بسم الله الرحمن الرحیم

# آشنایی با الگوریتم ژنتیک

# چرا الگوریتم های فراابتکاری؟

عوامل تاثیرگذار در پیچیدگی محاسباتی مدل های تصمیم گیری:

- غیرخطی بودن توابع هدف یا محدودیت ها
- گسسته بودن فضای حل مسئله
- اندازه مسئله

روش های دقیق در شرایط فوق قادر به پیدا کردن جواب بهینه در زمان قابل قبولی نیستند.

هدف روش های فراابتکاری: پیدا کردن بهترین جواب ممکن (نزدیک به بهینه) در زمان قابل قبول

# الگوریتم های فرااِبتکاری

در سی سال گذشته، نوع جدیدی از الگوریتم های تقریب ظهور یافته‌اند که اساساً هدف از آنها ترکیب روش‌های ابتکاری در چارچوبهای کلان تر به منظور کاوش کارا و اثربخش فضای جستجو می‌باشد. امروزه از این روش‌ها با عنوان روشهای فرا ابتکاری(متاهیوریستیک) نام برده می شود.

این واژه را اولین بار «گلوور» در ۱۹۸۶ به کار برد که از ترکیب دو واژه یونانی «متا» و «هیوریستیک» ساخته شده است. پیشوند «متا» به معنای فراتر یا در سطحی بالاتر است و «هیوریستیک» به معنای یافتن است.

یک روش فرا ابتکاری، در حقیقت، یک روش ابتکاری برای حل یک طبقه بسیار عمومی از مسائل است و ترکیبی کارآمد از توابع هدف یا روش‌های ابتکاری مبتنی بر توابع هدف می باشد.

# آشنایی با الگوریتم ژنتیک

- ✓ الگوریتم ژنتیک، الهامی از علم ژنتیک و نظریهٔ تکامل داروین است و بر اساس بقای برترین‌ها یا انتخاب طبیعی استوار است. یک کاربرد متداول الگوریتم ژنتیک، استفاده از آن بعنوان تابع بهینه‌کننده است. الگوریتم ژنتیک ابزار سودمندی دربازشناسی الگو، انتخاب ویژگی،درک تصویرو یادگیری ماشینی است.

- ✓ در الگوریتم‌های ژنتیکی، نحوه تکامل ژنتیکی موجودات زنده شبیه‌سازی می‌شود. این الگوریتم ها با الهام از روند تکاملی طبیعت مسائل را حل می نمایند .

- ✓ یعنی مانند طبیعت یک جمعیت از موجودات را تشکیل می دهند و با اعمالی بر روی این مجموعه به یک مجموعه بهینه و یا موجود بهینه دست می یابند.

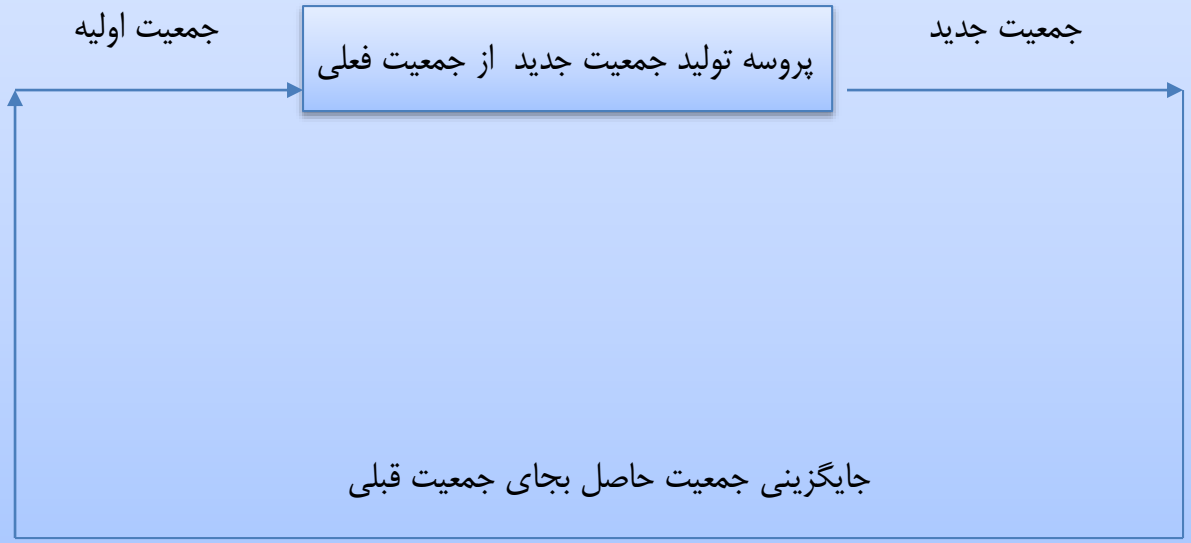
- ✓ با توجه به خصوصیات خاص خودشان به خوبی از عهده حل مسائلی که نیاز به بهینه سازی دارند بر می آیند.

