به نام خدا

مقدمه ای بر الگوریتم های ژنتیک (۱)

Genetic Algorithms

گردآورنده :

امیر علی بلورچیان

کلمات کلیدی

الگوریتم ژنتیک، GA

چکیده

در این مقاله سعی شده است تا خواننده با مفاهیم اولیه الگوریتم های ژنتیکی آشنا شود.

۱.الگوریتم ژنتیک چیست؟

الگوریتم های ژنتیک از اصول انتخاب طبیعی داروین برای یافتن فرمول بهینه جهت پیش بینی یا تطبیق الگو استفاده می کنند.الگوریتم های ژنتیک اغلب گزینه خوبی برای تکنیک های پیش بینی بر مبنای رگرسیون هستند.همان طور ساده،خطی وپارامتریک گفته می شود،به الگوریتم های ژنتیک می توان غیر پارامتریک گفت.

برای مثال اگر بخواهیم نوسانات قیمت نفت را با استفاده از عوامل خارجی وارزش رگرسیون خطی ساده مدل کنیم،این فرمول را تولید خواهیم کرد:قیمت نفت در زمان t=۱ ضریب نرخ بهره در زمان t+2ضریب نرخ بیکاری در زمان t+۱ ثابت . سپس از یک معیار برای پیدا کردن بهترین مجموعه ضرایب و ثابت ها جهت مدل کردن قیمت نفت استفاده خواهیم کرد.در این روش ۲ نکته اساسی وجود دارد.اول این روش خطی است و مسئله دوم این است که ما به جای اینکه در میان "فضای پارامترها"جستجو کنیم ،پارامترهای مورد استفاده را مشخص کرده ایم.

با استفاده از الگوریتم های ژنتیک ما یک ابر فرمول یا طرح تنظیم می کنیم که چیزی شبیه"قیمت نفت در زمان t تابعی از حداکثر ۴ متغیر است را بیان می کند. سپس داده هایی برای گروهی از متغیرهای مختلف،شاید در حدود ۲۰ متغیر فراهم خواهیم کرد.سپس الگوریتم ژنتیک اجرا خواهد شد که بهترین تابع و متغیرها را مورد جستجو قرار می دهد.روش کار الگوریتم ژنتیک به طور فریبنده ای ساده،خیلی قابل درک وبه طور قابل ملاحظه ای روشی است که ما معتقدیم حیوانات آنگونه تکامل یافته اند.هر فرمولی که از طرح داده شده بالا تبعیت کند فردی از جمعیت فرمول های ممکن تلقی می شود خیلی شبیه به این که بگوییم جرج بوش فردی از جمعیت انسان های ممکن است.

متغیر هایی که هر فرمول داده شده را مشخص می کنند به عنوان یکسری از اعداد نشان داده شده اند که معادل DNA آن فرد را تشکیل می دهند.

موتور الگوریتم ژنتیک یک جمعیت آغاز از فرمول ایجاد می کند .هر فرد در برابر مجموعه ای از داده ها ی مورد آزمایش قرار می گیرند و مناسبترین آنها شاید ۱۰ درصد از مناسبترین ها باقی می مانند.بقیه کنار گذاشته می شوند. مناسبترین افراد با هم جفتگیری (جابجایی عناصر DNA) وتغییر (تغییر تصادفی عناصر DNA) کرده اند.مشاهده می شود که با گذشت از میان تعدد ریادی از نسلها،الگوریتم ژنتیک به سمت ایجاد فرمول هایی که بیشتر دقیق هستند،میل می کنند.در حالی که شبکه های عصبی هم غیر خطی و غیر پارامتریک هستند،جذابیت زیاد الگوریتم های ژنتیک این است نتایج نهایی قابل ملاحظه ترند.فرمول نهایی برای کاربر انسانی قابل مشاهده خواهد بود،و برای ارائه سطح اطمینان نتایج می توان تکنیک های آماری متعارف رابر روی این فرمول ها اعمال کرد.فناوری الگوریتم های ژنتیک همواره در حال بهبود استفبرای مثال با مطرح کردن معادله ویروس ها که در کنار فرمول ها وبرای نقض کردن فرمول ها ی ضعیف تولید می شوندودر نتیجه جمعیت را کلاً قویتر می سازند.

مختصراً گفته می شود که الگوریتم ژنتیک یا GA یک تکنیک برنامه نویسی است که از تکامل ژنتیکی به عنوان یک الگوی حل مسئله استفاده می کند.مسئله ای که باید حل شود ورودی است و راه حلها طبق یک الگو کد گذاری می شودومتریک که تابع fitness هم نام دارد هر راه حل کاندید را ارزیابی می کندکه اکثر آنها به صورت تصادفی انتخاب می شوند.

الگوریتم ژنتیک GA یک تکنیک جستجو در علم کامپیوتربرای یافتن راه حل بهینه ومسائل جستجو است.الگوریتم های ژنتیک یکی از انواع الگوریتم های تکاملی اند که از علم زیست شناسی مثل وراثت، جهش،انتخاب ناگهانی ، انتخاب طبیعی و ترکیب الهام گرفته شده .

عموماً راه حلها به صورت ۲ تایی 0 و 1 نشان داده می شوند ولی روشهای نمایش دیگری هم وجود دارد.تکامل از یک مجموعه کاملاً تصادفی از موجودیت ها شروع می شود و در نسلهای بعدی تکرار می شود.در هر نسل،مناسبترین ها انتخاب می شوند نه بهترین ها.

