

رایانش تکاملی

EVOLUTIONARY COMPUTING

دکتر امیر فرید امینیان مدرّس

دانشگاه صنعتی سجاد مشهد

پاییز ۱۳۹۶

محاسبات تکاملی و زیستی

⊙ الگوریتم‌های تکاملی (evolutionary computing)

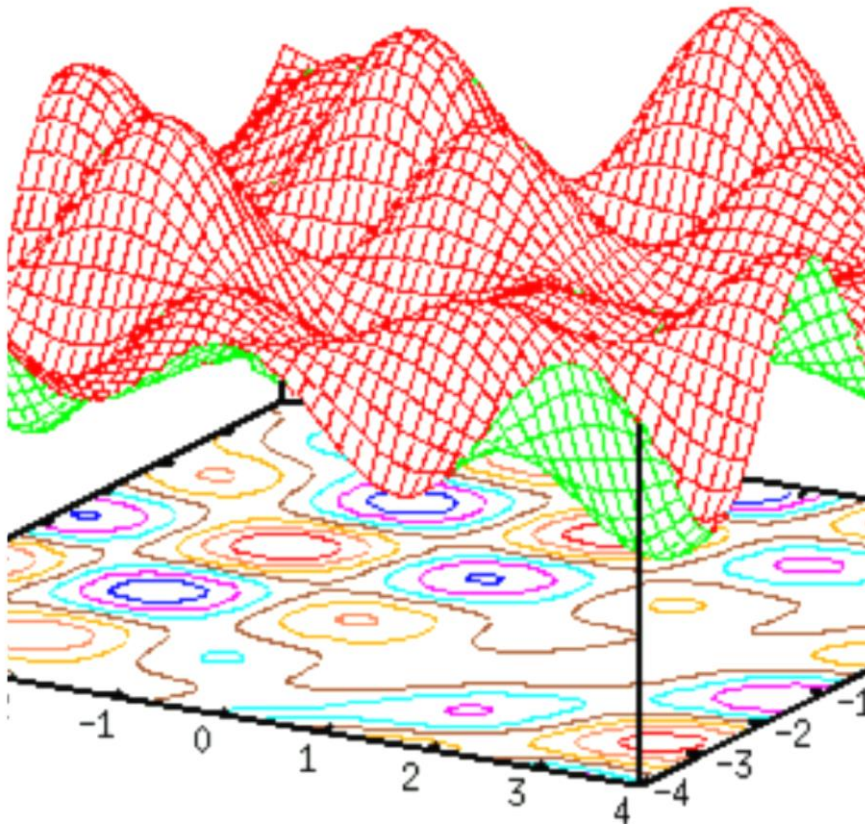
⊙ محاسبات زیستی (biological computation)

■ حل مسایل بهینه‌سازی، جستجو، یادگیری – با الهام از طبیعت و موجودات هوشمند طبیعی

■ شبیه‌سازی نظریه تکامل داروین (۱۸۵۹)

بخش اول – الگوریتم‌های جستجو

انواع الگوریتم‌های جستجو – طبقه‌بندی الگوریتم‌های فرامکاشف‌های
– فضای جستجو و دورنمای برآزش – پوشش و ارتفاع



مسایل بهینه‌سازی (OPTIMIZATION)

⊙ نمایش مساله

- یکی از ملزومات حل مساله، روش نمایش آن (problem representation) است.

- روشی برای تعیین هر وضعیت (state)

- هر حالت، یک راه‌حل برای مساله؛ بهینه یا غیر بهینه

⊙ الگوریتم جستجو

- روش یافتن وضعیت بهینه

- معیاری برای ایجاد تمایز بین راه‌حل‌ها و انتخاب راه حل بهینه

- سراسری - محلی

- راه حل رضایت‌بخش (satisfactory solution) - به اندازه کافی خوب (good.enough solution)

انواع روش‌های جستجو



جستجوی تحلیلی



جستجوی ناآگاهانه (کور)



جستجوی آگاهانه (مکاشفه‌ای)

جستجوی تحلیلی

◎ جستجوی تحلیلی (analytical search)

- ضابطه ریاضی مدل موجود است.
- جستجو برای حالت‌های بهینه بوسیله مفاهیم ریاضی هدایت می‌شود
 - بردار گرادیان
 - مشتق دوم
 - روش نیوتن-رافسون برای بدست آوردن ریشه تابع
- معایب
 - عدم کارآمدی در فضاهاى بزرگ
 - تعریف فقط در فضای پیوسته و مشتق پذیر
 - احتیاج به دانستن تمامی روابط ریاضی

جستجوی ناآگاهانه

● جستجوی ناآگاهانه (uninformed search)

■ جستجوی کور (blind search)

- فقط تدوین مساله؛ شیوه ارتباط حالت‌ها با یکدیگر؛ موجود است.
- اطلاعات اضافی درباره ماهیت مساله وجود ندارد.
- قدرت تشخیص یک حالت هدف از حالت غیرهدف وجود دارد.

■ جستجوی کامل

- پیمایش تمام فضای حالت و یافتن بهترین راه حل (هدف)
- تضمین یافتن راه حل در صورت وجود

■ جستجوی ناکامل

- پیمایش فضا تا هنگام یافتن یک راه حل؛
- تضمینی برای یافتن راه حل وجود ندارد.

■ جستجوی کامل می‌تواند بهینه یا غیربهینه باشد.

■ الگوریتم‌های جستجوی ناآگاهانه براساس ترتیب گسترش حالت‌ها متمایز شده‌اند:

- اول سطح؛ اول عمق؛ هزینه یکنواخت؛ بازگشتی؛ ...

جستجوی آگاهانه

◎ جستجوی آگاهانه (informed search) – جستجوی مکاشفه‌ای (heuristic search)

- علاوه بر تعریف فضای مساله، اطلاعات اضافی درباره ماهیت مساله دارند

- استفاده از یک تابع تخمینی برای بدست آوردن فاصله تا هدف، می‌تواند استراتژی بهتری و هوشمندانه‌ای برای رسیدن به هدف ایجاد کند.

- تابع هیوریستیک

- انواع روش‌های آگاهانه

- جستجوی اول بهترین: حریصانه؛ A^* ، نسخه‌های تکراری و حافظه محدود A^* .

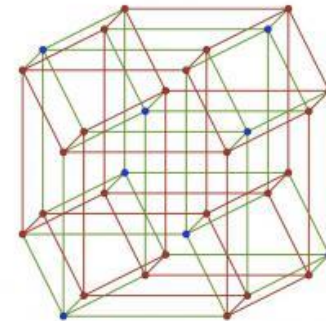
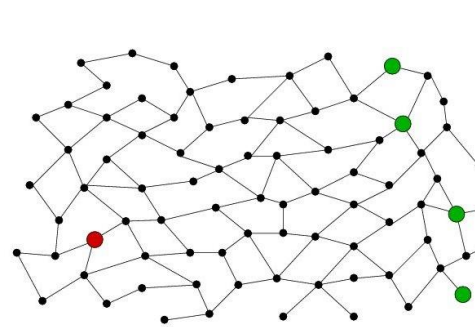
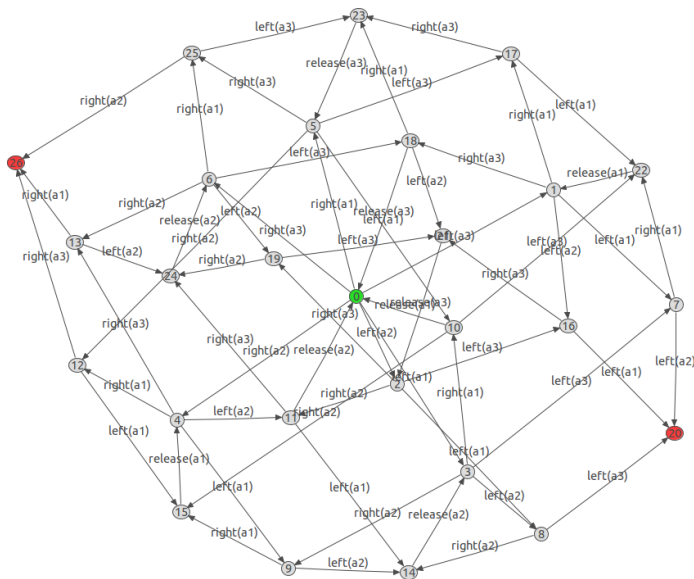
- جستجوی فرامکاشفه‌ای

