فرامکاشفه های تکاملی

فرامکاشفه های تکاملی

- الگوریتم ژنتیک
- برنامه نویسی ژنتیک
- استراتژی تکامل
- تکامل تفاضلی
- الگوریتم های ممتیک
- الگوریتم های فرهنگی
- الگوریتم ژنتیک تاگوچی
- الگوریتم هم تکاملی
- الگوریتم تکاملی دیپلوییدی
- بهینه سازی تولید مثل غیر جنسی
- سیستم ایمنی مصنوعی

الگوریتم ژنتیک

- تکامل ژنی را مدل می کند.
- از ژنوتایپ برای نشان دادن ویژگیهای موجودات استفاده می کند
- عملگرهای اصلی
  - انتخاب (برای مدل کردن قانون بقا اصلح) و
  - بازترکیب و جهش(برای مدل کردن تولید مثل)

الگوریتم ژنتیک

Genetic Algorithm

پیشگامان الگوریتم های ژنتیک

- ۱۹۵۷، Fraser، شبیه سازی الگوریتم های ژنتیک توسط کامپیوتر
- ۱۹۶۲، Bremermann، بهینه سازی با استفاده از تکامل و باز ترکیب
- ۱۹۶۷، Reed، شبیه سازی تکامل بیولوژیکی و یادگیری ماشین
- ۱۹۷۵، Holland، تطبیق در شبکه عصبی مصنوعی با روش های تکاملی

کاربرد های الگوریتم ژنتیک

- جستجو
- بهینه سازی پیوسته وترکیبیاتی
- یادگیری ماشین
- مهندسی کنترل
- طراحی
- زمان بندی کارها
- برنامه ریزی حرکت ربات
- پردازش سیگنال
- بازی

الگوریتم ژنتیک استاندارد هالند

بخش های الگوریتم ژنتیک ارائه شده توسط هالند که نام الگوریتم ژنتیک متعارف یا استاندارد نیز شناخته میشود:

- نمایش کروموزوم ها بصورت رشته بیتی
- روش انتخاب Proportional Selection
- روش برش تک نقطه ای (One-Point Crossover)
- روش جهش یکنواخت (Uniform Mutation)

شبه کد الگوریتم ژنتیک استاندارد

```
Function GA(problem) returns a state that is a local optimum
Input: Populationsize, Problemsize, Pcrossover, Pmutation
Output: Sbest

Population ← InitializePopulation(Populationsize, Problemsize);
EvaluatePopulation(Population);
Sbest ← GetBestSolution(Population);
while ¬ StopCondition() do
  Parents ← SelectParents(Population, Populationsize);
  Children ← ∅;
  foreach Parent1, Parent2 ∈ Parents do
    Child1, Child2 ← Crossover(Parent1, Parent2, Pcrossover);
    Children ← Mutate(Child1, Pmutation);
    Children ← Mutate(Child2, Pmutation);
  end
  EvaluatePopulation(Children);
  Population ← Replace(Population, Children);
  Sbest ← GetBestSolution(Population);
end
return Sbest;
```

الگوریتم ۳-۱ شبه کد الگوریتم ژنتیک استاندارد

عملگر بازترکیب

	نمایش دودویی	نمایش ممیز شناور
غیر جنسی		
جنسی (دو والد)	۱. تک نقطه ای هالند ۲. دونقطه ای رید ۳. یکنواخت ۲۰۰۲ و۱۹۸۷	۱. عملگر خطی رایت ۲. مکاشفه ای هدایت شده رایت ۳. هم برش مخلوط اشلمن و شیفر ۴. هندسی مایکل ویز
چند ترکیبی (چند والد)	۱. رای گیری بین هر ژن بریمن ۲. N نقطه ای چند والدینی بریمن ۳. تپه نوردی جونز	۱. حسابی مایکل ویز ۲. هندسی مایکل ویز ۳. هم برش سادک سوتسوی

روش های برش مختلف در یک دسته بندی کلی به سه دسته تقسیم میشوند:

- برش های تک والدی (asexual): فرزند از یک والد تولید میشود.
- برش های دو والدی (sexual): فرزند از دو والد تولید میشود.
- برش های چند والدی (multi-recombination): فرزند از ترکیب سه یا تعداد بیشتری والد تولید میشود.

تذکر:

✓ والدین با استفاده از یکی از روش های انتخاب، انتخاب میشود.

✓ والدین انتخاب شده با احتمال p_c تولید فرزند می نمایند. (معمولا p_c عدد بزرگی انتخاب میشود)

✓ در طی انتخاب تصادفی ممکن است دو (یا چند) والد یکسان برای عمل برش انتخاب شوند.

