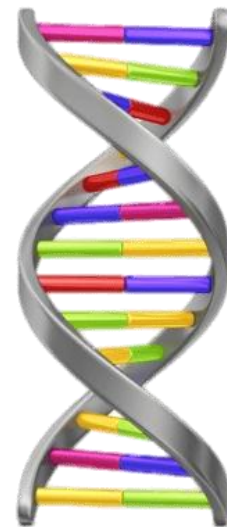
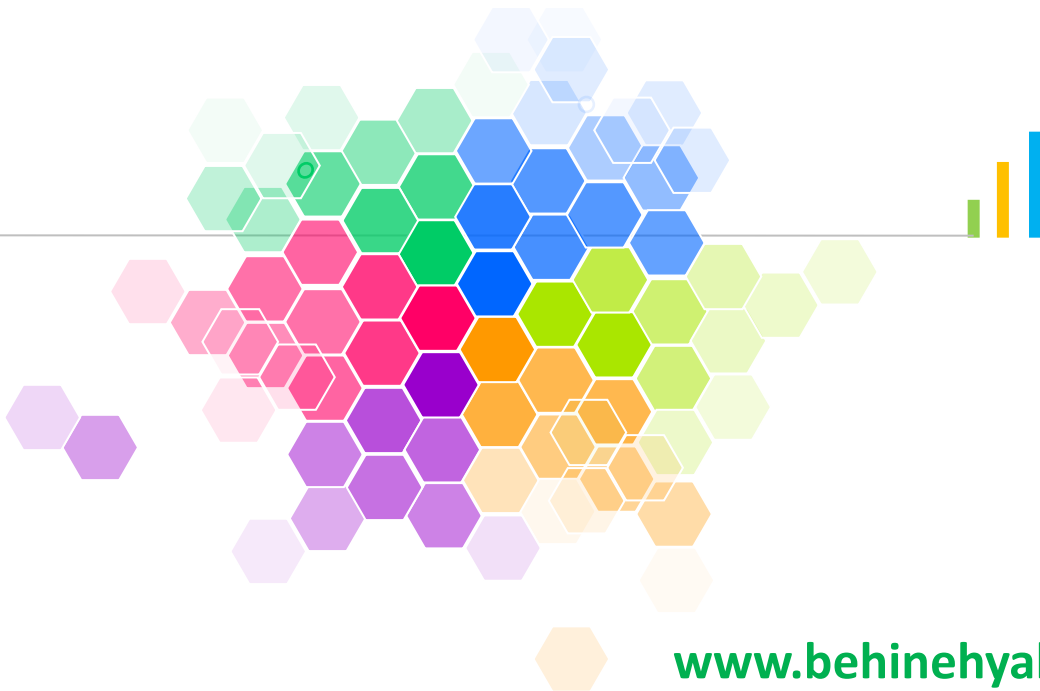


به نام خدا



# درس ۱۷: الگوریتم فراابتکاری ژنتیک



# فهرست مطالب



مقدمه

۱

مبانی الگوریتم فراابتکاری ژنتیک

۲

مراحل الگوریتم فراابتکاری ژنتیک

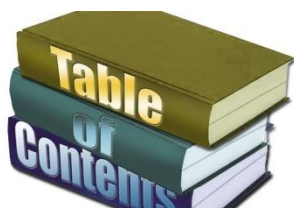
۳

اجزای الگوریتم فراابتکاری ژنتیک

۴

حل یک مثال

۵



## مقدمه

### مقدمه

محاسبه راه حل بهینه *Optimal solution* برای اکثر مسایل بهینه سازی که در خیلی از زمینه‌های کاربردی و عملی مشاهده می‌گردند، کار دشوار و سختی است. در عمل، معمولاً به راه حل‌های خوب که از الگوریتم‌های هیوریستیک *Heuristic* یا متاهیوریستیک (همان فراابتکاری) *Metaheuristic* بدست می‌آید، اکتفا می‌گردد.

## مقدمه

روش جستجوی متاهیوریستیک را می‌توان به صورت متدولوژی‌های عمومی سطح بالایی که می‌توانند به عنوان یک استراتژی راهنما در طراحی هیوریستیک‌های اختصاصی برای حل مسایل بهینه‌سازی تخصصی به کار روند، تعریف کرد.

برخلاف روش‌های دقیق، متاهیوریستیک‌ها (فراابتکاری‌ها) برای مسایل با اندازه‌های بزرگ کاربرد دشوار دارد و راه‌حلهایی راضی‌کننده‌ای در زمان معقولی ارائه می‌نمایند. در این الگوریتم‌ها، هیچ گونه ضمانتی برای یافتن جواب بهینه سراسری یا حدودی از آن وجود ندارد.

## مقدمه

فراابتکاری‌ها در خیلی از زمینه‌ها از قبیل موارد زیر کاربرد دارند:

- ✓ طراحی مهندسی، بهینه سازی توپولوژی، بهینه سازی مسایل الکترونیک، آئرو دینامیک، دینامک سیالات، مخابرات، رباتیک
- ✓ یادگیری ماشین و کاوش داده‌ها
- ✓ مدل سازی سیستم‌ها، شبیه سازی و تحقیق در شیمی، فیزیک، بیولوژی، کنترل، سیگنال، و پردازش تصویر

روش‌های مختلف الگوریتم‌های فراابتکاری یا متاهوریستیک تا به حال پیشنهاد شده است که به صورت زیر است:

- ✓ *Iterated local search* جستجوی محلی تکراری
- ✓ *Ant colony optimization* بهینه سازی کلونی مورچگان
- ✓ *particle swarm intelligent* بهینه سازی توده ذرات
- ✓ *Bee colony* بهینه سازی کلونی زنبوران
- ✓ *Simulated Annealing* انجماد تدریجی
- ✓ *cultural algorithms* الگوریتم‌های ترتیبی
- ✓ *Co-evolutionary algorithms* الگوریتم‌های با هم تکاملی
- ✓ *Genetic algorithm* الگوریتم ژنتیک

## مقدمه

در طراحی یک فراابتکاری، دو معیار متناقض شامل **کاوش** (*Exploration*) و در فضای جستجو (گوناگونی و تنوع) و **تبعیت** (*Exploitation*) از بهترین راه حل‌های پیدا شده، باید در نظر گرفته شوند.

در **کاوش** در ناحیه‌های جستجو نشده بررسی صورت می‌گیرد. در **تبعیت**، در ناحیه‌های امید بخش که تا به حال در آن ناحیه یک جواب خوب پیدا شده است بررسی بیشتر صورت می‌گیرد.

