: }\mathbf{x}^*\in P_s\}$

Pareto最优性

- 基本概念
  - 例：假设最小化一个MOP，它有二维独立变量**x** (n=2)，x取6个离散值 $\mathbf{x}^{(i)}$，i∈[1,6]。进一步假设两个目标(k=2)的函数值为：

$f_1(\mathbf{x}^{(1)}) = 1$，$f_2(\mathbf{x}^{(1)}) = 3$

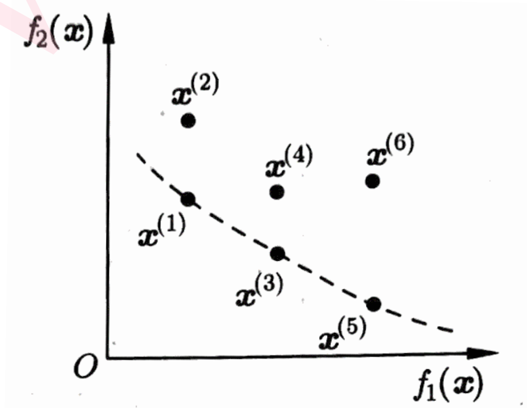
$f_1(\mathbf{x}^{(2)}) = 1$，$f_2(\mathbf{x}^{(2)}) = 4$

$f_1(\mathbf{x}^{(3)}) = 3$，$f_2(\mathbf{x}^{(3)}) = 3$

$f_1(\mathbf{x}^{(4)}) = 2$，$f_2(\mathbf{x}^{(4)}) = 3$

$f_1(\mathbf{x}^{(5)}) = 3$，$f_2(\mathbf{x}^{(5)}) = 1$

$f_1(\mathbf{x}^{(6)}) = 3$，$f_2(\mathbf{x}^{(6)}) = 3$



{$\mathbf{x}^{(1)}$，$\mathbf{x}^{(3)}$，$\mathbf{x}^{(5)}$}构成此多目标最小化问题的Pareto集，相应的函数向量构成Pareto前沿

Pareto最优性

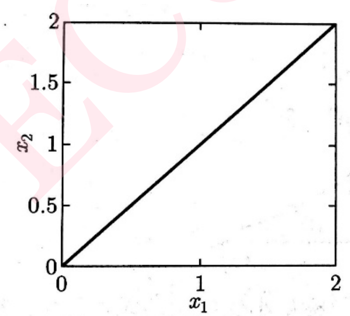
- 基本概念
  - 例：考虑MOP

$$\min f(\mathbf{x}) = \min\{f_1(\mathbf{x}), f_2(\mathbf{x})\} = \min\{x_1^2+x_2^2, (x_1-2)^2+(x_2-2)^2\}$$

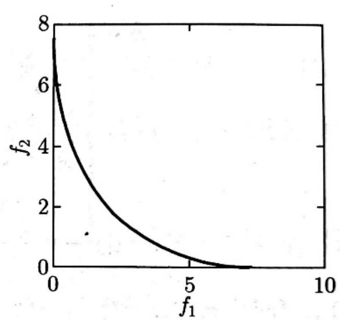
其中，$x_1 \in [0, 2]$, $x_2 \in [0, 2]$。

f1(x): (0, 0)，Pareto点；f2(x): (2, 2)，Pareto点

如果采用蛮力搜索找出所有的Pareto点，则得到Pareto集：Ps = $\{\mathbf{x}: x_1=x_2, x_1 \in [0, 2]\}$，相应的Pareto前沿为：$P_f = \{(f_1, f_2): f_1=2x_1^2, f_2=(x_1-2)^2, x_1 \in [0, 2]\}$