✓ در طی برش های مختلف، ممکن است بیش از یک بار، یک جفت والد برای برش انتخاب شوند.

مسایل مهم در بازترکیب

در انتخاب والدین، مسائل زیر باید در نظر گرفته شوند:

- به علت احتمالی بودن انتخاب، امکان دارد که یک موجود به عنوان هر دو والد انتخاب شود. در این صورت فرزندان تولید شده یک کپی از والدین خواهند بود. فرآیند انتخاب والدین باید دارای آزمونی برای جلوگیری از این عملیات غیر ضروری باشد.
- همچنین امکان دارد که یک موجود در بیش از یک عملیات بازترکیب شرکت کند. این حالت زمانی که از مدل ارزیابی انتخاب نسبی استفاده می‌شود، ایجاد مشکل می‌کند (همگرایی سریع جمعیت به یک جواب برتر).
- بعلاوه در عملگر بازترکیب می‌توانیم به گونه‌ای عمل کنیم که تنها در صورتی که فرزندان از والدین خود بهتر هستند آنها را جایگزین والدین‌شان نماییم.

عملگر بازترکیب

	نمایش دودویی	نمایش ممیز شناور
غیر جنسی		
جنسی	۱. **تک نقطه ای** **هالند** ۲. **دونقطه ای** **رید** ۳. **یکنواخت** **۲۰۰۲ و۱۹۸۷**	۱. عملگر خطی رایت ۲. مکاشفه ای هدایت شده رایت ۳. هم برش مخلوط اشلمن و شیفر ۴. هندسی مایکل ویز
چند ترکیبی	۱. رای گیری بین هر ژن بریمن ۲. N نقطه ای چند والدینی بریمن ۳. تپه نوردی جونز	۱. حسابی مایکل ویز ۲. هندسی مایکل ویز ۳. هم برش سادک سوتسوی

بازترکیب جنسی در نمایش دودویی

- تک نقطه ای
- دو نقطه ای
- یکنواخت

عملگر بازترکیب

	نمایش دودویی	نمایش ممیز شناور
غیر جنسی		
جنسی	۱. تک نقطه ای هالند ۲. دونقطه ای رید ۳. یکنواخت ۲۰۰۲ و۱۹۸۷	۱. عملگر خطی رایت ۲. مکاشفه ای هدایت شده رایت ۳. هم برش مخلوط اشلمن و شیفر ۴. هندسی مایکل ویز
چند ترکیبی	**۱. رای گیری بین هر ژن بریمن** **۲. N نقطه ای چند والدینی بریمن** **۳. تپه نوردی جونز**	۱. حسابی مایکل ویز ۲. هندسی مایکل ویز ۳. هم برش سادک سوتسوی

بازترکیب چند والدینی در نمایش دودویی

۱. رای گیری بین هر ژن بریمن

در این روش تعداد n_μ والد به نام های $x_1(t), x_2(t), \ldots, x_{n_\mu}(t)$ وجود دارند.
هر ژن از کروموزوم تولید شده (فرزند) با رابطه زیر محاسبه میشود:

$$\tilde{x}_{ij}(t) = \begin{cases} 0 & \text{if } n'_\mu \geq n_\mu / 2 \ , \ l = 1, \ldots, n_\mu \\ 1 & \text{otherwise} \end{cases}$$

که در رابطه فوق n'_μ تعداد والدینی است که در آن $x_{lj}(t) = 0$

۲. N نقطه ای چند والدینی بریمن

- به تعداد والدین منهای یک نقطه بازترکیب انتخاب می شود.
- یک فرزند توسط انتخاب یک بخش از هر والد تولید میشود.

بازترکیب چند والدینی در نمایش دودویی

۳. تپه نوردی جونز

- **با دو والد شروع میشود**
- **فرزندان گوناگونی با ترکیب این دو والد ایجاد میشود.**
- **تا جفت فرزندی که یکی از آنها دارای برازندگی بیش از بهترین والد باشد.**
- **سپس با این دو فرزند جدید به عنوان والدین جدید به ادامه جستجو می پردازد.**
- **اگر والدین بهتر پیدا نشدند بدترین والد با یک والد تصادفی جایگزین میشود.**

عملگر بازترکیب

	نمایش دودویی	نمایش ممیز شناور
غیر جنسی		
جنسی	۱. تک نقطه ای هالند ۲. دونقطه ای رید ۳. یکنواخت ۲۰۰۲ و۱۹۸۷	**۱. عملگر خطی رایت** **۲. هم برش مخلوط اشلمن و شیفر** **۳. هندسی مایکل ویز**
چند ترکیبی	۱. رای گیری بین هر ژن بریمن ۲. N نقطه ای چند والدینی بریمن ۳. تپه نوردی جونز	۱. حسابی مایکل ویز ۲. هندسی مایکل ویز ۳. هم برش سادک سوتسوی

بازترکیب جنسی در نمایش ممیز شناور

۱. عملگر خطی رایت (تعمیم در صفحه بعد)

- از والدین X1 , X2 سه فرزند ایجاد می شود.
  - X1 + X2
  - 1.5X1 - 0.5X2
  - -0.5X1+1.5X2
- دو تا از بهترین راه حلها به عنوان فرزندان انتخاب می شوند.