## مقدمه

معیارهای طبقه بندی زیادی ممکن است برای طبقه بندی فراابتکاری‌ها استفاده شود که در زیر به بعضی از آنها اشاره می‌شود:

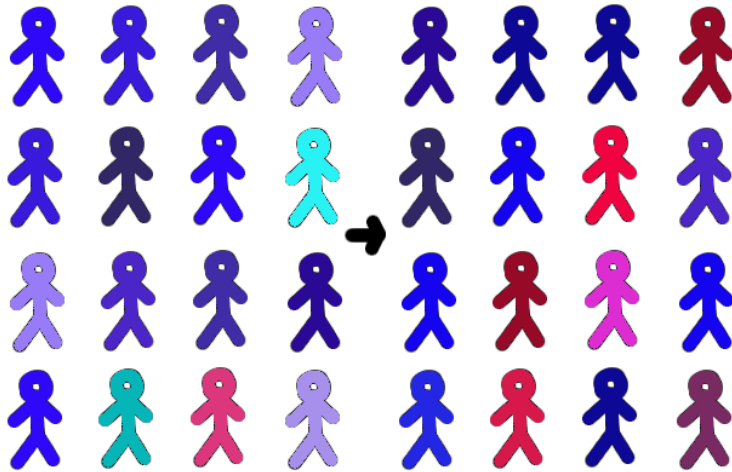
الهام گرفته از طبیعت در مقابل عدم الهام از طبیعت

نحوه استفاده از حافظه

قطعی در مقابل احتمالی

تکراری در مقابل حریصانه

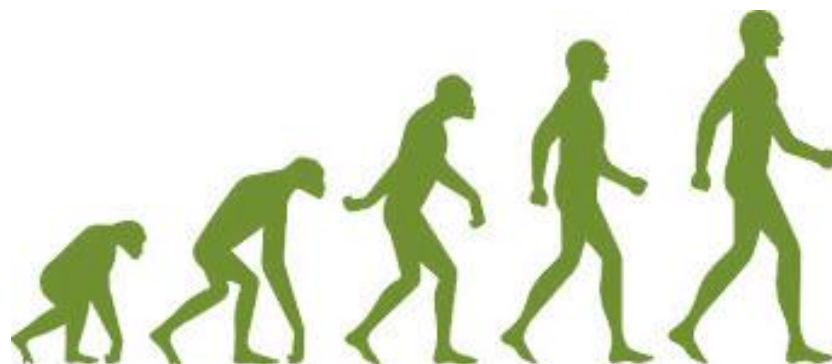
در ادامه الگوریتم **رُنْتیک** که الگوریتمی است که از طبیعت الهام گرفته است و در رده الگوریتم‌های احتمالی، و تکراری با حافظه قرار می‌گیرد صحبت می‌شود.



## مبانی الگوریتم ژنتیک

موجوداتی که توانایی بیشتری در دستیابی به منابع و تولید مثل داشته اند، در طبیعت نیز بیشتر مشاهده می شوند. موجوداتی که توانایی کمتری داشته اند، به دلایل مختلف، به مرور تعداد آنها کم و کمتر شده و حتی به نابودی کشیده شده اند. در طول زمان، گفته می شود که کل جمعیت اکوسیستم دچار **تکاملی** شده است به طور متوسط، شامل موجوداتی شده است که شایسته تر از موجودات قبلی آن می باشند، زیرا این موجودات مشخصاتی داشته اند که منجر به بقای بیشتر آنها شده است.

تکنیک محاسبات تکاملی یا *Evolutionary computation* یا *EC* قواعد تکامل را در الگوریتم‌هایی برای جستجوی راه‌های حل‌های بهینه مسایل به کار می‌گیرد. تفاوت اصلی الگوریتم‌های تکاملی نسبت به الگوریتم‌های دیگر این است که این الگوریتم‌ها مبتنی بر جمعیت عمل می‌نمایند. در این الگوریتم‌ها، معمولاً یک جمعیت اولیه ایجاد و سپس تکامل داده می‌شود.



# مبانی الگوریتم ژنتیک

الگوریتم ژنتیک یا *Genetic Algorithm* مشهورترین تکنیک در تحقیقات الگوریتم‌های **تکاملی** است.

الگوریتم ژنتیک با یک جمعیت **اولیه** از راه حل‌ها شروع می‌شود. هر راه حل از طریق یک کروموزوم نمایش داده می‌شود.

اپراتورهای تولید مثل، مستقیماً روی کروموزوم‌ها عمل نموده و سپس کروموزوم‌ها تحت اپراتور **جهش** و **ترکیب** قرار می‌گیرند.

## مبانی الگوریتم ژنتیک

الگوریتم ژنتیک یک رویه تکراری را به منظور تکامل جمعیت انجام می دهد که در آن فعالیت‌های زیر انجام می شود.

### انتخاب

اولین قدم شامل انتخاب افرادی از جمعیت برای تولید مثل است. این انتخاب به صورت تصادفی، با استفاده از یک احتمال متناسب با تابع برازندگی افراد انجام می گردد. بنابراین، همواره بهترین فرد **شانس بیشتری** برای انتخاب نسبت به فرد ضعیف تر دارد.

## مبانی الگوریتم ژنتیک

### تولید مثل

سپس عملگر تقاطع و جهش روی افراد انتخاب شده به کار گرفته شده و کروموزوم‌های جدید تولید می‌گردند. در این مرحله، فرزندان جدید تولید می‌شوند.

### ارزیابی

در این مرحله، میزان برازندگی فرزندان جدید تولید شده تعیین می‌شود.

### جابه‌جایی

در این مرحله، افرادی از جمعیت قبلی حذف و با افراد جدیدی که به تازگی تولید شده جایگزین می‌شوند.

# مراحل الگوریتم ژنتیک



پس از بیان اصول الگوریتم ژنتیک می توان مراحل این الگوریتم را به صورت زیر بیان کرد:

## شروع

در این مرحله، یک جمعیت اولیه شامل  $n$  کروموزوم تولید می گردد.

## برازندگی

مقدار برازندگی  $f(x)$  هر کروموزوم  $x$  از جمعیت ارزیابی می گردد.

## مراحل الگوریتم ژنتیک

### جمعیت جدید

ایجاد یک جمعیت جدید با تکرار مراحل زیر انجام گردد.

### انتخاب

انتخاب دو کروموزوم از جمعیت مطابق با مقدار برازندگی آنها. در این حالت افرادی

که برازندگی بیشتری دارند با احتمال بیشتری انتخاب می شوند.

## عملگر تقاطعی

با استفاده از عملگر تقاطعی روی والدین انتخاب شد، فرزندان جدید تولید می گردند.  
اگر عملگر تقاطع انجام نگیرد، فرزندان همان والدین خود خواهند بود.

## جهش

با استفاده از عملگر جهش، فرزندان جدید تحت عمل جهش قرار می گیرند. در این  
حالت، جهش روی هر ژن کروموزوم صورت می گیرد.

## پذیرش

فرزند جدید در جمعیت قرار داده می شود.

## جابه جایی

جمعیت جدید (فرزندان) با جمعیت قبلی (والدین) با همدیگر تشکیل یک نسل جدید را می دهند. در این حالت، بعضی از والدین حذف شده و بعضی از فرزندان جانشین آنها می شوند و دو جمعیت تبدیل به یک جمعیت شده تا اندازه جمعیت ثابت بماند.

# مراحل الگوریتم ژنتیک



## شرط توقف

اگر شرط توقف برآورده شده است توقف کنید و بهترین راه حل در جمعیت جاری را گزارش دهید.

## تکرار

در صورت عدم برآورده شدن شرط توقف، به **قدم ساخت جمعیت** است برگردید.

# اجزای الگوریتم ژنتیک

## اجزای الگوریتم ژنتیک

در ساختار الگوریتم ژنتیک بحث شده در قسمت قبلی، اجزایی به کار گرفته شده است که در ادامه آن‌ها مورد بحث قرار می‌گیرد.

