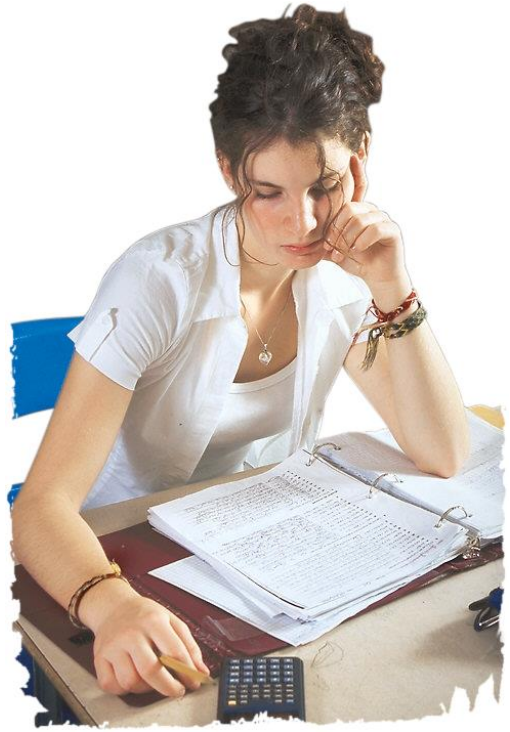
مدلسازی وبهسازی

دکتر پدرام پیوندی

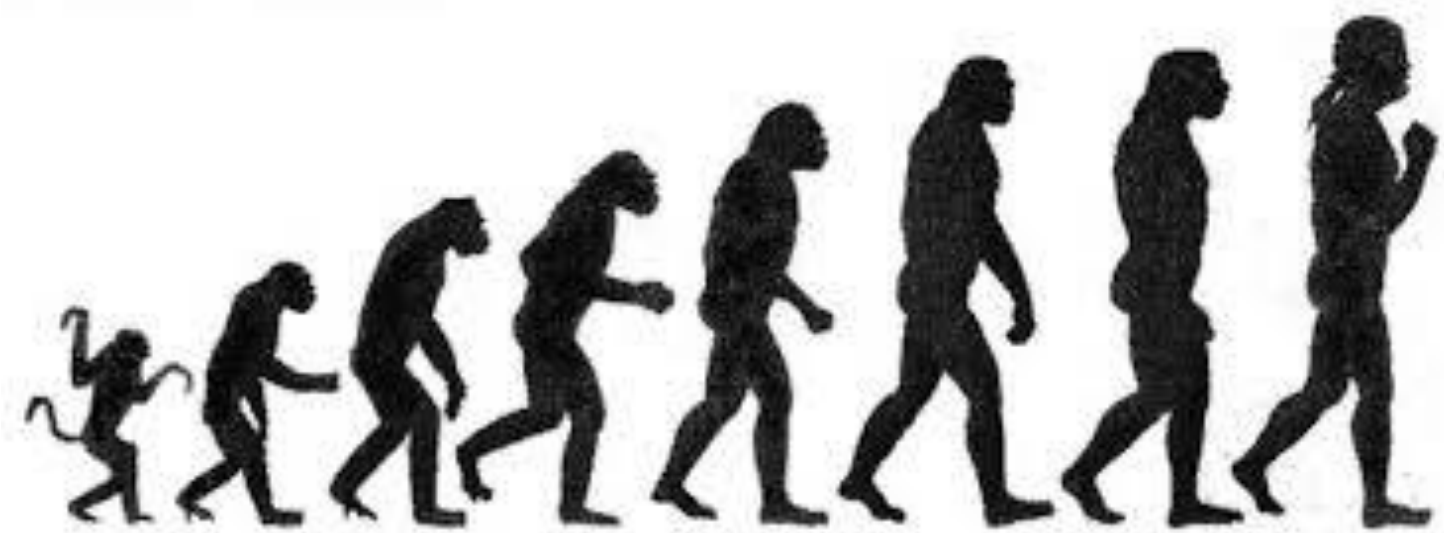
الگوریتم ژنتیک



DANGER
ARTIFICAL
INTELLIGENCE
AHEAD
www.says-it.com/safety

DANGER OF
ARTIFICIAL
INTELLIGENCE

HOW
ROBOTS
COULD
DESTROY
HUMANITY!



جهانی که در آن زندگی می کنیم ...

- به طور مداوم در حال تغییر است.
- هر موجودی که قصد ماندن در چنین محیطی را داشته باشد، بایستی بتواند خود را با شرایط اطراف تطبیق دهد.
- این فرآیند تطبیق، به نام‌های **تکامل** یا **فَرگشت** (Evolution) معروف است.

- موجودات زنده‌ای که امروزه در طبیعت مشاهده می‌کنیم، بیشترین تناسب و تطابق را با محیط زندگی‌شان دارند.
- به عنوان یک مثال بسیار شناخته شد، این موجودات را مقایسه کنید:

ژنتیک: دست خط طبیعت

- طبیعت برای آن که بتواند کار بهینه‌سازی را به خوبی انجام دهد، بایستی اطلاعات به دست آمده در طول میلیون‌ها سال را به نحوی ذخیره کند.
- زیربنای تشکیل دهنده موجودات زنده، عناصر شیمیایی هستند. لذا طبیعت از همین عناصر، برای ذخیره‌سازی اطلاعات مربوط به هر گونه زیستی، استفاده می کند.