- **روش ابتکاری Wright:**

با استفاده از رابطه زیر از دو والد، یک فرزند تولید میشود:

$$\tilde{x}_{ij}(t) = U(0,1)\left(x_{2j}(t) - x_{1j}(t)\right) + x_{2j}(t)$$

در رابطه فوق فرض شده است که والد x_2 به نسبت والد x_1 والد بهتری است

بازترکیب جنسی در نمایش ممیز شناور

۲. بازترکیب به هم برش مخلوط اشلمن و شیفر

بازترکیب جنسی در نمایش ممیز شناور

۳. بازترکیب هندسی دو والده مایکل ویز

صورتی که این فاصله بزرگ باشد، پس فاصله بین فرزندان و والدینشان بزرگ خواهد بود.

مایکل‌ویز [۸۱] بازترکیب هندسی دو والده را برای تولید یک فرزند به صورت زیر پیشنهاد نموده است:

$$\tilde{x}_{ij}(t) = (x_{1j}x_{2j})^{0.5} \qquad (۷-۳)$$

بازترکیب هندسی را می‌توان به تولیدمثل چند والده مانند زیر تعمیم داد:

$$\tilde{x}_{ij}(t) = (x_{1j}^{\alpha_1} x_{2j}^{\alpha_2} \dots x_{n_\mu j}^{\alpha_{n_\mu}}) \qquad (۸-۳)$$

به صورتی که n_μ تعداد والدین می‌باشد و $\sum_{i=1}^{n_\mu} \alpha_i = 1$

عملگر بازترکیب

	نمایش دودویی	نمایش ممیز شناور
غیر جنسی		
جنسی	۱. تک نقطه ای هالند ۲. دونقطه ای رید ۳. یکنواخت ۲۰۰۲ و۱۹۸۷	۱. عملگر خطی رایت ۲. هم برش مخلوط اشلمن و شیفر ۳. هندسی مایکل ویز
چند ترکیبی	۱. رای گیری بین هر ژن بریمن ۲. N نقطه ای چند والدینی بریمن ۳. تپه نوردی جونز	**۱. حسابی مایکل ویز** **۲. هندسی مایکل ویز** **۳. هم برش سادک سوتسوی**

بازترکیب چند ترکیبی در نمایش ممیز شناور

۱. بازترکیب حسابی

روش حسابی Michalewicz:

هر فرزند با استفاده از رابطه زیر از دو یا چند والد تولید میشود:

$$\tilde{x}_{ij}(t) = \sum_{l=1}^{n_\mu} \gamma_l x_{lj(t)}$$

در رابطه فوق باید $\sum_{l=1}^{n_\mu} \gamma_l = 1$

مثال: دررابطه فوق اگر $n_\mu = 2$ آنگاه $\tilde{x}_{ij}(t) = (1-\gamma)x_{1j(t)} + \gamma x_{2j(t)}$

بازترکیب چند ترکیبی در نمایش ممیز شناور

۳. بازترکیب هندسی چند والده مایکل ویز

صورتی که این فاصله بزرگ باشد، پس فاصله بین فرزندان و والدینشان بزرگ خواهد بود.

مایکل‌ویز [۸۱] بازترکیب هندسی دو والده را برای تولید یک فرزند به صورت زیر پیشنهاد نموده است:

$$\tilde{x}_{ij}(t) = (x_{1j}x_{2j})^{0.5} \qquad (۷-۳)$$

بازترکیب هندسی را می‌توان به تولیدمثل چند والده مانند زیر تعمیم داد:

$$\tilde{x}_{ij}(t) = (x_{1j}^{\alpha_1} x_{2j}^{\alpha_2} \dots x_{n_\mu j}^{\alpha_{n_\mu}}) \qquad (۸-۳)$$

