advantages of GA: ① conceptually simple ; ② do not require the search space to be: continuous, differentiable, unimodal

---

SGA example: max. $f(x)=x^2$, s.t. $0 \le x \le 31$

step 1: encoding the decision variable: $2^5=32$, use 5 bits

step 2: initialize the popln: randomly generate 4 solutions

01101 (13), 11000 (24), 01000 (8), 10011 (19)

step 3: fitness value

step 4: reproduction: use roulette wheel to obtain mating pool

step 5: Xover: select $\frac{n}{2}$ pair of parents. Use 1-point Xover, randomly generate Xover site k between (1, L-1). here $P_c = 1$.

step 6: mutation: assume $P_m = 0.001$. $20 \times 0.001 = 0.02$ ⇒ no change in the strings.

String #	initial popln	x	f(x)	%	# selected	Mating pool	Mate	Xover site
1	01101	13	169	14.4	1	01101	2	4
2	11000	24	576	49.2	2	11000	1	4
3	01000	8	64	5.5	0	11000	4	2
4	10011	19	361	30.9	1	10011	3	2
sum			1170	100.0	4			
avg			293					

⇒ new popln:

01100 . 11001

11011 . 10000

---

fitness scaling: in first few generations, highly fit solutions dominate the selection.

given +ve fitness $f(x)$. $g = af + b$, maintain $g_{avg} = f_{avg}$.

$g_{max} = C f_{avg}$ to restrict the fittest to have in mating pool.

$a = \frac{f_{avg}(C-1)}{f_{max} - f_{avg}}$, $b = f_{avg}(1-a)$, 若 $af+b<0$, 则 $a = \frac{f_{avg}}{f_{avg} - f_{min}}$, $b = f_{avg}(1-a)$

Sigma truncation: $f' = f - (f_{avg} - C \cdot \sigma)$, $C = 2 \sim 3$, σ is the variance

Power law scaling: $f' = g(f) = f^{\alpha}$

Coding Multivariable: if need 5 decimal point precision on x, then l is chosen as the smallest integer satisfying $10^5(b_i - a_i) \le 2^l - 1$

upper bound lower bound

decoding: $x_i = a_i + \text{decimal}(\text{string}_i)\left(\frac{b_i - a_i}{2^l - 1}\right)$

a population with n members and length l have schemas $2^l \sim n \cdot 2^l$ depending on diversity

(implicit parallelism)

4种 selection：① roulette wheel ② SUS ③ ranking ④ tournament

- Stochastic Universal Selection：wheel is rotated once with n markets on the wheel. 转一次 ⇒ 生成 n 个 strings (solutions)
- Ranking：the popln is sorted w.r.t. fitness value. each string is assigned with an offspring count.
  the selection proportion to the k-th ranked string is:
  $P_k = q_{max} - (\frac{k-1}{n-1})(q_{max} - q_{min})$, q_{max}, q_{min} are the allocation for the best and the worst string. $q_{max} + q_{min} = \frac{2}{n}$
- Tournament：随机选 2 个 string 进行 comparison，fitness 大的放进 mating pool，然后放回 popln. 重复 n 次选 n 个 solution 进行 Xover.

3种 Xover：① 1-point ② 2-point ③ uniform

- 1-point Xover cannot combine all possible schemas.
- 2-point Xover：example：11**.****.*1，****.1*1*.**
  $k_1 = 4$，$k_2 = 8$ ⇒ 11**1*1**1，**********
- uniform Xover：随机生成一个等长的 template，"1"的地方用爸的，"0"的用妈的
  example：1001011，0101101，template：1101001
  ⇒ offspring 1：1001101，offspring 2：0101011

4种 popln selection：① non-overlapping (全替换) ② elitist：用好的 parent 作为 offspring 进入下一个 G (只用最好的) ③ Steady-state reproduction：只用一部分 offspring 来代替 (保留一些 parent) ④ $\mu+\lambda$ selection：μ offspring + λ par

---

Building block：highly fit schemas (short $\delta(H)$, low order $O(H)$)

Fundamental Theorem：Short & low-order schemas with above popln average fitness will increase with exponential speed in subsequent Gs.

