

# مساله کد کردن

چگونگی کد کردن راه حل یک مساله به یک کروموزوم، موضوع کلیدی برای الگوریتم‌های ژنتیک است. در نگارش اولیه این الگوریتم و در کارهای Holland، کد کردن، توسط یک رشته دودویی انجام می‌شد.

برای بسیاری کاربردها، استفاده مستقیم از الگوریتم ژنتیک پایه با کدینگ باینری مشکل است، زیرا رشته دودویی یک روش کد کردن طبیعی نیست.

در خلال ۲۰ سال گذشته روش‌های مختلف کد کردن غیر دودویی ابداع شده است، مانند استفاده از اعداد حقیقی برای مسائل بهینه‌سازی با محدودیت (Constrained Optimization).

# مساله کد کردن

---

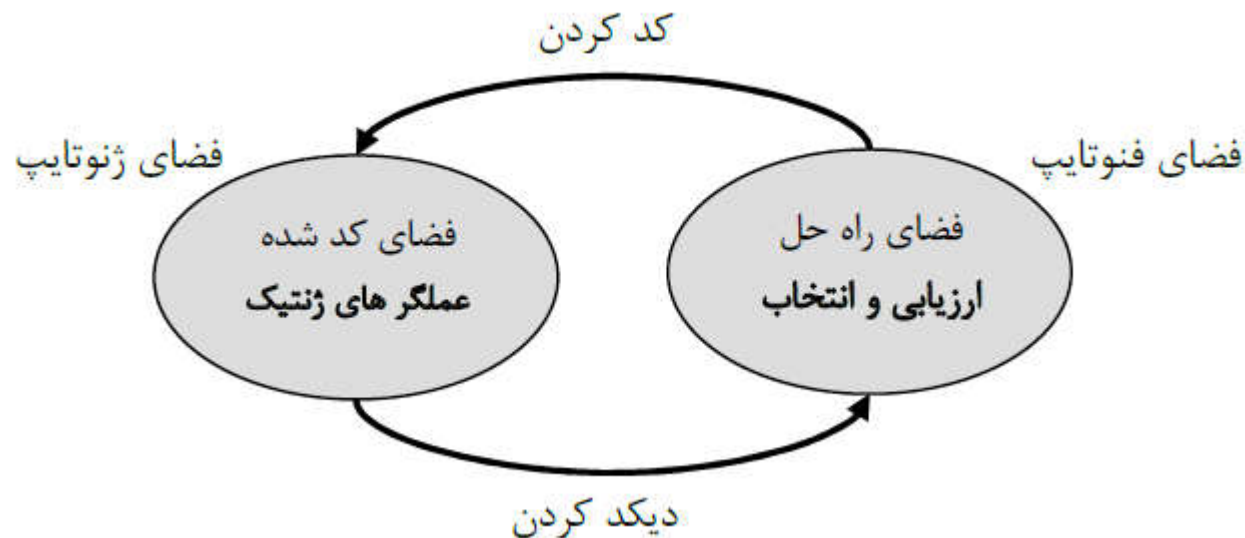
برای هر مساله کاربردی، لازم است که ابتدا یک بررسی برای روش نمایش راه حل مساله انجام شود.

یکی از ویژگی‌های الگوریتم‌های ژنتیکی این است که آن‌ها هم بر روی **فضای کد شده (Coding Space)** و هم بر روی **فضای راه حل (Solution Space)** کار می‌کنند.

عملگرهای ژنتیکی بر روی فضای کد شده (کروموزوم) کار کرده در حالیکه عملگرهای ارزیابی و انتخاب بر روی فضای راه حل، کار می‌کنند.

# مساله کد کردن

در این حالت انتخاب طبیعی همانند رابطی بین کروموزوم‌ها و پاسخ دیکد شده آنها عمل می‌کند.



# مساله کد کردن

---

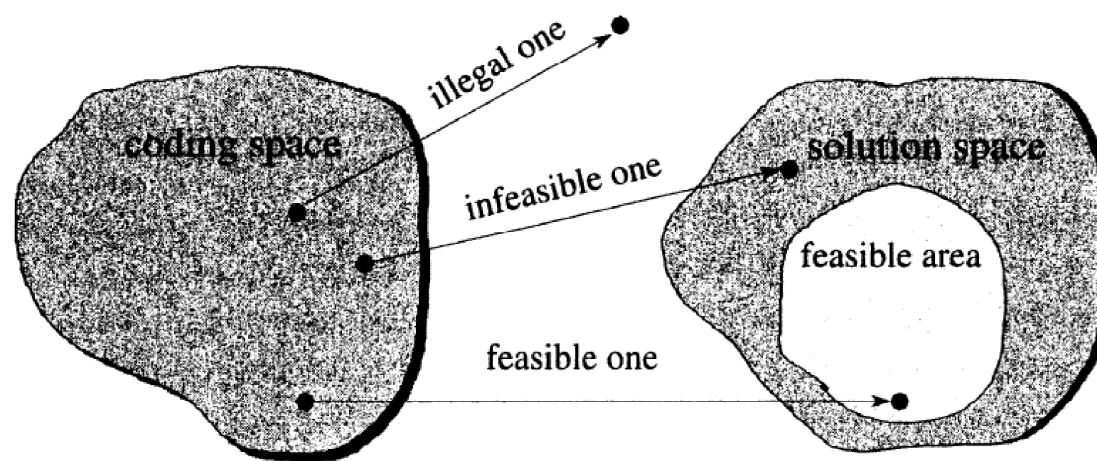
در روش کد کردن حقیقی، سه موضوع مهم بین کد کردن و دیکد کردن کروموزوم‌ها و راه حل‌ها وجود دارد:

- امکان پذیر بودن کروموزوم (Feasibility)
- قانونی (مجاز) بودن کروموزوم (Legality)
- یکتا بودن نگاشت (Uniqueness)

# مساله کد کردن

Feasibility به این معناست که آیا راه حل کد شده از یک کروموزوم، در ناحیه امکان پذیر قرار می گیرد، یا خیر.

Legality به این مساله اشاره داد که آیا یک کروموزوم، نشان دهنده یک راه حل برای مساله داده شده می باشد و یا خیر.