به صورتی که n_μ تعداد والدین می‌باشد و $\sum_{i=1}^{n_\mu} \alpha_i = 1$

بازترکیب چندترکیبی در نمایش ممیز شناور

- هم برش سادک سوتسوی گلدبرگ رندرز و بریسنی

سوتسوی[1]، گلدبرگ [۱۲۷] و رندرز[2]، برسینی[3] [۱۰۳] عملگر هم‌برش سادک[4] را به عنوان یکی دیگر از روش‌های هم‌برش برای نمایش ممیز شناور (اعداد حقیقی) معرفی کردند. آنها تعداد والدین را بزرگتر از ۲ در نظر گرفتند. در این روش بازترکیب، ابتدا بهترین و بدترین والدها تعیین شده و به ترتیب $x_1(t)$ و $x_2(t)$ نامیده می‌شوند. سپس مرکز والدین انتخاب شده، بدون در نظر گرفتن $x_2(t)$ محاسبه شده و با $\bar{x}(t)$ نام‌گذاری می‌شود. در نهایت یک فرزند با استفاده از رابطه زیر تولید می‌شود:

$$\tilde{x}(t) = \bar{x}(t) + (x_1(t) - x_2(t)) \qquad (۹-۳)$$

عملگر جهش

- جهش یکنواخت تصادفی
- جهش INORDER
- جهش بزرگ

- هدف عمل جهش یا mutation، تولید یک کروموزوم جدید با تغییر در کروموزوم های موجود، با هدف افزودن گوناگونی (*diversity*) به سیستم میباشد.
- استفاده از جهش در کنار برش باید این اطمینان را بوجود می آورد که برای تمامی فضای مسئله، امکان (شانس) در دست رس بودن وجود دارد.
- برای هر ژن از فرزند انتخاب شده برای جهش، عمل جهش با احتمال p_m انجام خواهد شد.

احتمال اینکه در یک کروموزوم عمل جهش انجام شود برابر است با

$$\mathrm{Prob}\left(\tilde{x}_i(i)\text{is mutated}\right) = 1-\left(1-p_m\right)^{n_x}$$

نتیجه عمل جهش بر روی کروموزم $\tilde{x}_i(t)$ را با $x_i'(t)$ نشان می دهند.

در نمایش رشته بیتی، عمل جهش را می توان به دو دسته تقسیم نمود:

- **جهش یکنواخت (Uniform Mutation):**

در این روش که نام جهش تصادفی نیز شناخته میشود، مقدار هر ژن با احتمالی معکوس (not) میشود.

- **جهش میانی (Inorder Mutation):**

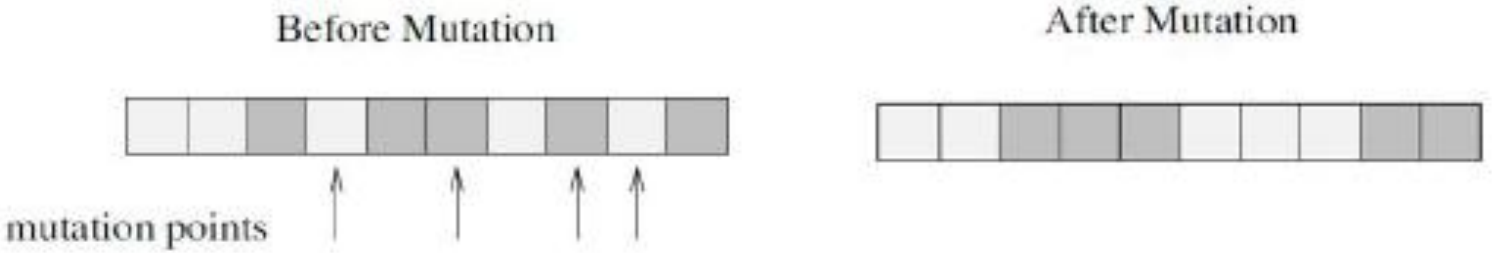
در این روش دو ژن انتخاب و فقط ژن های مابین آن دو مورد جهش قرار خواند گرفت. روش جهش استفاده شده در این روش همان جهش یکنواخت است

Uniform Mutation

جهش یکنواخت

Algorithm Uniform/Random Mutation

for $j = 1, \ldots, n_x$ **do**
  if $U(0,1) \leq p_m$ **then**
    $x'_{ij}(t) = \neg \tilde{x}_{ij}(t)$, where $\neg$ denotes the boolean NOT operator;
  end
end

