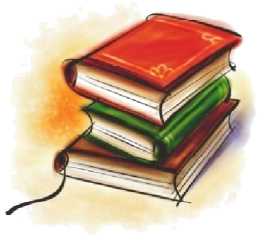


هوش مصنوعی پیشرفته

پردازش تکاملی

دانشگاه تهران - دانشکده ریاضی، آمار و علوم کامپیوتر

نیمسال اول 1397-1398



فهرست

○ الگوریتم‌های جستجو: معرفی و مفاهیم

○ پردازش تکاملی

- مفاهیم

- نمایش کروموزوم

- جمعیت اولیه

- تابع برازش

- انتخاب (Selection)

- تولید مثل: باز ترکیب (Recombination)/هم‌برش (Crossover)، جهش (Mutation)

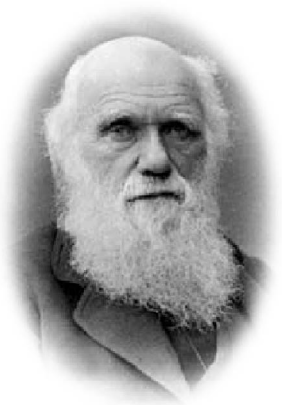
- جایگزینی

- شرایط توقف

- کنترل قابلیت پویش و انتفاع

○ الگوریتم ژنتیک: مثال

محاسبات زیستی



○ محاسبات زیستی (Evolutionary Computing)

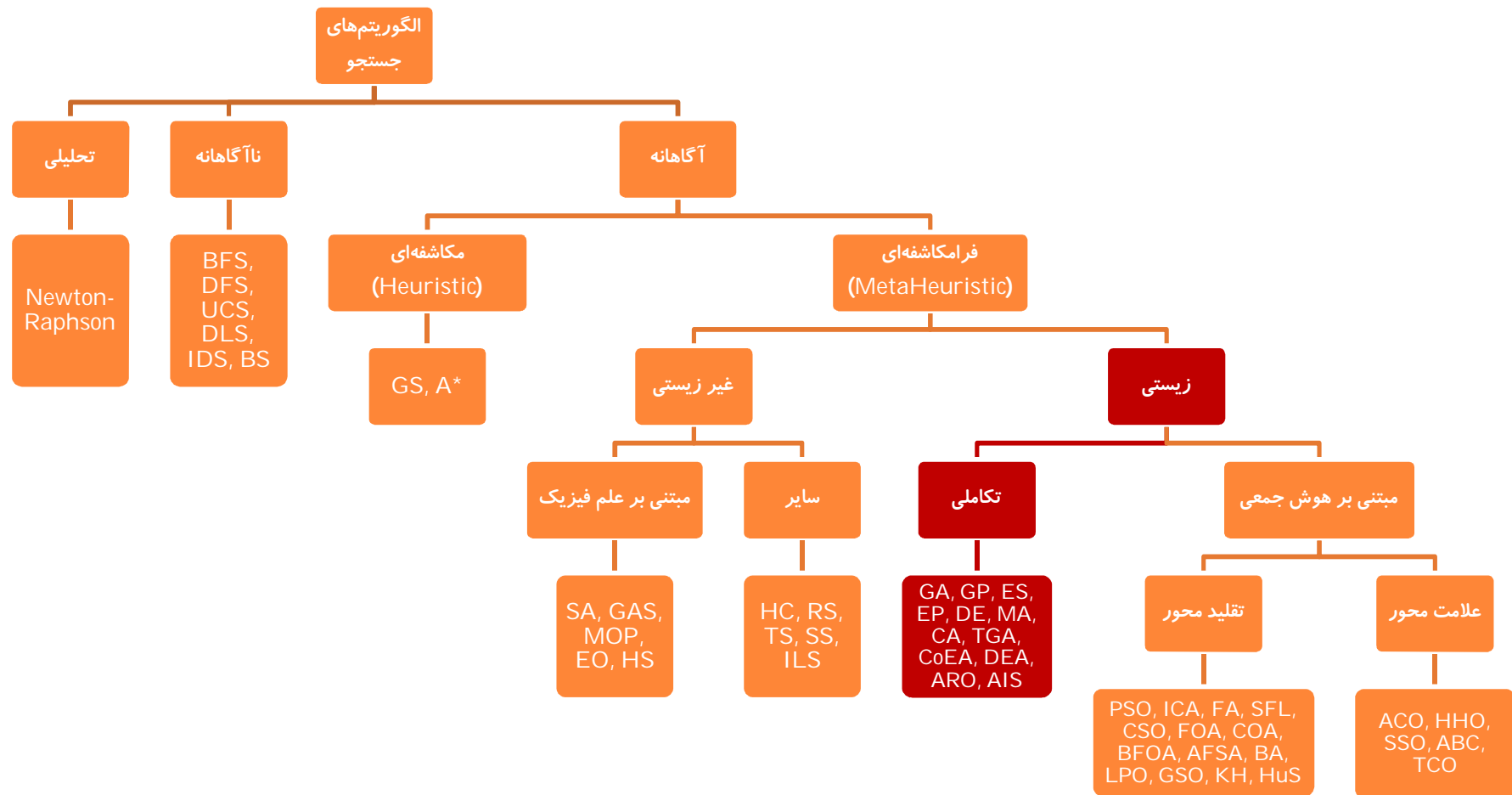
- حل مسائل بهینه‌سازی، جستجو و یادگیری ماشین با الهام از تکامل زیستی

- نظریه تکامل زیستی داروین (1859)

- حیوانات و گیاهان امروزی از نسل موجودات ماقبل تاریخ هستند -صدها میلیون سال از حیات می‌گذرد
- حیات تنها با یک یا تعدادی ارگانیسم ساده شروع شده و بعدها تکامل یافته و تبدیل به میلیون‌ها گونه متفاوت امروزی شده است
- تمامی فرآیند خلقت گونه‌های مختلف حیات، ناشی از یکی از نیروهای هدایت‌کننده در طبیعت با نام انتخاب طبیعی (Natural Selection) است

- از بین رفتن نمونه‌های ضعیف و زنده ماندن نمونه‌های برتر = تکامل تدریجی
- انتخاب طبیعی راز بقای برترین‌ها در طبیعت و انتقال خصوصیات برتر به نسل بعد = قانون بقای اصلح (Survival of the Fittest)

الگوریتم‌های جستجو ...



الگوریتم‌های جستجو ...

○ جستجوی تحلیلی (Analytical Search)

- استفاده از توابع و روش‌های ریاضی برای یافتن حل بهینه: گرادیان، مشتق دوم
 - نیوتون-رافسون (برای یافتن ریشه تابع)

○ جستجوی ناآگاهانه (Uninformed (Blind) Search)

- عدم وجود از اطلاعات جانبی درباره نقاط فضای جستجو: تنها تشخیص هدف از غیرهدف
- پیمایش فضای جستجو به صورت درختی
- دو نوع
 - کامل: پیمایش کامل فضای جستجو و تضمین یافتن یک راه حل (بهینه یا غیربهینه)
 - ناکامل: جستجو تا یافتن یک راه حل
- روش‌ها
 - جستجوی اول سطح (Breadth-First Search)
 - جستجوی هزینه یکنواخت (Uniform-Cost Search)
 - جستجوی اول عمق (Depth-First Search)

الگوریتم‌های جستجو

○ جستجوی آگاهانه (Informed Search)

- استفاده از یک تابع تخمین در مورد فضای جستجو (دانش مساله)

• جستجوی اول-بهترین (Best-First Search) - مکاشفه‌ای

- جستجوی Tree-Search (Graph-Search) - انتخاب یک گره بر اساس تخمین هزینه گسترش آن
- روش‌ها

○ جستجوی حریصانه (Greedy Search): جستجوی نزدیک‌ترین گره به هدف

○ جستجوی A^* (A^* Search): کمینه کردن کل هزینه برآورد شده - محاسبه زیاد و نیاز به حافظه زیاد

• جستجوی فرامکاشفه‌ای (Meta-Heuristic Search)

- روش‌های قدیمی: نیاز به نمایش فضای جستجو با درخت
- عدم امکان نمایش فضای جستجو با درخت (به ویژه برای فضاها بزرگ و نامنظم)
- الهام از پدیده‌های طبیعی برای جستجو (زیستی و غیرزیستی)

جستجو: مفاهیم ...

