Subject:

Year. Month. Date. (1)

پردازش های تکاملی

گردش مشکی
۰۹۱۲-۲۱۲۹۴۰۴

* استاد:

• محمدمهدی عبادزاده

• بتن - سخت افزار ← پردازش مصنوعی ← شبیه سازی مغز

• زمینه فعالیت: پردازش محاسباتی ← شبکه های عصبی

پردازش تکاملی ← محاسبات زیستی

سیستم های فازی

• محاسبات زیستی ← الهام گرفته از طبیعت ← شبکه های عصبی

فازی

Bicomputing

ایمنی مصنوعی

SA - ACO - PSO : جستجوهای ملهم از طبیعت

پردازش تکاملی

• IEEE ← انجمن پردازش محاسباتی

Neural Network

Bicomputing (Transaction)

Fuzzy

• Bicomputing : استفاده از روشهای ملهم از طبیعت برای کامپیوتر

Bioinformatics : استفاده از کامپیوتر در خدمت طبیعت

* مخففات این ادس ← PSO

ACO (بهینه سازی کلنی - خوشه بندی - بهینه سازی پیوسته)

SA

AIS (ایمنی مصنوعی)

ارائه

Subject:

Year. Month. Date. ( )

MA (الگوریتم‌های تکاملی)

DE (الگوریتم‌های تکاملی تفاضلی)

EDA (الگوریتم‌های تخمین توزیع)

QEA (الگوریتم‌های تکاملی کوانتومی)

...

* مباحث درس:

۱- مقدمه‌ای تکامل ← مالکولسلولیپ (داروین)

میکروسلولیپ (مقدمه داروین)

۲- معرفی الگوریتم‌های تکاملی به صورت عمومی

۳- استراتژی‌های تکاملی (ES)

۴- برنامه‌ریزی تکاملی (EP)

۵- الگوریتم ژنتیک (GA)

۶- برنامه‌ریزی ژنتیک (GP)

۷- سیستم‌های طبقه‌بندی (CS)

EA

۸- الگوریتم‌های تکاملی چند هدفه‌ای

۹- الگوریتم‌های تکاملی چندوجهی

۱۰- الگوریتم‌های تکاملی با محدودیت

* بهینه سازی

• یک تابع هدف ساخته شود و آنرا بهینه کرد.

↑ تابع سود ←
↓ تابع هزینه ←

• معمولاً مسئله مینیمم در نظر گرفته می شود. ←

Min [f(x)]

تابع ماکزیمم f(x) ←
مینیمم ← $\frac{1}{f(x)}$ ، $-f(x)$

• انواع محدودیت ← قید ← $h_i(x) = 0 \quad i: 1 \dots m_1$
حد ← $g_i(x) > 0$ یا $g_i(x) < 0$

• چند هدف

$$\text{Min } [f_1(x_1, \dots, x_{n_1}), \dots, f_m(x_1, \dots, x_{n_2})]$$

• فرم کلی

$$\text{Min } [f_1(x_1, \dots, x_n), \dots, f_m(x_1, \dots, x_n)]$$

$$g_i(x) \geq 0$$

$$h_j(x) = 0$$

* ارزیابی

• ۲ × پروژه ← نوشتن لتکی
• ۱ × مقاله ← ارائه شفاهی + پیاده سازی + نوشتن لتکی
• ۱ × امتحان ←
• هر هفته × تمرین ← سئوال و جواب از مسائل مطرح در کلاس

Subject:

Year. Month. Date. ( )

* نگارش

۱- چکیده ← معرفی مسئله
← چگونه حل شده است
← چه نتایجی بدست آمده
← خواننده را به خواندن مشتاق کند ← نوآوری
← نتایج جدید
← درشت نمایی

۲- مقدمه ← معرفی مسئله با جزئیات
← تاریخچه کارهای انجام شده ← حداقل کارهای جدید
← بهترین کارها
← معایب و مزایا هر روش

۳- تئوری و پیاده سازی ← معرفی تئوری به اندازه کافی ← توان تئوری آن پیاده شود
← توضیح کد لازم نیست
← ساختمان داده خاص ✓
← روش ابتکاری ✓

۴- آزمایشات و نتایج ← ۷۰٪ برگه

۵- نتیجه گیری کلی ← نتیجه کلی و ترکیبی، به همراه توجیه و ارجاع، خوبی یا بدی روش، حسنات

۶- منابع

* آزمایشات

- جدول معرفی آزمایش
- تیتر جدول = هدف آزمایش
- محتویات جدول = پارامترهای ثابت + نکات خاص

Subject:

Year. Month. Date. (3)

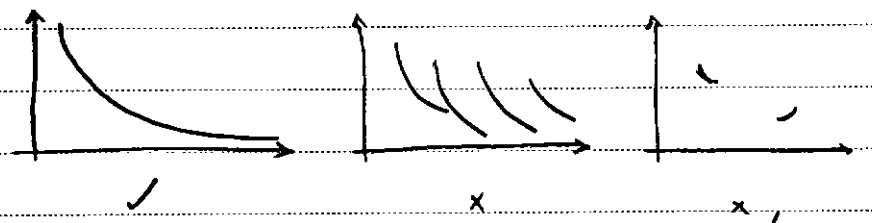
• نمایش نتایج ← متنی ✓ نمودار ✓

تصویر ✓ جدول ✗

نتایج باید روشن و واضح باشد و به شکل مرتبط باشد

شکل مشخص باشد.

☑ اگر تندی زیاد باشد، کارایی کم می‌شود.



✓ ✗ ✗

• تحلیل نتیجه ← توجیه آنکه چرا بدست آمده است.

ریاضی، منطقی، ...

• برآورد مالیش:

| جدول + نمودار + نتیجه + تحلیل |

• آزمایشها

تغییر پارامتر ۱

ترکیب پارامتر ۲

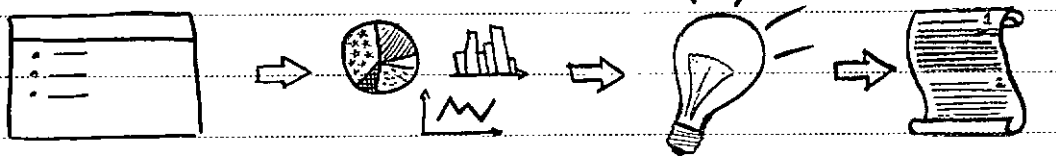
معیار کارایی

• معیار کارایی ← ارزیابی الگوریتم ← سرعت

دقت

تنوع

تعداد جواب



Subject:

Year . Month . Date . ( )

* تاریخچه :

1948 .

تورینگ پیشنهاد داده است که جستجوی تکاملی به رهم دیابد

1962 .

پیاده سازی شده است .

1960's .

ES : 64 —

EP : 65 —

GA : 75 —

1990 .

آن به بعد تمام روشهای تکاملی را تحت مجموعه EA قرار دادند . (در اولین کنفرانس PPSN)

1992 .

GP ← توسط koza ← برای جستجو مناسب است

* کنفرانس ها و مجلات

- ICGA : Intl. Conf. of Genetic Algorithm
- CEC
- PPSN
- „Evolutionary Computation" : MIT Press
- Evo Net : Network

معتبرترین ها : CEC ، PPSN ، GECCO .

- „Evolutionary Computation" : IEEE Transaction
- „Genetic Programming and Evolvable Machines"

Subject:

Year. Month. Date. 4

* آخه چرا؟

• "No Free lunch" ← می‌توان الگوریتمی ابداع کرد که تمام مسائل را به طور بهینه حل کند.

• برای پاسخ به نظریه بالا، الگوریتم‌های تکاملی ابداع شد ← الگوریتم عمومی
- بهینه تابل قبول
- جواب تابل قبول
- کلیه مسائل

• اثر الگوریتم‌های خوب از طبیعت الهام گرفته است.

* مزایا

• عمومی بودن ← مستقل از مسئله
• بهینه تابل قبول
• تابل تفسیر بودن جواب‌ها
• پیشنهاد چندین جواب به صورت همزمان ← در هر لحظه جمعیتی از جواب‌ها را دارد.
• مداری بودن الگوریتم

* معایب

• عدم ضمانت همگرایی به جواب بهینه در زمان محدود ← اثبات ریاضی ندارد و در بهترین حالت به جواب بهینه همگرا می‌شود.
• ضعف تئوری ← پیچیده است و درک آن زمان‌بر است
• نیاز به تنظیم پارامترها ← سرعت و همگرایی نسبت به پارامترها حساس است.
• کندبودن
• با تغییر پارامتر، الگوریتم باید از ابتدا اجرا شود ← چون هر نسل از نسل قبل به دست می‌آید، بجز GP

* مسائل مناسب

• مسائل پیچیده با تعداد زیاد پارامتر ← برخلاف روش‌های محاسباتی ریاضی

Subject:

Year. Month. Date. ( )

- مسائل دارای روابط پیچیده بین پارامترها ← با correlation
- مسائل با پارامترها از انواع متفاوت ← type های متفاوت
- مسائل با بهینه‌های محلی ← برخلاف دشتهای رادیان
- مسائل چند هدفی
- نویز در داده‌ها
- متغیر بودن شرایط مسئله ← بهینه مسئله در حال تغییر است، تعداد نسل‌های تولید شده باید متناسب با سرعت تغییر مسئله باشد.

* تئوری تکامل

- Search of Optimal forms (environment)
- Based On:
  - Assortment → بر مبنای ترکیبی مواد و ترتیب
  - Randomness
  - Selection → survival of the fittest برای بهترین‌ها

- اصول تئوری داروین: تنوع، انتخاب، تصادف
- موجودات برای زنده ماندن باید خود را با محیط تطابق دهند تا از منابع بهترین استفاده را بکنند تا زنده بمانند و تولید مثل کنند.
- علت تکامل ⇐ محدودیت منابع ← استفاده بهینه از منابع
  - پیدا کردن منابع
  - رقابت بر سر منابع
  - ☑ کالبدی
  - ☑ رفتاری
  - ☑ دستیابی جفت بالقوه
- در انتخاب، اصل «ماندن تنها شایسته‌ترین‌ها» نهفته است.
- تنوع برای تولید تمام حالات و در نتیجه بهترین حالت است.
- ☑ باید حداقل یک راه‌حل تولید و وجود داشته باشد که قدرتمند باشد.
- اصل تکامل بر اساس امیدوار بودن قرار گرفتن ویژگی‌های خوب پدر و مادر در کنار یکدیگر و تولید فرزند بهتر است.

حدیث: فرزند صالح یکی از خوبی‌های بهشت است.

Subject:

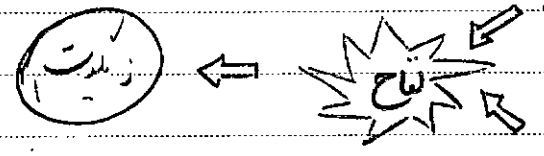
Year. Month. Date. 5)

- تقاوت های تولید دژنلیهای جدید است. هدف آن انتخاب نوع است. جهشهای بزرگ دارند دژنلیهای جدید پدید می آید.
- طبیعت با جهش مخالف است و سعی می کند جهش را از بین ببرد. ولی انتخاب نیاز (ضطراری به آن باشد
  - گذار از عصر دایناسورها به عصر پستانداران

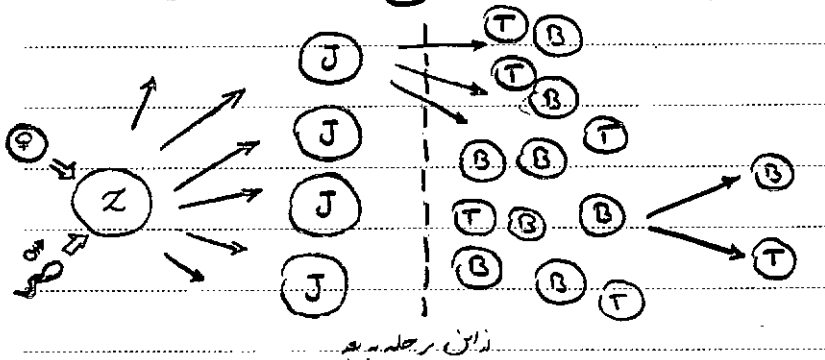
- محدودیت منابع ← تقابت و انتخاب ← تابول تنهای شایسته ترها
تقابل: سازش کنار هم قرارگرفتن دژنلیهای خوب پدر و مادر و تولید فرزند شایسته تر

* دید میکرو سلولی پیش

- سلول جنسی ماده ← تخمک ← ماهیت ماده
- سلول جنسی نر ← اسپرم ← ماهیت نر



- زایندگی ← تقسیم سلولی ← سلولهای بنیادی جنینی (قابل تبدیل به هر نوع سلولی)
- سلولهای بنیادی ← تبدیل ← سلولهای بنیادی بالغ (قابل تبدیل به سلول تخصصی خاص)
  ← سلولهای تخصصی
- سلولهای بنیادی بالغ ← کاملاً تخصصی ← قلب، مغز، مغز استخوان
  ← نیمه تخصصی ← قابل تبدیل ← برخی گروههای سلولهای تخصصی در مغز استخوان
  ← تقسیم سلولی ← سلول بنیادی بالغ + سلول تخصصی



Subject:

Year. Month. Date. ( )

• تقسیم سلولی ← غیر جنسی
↙ جنسی

* سلول

• کوچکترین واحد زنده موجود زنده

• سلول = هسته + غشاء سلولی + اندامک ها

• سلول ← هسته ← ژنوم ← کروموزوم ← DNA ← ژن ها

حاوی بخشی از اطلاعات ژنتیکی

مجموعه‌ای از کروموزوم‌ها

• کروموزوم‌ها ← پلی‌نوکلئیدی : یک رشته‌ای ( )

لوکوس ژنی

دیپلوئیدی : دو رشته‌ای ( )

جایگاه‌ها مشابه

اطلاعات کدشده می‌تواند متفاوت باشد

دو جایگاه یک ژنی ← ژن خالص

دو ژنی متفاوت ← غالب و مغلوب

جایگاه‌های مشابه ولی نه احیاناً لزوماً یکسان

جایگاه ژن چشم

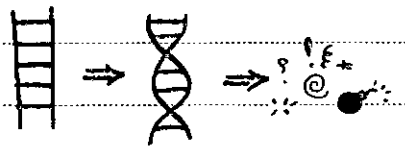
آبی / سبز

• کروموزوم دیپلوئیدی می‌تواند تا مدت‌ها ژن مغلوب را نگاه دارد و وقتی شرایط برای غالب شدن آن مناسب شود، بروز می‌کند ← تنوع بیشتر

Subject:

Year. Month. Date. (6)

• تمایش کروموزوم ← خطی (نردبانی)



پیچی

سه بعدی

چهار بعدی

• انسان ← ۲۳ جفت کروموزوم = ژنوم

کروموزوم ۲۳ ← در زنهای انسان (XX)

در مردها متفاوت (XY)

مربوط به تعیین جنسیت

ژنوم سلول دیپلوئیدی است

سلول های جنسی haploidی است ← تخمک : X

اسپرم : X یا Y

• ژن ← خصوصیات فیزیکی موجودات

ژنوتیپ ← کد ژنتیکی ← رشته ای از الفبا

فنوتیپ ← ویژگی فیزیکی

بین ژنوتیپ و فنوتیپ تناظر وجود دارد

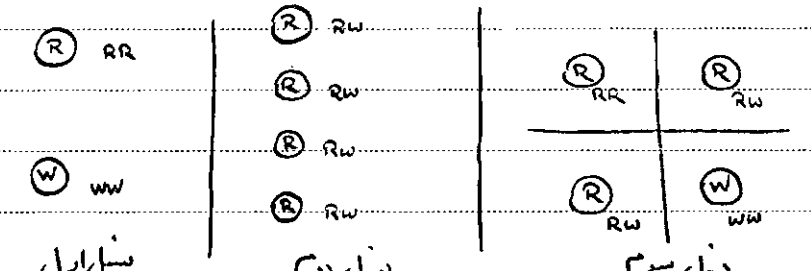
تناظر یک به یک ← الل های تکاملی

تناظر چند به چند ← موجودات زنده

آلل ← مجموعه مقادیر ممکن هر ژن ← هر مقدار یک ژنوتیپ

لوکوس ← جایگاه ژن در رشته DNA

آزمایش مندل ← نمایش غالب و مغلوب ← آزمایش آبادی



اگر نسل دوم تماماً قرمز باشد، آنگاه می فهمیم که نسل اول خالص بوده است.

Subject:

Year. Month. Date. ( )

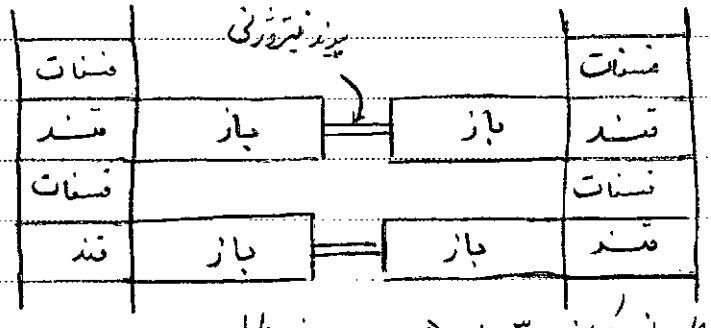
موجودات زنده
↓
ارگانها
↓
سلول
↙ ↘
هسته — پروتئین
کروموزومها — آمینواسید
↓ ↑
رشته DNA
↓
ژنها
↓
بازهای شیمیایی

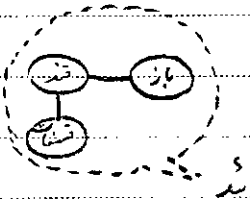
- آمینواسید = مجموعه‌ای ۳ تایی از النها (بازهای شیمیایی)
- بازهای شیمیایی = النها ← شامل ۴ نوع (T, A, C, G)

* DNA

Di oxy ribo nucleic acid
قند — هسته — اسید

- پرتون طول ثابتی ندارد و شروع و پایان دارد. دنباله ژنوم انسان دارای ۳۲۵۰۰۰۰۰ جفت باز است.
- شامل دو رشته طویل و پله‌ای نردبان
- یک مونومر از DNA، بخشی از پلیمر DNA است و متشکل است از:
  - * باز ← لوله‌ای ژنتیکی ← قسمت عرضی
  - * فسفات ← قسمت طولی
  - * قند





نوکلئوتید

- دارای جهت می‌باشد ← ارتباط از کربن ۳ یا ۵ ← پیوند طولی

Subject:

Year. Month. Date. (7)

• ۵ نوع باز

آدنین (A)
گوانین (G) } بازهای پورین
سیتوزین (C)
تیمین (T) } بازهای پریمیدین
یوراسیل (U) ← در RNA به جای T در DNA می‌نشیند

• پیوند ← از نوع هیدروژنی در بازها ← A=T و G≡C ← بین دو باز مکمل
در نردبان ← تعداد ـ نسبتها

• رشته ← آمینواسید = ۳ باز
دارای کد شروع و کد پایان
زنجیره‌ای از آمینواسیدها = پروتئین
ارتباط از طریق پیوندهای مکمل: کربن ۳ ⟺ کربن ۵

• پروتئین ← مایع ← آنزیم ← کاتالیزور واکنشهای شیمیایی
اجزای تشکیل دهنده سلول ← ژنهای نیز یکی محدود زنده

• همانندسازی ← یک سلول ⟸ دو سلول جدید
سلول اولیه باید اندامکهای مضاعف داشته باشد
DNA باید عیناً در دو سلول ساخته شود.
مراحل همانندسازی
حضور آنزیم هلیکاز ← کاتالیزور شروع عملیات
DNA باید از وسط نصف شود ← جدا شدن پیوندهای هیدروژنی
عملیات از ۳ به ۵ انجام می‌شود
نوکلئوتید مکمل + آنزیم پلیمراز ← چسباندن مکمل‌ها

Subject :

Year . Month . Date . ( )

ایجاد برش خطا ← تصحیح توسط آنزیم

وصل کردن فاصله ها ← توسط آنزیم

آنزیم‌ها

پرایمیز ← برای شناسایی را قرار دهد (پرایمر)

پلیمرایز ← چسباندن + اصلاح کردن

لیگاز = لیگ + ایز ← پلیمر + ایز

هلیکاز ← پیوند ها را جدا می کند

لیگاز ← فاصله ها را از بین می برد

کارها به صورت تصادفی از هر جای رشته

خطا

پلیمرایز باز نامناسب را می چسباند ☑ U و T

نقص سلولی جنسی ← جهش

نقص سلولی غیر جنسی ← سلول سرطانی

نوع دوم آنزیم پلیمرایز سعی می کند که خطاها را اصلاح کند

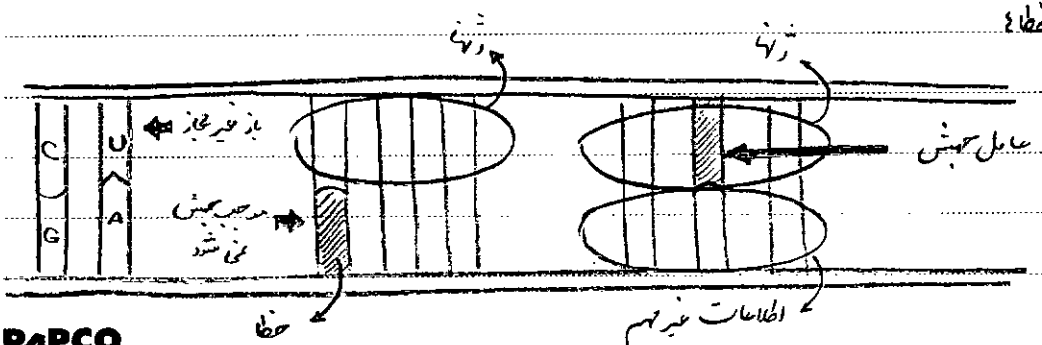
متابولیت بودن نوع باز ها، باعث می شود که از ترکیبات داخل سلول مولد، مواد مورد نظر را انتخاب کند ← باز های سلول

* علل عدم تمایل DNA به تغییر ← محافظت بوسیله غشای هسته

عدم تمایل تند آن (دی اکسی ریبوز) به شکست در دانشه

اصلاح چندباره به وسیله آنزیم‌ها

* نمایش خطاها



PAPCO

Subject:
Year. Month. Date. (8)

* اطلاعات روی DNA
  - Promotor ← شروع کروموزم از آنجا
  - Operon ← آیا ژن روشن است یا خاموش (برای تولید RNA پیشرو یا خیر)
  - Gene ← معمولاً دو ژن در نزدیکی روبروی هم قرار می گیرند.