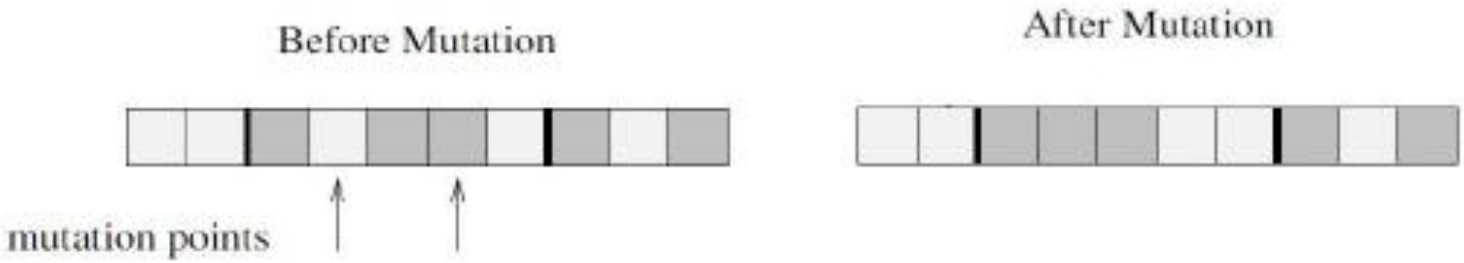


Inorder Mutation

جهش میانی

Algorithm Inorder Mutation

Select mutation points, $\xi_1, \xi_2 \sim U(1, \ldots, n_x)$:
for $j = \xi_1, \ldots, \xi_2$ **do**
  if $U(0,1) \leq p_m$ **then**
    $x'_{ij}(t) = \neg \tilde{x}_{ij}(t)$;
  end
end



جهش بزرگ
HEADLESS CHICKEN

- توسط جونز
- بازترکیب مجدد یک والد با یک پاسخ تصادفی

Floating-Point Mutation...

- روش Uniform Floating-Point Mutation:

در این روش، پس از آنکه ژن برای عمل جهش انتخاب شد، عددی باینری و تصادفی به نام k تولید میشود، سپس:

$$x'_{ij}(t) = \begin{cases} \tilde{x}_{ij}(t) + \Delta\left(t, x_{max,j} - \tilde{x}_{ij}(t)\right) & \text{if } k = 0 \\ \tilde{x}_{ij}(t) - \Delta\left(t, \tilde{x}_{ij}(t) - x_{min,j}\right) & \text{if } k = 1 \end{cases}$$

در رابطه فوق $\Delta(t,x)$ یک عدد تصادفی در بازهٔ $[0,\ x]$ میباشد.

کنترل پارامترهای P_m و P_c

- در اغلب از پیاده سازی ها مقدار P_m یک مقدار ثابت کوچک در نظر گرفته میشود.
- در اغلب از پیاده سازی ها مقدار P_c یک مقدار ثابت بزرگ در نظر گرفته میشود.

میتوان مقادیر P_m و P_c را بصورت **متغیر** نیز تعریف نمود.

یک نمونه از p_m که بر حسب زمان متغیر است

$$p_m(t) = \frac{1}{240} + \frac{0.11375}{2^t}$$

یک نمونه از p_m که بر حسب شماره بیت متغیر است

$$p_m(j) = \frac{0.3528}{2^{j-1}} \qquad j = 1, \ldots, n_b \text{ , } n_b \text{ is least significant bit}$$

یک نمونه از p_m که بر حسب زمان و شماره بیت متغیر است

$$p_m(j,t) = \frac{0.4026}{2^{t+j-1}}$$

انواع الگوریتم ژنتیک
براساس استراتژی جایگزینی

- **روش شکاف نسلی Generation Gap Method**

شکاف نسلی: میزان همپوشانی بین نسل کنونی و نسل بعدی

- الگوریتم ژنتیک نسلی Generatioal Genetic Algorithm GGA (شکاف نسلی صفر)همه والدین با فرزندان جایگزین می شوند.همپوشانی بین نسل جدیدو نسل کنونی وجود ندارد.
- الگوریتم ژنتیک پایا Steady State Genetic Algorithm SSGA (شکاف نسلی زیاد) بلافاصله بعد از تولید فرزند و جهش آن تصمیم میگیرد که میان فرزند و والد کدام را حفظ کند. همپوشانی بین جمعیت کنونی و بعدی وجود دارد.

- **الگوریتم ژنتیک آشفته Messy Genetic Algorithm**

استراتژی های جایگزینی برای SSGA

برخی از روشهای جایگزینی در روش SSGA

- Replace Worst: فرزند جانشین بدترین کروموزم خواهد شد
- Replace Random: فرزند تصادفی جانشین یکی از کروموزم ها خواهد شد
- Kill Tournament: یک دسته کروموزم تصادفی انتخاب و ...
- Replace Oldest: فرزند جانشین قدیمیترین کروموزم خواهد شد
- Conservative Selection: ...
- Elitist: ...
- Parent-Offspring: ...