## مساله کد کردن

امکان ناپذیر بودن (Infeasibility) یک کروموزوم، از طبیعت مساله بهینه سازی با محدودیت، سر چشمه می گیرد.

تمامی روش ها، چه روش های معمول و چه الگوریتم های ژنتیکی، باید محدودیت های حاکم بر مساله را رعایت نمایند.

در حل بسیاری از مسائل بهینه سازی، ناحیه Feasible (امکان پذیر) توسط تعدادی معادله یا نامعادله مشخص می شود. در چنین مواردی، تعدادی روش کارآمد **جریمه (Penalty)** وجود دارد که برای اداره کردن این محدودیت ها ارائه شده است.

## مساله کد کردن

در مسائل بهینه سازی با محدودیت، نقطه بهینه معمولاً در مرز نواحی Feasible و Infeasible یافت می‌شود. در این حالت رهیافت استفاده از جریمه، جستجوی ژنتیک را مجبور به نزدیک شدن به نقطه بهینه در منطقه Feasible می‌کند.

علت تولید کروموزوم‌های غیر مجاز (Illegal)، از طبیعت روش‌های کد کردن سرچشمه می‌گیرد. در حل بسیاری از مسائل بهینه سازی ترکیبی، از روش‌های کدینگی استفاده می‌شود که این روش‌ها، با یک عملگر ساده آمیزش تک نقطه‌ای نیز سبب تولید یک کروموزوم غیرمجاز می‌شوند.

## مساله کد کردن

---

چون این کروموزوم‌ها را نمی‌توان به عنوان راه حل‌های مساله مورد ارزیابی قرار داد، بنابر این روش جریمه را نمی‌توان بر روی این راه حل‌ها اعمال نمود.

از این روش‌های ترمیم، معمولاً برای تبدیل یک کروموزوم Illegal به کروموزوم Legal انتخاب می‌شود.



# مساله کد کردن

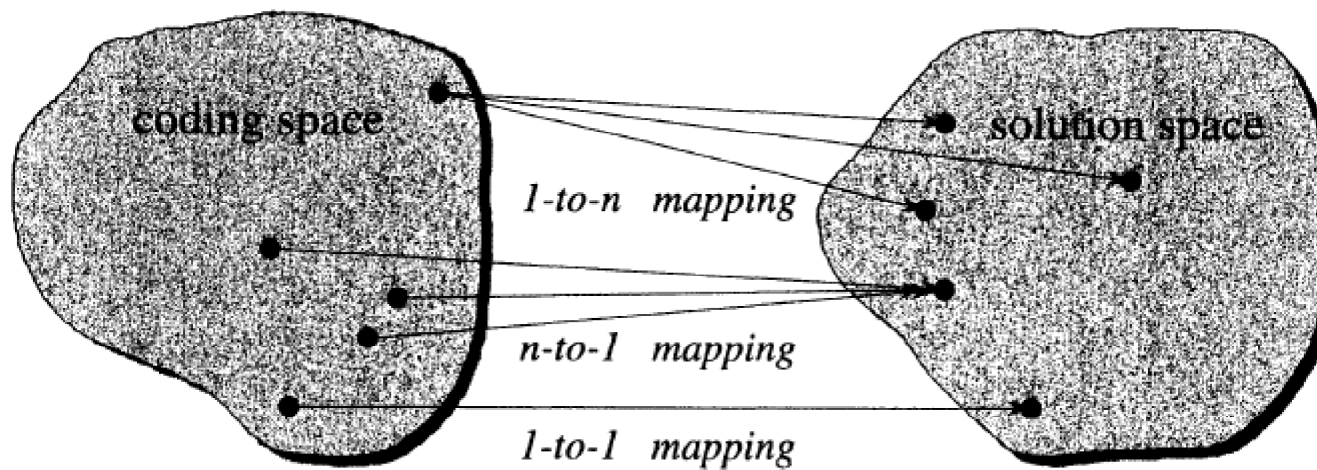
**نگاشت (Mapping)** از کروموزومها به راه حلها (دیکد کردن) به یکی از سه

روش زیر صورت می گیرد:

- نگاشت یک به یک
- نگاشت یک به  $n$
- نگاشت  $n$  به یک

نگاشت یک به یک بهترین روش در میان سه مورد است و نگاشت یک به  $n$  بدترین مورد از انواع نگاشت است.

# مساله کد کردن



نگاشت از کروموزوم‌ها به راه‌حل‌ها

# انتخاب

---

یکی از اصول مهم کارکرد الگوریتم ژنتیک، رویه انتخاب طبیعی داروین است. عملگر انتخاب، نیروی جلو برنده الگوریتم ژنتیک را فراهم می‌سازد که رویه انتخاب در آن نقش مهمی دارد.

در این حالت از یک سو ممکن است که رویه جستجو به صورت ناگهانی خاتمه یابد و از سوی دیگر ممکن است که سرعت جستجو از سرعت طبیعی آن پایین‌تر باشد.

# انتخاب

---

نیروی کم انتخاب، در ابتدای الگوریتم ژنتیک دلالت بر اکتشاف وسیع فضای جستجو است و افزایش آن در آخر جستجو، یادآور استخراج نواحی امیدوار کننده فضای جستجو می باشد.

در حقیقت عملگر انتخاب در الگوریتم ژنتیک رویه جستجو را به سمت نواحی امیدوار کننده فضای جستجو هدایت می کند.

# انتخاب

---

سه موضوع اصلی در مرحله انتخاب وجود دارد که عبارتند از:

- فضای نمونه برداری (Sampling Space)
- روش نمونه برداری (Sampling Mechanism)
- احتمال انتخاب (Selection Probability)

# فضای نمونه برداری

---

رویه انتخاب در هر چرخه از الگوریتم ژنتیک، نسل بعد را یا از تمام والدین و فرزندان تولید می کند و یا از قسمتی از آنها، که از آن تحت عنوان مساله فضای نمونه برداری یاد می شود. فضای نمونه برداری با دو عامل زیر مشخص می شود:

- اندازه (Size)
- اجزاء ترکیب (والدین و فرزندان) – Ingredient

## فضای نمونه برداری

در این حالت اگر  $PopSize$  نشان دهنده اندازه جمعیت اولیه و  $OffSize$  نشان دهنده جمعیت فرزندان تولید شده در هر نسل باشد، فضای نمونه برداری منظم (Regular Sampling Space) دارای اندازه جمعیت  $PopSize$  و در برگیرنده تمام فرزندان و قسمتی از والدین است.