* RNA ← پیوند A = U بیدارسیت تر از A = T است.
  از روی کدام RNA از روی DNA عملیاتی شبیه تکثیر DNA رخ می دهد ولی به جای T، U قرار می نشیند.
  چون تنه آن (دیوز) پیچ اکسیژن کم دارد، میل ترکیبی آن زیاد است.
  یک کپی از DNA است که از هسته خارج می شود ← تبدیل به پروتئین
  تنها روی رشته ای که ژن وجود دارد ترکیبی شود، و چون پیوند U با A سست است جدا می شود و رشته دیگر که تنها باقی مانده است دوباره مولد را به خود جذب می کند. در انتهای رشته تعداد A بیشتر است و پیوند سست تر.
  یک رشته است ← رشته مکمل ندارد.
  مراحل: Transcription و بعد Translation

DNA ☑

GATTCATCATGGATCATACTA
codon codon codon

* آمینو اسید ← هر codon یک آمینو اسید را کد می کند.
  تعداد codon ای ممکن $4^3 = 64$
  تعداد codon ای موجود ۲۰
  علت وجود کدهای استفاده نشده: کشف خطا
  یک کد شروع (AUG.) و سه کد پایان (، ،)
  ژن = رشته ای از آمینو اسیدها ← تنوع پروتئین‌ها = (۲۰) طول رشته

Subject:

Year. Month. Date. ( )

* تولید پروتئین

• وظیفه سلول ← از روی DNA تولید می‌کند ← کار دائم سلول تولید پروتئین است.

• از روی DNA کپی برداشته می‌شود و RNA تولید می‌شود ← از هسته بیرون می‌آید ← در روی کدون‌های آمینو اسیدها ساخته می‌شود ← از ترکیب آمینو اسیدها پروتئین ساخته می‌شود.

• رونویسی ← روی DNA

ترجمه ← پروتئین ها

رونویسی ⇄ ترجمه

• از روی DNA سه نوع RNA ساخته می‌شود: messenger - transfer - ribosome

mRNA ← از روی DNA کپی می‌شود ← تفاوت با DNA ← قند با اکسیژن کمتر

یک رشته‌ای

به جای T، U دارد

کدون‌های را حمل می‌کند

tRNA ← نقش حمل کننده آمینو اسید

rRNA ← اسکلت ریبوزوم را می‌سازد

• مراحل تولید پروتئین

۱- tRNA ← می‌آید ← شکل شبیه دارد ← پس از خروج از هسته این شکل را می‌گیرد

شامل سه برجستگی حامل codon

جای آمینو اسید به آن می‌چسبد

codon نوع آمینو اسید خود را تعیین می‌کند ☑

f-Met ≡ AUG

f-met

UAG ← Anti Codon

۲- ریبوزوم می‌آید ← دو تحت دارد ← تحت کوچک

تحت بزرگ

Subject:

Year. Month. Date. ۹۱

قسمت کوچک روی رشته قالبی لیزد ، +t-RNA در سایل روی آن قسمت بزرگ قرار می گیرد

بعد آن از r-RNA ساخته شده است.

سه جایگاه دارد ← P ← +t-RNA قالبی لیزد

← A ← +t-RNA جدید در آنجا قرار می گیرد

← E

۳- t-RNA روی رشته ی نشیند ← رشته سایل m-RNA است.

رشته حاوی کدهای ژنتیکی است.

t-RNA جایی ی نشیند که Anti Codon آن کامل باشد. در حقیقت Codon روی m-RNA ، نشان می دهد که چه آمینو اسیدی در آنجا قرار بگیرد

t-RNA ، m-RNA ، p-site با هم ترکیب می شوند.

۴- +t-RNA بعدی می آید ← t-RNA بعدی طوری ی نشیند که Codon در جایگاه A-site ترکیب شود.

آمینو اسید t-RNA جایگاه P به جایگاه A می چسبد.

۵- رشته m-RNA حرکت می کند ← از طرف ۵ به ۳ برای پپتید سازی

(از طرف ۳ به ۵ برای پایان سازی)

رشته یک Codon حرکت می کند A ← P ← E

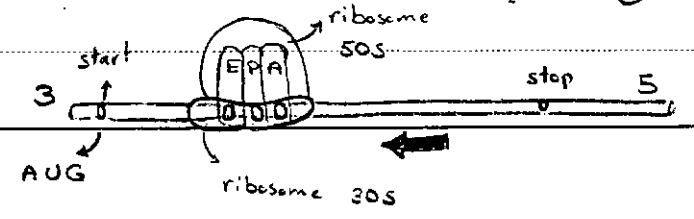
+t-RNA جایگاه E که آمینو اسیدش را جدا می شود

روند ادامه پیدا می کند ← شرط خاتمه : Codon توقف

آمینو اسیدی مربوط به آن نیست

رشته آمینو اسیدها با توجه به کد رشته m-RNA تولید می شود.

۶- ریبوزوم می رود پی کار خودش ← به این رشته دیگر چیزی کامل نشده است ، polypeptide می سازد.



Subject:

Year. Month. Date. ( )

• سلامتی بدن یا بیماری به حدود زیادی به خواص فیزیکی آن مرتبط است که این خواص از روی پروتئین تعیین می شود.

• کمبود آمینو اسید ← تولید کم پروتئین ← اختلال ارگانیسم

منابع آمینو اسید ← تجزیه مواد غذایی

تجزیه پروتئین های بازیافته سلول

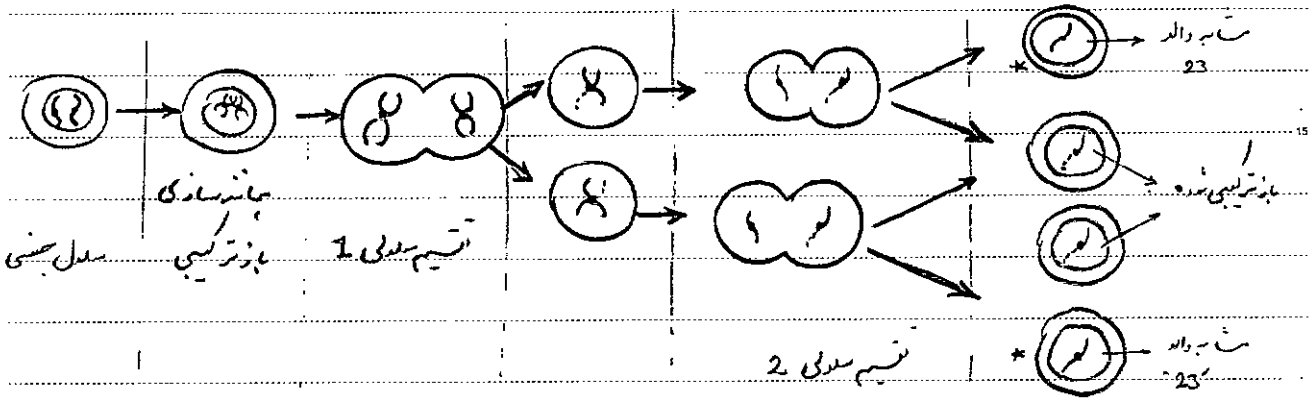
• خواص پروتئین ها وابسته است به ← کد ژنتیکی و رشته آمینو اسید

نوع ساختار شکل پروتئین ← خود این توسط کد کنترل می شود

DNA —Transcription→ RNA —Translation→ Amino Acid

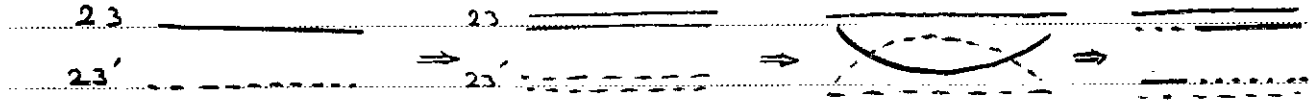
* تقسیم سلولی جنسی و وراثت (MEIOSIS)

• روی سلول های جنسی انجام می شود



• یک کروموزوم کاملاً مشابه با والد 1 و یک کروموزوم کاملاً مشابه با والد 2.

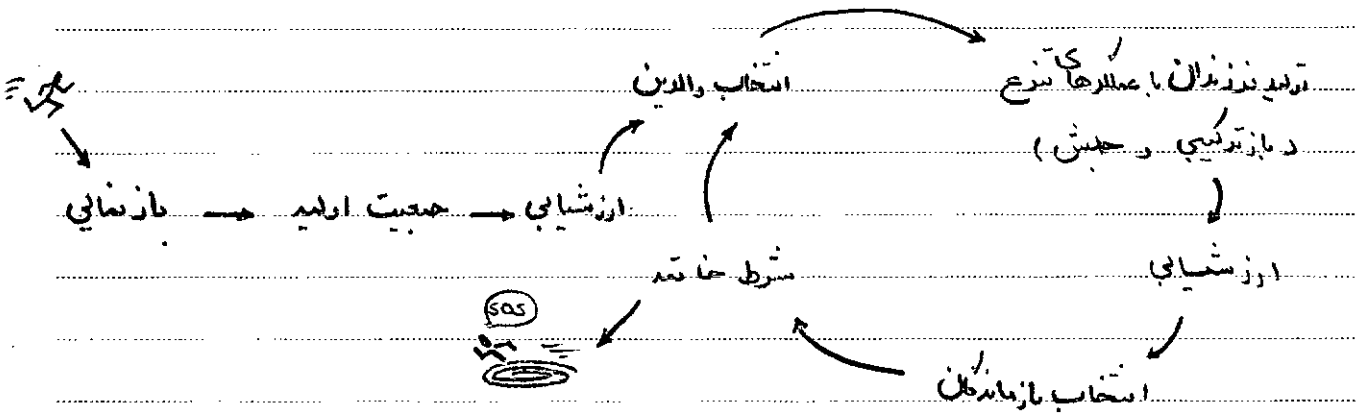
• تعداد کروموزوم 23 تا میشود



• برای cloning ، یکی از * ها برداشته و از روی آن کپیت را می سازند.

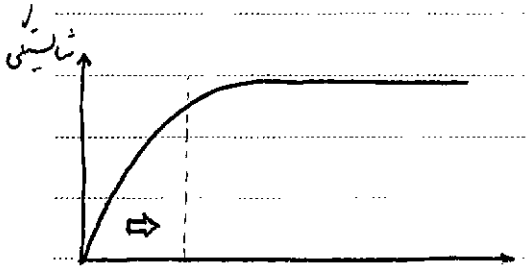
Subject:
Year. Month. Date. ( )

* مراحل الگوریتم تکاملی



* جمعیت اولیه

- بهتر است به صورت تصادفی ایجاد شود
- استفاده از الگوریتم heuristic
  - امکان نزدیک شدن به مینیمم محلی
  - تنها چند نسل طول می‌اندازد ← همگرا دارد؟



* شرط خاتمه

۱- تعداد نسل

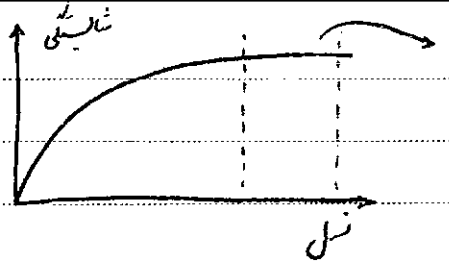
- عیب: به جواب نرسد، خیلی زودتر به جواب رسیده است
- مزیت: حلقه محدود، سادگی

۲- حد شایستگی

- عیب: حلقه بی‌نهایت، ممکن شایستگی بهینه و حد آن معلوم نباشد
- مزیت: اگر جواب پیدا شود تقریب خوبی از آن را دارد

۳- همگرایی ← در K نسل قبل میانگین یا متوسط شایستگی تغییر محسوسی ندارد.

Subject:
Year: Month: Date: ( )



- به جواب رسیده است یا در مینیمم محلی افتاده است.
- راه تشخیص همگرایی زودرس است.
- همگرایی زودرس یعنی یک نسل شدن جمعیت و نزدیک شدن مقدار شایستگی به هم ← تنوع از بین می‌رود ← تکامل از بین می‌رود.
- محور عمودی: میانگین شایستگی، شایستگی بهترین فرد
- عیب: حلقه بی نهایت، همگرایی زودرس

۴ ـ شکل ← تنوع در جمعیت از بین برود، در k نسل قبل از انحراف معیار شایستگی جمعیت از یک حد آستانه کمتر باشد.

- عیب: حلقه بی نهایت، همگرایی زودرس

۵ ـ ترکیب

- اگر شایستگی وجود داشته باشد ← ۱+۲+۳+۴
- اگر شایستگی وجود نداشته باشد ← ۱+۳+۴
- برای حل مشکل حلقه بی نهایت، راه حل ۱۰ همواره باید با راه‌های دیگر ترکیب شود.
- سرعت الگوریتم تکاملی ← تعداد دفعات محاسبه شایستگی

۵ معیارهای مقایسه الگوریتم‌های تکاملی

- تابع نسل ـ شایستگی ← متوسط شایستگی، بهترین شایستگی
- Robustness ← انحراف معیار شایستگی‌های در چند Run متوالی
- تعداد دفعات محاسبه شایستگی ← سرعت الگوریتم بدون دخالت تعداد نسل به عنوان شرط خاتمه

Subject:
Year. Month. Date. ( )

* بازنمایی
- کد کردن صورت مسئله
- استفاده از دانش مسئله برای کوچک کردن فضای مسئله
- کد کردن ژنوتیپ
- روش بایزی

- ساده
- طبیعت جهش یک تغییر کوچک ایجاد می کند ولی در روش بایزی نسبت به محل بیت، جهش می تواند تغییرات کوچک یا بزرگ ایجاد کند.
- طول کروموزوم ثابت
- مسائل چندبعدی و برای اعداد حقیقی نیاز دشوار در ژنوتیپ حل می شوند
- مزیت آن وجود انواع ژن و کدینگ بازترکیبی
- برای حل مسئله اولویت بیتها از Grey Code استفاده می شود تا اولویت بیتها چندان تأثیر نداشته باشد.
- جهش در بیتهای کم ارزش ← جستجوی محلی
- جهش در بیتهای پر ارزش ← جستجوی عمدی
- جهش با تغییر یک بیت صورت پذیرد

- روش اعداد صحیح
- روش اعداد حقیقی
- ماشین های متناهی حالت
- ساختارهای درختی
- گرامر

Subject:

Year. Month. Date. ( )

* انتخاب والدین

- ماهی اریات ، انتخاب والدین وجود ندارد و عاملی موجودات می‌توانند وارد فاز تولید مثل شوند.
- روشهای انتخاب والدین ، انتخاب نزدیکان است.

* معیارهای ارزیابی

۱- فشار انتخاب

SP = Selective Pressure

- احتمال انتخاب بهترین موجود در مقایسه با متوسط احتمال انتخاب همه موجودات
- عدد بزرگ ← شانس بالای انتخاب موجود بهتر
- عدد کوچک ← گرایش جمعیت به سمت شایستگی بهتر
- به طور متوسط در جمعیت جدید به تعداد SP تا از موجود بهتر ، جمعیت وجود خواهد داشت
- در جمعیت‌های اولیه باید SP کوچک باشد و در جمعیت‌های آخر باید بزرگ باشد.

۲- بایاس

Bias

- تعدد مطلق اختلاف بین شایستگی نرمال شده موجود احتمال انتخاب آن موجود
- صفر ← انتخاب متناسب با شایستگی
- دورتر از صفر ← کاهش متناسب انتخاب با شایستگی
- شایستگی نرمال شده ← شایستگی موجود تقسیم بر مجموع شایستگی‌های موجودات جمعیت

۳- Spread

- تعداد کپی‌های یک موجود پس از انتخاب
- $N_i : \mu . P_i$
- بهتر است ابتدا از هر موجود حداقل یک کپی وجود داشته باشد و در آخر کپی‌های موجود برتر بالا برود.

۴- از دست دادن تنوع

Loss of Diversity (LoD)

Subject:
Year. Month. Date. ( )

- در آخر باید زیاد باشد و در ابتدا کم باشد
- تعداد موجودات انتخاب شده به کل انتخاب

۵- واریانس انتخاب Selection Variance = SV

- امید ریاضی واریانس شایستگی جمعیت وقتی که انتخاب بردوی یک جمعیت با توزیع نرمال انجام شود. ← توزیع
- استانداردی برای مقایسه ← توزیع نرمال
- متوسط مقادیر واریانس شایستگی برای انتخاب‌های متفاوت و زیاد (> 20)

۶- تمایل انتخاب Selection Intensity = SI

- امید ریاضی متوسط شایستگی جمعیت بعد از اعمال عمل انتخاب بردوی جمعیت با توزیع نرمال
- توزیع نرمال مورد استفاده ← $N(0,1)$

* مسائل مطرح در انتخاب

- از کی انتخاب کنیم ← انتخاب والدین ← جمعیت محلی و همسایگی / چند جمعیتی
- انتخاب بازماندگان ← انتخاب از میان فرزندان (λ, μ) / انتخاب از میان فرزندان و جمعیت ($\mu+\lambda$)
- چگونه انتخاب کنیم

* مقایسه روش‌های انتخاب بازماندگان

- همیشه همرایی ← $\mu+\lambda$: با فرض انتخاب شایسته‌ترین‌ها تضمین می‌کند
  - μ, λ : وجود ندارد
- فراز از بهینم محلی ← $\mu+\lambda$: گیر می‌کند
  - μ, λ : چون امکان بدتر شدن شایستگی وجود دارد، می‌تواند

Subject:

Year. Month. Date. ( )

- سرعت ← $\mu+\lambda$: با انتخاب شایسته ترین ها سرعت بیشتری دارد
  μ,λ : سرعت کمتر
- دنبال کردن جمعیت‌های متحرک ← μ,λ : می‌تواند ← به خاطر فرار از مینیمم محلی
  $\mu+\lambda$: نمی‌تواند
- حافظه ← μ,λ : فراموشکار ← به دلیل فرار از مینیمم محلی
  $\mu+\lambda$: دارای حافظه
- سازگاری پارامترها ← μ,λ : می‌تواند پارامترهای بد را خراب کند
  $\mu+\lambda$: نمی‌تواند.

