

# Soft Computing

---

H. Nezamabadi-pour  
Electrical Eng. Dept.,  
University of Kerman, Kerman, Iran.  
[nezam@mail.uk.ac.ir](mailto:nezam@mail.uk.ac.ir)

# LECTURE3: Real-valued GA

---

□ فرض کنید که هدف از بهینه‌سازی، پیدا کردن بیشینه تابع  $f$  در یک دامنه مشخص باشد:

$$f(x_1, x_2, \dots, x_m)$$

$$x_i^{lo} \leq x_i \leq x_i^{hi} \quad \text{for } i = 1, 2, \dots, m$$

□ در این وضعیت، پیدا کردن مقادیری برای متغیرهای  $x_1$  تا  $x_m$  مد نظر است که تابع  $f$ ، به ازای آنها بیشترین مقدار را به خود بگیرد.

□ به عبارتی هدف از بهینه‌سازی، یافتن  $X^*$  است به گونه‌ای که

$$f(X^*) \geq f(X) \quad , \quad \forall X \in D_f$$

---

# LECTURE3: Basic Concepts

---

- در حل مساله با الگوریتم وراثتي، هر یک از متغیرهاي این مساله بصورت یک ژن در وراثت طبیعی در نظر گرفته می‌شوند.
  - از کنار هم قرار گرفتن تمام متغیرهاي یک مساله (ژنها)، یک کروموزوم ساخته می‌شود.
  - هلد برای اولین بار از رشته‌هاي بیتی برای بیان اطلاعات کروموزومها استفاده کرد
  - ساختار نمایش بیتی برای ارائه متغیرهاي پیوسته چندان مناسب نیست
-