یک راه حل برای مسئله مورد نظر،با یک لیست از پارامترها نشان داده می شود که به آنها کروموزوم یا ژنوم می گویند.کروموزوم ها عموماً به صورت یک رشته ساده از داده ها نمایش داده می شوند،البته انواع ساختمان داده های دیگر هم می توانند مورد استفاده قرار گیرند.در ابتدا چندین مشخصه به صورت تصادفی برای ایجاد نسل اول تولید می شوند. در طول هر نسل ،هر مشخصه ارزیابی می شود وارزش تناسب (fitness)توسط تابع تناسب اندازه گیری می شود

گام بعدی ایجاد دومین نسل از جامعه است که بر پایه فرآیندهای انتخاب ،تولید از روی مشخصه های انتخاب شده با عملگرهای ژنتیکی است:اتصال کروموزوم ها به سر یکدیگر و تغییر. برای هر فرد ،یک جفت والد انتخاب می شود.انتخابها به گونه ای اند که مناسبترین عناصر انتخاب شوند تا حتی ضعیفترین عناصر هم شانس انتخاب داشته باشند تا از نزدیک شدن به جواب محلی جلوگیری شود.چندین الگوی انتخاب وجود دارد: چرخ منگنه دار(رولت)،انتخاب مسابقه ای (Tournament)،... .

معمولاً الگوریتم های ژنتیک یک عدد احتمال اتصال دارد که بین 0.6 و 1 است که احتمال به وجود آمدن فرزند را نشان می دهد.ارگانیسم ها با این احتمال با هم دوباره با هم ترکیب می شوند . اتصال ۲ کروموزوم فرزند ایجاد می کند،که به نسل بعدی اضافه می شوند.این کارها انجام می شوند تا این که کاندیدهای مناسبی برای جواب،در نسل بعدی پیدا شوند. مرحله بعدی تغییر دادن فرزندان جدید است.الگوریتم های ژنتیک یک احتمال تغییر کوچک وثابت دارند که معمولاً درجه ای در حدود 0.01 یا کمتر دارد. بر اساس این احتمال ،کروموزوم های فرزند به طور تصادفی تغییر می کنند یا جهش می یابند.مخصوصاً با جهش بیتها در کروموزوم ساختمان داده مان.

این فرآیند باعث به وجود آمدن نسل جدیدی از کروموزوم ها یی می شود، که با نسل قبلی متفاوت است.کل فرآیند برای نسل بعدی هم تکرار می شود،جفتها برای ترکیب انتخاب می شوند،جمعیت نسل سوم به وجود می آیندو... .

این فرآیند تکرار می شود تا این که به آخرین مرحله برسیم.

شرایط خاتمه الگوریتم های ژنتیک عبارتند از:

- به تعداد ثابتی از نسل ها برسیم .
- بودجه اختصاص داده شده تمام شود(زمان محاسبه/پول).
- یک فرد(فرزند تولید شده) پیدا شود که مینیمم (کمترین)ملاک را برآورده کند.
- بیشترین درجه برازش فرزندان حاصل شود یا دیگر نتایج بهتری حاصل نشود.
- بازرسی دستی.
- ترکیبهای بالا.

توضیحی دیگر:

الگوریتم های ژنتیک قابلیت تبدیل فضای پیوسته به فضای گسسته را دارند. یکی از تفاوت های اصلی روش GA با روش های قدیمی بهینه سازی در این است که در GA با جمعیت یا مجموعه ای از نقاط در یک لحظه خاص کار میکنیم. در حالی که در روش های قدیمی بهینه سازی تنها برای یک نقطه خاص عمل میکردیم. این به این معنی است که GA تعداد زیادی از طرح ها را در یک زمان مورد پردازش قرار میدهد. نکته جالب دیگر این است که اصول GA بر پردازش تصادفی یا به تعبیر صحیحتر پردازش تصادفی هدایت شده (Guided Random) استوار است. بنابر این عملگرهای تصادفی فضای جستجو را را به صورت تطبیقی مورد بررسی قرار میدهند.

اصولا برای استفاده از GA باید سه مفهوم مهم زیر مشخص شوند:

- تعریف تابع هدف (Objective Function) یا تابع هزینه (Cost Function) .
- تعریف و پیاده سازی فضای ژنتیک (Genetic Representation) .
- تعریف و پیاده سازی عملگرهای GA .

۲.تاریخچه الگوریتم های ژنتیک

ایده اصلی الگوریتم های تکاملی در سال ۱۹۶۰ توسط Rechenberg مطرح گردید. الگوریتم های ژنتیک که منشعب از این نوع الگوریتم ها است، در حقیقت روش جستجوی کامپیوتری بر پایه الگوریتم های بهینه سازی و بر اساس ساختار ژن ها و کروموزوم ها است که توسط پروفسور Holland در دانشگاه میشیگان مطرح شد و پس از وی توسط جمعی از دانشجویانش توسعه یافت. همنطور که گفته شد ایده اساسی این الگوریتم انتقال خصوصیات موروثی توسط ژن‌هاست. فرض کنید مجموعه خصوصیات انسان توسط کروموزوم‌های او به نسل بعدی منتقل می‌شوند. هر ژن در این کروموزوم‌ها نماینده یک خصوصیت است. بعنوان مثال ژن ۱ می‌تواند رنگ چشم باشد ، ژن ۲ طول قد، ژن ۳ رنگ مو و الی آخر. حال اگر این کروموزوم به تمامی، به نسل بعد انتقال یابد، تمامی خصوصیات نسل بعدی شبیه به خصوصیات نسل قبل خواهد بود. بدیهیست که در عمل چنین اتفاقی رخ نمی‌دهد.

در واقع بصورت همزمان دو اتفاق برای کروموزوم‌ها می‌افتد. اتفاق اول موتاسیون (Mutation)است. موتاسیون به این صورت است که بعضی ژن‌ها بصورت کاملا تصادفی تغییر می‌کنند. البته تعداد این گونه ژن‌ها بسیار کم می‌باشد اما در هر حال این تغییر تصادفی همانگونه که پیشتر دیدیم بسیار مهم است. مثلا ژن رنگ چشم می‌تواند بصورت تصادفی باعث شود تا در نسل بعدی یک نفر دارای چشمان سبز باشد. در حالی که تمامی نسل قبل دارای چشم قهوه‌ای بوده‌اند. علاوه بر موتاسیون اتفاق دیگری که می‌افتد و البته این اتفاق به تعداد بسیار بیشتری نسبت به موتاسیون رخ می‌دهد چسبیدن ابتدای یک کروموزوم به انتهای یک کروموزوم دیگر است. این مساله با نام Crossover شناخته می‌شود. این همان چیزیست که مثلا باعث می‌شود تا فرزند تعدادی از خصوصیات پدر و تعدادی از خصوصیات مادر را با هم به ارث ببرد و از شبیه شدن تام فرزند به تنها یکی از والدین جلوگیری می‌کند.