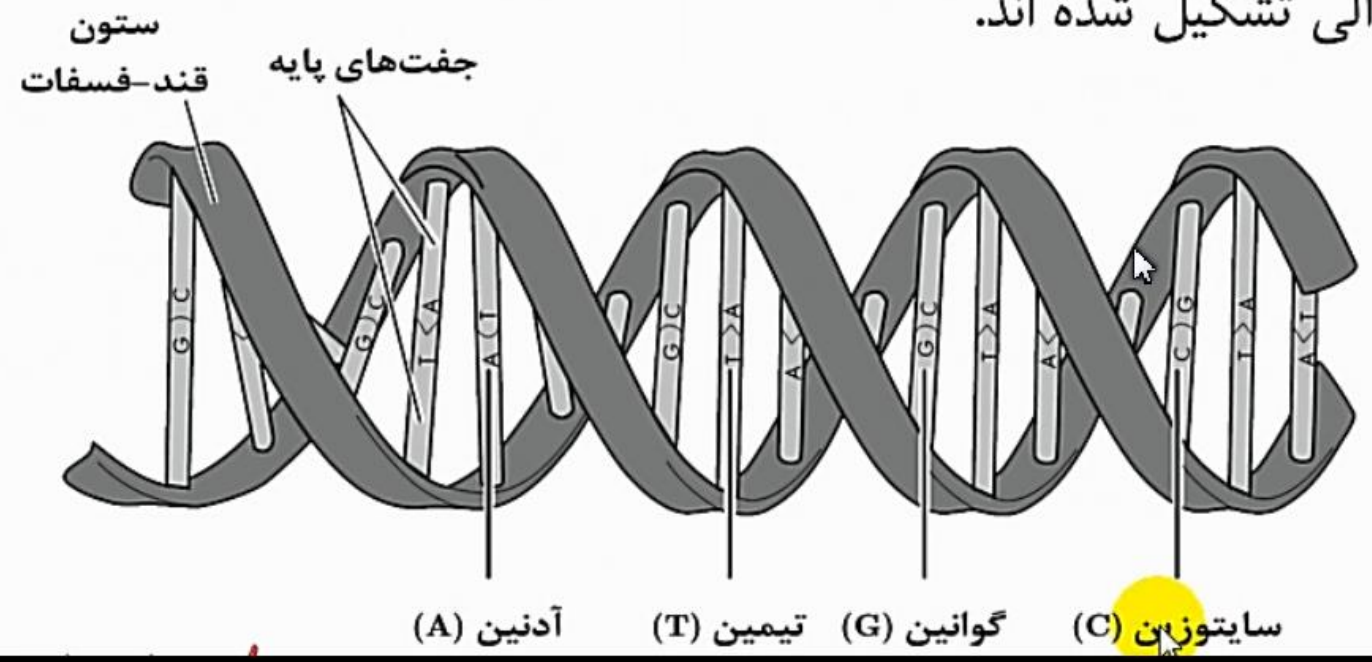
www.pedram-payvandy.com

امّا چگونه؟

- اطلاعات مربوط به هر موجود زنده، در یک ساختار پیچیده شیمیایی به نام DNA ذخیره شده‌اند.
- DNA حاوی تمام اطلاعات ضروری برای بازسازی یک موجود زنده است. لذا از DNA به عنوان تنظیم کننده اطلاعات وراثتی نیز یاد می‌شود.

ساختار DNA

مولکول DNA به شکل یک نردبان مارپیچ است که شاخه های اصلی آن از توالی قند و فسفات تشکیل شده اند. پایه های این نردبان، از اتصال چهار نوع باز آلی تشکیل شده اند.



ژن چیست؟

- اطلاعات مربوط به تمام موجودات زنده، در توالی و ترتیب‌های خاصی از ترکیب‌های شیمیایی معرفی شده در بخش قبل، ذخیره می شوند.
- ژن‌ها بخش‌های به خصوصی از ساختار DNA هستند که خواص و ویژگی‌های موجود زنده را تعریف می‌کنند.
- DNA انسان، شامل بیش از سه میلیارد عدد از ترکیبات پایه است که نوع و توالی بیش از ۹۹ درصد از این تعداد، برای تمام انسان‌ها مشترک می باشد. در واقع اطلاعات مربوط به هر فرد، در کمتر از یک درصد از ساختار DNA ذخیره سازی شده است.

www.pedram-payvandy.com

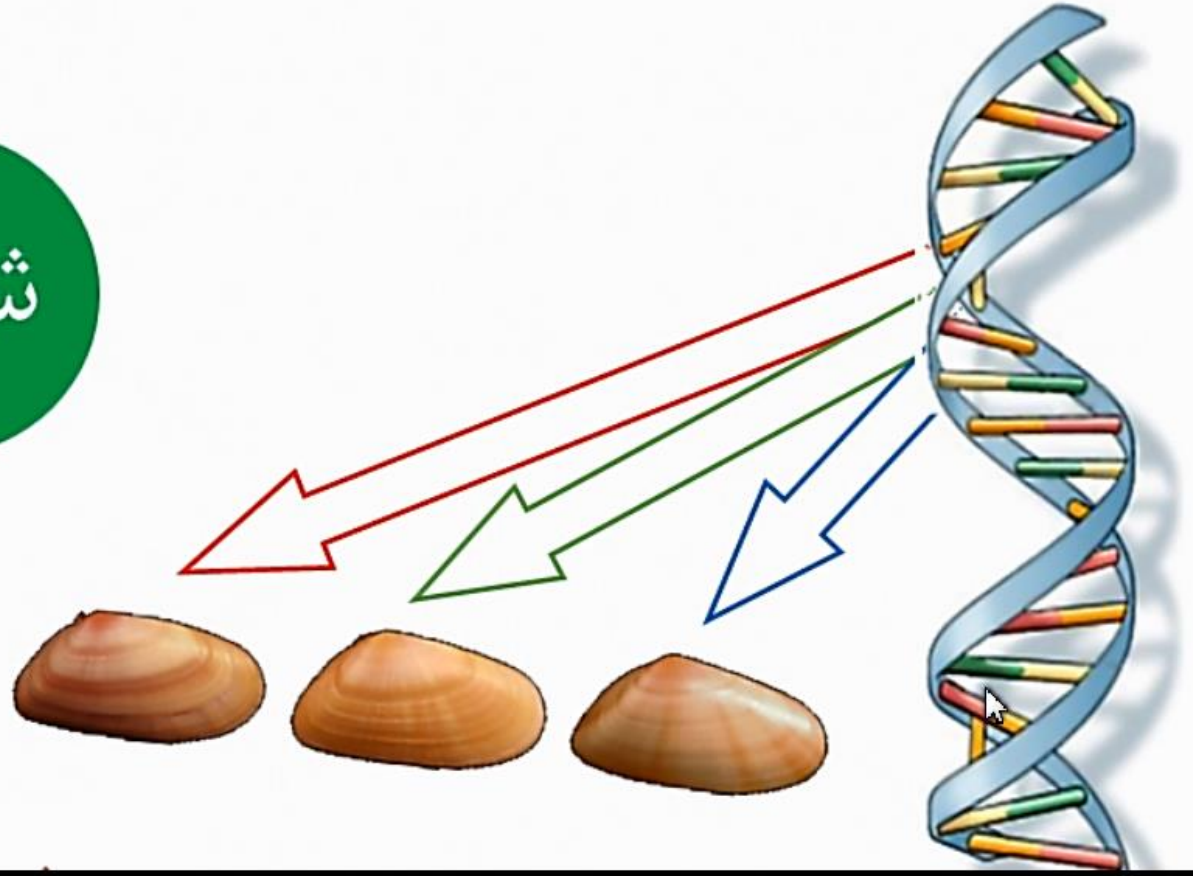
ژنتیک الفبای طبیعت است ...