- الهام از یک پدیده طبیعی برای جستجوی فضای حالت پیچیده، دشوار یا نامنظم

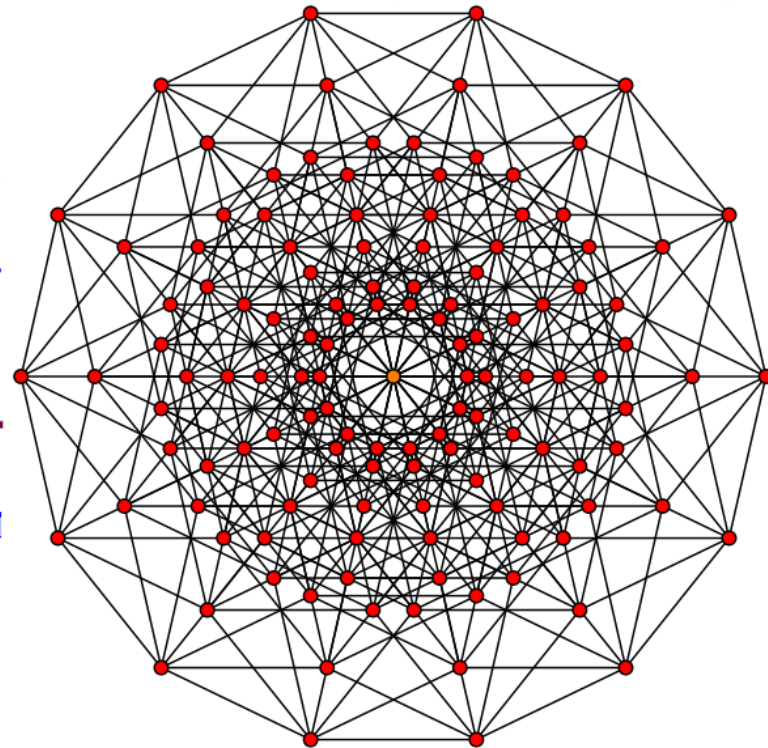
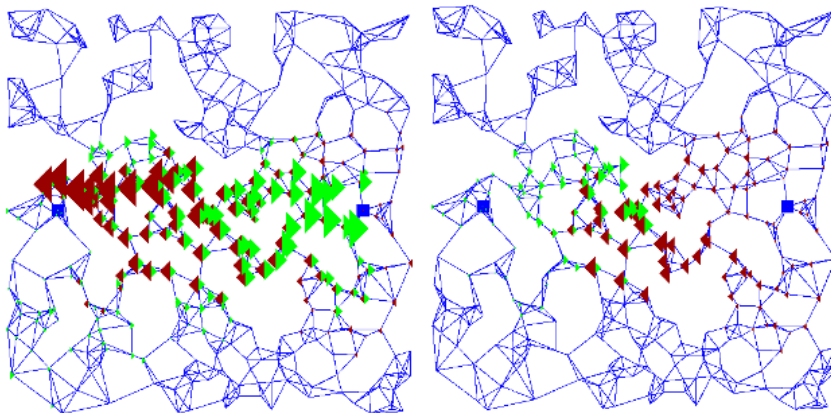
الگوریتم‌های فرامکاشفه‌ای (METAHEURISTIC)

- ⊙ یک هیوریستیک سطح بالا برای یافتن، تولید یا انتخاب هیوریستیکی که بتواند جواب‌های به اندازه کافی خوب برای مساله بهینه‌سازی تولید کند.
 - به خصوص در هنگام وجود اطلاعات ناکافی، ناقص یا محدود.
- ⊙ نمونه‌برداری از مجموعه‌ای از راه‌حل‌ها وقتی که مدل‌سازی تمامی آنها به خاطر اندازه بزرگ، مقدور نباشد.
- ⊙ داشتن فرضیات بسیار کم درباره مساله بهینه‌سازی که قرار است حل شود.
 - کارامدی زیاد برای دامنه بسیار وسیعی از مسایل متنوع
- ⊙ الهام از یک پدیده طبیعی برای جستجوی فضای حالت.

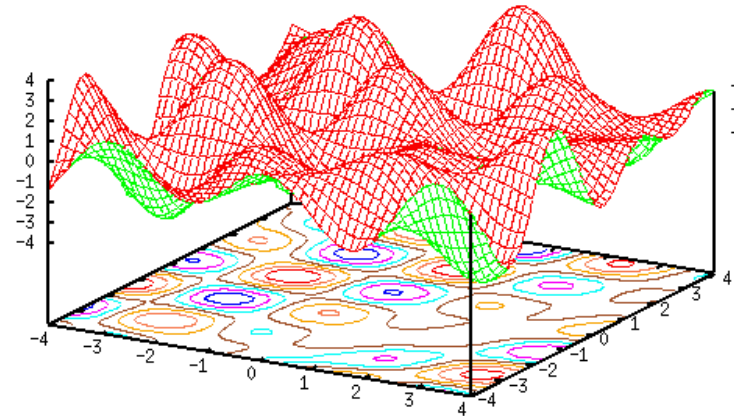
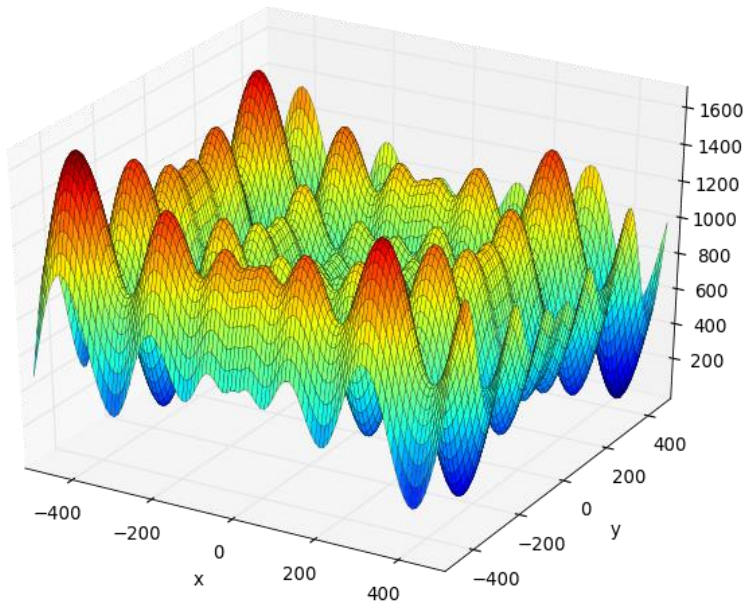
ساختار مساله و دورنمای فضای حالت



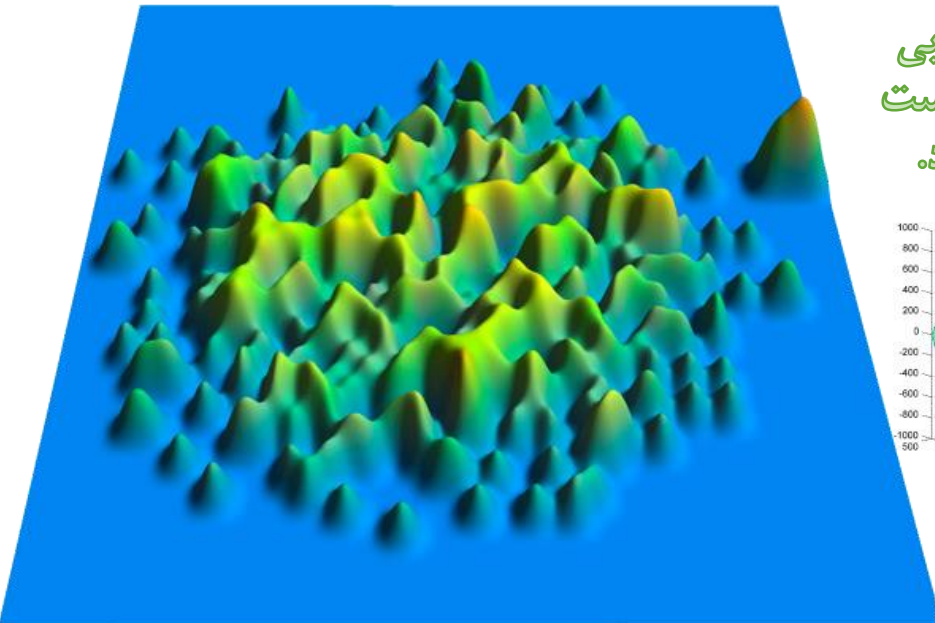
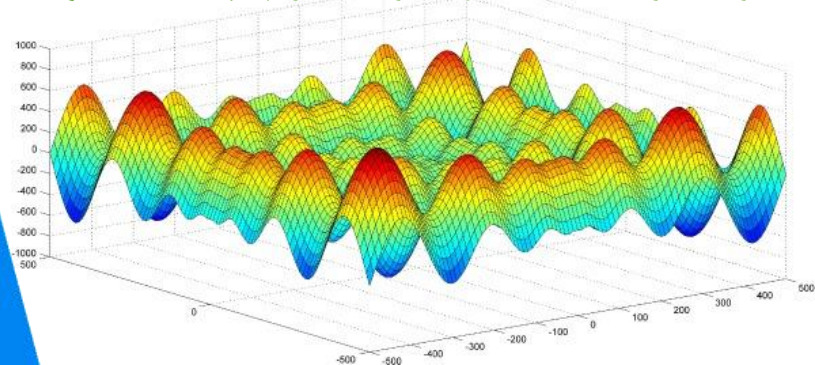
ساختار مساله (ابعاد و نحوه ارتباط حالتها)
ممکن است خیلی پیچیده و غیر یکنواخت باشد.



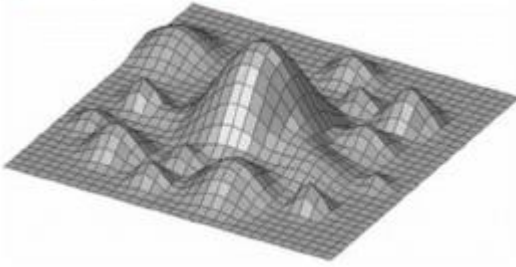
ساختار مساله و دورنمای فضای حالت



دورنما (نگاشت فضای حالت توسط تابع ارزیابی
به دامنه اعداد) کاملاً وابسته به ساختار مساله است
و معمولاً شکل غیریکنواخت و پیچیده‌ای دارد.