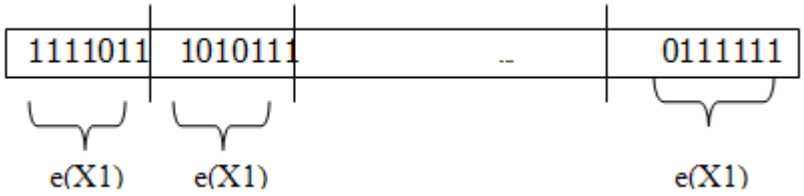
در ابتدا تعداد مشخصی از ورودی ها، X1,X2,…,Xnکه متعلق به فضای نمونه X هستند را انتخاب می کنیم و آنها را در یک عدد برداری X=(x1,x2,…xn) نمایش می دهیم..در مهندسی نرم افزار اصطلاحاً به آنها ارگانیسم یا کروموزوم گفته می شود.به گروه کروموزوم ها Colony یا جمعیت می گوییم.در هر دوره Colonyرشد می کند و بر اساس قوانین مشخصی که حاکی از تکامل زیستی است تکامل می یابند.

برای هر کروموزوم Xi ،ما یک ارزش تناسب (Fitness)داریم که آن را f(Xi) هم می نامیم.عناصر قویتر یا کروموزوم هایی که ارزش تناسب آنها به بهینه Colonyنزدیکتر است شانس بیشتری برای زنده ماندن در طول دوره های دیگر و دوباره تولید شدن را دارند و ضعیفترها محکوم به نابودی اند. به عبارت دیگر الگوریتم ورودی هایی که به جواب بهینه نزدیکترندرانگه داشته واز بقیه صرف نظر می کند.

یک گام مهم دیگر درالگوریتم،تولد است که در هر دوره یکبار اتفاق می افتد. محتویات دو کروموزومی که در فرآیند تولید شرکت می کنند با هم ترکیب میشوند تا ۲ کروموزوم جدید که ما انها را فرزند می نامیم ایجاد کنند.این هیوریستیک به ما اجازه می دهد تا ۲ تا از بهترین ها را برای ایجاد یکی بهتر از آنها با هم ترکیب کنیم. (evolution)به علاوه در طول هر دوره،یک سری از کروموزوم ها ممکن است جهش یابند (Mutation).