فضای نمونه برداری بسط یافته (Enlarged Sampling Space) دارای اندازه  $PopSize + OffSize$  و شامل تمامی فرزندان و والدین است.

# فضای نمونه برداری منظم

در الگوریتم ژنتیک پایه که توسط John Holland معرفی شد، فرزندان بلافاصله بعد از تولید جایگزین والدین خود می‌شدند.

به این روش، **جایگزینی نسلی** (Generational Replacement) گفته می‌شود. چون الگوریتم ژنتیک در اصل کورکورانه است، در این حالت ممکن است فرزندان بدتر از والدین خود شوند.

با استراتژی جایگزینی مستقیم هر کدام از والدین با فرزندان، ممکن است کروموزوم‌های بهتر از بین بروند. برای حل این مشکل استراتژی‌های جایگزینی گوناگونی معرفی شدند.



# فضای نمونه برداری منظم

در اولین اصلاح Holland پیشنهاد کرد تا هر فرزند تولید شده به صورت تصادفی جایگزین یک کروموزوم از نسل حاضر شود.

در ادامه De Jong **استراتژی ازدحام** (Crowding Strategy) را ارائه داد. در این مدل هنگامی که یک فرزند تولید می شود یک والد برای از بین رفتن انتخاب می شود.

والد حذف شونده آن والدی است که شباهت زیادی با فرزند دارد که میزان این شباهت با شمارش بیت های مشابه با فرزندان به دست می آید.

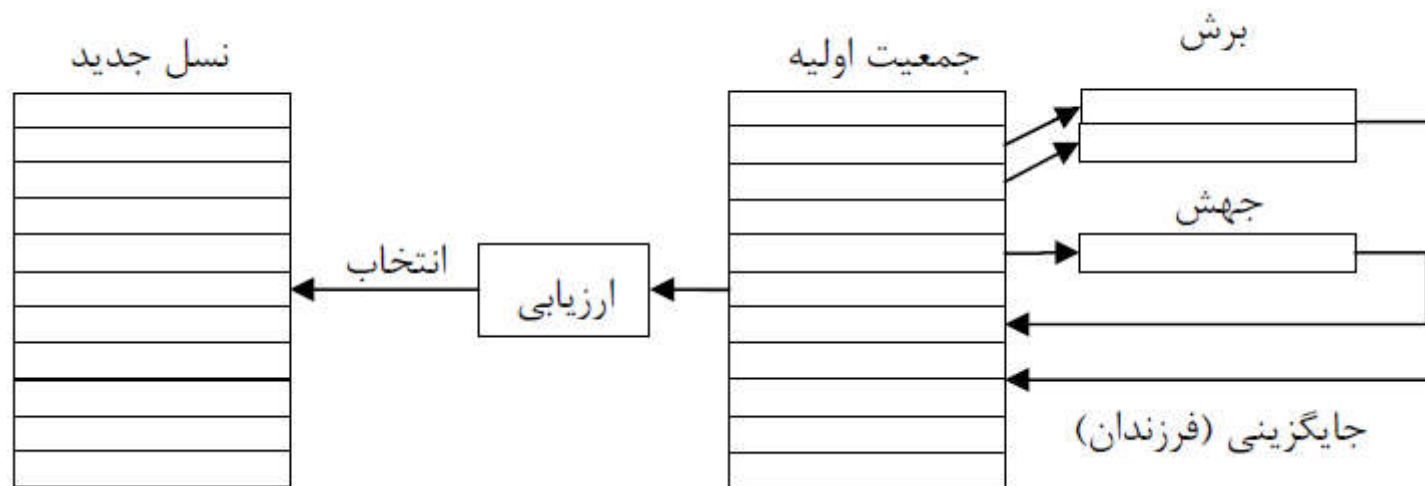
# فضای نمونه برداری منظم

---

نکته مهم این است که در کارهای Holland، رویه انتخاب اشاره به گزینش والدین به منظور انجام عمل آمیزش دارد.

# فضای نمونه برداری منظم

Michalewicz اصلاحاتی بر روی الگوریتم ژنتیک ساده ارائه کرد که فرزندان در هر نسل بلافاصله بعد از تولید، با استفاده از رویه چرخه رولت جایگزین والدینشان می‌شوند.



# فضای نمونه برداری بسط یافته

---

هنگام انجام عمل انتخاب بر روی فضای نمونه برداری بسط یافته، عموماً والدین و فرزندان دارای شانس یکسانی برای بقا هستند. یک نمونه ساده روش انتخاب  $\mu + \lambda$  است.

این روش در اصل در استراتژی‌های تکاملی ارائه شده است. Bäck و Hoffmeister استفاده از این روش را در الگوریتم ژنتیک معرفی کردند.