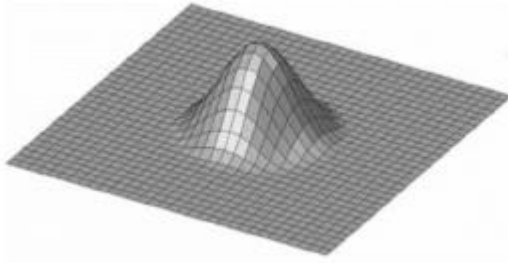


پویش و انتفاع



○ قابلیت پویش (exploration capability)

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



○ قابلیت انتفاع (exploitation capability)

- توجه به دستاوردهای الگوریتم در طول جستجو
- تمرکز بر پاسخها
- 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

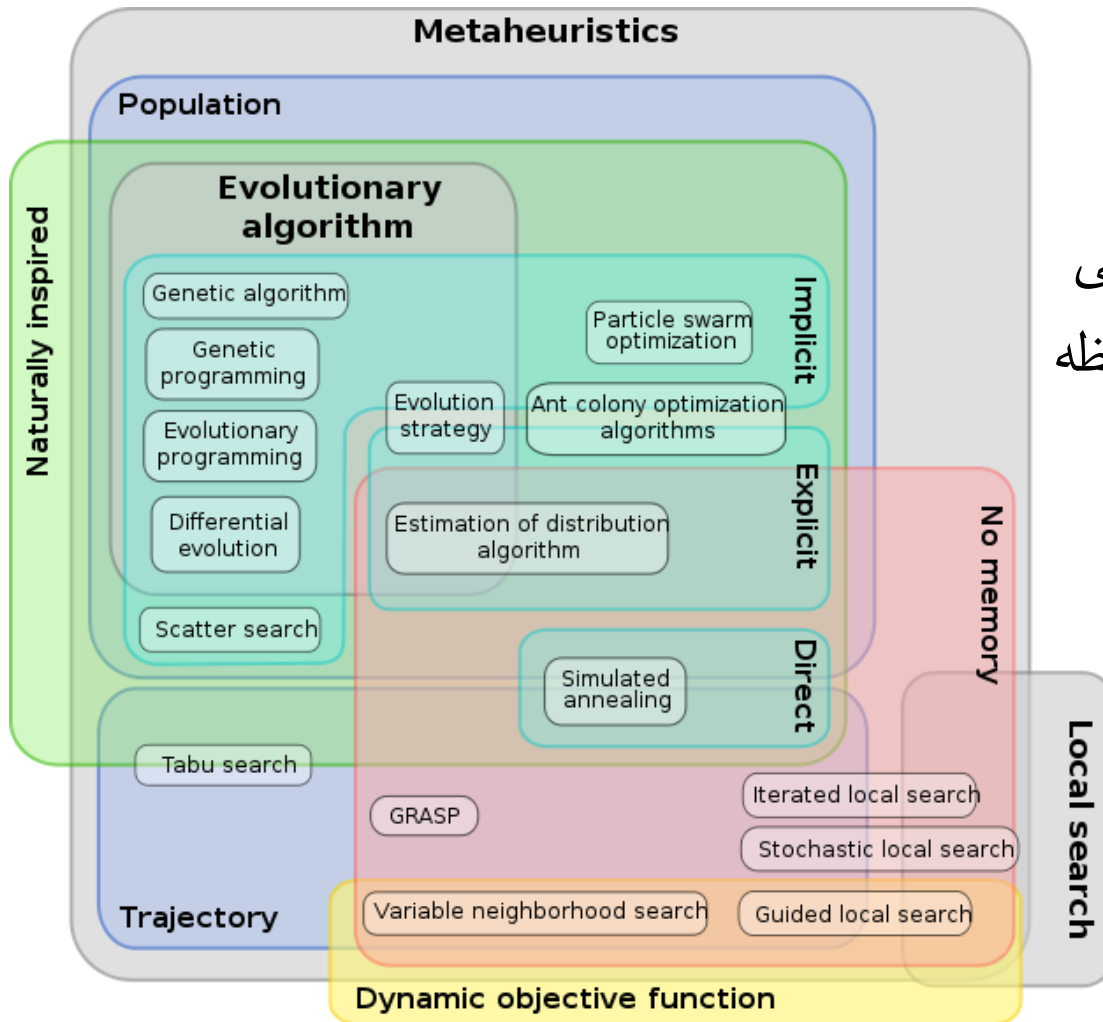
○ جستجوی با بیشترین پویش = جستجوی تصادفی Random Search

○ جستجوی با بیشترین انتفاع = جستجوی تپه نوردی Hill-Climbing Search

پویش و انتفاع



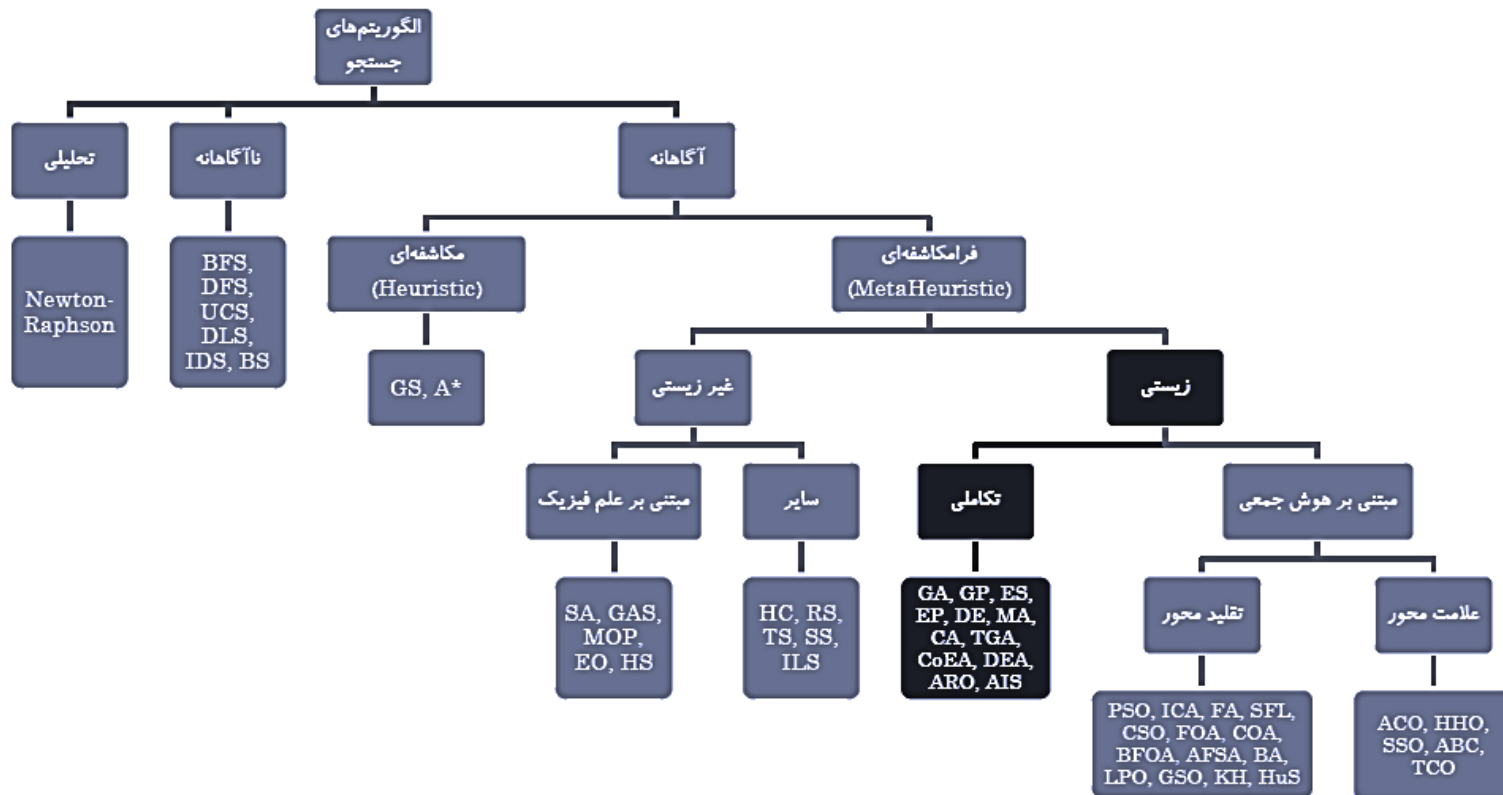
دسته‌بندی الگوریتم‌های فرامکاشفه‌ای

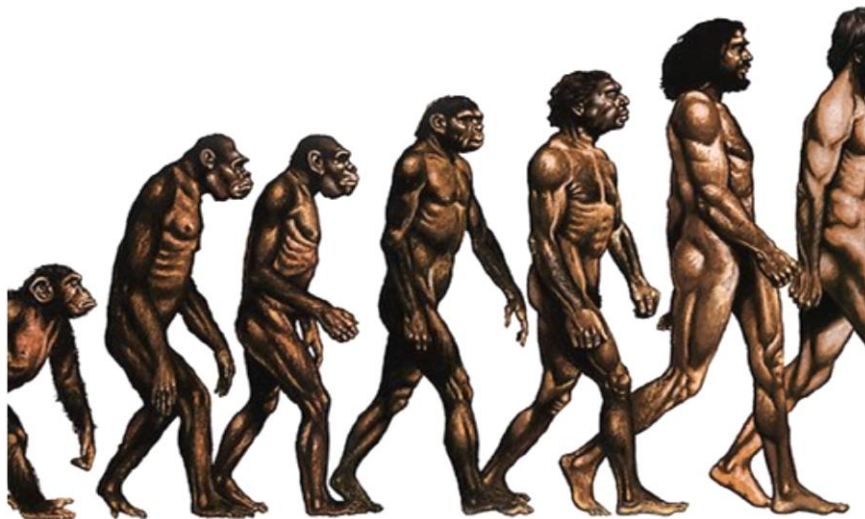


- زیستی و غیرزیستی
- تکاملی و غیرتکاملی
- جمعیتی و غیرجمعیتی
- باحافظه و بدون حافظه
- احتمالی و قطعی

دسته‌بندی انواع روش‌های جستجو

○ جایگاه الگوریتم‌های تکاملی و زیستی





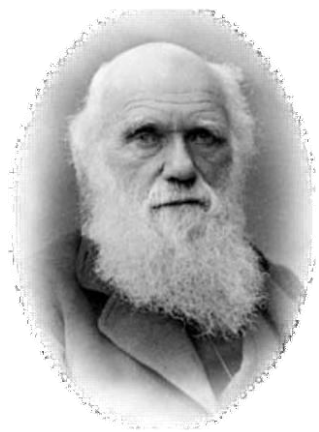
بخش دوم – پردازش تکاملی

نظریه داروین – مراحل الگوریتم تکاملی – شیوه‌های مختلف
پیاده‌سازی عملگرهای تکاملی – مباحث پیشرفته در پردازش تکاملی

نظریه تکامل لامارک

● انتقال موروثی

- موجودات در طول زندگی، با شرایط محیطی تطبیق پیدا میکنند.
 - ویژگی‌های غیرضروری از دست می‌رود.
 - ویژگی‌های جدید جایگزین می‌شوند.
- انطباق‌ها از طریق وراثت به فرزندان انتقال می‌یابد.
- موجودات تمایل به حفظ ویژگی‌های ژنتیکی خویش دارند.