# جدول هم ارزی مفاهیم بیولوژیکی و عناصر GA

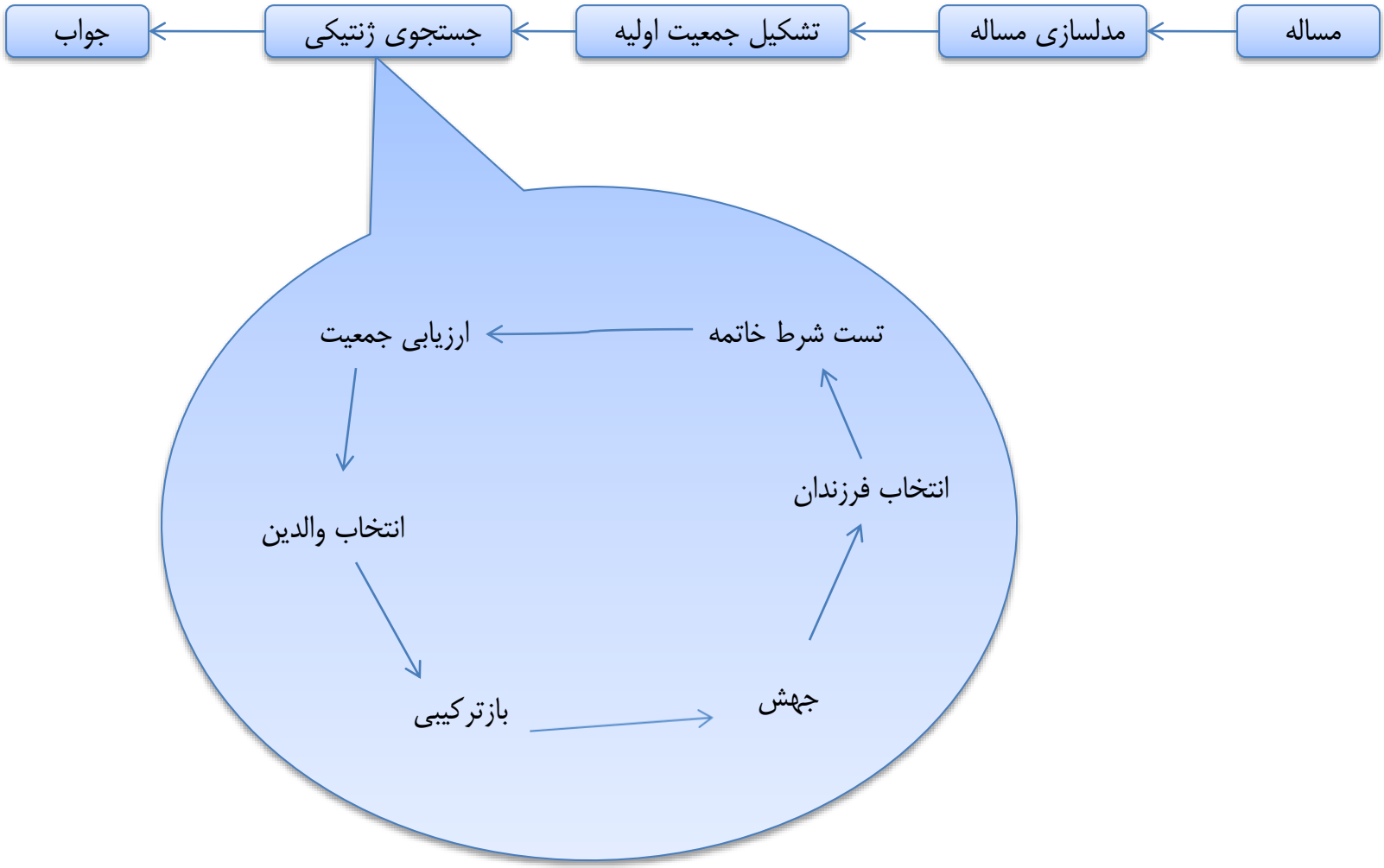
| سیر تکاملی طبیعی | GA |
| --- | --- |
| محیط | مساله برای حل |
| جمعیت | مجموعه جوابهای جایگزین |
| تولید | گام تکرار |
| فرد | جواب داوطلب |
| والدین | جوابهای برگزیده شده |
| اندازهٔ انطباق | مقدار برازندگی ( تابع ارزیاب) |