## فضای نمونه برداری بسط یافته

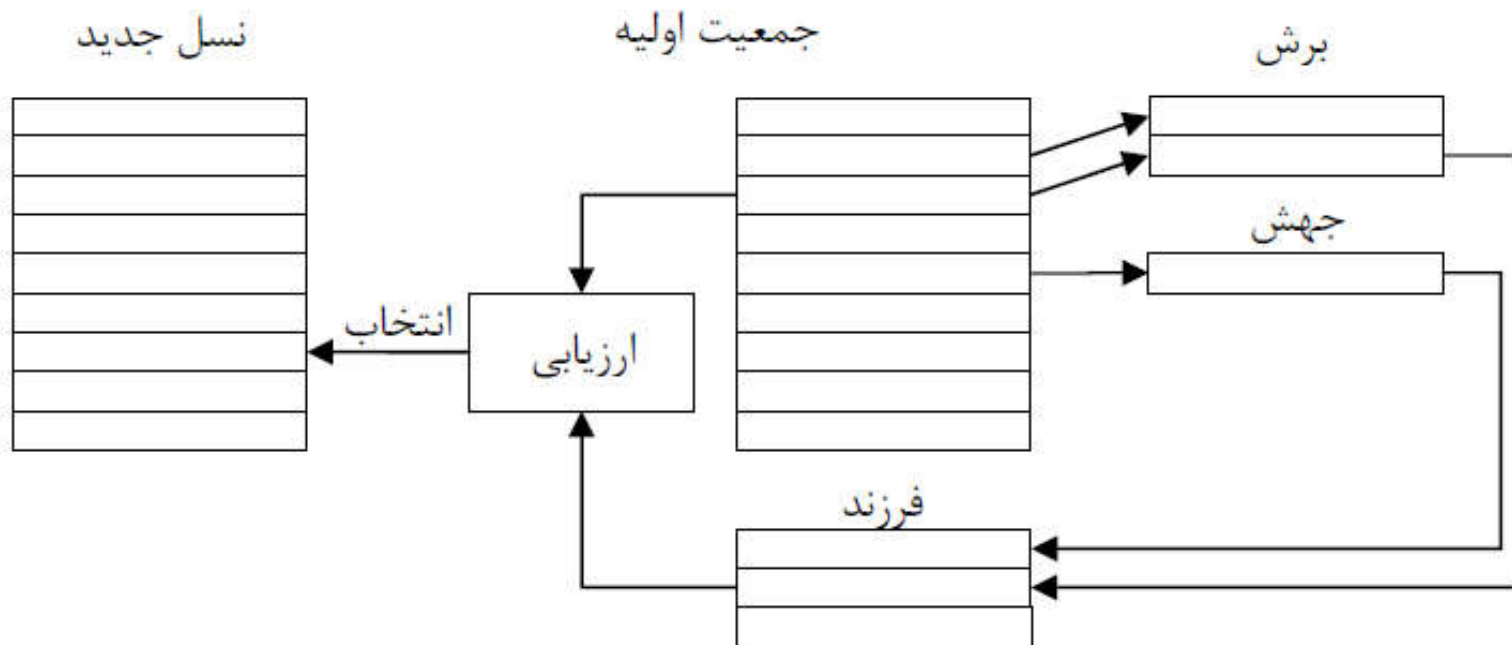
در این روش، تعداد  $\mu$  والد و  $\lambda$  فرزند برای بقا در نسل بعدی با هم رقابت می کنند و نهایتاً  $\mu$  پاسخ برای نسل بعد به دست می آید.

حالت بعدی، انتخاب  $(\mu, \lambda)$  است که  $\mu$  فرزند به عنوان والدهای نسل بعد انتخاب می شوند  $(\mu < \lambda)$ .

هر دو این روش ها، از جمله روش های قطعی هستند ولی می توانیم پیاده سازی احتمالی هم بر روی آنها انجام دهیم.

# فضای نمونه برداری بسط یافته

شکل زیر انتخاب بر اساس فضای نمونه بسط یافته را نشان می‌دهد:



## فضای نمونه برداری بسط یافته

---

مهم‌ترین فایده استفاده از این روش این است که می‌توانیم کارایی الگوریتم ژنتیک را با افزایش نرخ آمیزش و جهش بهبود بخشیم.

به عبارت دیگر، اگر روش انتخاب ما بر اساس فضای نمونه برداری بسط یافته باشد، مقادیر بالای نرخ عملگرهای ژنتیک، پراکندگی تصادفی زیادی تولید نمی‌کنند.

# روش‌های نمونه برداری

---

روش نمونه برداری به چگونگی انتخاب کروموزوم‌ها از فضای نمونه برداری مربوط است. سه روش پایه برای نمونه برداری کروموزوم‌ها وجود دارد که عبارتند از:

- نمونه برداری اتفاقی (Stochastic Sampling)
- نمونه برداری قطعی (Deterministic Sampling)
- نمونه برداری ترکیبی (Mixed Sampling)



## نمونه برداری اتفاقی

ویژگی معمول این کلاس‌ها همان طور که Baker اشاره کرده است، فاز انتخاب است که تعداد واقعی کپی‌های هر کروموزوم بر اساس احتمال بقا به دست می‌آید. بنابر این، فاز انتخاب ترکیبی از دو بخش است:

- تعیین امید ریاضی هر کروموزوم
- تبدیل امید ریاضی هر کروموزوم به تعداد فرزندان

## نمونه برداری اتفاقی

مقدار امید ریاضی یک کروموزوم، مقداری حقیقی است که تعیین کننده میانگین تعداد فرزندی است که باید از آن والد تولید شوند. در حقیقت در روال نمونه برداری، رویه‌ای برای تبدیل مقدار حقیقی امید ریاضی به تعداد فرزندان مورد انتظار استفاده می‌شود.

معروف‌ترین روش در این دسته، **انتخاب نسبی** یا **انتخاب چرخه رولت** است. ایده اصلی این روش، مشخص کردن احتمال انتخاب (احتمال بقا) برای هر کروموزوم متناسب با مقدار تطابق آن نسبت به تطابق کل جمعیت است.

## نمونه برداری اتفاقی

برای  $k$ -امین کروموزوم، با مقدار تطابق  $f_k$  احتمال انتخاب  $p_k$  به صورت زیر محاسبه می‌شود:

$$p_k = \frac{f_k}{\sum_{j=1}^{PopSize} f_j}$$

سپس می‌توانیم چرخه بر اساس این احتمالات درست کنیم. فرآیند انتخاب بر اساس چرخش چرخه رولت به تعداد  $PopSize$  است و در هر چرخش یک کروموزوم را انتخاب می‌کنیم که شرح جزییات آن در قبل گفته شده است.

# نمونه برداری اتفاقی

همچنین در این دسته Baker روش *Stochastic universal sampling* را معرفی کرد که از یک چرخه منفرد استفاده می کند. در این روش تعداد مورد انتظار  $e_k$  برای کروموزوم  $k$  به صورت زیر محاسبه می شود:

$$e_k = PopSize \times p_k$$

رویه انجام این روش انتخاب در ادامه آورده شده است.

# نمونه برداری اتفاقی

---

## Procedure : Stochastic Universal Sampling

```
begin
  sum = 0
  ptr = rand
  for k=1 to Pop_Size do
    sum = sum + ek
    while (sum > prt) do
      select chromosome k
      prt = prt + 1
    end
  end
end
end
```

# نمونه برداری اتفاقی

هدف اصلی از این رویه حفظ یک تعداد کپی از هر کروموزوم در نسل آینده است. این امر به دو دلیل صورت می‌گیرد:

- جلوگیری از به وجود آمدن یک **ابر کروموزوم (Super Chromosome)** که بر کل جمعیت غلبه نماید. عدم رعایت این مساله سبب همگرایی سریع الگوریتم به یک بهینه محلی می‌شود.
- حفظ **پراکندگی (Diversity)** جمعیت. این مساله سبب می‌شود تا جمعیت اطلاعات بیشتری را در خود برای هر تولید نسل در بر داشته باشد.