### کروموزوم

به رشته یا دنباله‌ای از بیت‌ها که به عنوان شکل کد شده یک جواب ممکن از مساله مورد نظر می‌باشد، کروموزوم می‌گوییم. یک کروموزوم که دارای ۱۰ ژن در شکل زیر نمایش داده شده است.

|   |   |   |   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 0 | 1 | 0 | 1 | 1 | 1 | 0 | 0 | 1 |
|---|---|---|---|---|---|---|---|---|---|

## جمعیت Population

مجموعه‌ای از کروموزوم‌ها را جمعیت گویند. یکی از ویژگی‌های الگوریتم ژنتیک این است که به جای تمرکز بر روی یک نقطه از فضای جستجو یا یک کروموزوم، بر روی جمعیتی از کروموزوم‌ها تمرکز می‌کند. در شکل زیر یک جمعیت با ۴ کروموزوم نمایش داده شده است.

|       |            |   |   |   |   |   |   |   |   |   |   |
|-------|------------|---|---|---|---|---|---|---|---|---|---|
| جمعیت | کروموزوم ۱ | ۱ | ۰ | ۱ | ۰ | ۱ | ۱ | ۱ | ۰ | ۰ | ۱ |
|       | کروموزوم ۲ | ۰ | ۱ | ۱ | ۱ | ۰ | ۰ | ۰ | ۱ | ۱ | ۱ |
|       | کروموزوم ۳ | ۱ | ۰ | ۱ | ۱ | ۱ | ۰ | ۰ | ۱ | ۰ | ۱ |
|       | کروموزوم ۴ | ۱ | ۱ | ۰ | ۱ | ۱ | ۰ | ۱ | ۱ | ۱ | ۱ |

## اجزای الگوریتم ژنتیک

### مقدار برازندگی

مناسب بودن یا نبود جواب با استفاده از مقدار تابع هدف سنجیده می شود. جواب هر چه مناسب تر باشد، به همان اندازه مقدار برازندگی بزرگتری دارد. معمولاً در مواردی که امکان دارد توصیه می شود تابع برازندگی در فاصله صفر و یک نرمال شود.

### کد گذاری

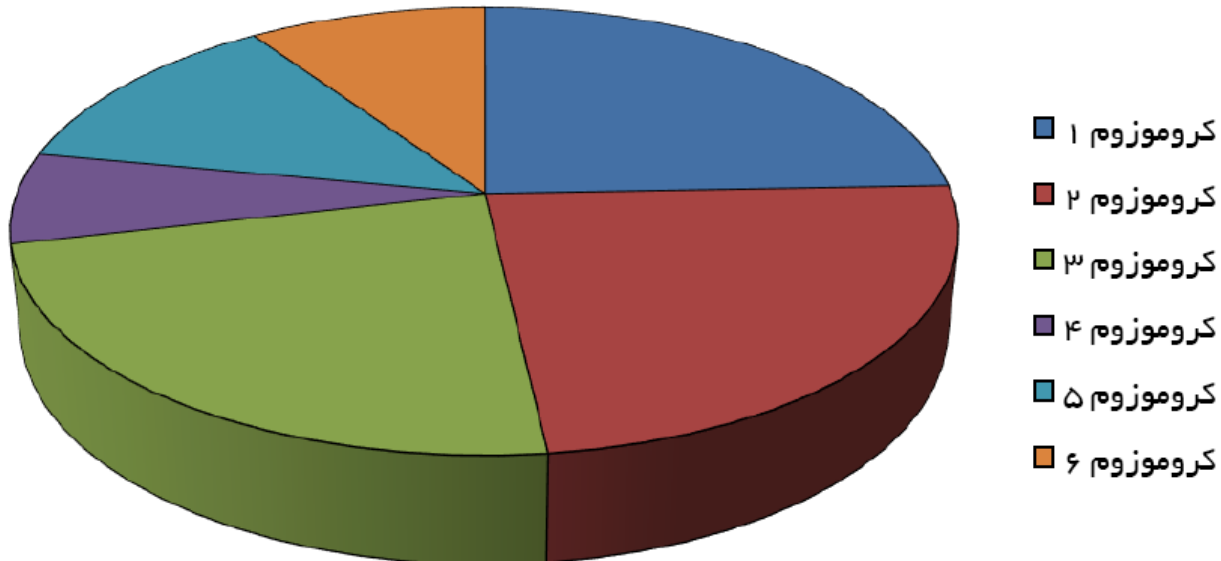
کدگذاری، فرایندی است که به واسطه آن، راه حل‌ها در فضای فیزیکی مساله تبدیل به راه حل‌هایی با کدهایی می گردد که قابل درک برای کامپیوتر است. نمایش اعداد در مبنای ۲ نوعی از کدگذاری‌های رایج است.

## انتخاب

انتخاب، فرایند انتخاب دو والد از جمعیت برای عمل تقاطع یا **Crossover** است. در این مرحله، در خصوص نحوه انتخاب والدین برای عمل تقاطع، نحوه تولید فرزندان و تعداد فرزندان تصمیم گیری گردد. هدف **انتخاب** این است که والدین شایسته تر انتخاب شده که منجر به تولید فرزندانی با برازندگی بالا گردد. چند روش برای رویه انتخاب در ادامه بیان می شود.

## روش چرخ رولت

روش چرخ رولت، یکی از روش‌های سنتی انتخاب در الگوریتم ژنتیک است. در این حالت یک کروموزوم از جمعیت براساس احتمالی متناسب با مقدار شایستگی اش انتخاب می گردد.

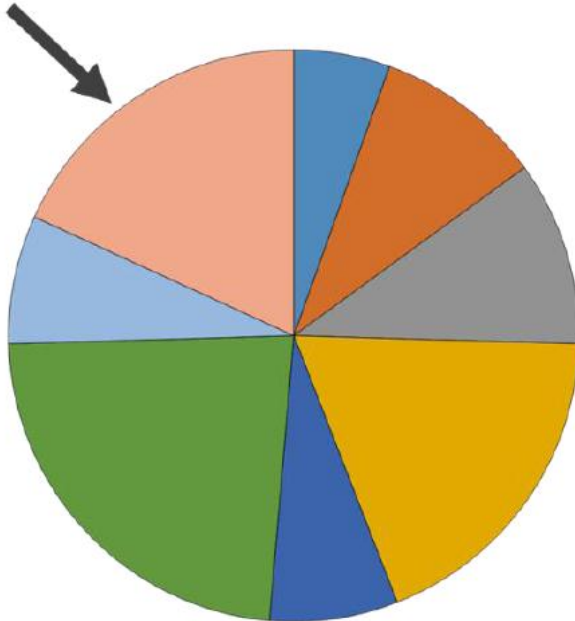


# اجزای الگوریتم ژنتیک

دو نوع چرخ رولت وجود دارد. در نوع اول با هر بار چرخش، یک کروموزوم انتخاب می گردد. در نوع دوم، با هر بار چرخیدن، بیش از دو کروموزوم انتخاب می شود. در شکل زیر دو نوع چرخ رولت را نشان می دهد.