ش

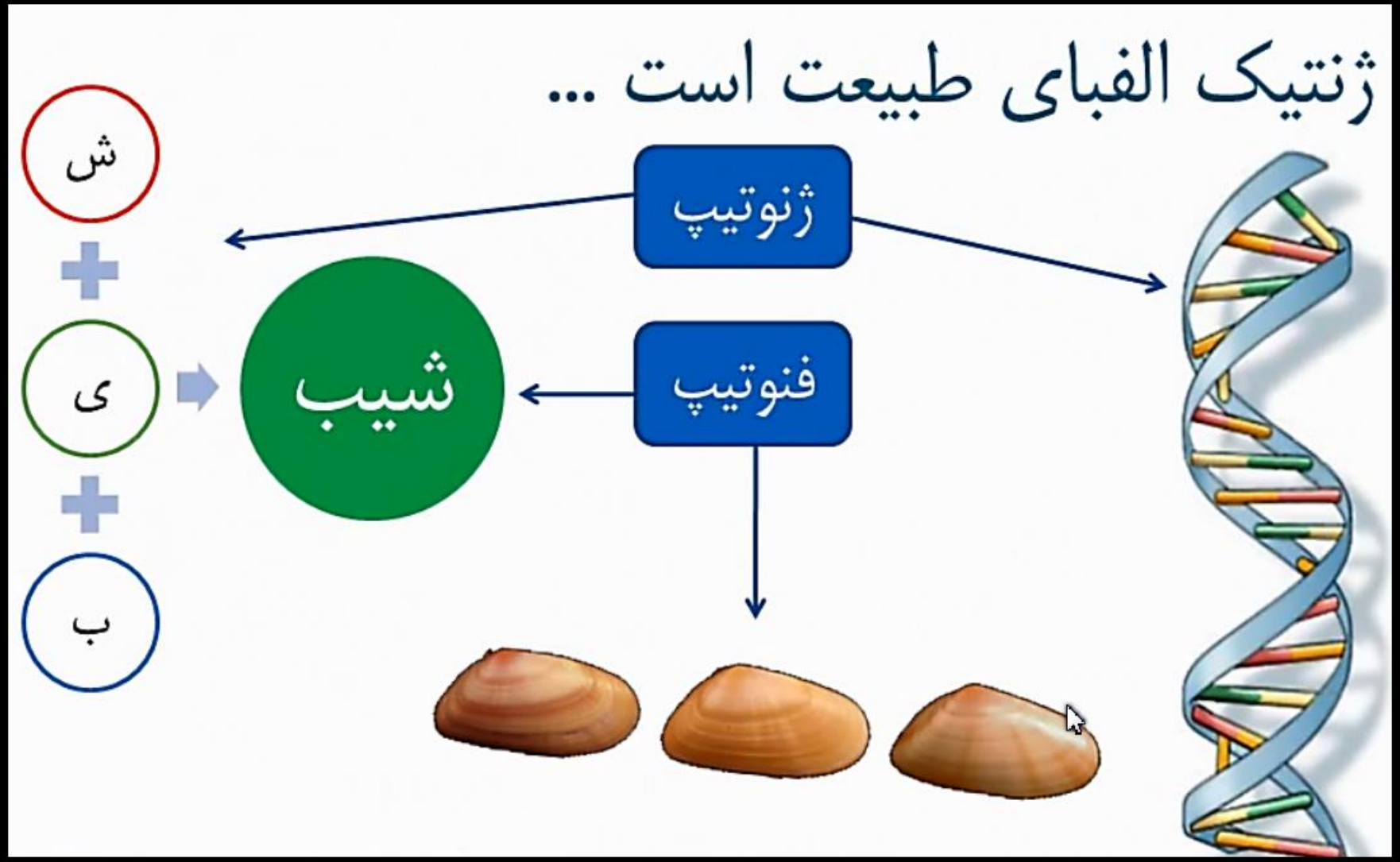
ی

ب

شیب



ژنتیک الفبای طبیعت است ...

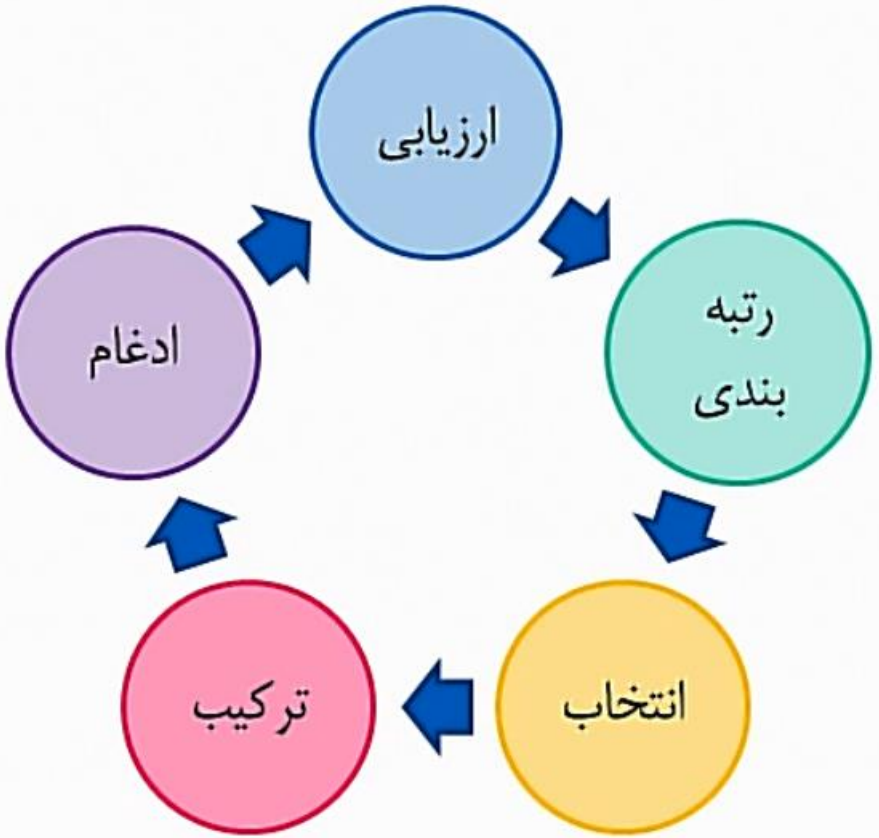


طرح اولیه یک الگوریتم تکاملی

مراحل الگوریتم تکاملی:

1. **ایجاد** مجموعه‌ای از جواب‌های تصادفی
2. **مقایسه** جواب‌ها، **رتبه‌بندی** آن‌ها و **انتخاب** بهترین ها
3. **ترکیب** جواب‌های به دست آمده، با شبیه‌سازی فرآیندهای طبیعی مانند تولید مثل، و ادغام جواب‌های جدید با جواب‌های قدیمی
4. **بازگشت** به مرحله ۲ (در صورت نیاز)

چرخه تکامل در یک الگوریتم تکاملی



الگوریتم ژنتیک

- از اوایل دهه ۱۹۵۰ میلادی، تلاش‌هایی برای شبیه‌سازی پدیده تکامل بر روی کامپیوترها آغاز شد، که در این میان توجه بسیاری از محققین حوزه‌های مربوط به علوم ریاضی و مهندسی، به این زمینه جلب شد. نهایتا در اوایل دهه ۱۹۷۰، **جان هالند**، در کتابش، الگوریتم ژنتیک را به عنوان ابزاری عمومی برای بهینه‌سازی، معرفی نمود.