Pareto集



Pareto前沿

本章内容

1. 背景知识

2. Pareto最优性

3. 多目标优化的目标

4. 基于Pareto的进化算法

5. 总结

多目标优化的目标

- 基本概念
  - 单目标优化的目标很直接：尽量快地找到使目标函数值最大/最小的解
  - 多目标优化：

1）在与真实Pareto集的某个距离之内找到最多个体

2）近似Pareto集与真实Pareto集的平均距离最小

3）在近似Pareto集中找到最具多样性的个体

4）目标函数空间中候选解到理想点（乌托邦点）的距离最小

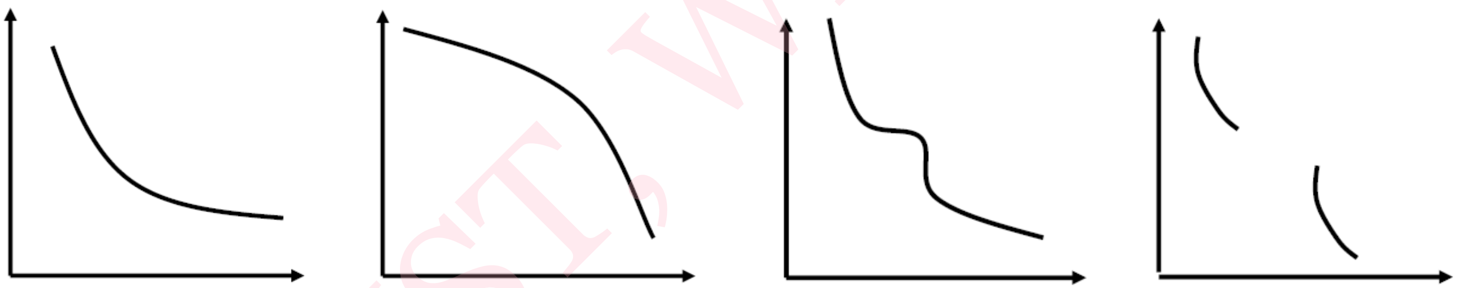
1）、2）：找出真实Pareto集“最好的”近似

3）：为决策者找出解的多样化集合

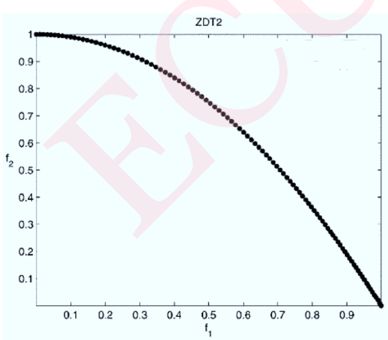
4）：找出与决策者的理想解尽可能接近的解

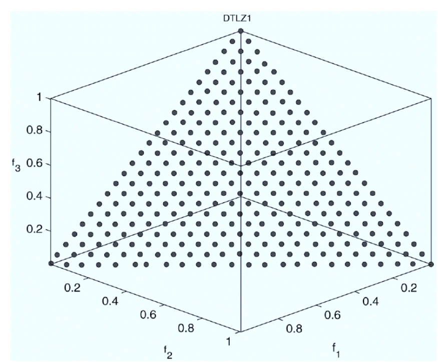
多目标优化的目标

- 基本概念
  - Pareto前沿的形状



convex　　concave　　non-convex　　discontinuous





评判标准：
分散、收敛

多目标优化的目标

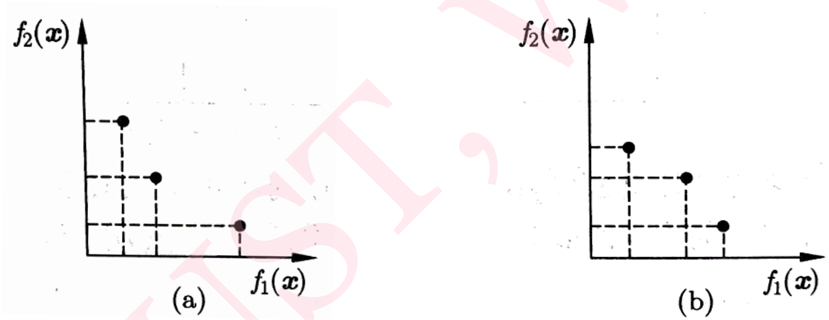
- 超体积（Hypervolume）
  - 常用的度量Pareto前沿质量的指标
  - 假设在近似Pareto前沿中找到M个点$P_f=\{\mathbf{f}(\mathbf{x}_j)\}$，$j\in[1,M]$，其中$\mathbf{f}(\mathbf{x}_j)$是一个k维函数。则其超体积为：

$$S(P_f)=\sum_{j=1}^{M}\prod_{i=1}^{k}f_i(\boldsymbol{x}_j)$$

  - 已知两个多目标优化算法，可以通过比较超体积比较两个算法找到Pareto前沿的优劣；对于最小化问题，超体积越小表明Pareto前沿近似得也越好

多目标优化的目标

- 超体积（Hypervolume）
  - 例：假设有两个多目标优化算法，对于一个两目标的最小化问题，其得到的近似Pareto前沿如下：



图(a)的近似Pareto前沿为$P_f(1) = \{f_1(\mathbf{x}_j), f_2(\mathbf{x}_j)\} = \{[1,5], [2,3], [5,1]\}$, 其超体积为5+6+5=16；

图(b)的近似Pareto前沿为$P_f(2) = \{f_1(\mathbf{x}_j), f_2(\mathbf{x}_j)\} = \{[1,4], [3,3], [4,1]\}$, 其超体积为4+9+4=17.