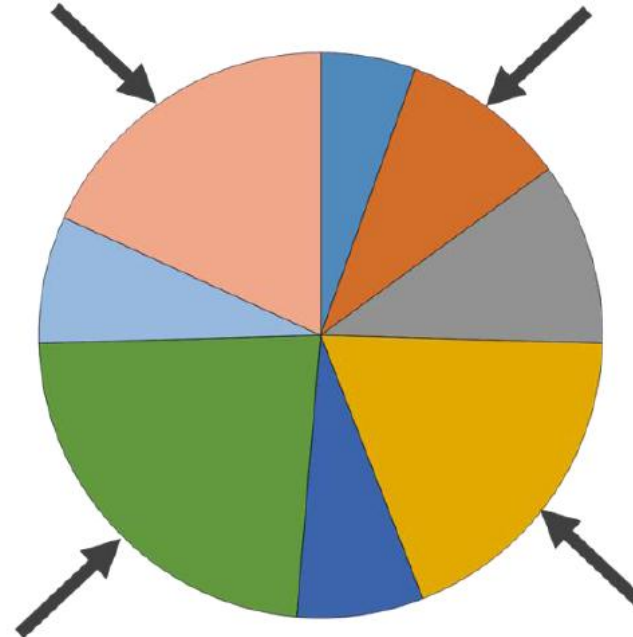
انتخاب تک رولت

نشانه



انتخاب چند رولت

نشانه ها



## اجزای الگوریتم ژنتیک

### عملگر تقاطعی crossower

عملگر تقاطعی، فرایندی است که دو والد را در نظر گرفته و براساس آن‌ها یک فرزند جدید تولید می نماید. عملگر تقاطعی در جمعیت به این امید فعالیت می کند که فرزند **بهتری** تولید گردد. این عملگر یک اپراتور ترکیب بندی است که در سه مرحله صورت می گیرد.

۱- اپراتور تولید مثل یک زوج از والدین را از جمعیت انتخاب می کند.

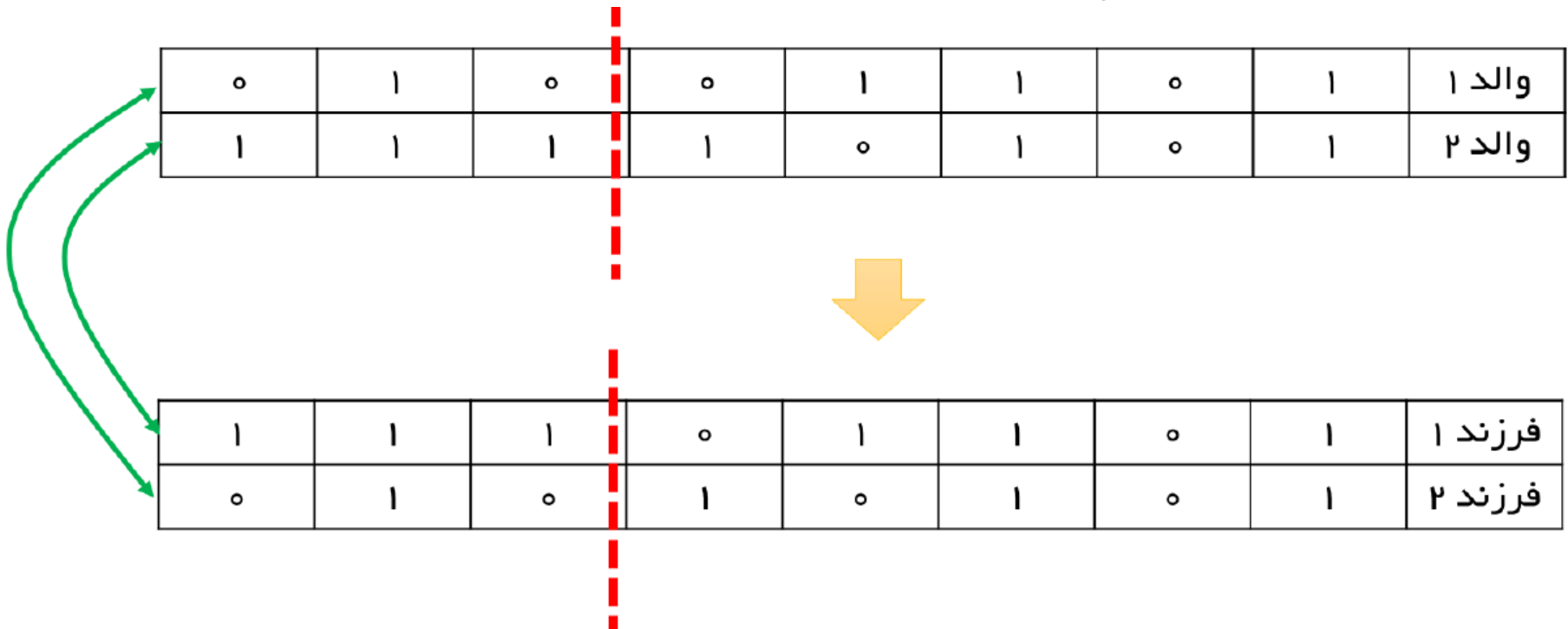
۲- یک نقطه تقاطع به صورت تصادفی در طول رشته انتخاب می شود.

۳- در نهایت، مقادیر رشته‌ها با توجه به نقطه تقاطع تعویض می گردند.

# اجزای الگوریتم ژنتیک

## تقاطع تک نقطه ای

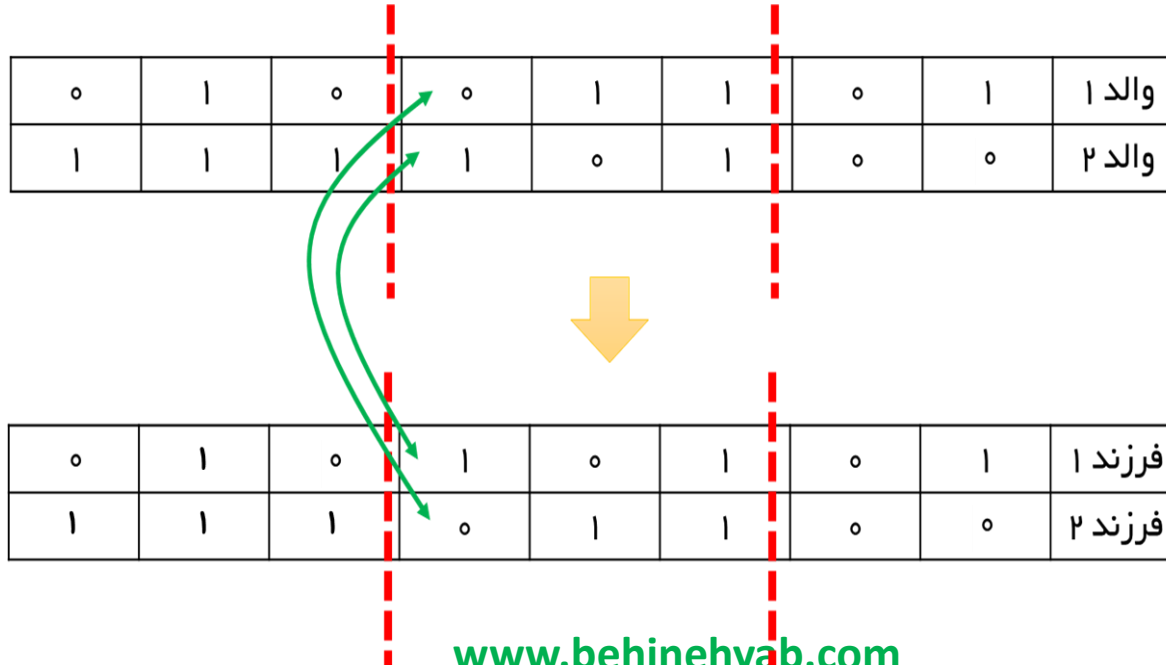
عملگرد تقاطعی تک نقطه ای، دو کروموزوم را به طور تصادفی از یک نقطه شکسته و بخش های شکسته شده دو کروموزوم را جابه جا می کند.