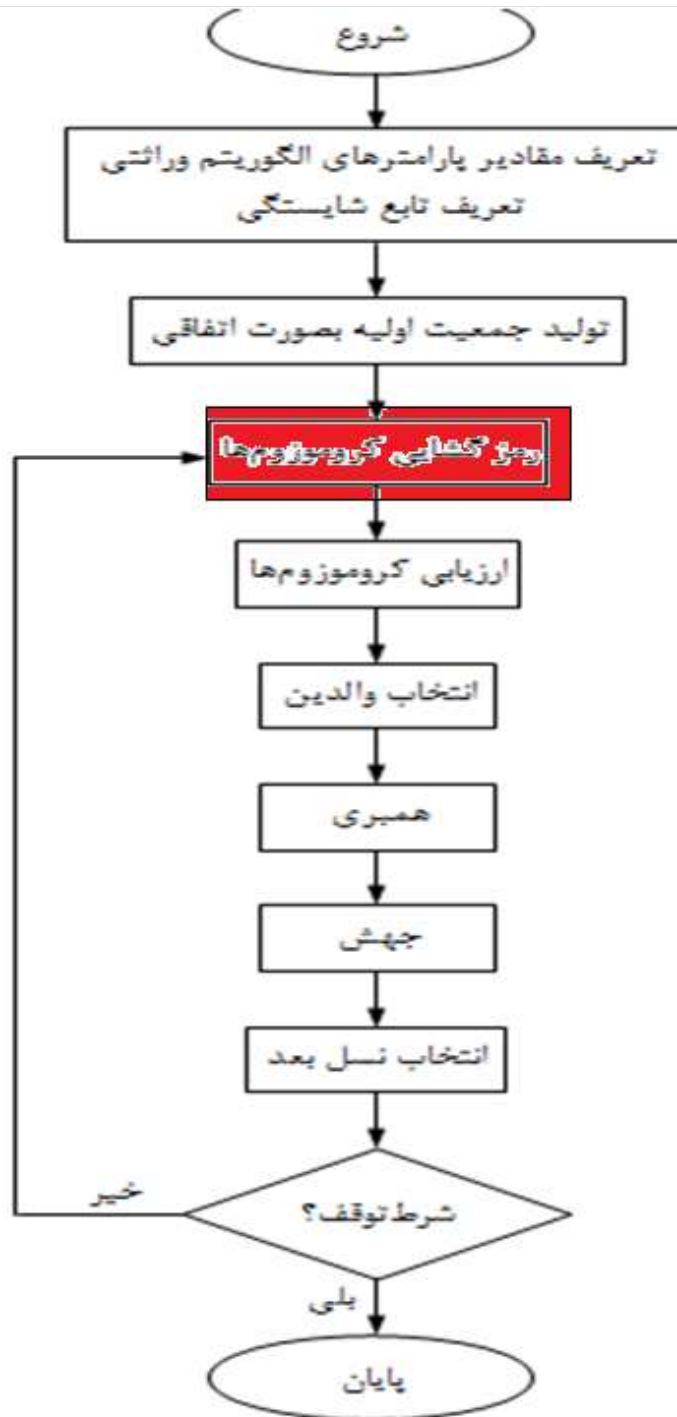
# LECTURE3: Basic Concepts

---

ساختار نمایش بیتی برای ارائه متغیرهای پیوسته چندان مناسب نیست

- جهشهای همبستگی یکی از مشکلات استفاده از رمز باینری برای متغیرهای پیوسته است
- مشکل دیگر وجود خطای چندبیتی است که الگوریتم باینری را در یافتن جواب بهینه با دقت دلخواه ناتوان میسازد. در الگوریتم وراثتی باینری برای افزایش دقت متغیرها و کاهش خطای چندبیتی، باید طول متغیرها برحسب بیت افزایش یابد. در حل مسأله‌های با تعداد متغیرهای زیاد، چنانچه طول متغیرها نیز زیاد در نظر گرفته شود، طول کروموزوم بیش از اندازه بزرگ خواهد شد. حل مسائل با طول کروموزومهای بزرگ، نیازمند آن است که اندازه جمعیت بزرگ در نظر گرفته شود. در این صورت، پیچیدگی محاسباتی الگوریتم افزایش قابل توجهی خواهد داشت.

# ساختار كلي الگوريتم وراثتي حقيقي



# مقدار دهی پارامترها

---

- ☐ تعیین تعداد اعضای جمعیت
  - ☐ نرخ همبري
  - ☐ نرخ جهش
  - ☐ تعداد متغیرها
  - ☐ طول کروموزوم
  - ☐ تعیین محدوده هر متغیر و دامنه جستجو
  - ☐ نحوه خاتمه الگوریتم
-

# نحوه نمایش کروموزوم ها

هر کروموزوم

$$[x_1 \quad x_2 \quad \dots \quad x_m]$$

جمعیت از تعدادی کروموزوم تشکیل شده است.

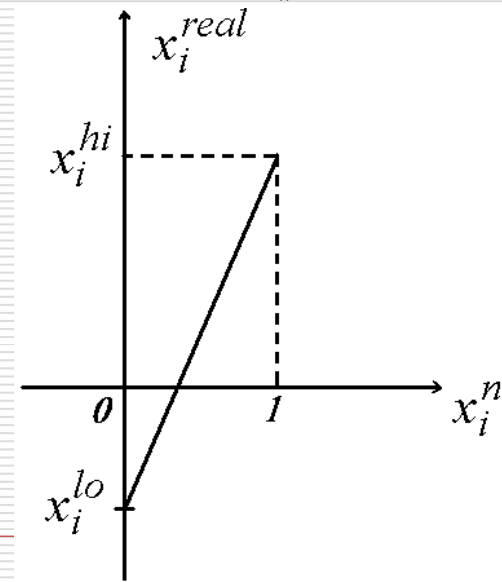
$$X = \begin{bmatrix} X_1 \\ X_2 \\ \vdots \\ \vdots \\ X_N \end{bmatrix} = \begin{bmatrix} x_1^1 & x_1^2 & \dots & \dots & x_1^m \\ x_2^1 & x_2^2 & \dots & \dots & x_2^m \\ & & \ddots & & \\ & & & \ddots & \\ x_N^1 & x_N^2 & \dots & \dots & x_N^m \end{bmatrix}$$

