الگوریتم ژنتیک

Genetic algorithm

امید یعقوبی
deopenmail@gmail.com
دانشکده‌ی فنی و مهندسی
دانشگاه علم و فرهنگ
دی ۱۳۹۳

چکیده

الگوریتم ژنتیک یک روش حل مسئله الهام گرفته از طبیعت است که از تئوری تکامل برای حل مسائلی که فضای جواب بزرگی دارند استفاده می‌کند. این مقاله به نحوه پیدایش و مقایسه آن با دیگر الگوریتمهای مشابه می پردازد. پس از توضیح تکامل و روش حل مسئله توسط الگوریتم ژنتیک وارد جزئیات آن مانند انواع روشهای انتخاب و تولید مثل می‌شود.

کلید واژه‌ها: الگوریتم ژنتیک، الگورتیم تکاملی، الگوریتم الهام گرفته از طبیعت، متاهیوریستیک، فرامکاشفه‌ای

۱. محیط مسئله

در اواخر دهه‌ی ۵۰ میلادی دو موضوع اصلی در هوش مصنوعی الگوریتمهای مکاشفه‌ای و شبیه‌سازی شبکه‌های عصبی بود. دکتر لارنس فوگل این دو دیدگاه را محدود به مدل انسان می‌دید و عقیده داشت که هوشمندی واقعی را می‌توان در فرایندی که موجب پدید آمدن موجودات روی کره‌ی زمین شده است مشاهده نمود. او نوعی از هوشمندی را مورد بررسی قرار داد که بر اساس آن رفتار یک عامل در تعامل با محیط شروع به سازگار شدن می‌کند. او برای اولین بار از پدیده‌ی تکامل برای حل مسئله استفاده نمود. در تئوری پیشنهادی او برنامه‌ای کار را با چندین جواب بالقوه شروع می کرد و در طول زمان این جوابها را تغییر می‌داد. در واقع جوابهایی که در زمان حال برنامه وجود داشتند به نوعی فرزندانِ جوابهای پیشین مسئله بودند. [۱]

با توجه به کارهای فوگل که مبنای استفاده از تئوری تکامل در علوم محاسباتی بود می‌توان دریافت که مسائلی با این روش قابل حل می‌باشند که بیش از یک جواب دارند و پیدا کردن جواب در آنها به تنهایی اهمیت ویژه‌ای ندارد، بلکه در فضای بزرگی از جوابها پیدا کردن بهترین جواب یا جواب بهینه است که اهمیت می‌یابد. پس واضح است که از آن نمی‌توان در مسائلی همچون مسائل تصمیم گیری استفاده نمود. زیرا در این مسائل ما نمی‌توانیم یک جواب بالقوه پیدا کنیم، که اگر می‌توانستیم مسئله به خودیِ خود حل شده بود (جواب در مسائل تصمیم گیری بلی یا خیر است).

تکنیکی که فوگل از آن استفاده کرده است جست‌وجوی محلی نام دارد. او بجای آنکه تعداد دفعات جست‌وجو را هر بار افزایش دهد برای هر جواب یک حول همسایگی در نظر می‌گیرد و آن را مورد بررسی قرار می‌دهد. فرض کنید که در کتابی دنبال مطلبی می‌گردید و کتاب فاقد فهرست موضوعی است. هوشمندانه نیست اگر بخواهید کل کتاب را از ابتدا تا انتها مورد بررسی قرار دهید؛ پس جایی از کتاب را به تصادف باز می کنید و قسمتی از آن را می‌خوانید. حال سوالی که پیش می‌آید این است که چه مقدار از صفحه ی باز شده را می خوانید؟ یک کلمه؟ یک جمله؟ یک پاراگراف؟ در واقع شما با بزرگتر کردن این بازه در جست و جویتان حول همسایگی آن را افزایش می دهید. در واقع اگر شما یک جواب داشته باشید و آنرا تغییر دهید یک همسایگی از آن به دست آورده اید، حال اگر تغییرات زیاد باشند حول همسایگی بزرگ و اگر کم باشند حول آن کوچک است. برای مثال شاید منطقی به نظر نیاید که با باز کردن کتاب به طور تصادفی کل فصل پیش آمده را بخوانید. و منطقی نیست اگر تنها یک جمله از آن را مورد مطالعه قرار دهید. البته لازم به ذکر است که شما علاوه بر خواندن آن قسمت از کتاب شباهت آن را با موضوع مورد نظرتان در ذهن ارزیابی می‌کنید و با توجه به شباهت آن اندازه‌ی همسایگی را کوچک یا بزرگ می‌گیرید. اگر نزدیک به آن چیزی بود که در ذهن دارید بازه‌ی بزرگتری را برای همسایگی آن قائل می شوید و اگر اگر بی‌ربط بود بازه‌ای بسیار کوچک را مد نظر قرار می‌دهید. البته این تعیین اندازه برای همسایگی در کار فوگل لحاظ نشده بود، بلکه در تحقیقات شخصی از دانشگاه میشیگان به نام جان هلند بود که برای اولین بار مطرح شد و پایه‌ی الگوریتم ژنتیک را بنا نهاد. هلند این کار را با ترکیب جوابهای خوب با هم دیگر انجام داد که در ادامه به آن خواهیم پرداخت. ویژگی مهم دیگری که با مثال جست‌وجو در کتاب باید به آن توجه کرد، عدم نیاز به نگه داری مسیر رسیدن به جواب است. برای شما مهم نیست که بدانید از چه مسیری باید به صفحه‌ی مورد نظرتان برسید. این ویژگی در الگوریتمهایی که از تکنیک جست‌وجوی محلی استفاده می‌کنند حائز اهمیت است و یکی از مزایای آن است چون نیازی به حافظه‌ی بزرگ در آن دیده نمی شود.