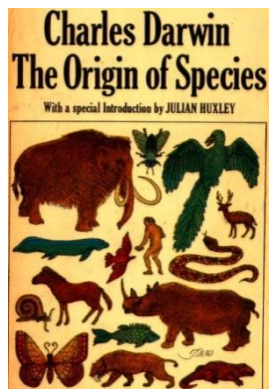
نظریه تکامل داروین

● طبیعت موجودی هوشمند است.

● تمامی موجودات امروزی از نسل موجودات ماقبل تاریخ هستند و صدها میلیون سال از حیات می‌گذرد.

● سابقه و ریشه تمام میلیون‌ها موجود زنده به یک ارگانیسم زنده ساده برمی‌گردد.

● فرایند ایجاد یا تغییر شکل گونه‌های مختلف حیات در اثر یک نیروی هدایت کننده طبیعی به نام انتخاب طبیعی (natural selection) است.



تکامل و انتخاب طبیعی

● انتخاب طبیعی، راز بقای برترین‌ها در طبیعت انتقال خصوصیات برتر به نسل بعد است.

■ تکامل تدریجی جامعه موجودات

■ تنازع بقا: قوی‌تر زنده می‌ماند.

○ نبرد برای زندگی (struggle for the life)

○ بقای اصلح (survival of the fittest)

■ از بین رفتن نمونه‌های ضعیف و زنده ماندن نمونه‌های برتر

انتخاب طبیعی

● مثال هایی از نیروی انتخاب طبیعی

- تکامل سیستم شنوایی و تضعیف سیستم بینایی خفاش به دلیل زندگی در غار
- تکامل سیستم بینایی عقاب به دلیل نیاز به شکار و پرواز در ارتفاعات بالا در کوهستان
- تکامل ویژگیهای استتار جهت پنهان ماندن از چشم صیاد در آفتاب پرست، مارمولک، پروانه، خرس قطبی
- وابستگی شکل آواز پرندگان به زیستگاه آنها
- محیط های جنگلی و پوشش گیاهی انبوه، صداهایی با فرکانس پایین و تحریر فاصله دار
 - تخریب صدای فرکانس بالا توسط محیط جنگل
- علفزارها و زیستگاه های باز، صداهایی با فرکانس بالا و تحریرهای سریع و پی در پی
 - تخریب صدا توسط باد

مفاهیم

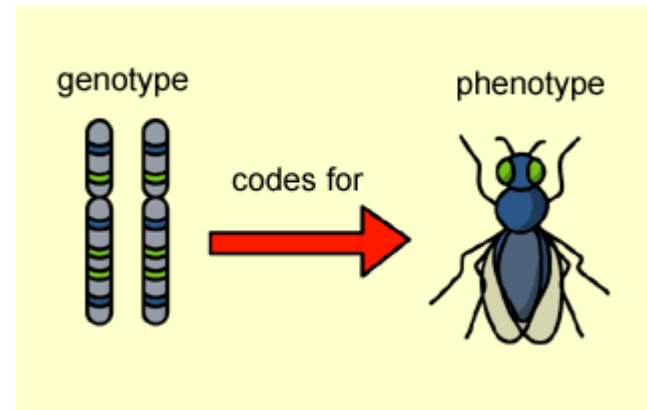
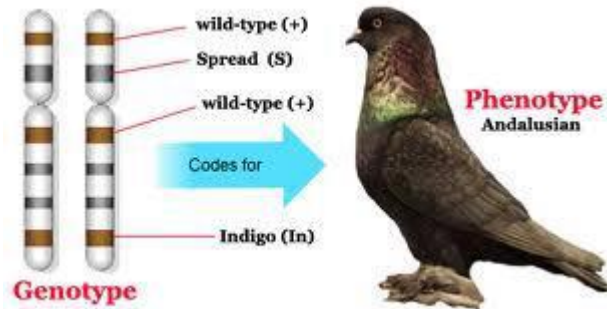
● ژنوتایپ (Genotype) ترکیب تمام ژن ها برای یک فرد مشخص

■ کد ژنتیکی یک موجود

● فنوتایپ (Phenotype) خصوصیات ظاهری یک فرد، حاصل شده از

رمزگشایی یک ژنوتایپ

■ خصوصیات نهایی حاصل از کد ژنتیکی



مفاهیم

● کروموزوم (Chromosome)

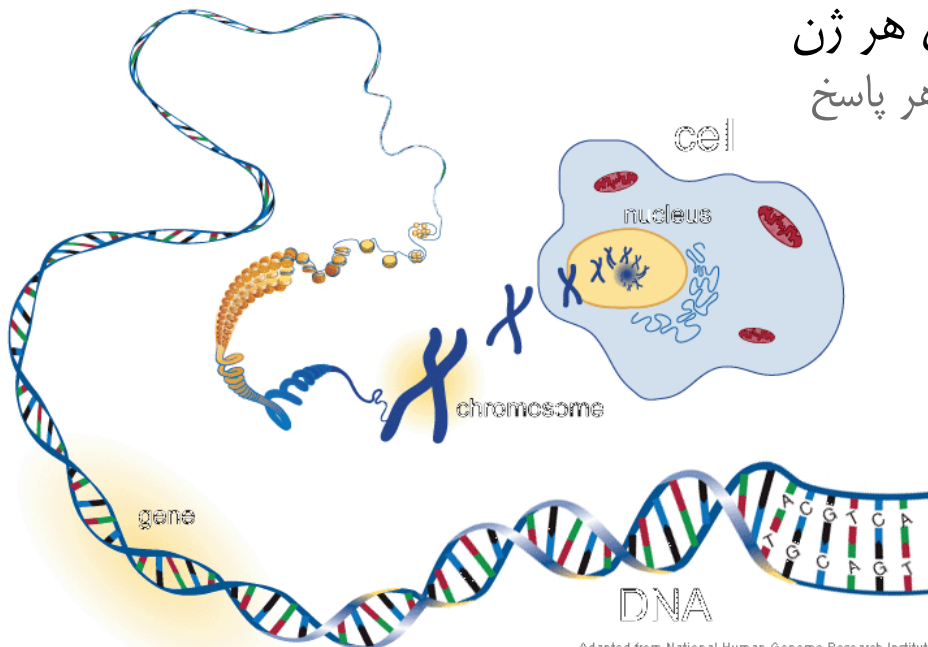
- محل ذخیره سازی اطلاعات ژنی یک موجود
- تشکیل شده از دنباله های DNA

● ژن (Gene)

- کروموزوم از واحدهای کوچک تری به نام ژن تشکیل شده است
- ویژگی (مشخصه) داده ها

● آلل (Allele) مقادیر مجاز برای هر ژن

- مقادیر مجاز برای مشخصه های هر پاسخ



دو مرحله مهم در حل مساله توسط الگوریتم تکاملی

①- کدگذاری

- تبدیل فنوتایپ به ژنوتایپ
- نگاشت از فضای واقعی مساله به فضای حالت
- در ساده‌ترین حالت، یک موجود به وسیله یک ساختار از داده‌ها نمایش داده می‌شود.
 - غالباً رشته باینری؛ اما انواع دیگری نیز وجود دارد: اعداد صحیح، اعداد حقیقی، درخت، ...
 - منشا تفاوت انواع الگوریتم‌های تکاملی

②- ارزیابی

- نگاشت از فضای حالت به مقادیر (قابل اندازه گیری و ترتیبی)
- هر موجود به وسیله تابع برازش (fitness function) ارزیابی می‌گردد.
 - تابع برازش، برای موجودات بهتر، مقدار بیشتری برمی‌گرداند. (متضاد تابع هیوریستیک)
- ارزیابی می‌تواند در فضای ژنوتایپ یا فنوتایپ انجام شود.
 - برای ارزیابی در فضای فنوتایپ، احتیاج به دیکد کردن ژنوتایپ داریم.