## تولید جمعیت اولیه

□ برای ایجاد جمعیت اولیه، کافی است که یک ماتریس با اندازه  $N \times m$  بطور اتفاقی با توزیع یکنواخت در بازه  $[0, 1]$  تولید شده، سپس هر یک از متغیرها به بازه‌ای از پیش تعریف شده نگاشت شوند. چنانچه متغیر  $i$  ام در بازه  $[x_i^{lo}, x_i^{hi}]$  تعریف شده باشد، نگاشت این متغیر با رابطه زیر امکان پذیر می‌شود.

$$x_i^{real} = x_i^{lo} + (x_i^{hi} - x_i^{lo}) \times x_i^n$$

for  $i = 1, 2, \dots, m$





# Initialization

## □ جمعیت اولیه

موقعیت اولیه هر کروموزوم به صورت تصادفی در فضای جستجو با یک توزیع یکنواخت در محدوده تعریف مسئله تعیین می شود  
 $X_{\min}^d$  و  $X_{\max}^d$  به ترتیب حد پایین و حد بالا از بعد  $d$  ام فضای جستجو می باشند.

$$x_i^d(t=1) \sim U(x_{\min}^d, x_{\max}^d)$$

$$x_i^{real} = x_i^{lo} + (x_i^{hi} - x_i^{lo}) \times x_i^n \quad \text{for } i = 1, 2, \dots, m$$

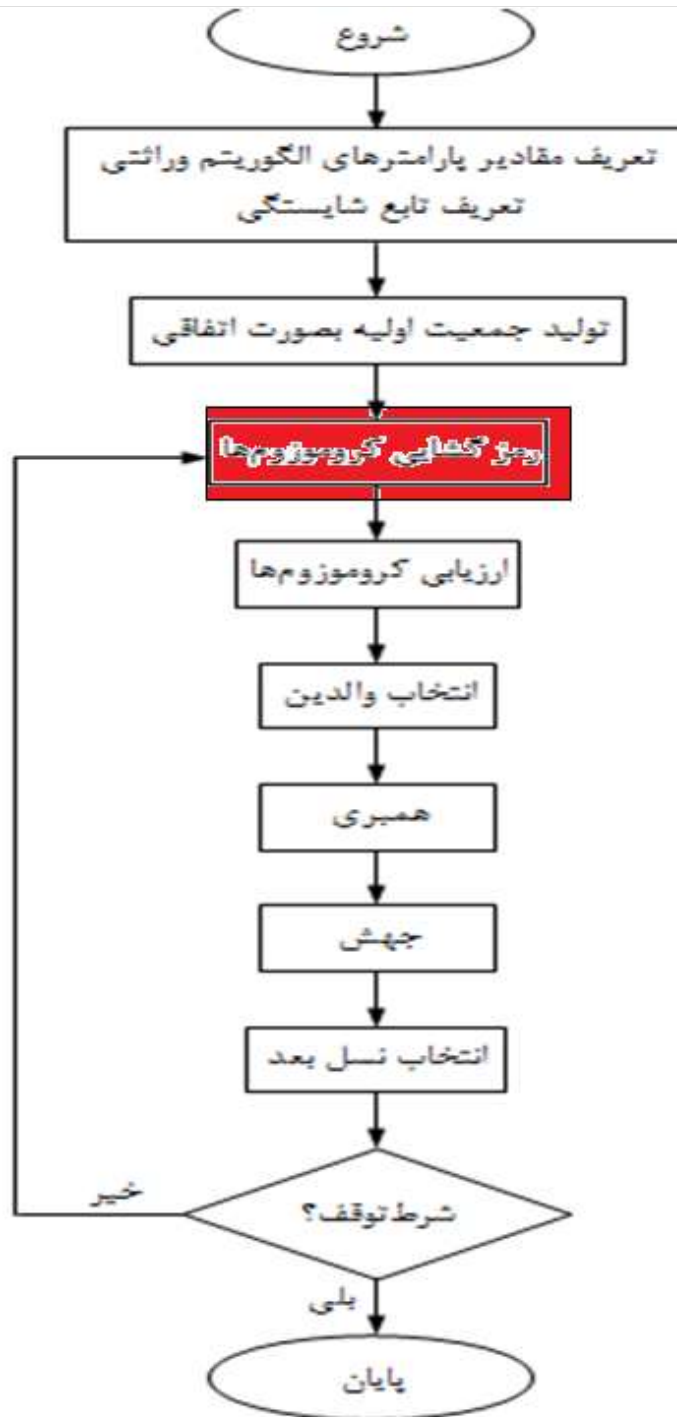
$$X = \begin{bmatrix} X_1 \\ X_2 \\ \vdots \\ \vdots \\ X_N \end{bmatrix} = \begin{bmatrix} x_1^1 & x_1^2 & \dots & \dots & x_1^m \\ x_2^1 & x_2^2 & \dots & \dots & x_2^m \\ & & \ddots & & \\ & & & \ddots & \\ x_N^1 & x_N^2 & \dots & \dots & x_N^m \end{bmatrix}$$