* معمولاً در الگوریتم‌های تکاملی یک مکانیسم «فرار از مینیمم محلی» کافی است ← μ,λ روشهای غیر شایسته سالاری

* انتخاب λ

- $\mu>\lambda$ ← $\mu+\lambda$
- $\mu<\lambda$ ← μ,λ یا $\mu+\lambda$
- $\mu=1$ ← $1,\lambda$ یا $1+\lambda$
- $\mu=\lambda=1$ ← $1+1$ (دلی $1,1$ نداریم چون معنای آن تصادفی محض است)
- $\lambda=1$ ← $\mu+1$ ← به خودش دشمن برخداری!

EA (1+1) ، EA(1,λ) ✓ ☑

EA (μ,μ) ← ✗ انتخاب معنی ندارد، چون هر فرزندی که تولید شد انتخاب می‌شود
✓ تنها در صورتی معنا دارد که انتخاب والدین بر مبنای شایسته سالاری باشد

این شرایط، حالتهای خاص هستند و دارای اسم هستند

EA (μ+μ)

Subject:
Year. Month. Date. ( )

* روشهای انتخاب

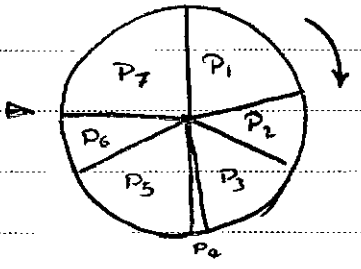
- تصادفی ناب یا ساده
  - لازم نیست از هر دو انتخاب، شایستگی خیلی باشد
  - از تمام جمعیت استفاده می‌شود

- شایستگی سالاری
  - مناسب برای استفاده با GA و NN ← مکانیسم فرار از min محلی وجود دارد.
  - انتخاب بهترین‌ها

- روشهای متناسب با شایستگی
  - چرخ رولت ← هر موجود دارای یک احتمال

P_i = شایستگی موجود / مجموع شایستگی‌ها

با این طرز کار می‌کند



پیاده‌سازی کامپیوتری

هر موجودی دارای احتمال خیلی بالایی باشد، برنده و باید بدید می‌آید. چون طبق Spread داریم:

$N = \mu \cdot P_i$

در این صورت اگر یک موجود شایستگی خیلی بالاتر از شایستگی متوسط جمعیت و کلیه از شایستگی جمعیت باشد، بعد از چند

Subject:

Year. Month. Date. ( )

نسل بعد تمام جمعیت از آن پر می شود. به این پدیده همگرایی زودرس می گویند.

اگر احتمال انتخاب موجودات نزدیک باشد ←

تنوع پایین است ←

فشار انتخاب کم است ←

حالت سکون پدید می آید.

حداقل Spread را هم کاهش می دهند.

$X \rightarrow P_X = 0.01$

$\mu = 10$

$N < 1 \Rightarrow N = 0$?!

مناسب با شایستگی ← SUS — Stochastic Universal Sampling

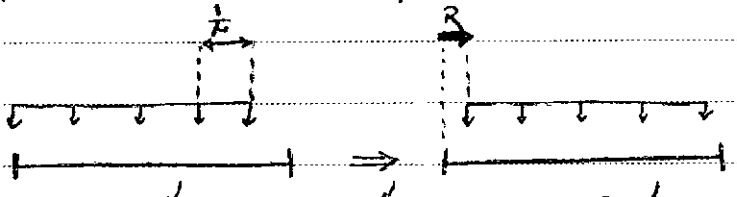
تضمین می کند که موجودات بد هم بتوانند انتخاب شوند.

خط نشان اول برمبنای احتمالات

P_1 P_2 P_3 P_4 P_5 P_6

خط دوم به طول $1 - \frac{1}{\mu}$ و با درجه بندی $\frac{1}{\mu}$

یک عدد تصادفی از بازه $R\ [0, \frac{1}{\mu}]$ به عنوان آغاز خط نشان دوم



به دلیل اینکه مناسب با شایستگی است ← همگرایی زودرس ← سکون

عیب روشهای مناسب با شایستگی ← همگرایی زودرس

$SP = \frac{P_{best}}{\frac{1}{\mu}\sum P_i} = \mu . P_{best} = N_{best}$

از این روشها فشار انتخاب بالاست.

$S.D(f) \rightarrow 0$ ← سکون

Subject:

Year. Month. Date. ( )

راه حل ٤ی روشهای مبتنی بر شایستگی

* رتبه بندی ← امتیاز براساس شایستگی ← رتبه بندی

1 2 ... μ

فاصله شایستگی ٤ متقابل می شود ← نرم ، نزاد

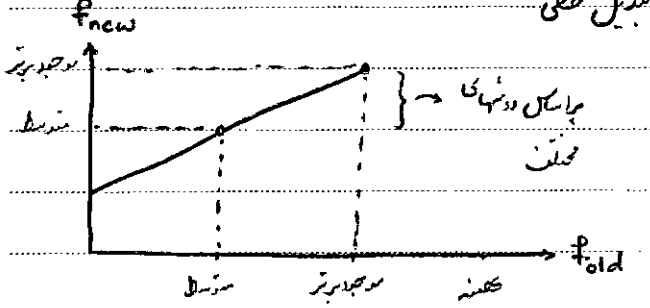
$$P_i = \frac{i}{\sum_{i=1}^{\mu} i} = \frac{2i}{\mu(\mu+1)}$$

حال از روش SUS یا RW استفاده می شود چون این دو روش مناسب با احتمال است (برخورد شایستگی)

حل شدن و بدرالی زودرس / احتیاج به محاسبه دقیق شایستگی نیست

* مقیاس بندی ← داشت فضای شایستگی برای مناسب کردن این فضا

تبدیل خطی



$f' = a f + b$

I) متوسط ثابت بماند ← $f'_{avg} = f_{avg}$

II) موجود برتر به میزان مناسبی ← $f_{max} = c \cdot f_{avg}$

از میانگین بالاتر برود

$a, b = ?$

معمولاً $c = 2$ تعداد ٤ی موجود برتر برابر 2 باشد

ممکن است شایستگی ٤ منفی باشد ← موجودات ضعیف را حذف می کنیم

برش سیگما

$f' = f - (f_{avg} - c.\sigma)$

σ : انحراف معیار جمعیت

مشکل شایستگی منفی را ندارد

P4PCO

پنجره بندی

$f' = f_i - \beta^t$

β : معمولاً بدترین موجود

Subject:

Year. Month. Date. ( )

Power Law

$f' = (f)^k$

$k = cte$ ⟵ مقدار ثابت دارد

$k > 0$ و $k < 0$ ⟵ انواع خیر

Boltzman

$P_t = \frac{e^{f_i/T}}{\sum e^{f_i/T}}$

T : دما

انواع روشهای متیاس بندی ⟵ حفظ ⟵ $af + b$

برش سیلی

ترانی

بولتزمان ⟵ غیرحفظ

• مبتنی بر رتبه

— بی نیاز از متیاستی است ⟵ تنها کافیست بدانیم کدام موجود بهتر از دیگری است

— سربار محاسباتی کم

—

— رتبه بندی حفظ

* استفاده از متیاس بندی برای کنترل فشار انتخاب

*

متیاس بندی $f_i = a f_i + b$

رتبه بندی $f_i = i$ ، $f_b = \mu$

$$\begin{cases} 1 \dots \mu \longrightarrow f_i' = ai + b \longrightarrow P_i' \longrightarrow SUS \text{ یا } RW \\ 1 \dots \mu \longrightarrow P_i' = ai + b \longrightarrow SUS \text{ یا } RW \end{cases}$$

$$P_i' = \frac{f_i'}{\sum f_i'}$$

گاهی اوقات شایستگی را تخمین می‌زنند ← NN و ... !

Subject:
Year. Month. Date.

$SP = \mu P_b = \mu(a\mu + b)$

$\sum P_i = 1 = \sum (ai + b)$ ⟹ $P_i = \frac{2-SP}{\mu} + \frac{2(i-1)(SP-1)}{\mu(\mu-1)}$

$SP = \mu \frac{f'_b}{\sum f'_i}$ ⟹ $SP = f'_b$

$\frac{1}{\mu} \sum f'_i = 1$

رتبه‌بندی نمایی

$P_i = a.e^{bi}$ * روش ۱

$P_i = \frac{1-e^i}{c}$ * ۲

$P_i = a + be^{ci}$ * ۳

* شرط

$\sum P_i = 1$

$SP = ?$

عیب ← هزینه Sort

• برگزاری مسابقه

روش q-tournament

* به صورت تصادفی q موجود انتخاب می‌شود ← بهترین انتخاب می‌شود
* μ بار تکرار می‌شود
* با جایگذاری و به‌ندرت بدون جایگذاری ← موجود برتر بیشتر انتخاب می‌شود.
* یک حالت نوعی از رتبه‌بندی است.
* $q = 1$ ← تصادفی
* $q = \mu$ ← شایستگی کامل
* لازم نیست شایستگی را کامل حساب کند ← بهترین موجود از q تا انتخاب می‌شود.
* تصادف ← بی‌عدالتی
* $q \uparrow$ ← شانس انتخاب ↑
* قابلیت قرار از min محلی ← می‌تواند بهترین انتخاب نشود

مثلاً مناسب با $\mu+2$

Subject:
Year Month Date ( )

* می‌توان در آن بهترین را برگزید.

* q_0 ثابت

exploration / exploitation ⟵ پویش تدریجی

استفاده از بازخورد محیط

* جمع‌بندی روش‌های انتخاب

۱- تقابلی انتخاب

۲- شایسته‌سالاری (برش)

۳- متناسب با برازندگی

۴- مقیاس‌بندی

• خطی

$f' = af + b \quad P_i = af_i + b$

• غیرخطی

$f' = ae^{bf} + c \quad P_i = ae^{bf_i} + c$

۵- رتبه‌بندی

۶- رتبه‌بندی مقیاس‌شده

• خطی

$f' = ai + b \quad P_i = ai + b$

• غیرخطی

$f' = ae^{bi} + c \quad P_i = ae^{bi} + c \quad SP = \mu P_b$

۷- q_0 تورنمنت

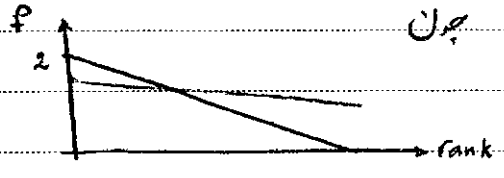
$\sum P_i = 1 \quad SP = \mu P_b$

Subject:
Year. Month. Date. ( )

* SGA
  - هر موجود فقط یک نسل عمر می کند. ← فرزندان جای والدین را می گیرند.
  - EA (μ, μ)
  - Generational Model
* Genitor
  - Steady State
  - EA $(\mu+1)$
  - یک فرزند جایگزین بدترین می شود. ← فرزند اگر از بدترین بهتر بود جایگزین بدترین می شود
* Generation Gap
  - درصدی از جمعیت جایگزین می شود.
  - EA (μ, λ) ← $\mu > \lambda$
  - حالت کلی دو حالت فوق

$P = \frac{\lambda}{\mu}$ ⟵ 0 : SGA
1 :

* در روش خطی ، $1 < SP < 2$ چون

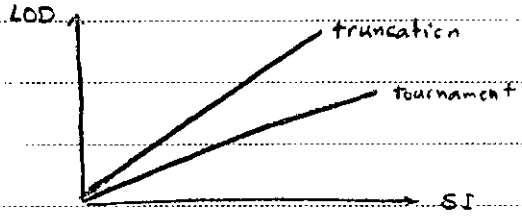


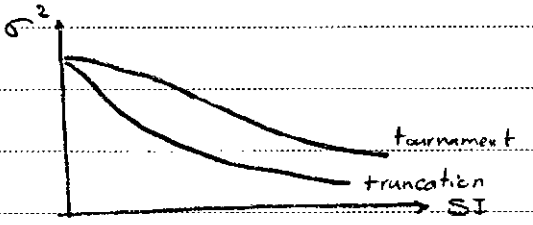
* فشار انتخاب ↑ ← خذف تنوع ↑
  - متوسط شایستگی ↑ ← خذف تنوع ↑
  - واریانس شایستگی ↓ ← خذف تنوع ↑
* انتخاب شایسته سالاری ← آستانه ↑ ← خذف تنوع ↓
  (پیشی) (٪ انتخاب)
  - واریانس ↑
  - متوسط شایستگی ↓

Subject:

Year Month Date ( )

* انتخاب مسابقه ← q_t ↑ ← متوسط شایستگی ↑
  - حذف تنوع ↑
  - واریانس شایستگی ↓
  - اعداد مناسب بین ۳ تا ۱۰
* ترکمنت q_t ادامه رتبه بندی خطی است (از لحاظ SP)
* میزان تنوع روش q_t ترکمنت نسبت به روش برش برای جمعیت کوچک بیشتر است.





پس q_t ترکمنت از روش برش بهتر است.

* SI ⟸ همگرایی ⟸ سرعت
* راه‌های ویژه در زمینه انتخاب:
  - Elitism ← گرامی داشتن بهترین ← سرایت دادن یا ندادن در عملگرهای تنوع
  - Genitor ← حذف کردن بدترین ← تولید مثل فرزند جایگزین کردن آن

Subject:

Year. Month. Date. ( )

* محاسبات پارامترهای عملکردی انتخاب

• رتبه بندی خطی

$$SI\ (SP) = \frac{(SP-1)}{\sqrt{\pi}}$$

$$LoD\ (SP) = \frac{(SP-1)}{4}$$

$$SV\ (SP) = 1 - \frac{(SP-1)^2}{\pi} = 1 - (SI)^2$$

• انتخاب برشی

F_β : مرجع‌هایی که شایستگی کمتر از F_β دارند انتخاب نمی‌شوند.

$$SI\ (\beta) = \frac{1}{\beta} \cdot \frac{1}{\sqrt{2\pi}} e^{-\frac{F_\beta^2}{2}}$$

$$LoD\ (\beta) = 1 - \beta$$

$$SV\ (\beta) = 1 - SI(\beta) \cdot (SI(\beta) - F_\beta)$$

• انتخاب q نوع‌علت

$$SI\ (q) \approx \sqrt{2\ln(q) - \ln\sqrt{4.14 \ln q}}$$

$$LoD\ (q) = q^{\frac{-1}{q-1}} - q^{\frac{-q}{q-1}}$$

$$SV\ (q) = \frac{0.918}{\ln(1.186 + 1.328q)}$$

Subject :

Year . . Month . . Date . ___( )

Subject:
Year. Month. Date. ( )

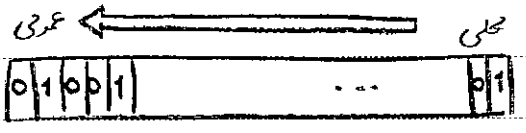
* عملگرهای تنوع

- تعداد عملوند (Arity)
  - 1 : Mutation
  - 2 : Cross Over
  - > 2 : Recombination
- وابسته به روش بازنمایی است.

GA ⟵ جهش اجباری ☑
GP ⟵ جهش ندارد

- برنامه ها می توانند جستجوی محلی یا عمومی انجام دهند.

☑ جهش در GA



بسته به محل جهش، نوع جستجو متفاوت است.

* بازترکیبی

- آیا بازترکیبی انجام شود یا خیر ⟵ احتمال P_c
- باینری
  - یک نقطه‌ای ⟵ محل آن تصادفی
    - P_c در GA بالا است ⟵ بین 0.6 تا 0.9
    - دو والد از محل بازترکیبی با هم جایشان را عوض می‌کنند.
    - ژن‌های کنار هم معمولاً پهلوی هم می‌مانند
    - ژن‌های ابتدا و انتهای یک فرد حفظ می‌شوند
    - دارای بایس ذاتی
    - کاربرد در مسائلی که ساختار آن مشخص باشد.
  - تصادفی ⟵ یک ژن تایپ

Subject:

Year Month Date ( )

• جایگشت

— روشهای بازیزی خاصیت جایگشت را برهم می زنند.

— عوامل مهم در این روشها ← حفظ ترتیب
حفظ همسایگی

— Order 1 ← یک زیررشته کپی می شود.
بقیه رشته با ترتیب رشته والد دوم کپی می شود.
ترتیب حفظ می شود.

1 2 3 | 4 5 6 7 | 8 9 ☑
9 3 7 | 8 2 6 5 | 1 4

3 8 2 | 4 5 6 7 | 1 9

— PMX ← یک زیررشته از والد اول کپی می شود
زیررشته اول با اعضا می کپی ← عدد i را به دست می آورد.

1 2 3 | 4 5 6 7 | 8 9 ☑
9 3 7 | 8 2 6 5 | 1 4

9 3 2 | 4 5 6 7 | 1 8

— Cycle ← وقت حفظ حلقه ها ست
همه حلقه ها را پیدا می کند
در یک فرزند یکی از حلقه ها و فرزند دیگر حلقه دومی را دست می زند.

Subject:
Year. Month. Date. ( )

Edge ← سمبایلی هر موجود تعیین می‌شود ← ۴ سمبایه
به جای تکراری ها + می‌گذاریم

برای شروع یکی را به طور تصادفی انتخاب می‌شود.
لیست سمبایه‌ها را می‌آوریم
بعدی را انتخاب می‌کنیم

۱- موجود با سمبایه‌ی تکراری
۲- موجود با کمترین لیست
۳- انتخاب تصادفی

این روش سمبایلی را حفظ می‌کند
معمولاً از دو والد، دو فرزند تولید می‌شود و نه بیشتر

Modified Edge ← مبتنی بر سمبایلی و ترتیب
سمبایلی‌های مشترک را حفظ می‌کند.
بقیه را از روی ترتیب والدین پر می‌کند.