الگوریتم ژنتیک آشفته
Messy Genetic Algorithm

- در ژنتیک استاندارد ممکن است موجودات برازنده ای تولید شوند که برخی از ژنهای آنها بهینه نباشد.
- یافتن مقادیر مناسب برای ژن ها از طریق عملیات بازترکیب و جهش بر روی کل موجود کار دشواری است.
- در ژنتیک آشفته راه حلها از طریق تکامل بلوکهای سازنده و ترکیب آنها یافته میشوند.
- موجودات دارای اندازه های گوناگون هستند. و به صورت جفت مقدار-موقعیت مشخص میشوند.
- این جفت را ژن آشفته میگویند.
- 0*10 (1,0)(3,1)(4,0)(1,1)
- ممکن است یک ژن دو بار یا اصلا مقدار نداشته باشد.

الگوریتم ژنتیک آشفته
Messy Genetic Algorithm

- برای ژنهای تعیین نشده از یک الگوی رقابتی استفاده می شود. مثلا 1011 که ژن قبل را به 0110 تبدیل می کند.
- **هدف** :
  - تکامل بلوک های سازنده
  - سپس ترکیب بلوک های سازنده بهینه برای شکل دادن یک راه حل بهینه
- **مراحل**:
  - مقداردهی اولیه: ایجاد بلوک های سازنده
  - مرحله اصلی : تولید بلوک های سازنده
  - الحاق: ترکیب بلوکهای سازنده

انواع الگوریتم های ژنتیگ – mGA و IEGA

(mGA) Messy Genetic Algorithm

گونه ای از پیاده سازی الگوریتم های ژنتیک که در آن تعداد ابعاد هر کروموزوم (تعداد ژنهای آن) میتواند متفاوت از دیگر کروموزوم ها باشد.

یکی از موارد استفاده mGA یا Messy Genetic Algorithm در پیاده سازی مسائل با ابعاد بزرگ است. زمانی که بخواهیم تعدادی از ابعاد مسئله را چشم پوشی نماییم.

Interactive Evolution Genetic Algorithm (IEGA)

گونه ای الگوریتم ژنتیک که در آن عمل ارزش دهی به کروموزوم ها و یا انتخاب کروموزم ها برای بقا در نسل بعد، توسط انسان انجام میشود

انواع الگوریتم های ژنتیگ – mGA و IEGA

(mGA) Messy Genetic Algorithm

گونه ای از پیاده سازی الگوریتم های ژنتیک که در آن تعداد ابعاد هر کروموزوم (تعداد ژنهای آن) میتواند متفاوت از دیگر کروموزوم ها باشد.

یکی از موارد استفاده mGA یا Messy Genetic Algorithm در پیاده سازی مسائل با ابعاد بزرگ است. زمانی که بخواهیم تعدادی از ابعاد مسئله را چشم پوشی نماییم.

Interactive Evolution Genetic Algorithm (IEGA)

گونه ای الگوریتم ژنتیک که در آن عمل ارزش دهی به کروموزوم ها وبا انتخاب کروموزم ها برای بقا در نسل بعد، توسط انسان انجام میشود

انواع الگوریتم های ژنتیگ – Island GA

Island GA

تولید موازی چند جمعیت (بهمراه تولید فرزندان، انتخاب نسل بعد و ...)
که کروموزوم های یک جمعیت میتوانند از جمعیت خود جدا و به جمعیت دیگری بپیوندند (migration).

هدف: پیاده سازی بصورت موازی بر روی چند پردازنده با تخصیص هر جمعیت به یک پردازنده مجزا، برای افزایش توام سرعت و کارایی.

سیاستهای مهاجرت (migration) در **Island GA**:

- پیروی از توپولوژی: ...
- نرخ یا میزان مهاجرت: ...
- مکانیزم انتخاب مهاجر: ...
- استراتژی انتخاب جایگزین: ...

مفهوم Niching

دسته بندی کلی الگوریتم های بهینه یابی:

- Unimodal: هدف یافتن یک هدف در فضای مسئله
- Multimodal: هدف یافتن تعدادی هدف در فضای مسئله

مبانی niching:

- در طبیعت بقای جانوران به رقابت در شکار، یافتن غذا، تعیین محل زندگی و ... وابسته است، که باعث میشود بعضی جانوران برای بقا به هم نیاز داشته باشند.
- هر niche در طبیعت، منطقه ای از زندگی جانوران است که در نوعی زندگی (متفاوت از niche های دیگر) در جریان است.
- یک گونه، به دسته ای از individualها گفته میشود که صفات مشترکی دارند و توانایی تولید مثل با گونه های دیگر را ندارند و در یک niche زندگی می کنند.
- در هر niche منابع زندگی محدود است و جانوران آن niche باید منابع را با هم تقسیم نمایند.

NichingGA – Fitness Sharing

Fitness Sharing

یکی از روش های پیاده سازی niching در GA است.