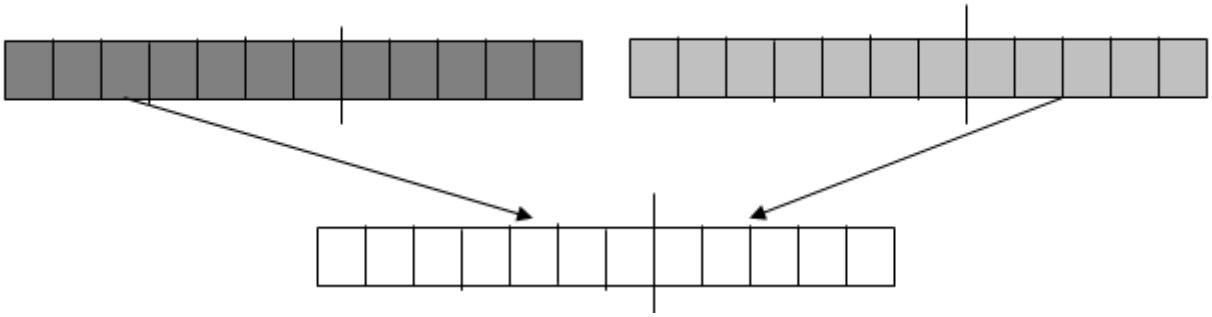
۳.الگوریتم

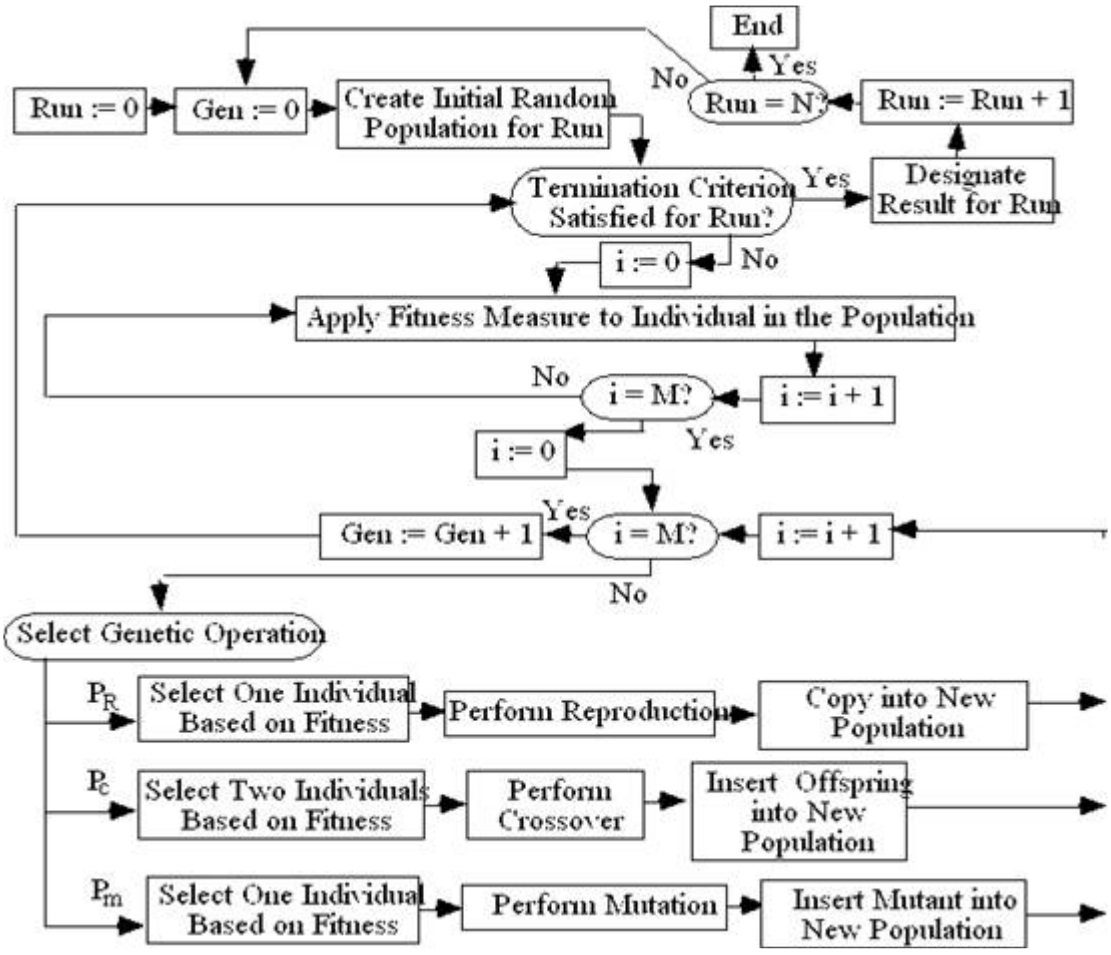
هر ورودی x در یک عدد برداری X=(x1,x2,..,xn) قرار دارد .برای اجرای الگوریتم ژنتیک مان باید هر ورودی را به یک کروموزوم تبدیل کنیم.می توانیم این را با داشتن log(n) بیت برای هر عنصرو تبدیل ارزش Xi انجام دهیم مثل شکل زیر .



(X1, X2,…,Xn)= (123, 87,…, 63)

می توانیم از هر روش کد کردن برای اعداد استفاده کنیم.در دوره ۰، یک دسته از ورودی های X را به صورت تصادفی انتخالب می کنیم.بعد برای هر دورهi ام ما ارزش مقدار Fitness را تولید،تغییر وانتخاب را اعمال می کنیم.الگوریتم وقتی پایان می یابد که به معیارمان برسیم.





۴.کد کردن مقادیر

بر اساس تعریف Hollandروش های متعددی برای نمایش ژن های منفرد وجود دارد. مثلا میتوان آنها را به صورت رشته (String) ، آرایه ، درخت یا لیست نشان داد که قصد داریم آنهارا به صورت رشته های بیتی مورد بررسی قرار دهیم.

۴-۱.کد مبنای دو(Binary)

مثال : کوله پشتی

در این مثاله فرض میکنیم که اشیایی با مقدار و اندازه مشخص وجود دارد و بخواهیم آنها را در یک کوله پشتی با ظرفیت مشخص قرار دهیم. نحوه انتخاب اشیا با توجه به حداقل فضایی که اشغال می کنند و استفاده بهینه از فضای کوله پشتی صورت میگیرد.

برای حل مساله فرض میکنیم هر بیت بیانگر حضور یا عدم حضور اشیا در کوله پشتی باشد. روش کد مبنای دو از روش های متداول در حل مسائل GA به شمار می آیند.

Chromosome	A	101101100011
Choromosome	B	010011001100

اصولا روش کد مبنای دو امکان تولید کروموزوم های بسیاری را با حداقل بیت ها فراهم میکند. لـذا ایـن روش کدگذاری در مسائل واقعی باید همراه با اصلاحاتی بعد از اعمال عملگرهای ژنتیکی صورت گیرد.

۴-۲.روش کدگذاری جایگشتی(Permutation Encoding)

این روش در حل مسائلی چون فروشنده دوره گرد (tsp) و یا مسائلی که به صورت ترتیبی هستند کاربرد دارد. همان طور که در جدول زیر مشاهده می کنید، در این روش کروموزوم هـا بـه صـورت رشـته ای از اعداد نمایش داده میشوند که هریک از این اعداد بر اساس یک ترتیبی قرار گرفته اند.

Chromosome A 1 5 3 2 4 7 9 8 6
Choromosome B 8 5 6 7 2 3 1 4 9

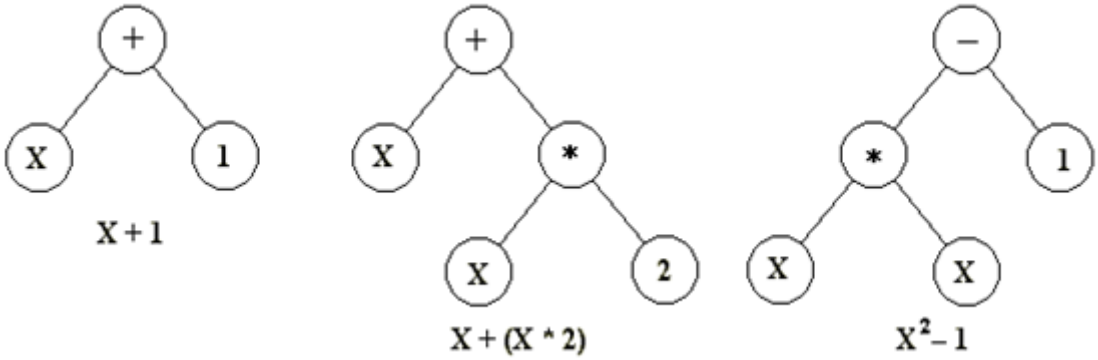
۴-۳.روش کد گذاری مقدار(Value Encoding)

در این روش هر کروموزوم به صورت رشته ای از مقادیر است که این مقادیر می توانن هرچیز مرتبط بـا مساله باشند . مثلا اعداد اعشاری و یا اشیا کد شده که در جدول زیر مثالی از این روش نشان داده شـده است.