GA : Genetic Algorithm

۱. ایجاد جمعیت تصادفی و ارزیابی آن ها

۲. انتخاب والدین و ترکیب آن ها برای ایجاد جمعیت فرزندان

۳. انتخاب اعضای جمعیت برای اعمال جهش و ایجاد جمعیت جهش یافتگان

۴. ادغام جمعیت اصلی، فرزندان و جهش یافتگان و ایجاد جمعیت اصلی جدید

۵. اگر شرایط خاتمه محقق نشده باشد، از مرحله ۲ تکرار می کنیم.

۶. پایان

Optimization بهینه سازی

Objective Function تابع هدف

Minimization کمینه سازی

Cost Func. تابع هزینه

Error Func. تابع خطا

Maximization بیشینه سازی

Fitness Func. تابع برازندگی

Profit Func. تابع سود

الگوریتم زنبک :

تعداد دفعات فراخوانی تابع هرف

NFE

$$= \text{تعداد جمعیت اصلی} + \left(\text{تعداد فرزندان} + \text{تعداد جهش یافتگان} \right) \times \text{تعداد تکرار}$$

روش های انتخاب والد

1) انتخاب تصادفی

2) انتخاب براساس شایستگی یا ربته

3) انتخاب رقابتی

```
function i=RouletteWheelSelection(P)

    r=rand;

    c=cumsum(P);

    i=find(r<=c,1,'first');

end
```

I : متغیر تصادفی مربوط به شماره والد انتخاب‌شده

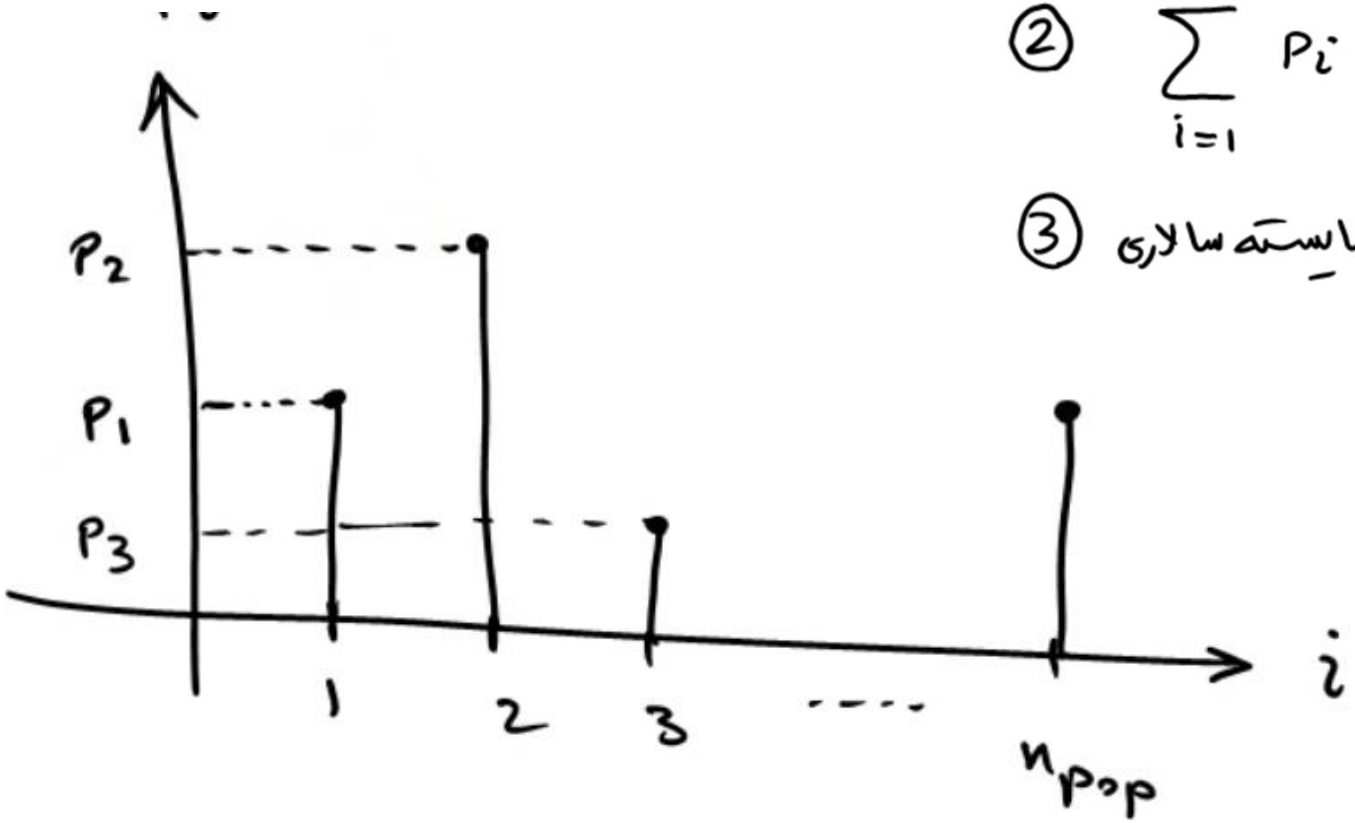
$$I \in \{1, 2, \dots, n_{pop}\} \qquad p_i = Pr\{I = i\}$$