# اجزای الگوریتم ژنتیک

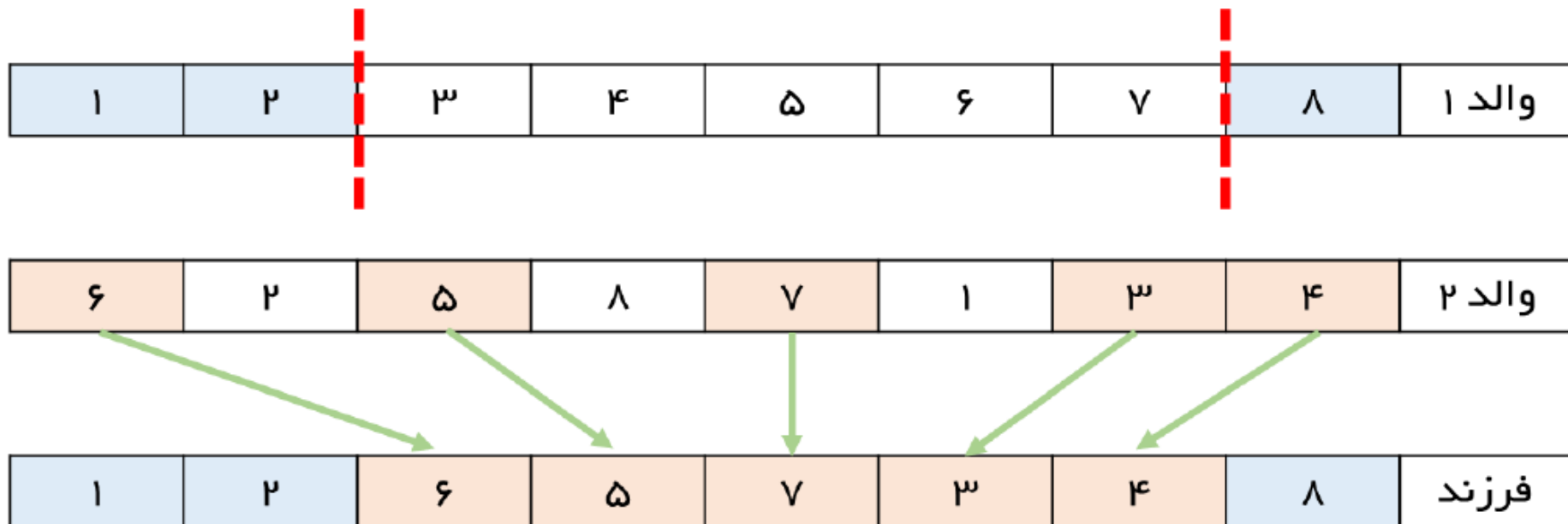
## تقاطع دو نقطه ای

در عملگر تقاطع دو نقطه ای، دو نقطه شکست در طول رشته جواب در نظر گرفته می شود. در این حالت، ابتدا دو مکان تصادفی را در طول رشته انتخاب شده و سپس به صورت یک میان قسمت‌های والد‌ها به فرزندان منتقل می شود.



# اجزای الگوریتم ژنتیک

عملگر تقاطع دو نقطه ای برای یک کروموزوم ترتیبی



## اجزای الگوریتم ژنتیک

### تقاطع یکنواخت

در این حالت، براساس یک توزیع تصادفی باینری مشخص می گردد که یک ژن از کدام والد انتخاب انتخاب گردد. مثلا اگر مقدار توزیع باینری ۱ باشد ژن از والد اول و در صورتی که صفر باشد، از والد دوم انتخاب می گردد. برای فرزند دوم، عکس فرزند اول در نظر گرفته شده است. تابع توزیع به صورت زیر می شود.

$$p_i = \begin{cases} 1 & \text{ژن از والد اول گرفته شود} \\ 0 & \text{ژن از والد دوم گرفته شود} \end{cases}$$

# اجزای الگوریتم ژنتیک

براساس تابع توزیع فوق، عملگر تقاطع یکنواخت برای یک مثال به صورت زیر می شود.

|              |   |   |   |   |   |   |   |   |
|--------------|---|---|---|---|---|---|---|---|
| والد ۱       | ۱ | ۰ | ۰ | ۱ | ۱ | ۰ | ۱ | ۰ |
| والد ۲       | ۱ | ۰ | ۱ | ۰ | ۱ | ۰ | ۱ | ۱ |
| مقدار تصادفی | ۱ | ۱ | ۰ | ۱ | ۰ | ۱ | ۱ | ۰ |
| فرزند ۱      | ۱ | ۰ | ۱ | ۱ | ۱ | ۰ | ۱ | ۱ |
| فرزند ۲      | ۱ | ۱ | ۰ | ۰ | ۱ | ۰ | ۱ | ۰ |

# اجزای الگوریتم ژنتیک

## تقاطع از سه والد یا Three Parent Crossover

در تکنیک تقاطع از سه والد، سه والد به طور تصادفی انتخاب می گردند. هر ژن از والد اول با همان ژن از والد دوم مقایسه می گردد. اگر مشابه باشند، همان ژن به فرزند منتقل می گردد. در صورتی که مشابه نباشد، ژن والد سوم در نظر گرفته می شود.

| شماره ژن | ۱            | ۲ | ۳            | ۴            | ۵            | ۶ | ۷ | ۸ |
|----------|--------------|---|--------------|--------------|--------------|---|---|---|
| والد ۱   | ۱            | ۱ | ۰            | ۱            | ۰            | ۰ | ۰ | ۱ |
| والد ۲   | <del>۱</del> | ۱ | <del>۱</del> | <del>۱</del> | <del>۱</del> | ۱ | ۱ | ۱ |
| والد ۳   | ۰            | ۱ | ۱            | ۰            | ۱            | ۱ | ۰ | ۰ |
| فرزند    | ۰            | ۱ | ۱            | ۰            | ۱            | ۰ | ۰ | ۱ |

## اجزای الگوریتم ژنتیک

### تقاطع PPC یا Precedence Preservation Crossover

تقاطع PPC برای مساله مسیریابی وسایل حمل و نقل و برای مسایل زمان بندی توسعه داده شد. در زیر عملگر PPC برای یک مسئله توالی شرح داده شده است:

۱- در ابتدا یک بردار به صورت کروموزوم در نظر بگیرید. بردار را با استفاده از اعداد ۱ و ۲ به طور تصادفی پر کنید.

۲- یک فرزند یا کروموزوم خالی را در نظر بگیرید. ژن‌های این کروموزوم به صورت زیر پر می شوند:

## اجزای الگوریتم ژنتیک

a. بردار تصادفی را بخوانید. در صورتی که یک باشد، ژن مربوطه از والد اول گرفته می شود و در صورتی که ۲ باشد، ژن مربوطه از والد دوم گرفته می شود.

b. در صورتی که این ژن قبلا به فرزند منتقل شده باشد، ژن قبلی از آن والد در نظر گرفته می شود.

c. برای تمامی ژن ها فرزند، عمل فوق انجام می گردد و کروموزوم فرزند تکمیل می گردد.

# اجزای الگوریتم ژنتیک

مثالی از عملگر  $PPC$  در شکل زیر نشان داده شده است.