$$X = \begin{bmatrix} X_1 \\ X_2 \\ \vdots \\ \vdots \\ X_N \end{bmatrix} = \begin{bmatrix} 0.8 & 3 & 0.1 & \dots & 10 \\ 1.2 & 2.9 & 0.3 & \dots & 8 \\ & & \vdots & & \\ & & \vdots & & \\ & & \vdots & & \\ 0.7 & 3.2 & 0.2 & \dots & 11 \end{bmatrix}$$

# ارزیابی کروموزومها

□ در این مرحله به ازای هر یک از بردارهای ورودی، مقدار تابع شایستگی محاسبه می شود. برای این کار، مقادیر بدست آمده برای متغیرها در تابع شایستگی قرار می گیرد. خروجی تابع شایستگی به ازای هر دسته از متغیرهای ورودی به عنوان شایستگی کروموزوم مربوط به آن در نظر گرفته می شود.

# ساختار کلی الگوریتم وراثتی حقیقی



# عملگر انتخاب

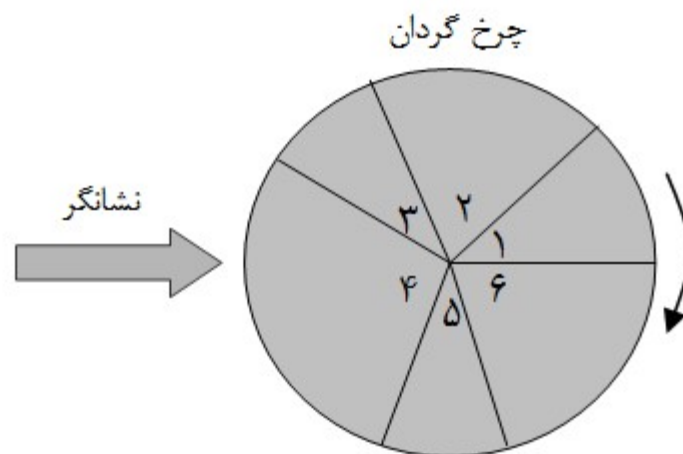
---

□ مانند الگوریتم وراثتی باینری

---

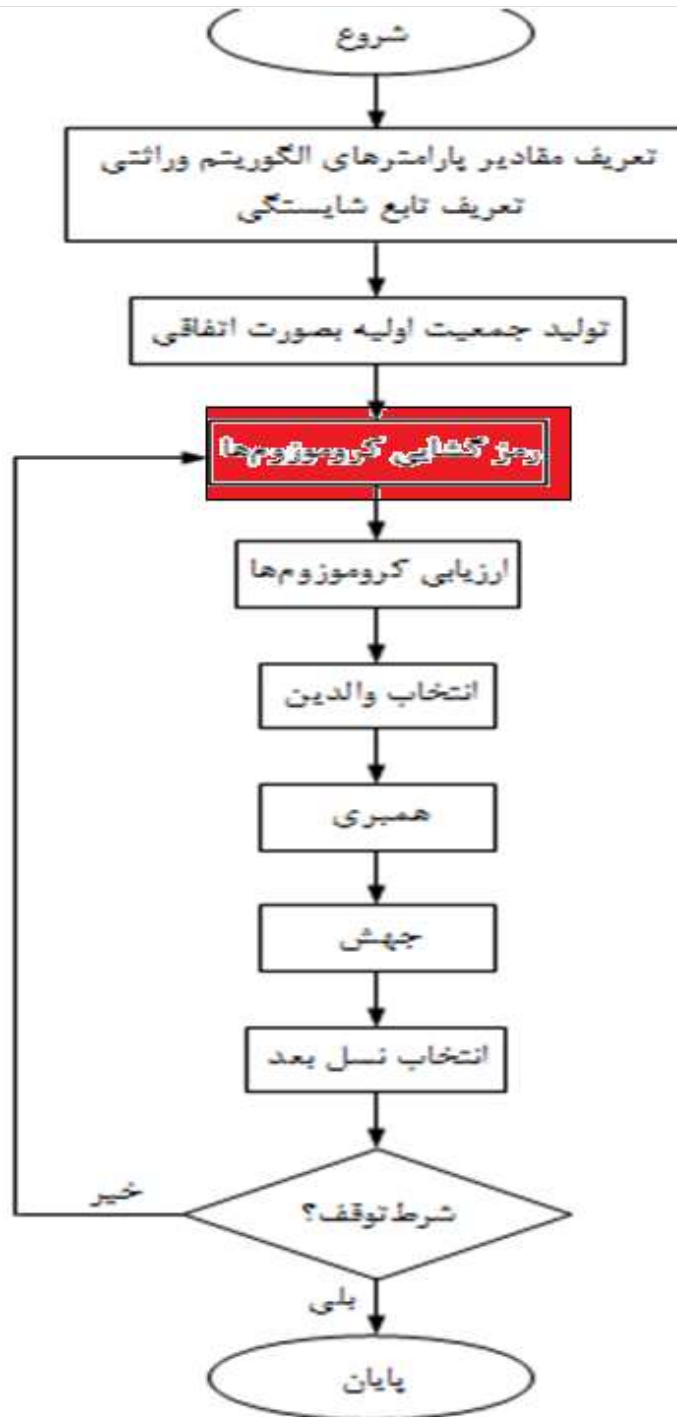