① $0 \leq p_i \leq 1$

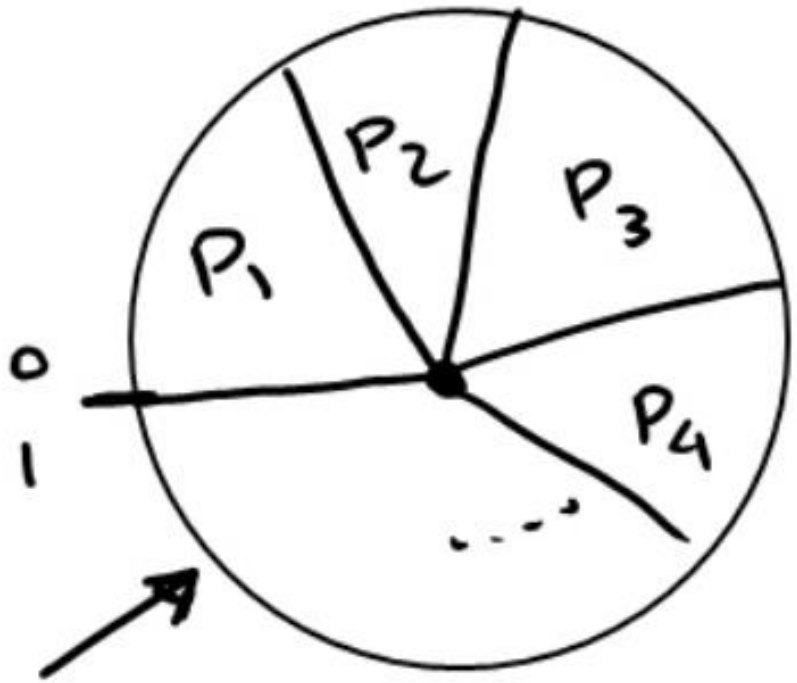
② $\sum_{i=1}^{n_{pop}} p_i = 1$

③ شایسته سالاری : $c_i \leq c_j \Leftrightarrow p_i \geq p_j$

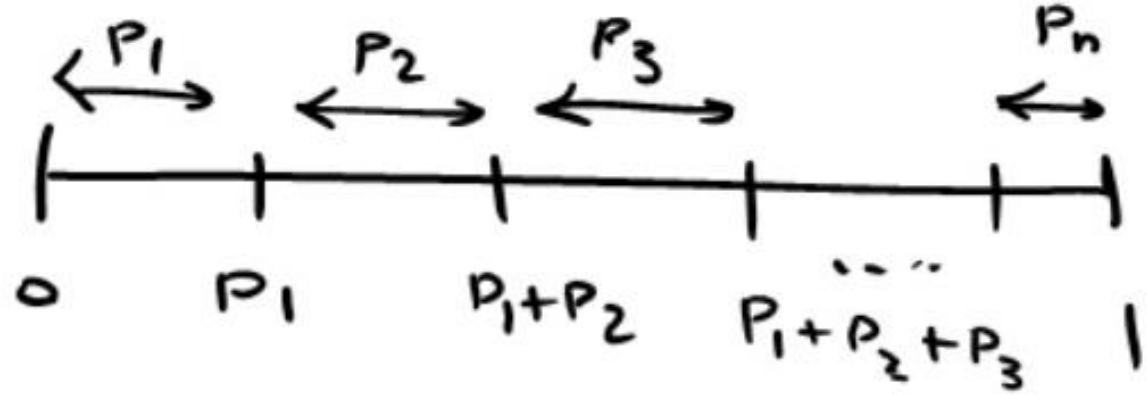
i بهتر از j باشد

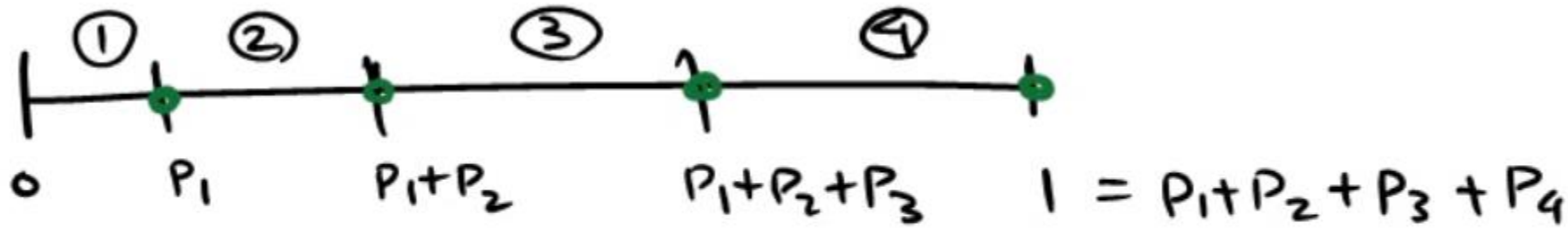


Roulette Wheel Selection (RWS)



$p_1 + p_2 + \cdots + p_n = 1$





$r \sim U(0,1)$ $\quad 0 \leq r \leq 1$

if ~~$r \geq 0$~~ & $r \leq P_1$ $\longrightarrow$ ①

if ~~$r > P_1$~~ & $r \leq P_1+P_2$ $\longrightarrow$ ②

if ~~$r > P_1+P_2$~~ & $r \leq P_1+P_2+P_3$ $\longrightarrow$ ③

if ~~$r > P_1+P_2+P_3$~~ & $r \leq 1$ $\longrightarrow$ ④

1. $r \sim U(0,1)$

RWS

2. $c_i = \sum_{j=1}^{i} p_j$

3. Find smallest i, where $r \leqslant c_i$.

$$i = \min \{ j \mid r \leqslant c_j \} \longrightarrow \text{find}$$

```
function i=TournamentSelection(pop,m)
    nPop=numel(pop);
    S=randsample(nPop,m);
    spop=pop(S);
    scosts=[spop.Cost];
    [~, j]=min(scosts);
    i=S(j);
end
```