Two-armed Bandit Problem：N coins with $\mu_1, \mu_2, \sigma_1, \sigma_2$ of 2 arms
① 对每个 arm，先试 n 次，找出 best arm ② 把剩下的 $N-2n$ 个投入 best arm 中
③ a probability $q(n)$ that the identification is wrong.
expected loss：$L(N, n) = n|\mu_1 - \mu_2| + q(n)(N-2n)|\mu_1 - \mu_2|$
determine n to minimize L：$\frac{dL}{dn} = [1 - 2q(n) + (N-2n)\frac{dq(n)}{dn}] \cdot |\mu_1 - \mu_2| = 0$.
$n^* \approx c\ln(\frac{N^2}{8\pi c^2 \ln(N^2)})$，$c = (\frac{\sigma_1}{\mu_1 - \mu_2})^2$，$N \approx e^{\frac{n^*}{2c}}\sqrt{8\pi c^2 \ln(N^2)} \approx N - n = $ # of trials given to best arm.

Minimal-Deceptive Problem：

- Local Processing

① edge pixel (x_0, y_0) in neighborhood of (x, y) is smailar with:

1. magnitude: $|\nabla f(x,y) - \nabla f(x_0, y_0)| \le E$
2. direction: $|\theta(x,y) - \theta(x_0, y_0)| < A$

同时满足两个条件 ⇒ (x_0, y_0) 与 (x, y) linked

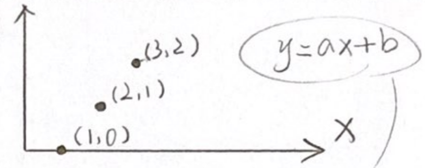
② Hough transform:

line segment $y = ax + b$ 经过某点 (x_i, y_i): $y_i = a x_i + b$

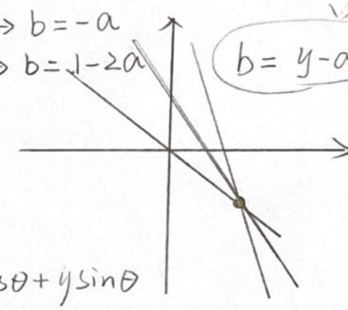
⇒ $b = y_i - a x_i$

极坐标表示: $\rho = x \cdot \cos\theta + y \cdot \sin\theta$

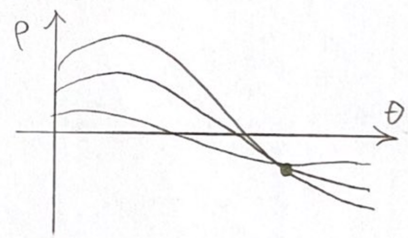
极坐标中的点 ⇔ (θ, ρ) 平面中的线
(共线) (共点)



$(1,0) \to b = -a$
$(2,1) \to b = 1-2a$



$\rho = x\cos\theta + y\sin\theta$



$(1,0) \to \rho = \cos\theta$
$(2,1) \to \rho = 2\cos\theta + \sin\theta$
$(3,2) \to \rho = 3\cos\theta + 2\sin\theta$

③ local dominant orientation:

gradient: 1st-order derivative of an image.

$$\nabla f(x,y) = \begin{bmatrix} G_x(x,y) \\ G_y(x,y) \end{bmatrix} = \begin{bmatrix} \partial f(x,y)/\partial x \\ \partial f(x,y)/\partial y \end{bmatrix}$$

* magnitude:

$$|\nabla f(x,y)| = [G_x^2(x,y) + G_y^2(x,y)]^{\frac{1}{2}}$$

* direction:

$$\varphi(x,y) = \tan^{-1}\left(\frac{G_y(x,y)}{G_x(x,y)}\right)$$

An image is modeled locally by

$$f(x,y) = h(ax + by)$$

$ax + by = t$ is the line representation of the orientation of $f(x,y)$.

$$y = -\frac{a}{b}x + t = \tan\theta\, x + t$$

$$\nabla f(x,y) = \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \partial f/\partial x \\ \partial f/\partial y \end{bmatrix} = \begin{bmatrix} a h'(t) \\ b h'(t) \end{bmatrix}$$

$$\nabla f(x,y) = G_x(x,y) + j G_y(x,y) = |\nabla f(x,y)| e^{j\theta(x,y)}$$

average the squared gradient: $[\nabla f(x,y)]^2 = G_x^2 - G_y^2 + j2 G_x G_y = |\nabla f|^2 e^{j2\theta}$

Discriminant Functions

$g_i(x) = \ln p(x|w_i) + \ln p(w_i)$,

decision boundary: $g_i(x) = g_j(x)$

- $p(x|w_i)$ 符合多元高斯分布:

$$p(x|w_i) = N(\mu_i, \Sigma_i)$$

则 $g_i(x) = -\frac{1}{2}(x-\mu_i)^T \Sigma_i^{-1}(x-\mu_i) + b_i$

$= -\frac{1}{2} d_\Sigma(x, \mu_i) + b_i$

$d_{\Sigma i} = (x, \mu_i) = (x-\mu_i)^T \Sigma_i^{-1}(x-\mu_i)$.