# نمودار گردشی فرآیند یک الگوریتم تکاملی

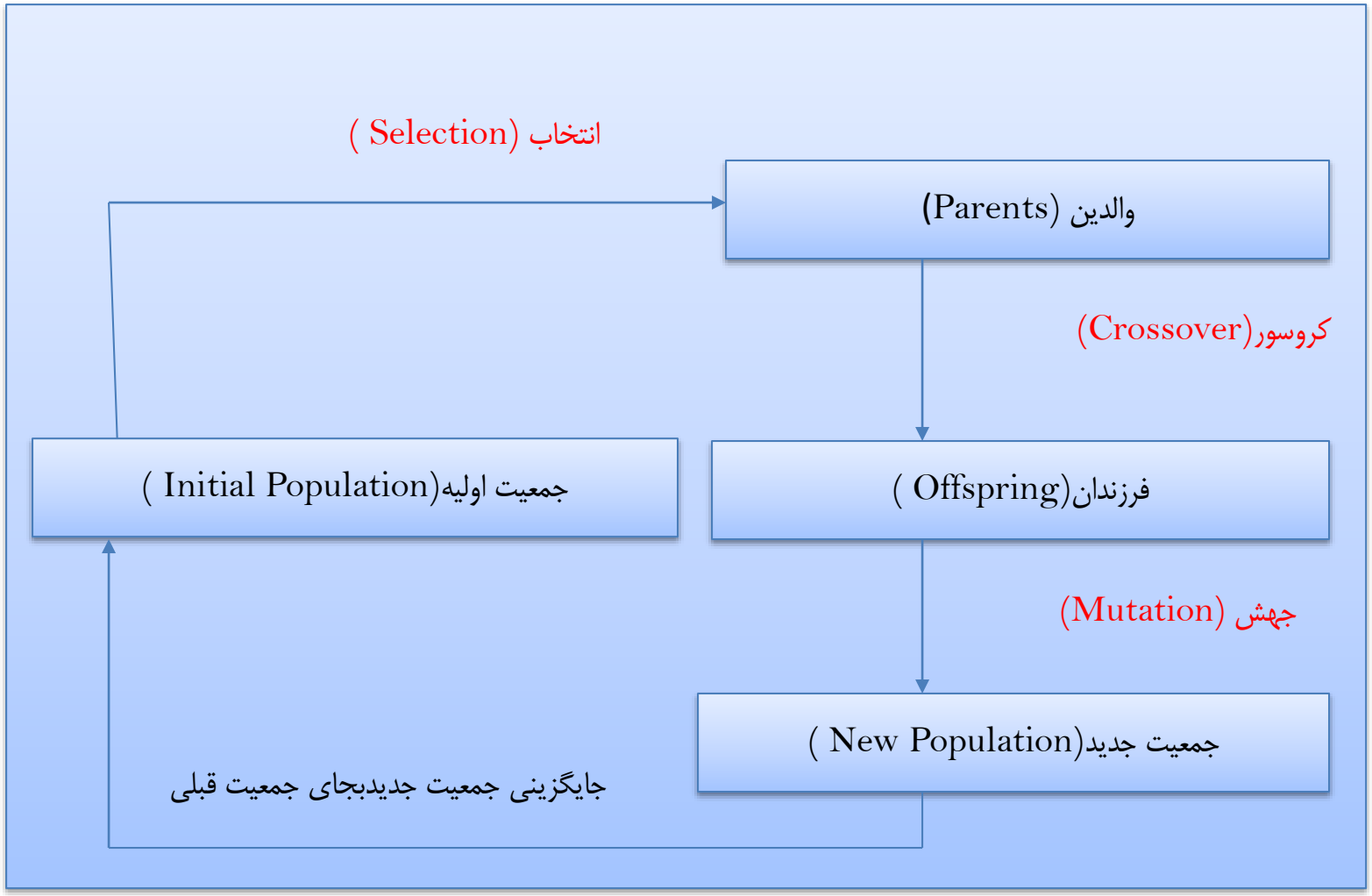
- فرایند تولید تا وقتی که جواب مورد نظر حاصل شود ادامه می یابد
  ( اغلب جمعیت اولیه بصورت تصادفی تولید می شود)



# ساختار الگوریتم های ژنتیک



# ساختار الگوریتم های ژنتیک



# مدلسازی مساله (بازنمایی)

برای اینکه بتوانیم یک مساله را بوسیله الگوریتم های ژنتیک حل کنیم، بایستی آنرا به فرم مخصوص مورد نیاز این الگوریتم ها تبدیل کنیم.

✓در این روند ما بایستی راه حل مورد نیاز مساله را به گونه ای تعریف کنیم که قابل نمایش بوسیله یک کروموزوم باشد.

✓اینکه چه نوع بازنمائی را برای مساله استفاده شود، به شخص طراح و فرم مساله بستگی دارد.

✓ چند نمونه از بازنمائی هایی را که معمولاً استفاده می شوند :

1. اعداد صحیح
2. رشته های بیتی
3. اعداد حقیقی در فرم نقطه شناور
4. اعداد حقیقی به فرم رشته های بیتی
5. یک مجموعه از اعداد حقیقی یا صحیح
6. ماشینهای حالت محدود
7. هر فرم دیگری که بتوانیم عملگرهای ژنتیک را بر روی آنها تعریف کنیم

# تابع برازش

## ارزیابی جمعیت Fitness

برای اینکه بتوانیم موجودات بهتر را درون جمعیت تشخیص بدهیم بایستی معیاری را تعریف کنیم که بر اساس آن موجودات بهتر را تشخیص دهیم. به این کار، یعنی تعیین میزان خوبی یک موجود، ارزیابی آن موجود می گویند.

ارزیابی، اینگونه است که بر حسب اینکه موجود چقدر خوب است یک عدد به آن نسبت می دهیم، این عدد که برای موجودات بهتر بزرگتر (یا کوچکتر) است را شایستگی آن موجود می نامیم.

به عنوان مثال در صورتی که به دنبال مینیمم یک تابع هستیم، مقدار شایستگی را می توانیم ورودیهایی که مقادیر تابع برای آنها کمتر است در نظر بگیریم که ورودیهای بهتری هستند.

✓بسته به نوع مساله ما می خواهیم شایستگی را بیشینه و یا کمینه کنیم.

# عملگر های الگوریتم ژنتیک

- ۱- انتخاب Selection (انتخاب والدین)

✓ سوق دادن جستجو به بخشهایی از فضا که امکان یافتن جوابهای با کیفیت بالاتر وجود دارد.

✓ نسل جدیدی از راه حل ها را با انتخاب والدینی که بالاترین Fitness را دارند تولید می کند.

والدین : در هر نسل تعدادی از عناصر جمعیت این فرصت را پیدا می کنند که تولید مثل کنند. به این عناصر که از میان جمعیت انتخاب می شوند، والدین می گویند.

روشهای مختلفی برای انتخاب والدین وجود دارند. در زیر به چند مورد از این روشها اشاره می کنیم:

# روش های انتخاب والدین

- انتخاب تمام جمعیت بعنوان والدین:

در واقع هیچگونه انتخابی انجام نمی دهیم (همه عناصر انتخاب می شوند) .

- انتخاب تصادفی:

بصورت تصادفی تعدادی از موجودات جمعیت را بعنوان والدین انتخاب می کنیم، این انتخاب می تواند با جایگذاری یا بدون جایگذاری باشد.