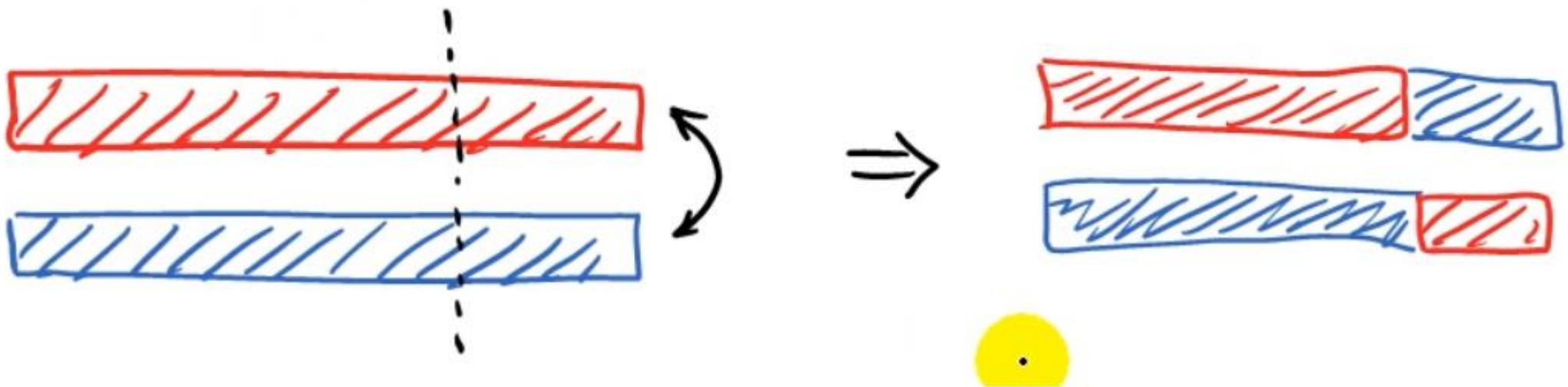
Tournament Selection

انتخاب رقابتی

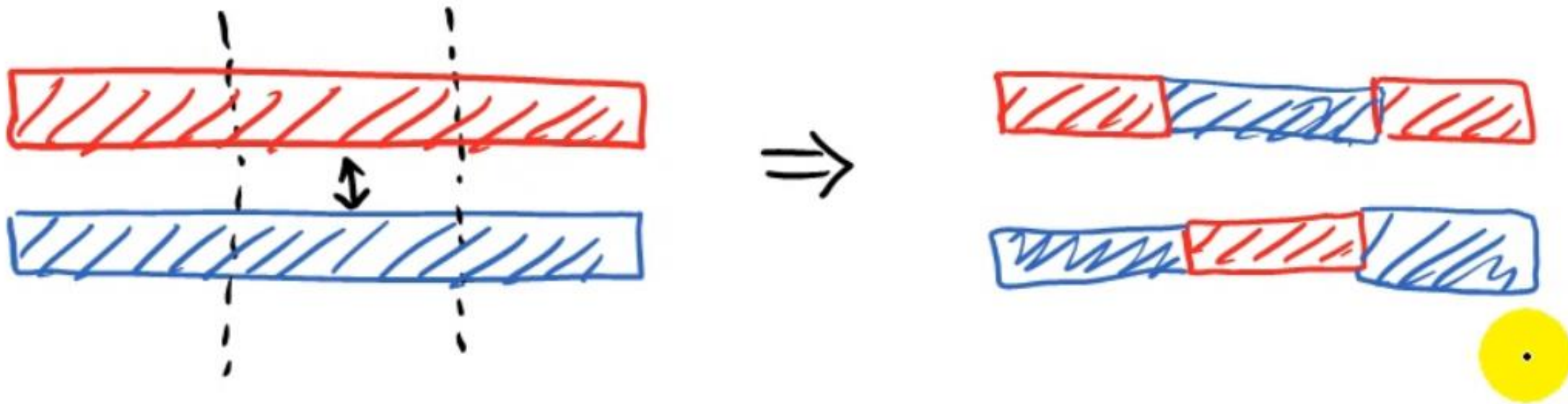
1. ابتدای m عضو از اعضای جمعیت به تصادف انتخاب می‌شوند (با احتمال یکسان)
2. بهترین عضو از میان m عضو منتخب، به‌عنوان نتیجه انتخاب رقابتی مشخص می‌شود.

Crossover :

① Single-Point Crossover



② Double-Point Crossover



③ Uniform Crossover

$x_1 = (x_{11}, x_{12}, \cdots, x_{1n})$

$x_2 = (x_{21}, x_{22}, \cdots, x_{2n})$

$\alpha = (\alpha_1, \alpha_2, \cdots, \alpha_n)$

$y_1 = (y_{11}, y_{12}, \cdots, y_{1n})$

$y_2 = (y_{21}, y_{22}, \cdots, y_{2n})$

$\alpha_i = 1 \rightarrow \begin{cases} y_{1i} = x_{1i} \\ y_{2i} = x_{2i} \end{cases}$

$\alpha_i = 0 \rightarrow \begin{cases} y_{1i} = x_{2i} \\ y_{2i} = x_{1i} \end{cases}$

$\alpha_i \in \{0, 1\}$

$y_{1i} = \alpha_i x_{1i} + (1-\alpha_i) x_{2i}$

$y_{2i} = \alpha_i x_{2i} + (1-\alpha_i) x_{1i}$

SPX احتمال $\rightarrow \pi_{spx}$

DPX احتمال $\rightarrow \pi_{dpx}$

UX احتمال $\rightarrow \pi_{ux}$

$\pi_{spx}+\pi_{dpx}+\pi_{ux}=1$

```
function [y1 y2]=SinglePointCrossover(x1,x2)

    nVar=numel(x1);

    c=randi([1 nVar-1]);

    y1=[x1(1:c) x2(c+1:end)];
    y2=[x2(1:c) x1(c+1:end)];

end
```

```
function [y1 y2]=UniformCrossover(x1,x2)

    alpha=randi([0 1],size(x1));

    y1=alpha.*x1+(1-alpha).*x2;
    y2=alpha.*x2+(1-alpha).*x1;

end
```

```
function [y1 y2]=DoublePointCrossover(x1,x2)

    nVar=numel(x1);

    cc=randsample(nVar-1,2);
    c1=min(cc);
    c2=max(cc);

    y1=[x1(1:c1) x2(c1+1:c2) x1(c2+1:end)];
    y2=[x2(1:c1) x1(c1+1:c2) x2(c2+1:end)];

end
```

$x_i \xrightarrow{\text{Mutate}} x_i^{new}$

$x_i^{new} \in \mathbb{X} - \{x_i\}$ ← از مجموعه $\mathbb{X}$ مقدار x_i را حذف می‌کنیم

به تصادف عضوی از مجموعه فوق را انتخاب می‌کنیم.

```
function y=Mutate(x,mu)

    nVar=numel(x);

    nmu=ceil(mu*nVar);

    j=randsample(nVar,nmu);

    y=x;
    y(j)=1-x(j);

end
```