# عملگر انتخاب: چرخ گردان

$$P_i = \frac{fit_i}{\sum_{j=1}^N fit_j}$$



$$EV_i = P_i \times N$$

# ساختار كلي الگوريتم وراثتي حقيقي

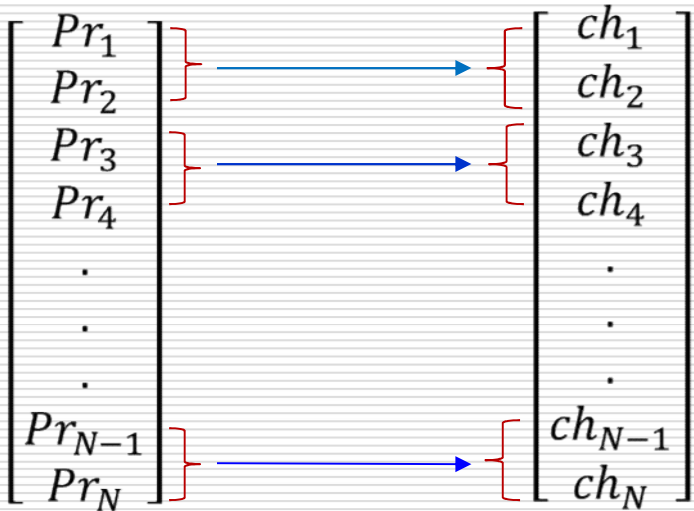


# عملگر همبري

- اين عملگر مهمترين عملگر در الگوريتم وراثتي است
- براي ساختن نسل بعد، دو کروموزوم از حوضچه ازدواج به عنوان والد انتخاب شده و با عمل همبري دو فرزند بوجود ميآيد.
- كه عملگر همبري بطور قطعي روي تمام والدين اجرا نمي شود (احتمال همبري  $P_c$  معمولاً عددي بين 0.2 و 0.7)

# عملگر همبري

- عملگر همبري در دو گام انجام مي شود:
- گام اول: از بين افراد موجود در حوضچه تزويج، دو نفر بصورت اتفاقي براي همبري انتخاب مي شوند.
- گام دوم: دو کروموزوم منتخب به عنوان والدين، طي يکي از روشهاي همبري فرزند يا انتقال زوج فرزندان توليد مي کنند.