多目标优化的目标

- 超体积（Hypervolume）
  - 不能盲目地将超体积作为衡量Pareto前沿质量的指标，根据定义：

$$S(P_f) = \sum_{j=1}^{M} \prod_{i=1}^{k} f_i(\boldsymbol{x}_j)$$

由上式可得：一个空的近似Pareto前沿（M=0）能得到S可能的最小值；因此，更精确地度量可能是正规化后的超体积$S_n(P_f)=S(P_f)/M$。但是这个量也可能不是近似Pareto前沿好的指标，原因如下：

  - 考虑某个近似Pareto前沿$P_f(1)$，其正规化后的超体积为$S_n(P_f(1))$，现在假设添加一个新的点到$P_f(1)$得到$P_f(2)$，于是，有可能让$S_n(P_f(2))> S_n(P_f(1))$，尽管$P_f(1)$和$P_f(2)$唯一的不同是$P_f(2)$多了一个点， $P_f(2)$显然比$P_f(1)$要好，但$S_n(P_f(2)) > S_n(P_f(1))$，这与我们的直觉不符。

多目标优化的目标

- 超体积（Hypervolume）
  - 因此考虑修改超体积测度，不是关于目标函数空间的原点来计算超体积，而是关于Pareto前沿之外的参考点来计算
  - 假设要比较Q个Pareto前沿$P_f(q)$，$q \in [1,Q]$，首先计算参考点向量$r = [r_1, r_2, ..., r_k]$，其中

$$r_i > \max_q \left[\max_{\boldsymbol{x} \in P_s(q)} f_i(\boldsymbol{x}) \right]$$

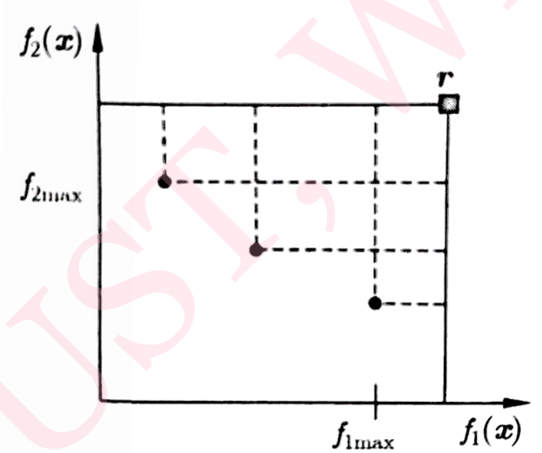
然后关于参考点计算超体积S'：

$$S'(P_{\boldsymbol{f}}(q)) = \sum_{j=1}^{M(q)} \prod_{i=1}^{k} (r_i - f_i(\boldsymbol{x}_j(q)))$$

其中$M(q)$是在第q个近似Pareto前沿中点的个数，$x_j(q)$是在第q个近似Pareto集中的第j个点。S'较大表示最小化问题的Pareto前沿较好。

多目标优化的目标

- 超体积（Hypervolume）
  - 更新后的超体积公式的示意图如下：



r为任意一个参考点，它的第i个分量大于近似Pareto前沿中的每个点的第i个分量；超体积较大表示最小化问题的近似Pareto前沿较好

多目标优化的目标

- 反世代距离（Inverted Generational Distance, IGD）
  - 多目标算法中的综合性能评价指标
  - 通过计算每个在真实Pareto前沿上的点到算法获得的近似Pareto前沿之间的最小距离和，公式如下：

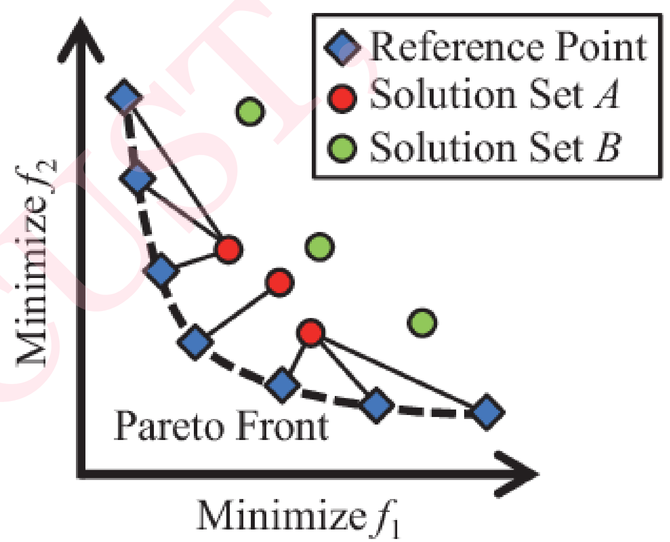
$$IGD(P,Q)=\frac{\sum_{v\in P}d(v,Q)}{|P|}$$