مسائل اعداد حقیقی (پیوسته)

n_{var} : تعداد مجهولات

$x = (x_1, x_2, \ldots, x_{nvar})$ $\quad x_i \in [x_{min}, x_{max}]$

$x_{min} \leq x_i \leq x_{max}$

Arithmetic Crossover

$x_1 = (x_{11}, x_{12}, \cdots, x_{1n})$

$x_2 = (x_{21}, x_{22}, \cdots, x_{2n})$

$\alpha = (\alpha_1, \alpha_2, \cdots, \alpha_n)$ $\quad 0 \leq \alpha_i \leq 1$

$y_1 = (y_{11}, y_{12}, \cdots, y_{1n})$

$y_2 = (y_{21}, y_{22}, \cdots, y_{2n})$

$\longrightarrow$

$y_{1i} = \alpha_i x_{1i} + (1-\alpha_i) x_{2i}$

$y_{2i} = (1-\alpha_i) x_{1i} + \alpha_i x_{2i}$

```
function [y1 y2]=Crossover(x1,x2,gamma,VarMin,VarMax)

    alpha=unifrnd(-gamma,1+gamma,size(x1));

    y1=alpha.*x1+(1-alpha).*x2;
    y2=alpha.*x2+(1-alpha).*x1;

    y1=max(y1,VarMin);
    y1=min(y1,VarMax);

    y2=max(y2,VarMin);
    y2=min(y2,VarMax);

end
```