عملگرهای چند والدی ← مبتنی بر ژنها ← رأی‌گیری
مبتنی بر بخش‌بندی ← چند نقطه‌ای مقطعی
حسابی ← برزنتل

Subject:
Year. Month. Date. ( )

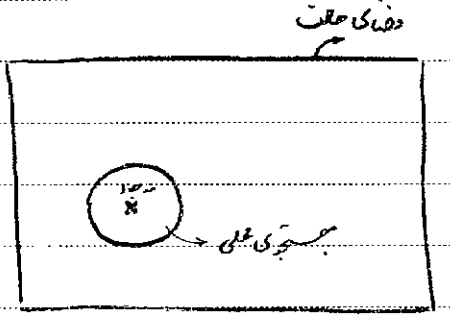
* عملگرهای جهش

- تضمین کننده تنوع
- حالت باینری ← مقدری بین bit را با حالت P_M تغییر می‌دهد

$$\frac{1}{Pop} < P_M < \frac{1}{Chromosome}$$

- اعداد حقیقی ← نویز نرمال $N(0,\sigma)$ به هر ژن اضافه می‌شود.
- روش جابجایی ← دو ژن را انتخاب می‌کند و کنار هم می‌آورد ← خط ترتیب
  ژن را انتخاب می‌کند و جای آن را عوض می‌کند ← جابجایی
  یک زیررشته را انتخاب می‌کند و معکوس می‌کند ← جابجایی
  یک زیررشته را انتخاب می‌کند و بهم می‌ریزد

* الگوریتم ژنتیک در برابر استنتاج



* شناسنامه الگوریتم ژنتیک

۱- باینری
۲- جمعیت اولیه ← N ، λ ، μ
۳- شایستگی ← نوع روش محاسبه
۴- انتخاب والدین
۵- بازترکیبی ← احتمال ، نوع
۶- جهش ← احتمال ، نوع
۷- انتخاب بازماندگان ← نوع ، $\mu + \lambda$ ، ...
۸- شرط خاتمه
۹- ویژگی‌های خاص

Subject:

Year Month Date ( )

کارگاه محاسبات تکاملی

۲۳ مرداد ۸۷

* سخنرانی کلیدی:

• [Zadeh] :

Fuzzy Logic
Neuro Computation
Evolutionary Computation
⟹ عدم ربط به قطعیت در زمینه سیستم های هوشمند مبتنی بر الگوهای کنترل بشری

• Expert : با انجام تکراری یک عمل تجربه کسب کنند تا Performance را داشته باشند.

• ضعف مدلسازی ریاضی ← در نظر نگرفتن تعامل (Interaction) بین اجزا

• ضعف الگوریتم ها در سیستم های واقعی ، ضعف اعداد و ریاضی در مدلسازی واقعیت

عدم قطعیت ← Fuzzy ← Membership x

معادلات بازگشتی ← دینامیک پویای غیر خطی

کوشش به طبیعت ✓

Cellular Automata ← تداخل بازگشتی و تعامل از قبل تعریف شده اند

— عدم قطعیت ناشی از غیر قابل پیش بینی بودن نتیجه تعامل است.

— نظریه آشوب ← شاه دلی

زبان اطلاعات (Logistic)

— مدل مغز ← بعد Fractal ← نماینده دینامیک مغز

در جهت همکاری و تعامل اجزا مغز

• شبکه های تکاملی:

Neuro Fuzzy —

Pure Delay —

Modular —

... —

Subject:

Year. Month. Date. ( )

* موضوعات ارائه

۱- PSO

۲- AC

۳- AIS

۴- SA

۵- الگوریتم تفاضلی DE

۶- الگوریتم ارتقایی QEA

۷- الگوریتم تخمین توزیعی EDA

۸- هم الگوریتم Coevolutionary ← همکاری / رقابتی

۹- Memetic MEA

۱۰- چند هدفی الگوریتم

۱۱- چند هدفهای الگوریتم

۱۲- با محدودیت الگوریتم

۱۳- GP

۱۴- CS ← Classifier System

Subject:
Year. Month. Date. ( )

Evolution Strategy *

• دهه شصت : استفاده از استراتژی به منظور بهبود سرعت و وقت تکامل

• قانون $\frac{1}{5}$ موفقیت ← روشنبرگ
ES(1+1)

• کاربرد ← بهینه سازی عدد ← به خصوص با ابعاد بالا

• ویژگی ← سرعت
مناسب برای فضای اعداد حقیقی
مبانی ریاضی بیشتر
خود تطبیقی اندازه قدم جهش

• جهش ← نویز گوسی با پارامتر σ
اندازه σ به روش استراتژی های مختلف متوسل می شود تا خود تطبیقی شود

• شناسنامه ES $(\mu \overset{+}{,} \lambda)$
- بازنمایی ← اعداد حقیقی
- جمعیت ← μ و λ ، تصادفی بی اهمیت
- شایستگی ← وابسته به کاربرد
- انتخاب والدین ← تصادفی عمداً
- بازترکیب ← گسسته و میانگین گیری
- جهش ← نویز گوسی $N(0,\sigma)$
- انتخاب بازماندگان ← $\mu+\lambda$ و μ,λ ← شایسته سالاری
- شرط خاتمه ← مهم نیست
- ویژگی خاص ← قدم جهش تطبیقی یا خود تطبیقی

Subject:
Year. Month. Date. ( )

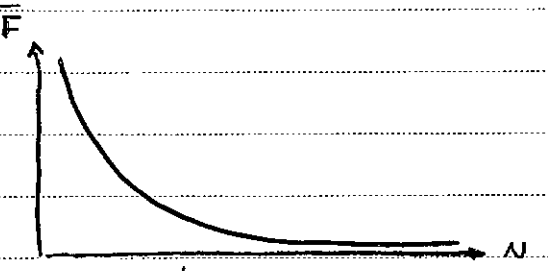
$E(1+1)$

$f: R^n \longrightarrow R$

بردارهای نقاط : R^n ، بازه‌ای

بازترکیبی : —

جهش : $N(0, \sigma)$



منحنی $ES(1+1)$ همیشه در حال کم شدن است و ثابت می‌ماند.

$$X_i' = X_i + \sigma_i N(0,1) = X_i + N(0, \sigma)$$

قانون 1/5 موفقیت در جهش : ثابت شده است که اگر تعداد موفقیت‌ها 1/5 باشد، سرعت تکامل بیشتر است.

$$\sigma_i' = \begin{cases} \sigma_i \div c & P_s > \frac{1}{5} \\ \sigma_i \times c & P_s < \frac{1}{5} \\ \sigma_i & P_s = \frac{1}{5} \end{cases}$$

$0.8 \leq c < 1$

موفقیت کم ← دور جواب نوسان می‌کند ← $\sigma \downarrow$

موفقیت زیاد ← به جواب نزدیک می‌شود ← $\sigma \uparrow$ (گام‌ها بزرگتر)

موفقیت = تعداد دفعاتی که فرزند بهتر از والد باشد.

$$P_s = \text{احتمال موفقیت} = \frac{\text{تعداد موفقیت در } k \text{ نسل قبل}}{k}$$

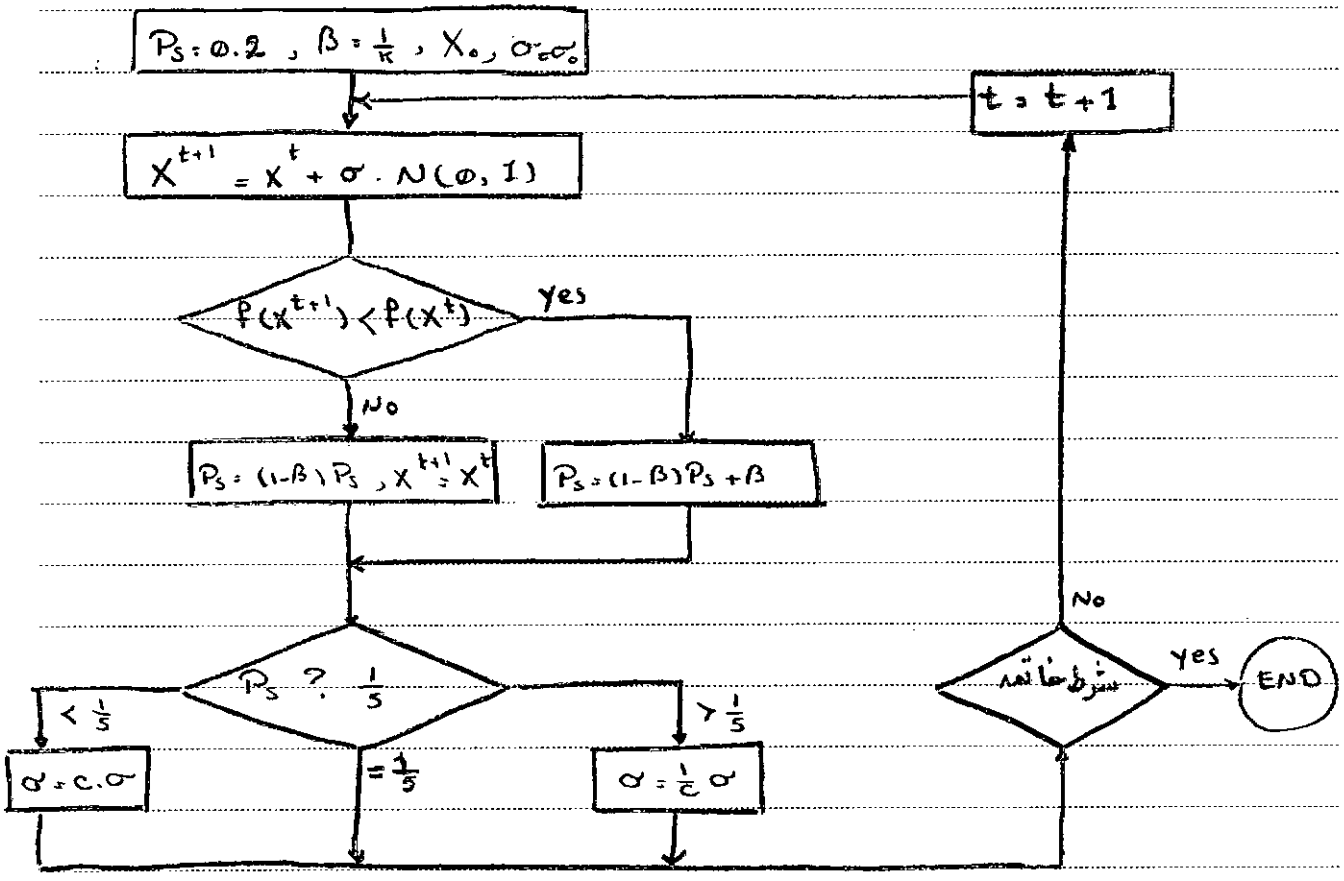
برای پیاده سازی :

$$\bar{A} = (1-\beta)A + \beta B$$

Subject:

Year . Month . Date . ( )

فلوچارت الگوریتم ES(1+1) :



اثبات قانون 1/5 :

x^t ⟵ * در لحظه t متغیر X مشخص در بار متغیر x^t است.

$\underline{x} = (x_1, \ldots, x_n)$ ⟵ برداری از ژن‌ها

$\underline{X}$ ⟵ متغیر تصادفی

$\underline{X}^{t+1} = \underline{x}^t + N(0, \sigma)$

$y = F(x^t) - F(x^{t+1})$ ⟵ متغیر تغییرات تابع هدف

معمولاً می‌توان این تابع هدف را این طوری گرفت: (مثل RMS) تابع هدف

Subject:
Year. Month. Date.

$F(x) = \sum_{i=1}^{n} x_i^2$ ← sphere تابع *

$y = F(x^t) - F(x^{t+1})$

$z_i \sim N(0,1)$ ← متغیر تصادفی نرمال

$$y = \sum (x_i^t)^2 - \sum (x_i^t + \sigma z_i)^2$$
$$= \sum (x_i^t)^2 - \sum (x_i^t)^2 - 2\sigma \sum x_i^t z_i - \sigma^2 \sum (z_i)^2$$
$$= -2\sigma \sum x_i^t . z_i - \sigma^2 \sum (z_i)^2$$

* خاصیت متغیر تصادفی نرمال

$z_i \sim N(\mu_i, \sigma_i^2)$

$a + \sum b_i z_i \sim N(\mu, \sigma^2)$

$\mu = a + \sum b_i . \mu_i$

$\sigma^2 = \sum \sum b_i b_j . c_{ij}$

$\mu_i = E(z_i)$

$c_{ij} = Cov(z_i, z_j)$

اگر z_i ها مستقل باشند:

$c_{ij} = 0 \quad (i \neq j)$

$\sigma^2 = \sum b_i^2 \sigma_i^2$

*

$z_i \sim N(0,1) \Rightarrow \sum b_i z_i \sim N(0, \sum b_i^2)$

$y = -2\sigma Y - \sigma^2 \sum z_i^2$

$\hookrightarrow Y \sim N(0, \sum (x_i^t)^2) \xrightarrow{r^t = \sum (x_i^t)^2} Y \sim r^t N(0,1)$

Subject:

Year. Month. Date. ( )

$y = -2\alpha c_i z - \alpha^2 \sum (z_i)^2$

* توزیع ← Chis توزیع های اساسی !!!

- $\text{Gamma}(a, b, c) = \dfrac{(x-c)^{a-1} \cdot e^{-(x-c)/b}}{\Gamma(a) \cdot b^a}$

- $c \leq x < +\infty \quad a, b > 0 \quad , \quad -\infty < c < +\infty$

- $\text{Chis1}(n) = \text{Gamma}(n/2, 2, 0) = \dfrac{x^{n/2-1} e^{-x/2}}{2^{n/2} \Gamma(n/2)}$

$E(X) = n \qquad V(X) = 2n$

- $\text{Chis2}(n, \alpha) = \text{Gamma}(n/2, 2\alpha^2, 0) = \dfrac{x^{n/2-1} \cdot e^{-x/2\alpha^2}}{2^{n/2} \Gamma(n/2) \alpha^n}$

$E(X) = n\alpha^2 \qquad V(X) = 2n\alpha^4$

- $\text{Chi}(n, \alpha) = \dfrac{2(n/2)^{n/2} \cdot x^{n-1} \cdot e^{-nx^2/2\alpha^2}}{\Gamma(n/2) \cdot \alpha^n}$

- $\text{Can}(a, b) = \dfrac{b}{\pi \left[(x-a)^2 + b^2\right]^2}$ توزیع کوشی

$E(X) = a \qquad -\infty < x < +\infty$

- $N(\mu, \alpha) = \dfrac{1}{\alpha\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\alpha}\right)^2}$

Subject:

Year. Month. Date. ( )

$Z_i \sim N(0,1)$ ستل $\sum Z_i \sim ChiS1(n) \sim ChiS2(n,1)$ *

با زیاد شدن n توزیع به توزیع نرمال تبدیل می‌شود.

اگر $n \gg 30 \longrightarrow \sum Z_i \sim \frac{1}{2}(Y_1)^2 \longrightarrow Y_1 = N(\sqrt{2n-1}, 1)$

$$y = -2\alpha r + Z_1 - \frac{\alpha^2}{2} Y_1^2$$

$$Y_1 = \sqrt{2n-1} + Z_2$$

$$y = -2\alpha r + Z_1 - \frac{\alpha^2}{2}(2n-1 + Z_2^2 + 2\sqrt{2n-1}\, Z_2)$$

$2n-1 \sim 2n$ ← $n \gg 30$ *

$Z_2^2 + 2\sqrt{2n-1}\, Z_2 \approx 0$ ↙

$$y = -2\alpha r + Z_1 - \frac{\alpha^2}{2}(2n)$$
$$= -2\alpha r + Z - n\alpha^2$$

* y را نرمال می‌کنیم ($\times \frac{n}{r^2}$) ← تغیری در توزیع ندارد.

$$y' = \frac{ny}{r^2} = -2\frac{n\alpha}{r_t} Z - \left(\frac{n\alpha}{r_t}\right)^2 = -2\alpha' Z - \alpha'^2$$

* سرعت همگرایی (Φ)

$$\Phi = E\{y'^+\}$$

تست منفی را الزوماً وجود ندارد. چون ES(1+1) است و اگر فرزند مثبتی به وجود بیاید، آنرا انتخاب نمی‌کنیم.

$$\Phi = \int_0^{+\infty} y' \cdot P_{y'} \cdot dy'$$

$y' = -2\alpha' Z - \alpha'^2 \Longrightarrow P_{y'} \sim N(-\alpha'^2, 4\alpha'^2)$

Subject:

Year. Month. Date. ( )

$$P_{y'} = \frac{1}{2\sigma'\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{y'+\sigma'^2}{2\sigma'}\right)^2} \cdot dy'$$

* تغییر متغیر

$$u = \frac{y'+\sigma'^2}{2\sigma'}$$

$$y' = 2\sigma' u - \sigma'^2$$

$$dy' = 2\sigma' du$$

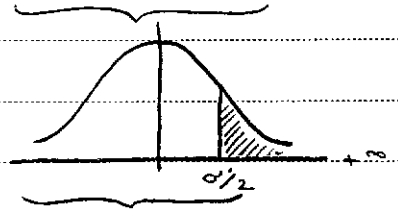
$$\Phi = \int_{\frac{\sigma'}{2}}^{+\infty} \frac{2\sigma' u - \sigma'^2}{\sqrt{2\pi}} e^{\frac{-1}{2}u^2} \cdot du$$

$$= \frac{2\sigma'}{\sqrt{2\pi}} \underbrace{\int_{\frac{\sigma'}{2}}^{+\infty} u e^{-\frac{1}{2}u^2}}_{} - \sigma'^2 \underbrace{\int_{\frac{\sigma'}{2}}^{+\infty} \frac{e^{-\frac{1}{2}u^2}}{\sqrt{2\pi}} \cdot du}_{}$$

$$\underbrace{-\int_{-\frac{\sigma'^2}{8}}^{-\infty} e^v \, dv}_{} \qquad \underbrace{1 - \int_{-\infty}^{\frac{\sigma'}{2}} \text{◎}}_{}$$

$$\underbrace{-e^v \Big|_{-\frac{\sigma'^2}{8}}^{-\infty}}_{} \qquad \underbrace{1 - \Phi\left(\frac{\sigma'}{2}\right)}_{}$$

$$\exp\left\{-\frac{\sigma'^2}{8}\right\}$$



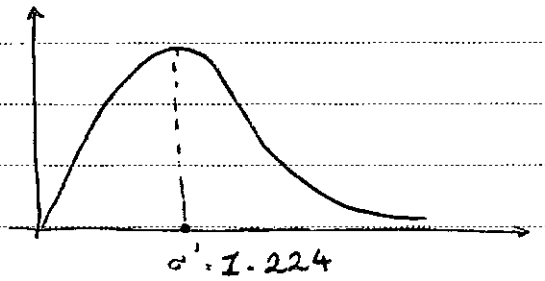
$$\Phi = \frac{2\sigma'}{\sqrt{2\pi}} e^{-\frac{\sigma'^2}{8}} - \sigma'^2 + \sigma'^2 \Phi\left(\frac{\sigma'}{2}\right)$$

$$\Phi(t) = \int_{-\infty}^{t} \frac{1}{\sqrt{2\pi}} e^{-\frac{t^2}{2}} \cdot dt$$

* تابع Φ جای جمعیت سازی نسبت به σ' مستقل گرفته بلایر صفر تقارن دهیم.

Subject:
Year. Month. Date. ( )

$\frac{d\varphi}{d\sigma'} = 0 \xrightarrow{\text{راه حل سدی}} \sigma' = ?$ شکل را رسم می‌کنیم



$\Rightarrow \sigma'^{*} = 1.224 \Rightarrow \varphi(1.224) \sim max$

*

$P(success) = P(F(x^t) \geq F(x^{t+1}))$

$= P\left(\frac{n}{r^2}(F(x^t) - F(x^{t+1})) \geq 0\right)$

$= \int_0^{+\infty} P_{y'} \cdot dy'$

$= \int_0^{+\infty} \frac{1}{2\sigma'^{*}\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{y'+\sigma'^{*2}}{2\sigma'^{*}}\right)^2} \cdot dy'$

$\simeq 0.27$

* به طور کلی برای تای زابع این احتمال موفقیت حدالی 1/5 در می آید. این عدد را معیار قرار دهیم، علاوه بر اینکه به σ (شایستگی فضای) بستگی دارد، بلکه برای هر مسئله شکل متفاوتی دارد و در هر مسئله σ'^{*} تغییر می‌کند.