در این روش مقدار fitness با توجه به مقدار fitness اولیه موجود (تابع fitness) از رابطه زیر محاسبه میگردد:

$$f_s(\mathbf{x}_i(t)) = \frac{f(\mathbf{x}_i(t))}{\sum_j sh(d_{ij})}$$

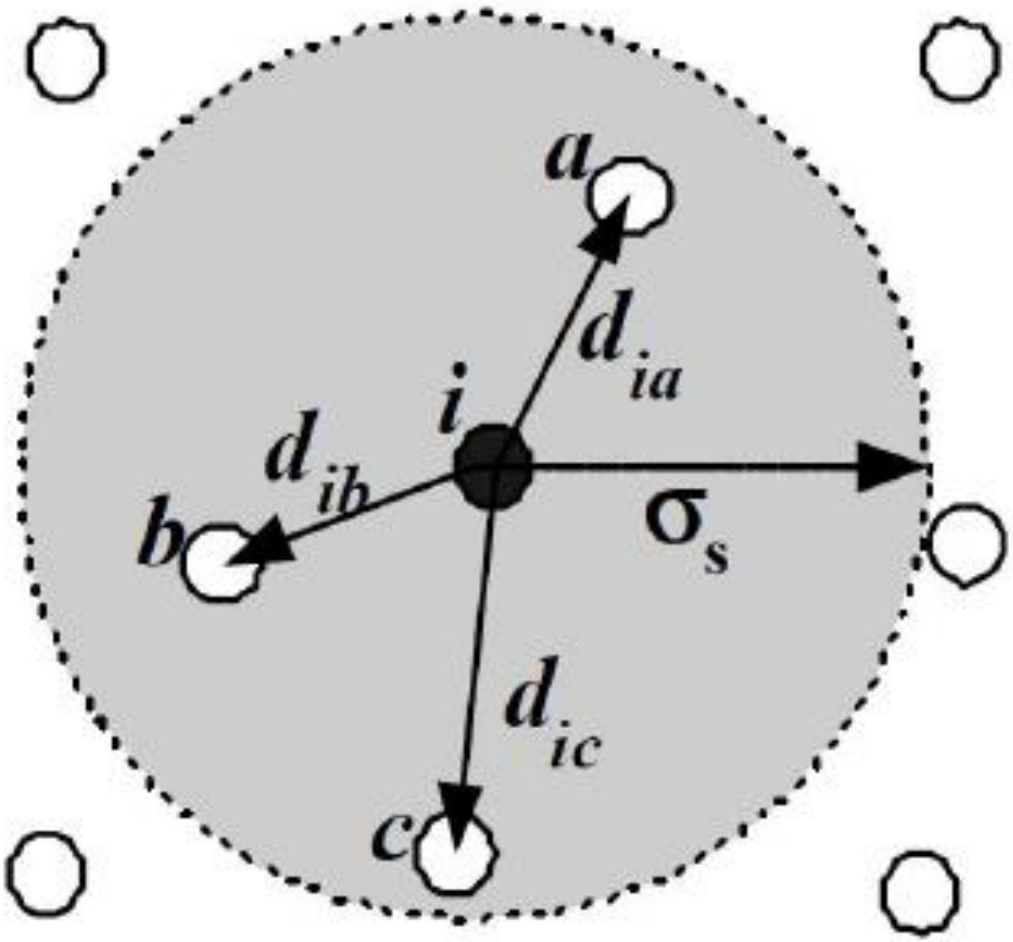
where $\sum_j sh(d_{ij})$ is an estimate of how crowded a niche is.
A common sharing function is the triangular sharing function,

$$sh(d) = \begin{cases} 1 - (d/\sigma_{share})^{\alpha} & \text{if } d < \sigma_{share} \\ 0 & \text{otherwise.} \end{cases}$$

The symbol d_{ij} represents the distance between individuals $\mathbf{x}_i$ and $\mathbf{x}_j$

NichingGA – Fitness Sharing

Fitness Sharing



۲ـ برنامه نویسی ژنتیک

برنامه نویسی ژنتیک

۳-۳-۱ شبه‌کد برنامه‌نویسی ژنتیک

همانطور که گفته شد، برنامه‌نویسی ژنتیک توسط کزا پیشنهاد شده است (الگوریتم ۳-۲). خصوصیات مهم برنامه‌نویسی ژنتیک عبارتند از:

- استفاده از نمایش درختی.
- طول متغیر برای کروموزوم‌ها (به عنوان تنها الگوریتم تکاملی که دارای این ویژگی است).
- بکارگیری یک جمعیت با تعداد اعضاء ثابت.
- عملگر بازترکیب در برنامه‌نویسی ژنتیک عملگر اصلی است.
- عملگر جهش در برنامه‌نویسی ژنتیک عملگر فرعی است.
- تنظیم احتمال بازترکیب P_c به مقادیر بزرگ (بزرگ‌تر از 0.9).
- در عملگر بازترکیب احتمال برش شاخه‌های غیرپایانی، زیاد (بزرگ‌تر از 0.9) است.
- در عملگر بازترکیب احتمال برش شاخه‌های پایانی، کم (کوچک‌تر از 0.01) است.
- تنظیم احتمال جهش P_m به مقادیر کوچک (کوچک‌تر از 0.01).
- عملگرهای اصلاح معماری[۲]، تنها منوط به تکثیر و یا حذف زیردرخت‌ها نمی‌باشند.

```
Function GP(problem) returns a state that is a local optimum
Input: Population_size, nodes_func, nodes_term, P_crossover, P_mutation, P_alteration
Output: S_best

Population ← InitializePopulation(Population_size, nodes_func, nodes_term);
EvaluatePopulation(Population);
S_best ← GetBestSolution(Population);
while ¬ StopCondition() do
  Children ← ∅;
  while Size(Children) < Population_size do
    Operator ← SelectGeneticOperator(P_crossover, P_mutation, P_alteration);
    if Operator ≡ CrossoverOperator then
      Parent1, Parent2 ← SelectParents(Population, Population_size);
      Child1, Child2 ← Crossover(Parent1, Parent2);
      Children ← Child1;
      Children ← Child2;
    else if Operator ≡ MutationOperator then
      Parent1 ← SelectParents(Population, Population_size);
      Child1 ← Mutate(Parent1);
      Children ← Child1;
    else if Operator ≡ AlterationOperator then
      Parent1 ← SelectParents(Population, Population_size);
      Child1 ← AlterArchitecture(Parent1);
      Children ← Child1;
    end
  end
  EvaluatePopulation(Children);
  Population ← Replace(Population, Children);
  S_best ← GetBestSolution(Population);
end
return S_best;
```

الگوریتم ۳-۲ شبه کد برنامه‌نویسی ژنتیک

برنامه نویسی ژنتیک

- **عملگرهای بازترکیب**

در برنامه‌نویسی ژنتیک، دو روش برای تولید فرزندان وجود دارد، که هر کدام در تعداد فرزندان تولید شده با یکدیگر متفاوت هستند.

- **تولید یک فرزند:** یک گره تصادفی در هر کدام از والدین انتخاب می‌شود. سپس عملگر بازترکیب از طریق جایگزینی زیر درخت متناظر در یک والد توسط والد دیگر انجام می‌پذیرد.
- **تولید دو فرزند:** یک گره تصادفی در هر کدام از والدین انتخاب می‌شود. در این مورد زیر درخت‌های متناظر برای ایجاد دو فرزند با یکدیگر تعویض می‌شوند.

برنامه نویسی ژنتیک

- **عملگرهای جهش**

عملگرهای جهش برای کاربردهای خاص توسعه یافته‌اند. در هر حال، بسیاری از عملگرهای جهش در برنامه‌نویسی ژنتیک، در حالت کلی نیز کاربرد دارند. عملگرهای جهش زیر را می‌توانیم برای برنامه‌نویسی ژنتیک بکار ببریم:

- **جهش گره تابع:** یک گره غیر پایانی یا گره تابع، به صورت تصادفی انتخاب می‌شود و با عنصری از مجموعه تابع که به صورت تصادفی انتخاب شده است جایگزین می‌شود.
- **جهش گره پایانی:** یک گره برگ یا گره پایانی، به صورت تصادفی انتخاب می‌شود و با یک گره پایانی جدید که به صورت تصادفی از مجموعه پایانی انتخاب شده است، جایگزین می‌شود.
- **جهش تعویضی:** یک گره تابع به شکل تصادفی انتخاب شده و پارامترهای آن تعویض می‌شوند.
- **جهش رشدی:** یک گره به صورت تصادفی انتخاب می‌شود و با یک زیر درخت تصادفی انتخاب شده جایگزین می‌شود. زیر درخت جدید دارای عمق محدود و از قبل مشخص شده است.
- **جهش گوسین:** یک گره پایانی که دارای مقداری ثابت است به صورت تصادفی انتخاب شده و با اضافه کردن یک مقدار تصادفی گوسین به آن مقدار ثابت جهش می‌یابد.
- **جهش برشی**[1]: یک گره تابع به صورت تصادفی انتخاب می‌شود و توسط یک گره پایانی تصادفی جایگزین می‌شود. این عملگر جهش، درخت را هرس می‌کند.