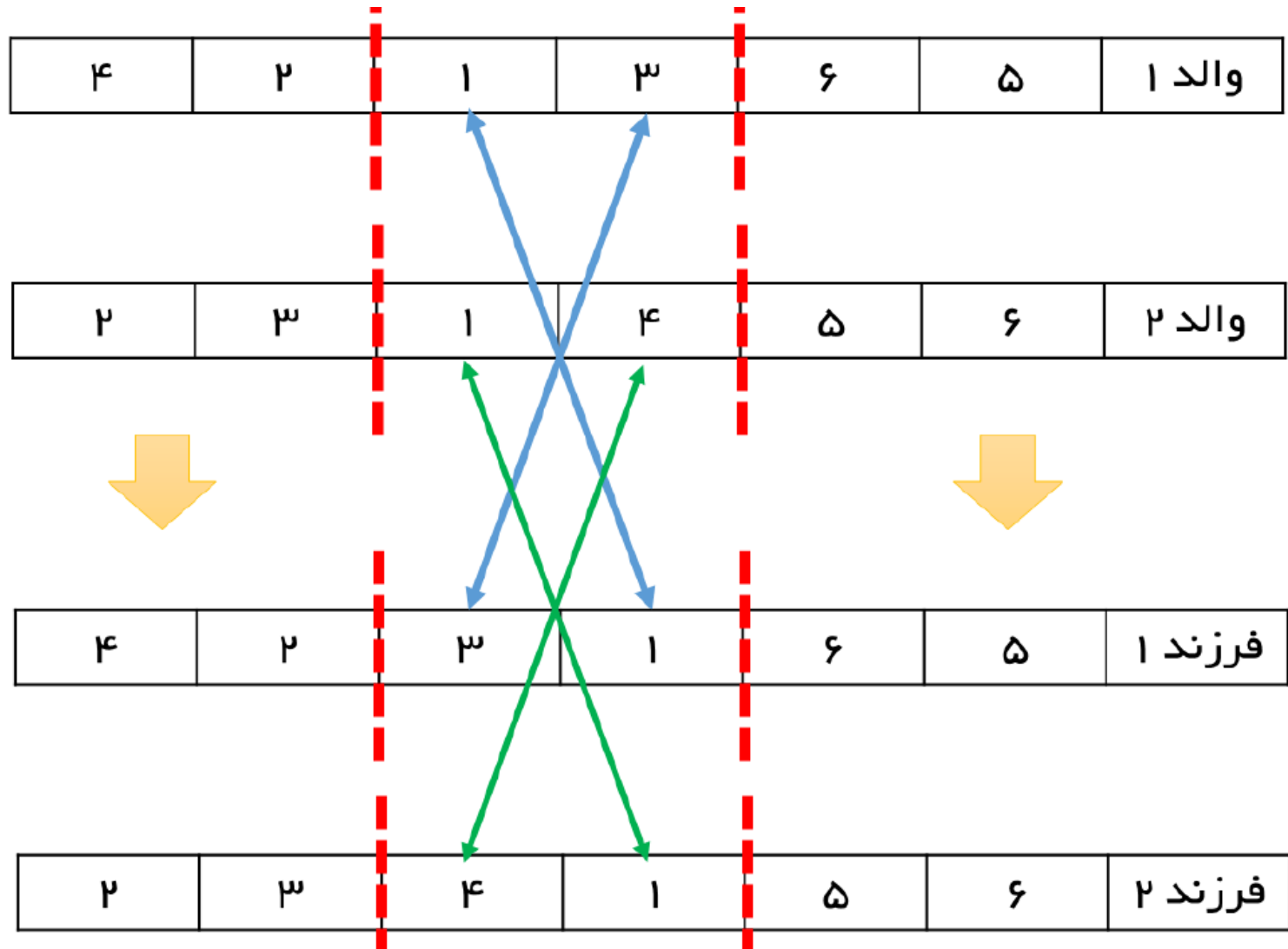
|              |                                                                |                                                                                                                      |                                                                                                                      |                                                                |
|--------------|----------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------|
| والد ۱       | A                                                              | B                                                                                                                    | C                                                                                                                    | D                                                              |
| والد ۲       | C                                                              | A                                                                                                                    | B                                                                                                                    | F                                                              |
| مقدار تصادفی | ۱                                                              | ۲                                                                                                                    | ۱                                                                                                                    | ۱                                                              |
| فرزند ۱      | A                                                              | C                                                                                                                    | B                                                                                                                    | D                                                              |
| توضیحات      | چون مقدار تصادفی ۱ است لذا والد ۱ انتخاب می شود و ژن A می شود. | چون مقدار تصادفی ۲ است، A می تواند انتخاب شود ولی چون ژن قبلی A بوده است C که ژن ما قبلی والد ۲ است C انتخاب می شود. | چون مقدار تصادفی ۱ است، C می تواند انتخاب شود ولی چون ژن قبلی C بوده است B که ژن ما قبلی والد ۱ است B انتخاب می شود. | چون مقدار تصادفی ۱ است لذا والد ۱ انتخاب می شود و ژن D می شود. |

## اجزای الگوریتم ژنتیک

### تقاطع مرتب ordered crossover

از تقاطع دو نقطه ای مرتب زمانی استفاده می شود که مساله مبتنی بر ترتیب باشد. با دو والد داده شده، دو نقطه تقاطع تصادفی انتخاب می گردد. که آن ها را به سه قسمت چپ، وسط و راست تقسیم می کند. عملگر به این ترتیب عمل می نماید. فرزند ۱ قسمت چپ و راست را از والد ۱ و قسمت وسط آن براساس ژن های قسمت وسط والد ۱ به نحوی که ترتیب آن براساس والد ۲ باشد تعیین می گردد. فرایند مشابه برای فرزند دوم صورت می گیرد. یک مثال از تقاطع مرتب در شکل زیر نمایش داده می شود.

# اجزای الگوریتم ژنتیک



# اجزای الگوریتم ژنتیک

## جهش Mutation

بعد از تقاطع، کروموزوم‌های تحت عملگر جهش قرار می گیرند. عملگر جهش از افتادن الگوریتم در بهینه محلی جلوگیری می نماید.

## معکوس کردن

معکوس کردن یک ژن شامل تغییر صفر به یک و یک به صفر براساس یک کروموزم جهش در یک بردار تصادفی تولید شده است. در این شکل، در کروموزم جهش، ۳ عدد ۱ وجود دارد و ژن مرتب از والد معکوس شده اند و کروموزم فرزند تولید شده است.

|   |   |   |   |   |   |   |   |                 |
|---|---|---|---|---|---|---|---|-----------------|
| ۱ | ۰ | ۱ | ۰ | ۱ | ۱ | ۰ | ۱ | والد            |
| ۱ | ۰ | ۰ | ۱ | ۰ | ۰ | ۰ | ۱ | کروموزوم تصادفی |
| ۰ | ۰ | ۱ | ۱ | ۱ | ۱ | ۰ | ۰ | فرزند           |

# اجزای الگوریتم ژنتیک

## تعویض

در این نوع جهش، دو موقعیت تصادفی از یک رشته انتخاب و ارزش‌های مرتبط آن‌ها با همدیگر تعویض می‌گردد. شکل زیر این نوع جهش را نشان می‌دهد.

|                  |   |   |   |   |   |   |   |       |
|------------------|---|---|---|---|---|---|---|-------|
| دو موقعیت تصادفی |   |   |   |   |   |   |   |       |
| ۱                | ۰ | ۱ | ۰ | ۱ | ۱ | ۰ | ۱ | والد  |
| ۱                | ۰ | ۰ | ۰ | ۱ | ۱ | ۱ | ۱ | فرزند |

## اجزای الگوریتم ژنتیک

### جابه جایی

جابه جایی آخرین مرحله از چرخه تولید مثل است. دو والد از یک جمعیت با اندازه ثابت، انتخاب شده و دو فرزند را به وجود آورده اند. نمی توان کل این چهار کروموزوم راه به جمعیت برگرداند و در نتیجه باید دو کروموزوم حذف گردد. روش های مختلفی برای انتخاب جمعیت جدید وجود دارد که به طور مثال می توان از دو روش زیر نام برد.