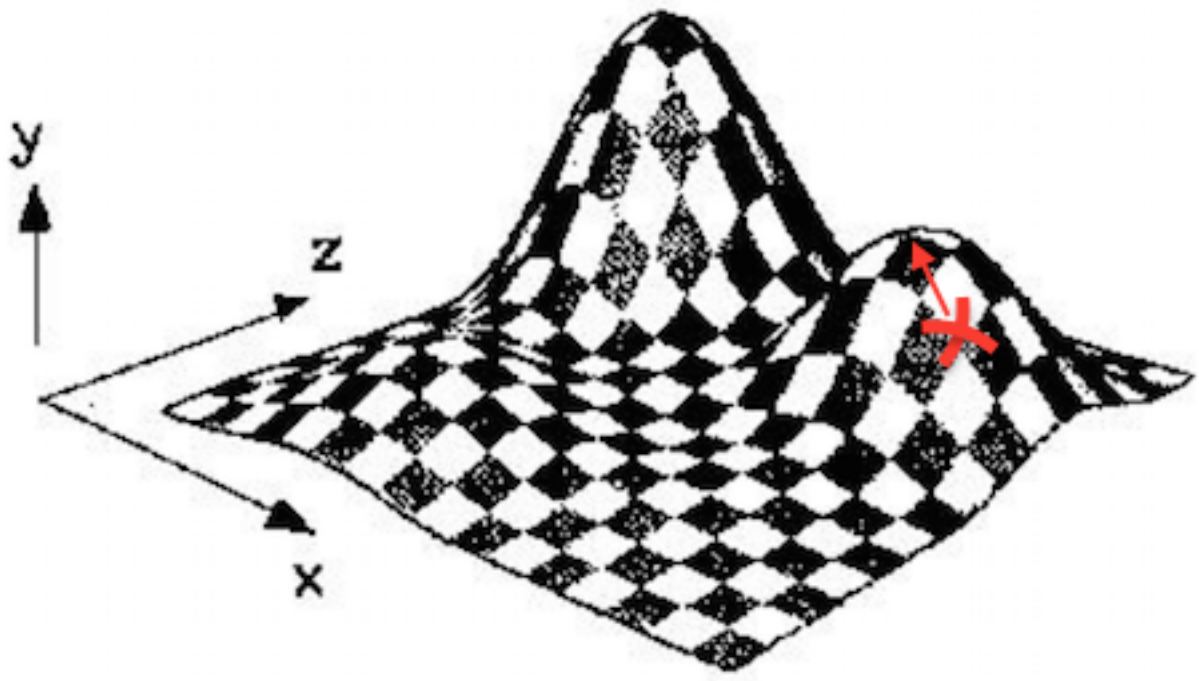
سوالی که پیش می‌آید این است که با چه چیزی می‌توان بهینگی یک جواب را مورد ارزیابی قرار داد؟ برای مثال شما از کجا می‌فهمید که صفحه‌ای را که مطالعه می‌کنید به موضوع مورد نظرتان نزدیک است یا دور؟ این وظیفه را تابع هیوریستیک بر عهده دارد. این تابع هر جواب بالقوه را بر اساس معیارهای داخلی خود می‌سنجد و به آن نمره می‌دهد. مثلا در جست‌وجوی کتاب فاصله‌ی موضوعی که به تصادف باز کردید با موضوعی که مورد جست‌وجویتان هست (هدف) مورد بررسی قرار می‌گیرد. این فاصله به طور ناخوداگاه و با برداشتی که شما از خواندن یک تکه به دست می آورید مورد ارزیابی می گیرد و این برداشت نتیجه‌ی آشنایی شما با ساختار زبان طبیعی است که در طول عمر خود با فرایند یادگیری به آن تسلط پیدا کردید. در مورد سیستمهای مصنوعی می‌توان از تئوریهای مطرح در پردازش زبان طبیعی برای این کار استفاده نمود. در یک راه حل ساده می توان از میزان اشتراک کلامات مرتبط با موضوع شما و کلمات موضوعی که به تصادف در حال مطالعه‌ی آن هستید برای محاسبه‌ی مقدار ارزیابی استفاده نمود و البته لازم به ذکر است که می بایست تنها کلمات کلیدی را مدنظر قرار دهید، چون حروف اضافه، برخی افعال و ...