$b_i = \ln p(w_i) - \frac{1}{2}\ln|\Sigma_i|$.

- Mahalanobis distance:

$$d_{\Sigma i} = (x-\mu_i)^T \Sigma_i^{-1}(x-\mu_i)$$

- Euclidean distance:

$$d_{Eu} = (x-\mu_i)^T(x-\mu_i)$$

- In 1-D:

$$d_\Sigma = \frac{(x-\mu_i)^2}{\sigma^2}, \quad d_{Eu} = (x-\mu_i)^2$$

- Case 1: $\Sigma_j = \Sigma$ (协方差矩阵相同)

$$g_i(x) = -\frac{1}{2}(x-\mu_i)^T \Sigma^{-1}(x-\mu_i) + \ln p(w_i)$$

with same prior probability $p(w_i)$:

$$g_i(x) = -\frac{1}{2}(x-\mu_i)^T \Sigma^{-1}(x-\mu_i)$$
$$= d_\Sigma(x, \mu_i)$$

$$g_i(x) = -\frac{1}{2}x^T\Sigma^{-1}x + \mu_i^T\Sigma^{-1}x - \frac{1}{2}\mu_i^T\Sigma^{-1}\mu_i + \ln p(w_i)$$
$$= \mu_i^T\Sigma^{-1}x - \frac{1}{2}\mu_i^T\Sigma^{-1}\mu_i + \ln p(w_i)$$
$$= w_i^T x + w_{i0}$$

decision boundary:

$g_i(x) = g_j(x) \Rightarrow (w_i - w_j)^T x + (w_{i0} - w_{j0}) = 0$

$$(\mu_i-\mu_j)^T\Sigma^{-1}x - \frac{1}{2}(\mu_i-\mu_j)^T\Sigma^{-1}(\mu_i-\mu_j) + \ln\left(\frac{p(w_i)}{p(w_j)}\right) = 0$$

Example:

class w_1: $\begin{bmatrix}1\\2\end{bmatrix}, \begin{bmatrix}2\\2\end{bmatrix}, \begin{bmatrix}3\\1\end{bmatrix}, \begin{bmatrix}3\\2\end{bmatrix}, \begin{bmatrix}2\\3\end{bmatrix}$

class w_2: $\begin{bmatrix}7\\9\end{bmatrix}, \begin{bmatrix}8\\9\end{bmatrix}, \begin{bmatrix}9\\8\end{bmatrix}, \begin{bmatrix}9\\9\end{bmatrix}, \begin{bmatrix}8\\10\end{bmatrix}$

Find the Bayes classifier and the decision boundary.

$$\mu_1 = \frac{1}{5}\left(\begin{bmatrix}1\\2\end{bmatrix} + \begin{bmatrix}2\\2\end{bmatrix} + \cdots + \begin{bmatrix}2\\3\end{bmatrix}\right) = \frac{1}{5}\begin{bmatrix}11\\10\end{bmatrix}$$

$$\mu_2 = \frac{1}{5}\left(\begin{bmatrix}7\\9\end{bmatrix} + \begin{bmatrix}8\\9\end{bmatrix} + \cdots + \begin{bmatrix}8\\10\end{bmatrix}\right) = \frac{1}{5}\begin{bmatrix}41\\45\end{bmatrix}$$

$$\Sigma_j = \left(\frac{1}{N_j}\sum xx^T\right) - \mu_j\mu_j^T,$$

$$\Sigma_1 = \frac{1}{5}\left(\begin{bmatrix}1\\2\end{bmatrix}[1\ 2] + \cdots + \begin{bmatrix}2\\3\end{bmatrix}[2\ 3]\right) - \frac{1}{5}\times\frac{1}{5}\times\begin{bmatrix}11\\10\end{bmatrix}\times[11\ 10]$$
$$= \frac{1}{25}\begin{bmatrix}14 & -5\\-5 & 10\end{bmatrix}.$$