# نمونه برداری قطعی

این رویکرد معمولاً به تعداد  $PopSize$  کروموزوم از جمعیت نسل فعلی انتخاب می‌کند. دو روش انتخاب  $\mu + \lambda$  و  $\mu, \lambda$  به این دسته تعلق دارد.

روش نمونه برداری قطعی Brindle بر پایه مفهوم **تعداد مورد انتظار** (Expected Number) است. در این روش تعداد مورد انتظار کپی از هر کروموزوم طبق معادله زیر حساب می‌شود.

$$p_k = \frac{f_k}{\sum_{j=1}^{Pop\_Size} f_j}$$

$$e_k = p_k \times PopSize$$

## نمونه برداری قطعی

---

هر کروموزوم به ازاء جز صحیح تعداد مورد انتظار، برای نسل بعد کپی خواهد داشت.

سپس جمعیت بر اساس قسمت اعشاری تعداد مورد انتظار مرتب شده و باقی مانده کروموزوم های لازم به منظور تکمیل اعضای جمعیت، از بالای لیست مرتب شده انتخاب خواهند شد.



## نمونه برداری قطعی

جایگزینی نسلی را نیز می‌توان جز روش‌های قطعی دانست. تغییری بر روی این روش، جانشین کردن  $n$  کروموزم فرزند به جای  $n$  کروموزم بد است که اولین بار توسط Whitley معرفی شده است.

Michalewicz یک نسخه تصادفی از این روش به نام الگوریتم ژنتیک بهبود یافته ارائه داد (modGA). کروموزوم‌هایی که باید باقی بمانند بر اساس احتمال زنده بودنشان انتخاب می‌شوند. در این روش کروموزوم‌هایی که از میانگین تطابق مجموع کروموزوم‌ها، بدتر هستند شانس بیشتری برای از بین رفتن دارند.

# نمونه برداری ترکیبی

این روش همزمان، شامل ویژگی‌های دو روش تصادفی و قطعی است. یک مثال از این روش **انتخاب مسابقه (Tournament Selection)** است که توسط Goldberg معرفی شده است. این روش به این صورت است که مجموعه‌ای تصادفی از کروموزوم‌ها انتخاب می‌شود و بهترین اعضای این مجموعه برای تولید جدا می‌شوند.

به اندازه جمعیت انتخابی، **اندازه مسابقه (Tournament Size)** گفته می‌شود. اندازه معمول برای اکثر روش‌ها برابر دو است و به آن **مسابقه دودویی (Binary Tournament)** نیز گفته می‌شود.

# نمونه برداری ترکیبی

روش انتخاب مسابقه تصادفی (Stochastic Tournament Selection)

توسط Wetzel ارائه شده است. در این روش، احتمالات انتخاب به صورت نرمال محاسبه می‌شوند و جفت کروموزوم‌ها توسط انتخاب چرخه رولت به دست می‌آیند.

بعد از قرعه کشی یک جفت کروموزوم، کروموزومی که تطابق بیشتری دارد در نسل جدید اضافه می‌شود. این فرآیند تا کامل شدن جمعیت ادامه می‌یابد.

# نمونه برداری ترکیبی

روش نمونه برداری تصادفی باقی مانده (Remainder Stochastic Sampling) توسط Brindle ارائه شد که تغییر یافته نسخه نمونه برداری قطعی خودش است.

در این روش، برای هر کروموزوم بر اساس جز صحیح تعداد مورد انتظار، نمونه اختصاص داده می‌شود، سپس کروموزوم‌ها برای باقی ماندن در نسل جدید بر اساس قسمت اعشاری تعداد مورد انتظار مسابقه می‌دهند.

## احتمال انتخاب

در این قسمت به چگونگی محاسبه احتمال انتخاب هر کروموزوم می‌پردازیم. به طور کلی احتمال انتخاب یک کروموزوم متناسب با مقدار تطابق آن کروموزوم است. با این حال این رویه چالش‌هایی را نیز در بر دارد.

برای مثال، در نسل‌های اولیه، تمایل الگوریتم بر روی انتخاب تعداد کمی سوپر کروموزوم است، حال آنکه در نسل‌های آخر که جمعیت همگرایی زیادی دارد، رقابت بین کروموزوم‌ها کم است و رفتار جستجوی تصادفی بروز می‌کند.

# احتمال انتخاب

مکانیزم‌های مقیاس بندی (Scaling) و رتبه بندی (Ranking) برای کم کردن این اثرات در حل مسائل ارائه شده‌اند.

روش مقیاس بندی، مقادیر تابع هدف خام را به تعدادی مقادیر صحیح مثبت نگاشت می‌کند و احتمال بقا برای هر کروموزوم بر اساس این مقادیر مشخص می‌شود.

روش رتبه بندی استفاده مستقیم از مقادیر خام واقعی تابع هدف را نادیده گرفته و از یک رتبه بندی برای برای مشخص کردن احتمال بقا استفاده می‌کند.

# احتمال انتخاب

روش Scaling با دو هدف انجام می‌شود:

- برای حفظ تفاوت معقول بین مقادیر تطابق کروموزوم‌ها
- برای جلوگیری از تمام شدن خیلی سریع الگوریتم ژنتیک به وسیله تعدادی اَبَر کروموزوم.

در بسیاری از روش‌های مقیاس بندی، پارامترهای مقیاس بندی وابسته به مساله هستند.

رتبه بندی تطابق، تأثیر مشابهی با مقیاس بندی دارد با این تفاوت که نیازی به تعیین پارامترها ندارد.

# احتمال انتخاب

در حالت کلی، تطابق Scale شده  $f'_k$  از تطابق خام  $f_k$  برای کروموزوم  $k$  می‌تواند به صورت زیر به دست آید:

$$f'_k = g(f_k)$$

که تابع  $g(.)$  تطابق خام را به تطابق مقیاس شده تبدیل می‌کند. تابع  $g(.)$  ممکن است صورت‌های مختلفی داشته باشد تا روش‌های مقیاس بندی مختلفی را تولید نماید، از جمله روش‌های Linear Scaling، Power Law Scaling، Logarithmic Scaling و غیره.