انواع الگوریتم های تکاملی

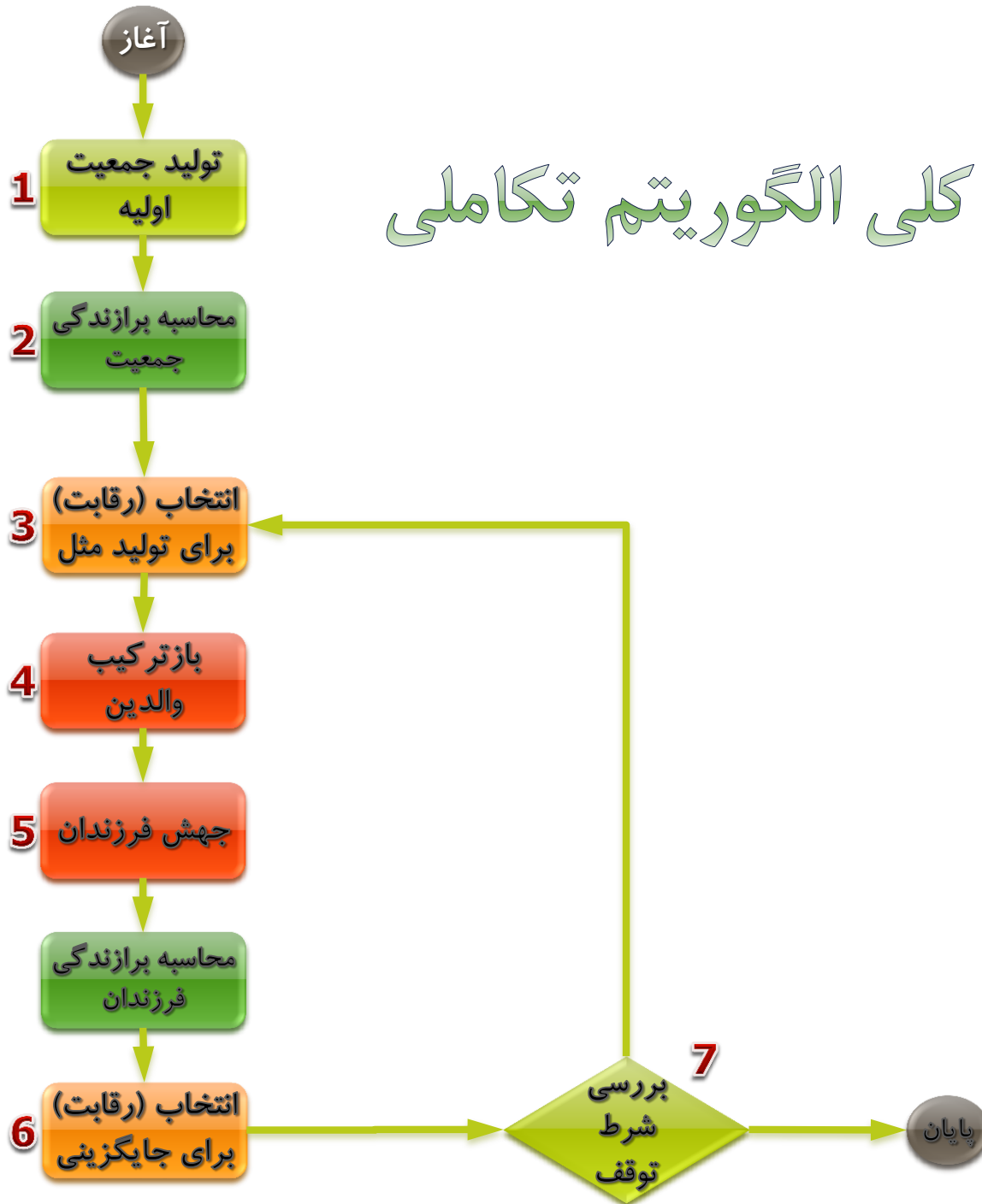
- الگوریتم ژنتیک (Genetic Algorithm)
- برنامه نویسی ژنتیک (Genetic Programming)
- استراتژی تکاملی (Evolutionary Strategy)
- برنامه نویسی تکاملی (Evolutionary Programming)

- تکامل تفاضلی (Differential Evolution)
- الگوریتم ممیتیک (Memetic Algorithm)
- الگوریتم فرهنگی (Cultural Algorithm)
- الگوریتم ژنتیک تاگوچی (Taguchi-Genetic Algorithm)
- الگوریتم هم تکاملی (Co-Evolutionary Algorithm)
- الگوریتم تکاملی دیپلوئیدی (Diploid Evolutionary Algorithm)
- بهینه سازی تولیدمثل غیرجنسی (Asexual Reproduction Optimization)
- سیستم ایمنی مصنوعی (Artificial Immune System)
- و ...

مراحل یک الگوریتم تکاملی

- تولید جمعیت اولیه
- محاسبه برازندگی (ارزیابی) جمعیت ورودی
- انتخاب برای تولید مثل
- باز ترکیب والدین (تولید مثل)
- جهش فرزندان تولید شده
- محاسبه برازندگی (ارزیابی) جمعیت فرزندان
- انتخاب برای جایگزینی
- بررسی شرط توقف

فلوچارت کلی الگوریتم تکاملی



1

جمعیت اولیه

⊙ روش تصادفی: تولید مقادیر تصادفی در بازه مجاز برای هر ژن

■ پوشش یکنواخت فضا

⊙ روش هیوریستیک: تولید کروموزم‌های با برازندگی نسبتاً بالا

■ پوشش بخش‌های مهم فضا

■ نیاز به دانستن اطلاعات کلی درباره مساله و دورنمای فضای حالت – حدس زدن جواب‌های نسبتاً خوب

⊙ اندازه جمعیت معمولاً ثابت است.

■ بر اساس محدودیت منابع، در تئوری داروین

⊙ اندازه جمعیت اولیه، مهم است

■ افزایش اندازه جمعیت اولیه

○ تقویت قابلیت پویش به خاطر پوشش دادن فضای جستجوی بزرگتر

○ تقویت قابلیت انتفاع به خاطر افزایش شانس عملگرهای تولید مثل

○ افزایش بار محاسباتی!

2

تابع برازش

⊙ تابع برازش، یک نگاشت یا شیوه تصویر کردن، از فضای نمایش کروموزم‌ها به یک مقدار عددی

■ مقدار عددی، به طور مستقیم میزان برازندگی یا خوب بودن کروموزم را نشان می‌دهد.

■ مقدار برازندگی، میزان رسیدن به هدف بهینه سازی را نشان می‌دهد و به شدت به کاربرد وابسته است.

■ ممکن است در بعضی مسایل، به سادگی محاسبه نشود.

عملگر انتخاب

⊙ یکی از عملگرهای اصلی در الگوریتم تکاملی

- ارتباط مستقیم به مفهوم بقاء اصلح در نظریه داروین
- هدف: انتخاب راه‌حل‌های برتر

⊙ فشار انتخاب (selective pressure)

- تاکید بر انتخاب راه‌حل‌های بهتر توسط اعمال عملگر انتخاب.
- عملگر با فشار انتخاب زیاد: تنوع در جمعیت سریع کاهش می‌یابد.
 - گیرافتادن در راه‌حل‌های بهینه محلی
 - محدود کردن قابلیت پویش – تقویت قابلیت انتفاع

⊙ انواع عملگرهای انتخاب

- انتخاب تصادفی
- انتخاب نسبی
- انتخاب رتبه‌ای
- انتخاب مسابقه‌ای
- انتخاب برشی

4

عملگرهای الگوریتم ژنتیک

● ترکیب یا تقاطع (crossover) – تولید مثل

■ فرض کنیم دو موجود برای تولید مثل، انتخاب شده‌اند. تولید پسین‌ها (فرزندان) به کمک این دو صورت می‌گیرد.

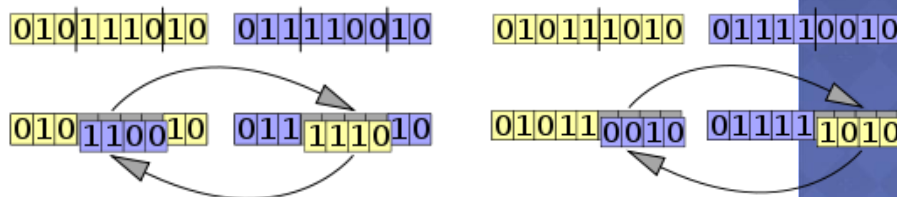
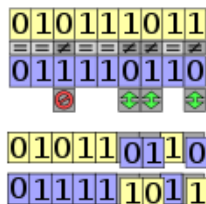
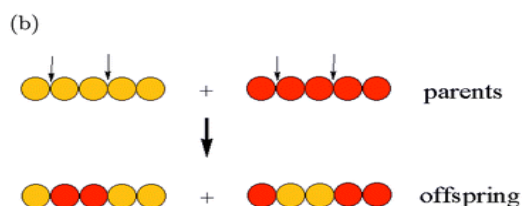
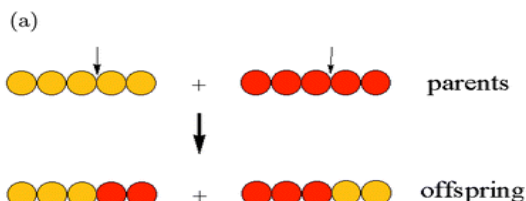
○ مانند تولید مثل در موجودات زنده و طبیعی؛ هر فرزند توسط والدین (دو والد) ایجاد می‌شود.

■ ترکیب یک‌نقطه‌ای (one-point crossover)

■ ترکیب چندنقطه‌ای (multi-point crossover)

■ ترکیب یکنواخت (uniform crossover)

■ ترکیب در حالت نمایش جایگشتی، درختی، عدد صحیح و اعشاری

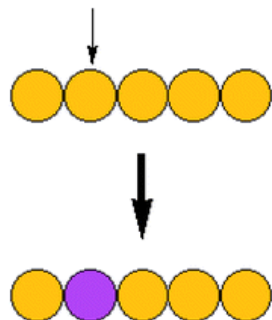


5

عملگرهای الگوریتم ژنتیک

جهش (mutation) – جهش ژنتیکی

■ در طبیعت، همیشه موجودات تمام خصوصیات خود را از والدین به ارث نمی‌برند.