○ فضای جستجو و دورنمای برآزش (Fitness Landscape) ...

- فضای جستجو: یک سرزمین حاوی کلیه مقادیر ممکن برای پاسخ مساله

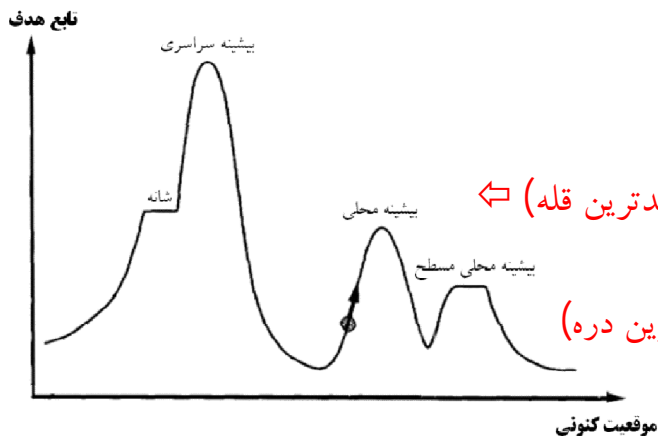
- گاهی بسیار بزرگ و نامنظم
- جستجوی هوشمندانه: پیمایش بخش مهم فضای پاسخ

• دورنمای برآزش: پستی و بلندی‌های سرزمین جستجو

- تابع برآزش (Fitness Function): تعیین پستی و بلندی‌های سرزمین جستجو

- تعریف بر اساس اطلاعات مساله

- در هر لحظه شامل نقطه پاسخ فعلی (موقعیت) و مقدار برآزش (شایستگی) آن نقطه (ارتفاع)



- ارتفاع متناظر با شایستگی: نقطه بهینه = بیشینه مقدار (بلندترین قله) ⇐

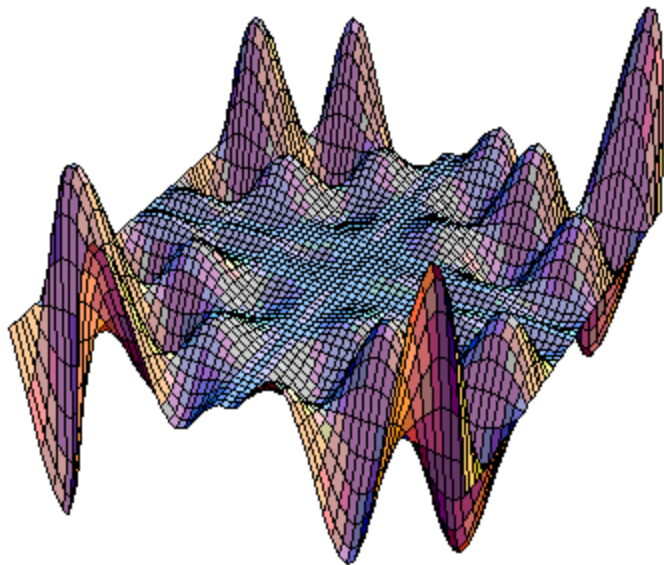
- ارتفاع متناظر با هزینه: نقطه بهینه = کمینه مقدار (عمیق‌ترین دره)

جستجو: مفاهیم ...

○ فضای جستجو و دورنمای برآزش (Fitness Landscape)

- در مسائل واقعی، دورنمای برآزش پیچیده است

- فضای چندقله‌ای یا خارپشتی (multimodal)



Fitness

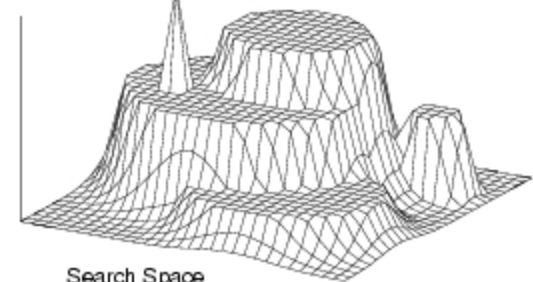
Search Space

دماغه
(Ridge)

Fitness

Search Space

فلات
(Plateau)



جستجو: مفاهیم ...

○ قابلیت پویش (Exploration)

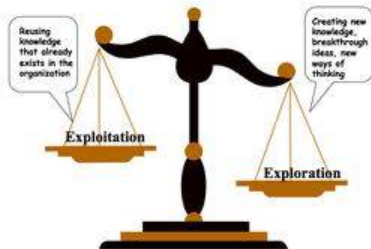
- جستجوی آزادانه کل فضا بدون توجه به دستاوردهای آن در طول جستجو
- رفتار تصادفی تر الگوریتم

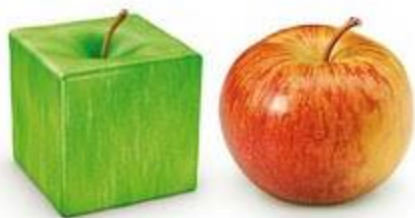
○ قابلیت انتفاع (Exploitation)

- توجه به دستاوردهای الگوریتم در طول جستجو
- رفتار حساب شده و محتاطانه

○ نیاز به تنظیم دو قابلیت بر اساس شرایط مساله

- ایجاد مصالحه (trade-off) بین این دو قابلیت با پارامترهای روش جستجو

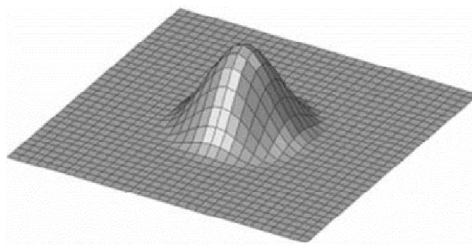




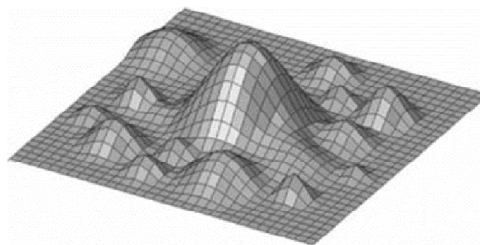
جستجو: مفاهیم ...

○ قابلیت پوشش و ارتفاع

- برای مسائلی با دورنمای برآزش منظم، باید ارتفاع را تقویت و پوشش را کم کنیم



- برای مسائلی با دورنمای برآزش نامنظم (خارپشتی)، باید قابلیت پوشش را زیاد کنیم



- جستجوی با بیشترین پوشش = جستجوی تصادفی (Random Search)
- جستجوی با بیشترین ارتفاع = جستجوی تپه نوردی (Hill-Climbing Search)

پردازش تکاملی ...

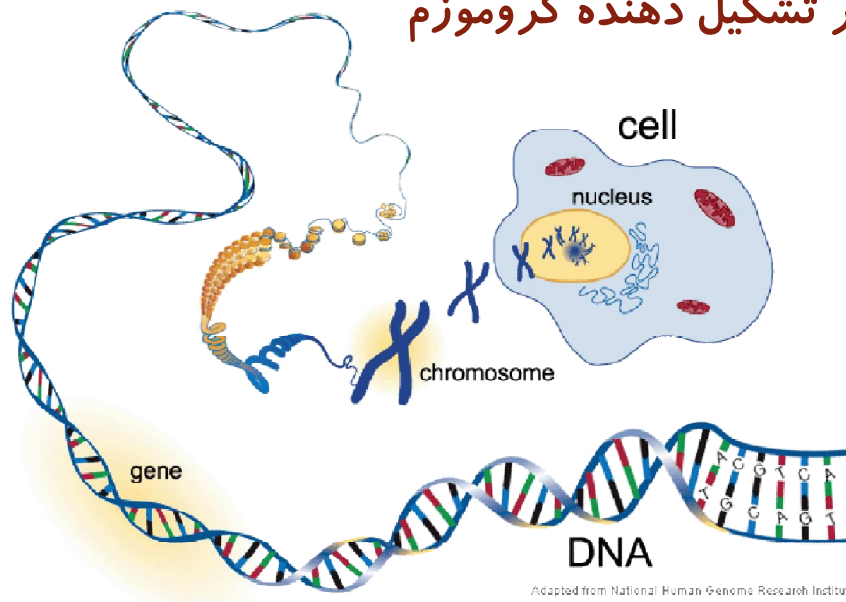
○ مفاهیم ...