نباید در این ارزیابی شرکت داشته باشند زیرا اینها در زبانهای طبیعی رابطه‌ی بسیار کمی با موضوع اصلی دارند و البته این قضیه وابسته به زبان مورد نظر نیز هست. برای مثال برخی افعال می توانند اطلاعات بسیار زیادی را در مورد موضوع جمله در اختیار قرار دهند. تابعی که در الگوریتمهای تکاملی این گونه ارزیابی‌ها را انجام می‌دهد به تابع *fitness* یا تابع سازگاری معروف است، زیرا سازگاری یک جواب را با محیط پیرامونش می‌سنجد. در ادامه در ارتباط با آن صحبت خواهم کرد.

حال اطلاعاتی در رابطه با محیطهایی که استفاده از روشهای مبتنی بر تئوری تکامل روی آنها جوابگو خواهد بود را داریم. می‌دانیم که این مسائل دارای فضای بسیار بزرگی از جوابها هستند و پیدا کردن جواب بهینه است که در آنها اهمیت دارد نه هر جوابی (مانند مسائل بهینه سازی). می دانیم که در این گونه مسائل مسیر رسیدن به جواب اهمیتی ندارد و همچنین می دانیم که برای تشخیص جواب بهینه نیاز به تابعی داریم که بتواند جوابها را مورد ارزیابی قرار دهد. به الگوریتمهایی که برای اینگونه مسائل طراحی شده‌اند، الگوریتمهای فرامکاشفه‌ای می‌گویند. حال برخی از این الگوریتمها را مورد بررسی کوتاه قرار می‌دهیم تا مزیت روشهای تکاملی را در مقایسه با آنها ببینیم.

۲. الگوریتمهای بهبود تکراری و فرامکاشفه‌ای

از ساده‌ترین الگوریتمهای خودبهبودی و فرامکاشفه‌ای می‌توان الگوریتم تپه‌نوردی را نام برد. در این الگوریتم ابتدا یک جواب تصادفی در فضای مسئله ایجاد می‌شود. سپس چند همسایگی برای آن مشخص می‌گردد. در بین همسایگی‌ها بهترین همسایه انتخاب و با نقطه‌ی فعلی مقایسه شده و در صورت بهتر بودن آن جایگزین شده و این عمل تا رسیدن به جایی که همسایه از نقطه‌ی فعلی بهتر نباشد ادامه پیدا می‌کند. دلیل نام گذاری این روش با نام تپه‌نوردی این بود که در مسائلی که نقاط بهینه نقاط ماکسیمم در محیط هستند حرکت نقطه در روند پیدا کردن جواب روی نمودار دوبعدی یا سه‌بعدی، مانند تپه‌نوردی است، یعنی نقطه برای رسیدن به ماکسیموم، رویِ منحنیِ برآمده به سمت بالا حرکت می کند و روی قله‌ی آن می‌ایستد.



شکل ۱. نقطه‌ای در حال ماکسیمم کردن خود در الگوریتم تپه نوردی

شکل ۲. نقطه‌ای در حال مقایسه شدن با بهترین همسایه‌اش

مشکل اصلی این الگوریتم افتادن در بهینه‌های محلی است. زمانی که نقطه به اولین جایی رسید که از همسایه‌هایش هیچ‌کدام بهتر نبودند جست‌وجو را متوقف می سازد. همین اتفاق در شکل ۱ خواهد افتاد و این الگوریتم بهترین جواب را برای مسئله پیدا نخواهد کرد. همچنین مشکلات دیگری مانند فلات (نقطه‌ای که همسایگانش همه اندازه‌ی هم‌اند) و تیغه‌ها (بهینه‌های محلی کوچک و پیاپی در راه رسیدن به نقطه‌ی بهینه‌ی اصلی) وجود دارد. برای حل این مشکل در الگوریتم تعمیم-یافته‌ی تپه‌نوردی پیشنهاد شده است که هنگام رسیدن به جواب نوعی جهش صورت گیرد تا الگوریتم محدوده‌ی بیشتری از مسئله را مورد کاوش قرار دهد. یا از آنجاکه نقطه‌ی شروع در این الگوریتم از اهمیت ویژه ای برخوردار است الگوریتم را پس از رسیدن به جواب در هر مرحله دوباره اجرا کنیم و اینکار را تا چندبار با نقاط شروع جدید تکرار کنیم.