其中P为均匀分布在真实Pareto前沿上的点集，|P|为分布在真实Pareto前沿上点集大小，Q为算法得到的近似Pareto前沿，d(v,Q)为P中的点v到Q的最小欧氏距离。

  - IGD通过计算真实Pareto前沿上点集到算法得到的Pareto前沿的最小距离平均值来评价算法的综合性能

多目标优化的目标

- 反世代距离（Inverted Generational Distance, IGD）
  - 评价收敛性能：当算法的收敛性能比较好时，d(v,Q)较小
  - 评价分布性能：当算法的分布性能比较好时，d(v,Q)较小



本章内容

1. 背景知识

2. Pareto最优性

3. 多目标优化的目标

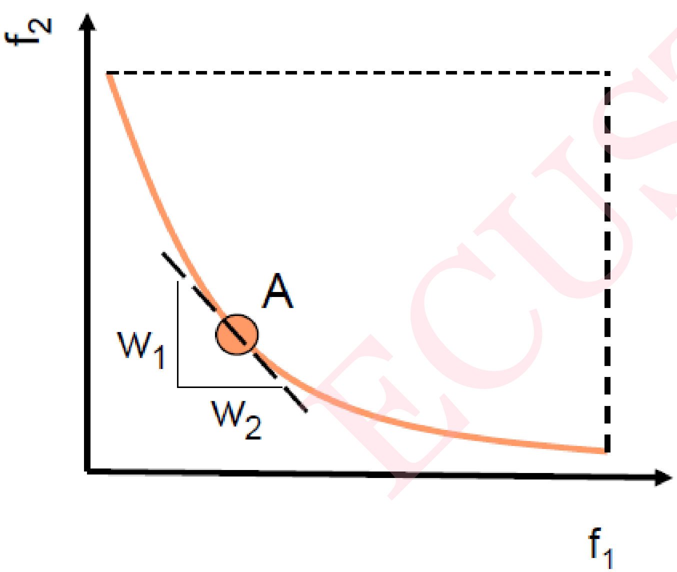
4. 基于Pareto的进化算法

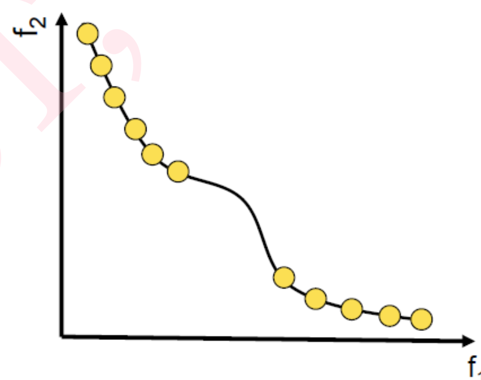
5. 总结

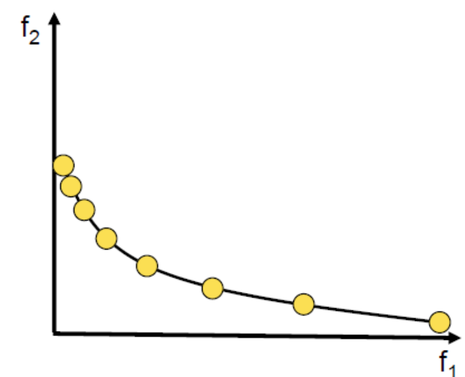
基于Pareto的进化算法

- 传统求解多目标优化问题的方法
  - 加权和法：

$$F(\mathbf{X}) = \sum_{m=1}^{M} w_m f_m(\mathbf{X})$$





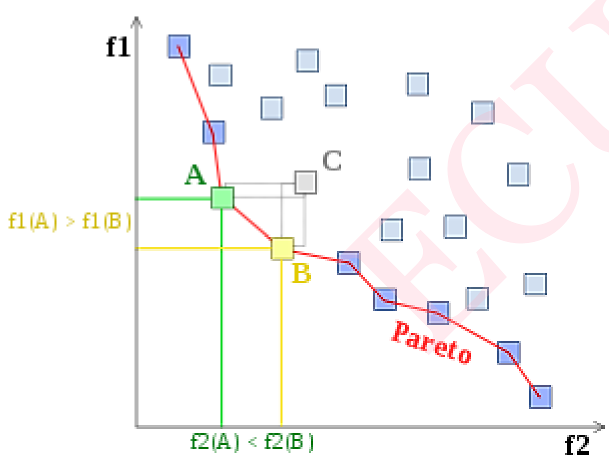


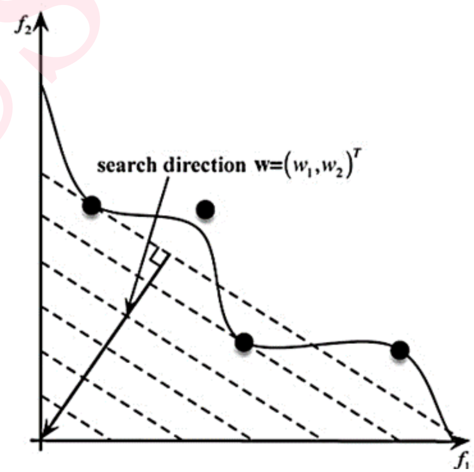
缺点：1）凹的区域；2）不一定能得到均匀的解

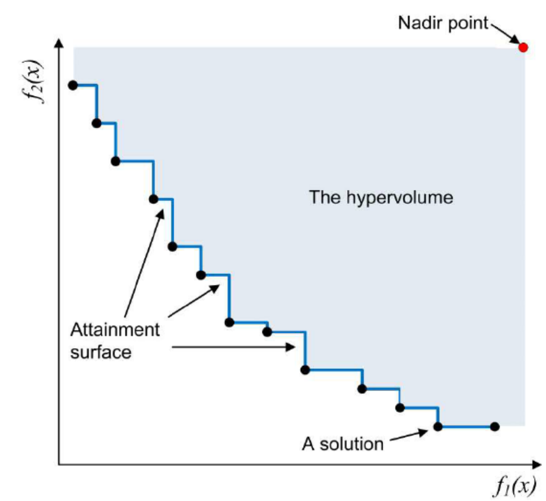
基于Pareto的进化算法

- 进化多目标优化（Evolutionary Multi-objective Optimization）
  - 进化算法是基于种群的，非常适合求解多目标优化问题
  - 相较于传统的方法，能够通过一次求解，得到一组解
  - 三大类进化多目标优化方法

  1）基于Pareto的方法；2）基于分解的方法；3）基于指标的方法