- کروموزوم (Chromosome) – محل ذخیره سازی اطلاعات ژنی یک موجود

- تشکیل شده از DNA
- انسان تقریباً 22 هزار ژن در DNA خود دارد
- رشته، گراف، درخت = پاسخ مساله

- ژن (Gene) – واحدهای کوچک تر تشکیل دهنده کروموزوم

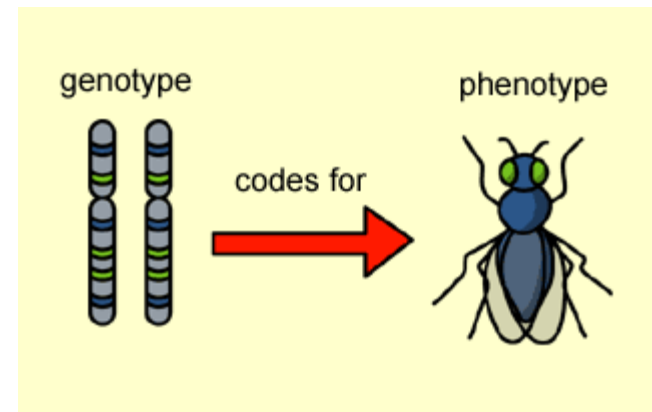
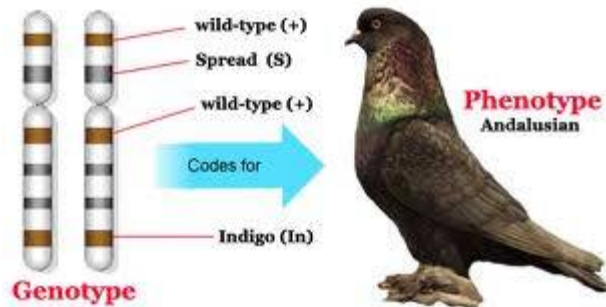
- ویژگی (مشخصه) داده ها



پردازش تکاملی ...

○ مفاهیم ...

- ژنوتایپ (Genotype) - ترکیب تمام ژن ها برای یک فرد مشخص
- فنوتایپ (Phenotype) - خصوصیات ظاهری یک فرد، حاصل شده از رمزگشایی یک ژنوتایپ
- آلل (Allele) - مقادیر مجاز برای هر ژن
 - مقادیر مجاز برای مشخصه‌های هر پاسخ
- برازش (Fitness) - میزان شایستگی یک موجود در جمعیت



پردازش تکاملی ...

○ مراحل یک الگوریتم تکاملی ...

1- تولید جمعیت اولیه (پاسخ‌های اولیه مساله)

2- محاسبه برآزش جمعیت ورودی

3- انتخاب (Selection) برای تولید مثل (Reproduction): انتخاب والدین شایسته‌تر

○ قانون بقای اصلح داروین

4- بازترکیب (Recombination) والدین انتخاب شده: تولید یک یا چند فرزند با ترکیب ژن‌های دو یا چند والد با هم‌برش (Crossover)

○ مفهوم جفت‌گیری

5- جهش (Mutation) فرزندان تولید شده: تغییر تصادفی ژن‌ها در یک کروموزوم

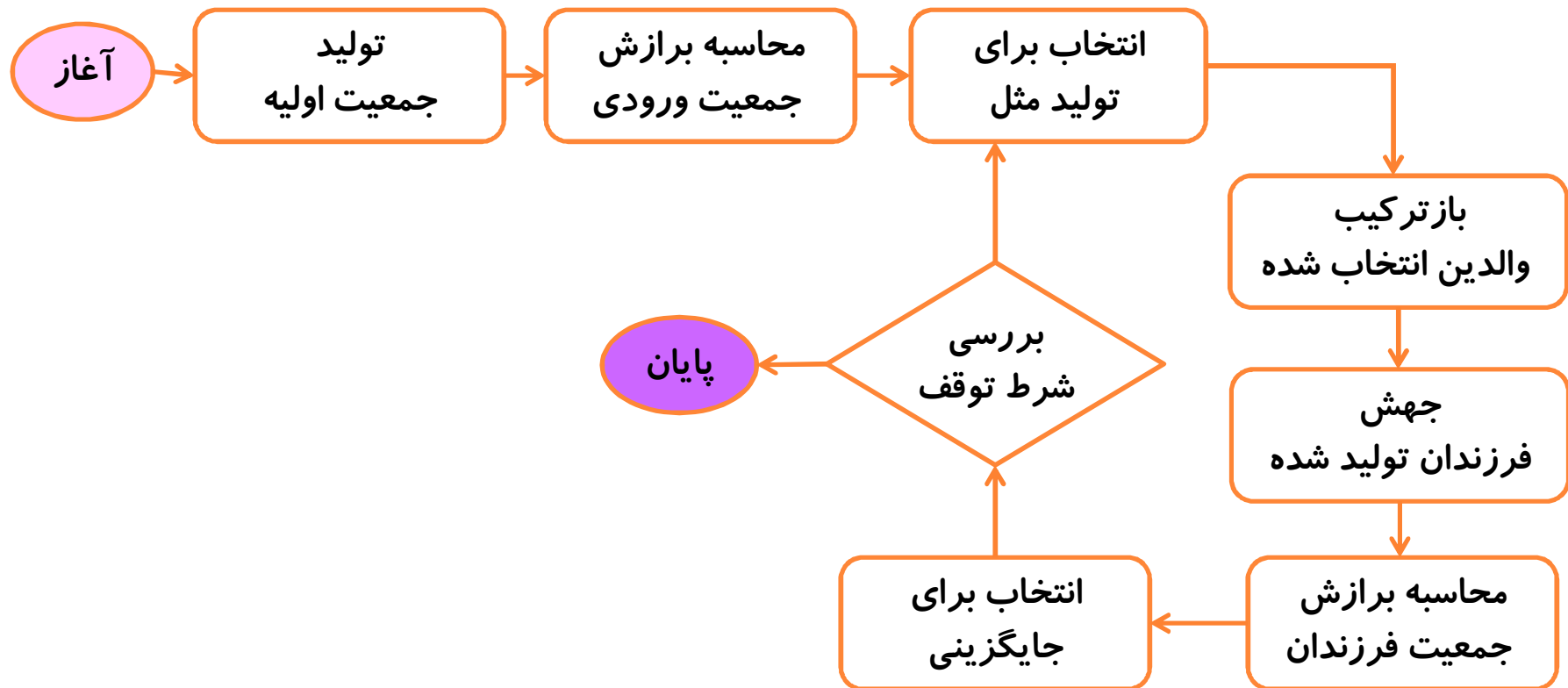
○ یافتن مقادیر جدید برای ژن فرزندان (ایجاد تفاوت با والدین)

6- محاسبه برآزش جمعیت فرزندان: محاسبه شایستگی فرزندان جدید

7- انتخاب برای جایگزینی (Replacement): تولید یک جمعیت به عنوان نسل جدید (از والدین قبلی و فرزندان جدید)

پردازش تکاملی ...