# عملگر همبري

---

□ عملگر همبري در دو گام انجام مي شود:      براي هر زوج

□ گام اول: از بين افراد موجود در حوضچه      If rand < Pc

تزوج، دو نفر بصورت اتفاقي براي همبري  
انتخاب مي شوند.

□ گام دوم: دو کروموزوم منتخب به عنوان والدين،  
طي يکي از روشهاي همبري فرزند يا زوج  
فرزنداني توليد مي کنند.

Else

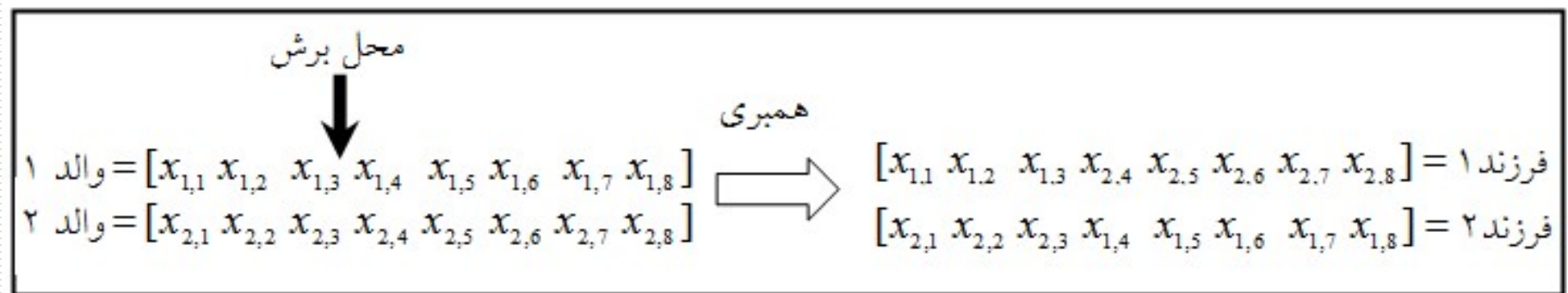
همبري

انتقال

end

---

## عملگر همبری تک نقطه ای



عملگر همبری یک نقطه‌ای در الگوریتم وراثتی حقیقی ( در این مثال فرض شده

است که هر کروموزوم ۸ ژن دارد)

## همبري خطي

$$Off_1 = \lambda_1 P_1 + \lambda_2 P_2$$

$$Off_2 = \lambda_1 P_2 + \lambda_2 P_1$$

$\lambda_1$  and  $\lambda_2$  are Rael

|  |               |  |
|--|---------------|--|
| $P_1 = [0.5 \quad 2 \quad -3 \quad 4.1]$   | ترکیب         | $Off_1 = [1.69 \quad -0.1 \quad -0.41 \quad 2.14]$ |
| $P_2 = [2.2 \quad -1 \quad 0.7 \quad 1.3]$ | $\Rightarrow$ | $Off_2 = [1.01 \quad 1.1 \quad -1.89 \quad 3.26]$  |
| $\lambda_1 = 0.3, \lambda_2 = 0.7$         |               |  |

عملگر ترکیب خطی در الگوریتم وراثتی حقیقی (در این مثال فرض شده است که هر کروموزوم ۴ ژن دارد)