Chromosome A	1.254 2.364 9.245 3.0058
Choromosome B	dfgtrgfdbynynh j
Choromosome C	right left back

۴-۴.روش کدگذاری درختی(Tree Encoding)

یک روش دیگر که توسط John Koza توسعه یافت،برنامه نویسی ژنتیک (Genetic programming)است.که برنامه ها را به عنوان شاخه های داده در ساختار درخت نشان می دهد.در این روش تغییرات تصادفی می توانند با عوض کردن عملگرها یا تغییر دادن ارزش یک گره داده شده در درخت،یا عوض کردن یک زیر درخت با دیگری به وجود آیند.



۵.تولید مثل

تولید مثل معمولا اولین عملی است که بر روی جمعیت اعمال میشود. در این روش یک سری کروموزوم از میان جمعیت به عنوان والد انتخاب شده که در نهایت با عمل ادغام(Cross Over) منجر به تولید فرزندان میشوند.

بر اساس نظریه حیات بهترین ها باید بهترین ها انتخاب شوند تا نسل بعدی بهتری را تولید کنند. به همین دلیل گاه به عملگر تولید مثل (Reproduction) عملگر انتخاب (Selection operator) نیز گفته میشود. روش های انتخاب بسیاری در GA برای انتخاب کروموزوم ها وجود دارد اما هدف اصلی در همه آنها انتخاب رشته هایی با میانگین بالا زا جمعیت قعلی و تولید کپی های چندگانه از آنها و قرار دادن آنها در یک مکان به نام استخر تولید مثل (Mating pool) بر اساس یک فرم احتمالی است.

۶.روش های انتخاب :

روش های مختلفی برای الگوریتم های ژنتیک وجود دارند که می توان برای انتخاب ژنوم ها از آنها استفاده کرد.اما روش های لیست شده در پایین از معمولترین روش ها هستند.

۶-۱.انتخاب *Elitist*

مناسبترین عضو هر اجتماع انتخاب می شود انتخاب Roulette : یک روش انتخاب است که در آن عنصری که عدد برازش(تناسب)بیشتری داشته باشد،انتخاب می شود.

۶-۲. انتخاب *Scaling*

به موازات افزایش متوسط عدد برازش جامعه،سنگینی انتخاب هم بیشتر می شودوجزئی تر.این روش وقتی کاربرد دارد که مجموعه دارای عناصری باشد که عدد برازش بزرگی دارند وفقط تفاوت های کوچکی آنها را از هم تفکیک می کند.

۶-۳. انتخاب *Tournament*

یک زیر مجموعه از صفات یک جامعه انتخاب می شوندواعضای آن مجموعه با هم رقابت می کنندو سرانجام فقط یک صفت از هر زیر گروه برای تولید انتخاب می شوند.

بعضی از روشهای دیگر عبارتند از Rank Selection, Generational Selection, Steady-State Selection: Hierarchical Selection.

۷.عملگرهای GA

در این قسمت به بررسی روش های مختلف برای تولید رشته های جدید و همچنین عملگرهای مهم دیگرGA مثل ادغام و جهش میپردازیم. عملگرهای وراثتی متعددی نیز برای تولید رشته های بهتر وجود دارد که هدف این عملگرها جستجوی فضای پارامترها و تا حد امکان حفظ اطلاعات نهفته در رشته است. چرا که این والدها بهترین موارد انتخاب شده توسط عملگرهای فاز تولید مثل هستند و نباید از دست بروند. از مهمترین عملگر های GA میتوان به موارد زیر اشاره کرد :

عمل معکوس کردن (Inversion)

عمل حذف کردن (Deletion)

عمل جداسازی (Segregation)

عمل نقل مکان (Migration)

عمل بخش بندی (Sharing)

عمل جفت گیری (Matin) یا ادغام (Cross Over)

عمل غالب شدن یا تسلط(Dominance)

عمل کپی کردن(Duplication)

که معمولا در یک الگوریتم ژنتیک ساده تنها از سه عملگر اصلی زیر استفاده میشود :

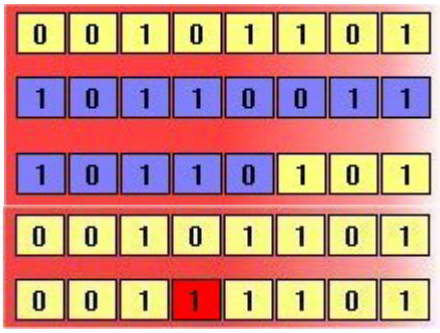
تولید مثل (Reproduction)

ادغام (Cross Over)

جهش(Mutation)

وقتی با روش های انتخاب کروموزوم ها انتخاب شدند،باید به طور تصادفی برای افزایش تناسبشان اصلاح شوند.۲ راه حل اساسی برای این کار وجود دارد.اولین وساده ترین جهش (Mutation) نامیده می شود.درست مثل جهش در موجودات زنده که عبارت است از تغییر یک ژن به دیگری، در الگوریتم ژنتیک جهش تغییر کوچکی در یک نقطه از کد خصوصیات ایجاد می کند.

دومین روش Crossover نام دارد و ۲ کروموزوم برای معاوضه سگمنتهای کدشان انتخاب می شوند.این فرآیند بر اساس فرآیند ترکیب کروموزوم ها در طول تولید مثل در موجودات زنده شبیه سازی شده. اغلب روش های معمول Crossover شامل Single-point Crossoverهستند ، که نقطه تعویض در جایی تصادفی بین ژنوم ها است.بخش اول قبل از نقطه ،و بخش دوم سگمنت بعد از آن ادامه پیدا می کند،که هر قسمت برگرفته از یک والد است،که ۵۰/۵۰ انتخاب شده.