○ مراحل یک الگوریتم تکاملی



پردازش تکاملی: نمایش کروموزوم

از مهم‌ترین مراحل

- فرموله کردن مساله برای الگوریتم تکاملی
- تاثیر زیاد بر کارایی و پیچیدگی الگوریتم

روش‌ها

- بردار دودویی با طول ثابت (هر ژن یک عدد دودویی است): متغیرهای گسسته و اسمی

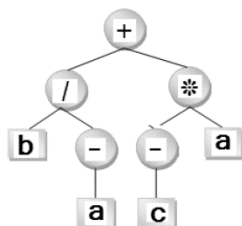
0	1	1	0	1	0	0	0	1
---	---	---	---	---	---	---	---	---

- نمایش مبتنی بر اعداد حقیقی (هر ژن یک عدد حقیقی): برای متغیرهای پیوسته

1.12	32.6	15.1	19.4	6.19	0.12	10.5	12.3	65.1
------	------	------	------	------	------	------	------	------

- جایگشت عناصر: مثلاً برای فروشنده دوره گرد

4	3	9	6	1	7	2	8	5
---	---	---	---	---	---	---	---	---



- نمایش درختی: برای برنامه‌ها و روابط ریاضی

پردازش تکاملی: جمعیت اولیه

○ جمعیت = تعدادی از راه‌حل‌های کاندید برای مساله

• جمعیت اولیه = پاسخ اولیه مساله

- روش تصادفی: مقدار تصادفی در بازه مجاز برای هر ژن (پوشش یکنواخت فضا)
- روش هوشمندانه: تولید کروموزم‌های اولیه با برازندگی بالا (پوشش بخش‌ها مهم فضا)

• اندازه جمعیت: معمولاً ثابت

○ افزایش اندازه جمعیت = تقویت قابلیت پوشش: پوشش دادن فضای جستجوی بزرگ‌تر

○ افزایش اندازه جمعیت = تقویت قابلیت انتفاع: افزایش شانس عملگرهای تولیدمثل

○ افزایش اندازه جمعیت = افزایش محاسبات

پردازش تکاملی: تابع برازش

○ تابع برازش: محاسبه میزان شایستگی پاسخ‌ها (نسل‌ها)

- نگاشت شایستگی هر کروموزوم به یک مقدار عددی

- کاربرد در عملگر انتخاب برای تعیین اعضای برازنده

- وابسته به کاربرد

- تعیین تابع برازش در برخی کاربردها کار مشکلی است



پردازش تکاملی: انتخاب ...

○ انتخاب = یکی از عملگرهای اصلی پردازش تکاملی

- بیانگر مفهوم بقای اصلح نظریه داروین
- هدف: یافتن پاسخ‌های برتر مساله در جمعیت جاری برای اعمال عملگر تولید مثل
- فشار انتخاب (Selective Pressure): میزان فشار عملگر انتخاب برای بردن جمعیت به راه‌حل‌های خوب
 - فشار انتخاب زیاد = توجه بیش از اندازه به اعضای برارنده = کاهش تنوع جمعیت = افزایش قابلیت انتفاع = کاهش قابلیت پوییش = همگرایی سریع (محلی)
- روش‌های مختلفی برای انتخاب
 - انتخاب تصادفی
 - انتخاب نسبی (Proportional)
 - انتخاب رتبه‌ای (Rank-based)
 - انتخاب مسابقه‌ای (Tournament)
 - انتخاب برشی (Truncation)

پردازش تکاملی: انتخاب ...

○ انتخاب تصادفی

- انتخاب هر کدام از اعضای جمعیت با احتمال برابر = عدم استفاده از برازش
- کمترین فشار انتخاب

برازش عضو آم
(محاسبه با تابع برازش)

○ انتخاب نسبی (Proportional)

- شانس بیشتر برای موجودات برتر
- محاسبه احتمال انتخاب عضو آم جمعیت:
$$p_i = \frac{f_i}{\sum_{j=1}^N f_j}$$
- N = تعداد اعضای جمعیت

• پیاده سازی انتخاب نسبی با چرخ روال (Roulette Wheel Selection)

- در نظر گرفتن یک دایره (چرخ دوار) که به تعداد N بخش (قطعه) تقسیم شده است
- هر قطعه مرتبط با هر عضو بوده که اندازه آن متناسب با احتمال آن عضو (برازندگی) است
- چرخاندن چرخ به تعداد N بار و انتخاب عضوی که چرخ روی آن توقف می کند

پردازش تکاملی: انتخاب ...

○ انتخاب نسبی (Proportional)

- فشار انتخاب زیاد چرخ رولت = همگرایی سریع در بهینه محلی

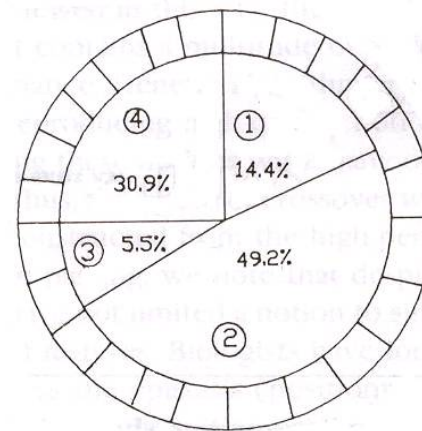
- بهبود: چرخ رولت با بیش از یک اشاره گر

- تعداد N اشاره گر

- یک بار چرخاندن

- تعدیل فشار انتخاب

No	Chromosome	F_i (Fitness)	p_i
1	01101	169	14.4
2	11000	576	49.2
3	01000	64	5.5
4	10011	361	30.9



پردازش تکاملی: انتخاب ...

○ انتخاب رتبه‌ای (Rank-based)

- استفاده از رتبه برازندگی اعضا به جای استفاده از مقدار مطلق برازندگی

○ برآزش برترین عضو N

○ برآزش دومین عضو برتر $N-1$

○ برآزش سومین عضو برتر $N-2$

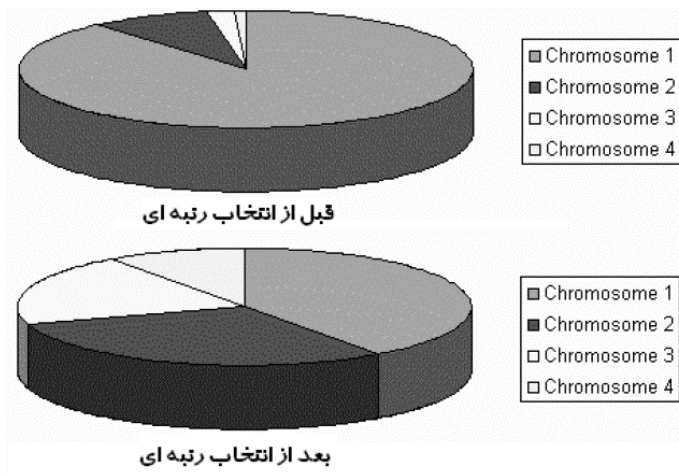
○ ...

○ برآزش آخرین عضو برتر 1

○ محاسبه احتمال بر اساس برآزش‌های جدید

$$p_i = \frac{f_i}{\sum_{j=1}^N f_j}$$

برآزش عضو i ام
(محاسبه با روش فوق)



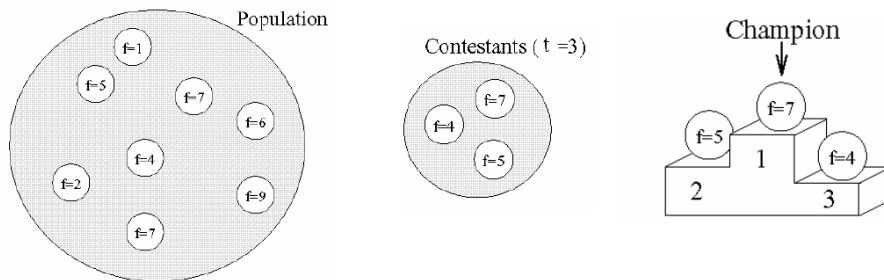
- کاهش فشار انتخاب در مقایسه با چرخ رولت

- عدم همگرایی زودرس

پردازش تکاملی: انتخاب ...

○ انتخاب مسابقه‌ای (Tournament)

- انتخاب یک گروه ($t < N$ عضو) از جمعیت به صورت تصادفی
- مقایسه برآزش اعضای انتخاب شده و انتخاب بهترین عضو
- اگر t خیلی بزرگ نباشد، از انتخاب برترین افراد جلوگیری می‌شود (فشار انتخاب کم)
 - برای $t = N$ ، همیشه بهترین فرد انتخاب می‌شود = بیشترین فشار انتخاب
- اگر t خیلی کوچک باشد، شانس انتخاب ضعیف‌ترین افراد افزایش می‌یابد
 - برای $t = 1$ ، الگوریتم انتخاب تصادفی
- مقدار معمول برای t : مقدار 2 یا 3



پردازش تکاملی: انتخاب ...

○ انتخاب برشی (Truncation)

- مرتب کردن اعضای جمعیت بر اساس شایستگی آنها
- انتخاب T درصد از اعضای برتر
- انتخاب N عضو به صورت تصادفی (از میان T درصد از برترین اعضا)
- مقدار T بزرگتر = فشار انتخاب کمتر
 - مقدار $T=100$ معادل انتخاب تصادفی

پردازش تکاملی: تولید مثل ...