در این روشها عناصر با شایستگی بیشتر شانس بیشتری برای انتخاب شدن بعنوان والدین را دارند.

- سایر روشها:

این روشها با استفاده از تکنیک هایی سعی می کنند که انتخاب هایی را ارائه دهند، که هم رسیدن به جواب نهایی را تسریع کنند و هم اینکه کمک می کنند که جواب بهینه تری پیدا شود.

# روش های انتخاب

معمول ترین  روش های انتخاب

: Elitist Selection

مناسب‌ترین عضو هر اجتماع انتخاب می‌شود.

:Roulette Selection

یک روش انتخاب است که در آن عنصری که عدد برازش (تناسب) بیشتری داشته باشد، انتخاب می‌شود.

ScalingSelection

به موازات افزایش متوسط عدد برازش جامعه، سنگینی انتخاب هم بیشتر می‌شود و جزئی‌تر. این روش وقتی کاربرد دارد که مجموعه دارای عناصری باشد که عدد برازش بزرگی دارند و فقط تفاوت‌های کوچکی آن‌ها را از هم تفکیک می‌کند.

Tournament Selection

یک زیر مجموعه از صفات یک جامعه انتخاب می‌شوند و اعضای آن مجموعه با هم رقابت می‌کنند و سرانجام فقط یک صفت از هر زیرگروه برای تولید انتخاب می‌شوند.

# عملگر های الگوریتم ژنتیک

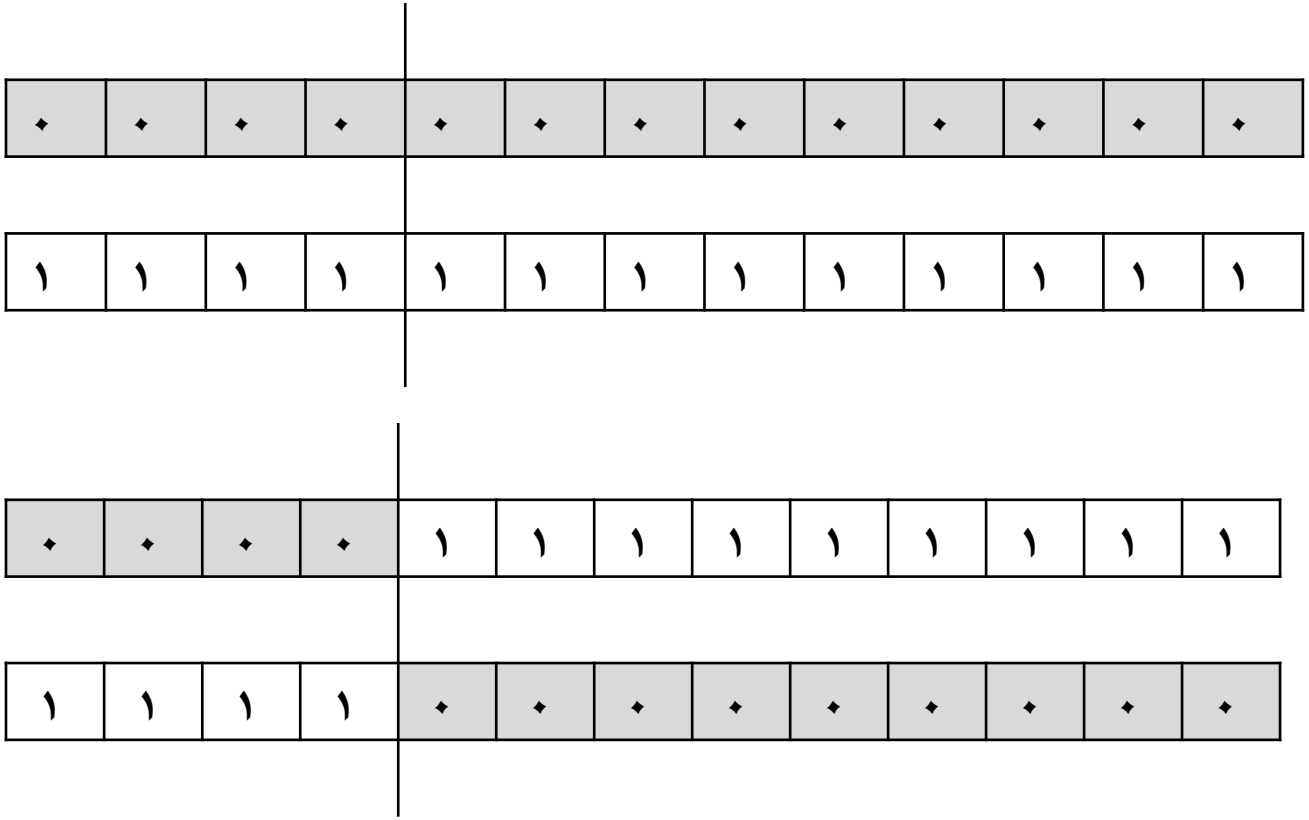
## ۲- تقاطع یا ترکیب مجدد (Recombination/Crossover)

- امکان ترکیب جوابهای جزیی (partial solutions) یافت شده و در نتیجه بدست آوردن جوابهایی با کیفیت بالاتر را فراهم می‌آورد.

در جریان عمل بازترکیبی به صورت اتفاقی بخشهایی از کروموزوم ها با یکدیگر تعویض می شوند. این موضوع باعث می شود که فرزندان ترکیبی از خصوصیات والدین خود را به همراه داشته باشند و دقیقاً مشابه یکی از والدین نباشند.