基于Pareto的进化算法

- 非支配排序遗传算法
  - Nondominated sorting genetic algorithm, NSGA
  - 根据每一个个体的支配水平为其分配适应度值
  - 关键：分批找出种群中的非支配个体

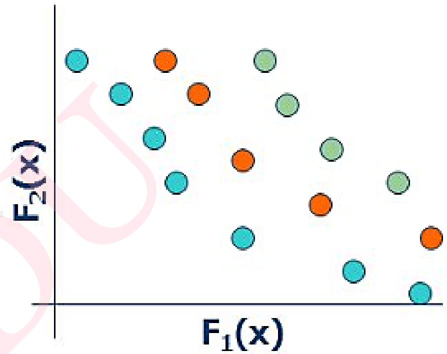
1）将所有个体复制给临时种群T

2）找出T中所有非支配个体，用集合B表示这些个体并为它们分配最低的适应度值

3）从T中去掉B，并在更新后的T中继续找出所有的非支配个体，为这些个体分配次低的适应度值

4）重复上述过程，直到T中没有个体

基于Pareto的进化算法

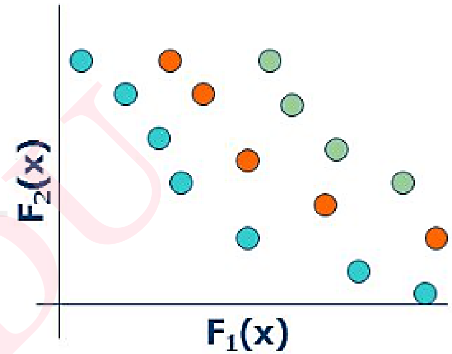


- 非支配排序遗传算法
  - 求解有k个目标的优化问题的NSGA伪代码

初始化候选解种群 $P = \{\boldsymbol{x}_j\}, j \in [1, N]$
While not (终止准则)
    临时种群 $T \leftarrow P$
    非支配水平 $c \leftarrow 1$
    While $|T| > 0$
        $B \leftarrow T$ 中非支配个体
        费用 $\phi(\boldsymbol{x}) \leftarrow c$ 对所有 $\boldsymbol{x} \in B$
        从 T 中去掉 B
        $c \leftarrow c + 1$
    下一个非支配水平
    $C \leftarrow$ 由 P 中个体重组生成的 N 个子代
    依概率变异 C 中的子代
    $P \leftarrow C$
下一代

- 计算全部N个个体所有k个适应度函数的值
- 将个体复制给临时种群T
- 为所有非支配个体分配适应度值为1
- 将这些个体从T中去除，找出缩减后的集合T中所有的非支配个体，为它们分配适应度值为2
- 重复上述过程直至每一个个体都分配到了适应度值
- 利用适应度值选择个体，交叉、变异