○ تولید مثل = تولید جمعیت جدید (فرزندان) از والدین انتخاب شده با

• بازترکیب (Recombination) یا هم‌برش (Crossover)

- معادل مفهوم جفت‌گیری
- تولید یک یا چند فرزند با ترکیب ژن‌های تصادفی انتخاب شده از دو یا چند والد
- اعمال روی اعضای (والدین) انتخاب شده در مرحله انتخاب (برازنده)
- تولید فرزندان مشابه والدین: انتقال ژن‌های کروموزوم‌های والدین به فرزندان
 - به ارث بردن ژن‌های والدین و تولید پاسخی با برازش بهتر
- اعمال عملگر بازترکیب روی اعضای جمعیت با احتمال p_c
- استفاده بیشتر از عملگر بازترکیب = افزایش قابلیت انتفاع

• جهش (Mutation)

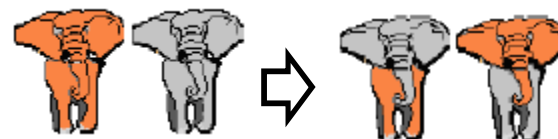
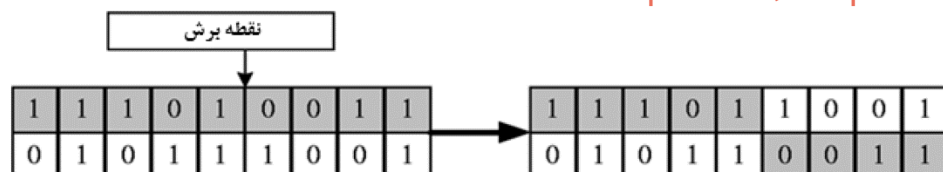
- تغییر تصادفی ژن‌ها در کروموزوم
- هدف: یافتن مقادیر جدید ژن برای فرزندان که در والدین نبوده است
 - افزایش تنوع ژنوتایی
- اعمال عملگر جهش روی فرزندان با احتمال p_m
- افزایش بیشتر از عملگر جهش = افزایش قابلیت پویش

پردازش تکاملی: تولید مثل (باز ترکیب) ...

○ حالت دودویی ...

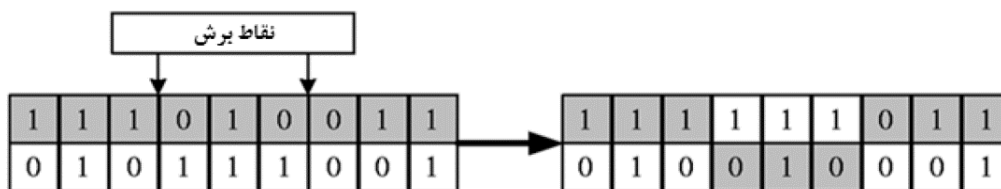
• هم‌برش (باز ترکیب) تک نقطه‌ای (One-point crossover)

- انتخاب یک نقطه تصادفی و برش کروموزوم‌ها از این نقطه
- بخش اول والد اول و بخش دوم والد دوم = فرزند اول
- بخش دوم والد اول و بخش اول والد دوم = فرزند دوم



• هم‌برش (باز ترکیب) دونقطه‌ای

- انتخاب دو نقطه تصادفی و برش کروموزوم‌ها از این نقاط
- بخش اول و سوم والد اول و بخش دوم والد دوم = فرزند اول
- بخش دوم والد اول و سوم والد اول و بخش اول والد دوم = فرزند دوم

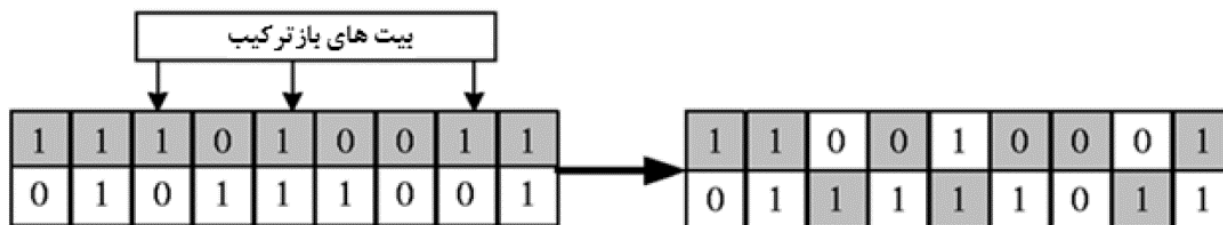


پردازش تکاملی: تولید مثل (باز ترکیب) ...

○ حالت دودویی

- هم‌برش (باز ترکیب) یکنواخت (Uniform crossover)

- انتخاب هر ژن فرزند از ژن متناسب یکی از دو والد
- استفاده از یک توزیع تصادفی برای انتخاب ژن فرزند
- شانس مشابه هر دو والد برای حضور در ژن فرزند
- برای ضریب ترکیب 50٪، شانس هر دو والد برابر خواهد بود

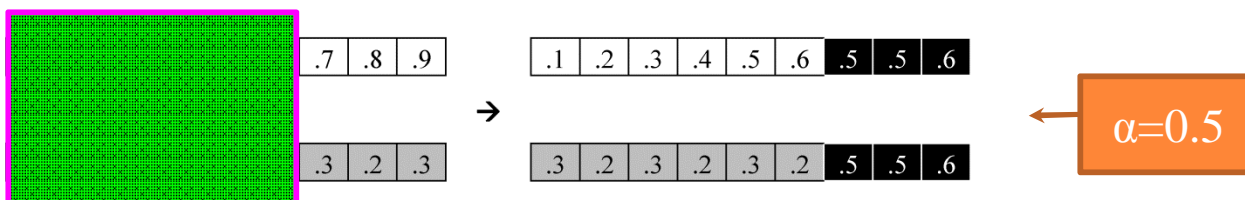


پردازش تکاملی: تولید مثل (باز ترکیب) ...

○ حالت حقیقی ...

• باز ترکیب ساده (Simple)

- انتخاب یک بخش از کروموزوم‌ها
- انتقال بخش اول از والد اول به فرزند اول
- انتقال بخش اول از والد دوم به فرزند دوم
- محاسبه ژن‌های بخش انتخاب نشده فرزند اول: جمع کردن مقدار ژن‌های دو کروموزوم و ضرب حاصل در α (بین 0 و 1)
- محاسبه ژن‌های بخش انتخاب نشده فرزند دوم: جمع کردن مقدار ژن‌های دو کروموزوم و ضرب حاصل در $1-\alpha$



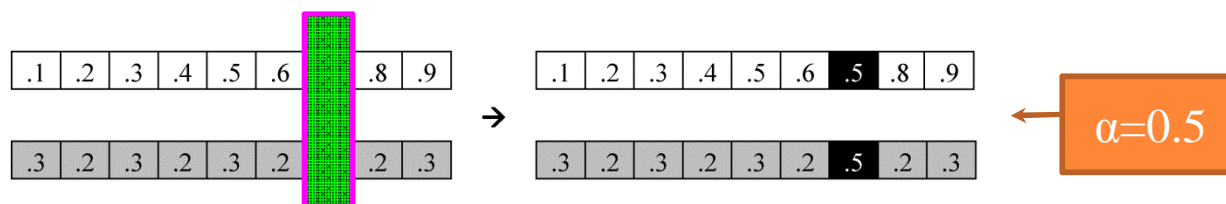
- توسعه: استفاده از عملگرهای دیگر، غیر از جمع

پردازش تکاملی: تولید مثل (باز ترکیب) ...

○ حالت حقیقی

- باز ترکیب حسابی ساده (Simple Arithmetic)

○ مشابه باز ترکیب ساده اما فقط یک ژن تغییر می کند



- باز ترکیب حسابی کامل (Whole Arithmetic)

○ مشابه باز ترکیب ساده اما تمام ژن ها تغییر می کند

○ فرزندان شباهتی به والدین ندارند: افزایش قابلیت پویا



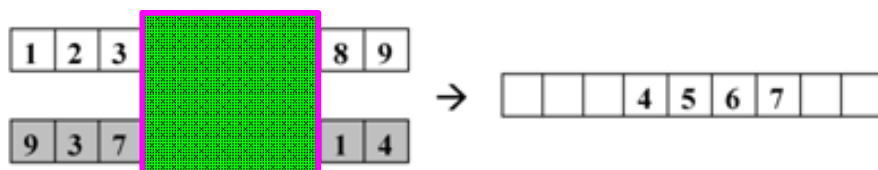
پردازش تکاملی: تولید مثل (باز ترکیب) ...