هدف تولید فرزند جدید می باشد به این امید که خصوصیات خوب دو موجود در فرزندشان جمع شده و یک موجود بهتری را تولید کند.

# روش انجام عمل بازترکیبی




نحوه انجام عملیات بازترکیبی

# روش انجام عمل بازترکیبی

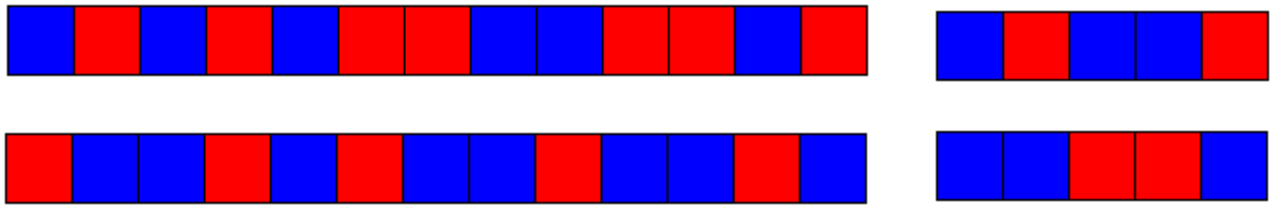
روش کار به صورت زیر است:

بصورت تصادفی یک نقطه از کروموزوم را انتخاب می کنیم
ژن های مابعد آن نقطه از کروموزوم ها را جابجا می کنیم

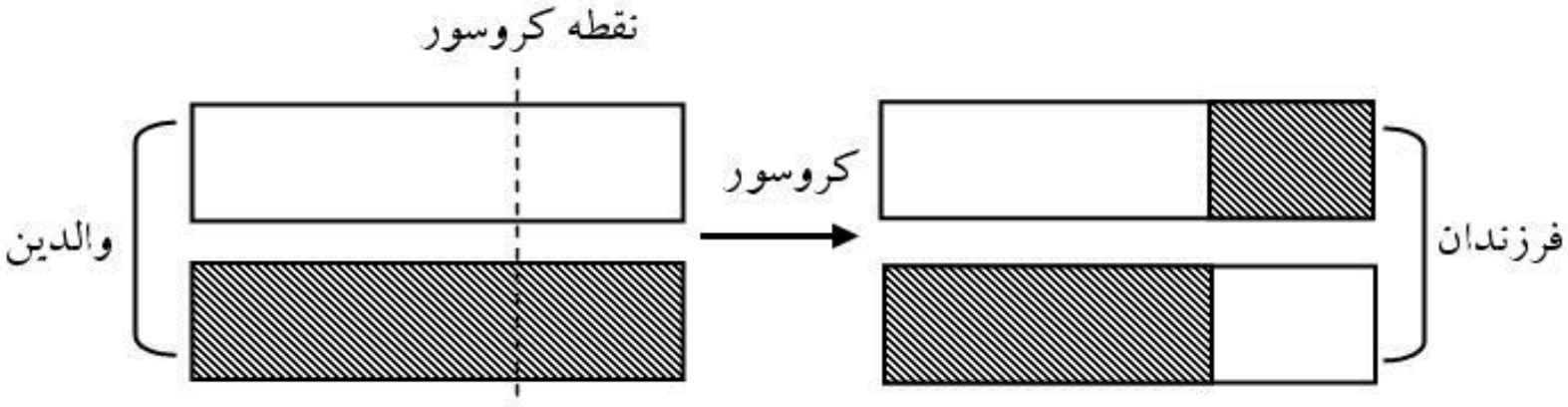
## بازترکیبی تک نقطه ای (Single Point Crossover)

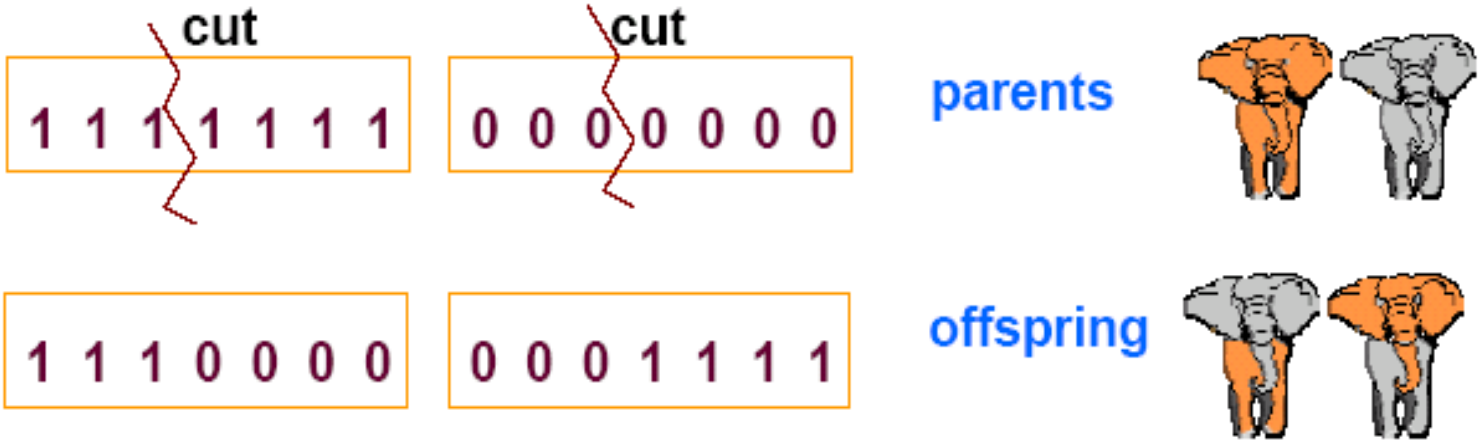
اگر عملیات بازترکیبی را در یک نقطه انجام دهیم به آن بازترکیبی تک نقطه ای می گویند.