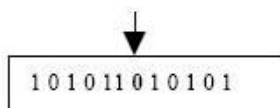


○ تولد نوزادی با چشم‌های رنگی از والدین با چشم‌های مشکی!

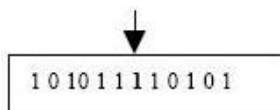
■ انتخاب یک کروموزوم و تغییر یکی از ژن‌های آن به صورت اتفاقی.

○ باعث ایجاد تنوع و پراکندگی در جمعیت.

○ در طبیعت، جهش ژنتیکی امر معمول عادی نیست؛ بنابراین در GA نیز نرخ وقوع این عملگر کم است.



■ جهش در نمایش جایگشتی، اعداد صحیح و اعشاری، درختی،...



6

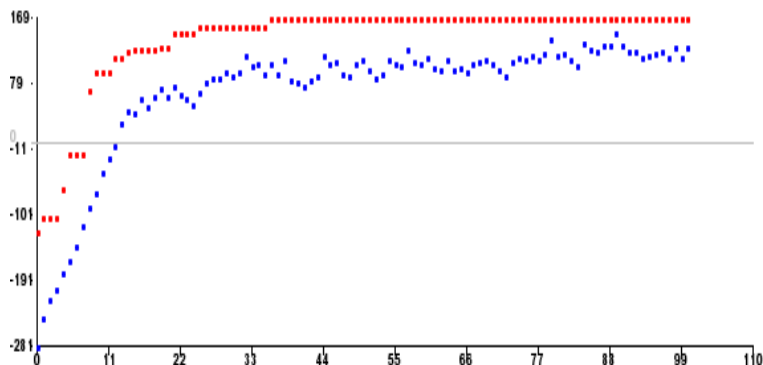
جایگزینی نسل

◎ جایگزینی حالت پایدار (steady state replacement)

- نگهداری بخش بزرگی از جمعیت والدین و جایگزینی بخش کوچکی (P_{rep}) از آن با بهترین فرزندان تولید شده.
- بافت کلی جمعیت و گوناگونی حفظ می شود.
- جلوگیری از همگرا شدن سریع به پاسخ های بهینه.
- هرچه P_{rep} بیشتر باشد، احتمال کاهش گوناگونی بیشتر شده و همگرایی الگوریتم سریعتر خواهد بود.

◎ جایگزینی نسلی (generational replacement)

- جایگزینی کل جمعیت والدین با کل جمعیت فرزندان.
 - احتمال از دست رفتن برترین پاسخ ها.
 - نخبه سالاری (elitism): جایگزینی بهترین عضو از جمعیت والدین به جای ضعیفترین فرزند.
 - روش انتخاب ($\mu + \lambda$): تعداد μ عضو برتر از مجموع μ والد و λ فرزند به نسل بعد منتقل می شوند.
 - روش انتخاب (μ, λ): تعداد μ عضو برتر از λ فرزند به نسل بعد منتقل می شوند. ($\mu < \lambda$)
- احتمال از دست رفتن ناگهانی تنوع جمعیتی



شرایط توقف

رسیدن به موجودی با برآزش مشخص

- در بعضی از مسایل، حد دلخواه تابع ارزیابی (تابع برآزش) قابل محاسبه و تعیین است.
- در این مسایل، شرط خاتمه مناسب، یافتن موجودی در جمعیت نسل فعلی است که مقدار برآزش دلخواه را داشته باشد.

عدم بهبود وضعیت جمعیت در طی نسل‌های گذشته

- ممکن است چند نسل بگذرد و مقدار برآزش بهترین موجود جامعه، تغییری نکند؛ یا اینکه تمام موجودات جامعه، یکسان شده باشند.
- همگرا شدن جمعیت؛ راکد (stagnant) شدن جمعیت.
- در این حالت، تولید نسل‌های بعد احتمالاً مفید نیست: خاتمه الگوریتم.

7

رسیدن تعداد نسل‌ها به حد مشخص

- همیشه الگوریتم تکاملی بعد از رسیدن شماره نسل‌ها به عدد مشخصی، خاتمه می‌یابد. (همیشه مثبت نیست)
- حداکثر تعداد نسل‌ها نباید عدد خیلی کوچکی باشد، چون فرصت پیمایش فضا برای الگوریتم تکاملی وجود نخواهد داشت.

کنترل پویش - انتفاع

● حفظ مصالحه بین پویش و انتفاع

- کاهش فشار انتخاب

- اجتناب از پاسخ‌های بهینه محلی

● اگر در زمان توقف الگوریتم تکاملی، نتایج حاصل شده رضایت‌بخش نباشد، باید با تغییر شرایط، موازنه پویش-انتفاع را تنظیم مجدد نمود.

● سه روش برای کنترل موازنه پویش-انتفاع

- کنترل پارامترهای موثر

- استفاده از توابع مناسب بازتولید

- حفظ تنوع جمعیتی

کنترل پارامترهای موثر در موازنه پویش-انتفاع

● پارامترهای الگوریتم‌های تکاملی که در موازنه موثرند:

- احتمال بازترکیب (P_c)
 - رابطه مستقیم با قابلیت انتفاع
- احتمال جهش (P_m)
 - رابطه مستقیم با قابلیت پویش
- درصد جایگزینی (P_{rep})
 - رابطه مستقیم با قابلیت انتفاع
- درصد اعضای مورد گزینش در انتخاب مسابقه‌ای (P_{tourn})
 - رابطه مستقیم با قابلیت انتفاع
- درصد اعضای مورد بررسی در انتخاب برشی (P_{trunc})
 - رابطه مستقیم با قابلیت پویش

پیاده سازی توابع بازتولید

● انتخاب روش پیاده سازی توابع بازتولید متناسب با ماهیت فضای جستجوی مساله

- چرخ رولت ساده
 - تقویت قابلیت انتفاع
- چرخ رولت با چنداشاره گر
 - تقویت قابلیت پوشش
- بازترکیب یک نقطه‌ای (یا با نقاط شکست کم)
 - تقویت قابلیت انتفاع
- بازترکیب یکنواخت
 - تقویت قابلیت پوشش
- جهش با بیشترین تخریب روی ساختار کروموزوم
 - تقویت قابلیت پوشش
- روش جایگزینی پایدار به جای روش جایگزینی نسلی
 - تقویت قابلیت پوشش

حفظ تنوع جمعیتی

● روشهای حفظ تنوع

■ کرانه سازی (niching): دادن اولویت به تنوع موجودات از طریق جلوگیری از رقابت ناعادلانه پاسخ‌های نامربوط یا غیرمشابه یا با فاصله زیاد

○ مشترک سازی برازش (fitness sharing)

• استفاده از ایده منابع محدود در طبیعت در یک منطقه؛ موجودات زیاد در یک ناحیه باعث می‌شوند برازش کلی آنها کم شود.

• ترغیب الگوریتم تکاملی به پوشش بیشتر فضا و عدم همگرایی زودرس

○ انبوه سازی (crowding)

• جایگزینی اعضای جدید با اعضای مشابه در جمعیت و حفظ تنوع.

• فرزند جدید تولید شده با شبیه‌ترین یا یکی از شبیه‌ترین موجودات نسل قبلی رقابت می‌کند و در صورت بهتر بودن، جایگزین می‌شود.

■ گونه‌سازی (speciation)

○ فقط آن دسته از موجودات که به اندازه کافی به یکدیگر شبیه هستند، حق بازترکیب شدن دارند؛

جلوگیری از ترکیب دو پاسخ با تفاوت‌های بنیادی و بسیار زیاد که معمولاً منجر به تولید پاسخ‌های مهلک (lethal solutions-ضعیف) می‌شود.



بخش سوم - فرامکاشفههای تکاملی

بررسی انواع الگوریتمهای تکاملی

الگوریتم ژنتیک GENETIC ALGORITHM

● الگوریتم ژنتیک استاندارد (Canonical GA - CGA)

■ توسط هالند.

- هالند، پدر الگوریتم‌های ژنتیک
- قبلاً ایده معرفی شده بود. اما به خاطر فعالیت‌های وی، الگوریتم ژنتیک توسعه زیادی یافت.

■ نکات مهم CGA

- استفاده از نمایش رشته بیتی
- طول ثابت و یکسان برای هر کروموزوم
- جمعیت با تعداد اعضای ثابت
- عملگر انتخاب نسبی برای والدین
- بازترکیب تک نقطه‌ای
- بازترکیب در CGA عملگر اصلی است.
- احتمال بازترکیب P_c مقادیر بزرگی است. معمولاً بیشتر از ۰.۹۵.
- جهش به شکل معکوس سازی بیت
- جهش در CGA عملگر فرعی است.
- احتمال جهش P_m مقادیر کوچکی است. مثلاً $1/L$ که L طول کروموزوم است.