○ حالت جایگشت ...

• باز ترکیب ترتیبی (Order Recombination)

- انتخاب دو نقطه تصادفی
- گام اول: استفاده از والد اول = انتقال بخش میانی والد اول به فرزند اول
- گام دوم: استفاده از والد دوم = شروع از نقطه اول بخش پایانی برای استفاده از ژنهای والد دوم در بخش پایانی فرزند اول
- اگر مقدار ژنی قبلاً در فرزند وجود داشته باشد از آن صرف نظر می شود
- ادامه این فرایند برای بخش آغازی (ابتدایی) فرزند اول

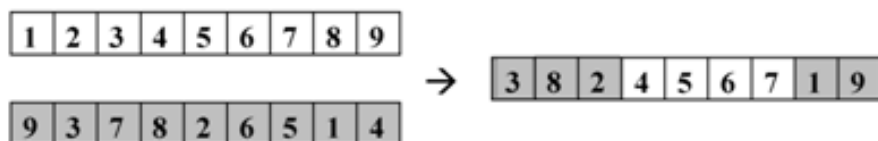
گام اول - استفاده از والد اول



○ برای فرزند دوم

- فرآیند قبل با عوض کردن جای دو

گام دوم - استفاده از والد دوم



پردازش تکاملی: تولید مثل (باز ترکیب) ...

○ حالت جایگشت

• باز ترکیب چرخشی (Cycle Recombination)

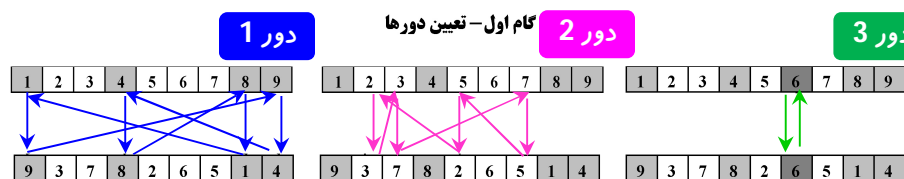
○ گام اول: تعیین دورها

- شروع از ژن اول والد اول و رفتن به شماره ژنی از والد دوم که در این ژن (اول از والد اول) نشان داده شده است
- تکرار گام فوق برای ژن انتخاب شده در والد دوم: رفتن به ژنی از والد اول که این ژن نشان می دهد
- دور: در صورت رسیدن به ژن اول والد اول

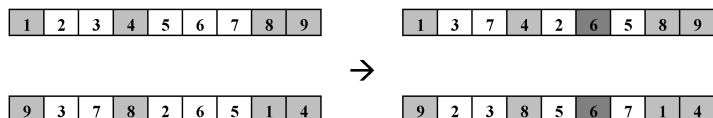
- تکرار الگوریتم فوق در صورت وجود ژن پیمایش نشده با شروع از اولین ژن پیمایش نشده در والد اول

○ گام دوم: تعیین ژن ها

- انتقال ژن های دور اول از والد اول به فرزند
- انتقال ژن های دور دوم از والد دوم به فرزند
- انتقال ژن های دور سوم از والد اول به فرزند
- انتقال ژن های دور چهارم از والد دوم به فرزند
- ...



گام دوم - کپی کردن دورها در فرزندان



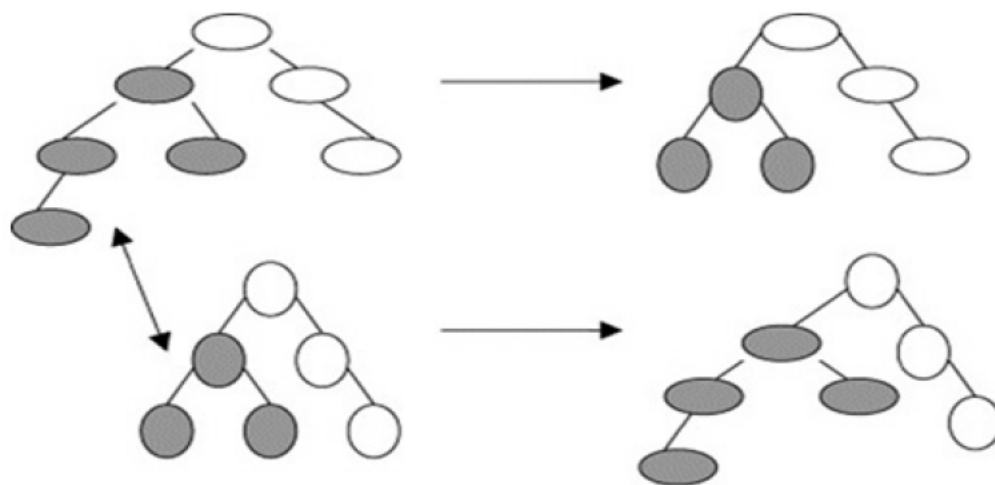
نقاط شکست در تشکیل فرزندان = دورها

- برای فرزند دوم: عوض کردن جای دو والد

پردازش تکاملی: تولید مثل (باز ترکیب)

○ حالت درخت

- در نظر گرفتن نقاط شکست در دو والد
- جابجایی زیردرخت‌های انتخاب شده



پردازش تکاملی: تولید مثل (جهش) ...

○ حالت دودویی: معکوس سازی بیت (Bit-flipping mutation)

- انتخاب یک یا چند بیت به صورت تصادفی
- تغییر مقدار آن بیت (0 به 1 و برعکس)

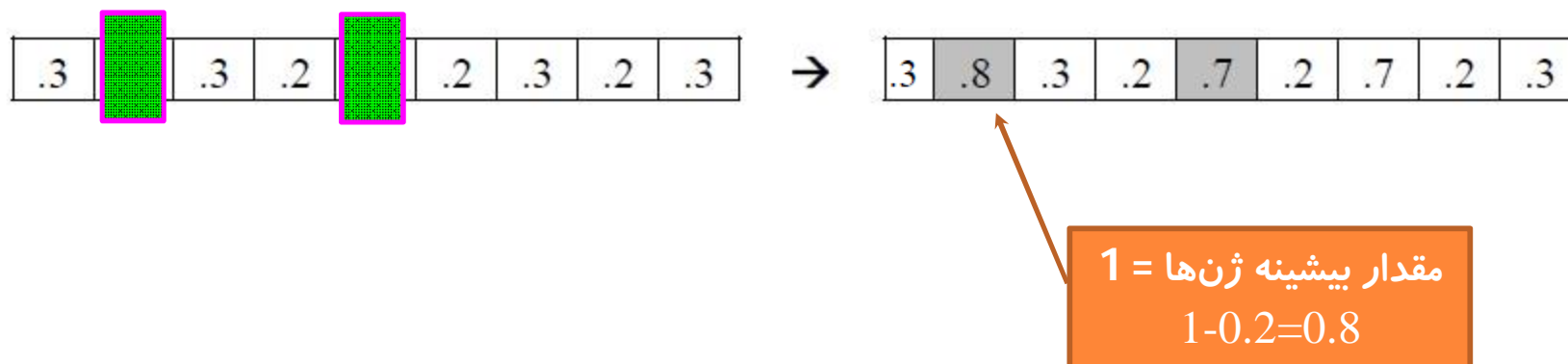
- عدم استفاده از اطلاعات موجود



پردازش تکاملی: تولید مثل (جهش) ...

○ حالت حقیقی: جهش مکمل (Complement Mutation)

- انتخاب یک یا چند ژن برای ژن
- کم کردن مقدار پیشینه ممکن برای آن ژن از مقدار جاری آن ژن
- جمع مقادیر قبلی و جدید ژن = مقدار پیشینه ممکن برای آن ژن

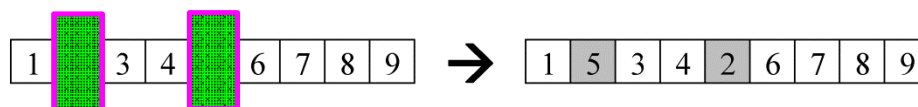


پردازش تکاملی: تولید مثل (جهش) ...