基于Pareto的进化算法



- 非支配排序遗传算法
  - 非支配排序的时间复杂度：

假设种群大小为N，优化目标个数为M

1）若要在种群中找出第一层非支配个体，对于每个个体，需要和其他N-1个个体进行比较，需要$O(MN)$次比较，那么总共需要$O(MN^2)$次比较；

2）找出第二层非支配个体，最坏的情况：$O(MN^2)$

3）最坏的情况，每一层只有1个个体，那么总共需要$O(MN^3)$次比较

  - 同一层的个体优劣度如何判断？

基于Pareto的进化算法

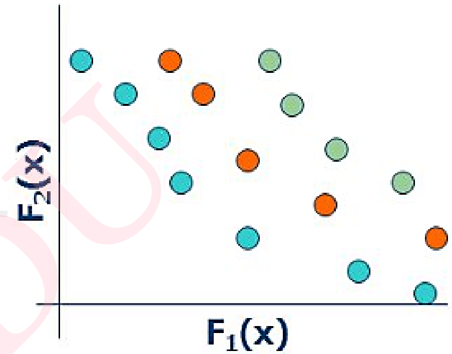
- 改进的非支配排序遗传算法
  - Nondominated sorting genetic algorithm-II, NSGA-II
  - 迄今为止最经典的进化多目标优化算法（IEEE Transactions on Evolutionary Computation, 2002, 6(2): 182-197）
  - 截止至2020年6月， Google Scholar: 被引32000余次； Web of Science: 被引19000余次
  - 针对NSGA进行的3大改进之处：

1）快速非支配排序方法（fast nondominated sorting approach）

2）个体拥挤距离算子（crowding distance-based operator）

3）精英策略选择算子（elitism strategy-based selection operator）

基于Pareto的进化算法



- 改进的非支配排序遗传算法
  - 快速非支配排序方法：

对于集合P中的每一个个体p，计算以下两个值：

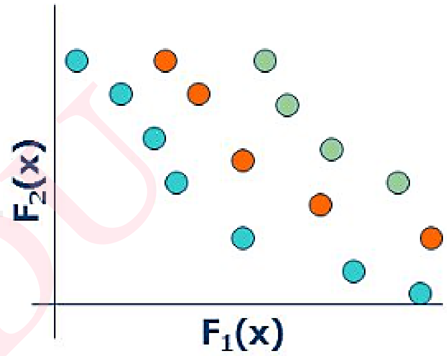
1）多少个体支配它？（支配该个体的个体数记为n_p）

2）它支配的个体集合（该集合记为S_p）

✓ 对于第一层的每个个体，n_p=0，访问该个体对应的S_p，并将其中的每个个体的n_p值减1；

✓ 若S_p中任一个体q的n_p值为0，则将q放入集合Q中，这些个体属于第二层

基于Pareto的进化算法

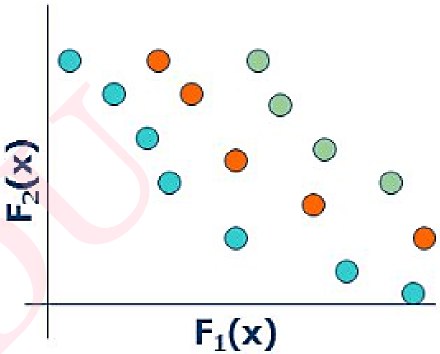


- 改进的非支配排序遗传算法
  - 快速非支配排序方法：

$\underline{\texttt{fast-non-dominated-sort}(P)}$

for each $p \in P$
    $S_p = \emptyset$
    $n_p = 0$
    for each $q \in P$
        if $(p \prec q)$ then — If p dominates q
            $S_p = S_p \cup \{q\}$ — Add q to the set of solutions dominated by p
        else if $(q \prec p)$ then
            $n_p = n_p + 1$ — Increment the domination counter of p
    if $n_p = 0$ then — p belongs to the first front
        $p_{\text{rank}} = 1$
        $\mathcal{F}_1 = \mathcal{F}_1 \cup \{p\}$
$i = 1$ — Initialize the front counter
while $\mathcal{F}_i \neq \emptyset$
    $Q = \emptyset$ — Used to store the members of the next front
    for each $p \in \mathcal{F}_i$
        for each $q \in S_p$
            $n_q = n_q - 1$
            if $n_q = 0$ then — q belongs to the next front
                $q_{\text{rank}} = i + 1$
                $Q = Q \cup \{q\}$
    $i = i + 1$
    $\mathcal{F}_i = Q$

基于Pareto的进化算法



- 改进的非支配排序遗传算法
  - 快速非支配排序的时间复杂度：

1）对于集合P中的每一个个体p，计算以下两个值：多少个体支配它？（支配该个体的个体数记为n_p），以及它支配的个体集合（该集合记为S_p），需要$O(MN^2)$次比较；