شکل های بالا تاثیر هر یک از عملگر های ژنتیک را روی کروموزوم های ۸ بیتی نشان می دهد. شکل بالاتر ۲ ژنوم را نشان می دهد که نقطه تعویض بین ۵امین و ۶امین مکان در ژنوم قرار گرفته،ایجاد یک ژنوم جدید از پیوند این ۲ والد بدست می آیند.شکل ۲وم ژنومی را نشان می دهد که دچار جهش شده و ۰ در آن مکان به ۱ تبدیل شده .

۷.نقاط قوت الگوریتم های ژنتیک

اولین و مهمترین نقطه قوت این الگوریتم ها این است که الگوریتم های ژنتیک ذاتاً موازی اند .اکثر الگوریتم های دیگر موازی نیستند و فقط می توانند فضای مسئله مورد نظر را در یک جهت در یک لحظه جستجو کنند واگر راه حل پیدا شده یک جواب بهینه محلی باشدویا زیر مجموعه ای از جواب اصلی باشد باید تمام کارهایی که تا به حال انجام شده را کنار گذاشت ودوباره از اول شروع کرد.از آنجایی که GA چندین نقطه شروع دارد،در یک لحظه می تواند فضای مسئله را از چندجهت مختلف جستجو کند. اگر یکی به نتیجه نرسید سایر راه ها ادامه می یابند و منابع بیشتری را در اختیار شان قرار می گیرد.در نظر بگیرید: همه ۸ عدد رشته باینری یک فضای جستجو را تشکیل می دهند،که می تواند به صورت ××××××× نشان داده شود.رشته ۰۱۱۰۱۰۱۰ یکی از اعضای این فضاست.همچنین عضوی از فضاهای ×××××××و۰۱×××××و۰ ×××××و۰×۱×۱×۱×۰و ۰××۰۱×۰۱ والی آخر باشد.

به دلیل موازی بودن واین که چندین رشته در یک لحظه مورد ارزیابی قرار می گیرند GA ها برای مسائلی که فضای راه حل بزرگی دارند بسیار مفید است .اکثر مسائلی که این گونه اند به عنوان "غیر خطی" شناخته شده اند.در یک مسئله خطی،Fitness هر عنصر مستقل است،پس هر تغییری در یک قسمت بر تغییر وپیشرفت کل سیستم تاثیر مستقیم دارد.می دانیم که تعداد کمی از مسائل دنیای واقعی به صورت خطی اند.در مسائل غیر خطی تغییر در یک قسمت ممکن است تاثیری ناهماهنگ بر کل سیستم ویا تغییر در چند عنصر تاثیر فراوانی بر سیستم بگذارد. خوشبختانه موازی بودن GA باعث حل این مسئله می شود ودر مدت کمی مشکل حل می شود.مثلاً برای حل یک مسئله خطی ۱۰۰۰ رقمی ۲۰۰۰ امکان حل وجود دارد ولی برای یک غیر خطی ۱۰۰۰ رقمی ۲۱۰۰۰ امکان .

یکی از نقاط قوت الگوریتم های ژنتیک که در ابتدا یک کمبود به نظر می رسد این است که :GA ها هیچ چیزی در مورد مسائلی که حل می کنند نمی دانندو اصطلاحاً به آنهاBlind Watchmakers می گوییم . آنها تغییرات تصادفی را در راه حل های کاندیدشان می دهند وسپس از تابع برازش برای سنجش این که آیا آن تغییرات پیشرفتی ایجاد کرده اند یا نه، استفاده می کنند.مزیت این تکنیک این است که به GA اجازه می دهند یا با ذهنی باز شروع به حل کنند.از آنجایی که تصمیمات آن اساساً تصادفی است،بر اساس تئوری همه راه حلهای ممکن به روی مسئله باز است،ولی مسائلی که محدود به اطلاعات هستند باید از راه قیاس تصمیم بگیرند ودر این صورت بسیاری از راه حلهای نو وجدید را از دست می دهند.

یکی دیگر از مزایای الگوریتم ژنتیک این است که آنها می توانند چندین پارامتر را همزمان تغییردهند.بسیاری ازمسائل واقعی نمی توانند محدود به یک ویژگی شوند تا آن ویژگی ماکسیمم شود یا مینیمم و باید چند جانبه در نظر گرفته شوند.GAها در حل این گونه مسائل بسیار مفیدند،و در حقیقت

قابلیت موازی کار کردن آنها این خاصیت را به آنها می بخشد.و ممکن است برای یک مسئله ۲ یا چند راه حل پیدا شود،که هر کدام با در نظر گرفتن یک پارامتر خاص به جواب رسیده اند.

۸.محدودیتهای GA

یک مشکل چگونگی نوشتن عملگر Fitness است که منجر به بهترین راه حل برای مسئله شود.اگر این کارکرد برازش به خوبی و قوی انتخاب نشود ممکن است باعث شود که راه حلی برای مسئله پیدا نکنیم یا مسئله ای دیگر را به اشتباه حل کنیم. به علاوه برای انتخاب تابع مناسب برای Fitness ،پارامترهای دیگری مثل اندازه جمعیت،نرخ جهش وCrossover ،قدرت ونوع انتخاب هم باید مورد توجه قرار گیرند.

مشکل دیگر،که آن را نارس می نامیم این است که اگر یک ژنوم که فاصله اش با سایر ژنوم های نسل اش زیاد باشد(خیلی بهتر از بقیه باشد)و خیلی زود دیده شود(ایجاد شود)ممکن است محدودیت ایجاد کند و راه حل را به سوی جواب بهینه محلی سوق دهد.این اتفاق معمولاً در جمعیت های کم اتفاق می افتد.روش های Scaling, Rank tournament selection بر این مشکل غلبه می کنند.

۹.چند نمونه از کاربرد های الگوریتم های ژنتیک

نرم‌افزار شناسایی چهره با استفاده از تصویر ثبت شده به همت مبتکران ایرانی طراحی و ساخته شد. در این روش، شناسایی چهره براساس فاصله اجزای چهره و ویژگی‌های محلی و هندسی صورت می‌گیرد که تغییرات ناشی از گیم، تغییرات نور و افزایش سن کمتین تأثیر را خواهد داشت.