# عملگر جهش Mutation

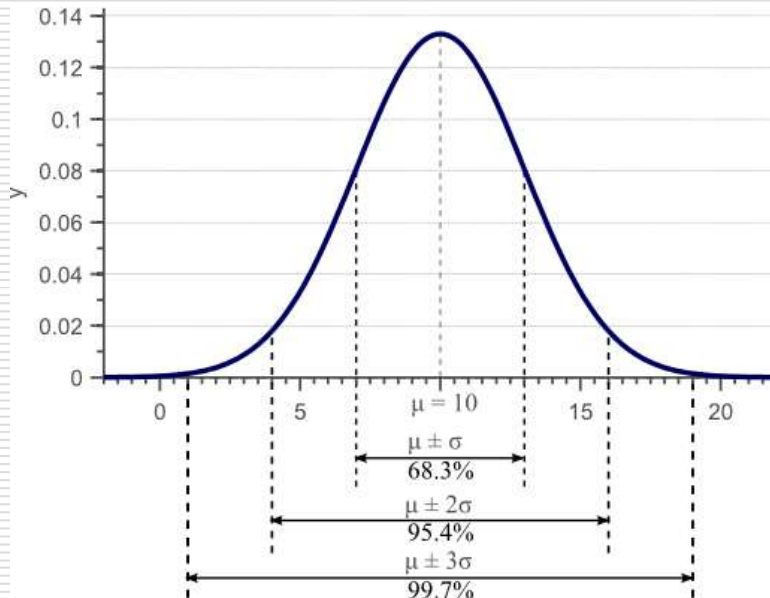
- این عملگر بطور عام بعد از عملگر همبري اعمال مي شود.
- احتمال رخداد جهش در الگوريتم وراثتي حقيقي با  $P_m$  نمايش داده مي شود که معمولاً بين 0.1 تا 0.4 انتخاب مي شود.
- عملگر جهش به هر يك از افراد جمعيت به تنهائي اعمال مي شود.

# عملگر جهش حقیقی (جهش با توزیع نرمال)

□ برای هر متغیر که به طور تصادفی جهش پیدا کند، مقدار متغیر به صورت زیر تغییر می کند:

$$x'_i = x_i + N(0, \sigma_i)$$

□  $N(0, \sigma)$  یک عدد تصادفی است که با تابع توزیع احتمال نرمال (گوسی) با میانگین 0 و واریانس  $\sigma$  تولید می شود.



## سایر مراحل

---

□ جایگزینی نسل بعد

□ چک کردن شرط توقف

---

# ارزیابی

| t=1  | t=2  |       | t=T  |
|--|--|-------|--|
| $\begin{bmatrix} fit_1(1) \\ fit_2(1) \\ fit_3(1) \\ fit_4(1) \\ \vdots \\ fit_{N-1}(1) \\ fit_N(1) \end{bmatrix}$ | $\begin{bmatrix} fit_1(2) \\ fit_2(2) \\ fit_3(2) \\ fit_4(2) \\ \vdots \\ fit_{N-1}(2) \\ fit_N(2) \end{bmatrix}$ | ..... | $\begin{bmatrix} fit_1(T) \\ fit_2(T) \\ fit_3(T) \\ fit_4(T) \\ \vdots \\ fit_{N-1}(T) \\ fit_N(T) \end{bmatrix}$ |

mean

Fitness=[mf(1)

mf(2)

.....

mf(T)]

میانگین

هر تکرار

best

So far =[bf(1)

bf(2)

.....

bf(T)]

بهترین تا

هر تکرار

## ارزیابی در چند اجرای مستقل الگوریتم

|       |        |       |       |        |
|-------|--------|-------|-------|--------|
| Run1  | [bf(1) | bf(2) | ..... | bf(T)] |
| Run2  | [bf(1) | bf(2) | ..... | bf(T)] |
| Run3  | [bf(1) | bf(2) | ..... | bf(T)] |
| .     |        |       |       |        |
| .     |        |       |       |        |
| .     |        |       |       |        |
| .     |        |       |       |        |
| Run20 | [bf(1) | bf(2) | ..... | bf(T)] |

---

|              |         |        |       |         |
|--------------|---------|--------|-------|---------|
| Average best |         |        |       |         |
| So far =     | [abf(1) | abf(2) | ..... | abf(T)] |

---