# احتمال انتخاب

---

این روش‌ها به دو گروه زیر تقسیم می‌شود:

- سنجش ایستا
- سنجش پویا

رابطه نگاشت بین تطابق مقیاس شده و تطابق خام اگر ثابت باشد، سنجش ایستا به دست می‌آید و اگر بنابر برخی فاکتورها متغیر باشد روش پویا نامیده می‌شود.

# احتمال انتخاب

---

روش‌های مقیاس بندی پویا به دو بخش تقسیم می‌شوند:

- پارامترهای مقیاس بندی به صورت تطبیقی بر اساس میزان پراکندگی مقدار تطابق در هر نسل، به منظور ثابت نگه داشتن نیروی انتخاب، تنظیم شوند.
- پارامترهای سنجش به صورت پویا، بر اساس بالا رفتن شماره نسل، به منظور زیاد کردن نیروی انتخاب تغییر کنند.

# احتمال انتخاب

## مقیاس بندی خطی (Linear Scaling)

مقیاس بندی خطی مقادیر تطابق تمام کروموزوم‌ها را تنظیم می‌کند به طوری که بهترین کروموزوم یک عدد ثابت را به عنوان تعداد فرزندان مورد انتظار دریافت می‌کند. بنابراین از تولید خیلی زیاد آن کروموزوم جلوگیری می‌شود.

در این حالت تابع  $g(.)$  فرم خطی خواهد داشت و می‌توان آن را به صورت زیر تعریف نمود:

$$f'_k = a \times f_k + b$$

## احتمال انتخاب

---

پارامترهای  $a$  و  $b$  به صورت خطی انتخاب می‌شوند بنابراین، کروموزوم‌های متوسط به طور میانگین یک فرزند تولید می‌کنند و بهترین کروموزوم تعداد مشخصی کپی از خود تولید می‌کند که معمولاً این تعداد برابر دو است.

# احتمال انتخاب

## مقیاس بندی خطی پویا (Dynamic Linear Scaling)

هنگامی که پارامتر  $b$  با شماره نسل تغییر کند، رویه زیر قابل تعریف است:

$$f'_k = a \times f_k + b_t$$

یک تعریف ممکن برای  $b_t$  اختصاص کمترین مقدار تطابق خام نسل حاضر است.

$$b_t = -f_{\min}$$

# احتمال انتخاب

## مقیاس بندی توانی (Power Law Scaling)

این روش توسط Gillies ارائه شد به طوریکه تابع  $g(.)$  را به صورت توانی از تطابق خام در نظر می گیرد و به صورت زیر است:

$$f'_k = f_k^\alpha$$

به طور کلی مقدار  $\alpha$  وابسته به مساله است ولی Gillies مقدار ۱.۰۰۵ را برای آن ارائه داده است.

# احتمال انتخاب

---

در این روش فاصله بین تطابق مقیاس شده بهترین و بدترین کروموزوم با مقدار  $\alpha$  بیشتر می شود.

زمانی که  $\alpha$  به صفر میل می کند، این فاصله هم به صفر میل می کند و نمونه برداری به جستجوی تصادفی تبدیل می شود.

هنگامیکه  $\alpha > 1$  باشد این فاصله زیاد می شود و نمونه برداری به اختصاص کروموزوم برتر تبدیل می شود.

## احتمال انتخاب

---

در یک اجرا ممکن است لازم باشد تا مقدار  $\alpha$  به صورت پویا تنظیم شود تا این فاصله را کم و زیاد کند. از این رو تعریف دیگری برای مقیاس بندی توانی به صورت زیر است:

$$f'_k = (a \times f_k + b)^\alpha$$



# احتمال انتخاب

## مقیاس بندی لگاریتمی (Logarithmic Scaling)

این روش توسط Fitzpatrick به منظور نگاشت تابع هدف برای مسائل بهینه سازی معرفی شده است که در آن تابع  $g(\cdot)$  دارای فرم لگاریتمی به صورت زیر است:

$$f'_k = b - \log(f_k)$$

که در آن  $b$  به نحوی انتخاب می شود که بزرگتر از هر مقدار  $\log(f_k)$  باشد.

# احتمال انتخاب

## پنجره سازی (Windowing)

فرض کنیم که دارای دو کروموزوم با مقدار تطابق ۲ و ۱ هستیم. در این حالت تعداد فرزندان کروموزوم اول به طور متوسط ۲ برابر تعداد فرزندان کروموزوم دوم است.

حال اگر تابع مورد بررسی تنها با افزودن ۱ واحد به آن تغییر کند مقدار تطابق حاصل برای دو کروموزوم فوق ۳ و ۲ خواهد بود که نشان دهنده نسبت ۱.۵ برای تولید فرزندان خواهد بود.

## احتمال انتخاب

در این حالت مشاهده می‌شود که تغییر کوچکی در تابع مورد بررسی، تاثیر زیادی بر روی رویه پیشرفت خواهد داشت.

روش پنجره سازی یک تکنیک Moving Baseline را معرفی می‌کند تا مقدار ثابتی از نیروی انتخاب را در طول رویه تکامل حفظ کند. این روش به صورت زیر قابل بیان است:

$$f'_k = f_k - f_w$$

که در آن  $w$  به عنوان اندازه پنجره شناخته می‌شود و عموماً دارای مرتبه ۲ الی ۱۰ است و  $f_w$  بدترین مقدار دیده شده در  $w$  گام قبلی است.

# احتمال انتخاب

## نرمال سازی (Normalization)

نرمال سازی توسط Cheng و Gen به عنوان تکنیکی از گروه مقیاس بندی پویا ارائه شد. این روش برای مسائل بیشینه سازی از رابطه زیر استفاده می کنند:

$$f'_k = \frac{f_k - f_{\min} + \gamma}{f_{\max} - f_{\min} + \gamma}$$

که  $f_{\max}$  و  $f_{\min}$  به ترتیب بدترین و بهترین تطابق خام در نسل حاضر است.  $\gamma$  یک عدد حقیقی مثبت است که معمولاً در باز (0,1) محدود است.