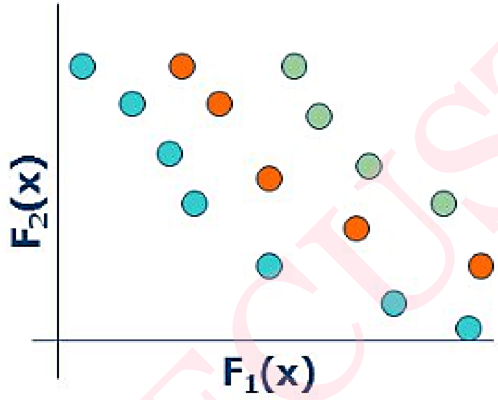
2）对于在第二层或更高层的每个个体p，n_p值最多为N-1，因此，在p的n_p值减为0时，需要被访问最多N-1次（此时p已被分至某个非支配层，此后不会再被访问），这一步的复杂度为$O(N^2)$

因此，快速非支配排序总的时间复杂度为$O(MN^2)$

基于Pareto的进化算法

- 改进的非支配排序遗传算法
  - 个体拥挤度距离算子：

解决问题：在相同非支配层的个体如何区分解的好坏？



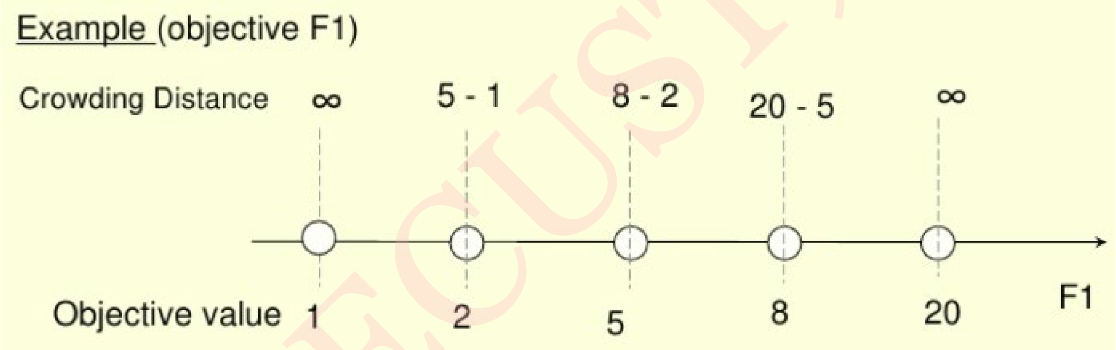
左图中的3层非支配层中，每一层中的个体是否有优劣？

基于Pareto的进化算法

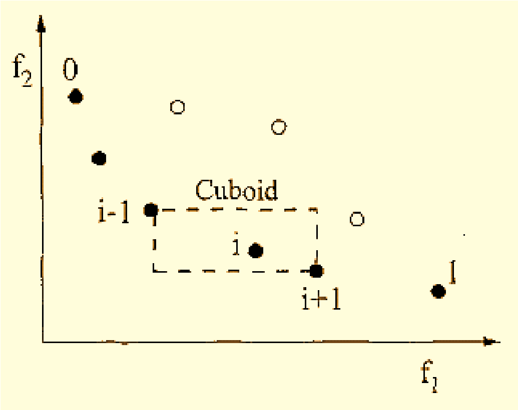
- 改进的非支配排序遗传算法

  – 个体拥挤度距离算子：

拥挤距离：计算某个个体的拥挤距离，即根据该个体每一目标函数计算个体两侧的两个个个体的距离



单个优化目标

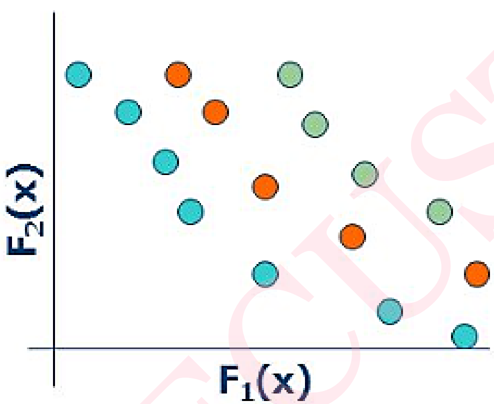


两个优化目标

基于Pareto的进化算法

- 改进的非支配排序遗传算法