Subject:

Year Month Date ( )

$X_i \sim N(\mu_i, \sigma_i^2)$ *

مستقل: $a\sum b_i X_i \sim N(\mu, \sigma^2)$

$\mu = a + \sum b_i \mu_i$

$\sigma^2 = \sum b_i^2 \sigma_i^2$

غیر مستقل: $\mu = a + \sum b_i \mu_i$

$\sigma^2 = \sum\sum b_i b_j c_{ij}$

$\mu_i = E(X_i)$

$c_{ij} = Cov(X_i, X_j)$

*

$X_i \sim N(0,1)$ —مستقل→ $\sum X_i^2 \sim chiS1(n) \sim chiS2(n,1)$

$X_i \sim N(0,\sigma^2)$ —مستقل→ $\sum X_i^2 \sim chiS2(n,\sigma) \sim \sigma^2 chiS1(n)$

$X_i \sim N(0,\sigma^2)$ —مستقل→ $\sqrt{\frac{1}{n}\sum X_i} \sim Chi(n,\sigma) \sim \sqrt{\frac{1}{n} chiS2(n,\sigma)}$

$X_i \sim N(\mu,\sigma^2)$ —مستقل→ $\frac{nS^2}{\sigma^2} \sim chiS1(n)$

$nS^2 \sim \sigma^2 chiS1(n)$

$S^2 = \frac{1}{n}\sum (X_i - \mu)^2$

$X_i \sim N(M,\sigma^2)$ —مستقل→ $\frac{n-1}{\sigma^2} S^2 \sim chiS1(n-1)$

$(n-1)S^2 \sim \sigma^2 chiS1(n-1)$

$S^2 = \frac{1}{n-1}\sum (X_i - \bar{X})^2$

$\bar{X} = \frac{1}{n}\sum X_i$

$X_i \sim Cau(a_i, b_i)$ —مستقل→ $\sum X_i \sim Cau(\sum a_i, \sum b_i)$

$X, Y \sim N(0,1)$ —مستقل→ $X/Y \sim Cau(0,1)$

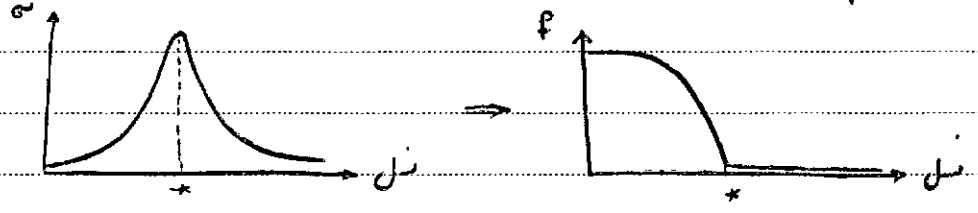
Subject:
Year Month Date ( )

$X_i \sim N(0,1) \xrightarrow[n \gg 30]{\text{نرمال}} \sum X_i^2 \sim ChiSq(n) \sim \frac{1}{2}Y^2$

$Y \sim N(\sqrt{2n-1}, 1)$

• قانون 1/5 موفقیت برای تابعی unimodal کاربرد دارد و در تابعی multi-modal صادق نیست.

• ابتدا : σ بزرگ ← سرعت
انتها : σ کوچک ← دقت



* پیاده سازی ES خود تطبیق

• پیدا کردن σ^* کار بسیار دشواری است. در این وظیفه نیز به الگوریتم تکاملی سپرده شده، کار راحت می‌شود. ابتدا در آن σ حدس زده می‌شود و سپس مجتمع‌ی عامل جستجو به کمک آن تغییر می‌کند.

• روش اول:

$X_i' = X_i + N(0, \sigma)$

representation: $\langle x_1, \ldots, x_n, \sigma \rangle$

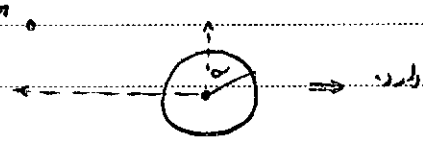
mutation : $\sigma' = \sigma \cdot \exp\{\tau \times N(0,1)\}$

- $\sigma' > 0$
- تغییرات σ بزرگ نخواهد بود
- $\tau \propto 1/\sqrt{n}$ ← نرخ یادگیری
- $\sigma' < \varepsilon_0 \rightarrow \sigma' = \varepsilon_0$

ε_0 آستانه‌ای است که دقت الگوریتم به آن وابسته است.

Subject:
Year Month Date ( )

مشکل این روش این است که به علت ثابت بودن σ یکی است، روی یک دایره حرکت می‌کند و هیچ‌وقت به مرکز نزدیک نمی‌شود.



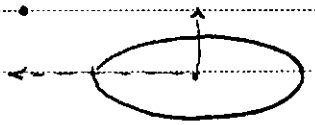
• روش دوم:

$$X_i = X_i + \sigma_i' \cdot N_i(0,1)$$

representation : $\langle x_1, \ldots, x_n, \sigma_1, \ldots, \sigma_n \rangle$

mutation : $\sigma_i' = \sigma_i \cdot \exp\{\tau' N(0,1) + \tau N_i(0,\frac{1}{2})\}$

- $\tau' \propto 1/\sqrt{n}$ → نرخ یادگیری سراسری
- $\tau \propto 1/\sqrt{2\sqrt{n}}$ → نرخ یادگیری
- τ' تغییر مشترک را مشخص می‌کند
- τ تغییر را در هر بعد مشخص می‌کند
- $\sigma_i' < \varepsilon_0 \Rightarrow \sigma_i' = \varepsilon_0$ توان بزرگ



مشکل این روش این است که x_i و σ_i با هم ارزیابی می‌شوند و ممکن است x_iهای خوب σ_iهای بدی داشته باشند. ولی اگر x_i خوب باشد احتمال اینکه σ_i آن خوب باشد زیاد است. این مشکل ناشی از این است که σ_i بر مبنای x_i ارزیابی می‌شود.

این روش، ابتدا با $\tau' N(0,1)$ جهش می‌کند و سپس بر هر بعد به اندازه $\tau_i N_i(0,1)$ دوباره جهش می‌کند.

اگر بتوان بطور مؤثر به سمت بیشینه (محلی) پرید، ممکن است سرعت آن افزایش یابد. با توجه به همبستگی ابعاد این کار قابل اجراست. پس نیاز به ماتریس کوواریانس، ماتریس‌های چرخش، بردارهای ویژه، وجود دارد. ماتریس کوواریانس همبستگی ابعاد را در بر دارد. ماتریس‌های چرخش زاویه بین ابعاد را محاسبه می‌کند. بردارهای ویژه طوری دستگاه مختصات را می‌چرخانند که ابعاد از هم مستقل شوند.

روش‌های خودتطبیقی حتماً از (μ, λ) استفاده می‌شود و برنده روش تورنمنتی بهتر جواب می‌دهد. (μ, λ) پارامترهای بد را سریع‌تر حذف می‌کند و مناسب‌تر است.

$ES(1+1)$ ⟵ بازترکیبی ندارد

$ES(\mu+\lambda)$ ⟵ q-tournament

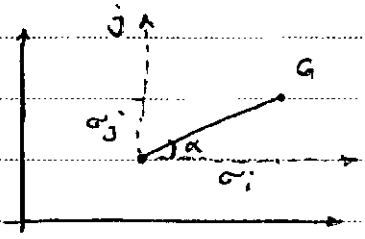
• روش سوم:

ماتریس کوواریانس

representation: $\langle x_1, \dots, x_n, \sigma_1, \dots, \sigma_n, \alpha_1, \dots, \alpha_k \rangle$ $\quad k = \binom{n}{2}$

covariance matrix:

$$C' = \begin{bmatrix} \sigma_1^2 & p_{12}\sigma_1\sigma_2 & \dots \\ \vdots & \ddots & \\ & & \sigma_n^2 \end{bmatrix} \rightarrow \begin{cases} i = j \rightarrow c_{ij} = \sigma_i^2 \\ i \neq j \rightarrow c_{ij} = p_{ij} c_i c_j \end{cases}$$



$$\tan\alpha = \frac{\sigma_j}{\sigma_i}$$

$$\tan 2\alpha = \frac{2\tan\alpha}{1-\tan^2\alpha} = \frac{2\sigma_i\sigma_j}{\sigma_i^2 - \sigma_j^2}$$

$$c_{ij} = \frac{1}{2}(\sigma_i^2 - \sigma_j^2)\tan 2\alpha$$

این روش ← دلیل اینکه p_{ij} ← علاقه آنها را در نظر نمی‌گیرد، جواب نمی‌دهد.

$$\sigma_i' = \sigma_i \cdot \exp\{\tau' N(0,1) + \tau N_i(0,1)\}$$
$$\alpha_j' = \alpha_j + \beta \cdot N(0,1)$$
$$x' = x + N(0, C')$$

Subject:

Year. Month. Date. ( )

- $T' \propto 1/\sqrt{2n}$
- $T \propto 1/\sqrt{2\sqrt{n}}$
- $\beta \approx 5^\circ$
- $\alpha_i' < \epsilon_0 \Rightarrow \alpha_i' = \epsilon_0$
- $|\alpha_j'| > \pi \Rightarrow \alpha_j' = \alpha_j' - 2\pi . \text{Sign}(\alpha_j')$

Subject:

Year. Month. Date. ( )

Subject:
Year. Month. Date. ( )

* برنامه نویسی تکاملی — Evolutionary Programming

- کاربرد:
  - یادگیری ماشین با FSM
  - بهینه سازی
- بازترکیبی وجود ندارد و تنها جهش در آن حاکم است.
- می‌تواند مشابه ES خودتطبیقی داشته باشد.
- شناسنامه:

— بازنمایی	FSM ، اعداد حقیقی
— جمعیت اولیه	تصادفی/دانش مسئله
— شایستگی	نسبت به مسئله
— انتخاب والدین	ندارد.
— بازترکیبی	فقط FSM ها بحث می‌شود.
— جهش	برای اعداد حقیقی مشابه ES (حتی با مدل‌های دیگر)
— بازماندگان	$\mu+\mu$ ، سانه
— خاتمه	
— ویژگی خاص	بازترکیبی ندارد.
	تا بیت خودتطبیقی دارد.
	می‌تواند روی پراکنش عمل کند.

Subject:

Year. Month. Date. ( )

Subject:

Year. Month. Date. ( )

☑ در EP استاندارد

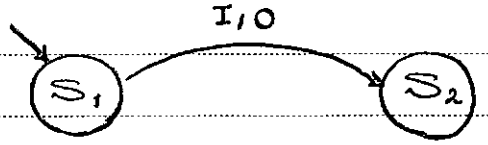
والدین : همه انتخاب می‌شوند

بازماندگان : متناسب با شایستگی

EP(μ+μ)

• هدف EP دادن پاسخ‌هایی به FSM است تا بتواند پیش‌بینی کند.

• FSM



$\delta : S \times I \longrightarrow S \times O$

برای پیش‌بینی حالت بعدی ، از روی خروجی حالت فعلی ، باید تبدیل حالت بعدی را تولید کند.

تعداد حالتهای FSM و جدول انتقال قابل پذیری هستند.

• شایستگی ماشین پیش‌بینی باینری

شایستگی = درصد درستی پیش‌بینی رشته ورودی با توجه به خروجی ⟹ رشته ورودی / رشته خروجی

☑

I: 0 1 1 1 0 1

O: 1 1 0 1 1 1

✓ ✓ x x ✓ ⟶ 3 out of 5 ⟶ 0.6

☑ پیش‌بینی اعداد اول

I : 1 2 3 4 ... N

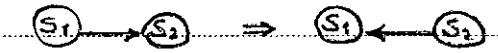
O : 0 : not prime 1 : prime

Penalty : wrong predictions / total

Subject:

Year. Month. Date. ( )

Mutation : $0 \rightarrow 1$ و $1 \rightarrow 0$ output



transition direction

insert state

delete state

initial state

ES(1+1)

$$\sigma' \quad 1.2 = \frac{n\sigma}{r} \Rightarrow \sigma^* = \frac{1.2\, r}{n} \Rightarrow \sigma^* \propto r \times \frac{1}{(\text{fitness})}$$

• در EP برخلاف ES از جذر واریانس ($\sigma = \sqrt{var}$) در جهش استفاده می‌شود و این جهش خطی است (و نمایی نیست) و تفاوت ترتیب جهش σ و x چندان فرقی ایجاد نمی‌کند

• انتخاب بازماندگان

* تعداد زادی سابقه (q)
* هر فرد با سابقه‌ی امتیازی گیرد ← امتیاز برحسب حداکثر q
* در پایان n مجدد که برنده شده‌اند انتخاب می‌شوند.

• تفاوت‌های EP و ES

* $\mu+\lambda$ ⚡ $\mu+\mu$
* سابقه سالاری ⚡ q-tournament
* بازترکیبی نداریم ⚡ بازترکیبی داریم
* تغییرات خطی σ ⚡ تغییرات نمایی σ

EP(1+1) ⟶ $x' = x + \sqrt{\frac{f(x)}{n}}\, N(0,1)$ ☑

EP($\mu+\mu$) ⟶ $x' = x + \sqrt{\alpha f(x) + \beta}\, N(0,1)$

Subject:

Year. Month. Date. ( )

Genetic Algorithm

- کاربرد
  - مسائل پیچیده
  - مسائل ترکیبی
- نمایش باینری
  - خیلی سریع نیست
- بازترکیبی
  - خیلی روی بازترکیبی تاکید می‌شود و نقش جهش را پررنگ می‌داند
- شامل عملگر
  - reproduction ← کپی فرزندان کاملاً شبیه والد
  - بازترکیبی روی دو والد با احتمال P_c

2 parents $\xrightarrow{P_c}$ 2 combined offspring

$\xrightarrow{1-P_c}$ 2 offspring completely similar to parent

- این عملگرها نحوه انتخاب از μ و λ را نشان می‌دهد.

- جهش

$0.001 < P_m < 0.01$

- تعداد جمعیت زیاد $\frac{1}{\text{تعداد جمعیت}} < P_m < \frac{1}{\text{تعداد ژن‌ها}}$

- انتخاب بازماندگان
  - فرزندان جایگزین والدین می‌شوند
  - علت وجود عملگر Reproduction است.
  - $GA(\mu, \mu)$
- انتخاب والدین
  - متناسب با شایستگی

Subject:

Year. Month. Date. ( )

- تبدیل به نمایش باینری

coding *

binary *

$$C_i = C_{min} + \frac{C_{max} - C_{min}}{2^l - 1} \times B_i$$

* روش Ant Colony

- مزیت این است به روشهای کامل در مسائل مربوط به کلاف است.
- در مجیطهای گسسته عمل می‌کند. ویژگی جدید برای مجیطهای پیوسته هم آمده است.
- نوری را شبیه می‌کنیم که بلد از آن عبور کرده باشد و نیز سیر مستقیم به آن دارد. به این مجموعه J می‌گوییم.

- T_{ij} ← فرومون
  - ← میزان اطلاعات عمومی

- η_{ij} ← جستجوی محلی
  - ← معمولاً نسبت معکوس با فاصله دو نود

$$\eta_{ij} = \frac{1}{D_{ij}}$$

- α و β ← اکتشاف در برابر استخراج
  - ← ترجیح: حرکت $\alpha \downarrow$ و $\beta \uparrow$

$$P_{ij} = \frac{(\tau_{ij})^{\alpha} (\eta_{ij})^{\beta}}{\sum_{j \in J} (\tau_{ij})^{\alpha} (\eta_{ij})^{\beta}}$$

☑ پیدا کردن کوتاهترین سیر در اینترنت ← روی مقدار بالا و پایین P_{ij} حدی می‌گذارند تا جلوی سیر در یک سیر جمعیت متوقف شود

PAPCO

• قانون بروز رسانی

در مسیر بهینه نباشد : $\tau_{ij} = (1-\eta)\ \tau_{ij}(old)$ ← تبخیر

در مسیر بهینه باشد : $\tau_{ij} = (1-\xi)\ \tau_{ij}(old) + \xi\ \Delta\tau_{ij}$

$\Delta\tau_{ij} \approx \begin{cases} 0 & , ij \notin Path \longrightarrow \text{تبخیر} \\ \frac{Y}{D} & , ij \in Path \longrightarrow \text{تقویت} \end{cases}$

• تعداد مورچه‌ها ← TSP : برای هر شهر یک مورچه

Routing : $\frac{1}{2}$ مورچه‌ها در مبدأ

• پارامترهای آزاد ← $\alpha, \beta, \gamma, \xi$

Genetic Programming*

• قدرت ساخت مدل را بدست می دهد.

• مزیتها نسبت به شبکه عصبی ← فرار از مینیمم محلی
عدم محدود شدن به ساختار شبکه عصبی

• کاربردها ← یادگیری ماشین
پیشگویی
طبقه بندی

• معایب ← کند ← نیاز به جمعیت اولیه زیاد ← البته وقتی مدل درست شده جواب نهایی بسیار سریع از روی مدل پاسخ داده می شود.

نیاز به جمعیت اولیه زیاد ← میلیونها

• بازنمایی : درختی - غیرخطی

• جهش ← جمعیت کم ← مدت نیاز است.
← جمعیت زیاد ← $P_M = 0$

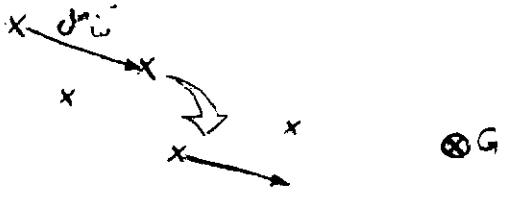
- شناسنامه

ارزیابی : ساختار درختی
بازترکیبی : تقاطع
جهش : تغییر زیردرخت
انتخاب بازماندگان : همه را انتخاب می‌کند.
انتخاب والدین : متناسب با شایستگی

- درخت فقط حالتهای مناسب را نمایش می‌دهد. اگر حالتی در درخت صدق نکرد، راه‌حل مناسب نیست.

☑ تولید برنامه خودکار
تولید قانون‌های انعطاف‌پذیر

* Differential Evolutionary
- سرعت حرکت به سمت هدف را تنظیم می‌کند.
- در یک جهت موجود تصادفی انتخاب می‌کند



- موجودات به سمت هدف می‌روند. معمولاً جهت آنها به سوی هدف است. پس تفاضل آنها برداری در راستای وکتور هدف (در شبکه و نرمالیزه شده) است. این تفاضل به عنوان بردار احتمال موجودی در جهت استفاده می‌شود. این موجود جهش یافته است.

$$V_i = X_{g1} + F (X_{g2} - X_{g3})$$

- از بین موجودات اولیه و موجودات جهش یافته، با توجه به معیار cross over می‌انتخاب می‌کند.

Subject :

Year. Month. Date.

Cross Over 1 •

/DE/rand/1/bin/

Next Gen. $\frac{1}{7}$ { Rand { $< CR \longrightarrow$ use $X_{\gamma 4}$; $> CR \longrightarrow$ use $V_i = X_{\gamma 1} + F(X_{\gamma 2} - X_{\gamma 3})$ } ; One of Mutated Individual

NFE ← معیار سرعت ← Number of Fitness Eval. •

/DE/best/1/bin/ •

$V_i = X_{best} + F(X_{\gamma 1} - X_{\gamma 2})$

/DE/best/2/bin/ •

$V_i = X_{best} + F((X_{\gamma 1} + X_{\gamma 2}) - (X_{\gamma 3} + X_{\gamma 4}))$

* ادامه GP

• جهش ← P_m : آیا جهش را انجام بدهیم. ← بیا رولت
روی کدام فرد جهش را انجام بدهیم.
آیا این جهش نتیجه درستی بدست می دهد ← ☑ علم ریاضی برای مقادیر منطقی
اگر جمعیت بزرگ باشد (تنوع زیاد) به این روش توصیه می شود ← جهش نمایی
نسبت • برای جبران محدودیت اندازه جمعیت، می توان از جهش گره برد.