# احتمال انتخاب

فواید استفاده از این تبدیل به شرح زیر است:

- جلوگیری از تقسیم بر صفر.
- امکان تنظیم رفتار رویه انتخاب از گزینش متناسب با تطابق تا انتخاب تصادفی.

برای کمینه سازی از تابع زیر استفاده می شود:

$$f'_k = \frac{f_{\max} - f_k + \gamma}{f_{\max} - f_{\min} + \gamma}$$

# احتمال انتخاب

---

## انتخاب بولتزمن (Boltzman)

این روش دارای رابطه زیر است:

$$f'_k = e^{f_k / T}$$

در این حالت نیروی انتخاب زمانی که مقدار  $T$  زیاد باشد کاهش پیدا می کند.

# احتمال انتخاب

## رتبه بندی (Ranking)

Baker مفهوم انتخاب رتبه بندی را برای الگوریتم های ژنتیک ارائه داد تا مشکل روش های سنجش بر پایه تطابق را حل کند. ایده آن به صورت زیر است:

جمعیت را بر اساس تطابق از بهترین به بدترین کروموزوم مرتب می کنیم. سپس احتمال انتخاب هر کروموزوم را بر اساس رتبه آن حساب خواهیم کرد و نه براساس تطابق خام آن.

# احتمال انتخاب

---

دو روش برای رتبه‌بندی وجود دارد:

- رتبه‌بندی خطی
- رتبه‌بندی نمایی

اگر  $p_k$  احتمال انتخاب برای  $k$ -امین کروموزوم در جمعیت رتبه‌بندی شده باشد، رتبه‌بندی خطی به صورت زیر است:

$$p_k = q - (k - 1) \times r$$



## احتمال انتخاب

که  $q$  احتمال بهترین کروموزوم است. اگر  $q_0$  احتمال برای بدترین کروموزوم باشد پارامتر  $r$  به صورت زیر محاسبه می شود:

$$r = \frac{q - q_0}{PopSize - 1}$$

مقدار تطابق کروموزوم‌های میانی از  $q$  به  $q_0$  بر اساس رتبه‌شان کاهش می‌یابد. اگر  $q_0$  برابر با صفر باشد باعث بروز نیروی بیشتر انتخاب می‌شود. روش رتبه بندی نمایی به صورت زیر قابل تعریف است:

$$p_k = q(1 - q)^{k-1}$$

# Exploration vs. Exploitation

---

فازهای مختلف جستجو اغلب برحسب موارد زیر دسته بندی می شوند:

**Exploration (اکتشاف):** تولید افراد جدید در نواحی آزمایش نشده فضای جستجو

**Exploitation (بهره برداری):** تمرکز جستجو در نزدیکی راه حل های خوب

# Exploration vs. Exploitation

---

در فرآیند جستجوی تکاملی باید بین این دو توازن برقرار شود:

Exploration زیاد ← عدم کارآیی جستجو، پاسخ بهینه

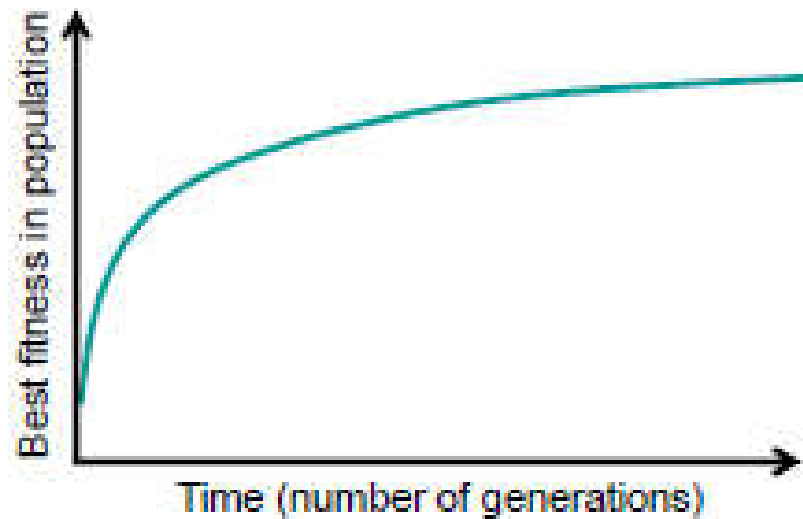
Exploitation زیاد ← افزایش میزان حریصانه بودن جستجو، سریع، پاسخ غیر بهینه

**همگرایی زودرس (Premature Convergence):**

از دست دادن سریع تنوع جمعیت و افتادن در دام بهینه‌های محلی

# نمودار اجرا

## نشانهگر Anytime Behavior

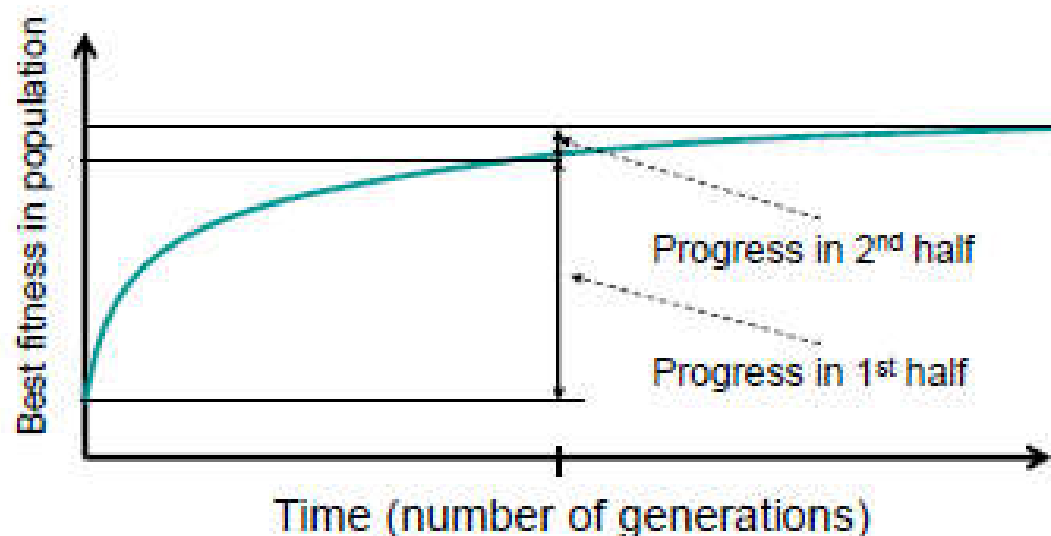


این بدان معنی است که جستجو را می‌توان در هر زمان دلخواه متوقف نمود و یک راه حل به دست آورد (هرچند این راه حل Sub-Optimal باشد)