  - 个体拥挤度距离算子：

解决问题：在相同非支配层的个体如何区分解的好坏？

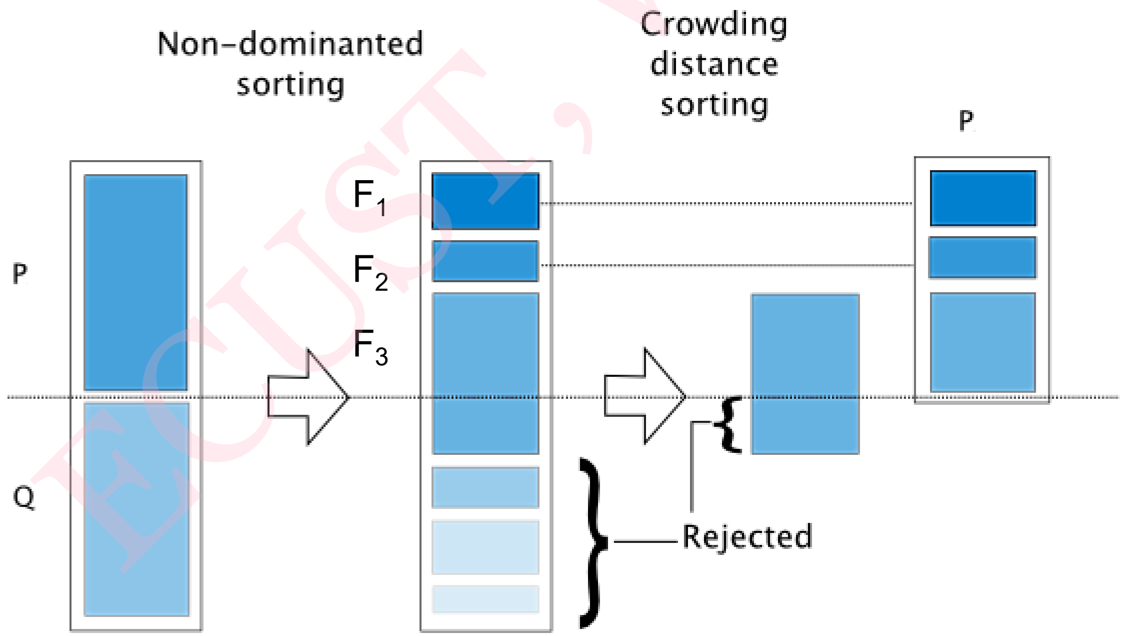


答：优先选择拥挤距离大的个体！

基于Pareto的进化算法

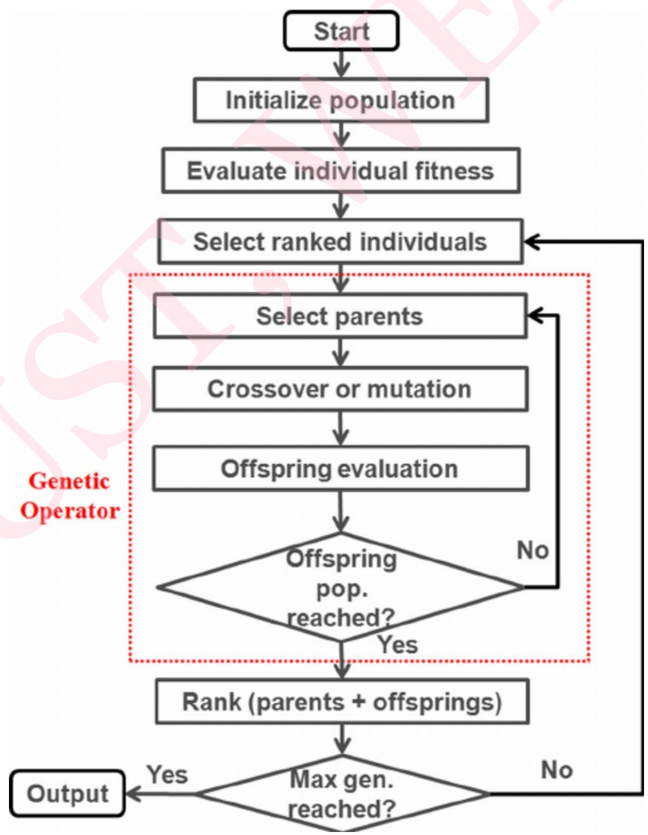
- 改进的非支配排序遗传算法
  - 精英策略选择算子：

保留父代中好的个体直接进入子代，加速收敛



基于Pareto的进化算法

- 改进的非支配排序遗传算法
  - NSGA-II算法框架



本章内容

1. 背景知识

2. Pareto最优性

3. 多目标优化的目标

4. 基于Pareto的进化算法

5. 总结

总结

- 多目标优化
  - 多目标优化问题
  - Pareto最优性
  - 进化优化求解多目标优化问题：进化多目标优化(Evolutionary Multi-objective Optimization, EMO)，多目标进化算法（Multi-objective Evolutionary Algorithm, MOEA）
  - 基于Pareto的、基于分解的、基于指标的
  - 多目标、高维多目标

结束