• بازترکیبی ← P_c : آیا انجام بدهیم.
روی کدام زیر درختهای والدین اعمال شود.
تصادفی مناسب ✗ ← 0.1 برگ و 0.9 تابع ✓

• انتخاب بازماندگان ← original : نداریم
steady state : جدید

• انتخاب والدین ← ① برتر برگزیدن براساس شایستگی
② گروه بندی: x % بهترین گروه اول (100-x)% گروه دوم

Subject :

Year . Month . Date . : :

① 80٪ از گره اول انتخاب می‌شود و 20٪ از گره دوم ← شانس سلکتی شدن x وابسته به اندازه جمعیت

• تولید جمعیت اولیه ← F : نودهای میانی

T : نودهای برگ

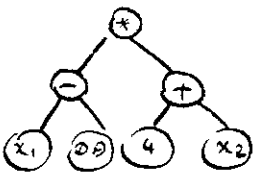
روش full method

depth $\begin{cases} < D_{max} \longrightarrow x_i = \text{rand from } (F) \\ = D_{max} \longrightarrow x_i = \text{rand from } (T) \end{cases}$

Function Set : $(+, -, *, /)$

Terminal Set : (R, x_1, x_2)

$D = 3$



50٪ ، 50٪ از پاپ‌اپ، یک درصد از تعداد جمعیت را

سپس انتخاب تصادفی عدد است بین برگ‌ها از گره‌ها

روش grow method

depth $\begin{cases} < D_{max} \longrightarrow x_i = \text{rand from } (F \cup T) \\ = D_{max} \longrightarrow x_i = \text{rand from } (T) \end{cases}$

۲ روش half_and_half

grow method و full method نیمی از جمعیت از

نیمی درختی :(

برای پر کردن باید half_and_half را استفاده کنید.

Subject :
Year . Month . Date . ( )

مشکلات GP

* چاقی درختها ← Bloat - Survival of the fattest
* نمی تواند تمام ساختارها را تولید کند ← تنوع ↓

• چاقی درختها
* علت تولید آن تابع شایستگی است.
* راههای جلوگیری
  ← بازترکیبی از نقاطی که عمق نزدیک به هم دارند.
  ← در شایستگی برای اندازه درخت جریمه قرار دادن ← ضریب خست Parsimony
  ← جلوی فرزندان تولید شده بزرگ گرفته شود.
* روش فشار خست نسبت به پیشگیری از تولید فرزندان مزیت دارد چون ممکن است واقعاً جواب بهینه درخت بزرگی داشته باشد.

• محاسبه تابع شایستگی
* استفاده از ماتریس ← چون روی تمام داده های آموزشی باید اعمال شود.
  ← ارزیابی مدل ← به همین دلیل NFE معیار سرعت است.

Subject :

Year . Month . Date . ( ) :

• Bloat in Genetic Programming

★ تعریف

تمایل GP برای بزرگ شدن درخت ها در طول زمان

- از لحاظ تعداد گره ها
- از لحاظ عمق

★ معایب

- فضای الگوریتم ← نمی تواند به کارش برسد
- منبع و پردازنده ی زیادی می برد ← همه به بزرگترین درخت همگرا می شوند

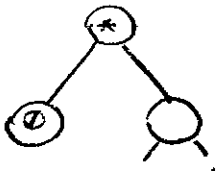
★ دلیل

Cross Over + Selection ←

Inviable ← جایگزینی آن زیر درخت با

هر زیر درختی تغییری در f به

وجود نمی آورد



★ توجیه

- Destructive Hypothesis ← طی cross over تخریبی است.
- Diffusion Theory ← over fitting
- Removal Bias
- Other ← Parasite ، ...

★ نقش بازترکیبی

- در ۵۰٪ موارد تأثیری ندارد
- در فرزند ۱ می احتمالاً کوچکتر و در ۲ احتمالاً بزرگتر
  - تابع f بزرگتر را برمی دارد و کوچکتر حذف می شود.
- Effective Length ← Inviable پس از برداشتن

★ کد بی اثر

- Inviable , Intron , Neutral
- تمام زیر درخت های کد نکرده بی اثر است.
- والدین هر گره viable ، viable خواهند بود.

Subject:

Year. Month. Date. ( )

Destructive Hypothesis *

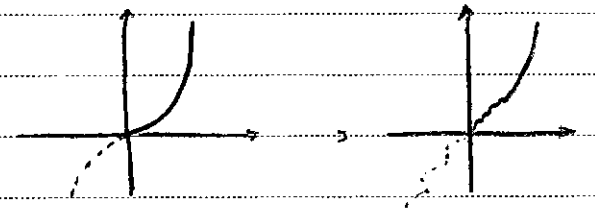
- اگر بازترکیبی روی لوکوس بیاید شود، به احتمال زیاد F خراب می‌شود.
- بنابراین در حالت میانگین، بازترکیبی یا بی‌تأثیر است یا F را خراب می‌کند.
- پس بازترکیبی روی زیردرختهای Inviable بی‌تأثیر است و این مطلب (با توجه به حالت میانگین)
- پس بازترکیبی میل به عمل روی زیردرختهای بی‌تأثیر دارد.

Removal Bias *

- زیردرخت غیرمثمر اگر بزرگ شود هم F ثابت می‌ماند.
- درختهای بزرگتر معمولاً زنده می‌مانند.

Diffusion Theory *

- اگر راه حلهای کوچک پیدا شود، به جای آن معادله‌ی بزرگتری کوچک شده است باقی می‌ماند.
- $\sin^2 x + \cos^2 x = 1$
- در حالت تقسیم داده شده، که ترتیب نوترونهای زنده عدد نمی‌شود، پس از دوره حالتهای دیده نشده عکس العمل می‌کند.



حال اگر بازه از ۰ به ۱ برود F خراب می‌شود.

Other Causes *

- زیردرختهای که لوکوس را خنثی می‌کند
- Parasite : راه حل خوبی موجود بوده است، با چسبیدن یک عامل خنثی کوچک و تقریباً Inviable، آن اثر را خنثی می‌کند و درجه بالایی خواهد بود ← راه حل:

PAPCO Migration

Subject:

Year: ___ Month: ___ Date: ( )

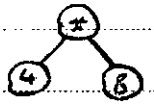
Solution *

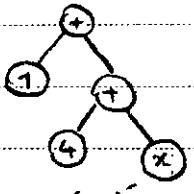
Code Editing ← ساده کردن اعداد با هم

در عین یک

ساده کردن اعداد در

عین دو





بالا بردن عق برای ازین بردن تضاد

وابسته به مسئله

محاسبه شایستگی هر گره و ذخیره سازی آن

output tree → output node

از پایین به بالا صدا می‌زنیم (Recursive)

گره‌های Inviable را پیدا می‌کنند

Parsimony Pressure ← سایز درخت در محاسبه f دخیل شود.

جریمه برای سایزهای بزرگ

$f = w_1 * f(size) + w_2 * f_{raw}$

دو پارامتر آزاد ×

Size Limiting ←

Static ← محدودیت روی عق (مثلاً 17)

محدودیت روی تعداد گره‌ها

Dynamic ← محدودیت روی تعداد گره‌های کل جمعیت

برای درون جمعیت

$f \begin{cases} < f_{best} \Rightarrow \text{درود} \\ > f_{best} \rightarrow \frac{\text{new size}}{\text{best size}} < C \Rightarrow \text{درود} \end{cases}$

اشکال آن است که ممکن است جواب در درختهای بزرگ باشد.

Subject :

Year . Month . Date . ( )

Hill Climbing ← فرزند باید از والد بهتر باشد یا مانیز آن را بهتر باشد تا در جمعیت قرار بگیرد.

شکل آن رشدن تعداد زیادی جمعیت است.

NFE↑

کوچکتر (نه ≥) چون در بهینه‌ی Inviable را راه می‌دهد.

* Comprehensive learning PSO

PSO .

* تعدادی پرنده ← برنامه‌ای راه حل

* تجربه شخصی ← جستجوی محلی

بهترین موقعیتی است که این پرنده تا حال بدست آورده است.

$P_i = X_i$ بردار تمایل به بهینه شخصی

* تجربه گروهی ← جستجوی عمومی

بهترین موقعیتی است که تمام پرنده‌ها تا کنون تجربه کرده‌اند.

$G = X_i$ بردار تمایل به بهینه عمومی

* جابجایی هر پرنده.

$$X_i^{t+1} = X_i^t + V_i^{t+1}$$

* سرعت پرنده ← دارای سرعت اولیه تصادفی

دارای اینرسی ω ← $\omega < 1$ تا همگرا شود

ابتدا بزرگ و سپس کوچک باید باشد.

می‌تواند به طور خطی کاهش یابد.

$$\omega_t = \omega_{max} - \frac{\omega_{max} - \omega_{min}}{N} \times t$$

دارای عدد تصادفی

$0 < r_1, r_2 < +1$

ضرایب معمولاً برابر 2 (می‌تواند تطبیقی هم باشد)

PAPCO $c_1 = c_2 = 2$

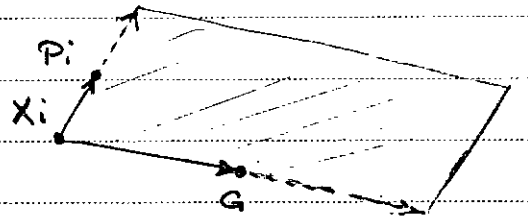
Subject:
Year Month Date ( )

* در نسل های اولیه به دلیل سرعت تصادفی اولیه جستجوی عمومی دارد. پس از آن است در نسل های اولیه به بهینه عمومی نزدیک میشود، جدا می شود.

* بروزرسانی سرعت

$$V^{t+1} = \omega . V^{t} + c_1 . r_1 (P_i - X_i) + c_2 . r_2 (G - X_i)$$

* منطقه جستجو



• راه حل سرانگشتی زودرس

* هر پرنده می تواند P_{best} پرنده‌ای استفاده کند این مسئله برای هر بعد مطرح است. این کار تنوع ایجاد می کند.

* احتمال پادگیری (P_c) ← P_c: استفاده از P_{best} خود پرنده
$1-P_c$: استفاده از P_{best} سایر پرندگان

* استفاده از G حذف شده است.

*

$$V_i^{t+1} = \omega . V_i^{t+1} +$$

* هر بعد پرنده پس از m بار، پرنده جدیدی را برای یادگیری استفاده می کند.

• توانست فضای جستجو

* برای توابع Unimodal جندان خوب نمی کند.

* برای توابع Multimodal روش CLPSO بهتر عمل می کند.

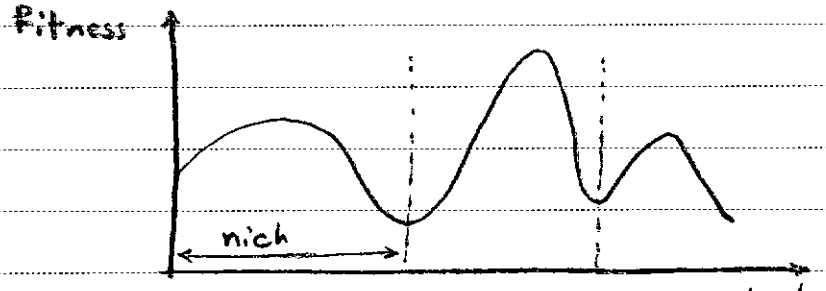
• احتمال یادگیری در ذرات مختلف متفاوت است. ← تنوع ↑

• بهترین PSO موجود

Subject:
Year. Month. Date. ( )

* مسائل Multimodal

• خیلی از مسائل دارای چند بهینه محلی هستند و ما می‌خواهیم همه آنها را بدست آوریم.



• در تمام جمعیت‌ها بهینه سراسری پیدا است.

• کاربرد:

* مسائل چندهدفی
* حل سیستم‌های محلی مطلوب باشد.
* حل مسائل Genetic Drift

• Genetic Drift

* الگوریتم تکاملی تنها تمایل دارد به سمت یکی از قله‌های تکامل یابد چون میل جمعیت به آن بیشتر می‌شود (احتمال آن به مرور زمان بالا می‌رود.)

• روشهای حل (Niching)

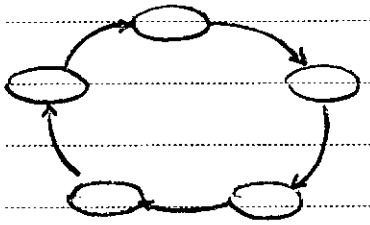
* سیستم صریح ← چند جمعیتی
  ← جمعیت با همسایگی محلی
  ...

* غیر صریح ← Fitness Sharing
  ← Crowding
  ...

Subject:
Year Month Date ( )

• روشهای ارتباط جمعیتها

* Island (Ring)



← ارتباط یک طرفه (مهاجرت)

← الیلهای چند جمعیتی با پراکنش سوزنی دارند

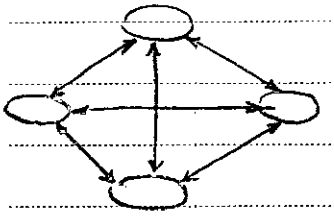
← سرعت آن بالا بود، با بهترین جواب

هر niche پیدا شود.

← پراکنش سوزنی: مهاجرت متناسب با شایستگی

← multimodal: مهاجرت موجودات به جمعیت شبیه خود

* Full Connected



* Star

• مهاجرت

* هر نسل؟ — هر چند نسل؟ — چه مدل؟ — استراتژی؟

* فاصله مهاجرتها

هر نسل: حالتی است که یک جمعیت داریم.

حتی وقتی که نرخ آن بالا باشد، جمعیتها بهتر نمی شود

پس هر چند نسل یکبار ← پارامتر آزاد ← نرخ

محضرضا: پراکنش سوزنی که متناسب با جمعیت

سرعت همگرایی

فاصله ← کم: بی توجهی به چند جمعیتی ← سرعت ↑ ← تنوع ↓

زیاد: بی اطلاع از سایر جمعیتها ← سرعت ↓ ← تنوع ↑

☑ ازدواج بین شهرهای مختلف

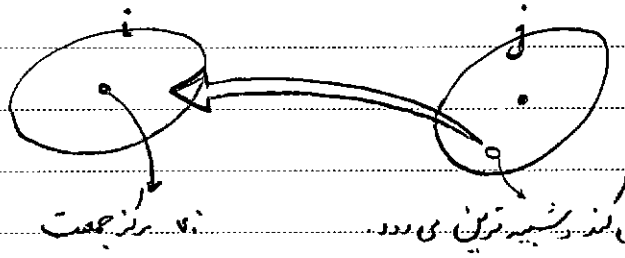
Subject:
Year. Month. Date. ( )

• بد Multimodal

* در جهت ← عمودی
* بین جهتها ← متنوع

• مهاجرت

* بین ۲۵ تا ۱۵۰ نفر
* بر بار ۲ تا ۵ فرد
* شباهت نژاد ← مثلاً فاصله اقلیدسی کم بین آنها
* جایگزین کردن ← اقتصادی ← multimodal ← براساس شباهت
  بدترینها ← برداشتن مداری

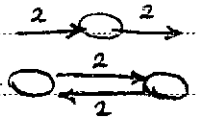


i: برای جهت

فاصله را با i محاسبه می کند و شبیه ترین را می دهد.

* آرشیو کردن ← برداشتن مداری ✓
  هزینه ← multimodal ← ممکن است مال این Nich نباشد ✗

* هر مرحله هر چند بدون مهاجرت انجام جایگزین می شود.



Subject:

Year. Month. Date. ( )

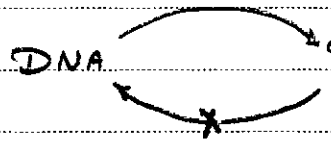
Subject:

Year Month Date ( )

از الگوریتم خود

* الگوریتم Memetic

- EA ← صنف در جستجوی محلی
- LS ← صنف در جستجوی پراکنده Local Search
- در الگوریتم‌های تکاملی یادگیری منتقل نمی‌شود، چون:



- خلق و حذف قابل یادگیری و انتقال به نژادان است. روانشناسان به آن Mem می‌گویند.
- در الگوریتم هم سعی می‌کنند این را مدل کنند منتقل کنندهٔ خلق و حذف در آن، معمولاً الگوریتم‌های جستجوی محلی هستند.
- معمولاً در الگوریتم‌های Memetic از Hill Climbing استفاده می‌کنند.
- بدین ترتیب که پس از هر نسل تکامل Memetic، چند قدم جستجوی محلی انجام می‌دهند.
- تعداد این قدم‌ها مورد بررسی است ← خطی، نمایی، ...
  - تطبیقی
  - ثابت!
- از جدیدترین روش‌ها استفاده از Simulated Annealing در ترکیب با تکامل است در min‌های محلی گیر نمی‌کند و می‌تواند از آن‌ها رد شود.

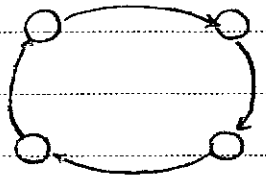
* ادامه روش‌های چندمدلی

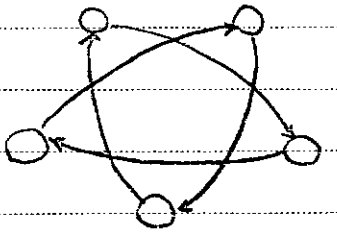
- Multimodal = niching = چندمدلی = حفظ تنوع
- هدف: پیدا کردن مدل‌های داخل niche‌ها و تنوع بین niche‌ها

- چندجمعیتی ← جمعیت با تعداد ثابت $\mu_1, \ldots, \mu_p$
  - مدل ← مطلقه ← ناصفر 1
    - (از نظر بهینگی)
    - ناصفرهای بیشتر از 1

Subject:

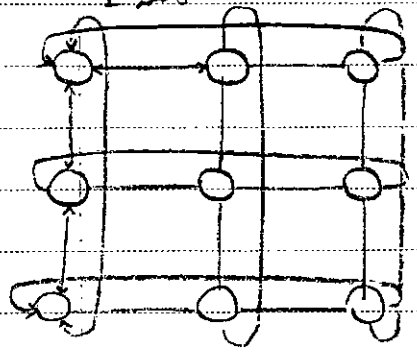
Year Month Date ( )





فاصله 2

فاصله 1



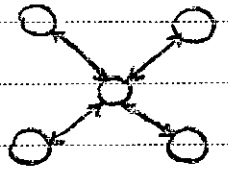
مدل همسایگی

همسایگی دسته ای

همسایگی تقاطعی

Full - connected

star



مهاجرت

هرچند نسل یکبار؟ ← 25 ~ 150 Generation = 1 Epoch

← مقدار بالا ← عدم همگرایی

چند موجود؟ ← 2 ~ 5 individual

← مقدار بالا ← همگرایی نسبتا دیر

← از بین رفتن تنوع

چه موجوداتی؟ ← پردازش موازی ← متناسب با شایستگی

← حفظ تنوع ← شباهت ← ژنتیپی ← فاصله اقلیدسی

← فنوتیپی ← شایستگی

جایگزین چه موجوداتی ← پردازش موازی ← بدترین

← حفظ تنوع ← متفاوت ترین

Subject:

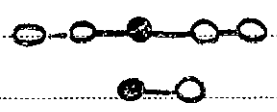
Year. Month. Date. ( )

• جمعیت با همسایگی محلی

یک بعدی ← رینگ

نقاط فرعی نداخت ← تعادل همسایگی برقرار نیست

full ring vs. half ring



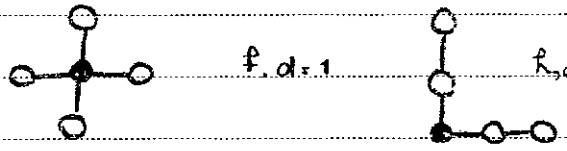
full , dist = 2

half , dist = 1

دو بعدی ← رینگ با نقاط فرعی نداخت

full cross

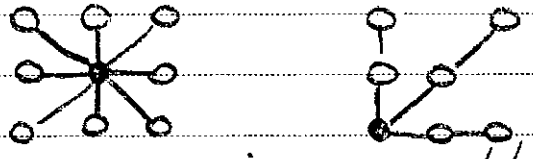
half cross



f, d=1 r, d=2

full star

half star



فاصله ← کوچک ← سرعت انتشار اطلاعات ↓

تنوع ↑

بازترکیبی و انتخاب بازماندگان داخل همسایگی

انتخاب والدین ← نقاط فرعی نداخت به عنوان والد اول $\lambda/2$

یک طرفی از داخل همسایگی به عنوان والد دوم $\lambda/2$

انتخاب بازماندگان ← درون همسایگی

می تواند جایگزین بدترین شود.