شبه کد الگوریتم ژنتیک استاندارد

Function GA(*problem*) **returns** a state that is a local optimum

Input: Populationsize, Problemsize, P_{crossover}, P_{mutation}

Output: S_{best}

Population \leftarrow InitializePopulation(Populationsize, Problemsize);

EvaluatePopulation(Population);

S_{best} \leftarrow GetBestSolution(Population);

while \neg StopCondition() **do**

 Parents \leftarrow SelectParents(Population, Populationsize);

 Children $\leftarrow \emptyset$;

foreach Parent₁, Parent₂ \in Parents **do**

 Child₁, Child₂ \leftarrow Crossover(Parent₁, Parent₂, P_{crossover});

 Children \leftarrow Mutate(Child₁, P_{mutation});

 Children \leftarrow Mutate(Child₂, P_{mutation});

end

 EvaluatePopulation(Children);

 Population \leftarrow Replace(Population, Children);

 S_{best} \leftarrow GetBestSolution(Population);

end

return S_{best};

ویرایش‌های GA

⊙ مبتنی بر نوع استرژری جایگزینی نسل‌ها

- الگوریتم‌های ژنتیک نسلی (Generational GA - GGA)
- الگوریتم‌های ژنتیک پایا (Steady State GA - SSGA)

⊙ شکاف نسلی (Generation Gap)

- میزان همپوشانی بین نسل کنونی و نسل بعدی
 - GGA شکاف نسلی صفر (یا در صورت نخبه‌گرایی، خیلی کم)
 - SSGA شکاف نسلی زیاد

برنامه نویسی ژنتیکی (GENETIC) (PROGRAMMING)

⊙ برنامه نویسی ژنتیکی (GP)

- یک الگوریتم تکاملی استاندارد
- به عنوان یکی از انواع الگوریتم ژنتیکی
- هم GA و هم GP متکی بر تکامل ژنوتایپی با تاکید بر تولید مثل هستند.

⊙ تفاوت GA و GP در شیوه نمایش موجودات است.

- در GP از نمایش درختی استفاده می شود.

⊙ پیشنهاد اولیه GP برای تکامل برنامه های کامپیوتری بوده است!

- قابل استفاده برای تکامل، ایجاد و بهینه سازی: عبارات بولی، مسایل برنامه ریزی، حل معادلات، تولید مفهوم، برنامه نویسی خودکار، شناسایی الگو، طراحی شبکه های عصبی، درخت تصمیم، مسایل بازی، بیوانفورماتیک، داده کاوی و رباتیک.

نکات GP

⊙ خصوصیات مهم GP

- استفاده از نمایش درختی

- طول متغیر برای کروموزوم‌ها

- تنها الگوریتم تکاملی که دارای این ویژگی است!

- تعداد ثابت اعضای جمعیت

- عملگر بازترکیب عملگر اصلی است.

- احتمال بازترکیب P_c مقادیر بزرگ (بیشتر از ۰/۹)

- احتمال برش شاخه‌های غیرپایانی زیاد و احتمال برش شاخه‌های پایانی کم است.

- عملگر جهش عملگر فرعی است.

- احتمال جهش P_m مقادیر کوچک (کمتر از ۰/۰۱)

- استفاده از عملگر ویژه‌ای به نام اصلاح ساختاری (architecture alteration operator)

انتخاب‌های مختلف GP

● جمعیت اولیه

- تولید موجودات به صورت تصادفی با توجه به محدودیت‌های معنایی که می‌تواند ساختار درخت کروموزوم داشته باشد.
- عمق درخت، فاکتور انشعاب، ...

● تابع برازش

- موجودات نماینده یک برنامه یا یک ساختار هستند، بنابراین برای محاسبه برازندگی معمولاً توسط یک برنامه آزمون که آن را در شرایط خاصی اجرا می‌کند، صورت می‌گیرد.

● عملگرهای بازترکیب

- دو روش متداول در GP
 - تولید یک فرزند با ترکیب دو والد
 - تولید دو فرزند با جابجایی زیردرختها

انتخاب‌های مختلف GP

◎ عملگرهای جهش

- جهش گره تابع
 - جهش یک گره داخلی از بین مقادیر تابعی مجاز
- جهش تعویضی
 - تعویض تصادفی پارامترهای یک گره داخلی
- جهش گره پایانی
 - جهش یک گره برگ از بین مقادیر پایانی مجاز
- جهش گوسین
 - جهش مقدار یک گره پایانی به شیوه گوسین
- جهش رشدی
 - جایگزینی یک گره با یک زیردرخت تصادفی با عمق از قبل مشخص
- جهش برشی
 - جایگزینی تصادفی یک گره داخلی با یک مقدار تصادفی پایانی (برگ) – هرس درخت

انتخاب‌های مختلف GP

● عملگرهای اصلاح ساختاری

○ باز ترکیب غیرجنسی؛ اصلاح ساختار درختی یک موجود

■ عملگر جایگشت

○ مشابه جهش تعویضی؛ تولید یک جایگشت تصادفی از $n!$ جایگشت ممکن بین فرزندان یک گره

■ عملگر ویرایش

○ ویرایش ساختار درخت بر اساس قوانین حاکم در دنیا که از قبل تعریف شده‌اند.

○ مثلاً زیردرختی که عبارت $x \text{ AND } x$ را نشان می‌دهد با یک گره با برچسب x جایگزین می‌شود.

■ عملگر بلوک سازنده

○ بلوک‌های سازنده مفید تشخیص داده شده و با یک گره ویژه جایگزین می‌شوند. در این صورت دیگر ساختار آنها برهم نمی‌خورد.

شبیه‌گد برنامه نویسی ژنتیکی - GP

Function GP(*problem*) **returns** a state that is a local optimum

Input: Populationsize, nodesfunc, nodesterm, Pcrossover, Pmutation, Palteration

Output: Sbest

Population \leftarrow InitializePopulation(Populationsize, nodesfunc, nodesterm);

EvaluatePopulation(Population);

Sbest \leftarrow GetBestSolution(Population);

while \neg StopCondition() **do**

Children $\leftarrow \emptyset$;

while Size(Children) < Populationsize **do**

Operator \leftarrow SelectGeneticOperator(Pcrossover, Pmutation, Palteration);

if Operator \equiv CrossoverOperator **then**

Parent1, Parent2 \leftarrow SelectParents(Population, Populationsize);

Child1, Child2 \leftarrow Crossover(Parent1, Parent2);

Children \leftarrow Child1;

Children \leftarrow Child2;

else if Operator \equiv MutationOperator **then**

Parent1 \leftarrow SelectParents(Population, Populationsize);

Child1 \leftarrow Mutate(Parent1);

Children \leftarrow Child1;

else if Operator \equiv AlterationOperator **then**

Parent1 \leftarrow SelectParents(Population, Populationsize);

Child1 \leftarrow AlterArchitecture(Parent1);

Children \leftarrow Child1;

end

end

EvaluatePopulation(Children);

Population \leftarrow Replace(Population, Children);

Sbest \leftarrow GetBestSolution(Population);

end

return Sbest;

استراتژی تکامل (EVOLUTION) (STRATEGY)

◎ استراتژی تکامل ES

- بهینه سازی فرآیند تکامل، همزمان با یافتن موجود بهینه!
- فرایندهای زیست شناسی توسط تکامل بهینه می شوند. تکامل خود یک فرایند زیست شناسی است؛ پس باید بتوان تکامل را نیز بهینه نمود!!
- هر موجود توسط بلوک های سازنده ژنی و همچنین مجموعه ای از پارامترهای استراتژی که رفتار موجودات را مدل می کند، نمایش داده می شود.
- ویژگی های ژنی و پارامترهای استراتژی همزمان تکامل می یابند.
- تکامل ویژگی های ژنی توسط پارامترهای استراتژی کنترل می شود.

نکات استراتژی تکامل ES

● خصوصیات ES

- استفاده از نمایش اعداد حقیقی ممیز شناور
- جمعیت با تعداد اعضای ثابت
- پارامترهای استراتژی همراه هر کدام از موجودات است تا شیوه و جهت جستجو در هر مرحله برای آن تعیین شود.
- عملگر جهش در ES عملگر اصلی است.
- در ES، باز ترکیب وجود ندارد. هرچند در نسخه‌هایی از باز ترکیب هم استفاده شده است؛ که به ماهیت ES لطمه زده و از آن دور می‌شود.
- الگوریتم عملگر جهش و پارامترهای آن نیز طی فرایند استراتژی تکامل، تکامل می‌یابند.
- نرخ جهش با توجه به برازش موجودات، به صورت تطبیقی تنظیم می‌شود.
 - پاسخ برازنده، نرخ جهش کم و پاسخ با برازندگی کم، نرخ جهش بالا
- نرخ جهش با توجه به موفقیت بهبود پاسخ، تنظیم می‌شود.
- اگر درصد موفقیت بیشتر از ۲۰٪ بود، نرخ جهش افزایش و در غیر اینصورت کاهش می‌یابد.
- احتمال جهش با توجه به یک تابع توزیع احتمال گوسی محاسبه می‌شود.
- جایگزینی انتخاب $(\mu + \lambda)$
- جایگزینی انتخاب (μ, λ)
- $1 \leq \mu \leq \lambda$

شبهه کد استراتژی تکامل - ES

Function ES(*problem*) **returns** a state that is a local optimum

Input: μ , λ , ProblemSize

Output: S_{best}

Population \leftarrow InitializePopulation(μ , ProblemSize);

EvaluatePopulation(Population);

$S_{best} \leftarrow$ GetBestSolution(Population);

while \neg StopCondition() **do**

 Children $\leftarrow \emptyset$;

for $i = 0$ to λ **do**

$P_i \leftarrow$ GetParent(Population, i);

$S_i \leftarrow \emptyset$;

$S_{iproblem} \leftarrow$ Mutate($P_{iproblem}$, $P_{istrategy}$);

$S_{istrategy} \leftarrow$ Mutate($P_{istrategy}$);

 Children $\leftarrow S_i$;

end

 EvaluatePopulation(Children);

 Population \leftarrow Replace(Population, Children, μ);

$S_{best} \leftarrow$ GetBestSolution(Population);

end

return S_{best} ;

برنامه نویسی تکاملی (EVOLUTIONARY PROGRAMMING)

● ایده اصلی برنامه نویسی تکاملی EP

- هوش، ویژگی‌ای که به سیستم اجازه می‌دهد تا رفتار خود را برای رسیدن به هدف خاص، در محیط‌های مختلف، تطبیق دهد!
 - تقلید از تکامل خصیصه‌های رفتاری
 - توسعه مدل‌های رفتاری، نه الزاماً توسعه مدل‌های ژنی
- شیوه نمایش اولیه، ماشین‌های حالت متناهی
 - البته روش‌های دیگر نیز استفاده شده است.
- کاربردهای وسیع در مسایل دنیای واقعی
 - زمان‌بندی، مسیریابی، طراحی ساختارها، رباتیک، پردازش تصویر، ...