همچنین گراف‌ها برای چهره‌های جدید با استفاده از الگویتم‌های ژنتیک ساخته شده و با استفاده از یک تابع تشابه، قابل مقایسه با یکدیگر هستند که این امر تأثیر به‌سزایی در افزایش سرعت شناسایی خواهد داشت.

توپولوژی های شبکه های کامپیوتی توزیع شده.

بهینه سازی ساختار ملکولی شِمیایی (شیمی)

مهندسی برق برای ساخت آنتنهای Crooked-Wire Genetic Antenna

مهندسی نرم افزار

مهندسی مواد

مهندسی سیستم

رباتیک(Robotics)

تشخیص الگوو استخراج داده(Data mining)

حل مسئله فروشنده دوره گرد
آموزش شبکه های عصبی مصنوعی
یاددهی رفتار به رباتها با GA .
یادگیری قوانین فازی با استفاده از الگویتم های ژنتیک.

یک مثال ساده

ما یک مربع ۳×۳ داریم که می خواهیم اعدادی بین ۱تا۱۵ را در این مربع قرار دهیم به طوری که جمع اعداد در هر سطرو ستون برابر ۲۴ شود.

N	N	N	=24
N	N	N	=24
N	N	N	=24
=	=	=	
24	24	24	

ابن مسئله تا حدودی پیچیده است.ممکن است یک انسان بتواند آن را در مدت زمانی مشخص حل کند ولی هیچ گاه یک کامپیوتر نخواهد توانست آن رادر مدت زمان کوتاهی با استفاده از اعداد تصادفی حل کند. ولی الگوریتم ژنتیک می تواند این مشکل را حل کند.

نسل اول

اولین گام ایجاد کردن یک نسل ابتدایی برای شروع کار است که شامل تعدادی ژنوم تصادفی است.این ژنوم ها به صورت باینری(۰و۱) نشان داده می شوند. حالا مثال مان:
اول یکسری عدد به صورت تصادفی تولید می شوند. هر ژنوم یا کروموزوم شامل اطلاعاتی برای هر ۹ جای خالی است .چون این اعداد مقادیر بین ۰تا۱۵ دارند می توان آنها را با ۴ بیت یا ژن داده نمایش داد. پس هر ژنوم شامل ۳۶ بیت است.
یک نمونه ژنوم می تواند به شکل زیر باشد:

Bits (Genes)	0110	1100	1111	1011	0100	1010	0111	0101	1110
Values(Traits)	6	12	15	11	4	10	7	5	14

حالاباید به هر ژنوم در مجموعه یک عدد تناسب(Fitness) بنابر تاثیر آن در حل مسئله نسبت داد.فرآیند وروش محاسبه این عدد برای هر مسئله فرق می کند.انتخاب الگوی مناسب برای مسئله مشکلترین و حساسترین بخش در حل مسئله ژنتیک است.دراین مثال ما اعداد را در مکان هایشان جایگذاری می کنیم و بررسی می کنیم که چقدر با جواب اصلی فاصله دارند

۶	۱۲	۱۵	=۳۳
۱۱	۴	۱۰	
۷	۵	۱۴	=۲۵
= 24	= 21	= 39	

مقادیر معادل عبارتند از ۳۳و۲۵و۲۶و۲۴و۲۱و۳۹.واضح است که این مقادیر مسئله را حل نمی کنند.پس باید مقادیر تناسب را برای این ژنوم محاسبه کرد.برای این کار ابتدا فاصله هرمجموع را از۲۴ محاسبه کرده،سپس معکوس مجموع تفاصل آنها را محاسبه می کنیم .

$$\frac{1}{|33-24|+|25-24|+|26-24|+|24-24|+|21-24|+|39-24|}$$

$$\frac{1}{9+1+2+0+3+15}$$

$$\frac{1}{30} \approx 0.033$$

بنابراین درجه تناسب برای این ژنوم تقریباً برابر ۰.۰۳۳ است.هرچقدر که اعداد ما به جواب نزدیکتر باشند عدد تناسب بزرگتر خواهد شد.اما اگر مخرج ما برابر ۰شود چه اتفاقی می افتد؟ دراین صورت همه اعداد ما برابر ۲۴ شده اند وما به جواب رسیده ایم.

نسل بعدی

دو ژنوم به طور تصادفی برای تولید نسل بعدی انتخاب می شوند. این اصلی ترین بخش الگوریتم ژنتیک است که از ۳ مرحله تشکیل شده:

انتخاب

دو ژنوم به طور تصادفی از نسل قبل انتخاب می شوند.این ژنوم ها دارای اعداد تناسب بزرگتری هستند و بعضی صفات آنها به نسل بعدی منتقل می شوند. این بدین معنی است که عدد تناسب در حال افزایش خواهد بود.

بهترین روش برای تابع انتخاب(Fitness) در این مسئله روشی به نام رولت(Roulette) است.اول یک عدد تصادفی بین ۰ وعدد تناسب نسل قبلی انتخاب می شود. تابع انتخاب به صورت زیر خواهد بود:

```
RouletteSelection()
{
   float ball  = rand_float_between(0.0, total_fitness);
   float slice = 0.0;
```

```
    for each genome in population
    {
      slice += genome. fitness;

      if ball < slice
        return genome;
    }
}
```

تغییر از یک نسل به نسل بعدی (Cross over)

حالا دو ژنوم بخشی از ژنهایشان را برای ایجاد نسل بعدی اهدا می کنند. اگر آنها تغییر پیدا نکنند همانطور بی تغییر به نسل بعدی منتقل خواهند شد.درجه Crossover نشان دهنده این است که هر چند وقت یکبار ژنوم ها تغییر پیدا خواهند کرد و این عدد باید در حدود ۶۵-۸۵٪ باشد.