در این روش یک مکان تصادفی در طول رشته انتخاب می شود و gene ها از این مکان به بعد جابجا می شوند.



# روش انجام عمل بازترکیبی

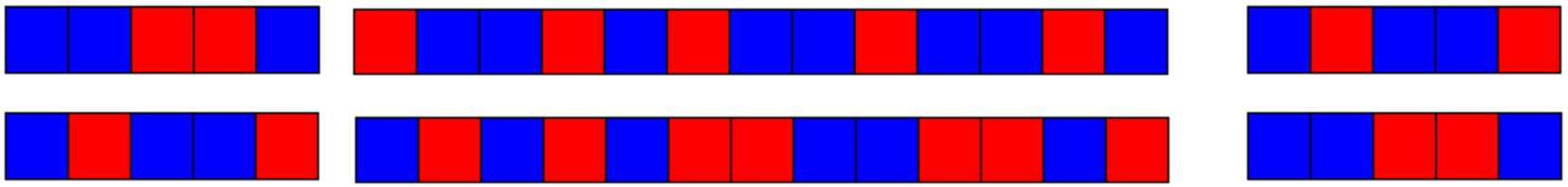


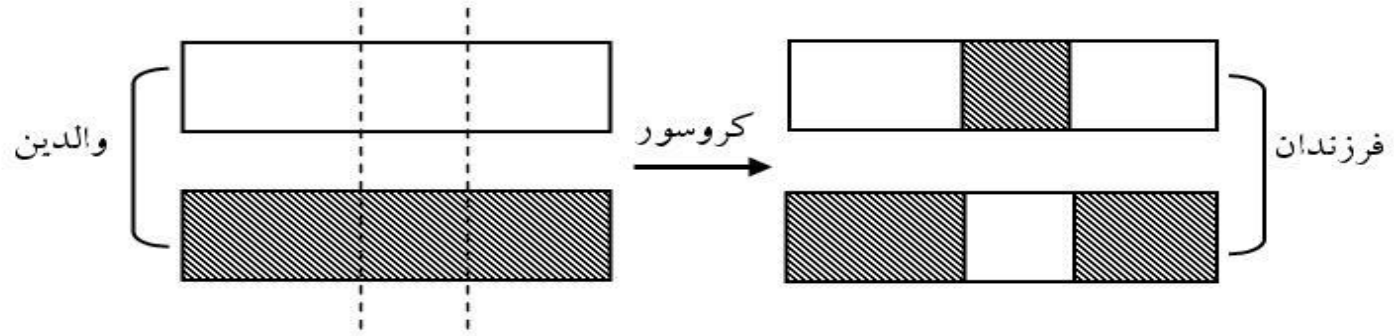



# روش انجام عمل بازترکیبی

## روش ادغام دو نقطه ای(Two-point CrossOver)

✓در این روش دو مکان را به صورت تصادفی انتخاب کرده و مقادیر بین این دو نقطه را جابجا می کنیم.





# روش انجام عمل بازترکیبی

## ادغام چند نقطه ای (Multipoint Crossover) :

می توانیم این عملیات را در چند نقطه انجام دهیم ، که به آن بازترکیبی چند نقطه ای می گویند .

## ادغام جامع (Uniform Crossover)

اگر تمام نقاط کروموزوم را بعنوان نقاط بازترکیبی انتخاب کنیم به آن بازترکیبی جامع می گوئیم.

روش کار را برای این دو مورد اخیر بدین صورت است:

با احتمال ثابتی مثل $P_c$ عمل بازترکیبی را انجام می دهیم

روش کار به صورت زیر است:

- به ازای هر یک از قسمت های کروموزوم:
  - یک عدد تصادفی بین صفر و یک تولید می کنیم
  - اگر این عدد از مقدار ثابتی مثل $P_c$ کوچکتر باشد، ژنهای مابعد آن نقطه از کروموزوم ها را جابجا می کنیم.

# عملگر های الگوریتم ژنتیک

## ۳- جهش (mutation)

- ویژگی تصادفی بودن و امکان فرار از نقاط بهینه محلی را فراهم می‌آورد.

برای انجام جهش به این صورت عمل می کنیم:

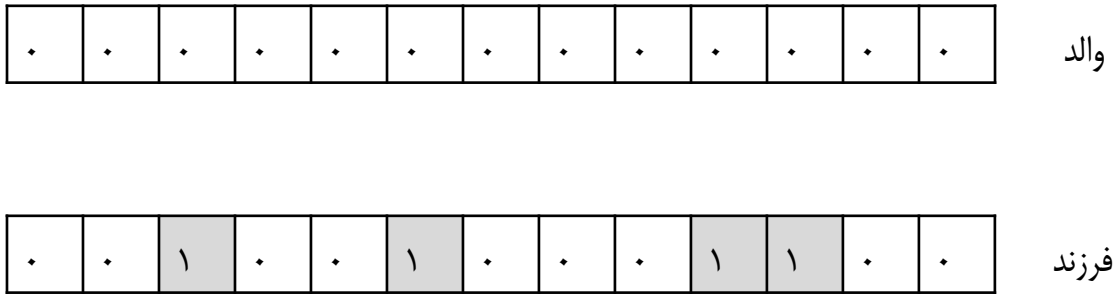
بصورت تصادفی تعدادی از کروموزوم های فرزند را انتخاب می کنیم به صورت تصادفی مقادیر یک یا چند ژن وی را تغییر می دهیم.
همچنین در هنگام پیاده سازی به صورت زیر عمل می کنیم:

به ازای هر کروموزوم اعمال زیر را انجام می دهیم:
یک عدد تصادفی بین صفر و یک تولید می کنیم

اگر عدد تولید شده کوچکتر از $P_m$ بود، به ازای هر ژن اعمال زیر را انجام می دهیم، در غیر اینصورت از جهش دادن کروموزوم صرف نظر می کنیم
یک عدد تصادفی بین صفر و یک تولید می کنیم
اگر عدد تولید شده کوچکتر از $P_g$ بود، ژن مربوطه را جهش می دهیم

# عملگر های الگوریتم ژنتیک

بعنوان مثال جهش برای کروموزوم های به فرم باینری به صورت زیر می باشد:

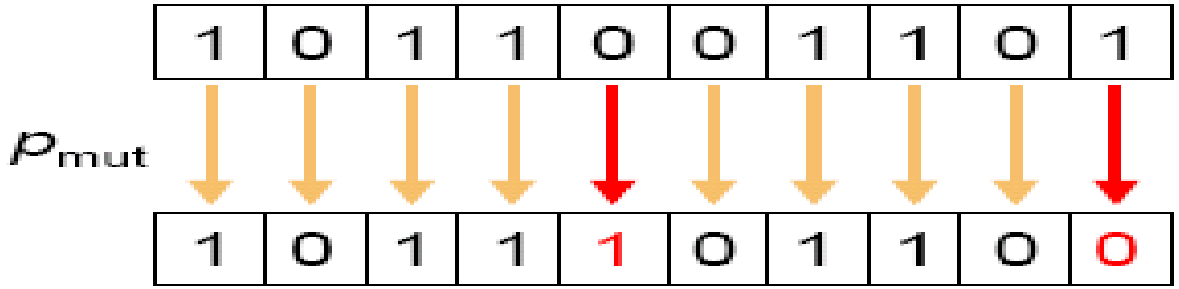


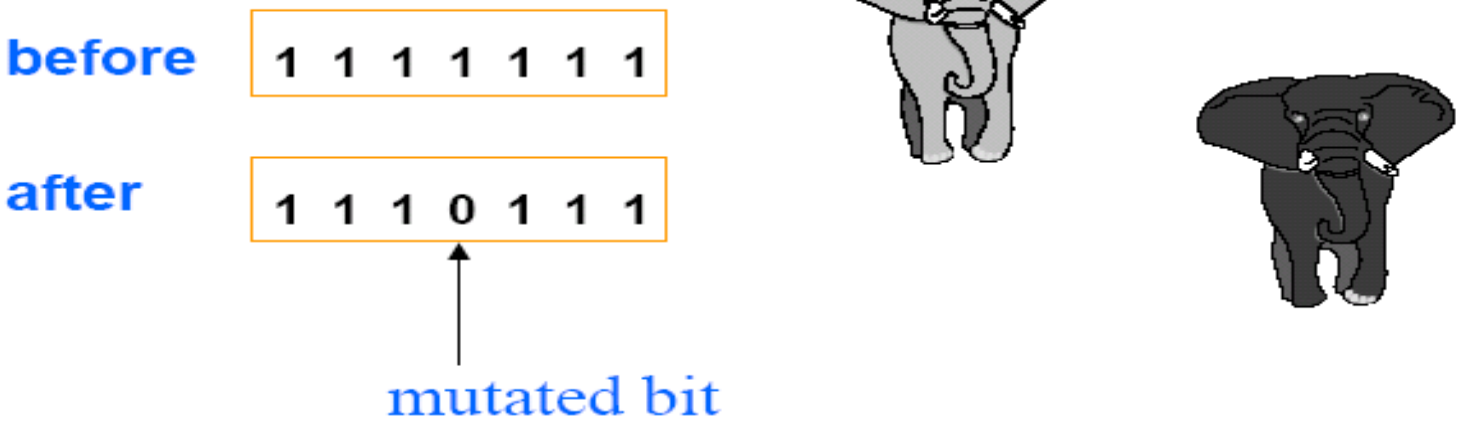

نحوه انجام عملیات جهش

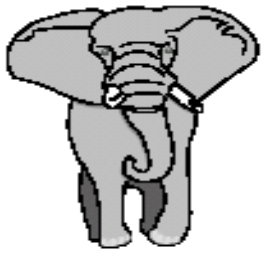
* جهش، برای بازنمایی های که از مقادیر حقیقی استفاده کرده اند، به این صورت پیاده سازی می شود که یک عدد حقیقی بصورت تصادفی در یک محدوده خاص تعیین و جایگزین عدد قبلی می گردد و یا اینکه عدد اصلی با یک مقدار خاص جمع گردد و .... برای سایر مدل های بازنمایی مساله نیز انواع خاصی از جهش پیشنهاد شده است.

# یک مثال

## جهش بیتی ( Bitwise Mutation)



# شرط خاتمه الگوریتم

چون که الگوریتم های ژنتیک  بر پایه تولید و تست می باشند، جواب مساله مشخص نیست و نمی دانیم که کدامیک از جواب های تولید شده جواب بهینه است تا شرط خاتمه را پیدا شدن جواب در جمعیت تعریف کنیم. به همین دلیل، معیارهای دیگری را برای شرط خاتمه در نظر می گیریم:

1. تعداد مشخصی نسل: می توانیم شرط خاتمه را مثلاً ۱۰۰ دور چرخش حلقه اصلی برنامه قرار دهیم.
2. عدم بهبود در بهترین شایستگی جمعیت در طی چند نسل متوالی
3. واریانس شایستگی جمعیت از یک مقدار مشخصی پائین تر بیاید و یا اینکه در طی چند نسل متوالی مشخص، تغییر نکند.
4. بهترین شایستگی جمعیت از یک حد خاصی کمتر شود.