$$\Sigma_2 = \Sigma_1 = \Sigma.$$

$$\Sigma^{-1} = \frac{5}{23}\begin{bmatrix}10 & 5\\5 & 14\end{bmatrix}$$

with $p(w_1) = p(w_2)$, $\Sigma_1 = \Sigma_2 = \Sigma$:

$$d_j(x) = x^T\Sigma^{-1}\mu_j - \frac{1}{2}\mu_j^T\Sigma^{-1}\mu_j$$

$$\Sigma^{-1}\mu_1 = \frac{5}{23}\begin{bmatrix}10 & 5\\5 & 14\end{bmatrix}\times\frac{1}{5}\begin{bmatrix}11\\10\end{bmatrix} = \frac{1}{23}\begin{bmatrix}160\\195\end{bmatrix},$$

$$\mu_1^T\Sigma^{-1}\mu_1 = \frac{1}{5}[11\ 10]\times\frac{1}{23}\begin{bmatrix}160\\195\end{bmatrix} = \frac{742}{23}.$$

$$\Rightarrow d_1(x) = \frac{160}{23}x_1 + \frac{195}{23}x_2 - \frac{742}{46}$$

similarly, we obtain:

$$d_2(x) = \frac{635}{23}x_1 + \frac{835}{23}x_2 - 276.56$$

$d_1 - d_2 = 0 \Rightarrow$ decision boundary

Comprehensive Strategy:

① instead of learning from two exemplars (pbest & gbest), each dimension learns from just $pbest_{fi(d)}$

② instead of learning from pbest & gbest for all dim, each dim can learn from a differen exemplar.

VRS Implementation:

$$\min.\ f(X)$$
$$\text{s.t.}\ g_i(X) \le 0,\quad i = 1, \cdots, p$$
$$h_j(X) = 0,\quad j = 1, \cdots, m$$
$$l_k \le x_k \le u_k,\quad k = 1, \cdots, n$$

$$\min.\ f(X)$$
$$\text{s.t.}\ g_i(X) \le 0,\quad i = 1, \cdots, p$$
$$h_j(x_k \mid k \in C) = 0,\quad j \in M_2$$
$$l_k \le R_{k,j}(\{x_l \mid l \in \Omega_j, l \ne k\}) \le u_k,\quad k \in C_1, j \in M_1$$
$$l_k \le x_k \le u_k,\quad k \in C_2$$

Weeks 4~5

HCL-PSO: Diversity Measure

Swarm diversity: distance to average point

$$\text{Diversity}(S(t)) = \frac{1}{N} \sum_{i=1}^{N} \sqrt{\sum_{d=1}^{D} (x_i^d(t) - \overline{x^d(t)})^2},\quad \overline{x^d(t)} = \frac{1}{N} \sum_{i=1}^{N} x_i^d(t)$$

exploration subgroup g_1: high diversity

exploitation subgroup g_2: low diversity

- The explorative particles are not allowed to access information from exploitative particles.
- Even if exploitation groups suffer from premature convergence, exploration group can rescue them from local optimum.

Dynamic-Multiple-Swarm PSO (DMS-PSO)

- Constructed based on local version of PSO
- Two characteristics: ① small sized swarms, ② randomized re-grouping scheme
- Popln is divided into sereral sub-swarms randomly.
- Each sub use its own particles to search optimum.
- The whole popln is re-grouped into new sub-swarms.

minimal deceptive problem (MDP). $O(H) = 2$, $\delta(H) = 10 - 4 = 6$

	1	2	3	4	5	6	7	8	9	10	11
f_{00} :	*	*	*	0	*	*	*	*	*	0	*
f_{01} :	*	*	*	0	*	*	*	*	*	1	*
f_{10} :	*	*	*	1	*	*	*	*	*	0	*
f_{11} :	*	*	*	1	*	*	*	*	*	1	*

assume f_{11} is the global optimum

$f_{0*} > f_{1*}$, i.e., $\frac{f_{00} + f_{01}}{2} > \frac{f_{10} + f_{11}}{2}$

$\therefore f_{00} > f_{10}$, $f_{01} > f_{10}$

Type I : $f_{00} > f_{01}$: 凹 concave

Type II : $f_{00} < f_{01}$: 凸

Table : results of 1-point Xover (site is between fixed positions)