جایگزین کردن فرزند بهتر از بدترین به جای بدترین

جایگزین کردن فرزند بهتر از بدترین به جای والدین

نقاط فرعی نداخت

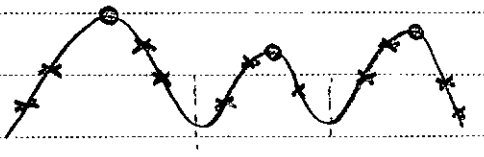
PqPCO

جایگزین کردن فرزند بهتر از بدترین به جای صفر نقاط فرعی همسایگی

Subject:
Year Month Date ( )

• علامت‌گذاری ← در کروموزوم چندین اضافه می‌شود که نشانه niche باشد.
Tagging ← اگر niche یکسان بودند با هم بازترکیبی شوند.
چندین اضافه

قله niche ← جستجوی محلی همان موجود

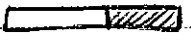


شباهت نزدیک
شباهت نزدیک

الگوریتم Memetic

۱- Local Search ژن‌ها را تغییر می‌دهد سعی می‌کند تا آن‌ها را بهبود بخشد و سپس جواب را کامل کند.

۲- در الگوریتم‌های Memetic جدیدی، ژن‌ها دست نخورده می‌مانند، سپس LS بهترین شایستگی بدست آمده را در نسل‌های متوالی اضافه می‌کند. (مقدار یادگیری)



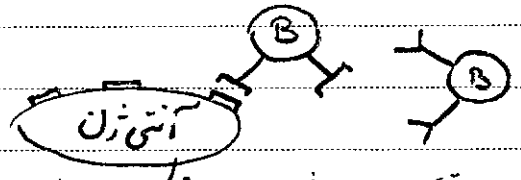
AIS *

• آنتی ژن: عوامل بیماری‌زا
• آنتی بادی: دفاع بدن در برابر آنتی ژن
• لایه‌های سیستم ایمنی
① ذاتی: پوست، pH خون، اسید معده، ...
② تطبیقی: سلول‌های B، سلول‌های T

• سلول‌های APC آنتی ژن را می‌بلعد و پپتیدهای را آزاد می‌کند.
• سلول‌های T به این پپتیدها می‌چسبند.

Subject:
Year. Month. Date. ( )

- روی آنتی ژن ها گیرنده‌های پروتئینی روی سطح خود هستند که توسط سلول‌های B و سلول‌های T شناسایی می‌شوند. ← نشانه = پیتید
- براساس نشانه پروتئینی آنتی ژن مد بدن مقابله ایجاد می‌شود.



- سلول B با آنتی بادی خود، شکل نشانه‌های پروتئینی را می‌شناسد و تحریک می‌شود و پلاسما تولید می‌شود.
- از پلاسما، گیرنده‌های مربوط به ویروس (آنتی بادی) تولید می‌شود به تعداد زیاد و به ویروس می‌چسبد و پس از آن سلول‌های ماکروفاژی توانند ویروس را ببلعند.

- سلول‌های بدن نیز نشانه‌های پروتئینی دارند.
- سلول‌های B در مغز استخوان تولید می‌شوند. فرق بین سلول‌های خودی و غیرخودی را نمی‌دانند. آنها فقط به نشانه حساس است.
- سلول‌های T در دوران تمرین هستند. دوران تمرین تمام انواع سلول‌های خودی را دیده دارد. اگر سلول‌های T آنها را شناخت از بین می‌روند، در غیر این صورت بالغ می‌شوند و بیرون می‌آیند.
- وقتی سلول T با عامل خارجی برخورد می‌کند، از خود عامل محرک آزاد می‌کند که سلول B را تحریک می‌کند و سلول B به آن عامل حمله می‌کند.
- گیرنده‌های B سه‌بعدی هستند و گیرنده‌های T خطی هستند.
- APC ها جزو سیستم ایمنی ذاتی هستند، به طور خودی می‌دانند که کدام سلول خودی هستند (MHC I) و کدام غیرخودی (MHC II)
- این APC سلول‌های مشکوک به غیرخودی را می‌بلعد و گیرنده‌های آنها از حالت سه‌بعدی به خطی تبدیل می‌کند و به روی سطح سلول می‌آورد (توسط پروتئین MHC) حال گیرنده‌های سلول T می‌توانند آنها را تشخیص دهد.
- تولید انواع سلول‌های خودی در تیموس به صورت شانسی است.

Subject:

Year. Month. Date. ( )

• آنتی بادی لازم جهت کاملاً پپتید منطبق شود.

• عمر سلولهای B و T در تا سه روز است. ولی وقتی T به سلول حافظه تبدیل شود، عمرآن تا ۱۵ سال می ماند.

• سلولهای حافظه دقیقاً تقسیم سلولی می شوند به یک سلول پلاسما و یک سلول حافظه دیگر تقسیم می شود.

• انتخاب Clonal ، آنتی بادی ای که بیشترین تطبیق با آنتی ژن دارد فعال می کند.

• در مغز استخوان تمام سلولهای B و T لازم تولید می شود و با یک شعاع E با آن فاصله دارد.

• آنتی ژن ممکن است با آنتی بادی فاصله E را داشته باشد.

• در برابر آنتی ژن ، تمام آنتی بادی ای که با آن شباهت دارد را فعال می کند. این سلول به تناسب شباهتی که با آنتی بادی دارد تولید مثل می کند. این سلول B جدید پس جهش می یابد متناسب با میل شایستگی. سپس مراحل تکرار می شود.

• Clonal Selection

۱- سلول B اولیه

۲- شباهت با آنتی ژن ← ↑

۳- متناسب با ↑ تولید مثل می کند.

۴- متناسب با میل ↑ جهش می یابد.

۵- برو به ۱

• Clonal Selection ← جستجوی محلی بد

— جستجوی محلی بسیار سریع

— سریعاً به min محلی می رسد.

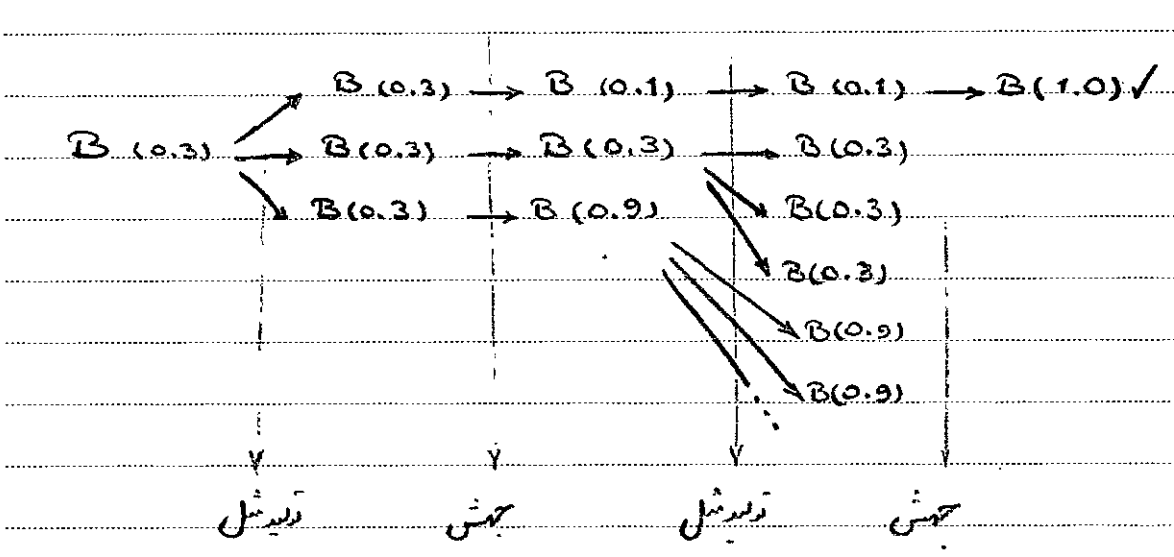
• Bone Marrow ← جستجوی عمومی بد

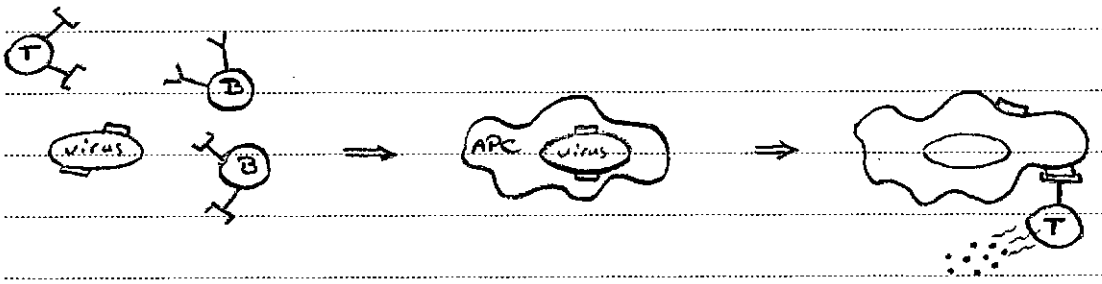
— ۷۰٪ جمعیت تولید شده شبیه جمعیت موجود

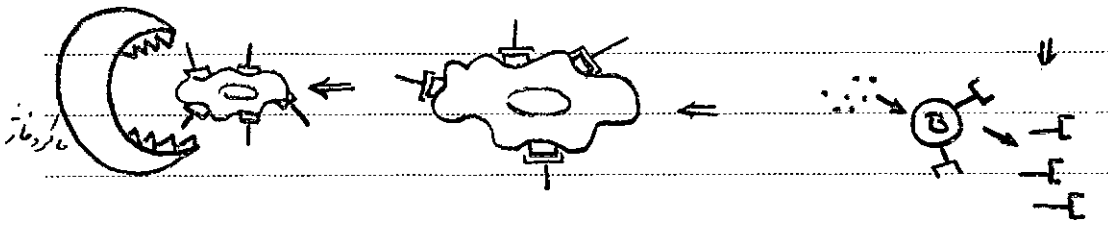
— ۳۰٪ سلول های تصادفی

Subject:

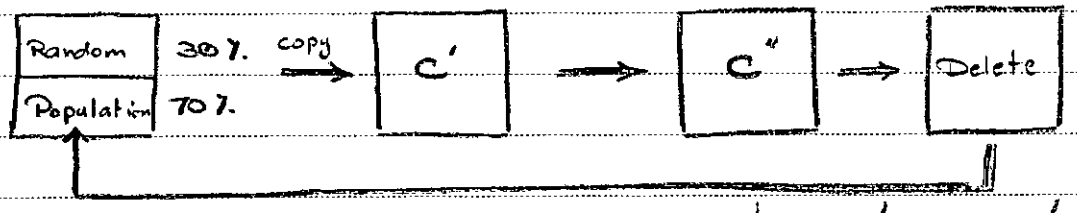
Year. Month. Date. ( )







AiNet



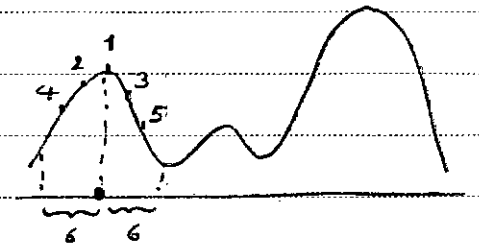
روشی که حذف می کند: سمدایی

روشی که حذف می کند: نزع ← موجودات شبیه به هم به غیر از بهترین حذف می شوند

Subject:
Year Month Date ( )

* Fitness Sharing ← شایستگی به اشتراک گذاشته می‌شود.

برای محاسبه شایستگی یک شعاع همسایگی برای هر نقطه در نظر گرفته می‌شود. در این شعاع، تعداد موجودات شمرده می‌شود و هرچه موجود دورتر باشد اهمیت کمتری می‌یابد.



شایستگی موجود، برابر با شایستگی آن تقسیم بر این تعداد و مقدار همسایه است.

$$f'(i) = \frac{f(i)}{\sum_{j=1}^{\mu} sharing(d_{ij})}$$

$$sharing = \begin{cases} \frac{1-d}{\sigma} & d < \sigma \text{ شعاع} \\ 0 & o.w \end{cases}$$

قله‌ای که شایستگی بالاتر دارد تعداد بیشتری موجود در آن قرار می‌گیرد، و شایستگی آنها کاهش می‌یابد. چون از طرفی شایستگی بیشتر جمعیت بیشتری را جذب می‌کند و از طرفی جمعیت باعث کم شدن شایستگی می‌شود.

این روش برای حل مشکل تجمع جمعیت در بهترین قله است. در هر جمعیت موجود جزاهدبرد و موجودات هر جمعیت شایستگی بالایی دارند. بهتر است برای niche سیمای متفاوت در نظر گرفت. ولی مانعی هم که چه سیمای مناسب ← خود تطبیقی خوب است؟

PAPCO

Subject:

Year. Month. Date. ( )

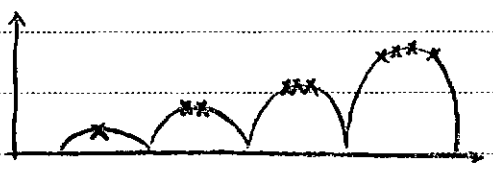
* Crowding ← در این روش معمولاً با روش سهمیه‌ای استفاده می‌شود، مثل آنست.
از دو والد، دو فرزند به وجود آمده است. فاصله هر والد تا فرزند خود دیگری محاسبه می‌شود. در صورتی که نامساوی زیر برقرار باشد:

$$d(P_1, O_1) + d(P_2, O_2) < d(P_1, O_2) + d(P_2, O_1)$$

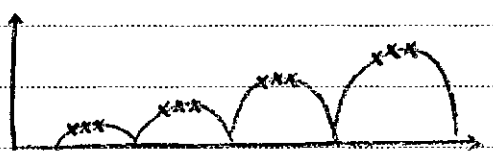
آنگاه اگر شایستگی فرزند کمتر از شایستگی والد بود، آنها جایگزین والد خود می‌کنیم.
اگر معادله برقرار نشود، آنگاه هر والد با فرزند دیگری مقایسه می‌شود و در صورت کمتر بودن فرزند آنها جایگزین می‌شوند.
در این حالت فرزندان به هم شبیه خواهند بود. nich را خراب می‌کند.

if ($d(P_1, O_1) + d(P_2, O_2) < d(P_2, O_1) + d(P_1, O_2)$)
  if ($f(O_1) > f(P_1)$) then $P_1 = O_1$
  if ($f(O_2) > f(P_2)$) then $P_2 = O_2$
else
  if ($f(O_2) > f(P_1)$) then $P_1 = O_2$
  if ($f(O_1) > f(P_2)$) then $P_2 = O_1$

* مقایسه Crowding و Fitness Sharing



Fitness Sharing:
تعداد افراد متناسب با nich



Crowding:
تعداد افراد نامتناسب با nich

Subject:
Year. Month. Date. ( )

* مسائل چندهدفی

• در مسائل تک هدفی تنها یک Fitness Function داریم

• در مسائل چندهدفی مجموعه‌ای از توابع هدف داریم که می‌توانند متضاد هم باشند.
می‌خواهیم این توابع را همزمان min یا max کنیم

$$\min \{ f_1(x), f_2(x), \dots, f_m(x) \}$$

• می‌توان هر دو حالت min و max را داشت، همه را باید به یک شکل کرد.

$$\min f_1(x) \quad \max f_2(x) \Longrightarrow \begin{cases} \min \{ f_1(x), -f_2(x) \} \\ \min \{ f_1(x), M - f_2(x) \} \\ \min \{ f_1(x), 1/(f(x)+\epsilon) \} \end{cases}$$

• روش اول:

$$f(x) = \sum w_i \cdot f_i(x)$$

- پیدا کردن وزن‌های بهینه x
- وزن ↑ ⇐ تمایل به آن می‌پردازد.
- وزن ↓ ⇐ بی‌توجهی
- بازه f_i : متناسب است x
- مقدار تمام ادله این 0 و 1 بیاوریم آن‌ها را نرمال کنیم.

$$0 \le f_i(x) \le 1, \quad \sum w_i = 1 \Longrightarrow 0 \le f(x) \le 1$$

- جواب‌هایی که پارامتر روی آن‌ها اختلاف نمی‌دارد ولی دیگری اختلاف زیادی دارد به خوبی دیده نمی‌شود.
- باید این روش با ترکیبات مختلف وزن آزمایش شود.

Subject:
Year. Month. Date. ( )

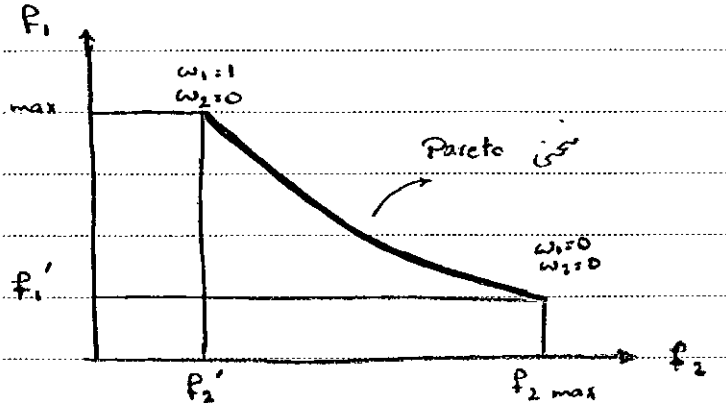
- باید برای w_1 و w_2 مقادیر مختلف امتحان شود.

$(w_1, w_2) = (0.1, 0.9), (0.2, 0.8)$

- ایده Pareto

- الگوریتم تعاملی چندهدفی که در نهایت جمعیت بهینه شده به دست‌هایی که w های مختلف دارد.

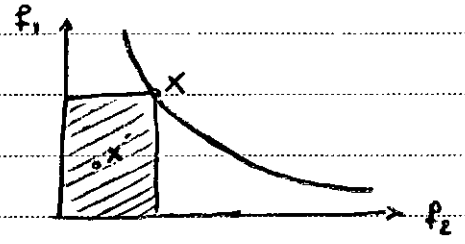
- دو تابع هدف f_1 و f_2



- غالب بودن: x_1 بر x_2 غالب است در صورتی که حداقل یک جواب مانند i پیدا شود که بدان $f_i(x_1) > f_i(x_2)$ و برای بقیه اهداف مانند j

$f_j(x_1) \geq f_j(x_2)$

- در بهینه Pareto موجود نقطه x بر تمام ناحیه‌های شدنی غلبه دارد.



x بر x' غلبه دارد $\iff$

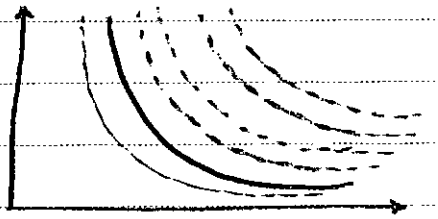
$f_i(x) > f_i(x') \quad \exists i$

$f_j(x) \geq f_j(x') \quad \forall i \neq j$

Subject:

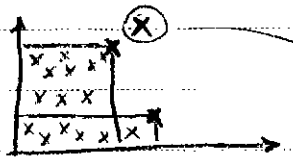
Year. Month. Date. ( )

- در تکامل تکامل دادن Pareto به سمت منحنی بهینه آن است.