○ حالت جایگشت ...

- جهش جابجایی (Swap Mutation)

- انتخاب دو ژن به صورت تصادفی و جابجا کردن مقادیر آنها

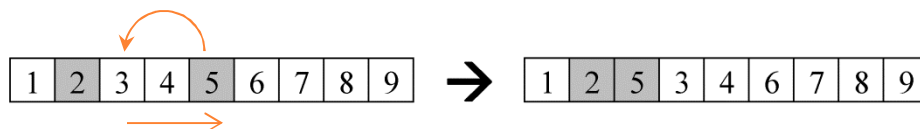


- جهش درجی (Insert Mutation)

- انتخاب دو ژن به صورت تصادفی

- کپی کردن ژن دوم در ژن بعد از ژن اول

- شیفت دادن ژن‌های دیگر به راست



پردازش تکاملی: تولید مثل (جهش) ...

○ حالت جایگشت

• جهش درهم‌سازی (Scramble Mutation)

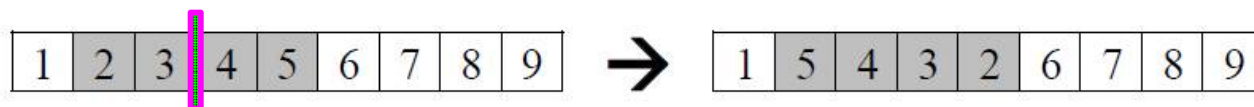
- انتخاب دو ژن به صورت تصادفی
- جابجایی تصادفی مقادیر ژن‌های بین دو نقطه



- افزایش فاصله بین دو نقطه انتخابی = قابلیت پویش بیشتر

• جهش معکوس (Inversion Mutation)

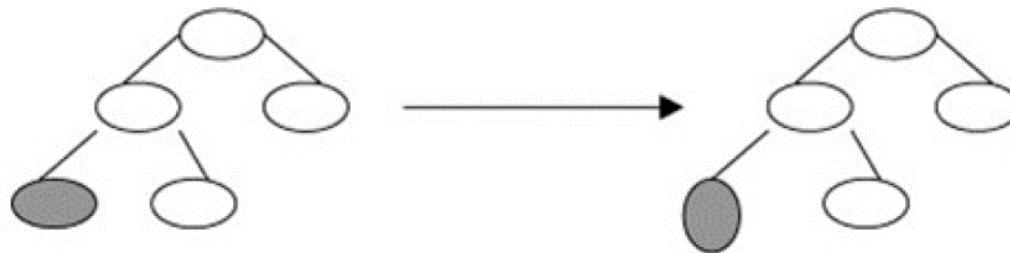
- انتخاب دو ژن به صورت تصادفی
- جابجا کردن مقادیر ژن‌های بین دو نقطه به صورتی که نسبت به وسط آنها وارونه باشند
- گذاشتن آینه در نقطه وسط بین دو نقطه



پردازش تکاملی: تولید مثل (جهش)

○ حالت درخت

- تغییر مقدار مربوط به یک گره
- مقدار جدید = تصادفی



پردازش تکاملی: جایگزینی ...

○ گزینش جمعیت جدید (پاسخ‌های جدید)

- از روی جمعیت والدین (پاسخ‌های فعلی) و جمعیت فرزندان (پاسخ‌های جدید)
- دو نوع کلی

○ جایگزینی حالت پایدار (پایا) (Steady State Replacement)

○ جایگزینی نسلی (Generational Replacement)

○ جایگزینی حالت پایدار (پایا) (Steady State Replacement) ...

- حفظ کردن بخش بزرگی از جمعیت والدین
- جایگزینی بخشی از والدین با بهترین فرزندان تولید شده
- حفظ بافت قبلی جمعیت و گوناگونی جمعیت: جلوگیری از همگرایی به بهینه محلی
- پارامتر کنترلی: p_{rep} = درصدی از والدین که تغییر می‌کنند
 - افزایش این پارامتر = کاهش تنوع = همگرایی سریع
 - شکاف نسلی (generation gap): میزان همپوشانی نسل کنونی و نسل بعد

پردازش تکاملی: جایگزینی

○ جایگزینی نسلی (Generational Replacement)

- جایگزینی کل والدین (نسل قبل) با کل فرزندان (نسل جدید)
- جایگزین شدن بهترین عضو در جمعیت والدین با ضعیف‌ترین عضو در جمعیت فرزندان
- نخبه سالاری (Elitism): جلوگیری از نابودی بهترین پاسخ

- همگرایی سریع الگوریتم

• در استراتژی تکاملی: روش انتخاب (μ, λ) و $(\mu + \lambda)$

- روش $(\mu + \lambda)$: تعداد μ عضو برتر والدین و تعداد λ فرزند انتخاب می‌شوند
- روش (μ, λ) : تعداد μ عضو برتر از میان λ فرزند انتخاب شده و به نسل بعد منتقل می‌شوند

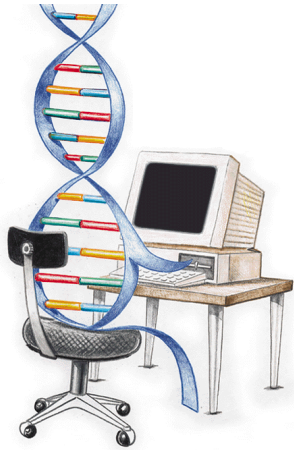
پردازش تکاملی: شرایط توقف

- رسیدن به بهترین پاسخ
 - برای حالتی که مقدار برازش بهترین پاسخ را داریم
 - همیشه ممکن نیست
- محدود کردن تعداد نسل‌ها (تعداد تکرار الگوریتم)
- راکد شدن (Stagnant) جمعیت
 - عدم تغییر جمعیت در نسل‌های متوالی
 - شمارش تعداد نسل‌هایی که بهترین پاسخ تغییر نکرده است

پردازش تکاملی: کنترل قابلیت پویش و انتفاع ...

○ به کارگیری روش مناسب برای انتخاب، بازترکیب، جهش و جایگزینی

- انتخاب چرخ رولت ساده: تقویت قابلیت انتفاع
- انتخاب چرخ رولت با چنداشاره گر: تقویت قابلیت پویش
- بازترکیب یکنواخت: بهترین انتخاب برای تقویت قابلیت پویش
- جهش: تخریب بیشتر در کروموزوم‌ها = قابلیت پویش بیشتر
- جایگزینی حالت پایدار در مقایسه با جایگزینی نسلی قابلیت پویش بیشتری دارد



پردازش تکاملی: انواع الگوریتم‌ها

- الگوریتم ژنتیک (Genetic Algorithm)
- برنامه‌نویسی ژنتیک (Genetic Programming)
- استراتژی تکامل (Evolutionary Strategy)
- برنامه‌نویسی تکاملی (Evolutionary Programming)
- تکامل تفاضلی (Differential Evolution)
- الگوریتم فرهنگی (Cultural Algorithm)
- الگوریتم هم‌تکاملی (Co-Evolutionary Algorithm)
- الگوریتم ممیک (Mimetic Algorithm)
- بهینه‌سازی تولیدمثل غیرجنسی (Asexual Reproduction)
(Optimization)
- ...

الگوریتم ژنتیک ...

○ تاریخچه

- اولین بار توسط Fraser در 1957
- ادامه توسط Bremermann در 1962 و Reed در 1967
- تکمیل و توسعه توسط Holland در 1975
- هالند = پدر الگوریتم ژنتیک

○ کاربرد

- جستجو، بهینه‌سازی، یادگیری ماشین، کنترل، زمان‌بندی کارها، رباتیک

○ عملگرها

- انتخاب (مدل‌سازی قانون بقای اصلح)
- تولید مثل: بازترکیب و جهش

الگوریتم ژنتیک ...