✓ تمامی اعضای جمعیت جدید از میان کروموزوم های فرزندان انتخاب شوند.

✓ تعدادی از افراد جمعیت مرحله بعد، همان افراد جمعیت مرحله قبل بوده و مابقی از

میان فرزندان جدید انتخاب گردند. البته در هر مورد، شایسته ترین کروموزوم ها

انتخاب می شود.

## اجزای الگوریتم ژنتیک

### قاعده توقف

قواعد توقف متعددی برای الگوریتم ژنتیک وجود دارد که در زیر به طور خلاصه به بعضی از

آن اشاره می‌نمایم.

✓ حداکثر تولید نسل: با استفاده از این قاعده، الگوریتم ژنتیک زمانی متوقف می‌گردد که

تعداد مشخصی از تولید نسل اتفاق افتاده باشد

✓ زمان سپری شده: زمانی که فرایند الگوریتم ژنتیک زمان خاصی را سپری کرد،

الگوریتم متوقف می‌گردد.

✓ عدم بهبود در برازندگی: در این حالت، در صورتی که هیچ تغییری در بهترین

برازندگی جمعیت بعد از تعداد مشخصی تولید نسل به وجود نیاید، الگوریتم ژنتیک

متوقف می‌گردد.

## حل یک مثال

**EXAMPLE**

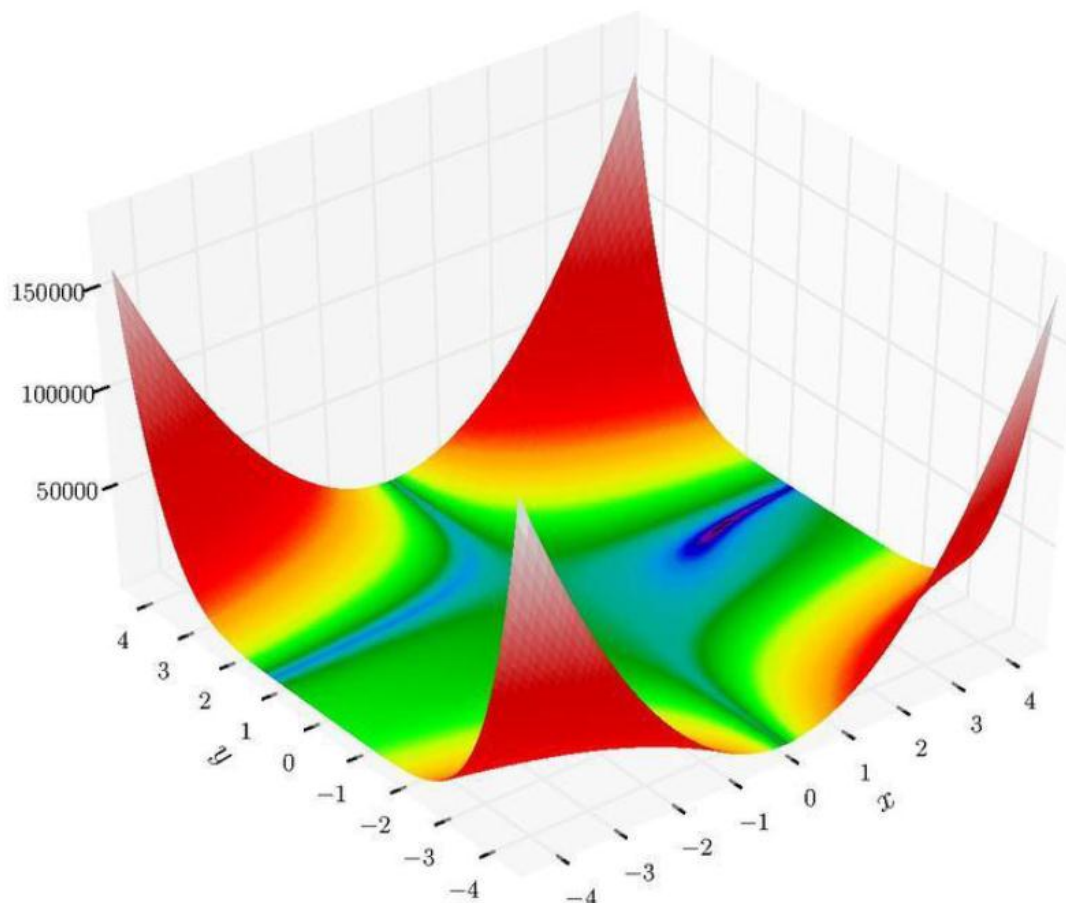
### مثال

در این قسمت می خواهیم یک مسئله بهینه سازی بدون محدودیت را با استفاده از روش الگوریتم ژنتیک حل نماییم. در این مثال به دنبال کمینه کردن تابع هدف زیر هستیم.

$$f(x, y) = (1.5 - x + xy)^2 + (2.25 - x + xy^2)^2 + (2.625 - x + xy^3)^2$$

مقدار بهینه تابع هدف این مدل برابر با صفر و مقدار متغیرهای برابر با  $(x^*, y^*) = (3, 0.5)$  هستند. نمایش سه بعدی این تابع به صورت زیر است.

# حل یک مثال



## حل یک مثال



برای یافتن جواب بهینه این تابع مراحل زیر باید طی شود.

### تنظیمات الگوریتم

در این قدم تنظیمات اولیه برای اجرای الگوریتم آورده می شود. نرخ جهش برابر  $0.1$  و تعداد جمعیت برابر  $20$  در نظر گرفته می شود. بازه ی تغییرات دو متغیر  $x \in [0,10]$  و  $y \in [0,10]$  است.

## حل یک مثال

### تشکیل جمعیت اولیه

برای نمایش متغیرهای  $x$  و  $y$  به صورت ژن و از رویکرد مبنای دو استفاده شد. برای افزایش دقت و تولید اعداد بین صفر تا ۱۰، اعدادی بر مبنای دو تولید می شود که بین صفر تا ۱۰۲۳ است و سپس عدد تولید شده بر عدد ۱۰۰ تقسیم می شود. برای نمونه مثال زیر را در نظر بگیرید.

$$(x'')_2 = (0, 0, 1, 1, 0, 0, 1, 1, 0, 0)$$

$$\begin{aligned} \rightarrow (x')_{10} &= 0 \times 2^0 + 0 \times 2^1 + 1 \times 2^2 + 1 \times 2^3 + 0 \times 2^4 + 0 \times 2^5 + 1 \times 2^6 + 1 \times 2^7 + 0 \times 2^8 + 0 \times 2^9 \\ &= 4 + 8 + 64 + 128 = 204 \end{aligned}$$

برای تولید یک مقدار برای متغیر  $x$  تنها کافی است مقدار تولید شده در بالا را بر ۱۰۰ تقسیم

نماییم. در این صورت مقدار تولید شده برای  $x$  برابر با ۲.۰۴ می شود.