	00	01	10	11
00	same	same	same	01 10
01	same	same	00 11	same
10	same	11 00	same	same
11	10 01	same	same	same

- The proportions of schemas at time t are : P_{00}^t, P_{01}^t, P_{10}^t, P_{11}^t
- The probability that a Xover site falling between two fixed positions is $P_c \frac{\delta(H)}{l-1} = 1 \times \frac{6}{10} = 0.6$
- Deriving population proportions at time $t+1$:

consider schema "00", the proportion P_{00} always survive except :

(a) Xover with "11" with k between fixed position

(b) when "01" and "11" Xover between fixed position, then "00" is created

$$P_{00}^{t+1} = \frac{f_{00}}{\bar{f}} P_{00}^t - 0.6 \frac{f_{00}}{\bar{f}} P_{00}^t \frac{f_{11}}{\bar{f}} P_{11}^t + 0.6 \frac{f_{01} f_{10}}{\bar{f}^2} P_{01}^t P_{10}^t .$$

$\bar{f} = f_{00} P_{00}^t + f_{01} P_{01}^t + f_{10} P_{10}^t + f_{11} P_{11}^t$ is the average fitness.

00	00	$2 \frac{f_{00}}{\bar{f}} P_{00}^t \frac{f_{00}}{\bar{f}} P_{00}^t$
00	01	$\frac{f_{00}}{\bar{f}} P_{00}^t \frac{f_{01}}{\bar{f}} P_{01}^t$
00	10	$\frac{f_{00}}{\bar{f}} P_{00}^t \frac{f_{10}}{\bar{f}} P_{10}^t$
01	10	$0.6 \frac{f_{01}}{\bar{f}} P_{01}^t \frac{f_{10}}{\bar{f}} P_{10}^t$
00	10	$\frac{f_{00}}{\bar{f}} P_{00}^t \frac{f_{11}}{\bar{f}} P_{11}^t - 0.6 \frac{f_{00}}{\bar{f}} P_{00}^t \frac{f_{11}}{\bar{f}} P_{11}^t$

the parent schemas will survive if the Xover site is outside fixed pos.

Neighbor Mutation-based DE for Multi-modap Optimization

Weeks 6-7

Input population of solutions at current generation

Step 1 For $i = 1 : NP$ (NP is the popln size)

1. calculate the Euclidean distance between solution i and others
2. select m members with smallest E distance to solution i to construct a subpopln
3. produce an offspring u_i using DE within subpopln, i.e., pick r_1, r_2, r_3
4. reset u_i within bounds if any of the dimensions exceed the bounds
5. evaluate u_i using fitness value

End for

Step 2 Select NP fitter solutions for next G w.r.t. niching strategy

Output popln of solutions for next G

Difference between multi-obj and single-obj optimization:

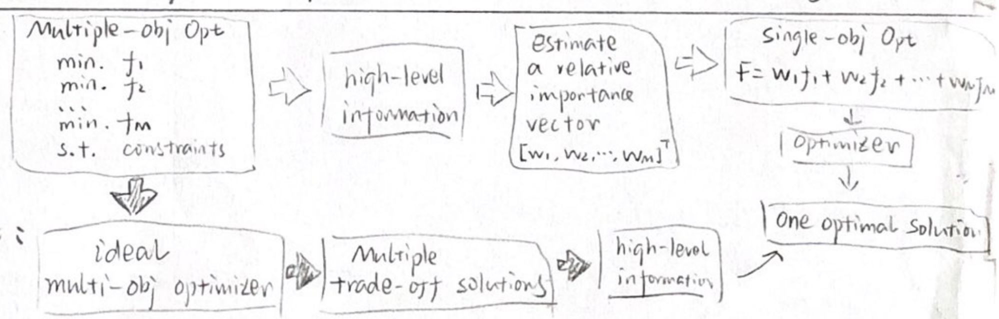
① one optimum versus multiple optima ② require search and decision making
③ two spaces-of-interest (objective space & decision space), instead of one

---

Weighted-Sum:

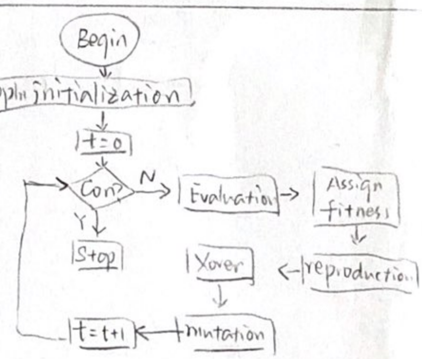
M obj

Ideal-Multi-obj-Opt:



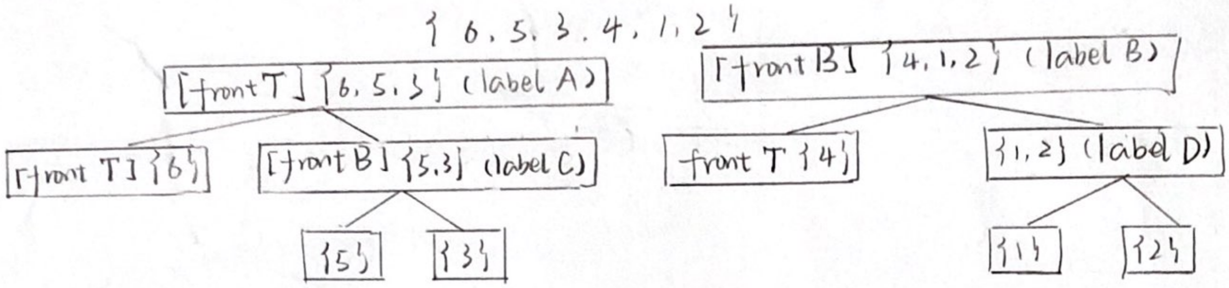
Changes of using SGA to solve multi-obj Opt:

① Modify the fitness computation
② Emphasize non-dominated solutions for convergence
③ Emphasize less-crowded solutions for diversity



Example of Kung Algorithm:

- Step 1: sort the popln w.r.t. the 1st obj in descending order: P = {6, 5, 3, 4, 1, 2}
- Step 2: the size of P' is not 1. divide recursively until it is 1:



- 5 & 3 do not dominate each other → {5, 3} at C
- 6, 5, 3 do not dominate each other → {6, 5, 3} at A
- 1 dominates 2 → discard 2 to have {1} at D
- 4 & 1 do not dominate each other → {4, 1} at B

- Step 3: when all the size are 1, work upward.
  finally we have 2 non-dominated sets {6, 5, 3} 和 {4, 1}
  in this exemple 4 & 1 are dominated by 3 & 5. → final set {6, 5, 3}

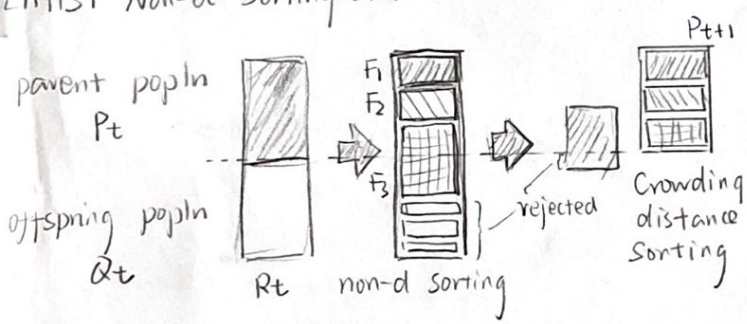
A simple Non-dominated Sorting: ($O(MN^2)$)

1. identify the best non-dominated set
2. discard them form popln
3. identify the next-best non-dominated set
4. Continue till all solutions are classified

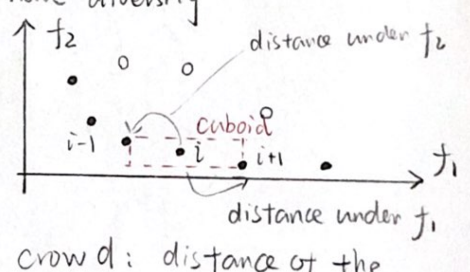
Vector-evaluated GA (VEGA)

1. divide popln into M equal blocks
2. each block is reproduced with 1 obj
3. compete solutions by Xover & mutation
4. bias towards solutions to best solution
5. A non-d selection: non-d solutions are assigned more copies
6. mate selection: 2 distance in parameter space are mated

Elitist Non-d Sorting GA (NSGA-II)



- Fronts F_1 & F_2 are chosen by elitism
- Only a part of F_3 can be retained in the popln of N
- Hence, choose diverse solutions in F_3 to have diversity



crowd: distance of the neighbor is small