○ خصوصیات الگوریتم ژنتیک استاندارد

- استفاده از نمایش رشته بیتی
- طول ثابت و یکسان برای کروموزوم‌ها
- تعداد اعضای جمعیت ثابت
- استفاده از عملگر انتخاب نسبی برای انتخاب والدین
- استفاده از باز ترکیب تک نقطه
- استفاده از جهش معکوس سازی بیت
- احتمال باز ترکیب بالا (حدود 0.95 و بیشتر)
- احتمال جهش پایین: برابر با $1/L$ که L طول کروموزوم است

الگوریتم ژنتیک: مثال

○ فروشنده دوره گرد (Travelling Salesman Problem)

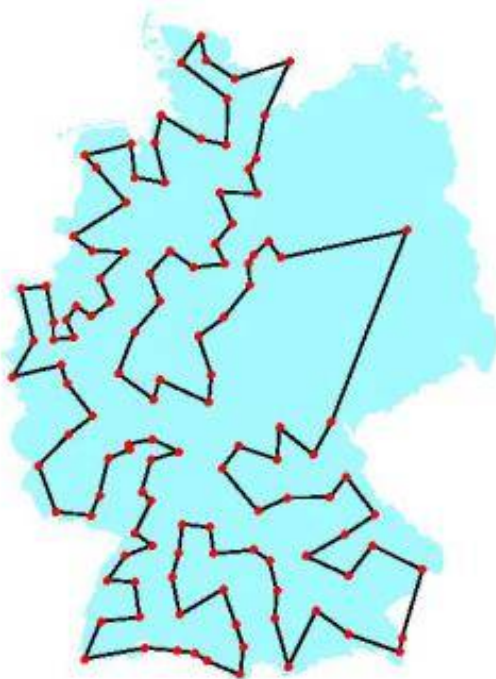
- یافتن کوتاه ترین مسیر برای یک فروشنده با عبور از n شهر
- از تمامی شهرها دقیقاً یک بار بگذرد و به شهر اول برگردد
- جزو مسائل NP-Hard
- تعداد کل راه حل ها برای n شهر $= 0.5(n-1)!$

○ برای 5 شهر $= 12$

○ برای 10 شهر $= 1.814.400$

○ برای 30 شهر $= 1.3 * 10^{32}$

- حل با الگوریتم ژنتیک



الگوریتم ژنتیک: مثال (فروشنده دوره گرد) ...

○ نمایش کروموزوم: جایگشت

- برای شهرهای تهران (1)، اصفهان (2)، مشهد (3)، شیراز (4)، تبریز (5)، زاهدان (6)

○ [136425]

○ برآزش: جمع کل فاصله (هزینه) تور

- فاصله کم تر = برآزش بهتر

	تهران	اصفهان	مشهد	شیراز	تبریز	زاهدان
تهران	0	439	894	924	1567	1567
اصفهان	439	0	1222	485	1190	1190
مشهد	894	1222	0	1374	951	951
شیراز	924	485	1374	0	1523	1100
تبریز	1567	1190	951	1523	0	2166
زاهدان	1567	1190	951	1100	2166	0



○ انتخاب: مسابقه‌ای، رتبه‌ای، ...

○ بازترکیب: ترتیبی یا چرخشی

○ جهش: جابجایی، درجی، جهش، معکوس

الگوریتم ژنتیک: مثال (فروشنده دوره گرد)

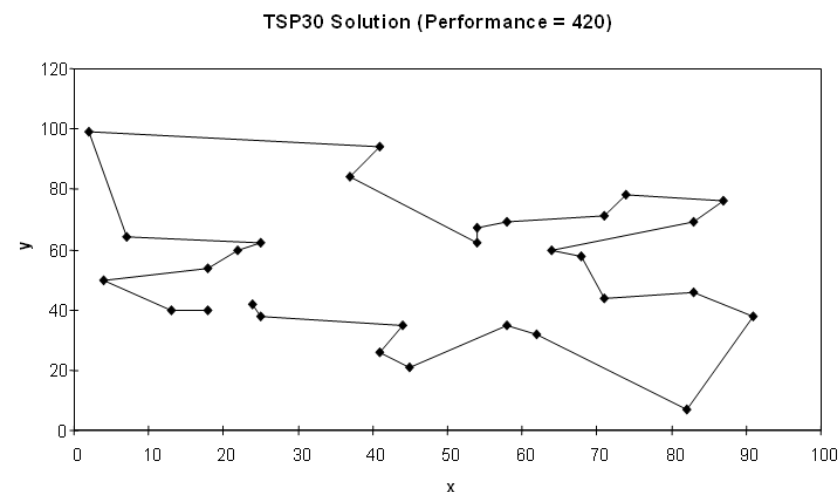
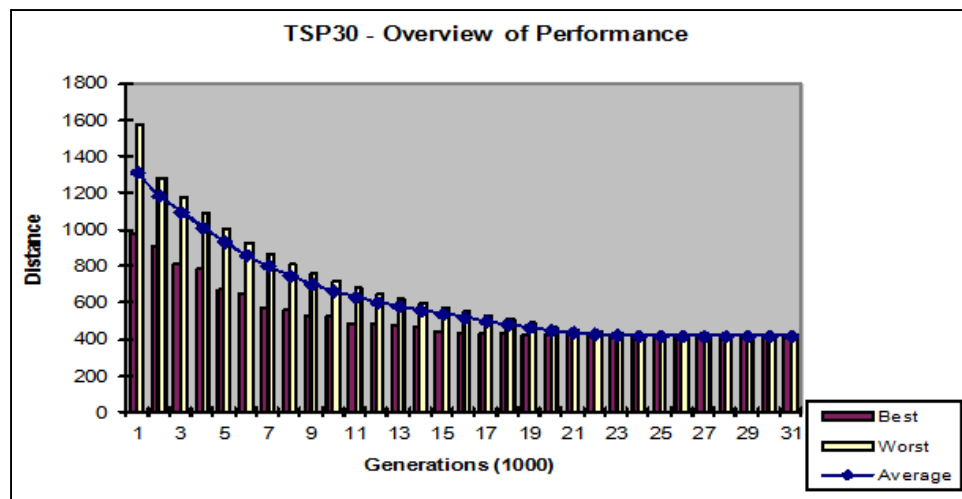
○ پارامترها: برای 30 شهر

• جمعیت: 50

○ جمعیت اولیه: تصادفی (می تواند هوشمندانه هم باشد)

• تعداد نسل ها (تکرار): 10.000

• نرخ جهش (p_m) = 5%



برنامه‌نویسی ژنتیک

○ برنامه‌نویسی ژنتیک (Genetic Programming)

- ارائه شده توسط Koza برای تکامل برنامه‌های کامپیوتری
 - هدف: پیدا کردن برنامه بهینه
- کاربردها: برنامه‌نویسی خودکار، برنامه‌ریزی، درخت تصمیم، طراحی شبکه عصبی، ...
- می‌توان آن را یکی از الگوریتم‌های ژنتیک دانست!
- تفاوت اصلی با الگوریتم ژنتیک: استفاده از نمایش درختی
- ویژگی‌ها
 - طول متغیر برای کروموزوم (ویژگی انحصاری)
 - تعداد اعضای جمعیت ثابت است
 - استفاده از عملگرهای بازترکیب (با احتمال بیشتر از 90%) و جهش (کمتر از 10%)
 - استفاده از عملگر تغییر معماری (Architecture Alternation)
 - ویرایش برخی قوانین (مثلاً جایگزینی x با x and x)
 - شناسایی بلوک‌های سازنده و جلوگیری از تغییر بلوک‌های مفید با عملگرهای تولید مثل
- تابع برازش: کارایی موجود (برنامه) بر روی یک مجموعه آزمون

استراتژی تکامل

○ استراتژی تکامل (Evolutionary Strategy)

- ارائه شده توسط ریچنبرگ در سال 1960
- هدف: بهینه‌سازی فرآیند تکامل است
- علاوه بر ویژگی‌های ژنی برای هر موجود، پارامترهای استراتژی هم وجود دارد
 - مدل‌سازی رفتار موجود در محیط
 - تکامل هم‌زمان ویژگی‌های ژنی و پارامترهای استراتژی
- کاربردها: بهینه‌سازی، طراحی کنترل‌گر، سیستم‌های قدرت
- ویژگی‌ها
 - نمایش اعداد حقیقی
 - تعداد ثابت برای اعضای جمعیت
 - استفاده از عملگر جهش با نرخ تطبیقی (بازرسی در موارد محدودی استفاده می‌شود)
 - پاسخ‌های بهتر دارای نرخ جهش کم و برعکس
 - هر موجود دارای پارامترهای استراتژی مربوط به خود است
 - استفاده از جایگزینی‌های $(\mu + \lambda)$ و (μ, λ) [مراجعه به چند اسلاید قبل]