# آیا اجرای طولانی مفید می باشد؟

پاسخ:

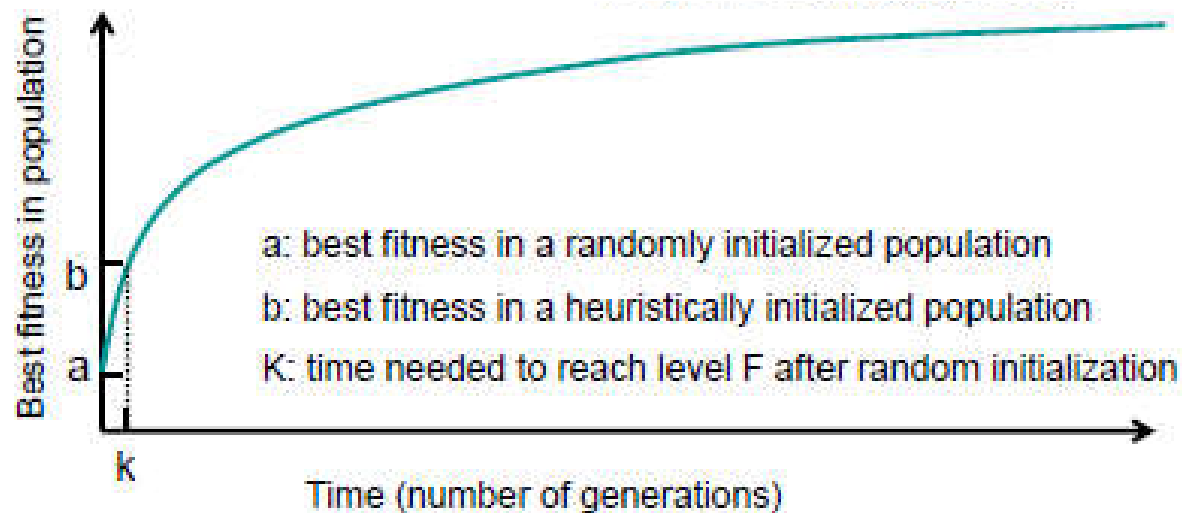
بستگی به میزان برازندگی مورد نیاز دارد ولی در کل بهتر است از چندین اجرا با زمان اجرای کوتاه استفاده شود.



# تولید جمعیت اولیه

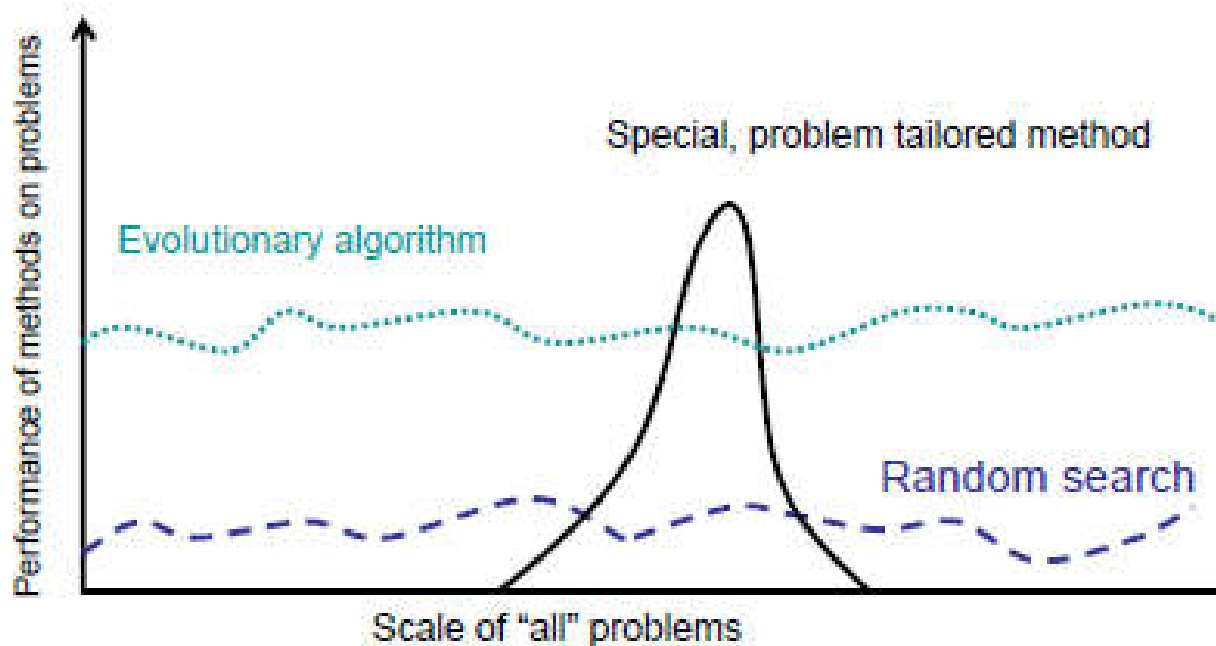
آیا تلاش برای ایجاد جمعیت اولیه به صورت هوشمند ارزش دارد؟

اگر راه حل‌ها یا روش‌های خوبی وجود داشته باشد مناسب است هرچند این روش‌ها عموماً بسیار زمان بر هستند



# الگوریتم‌های تکاملی به عنوان حل کننده مسئله

دیدگاه Goldberg (۱۹۸۹)



# تئوری عدم وجود ناهار مجانی!

---

بر اساس این اصل داریم:

- عملکرد هیچ الگوریتمی روی همه مسائل نمی‌تواند به طور متوسط بهتر از عملکرد جستجوی تصادفی باشد.
- یعنی، نمایش منحنی عملکرد EA همواره در بالای منحنی عملکرد جستجوی تصادفی اساساً نادرست می‌باشد.