- مزایای حل کردن مسائل چندهدفی با تکامل
  - نیاز به حدسی در مورد روابط بین متغیرها نیست تا ترکیب خوبی بدست آید.
  - مقید به شروط عادی مسائل چندهدفی نیست.
    - محدب بودن
    - مشتق پذیری
  - بسیاری تمام جواب‌های روی Pareto را بدست می‌آورند.
    - هم به Pareto بهینه نزدیک می‌شوند.
    - تمام جواب‌های روی Pareto را بسیاری دارد و جمعیت نهایی تکاملی آنها را نشان می‌دهد.

- برای حل مسائل Pareto
  - به یک جواب بهینه نشود و تمام جواب‌ها را داشته باشد ⟸ Multi Modal
  - نیاز به تنوع وجود دارد.
  - باید شایستگی را براساس Pareto تعریف کرد
    - افرادی که غالب هستند شایسته‌ترند.
  - بهترین‌ها (Pareto) آرشیو شوند ⟸ روش فراوانسار نباشد
    - ☑ شایستگی = تعداد افرادی که بر آن غالب شوند. ✓
    - ☑ شایستگی = به چند متجاوز دیگر غلبه می‌کند. ✗
    - مثال



Pareto نسبت به هیچ Pareto غالب است

Subject:
Year. Month. Date. ( )

• پیشنهاداتی برای شرط Pareto
— شایستگی : تعداد موجوداتی که بر آن غالب است ، ...
— تنوع : Fitness Sharing ، ...
— آرشیو کردن : (μ+λ) با جایگزینی ، آرشیو موجودات غالب

☑ برای توضیح بر الگوریتم چند پدری باید این سه مورد مشخص شود.
معیار خوبی دوش ← چند Pareto رابیشتر می دهد.
چند Pareto بهینه نزدیک می شود.

• محدودیت ها برای شایستگی

$$\min \{ f_1(x), \dots, f_m(x) \}$$
$$g_i(x) = 0 \qquad i = 1, \dots, q$$
$$h_i(x) = 0 \qquad i = q+1, \dots, m$$

تابع را می توان به این روش نشان داد.

☑ $x < t \longrightarrow x > -t$

☑ $c_{min} < x < c_{max} \longrightarrow c_{max} - x > 0$
$x - c_{min} > 0$

• برای کنار آمدن محدودیت ها در تکامل
① فرزندانی که محدودیت ها را ارضا نمی کنند حذف کنیم ← تنوع ↓ :(

② محدودیت ها را با یک روش وارد تابع هدف کرد.

$$f(x) = f(x) + \sum w_i \cdot f_i(x)$$

$$f_i(x) = \begin{cases} g_i(x) , & 1 \le i \le m \\ |h_i(x)| , & q+1 \le i \le m \end{cases}$$

Subject:
Year. Month. Date. ( )

w بزرگ ← موجوداتی را تولید می‌کند که فقط محدودیت را ارضا کنند.
w کوچک ← موجوداتی که زیاد به محدودیت توجهی ندارند. ← w ؟!

* پارامترها در الگوریتم‌های تکاملی

• دو راهکار

تنظیم قبل از اجرا ← Parameter Tuning
تنظیم حین اجرا ← Parameter Control

• Tuning

- محاسبه مقدار بهینه قبل از اجرا
- ماهیت تصادفی الگوریتم ← درباره مقدار بهینه صحبت
  ↙ مقدار بهینه برای مشروط حالات است.
- هزینه زیاد طراحی ← محاسبات زیاد
  ↙ آزمایشات زیاد

Orthogonal Genetic Alg. : 2^n آزمایش ← n آزمایش ☑
Fractional Factoriel

• Control

- محاسبه مقدار پارامتر در حین اجرا با توجه به ماهیت دینامیک آن
- بر سه نوع است ← Deterministic
  Adaptive
  Self-Adaptive

- Deterministic ← تغییر اندازه قدم کاهش : خطی
  تعیین اندازه وزن در ارضای نسل : نمایی
- Adaptive ← تغییر اندازه قدم : تا نزل $\frac{1}{5}$ موفقیت
  تغییر جزیی : تغییر جزیی نسبت به تعداد عضوهایی که نسل را ارضا می‌کنند.

Subject:
Year. Month. Date. ( )

Self - Adaptive ← اندازه قدم: ثابت ، چند ؟
اندازه جریمه: ثابت ، چند ؟

• مزایا و معایب

Deterministic ← :(اندازه قدم مناسب ؟!
Adaptive ← :(پیاده کردن feedback مناسب
:) ← اگر feedback خوب باشد، شبیه کلیه است.
Self - Adaptive ← :(ممکن است پارامترها را بگیریم ولی دوم مرحله‌ات نیز دارد.

$F = f + \frac{w}{\gamma} g$

* الگوریتم‌های تصمیم + گرادیان

• روش‌های جستجو ← لاماركی ← توالی تغییر می‌کند و به نسل بعدی می‌رود.
داروینی

• محاسبه گرادیان ← FD ← هزینه محاسباتی بالا
در همه حالات جواب می‌دهد.

Analytic Gradient — AG ← دقت بالا
شرایط خاص
ستون مستقیم

SPSA ← تخمین احتمالاتی با توزیع برنولی
دقت کمتر
کمترین محاسبات سالیانه

• روش‌های گرادیان ← تندترین شیب (Steepest Descend)
روش نیوتن ← محاسبات بالا
روش شبه نیوتنی ← تخمین تابع Hessian
روش DFP

Subject:
Year. Month. Date. ( )

Classifier Sys ✱

- ① محیط Sense می‌شود.
② با کمک قانون تطابق داده می‌شود. ← Match Set
③ عمل براساس قانون‌ها عمل می‌کند.

- چند قانون Match می‌شود.
↙ کدام قانون اجرا شود ← شایستگی (Strength)
↙ با یادگیری تقویتی اصلاح می‌شود.

- قانون نامناسب است
↙ استفاده از Internal Message

- قانون با الگوریتم تکاملی GA تولید می‌شود.
↙ Pitt (پیتزبورگ) : هر کروموزم مجموعه‌ای از قانون است.
↙ مزیت : شایستگی کل سیستم تعیین می‌شود و هر عضو جداگانه نیاز ارزیابی نیست.
↙ Michigan : هر کروموزم یک قانون است.
↙ عیب : شایستگی تمام مجموعات تعیین می‌شود و تخلص یک عضو نیست.
↙ راه حل : niching ← match set
action set ↙

- مراحل CS
↙ آموزش : عمل تصادفی از فضای match set
اجرا : عمل با استفاده از قدرت قانون

- هر Episod یکبار الگوریتم GA یک قانون را اصلاح می‌کند.
- اگر Match List خالی باشد، قانون را برای تطابق با محیط می‌سازیم و action آن تصادفی

Subject:
Year. Month. Date. ( )

* ادامه Classifier System

ZCS .

- فایل تا زن <State , Action , Strength>

r_i : < C:a → S >

- Match Set : تمام قوانینی که قسمت شرط با محیط مطابقت دارد.
- Action Set : قوانینی از MS که دارای action یکسانی هستند.
- برای هر عمل یک AS ساخته می‌شود و برای آن یک قدرت در نظر می‌گیریم

$$P_{a_i} = \sum_{r \in AS_{a_i}} \frac{S_r}{n\,AS_{a_i}}$$

- در فاز آموزش، عمل را به طور تصادفی انتخاب می‌کنیم.
- در فاز عملکرد، عملی را انتخاب می‌کنیم که بیشترین P را دارد.

وضعیت : 0101

MS
- 01#0 : 01 → 10 ⇝ AS_1
- 01## : 10 → 15 ⇝ AS_2
- #101 : 00 → 12 ⇝ AS_3
- 010# : 01 → 15 ⇝ AS_4

$P_{a_i} = \frac{10+15}{2}$

- AS رابطی است که به نتایج عمل مربوط به آن اجرا شده است.

$r_i \Rightarrow MS \Rightarrow AS_{a_1}, \ldots, AS_{a_2} \Rightarrow AS$

- در فاز آموزش دقتی عملی انتخاب می‌شود، قدرت آن با پاداش تنبیهی تنظیم می‌شود.

Subject:
Year. Month. Date. ( )

I. $\forall r \in MS^t \notin AS^t \Rightarrow S_r = \gamma . S_r \quad , \quad 0 < \gamma < 1$

II. $\forall r \in AS^t \Rightarrow S_r = (1-\beta) S_r \quad , \quad 0 < \beta < 1$

$\Rightarrow \alpha = \sum_{r \in AS^t} \beta . S_r$

III. $\forall r \in AS^{t-1} \Rightarrow S_r = S_r + \frac{\gamma \alpha}{n_{AS^{t-1}}} \quad , \quad 0 < \gamma < 1$

IV. $\forall r \in AS^t \Rightarrow S_r = S_r + \frac{\beta R}{n_{AS^t}}$

I. اگر عمل مربوط به این AS نباشد، باید مقدار این عمل کم شود. اینکار با ضریب کاهش γ (معمولاً 0.9) انجام می‌شود.

II. اگر عمل مربوط به این AS باشد، کمی از مقدار آن مربوط عمل مرحله قبل است. بنابراین این مقدار α است.

III. اگر عمل مرحله t مربوط به AS^t باشد، کمی از مقدار آن به طور مساوی بین تمام عملهای منجر به آن $(t-1)$ تقسیم می‌شود.

IV. در نهایت پاداش به عمل مربوط می‌رسد.

XCS.

$r_i : \langle C_r : a_r : p_r : \epsilon_r : F_r \rangle$

خطای پیش بینی شده ← امید پاداش پیش بینی شده

$MS \Rightarrow AS_{a_i} \Rightarrow AS^t$

$p_{a_i} = \sum_{r \in AS_{a_i}} w_{r_i} . p_r$

Subject:
Year.. Month. Date. ( )

$$w_{r,i} = \frac{F_r}{\sum_{r \in AS} F_r}$$

وزن با شایستگی متناسب است.

شایستگی در XCS برخلاف ZCS همان دقت نیست.

شایستگی در اینجا همان دقت است که عبارتست از نزدیک بودن به پیش بینی.

$$F_r' = (1-\beta) F_r + \beta K'_r$$

$$K'_r = \frac{K_r}{\sum_{r \in AS} K_r}$$

دقت نرمال شده

$$K_r = e^{(\ln \alpha)\frac{\varepsilon_r - \varepsilon_0}{\varepsilon_0}}$$

خطای پیش بینی شده

دقت ↑ ⇔ خطا ↓

$$P_r = P_r + \beta(P - P_r) = (1-\beta) P_r + \beta P$$

پاداش دریافتی

$$\varepsilon_r = \varepsilon_r + \beta(|P - P_r| - \varepsilon_r) = (1-\beta)\varepsilon_r + \beta(|P - P_r|)$$

خطا

$$P = (1-\gamma) P_{a_1} + \gamma R$$

- برای استفاده از این روش در حالت تکمیلی باید چند Episod بین دو اجرای GA فاصله باشد تا XCS همگرا شود.

پس از آن GA وارد می‌شود و Cross Over قانون جدیدی می‌سازد.

در قانون هیچ بعدی در CS نداشته باشد، آنرا در GA می‌سازیم و بعضی از آنها را don't care می‌کنیم.

در don't care شکل مرتب عوض نشود، مناسب نیست.

برای بهتر شدن علاوه بر 1 ⇄ 0 باید 1 ⇄ # و 0 ⇄ # تبدیل شود و بعکس.

P4PCO

Subject:

Year. Month. Date. ( )

در XCS از GA یا AS تنها، استفاده می‌کنیم (چندجمله‌ای)

در XCS از fitness sharing استفاده می‌شود

Subject:
Year. Month. Date. ( )

۵ خلاصه تکاملی

* مزایای EA

عمومی بودن ـ پذیرش تابع هدف قابل قبول ـ جواب‌های قابل قبول ـ پیشنهاد چندین جواب ـ جوابی

* معایب

عدم ضمانت همگرایی در زمان محدود ـ ضعف تئوری ـ پارامترهای آزاد ـ کند ـ نیاز به تابع هدف محدب یا تغییر تابع هدف

* تئوری تکامل

- اصل تئوری داروین: تنوع، انتخاب، تضاد
- علت تکامل: محدودیت منابع (استعداد بقیه، بیماری‌ها، رقابت)
- انتخاب ← ماندن بقای شایسته‌ترین‌ها
- تضاد ← تولید ژن‌های جدید ← مخالف روند طبیعت
- تنوع ← تولید حالات زیاد و در نتیجه بهترین حالت
- تکامل ← شدن تمام هم‌تراز کردن ژن‌های خوب پدر و مادر و تولید فرزند شایسته‌تر

Subject:
Year Month Date ( )

* الگوریتم تکاملی

• جمعیت اولیه ← تصادفی باشد است ✓ ، heuristic ✗

• شرط خاتمه ← تعداد نسل
  - تعداد نسل ← ممکن است شایستگی بهینه مشخص نباشد
  - همگرایی ← در k نسل قبل مقدار یا متوسط f تغییر محسوسی نکند
  - همگرایی ← در k نسل قبل واریانس f از حد آستانه‌ای کمتر باشد

• بازنمایی ← باینری ، اعداد صحیح ، اعداد حقیقی ، FSM ، درخت ، گراف

• همگرایی زودرس ← همگرایی به یک نقطه بهینه در نسلهای ابتدایی ← نزدیک شدن f به هم ← از بین رفتن تنوع

• سکون ← نزدیک شدن متوسط به هم ← از بین رفتن تنوع

• انتخاب والدین ← جمعیت محلی ، همسایگی ، چند جمعیتی

• استراتژی ها

	تضمین همگرایی	نزاز میمیم محلی	سرعت	حافظه سابقه پارامتر ع	بهینه متحرک
μ, λ	✗	✓	↓	✗	✓
$\mu + \lambda$	با فرض انتخاب شایسته ترین‌ها	✗	با شایسته سالاری ↑	✓	✗

• روشهای انتخاب ← تصادفی
  ← برشی ← شایسته سالاری
  ← متناسب با شایستگی ← SUS , RW ← همگرایی زودرس و سکون
  ← مقیاس بندی ← خطی ($af+b$ ، برش سیگما) و غیرخطی (توانی ، بولتزمن)
  ← رتبه بندی ← بیاز محاسبه دقیق شایستگی هست
  ← رتبه بندی تعدیل شده ← خطی و غیرخطی
  ← مسابقه

• حالتهای ویژه ← Generational Model ← GA(μ, λ) ← عمر موجود = 1 نسل
  ← Steady State ← EA($\mu+1$)
  ← Generation Gap ← تنها درصدی از جمعیت جایگزین می‌شود
  ← Elitism ← آرشیو کردن بهترین

PAPCO

Subject:

Year. Month. Date. ( )

* معیارهای ارزیابی

۱- فشار انتخاب (Selective Pressure) $SP: \frac{P_b}{\frac{1}{\lambda}\sum P_i}$

• احتمال انتخاب بهترین موجود در مقایسه با متوسط احتمال انتخاب بقیه موجودات

۲- بایاس (Bias) $B_i: |P_i - \frac{f_i}{\frac{1}{\lambda}\sum f_i}|$

• قدر مطلق اختلاف بین شایستگی نرمال شده موجود و احتمال انتخاب آن موجود

۳- پخش (Spread) $N_i = \mu_x P_i$

• تعداد کپی‌های یک موجود پس از انتخاب

۴- درجه اول تنوع (LoD)

• تعداد موجودات انتخاب شده به کل موجودات

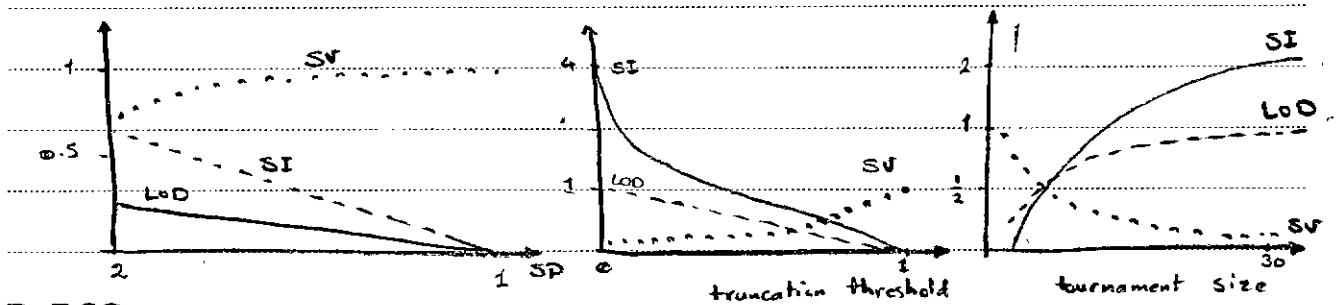
۵- واریانس انتخاب (Selection Variance)

• امید ریاضی واریانس f جمعیت وقتی که انتخاب بر روی یک جمعیت با توزیع نرمال انجام شود.

۶- تمایل انتخاب (Selection Intensity)

• امید ریاضی متوسط f جمعیت بعد از اعمال عامل انتخاب بر روی جمعیت با توزیع نرمال

	معیار بزرگ	معیار کوچک	صفر	اول الگوریتم	آخر الگوریتم
SP	شانس بالای انتخاب موجود برتر	—	—	↓	↑
BI	انتخاب نامناسب با شایستگی	—	انتخاب متناسب با شایستگی	—	—
Spread	کپی‌های بیشتر و نسل بعد	—	—	حداقل ۱	N_b ↑
LoD	کاهش تنوع		سکون (؟)	↓	↑
SV	تنوع زیاد	تنوع کم	سکون	↑	↓
SI	دور از جواب	نزدیک جواب	جواب بهینه (؟)		کمتر شود ↓



Subject:

Year. Month. Date. ( )

	ES	EP	GA	GP
بازنمایی	اعداد حقیقی	FSM ، اعداد حقیقی	باینری	درختی - غیرخطی
جمعیت اولیه	μ ، λ ، تصادفی	تصادفی / بی‌اریبی	μ	تعداد حقیقی زیاد ← full method / grow method / half-n-half
انتخاب والدین	تصادفی / بی‌اریبی	همه	مناسب با شایستگی	برده‌بری ← 80٪ برده اول و بقیه دوم
بازترکیبی	نیست و بیشتر نیری	ندارد	باینری (تهم)	۰.۱ برد، ۰.۹ نود ← ترکیب زیردرخت
جهش	$N(0,\sigma^2)$	روی FSM	Flip	تنها برای جمعیت‌های کم ← تغییر زیردرخت
انتخاب بازماندگان	شایسته‌سالاری	مسابقه	همه	steady state
استراتژی بقا	ES $(\mu+\lambda)$	EP $(\mu+\mu)$	GA (μ,μ)	GP $(\mu+1)$
ویژگی‌های خاص	گام جهش تطبیقی	تابع جهش تطبیقی / برازندگی تورنامنتی لذت	عملگر reproduction	حافظه دفتر کم
	تا نزدیک $\frac{1}{5}$ موفقیت برای $(1+1)$ ES ، unimodal توابع			

* تفاوت‌های ES ، EP

	استراتژی بقا	انتخاب بازماندگان	بازترکیبی	تغییرات σ
ES	$\mu+\lambda$	شایسته‌سالاری	✓	σ^2
EP	$\mu+\mu$	مسابقه	✗	$\sqrt{var} = \sigma$

* جهش تطبیقی

۱ ، σ ⟶ $\sigma' = \sigma \times \exp\{\tau \times N(0,1)\}$ ⟶ $\tau \propto 1/\sqrt{n}$

n ، σ ⟶ $\sigma_i' = \sigma_i \times \exp\{\tau' \times N(0,1) + \tau \times N_i(0,1)\}$ ⟶ $\tau \propto 1/\sqrt{n}$ ، $\tau' \propto 1/\sqrt{2\sqrt{n}}$

n × σ ، n × α ⟶ کوواریانس

* حل مشکل چاقی

Hill Climbing ، Size Limiting ، Parsimony ، Code Editing .