✓شرایط دیگری نیز می توانیم تعریف کنیم و همچنین می توانیم ترکیبی از موارد فوق را به عنوان شرط خاتمه به کار ببندیم.

# کاربرد در مسئله TSP

- **تقاطع مطابقت جزئی یا PMX ( Partially Matched Crossover)**
- اگر دو رشته مقابل را داشته باشیم :

8521|364|7 و 1234|567|8

- و یک تقاطع دو نقطه ای انجام دهیم خواهیم داشت:

V1=1234|364|8 و V2=8521|567|7

- که بطور بدیهی غیرمجازهستند

# کاربرد در مسئله TSP

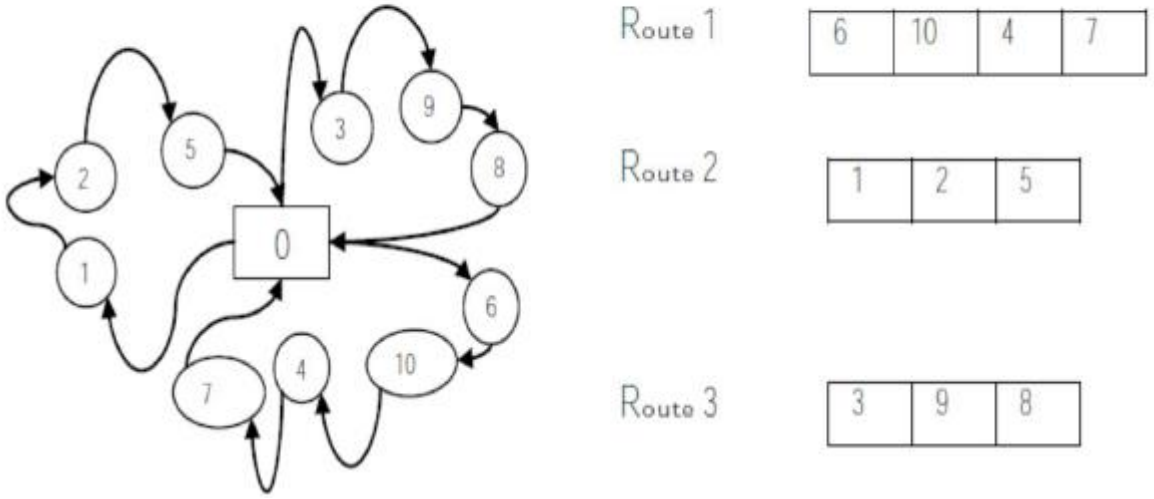
8521|364|7 و 1234|567|8

- با استفاده از PMXرشته های زیر ساخته می شوند:

8 3 2 1 5 6 7 4 و 1 2 5 7 3 6 4 8

# کاربرد در مسئله VRP

- یک کروموزوم یا جواب در VRP : تعداد وسایل نقلیه، مشتریانی که هر مسیر شامل می شود و ترتیب ملاقات برای هر مسیر را مشخص کند.



# کاربرد در مسئله VRP

- فرزندان به عمل آمده از تقاطع می توانند مورد عمل جهش واقع شوند. ۴ عملگر جهش مطرح است:

- وارون سازی یا Inversion
- جابجایی یا Displacement
- درج یا Insertion
- معاوضه یا swap

# کاربرد در مسئله VRP

- **وارون سازی** یک زیرمسیر در نظر می گیرد و ترتیب مشتریان ملاقات شده ای که به آن زیرمسیر تعلق دارند را وارون می کند.

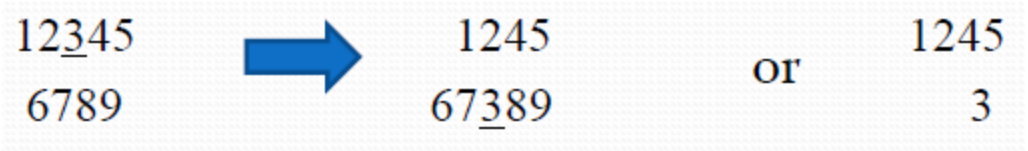
$$12\underline{345} \rightarrow 12\underline{543}$$

- **جابجایی** یک زیرمسیر انتخاب می کند و آنرا به مکان دیگری که به صورت درونی یا بیرونی می تواند جابجا کند، وارد می کند (قسمت انتخاب شده می تواند هم در همان مسیر و هم در مسیر دیگری وارد شود).

$$12\underline{34}5\underline{67} \rightarrow 12\underline{63}45\underline{7}$$

# کاربرد در مسئله VRP

- عملگر **درج** یک مشتری انتخاب می کند و آنرا به مکان دیگری وارد می کند. مسیر درج شونده به صورت تصادفی انتخاب می شود و نیز امکان دارد این مشتری واحد یک مسیر جدید ایجاد کند.




- عملگر **معاوضه** دو مشتری انتخاب می کند و جای آنها را با هم تعویض می کند. این دو مشتری می توانند متعلق به یک مسیر یا مسیرهای متفاوت باشند. همانند عملگر قبلی ممکن است مسیر با توالی جدیدی ایجاد شود.