# حل یک مثال

ژن‌های مربوط به متغیر  $Y$

ژن‌های مربوط به متغیر  $X$

| شماره ژن |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   | شماره جمعیت |
|----------|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|-------------|
| ۱۹       | ۱۸ | ۱۷ | ۱۶ | ۱۵ | ۱۴ | ۱۳ | ۱۲ | ۱۱ | ۱۰ | ۹ | ۸ | ۷ | ۶ | ۵ | ۴ | ۳ | ۲ | ۱ | ۰ |             |
| 1        | 0  | 1  | 0  | 1  | 1  | 1  | 0  | 0  | 0  | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | ۱           |
| 1        | 1  | 0  | 0  | 1  | 0  | 0  | 0  | 0  | 1  | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | ۲           |
| 1        | 0  | 0  | 1  | 0  | 1  | 0  | 1  | 1  | 1  | 1 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 1 | ۳           |
| ⋮        |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |             |
|          |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |             |
|          |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |             |
|          |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |             |
| 0        | 1  | 1  | 0  | 0  | 1  | 1  | 1  | 0  | 0  | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | ۱۷          |
| 0        | 1  | 0  | 0  | 1  | 1  | 0  | 0  | 1  | 0  | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | ۱۸          |
| 1        | 0  | 0  | 0  | 0  | 1  | 1  | 0  | 1  | 0  | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | ۱۹          |
| 1        | 1  | 0  | 0  | 1  | 1  | 1  | 0  | 0  | 1  | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | ۲۰          |

## حل یک مثال

### روش چرخه رولت

در این قسمت از میان جمعیت دو عضو برای انجام عملیات تقاطع به عنوان والد انتخاب می شود. با توجه به این که در این مسئله به دنبال کمینه کردن مقدار تابع هدف هستیم، لذا در روش چرخه رولت، معکوس تابع هدف را به عنوان تابع برازندگی این چرخه در نظر می گیریم. هر چه مقدار تابع هدف بیشتر باشد برازندگی آن کمتر است و در این صورت در روش چرخه رولت با احتمال کمتری انتخاب می شود.

## حل یک مثال

### عملگر تقاطع

در این قسمت از عملگر ساده تقاطع استفاده می شود به این صورت که دو عضو از جمعیت با استفاده از چرخه رولت به صورت تصادفی انتخاب می شود. سپس یک عدد بین ۰ تا ۱۹ انتخاب می شود و از آن محل تقاطع، عملگر تقاطع بین این دو والد انجام می شود و نتیجه دو فرزند می شود. مقدار تابع هدف این دو فرزند محاسبه می شود. در صورتیکه مقدار تابع هدف این دو فرزند بهتر از اعضای جمعیت باشد در جمعیت جایگزین می شود.

## حل یک مثال

### عملگر جهش

عملگر جهش در این مسئله برابر با ۰, ۱ در نظر گرفته شد و به این صورت عمل می شود که یک عدد تصادفی بین صفر و یک تولید می شود. اگر عدد تولید شده کمتر از ۰, ۱ باشد، این عملگر اجرایی می شود و در این صورت یک عدد تصادفی بین صفر و ۱۹ تولید می شود. ژن مربوط به عدد تولید شده در صورتی که صفر باشد به یک و در صورتیکه یک باشد به صفر تغییر می کند.

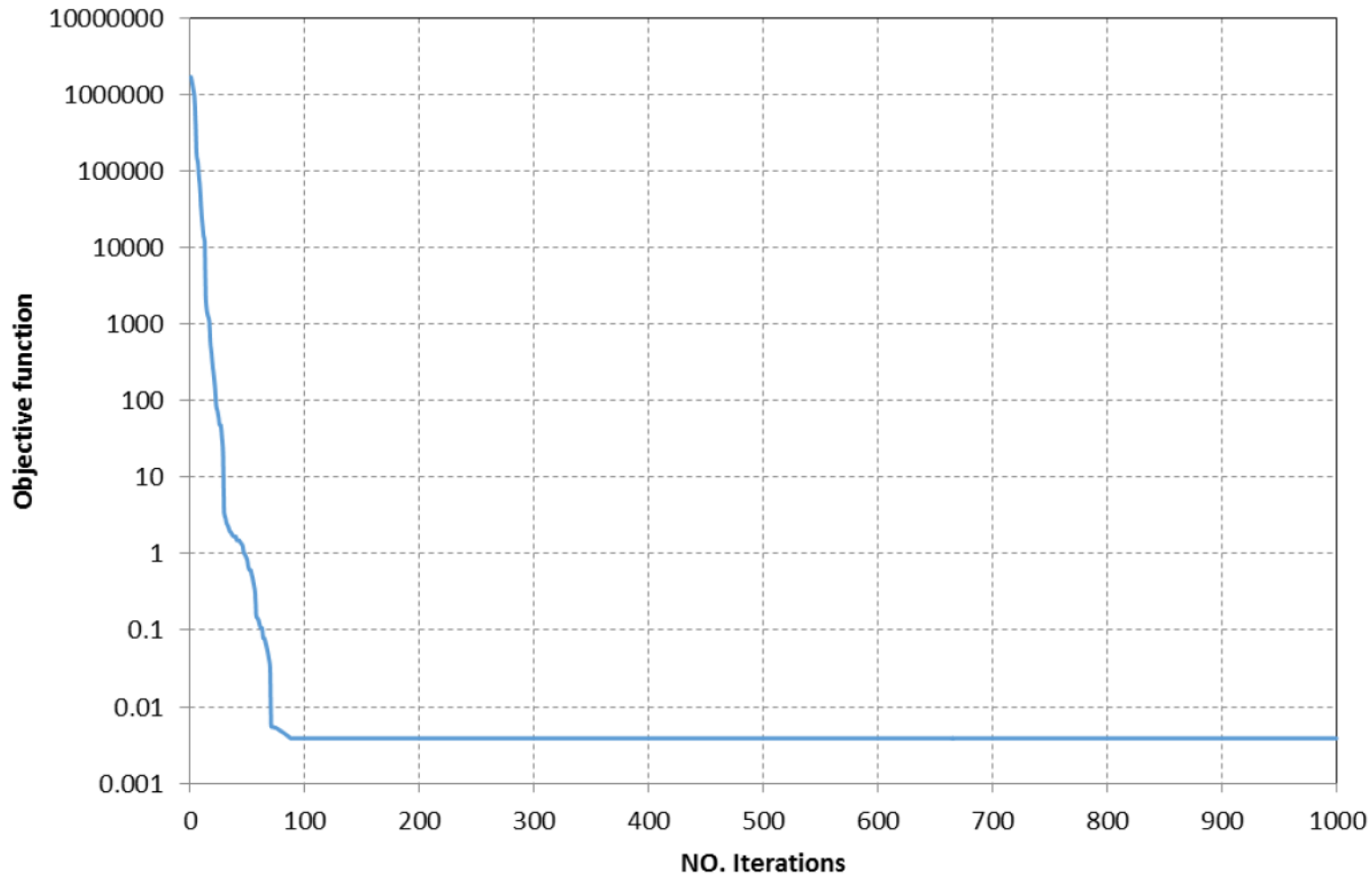
## حل یک مثال

### شرط توقف

شرط توقف زمانی برقرار می شود که ۱۰۰۰ تکرار انجام شود.

پس از اجرای برنامه فوق، الگوریتم ژنتیک توانست به جواب بهینه تابع هدف با دقت مناسبی نزدیک شود. در شکل زیر تغییرات مقدار میانگین تابع هدف جمعیت برای تکرارهای مختلف آورده شده است.

# حل یک مثال



# با تشکر

راه های ارتباطی با ما

[www.behinehyab.com](http://www.behinehyab.com)

[behinehyab@gmail.com](mailto:behinehyab@gmail.com)