عملگر تغییر در ژنوم های باینری مثال ما با انتخاب یک مکان تصادفی در ژنوم برای تغییر آغاز می شود. بخش اول ژنهای پدر و بخش دوم ژنهای مادر با هم ترکیب می شوند(و بالعکس) تا۲ فرزند تولید شوند. در زیریک عمل تغییر را می بینیم.

Before Crossing
Father 011110010011 001011011000111011010000
Mother 010100111110 010101111101000100010010
After Crossing
Child1 011110010011 010101111101000100010010
Child2 010100111110 001011011000111011010000

جهش (Mutation)

قبل از این که ژنوم ها در نسل بعدی قرار بگیرند،احتمال دارد دچار جهش یا تغییر ناگهانی شوند شوند.جهش یک تغییر ناگهانی در ژن است.در ژنهای باینری این تغییر به معنای تغییر یک بیت از ۰به ۱ یا از ۱ به ۰ است. درجه جهش نشان دهنده احتمال بروز جهش در یک ژن است و تغریباً بین ۱-۵٪ برای ژنهای باینری و ۵-۲۰٪برای ژنهای عددی است.

۱۰.هایپر هیوریستیک

مثال های بسیاری وجود دارد که برای حل ان ها از الگوریتم ژنتیک استفاده شده است از جمله Traveling salesman– bin packing – scheduling .مسئله ی جدول زمانی پرسنل وکارکنان با استفاده از الگوریتم ژنتیک به صورت موفقیت آمیزی حل شده است .aickelin- dowslamd از الگوریتم ژنتیک لیست کار پرستاران در یکی از بیمارستان های بزرگ انگلستان به کا ر برده شد mansour-esto از الگوریتم ژنتیک برای حل مسئله ی جدول زمان بندی کار استفاده کردند .در واقع برای حل این مسئله از الگوریم ژنتیک توزیع شده که به صورت موازی روی شبکه ی محل کار وجود داشت دست یافتند.آنچه از این تحقیق بدست آمد سه دسته ی متفاوت زمان بندی برای مجموعه ای از کارهای متفاوت بود. این محققین مسائلی را که با هیوریستیک و متا هیوریستیک کار میکرند با هم مقایسه کردند و نتیجه ان بود که الگوریتم ژنتیک بهتر از همه ی آنها کار می کرد در نتیجه الگوریتم های ژنتیک با کروموزوم های مستقیم و کروموزوم های غیر مستقیم به صورت گسترده ای مورد مطالعه قرار گرفت . به عنوان مثال hart & ross& nelson از الگوریتم ژنتیک با کروموزوم های مستقیم برای حل مسئله ی زمان بندی امتحانات طراحی شد. آنها استراتژی های به عنوان پارامتر در ده خانه ای ارایه ایجاد کردند. بنابراین کروموزوم ها ساختار جدول زمان بندی را ایجاد می کنند نه خود جدول زمان بندی. کروموزوم های غیر مستقیم برای رفع محدودیت های کروموزوم های مستقیم مورد استفاده قرار می گیرند. که این محدودیت همان شکست هماهنگ بین قسمت های مختلف حل مسئله وجود داشت.

الگوریم ژنتیک بیشتر در مسائلی مورد استفاده قرار می گرفت که روی مسئله ی زمان و زمان بندی تاکید داشت و در این نوع مسائل ما نیاز به دامنه ی دانش مسئله داریم . در واقع کروموزوم ها که به عنوان ساختار حل مسئله بودند دانش مسئله در طراحی کروموزوم ها در الگوریتم ژنتیک بسیار لازم و ضروری است . وابستگی زیاد این طراحی به دامنه ی دانش مسئله موجب می شود تا نتوانیم الگوریتم ژنتیک بدست آمده در مسائل دیگر به کار ببریم.

الگوریتم های ژنتیک که از کروموزوم های غیر مستقیم بر پایه ی سلسله مراتب تکاملی هیوریستیک طراحی شده اند را در مسائلی همچون مسئله ی زمان بندی کار پرسنل می توان به صورت عمومی ودر همه ی مکان ها استفاده کرد.

متد هایپر هیوریستیک که از آن در مسئله ی حمل و نقل مرغ های زنده تو سعه و تکامل دادند این مسئله به این شیوه حل شد که این مسئله را به دو زیر مسئله تقسم کردند هر یک از این دو زیر مسئله از الگوریتم ژنتیک جدا استفاده میکردند. هر کدام از این دو الگوریتم ژنتیک یک استراتژی برای تولید برنامه معین می کنند نه اینکه خودشان یک برنامه باشند تمامی اطلاعات شرکت به عنوان یک سری قانون است که به وسیله ی توانایی جستجوی الگوریتم ژنتیک پشتیبانی می شود. سلسله مراتب در الگوریتم هیوریستیک معین میکند که کدام هیوریستیک برای قرار دادن کارها در برنامه استفاده شود.

الگوریتم هایپرهیوریستیک غیر مستقیم الگوریتمی است که از یکسری هیوریستیک سطح پایین و یکسری هیوریستیک سطح بالا که هایپر هیوریستیک که فقط میداند که کی توابع هدف ماکسیمم یا مینیمم هستندو هیچ اطلاعاتی در مورد اینکه تابع هدف چه چیزی را نشان میدهد ندارد و هیچ گونه دامنه ی دانشی در هایپر هیوریستیک و هیوریستیک های سطح پایین مرتبط با ان موجود نمی باشد . همچنین با توجه به چهارچوب کلی یکسری توابع انتخاب داریم که که تصمیم میگیرند کدام هیوریستیک را صدا بزنند.

منابع

کتاب الگوریتم های ژنتیک و کاربردهای آن ترجمه : مهندس مهدی علیرضا
مقاله ای از مرکز تحصیلات تکمیلی در علوم پایه زنجان
مجله علم و کامپیوتر www.ccwmagazine.com
پاورپوینتKoza

گردآورنده : امیر علی بلورچیان amirali@eca.ir