$$y_1 = \alpha x_1 + (1-\alpha) x_2$$

$$y_2 = \alpha x_2 + (1-\alpha) x_1$$

$0 \leq \alpha \leq 1$

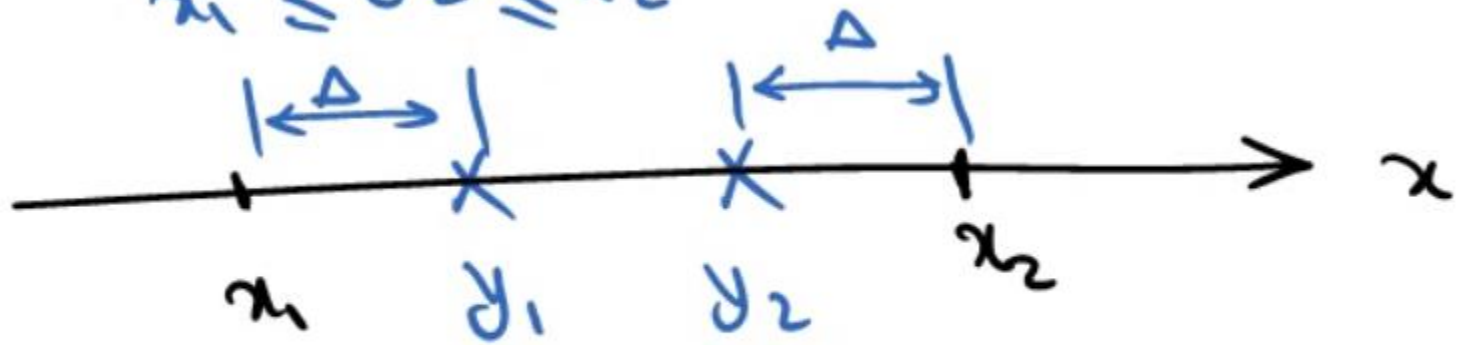
$x_1 \leq x_2$

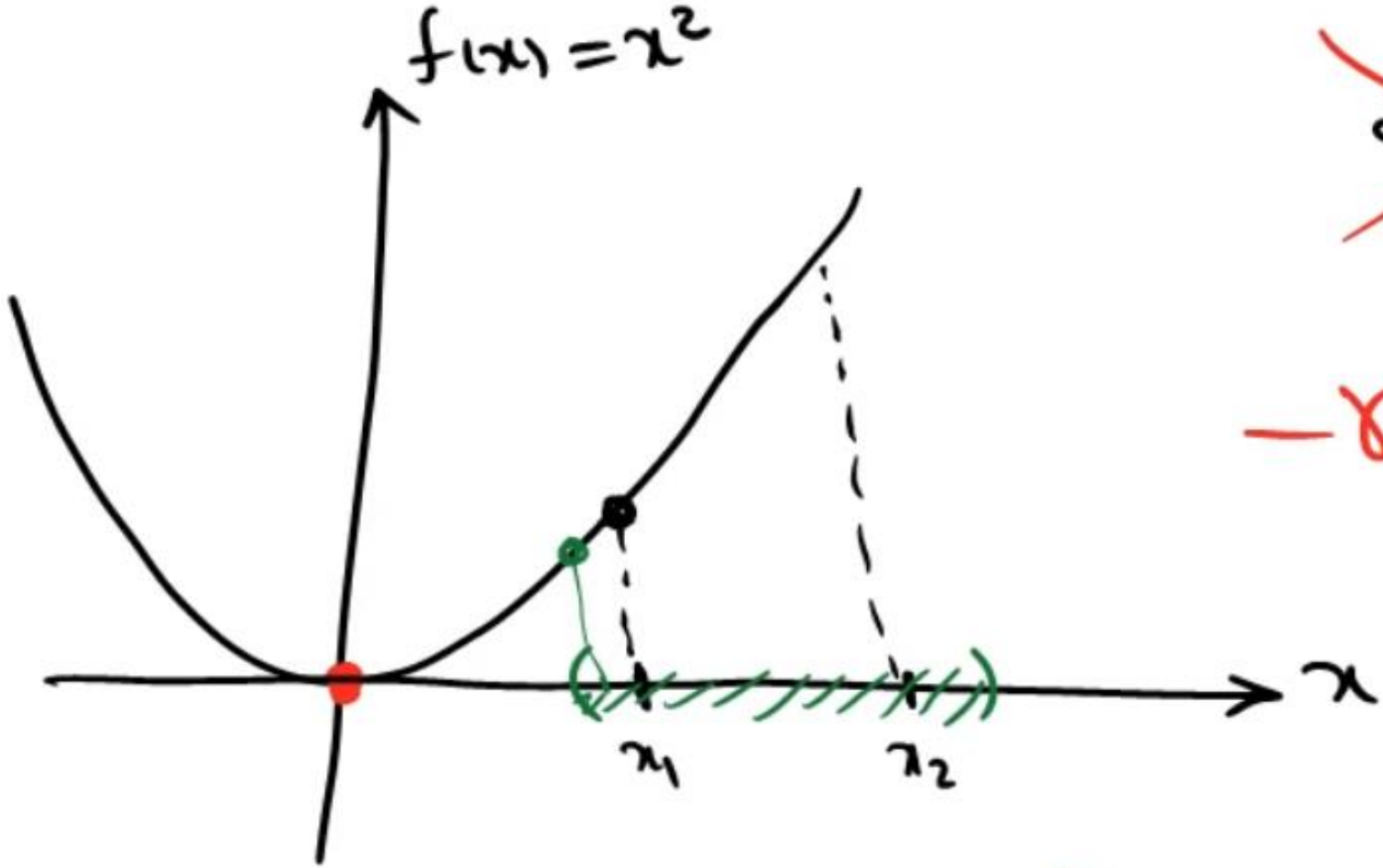
$x_1 \leq y_1 \leq x_2$

$x_1 \leq y_2 \leq x_2$

,

$y_1 + y_2 = x_1 + x_2$

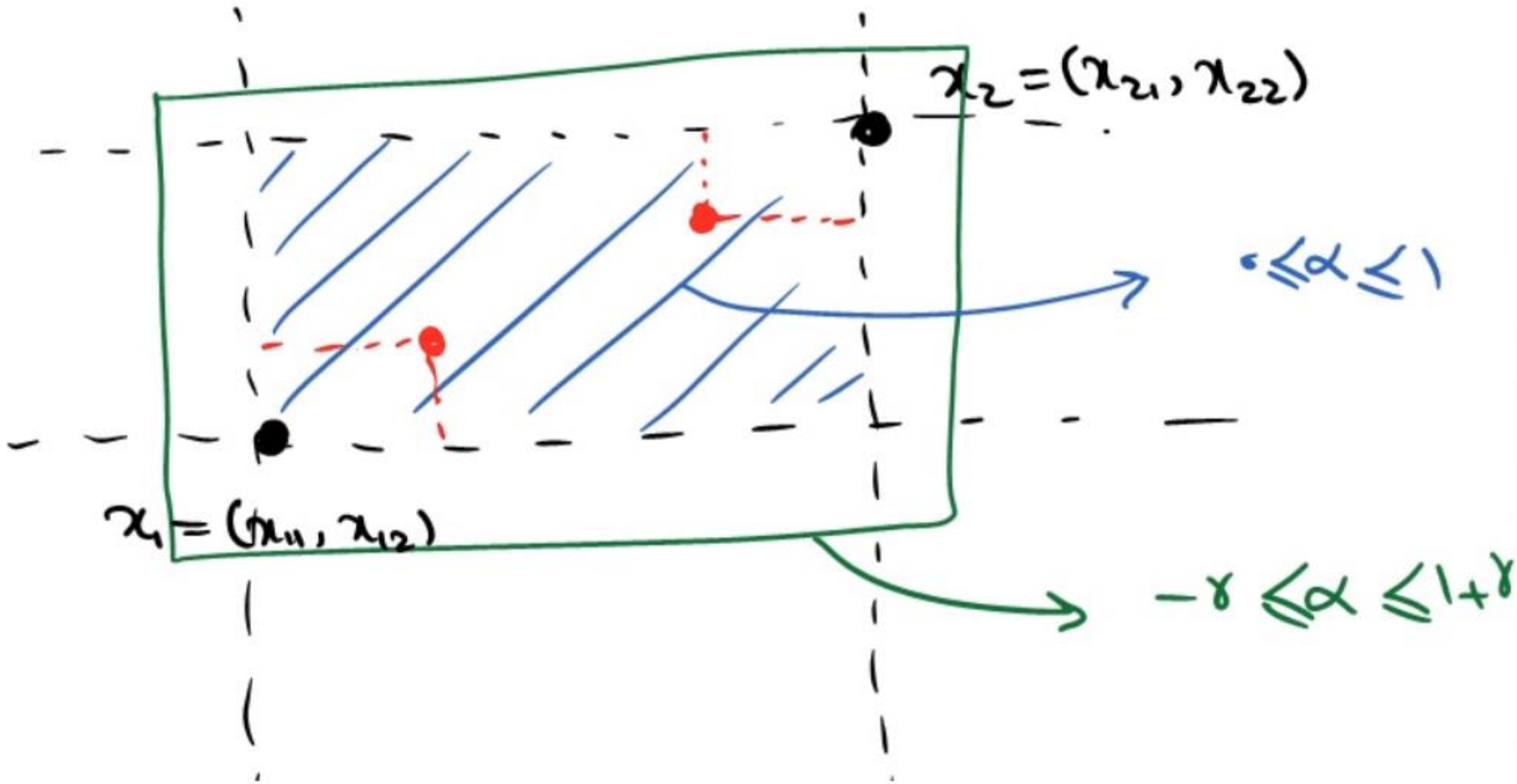




~~$0 \leq \alpha \leq 1$~~

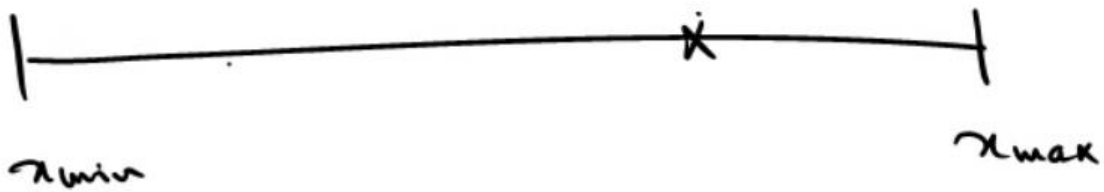
$-\gamma \leq \alpha \leq 1+\gamma$

$\gamma = 0.1 \longrightarrow -0.1 \leq \alpha \leq 1.1$


$x_2 = (x_{21}, x_{22})$
$0 \leq \alpha \leq 1$
$x_1 = (x_{11}, x_{12})$
$-\gamma \leq \alpha \leq 1+\gamma$

Mutation :

$$x_i^{new} \sim p(X) \quad X \in [x_{min}, x_{max}]$$




```
function y=Mutate(x,mu,VarMin,VarMax

    nVar=numel(x);

    nmu=ceil(mu*nVar);

    j=randsample(nVar,nmu);

    sigma=0.1*(VarMax-VarMin);

    y=x;
    y(j)=x(j)+sigma*randn(size(j));

    y=max(y,VarMin);
    y=min(y,VarMax);

end
```

$$x_i^{new} \sim N(x_i, \sigma^2)$$

$$\sim x_i + \sigma N(0,1)$$

$$\sigma = \mu (x_{max} - x_{min}) \qquad \mu = 0.1$$

Evolution Strategy (ES)

Schwefel's 1/5 rule:

* اگر نسبت جهش‌های موفق بیش از $1/5$ باشد، طول گام را بزرگتر کنید.
* اگر نسبت جهش‌های موفق کمتر از $1/5$ باشد، طول گام را کمتر کنید.

انواع شرایط خاتمه :

۱- رسیدن به حد قابل قبولی از پاسخ

۲- سپری شدن زمان / تکرار معین

۳- سپری شدن زمان / تعداد تکرار معین بدون مشاهده بهبود خاصی در نتیجه

شرایط خاتمه :