نکات برنامه نویسی تکاملی EP

● خصوصیات EP

- استفاده از نمایش نمادی – برای حالت‌ها و انتقال‌های ماشین‌های حالت متناهی
- تعداد ثابت اعضای جمعیت
- عملگر جهش در EP عملگر اصلی است.
 - باز ترکیب وجود ندارد.
 - شیوه عملکرد و پارامترهای عملگر جهش طی فرایند الگوریتم تکامل، تکامل می‌یابند.
 - استفاده از حالت‌ها و انتقال‌های جدید در ماشین‌های متناهی
 - پارامترهای جهش (گام و نرخ جهش) در هر موجود نگهداری می‌شود.
 - خودتطبیقی بودن؛ هر موجود بهترین شرایط خودش را تعیین می‌کند.
 - تنظیم پارامترهای جهش با توجه به برازش پاسخ‌ها به صورت تطبیقی
 - برای پاسخ‌های برازنده گام جهش کوتاه و برای پاسخ‌های نامطلوب، گام جهش بلند
 - احتمال جهش توسط تابع توزیع احتمال گوسی محاسبه می‌شود.
 - هر والد یک فرزند را از طریق عملگر جهش ایجاد می‌کند.
 - روش انتخاب برای جایگزینی، رقابت بین والدین و فرزندان است.

شبهه کد برنامه نویسی تکاملی - EP

Function EP(*problem*) **returns** a state that is a local maximum

Input: Populationsize, ProblemSize, BoutSize

Output: Sbest

Population \leftarrow InitializePopulation(Populationsize, ProblemSize);

EvaluatePopulation(Population);

Sbest \leftarrow GetBestSolution(Population);

while \neg StopCondition() **do**

Children $\leftarrow \emptyset$;

foreach Parent_i \in Population **do**

P_i \leftarrow GetParent(Population, i);

S_i $\leftarrow \emptyset$;

S_iproblem \leftarrow Mutate(P_iproblem, P_istrategy);

S_istrategy \leftarrow Mutate(P_istrategy);

Children \leftarrow S_i;

end

EvaluatePopulation(Children);

Union \leftarrow Population + Children;

foreach S_i \in Union **do**

for 1 to BoutSize **do**

S_j \leftarrow RandomSelection(Union);

if Fitness(S_i) > Fitness(S_j) **then**

S_iwins \leftarrow S_iwins + 1;

end

end

end

Population \leftarrow SelectBestByWins(Union, Populationsize);

Sbest \leftarrow GetBestSolution(Population);

end

return Sbest;

الگوریتم‌های ممیک (MEMETIC) (ALGORITHM)

● انتقال فرهنگ و رفتار از طریق مِم (meme)

- عنصر فرهنگی یا رفتاری که به وسیله عوامل غیرژنی منتقل می‌شود.
- خصایص رفتاری که در طول زندگی یک موجود از طریق تجربه و تقلید فراگرفته شده است.
- بخشی از تمدن که ژنها در به ارث رسیدنشان نقش ندارند.
- رفتارها، مد، علوم و دانش، ادبیات، موسیقی، ...
- شباهت ژن و مِم:
- ژن‌ها از کروموزومی به کروموزوم دیگر انتقال می‌یابند، و مِم‌ها از مغزی به مغز دیگر انتقال می‌یابند.
- در تکامل، بهترین ژن‌ها و بهترین مِم‌ها باقی می‌مانند.
- تفاوت ژن و مِم:
- ژن‌ها مقادیر از قبل مشخصی دارند (از والدین) اما مِم‌ها قبل از زندگی قابل تعیین نیستند.
- ژن‌ها در طول زندگی ثابت هستند، اما مِم‌ها در طول دوره زندگی تغییر می‌کنند.
- مِم‌ها اصلاح را در یک دوره زندگی میسر می‌کنند، اما ژن‌ها این کار را در زمان طولانی انجام می‌دهند.

شبهه کد الگوریتم ممیتیک - MA

```
Function MA(problem) returns a state that is a local optimum  
Input: Populationsize, Problemsize, Pcrossover, Pmutation, MemePopsize  
Output: Sbest  
  
Population  $\leftarrow$  InitializePopulation(Populationsize, Problemsize);  
EvaluatePopulation(Population);  
Sbest  $\leftarrow$  GetBestSolution(Population);  
while  $\neg$  StopCondition() do  
    Parents  $\leftarrow$  SelectParents(Population, Populationsize);  
    Children  $\leftarrow \emptyset$ ;  
    foreach Parent1, Parent2  $\in$  Parents do  
        Child1, Child2  $\leftarrow$  Crossover(Parent1, Parent2, Pcrossover);  
        Children  $\leftarrow$  Mutate(Child1, Pmutation);  
        Children  $\leftarrow$  Mutate(Child2, Pmutation);  
    end  
    EvaluatePopulation(Children);  
  
    MemeticPopulation  $\leftarrow$  SelectMemeticPopulation(Children, MemePopsize);  
    foreach Si  $\in$  MemeticPopulation do  
        Si  $\leftarrow$  LocalSearch(Si);  
    end  
  
    Population  $\leftarrow$  Replace(Population, Children);  
    Sbest  $\leftarrow$  GetBestSolution(Population);  
end  
return Sbest;
```

نکاتی درباره MA

⊙ اصالت وجودی مم از ژن بیشتر است.

▪ ژن پس از مرگ موجود از بین می‌رود، اما مم باقی می‌ماند.

⊙ دو شیوه مهم الگوریتم MA

▪ روش لامارکی (Lamarckian MA)

○ در قسمت جستجوی محلی، اگر کروموزوم y از x بهتر باشد، جایگزین آن می‌شود.

▪ روش بالدوینی (Baldwinian MA)

○ در قسمت جستجوی محلی، اگر کروموزوم y از x بهتر باشد، فقط مقدار برازندگی آن جایگزین مقدار برازندگی x می‌شود.

⊙ روش لامارکی سرعت بیشتر و روش بالدوینی پوش بیشتری دارند.

الگوریتم‌های فرهنگی (CULTURAL) (ALGORITHM)

⊙ بهبود جوامع فقط تابع تکامل ژنتیکی نیست، بلکه تابع تکامل فرهنگی نیز هست!

⊙ تعریف فرهنگ:

- مفاهیم و نمادهایی که بصورت اجتماعی و تاریخی میان گروه‌ها منتشر می‌شود.
- ذخیره‌سازی دانش، تجربه، باورها، ارزش‌ها، افکار، مهارت‌ها، ... در گروهی از مردم در طول یک نسل که حاصل تلاش گروهی و فردی افراد است.
- رفتارهای یادگرفته شده از مردم؛ سنت
- برنامه ریزی اجتماعی از تمایلات فکری که رفتارهای مردم را هدایت می‌کند.

الگوریتم‌های فرهنگی (CULTURAL) (ALGORITHM)

● الگوریتم CA یک سیستم دو-وراثتی است.



■ دارای دو فضای جستجو است.

○ فضای جمعیت: تکامل ژنتیکی

○ فضای باور: نمایش مولفه‌های فرهنگی

■ هر دو فضا تکامل می‌یابند.

■ ارتباط دو فضا

○ فضای باور، مدل سازی اطلاعات فرهنگی جمعیت

○ فضای جمعیت، تبعیت از باورها

■ در هر تکرار، موجوداتی از جمعیت فعلی برای تاثیر گذاردن روی باورها، پذیرش می‌شوند.

■ سپس تجربه موجودات پذیرفته شده برای تنظیم باورها مورد استفاده قرار می‌گیرد.

■ در ادامه، از باورهای تنظیم شده برای تاثیر بر روی جمعیت استفاده می‌شود.

شبیه‌کد الگوریتم‌های فرهنگی - CA

Function CA(*problem*) **returns** a state that is a local optimum

Input: Problemsize, Populationnum

Output: Sbest

Population \leftarrow InitializePopulation(Problemsize, Populationnum);

Belief Space \leftarrow Initialize BeliefSpace (Problemsize, Populationnum);

Sbest \leftarrow GetBestSolution(Population);

while \neg StopCondition() **do**

 EvaluatePopulation(Population);

 Children \leftarrow ReproduceWithInfluence(Population, Belief Space);

 Population \leftarrow Select(Children, Population);

 Beliefcandidate \leftarrow AcceptBelief(Population);

 UpdateBelief Space(Belief Space, Beliefcandidate);

 Sbest \leftarrow GetBestSolution(Population);

end

return Sbest;

الڱورٽم هم ٽڪاملى

(CO-EVOLUTIONARY ALGORITHM)

الگوریتم تکاملی دیپلوئیدی (DIPLOID) (EVOLUTIONARY ALGORITHM)

بهینه سازی تولید مثل غیر جنسی (ASEXUAL) (REPRODUCTION OPTIMIZATION)

سیستم ایمنی مصنوعی (ARTIFICIAL (IMMUNE SYSTEM