فرض کنید الگوریتم تپه‌نوردی را طوری بسط دادیم که به جای یک نقطه در یک لحظه جمعیتی از نقاط در حال جست‌وجو بودند و این جمعیت می توانستند از اطلاعات یکدیگر استفاده کنند (از هم مستقل نبودند). این دیدگاه به نوعی همان الگوریتم ژنتیکی است که به آن خواهیم پرداخت. در آنجا نقاط خوب باهم ترکیب و در نقطه‌ای بینابین قرار می‌گیرند و شیوه‌ی جست‌وجو در آن برخلاف تپه‌نوردی موازی است.

قبل از آنکه به الگوریتم ژنتیک پرداخته شود الگوریتم فرامکاشفه‌ای دیگری را مورد بررسی قرار می دهیم که با اجازه‌ی ورود نقاطی که بهتر از نقطه‌ی فعلی نیستند با احتمالی خاص مشکلات الگوریتم تپه‌نوردی را برطرف خواهد کرد. در الگوریتم simulated annealing همه چیز مانند الگوریتم تپه‌نوردی است به جز آنکه به احتمالی که در ابتدا زیاد و به تدریج کاهش میابد به نقاطی که از نقطه‌ی فعلی بهتر نیستند اجازه می‌دهیم تا با نقطه‌ی فعلی جایگزین شوند. با این کار سرعت حرکت نقطه در ابتدای کار زیاد و با گشت‌وگذاری اولیه در فضای جواب به تدریج در نقطه‌ای نزدیک به بهینه‌ی عمومی کاهش میابد و نقطه‌ی بهینه‌ی عمومی را به احتمال بسیار قوی‌ای پیدا خواهد کرد. مشکل این الگوریتم اول آن است که در هر مرحله تنها یک نقطه در فضا در حال بهبود یافتن است و همچنین احتمال بررسی شدن چندین‌وچند مرتبه‌ی یک نقطه ی بهینه بسیار زیاد است.

۳. الگوریتم ژنتیک

جان هنری هلند در اوایل دهه‌ی ۷۰ میلادی در مقاله‌ای با عنوان «سازگاری در سیستمهای طبیعی و مصنوعی» الگوریتم ژنتیک را پایه‌گذاری نمود. او پیش از آن به اتوماتای سلولی علاقه‌مند شده بود. در مدل اتوماتای سلولی هر سلول می‌تواند دارای بازه‌ای متناهی از حالات ممکن باشد و در شبکه‌ای این سلولها کنار هم قرار می‌گیرند. در زمانهای گسسته هر نسل خروجی تابعی است که روی تک تک سلولها اجرا می‌شود. تابع با توجه به سلولهای همسایه موقعیت سلول فعلی را تعیین می‌کند. برای مثال می‌توانیم قانونی تعریف کنیم که در آن اگر همسایه‌های یک سلول کم یا زیاد باشند سلول بعلت انزوا یا ازدحام جمعیت بمیرد و ... این مدل با آنکه دارای اجزای ساده‌ای است سیستمی غیرقابل پیش بینی را به وجود می‌آورد (سیستم پیچیده). و اگر قدری توابع آن را پیچیده‌تر کنیم حتا قادر است تولید مثل کند. دانشمندان عقیده دارند که اتومات سلولی به نوعی ظهور حیات در دریاهای اولیه را نشان می‌دهد. او از این طریق به پدیده‌ی تکامل و انتخاب طبیعی داروین علاقه‌مند شد.

در انتخاب طبیعی داروین ادعا کرد که منابع محدودند و تولید مثل موجودات نامحدود، پس حتما باید رقابتی برای رسیدن به منابع وجود داشته باشد. همچنین با مشاهده‌ی موجودات کره‌ی زمین دریافت که خصوصیات بسیار گوناگونی در آنها وجود دارد. او ابتدا به تغییر شکلی که در نسل حیوانات و گیاهان اهلی پدید آمده بود علاقه‌مند گشت. برای مثال او اشاره کرد که پستان گاوها و بزها در سرزمینهایی که دوشیدن شیر آنها عملی رایج است بطور ارثی و قابل توجهی بزرگتر از دیگر هم‌نوعانشان هست. یا وزن استخوانهای پای اردک اهلی بیشتر از اردک وحشی اما وزن استخوانهای بال او کمتر است زیرا اردک اهلی بیشتر راه می‌رود اما کمتر می‌دود. داروین ادعا نمود که هرگونه موجود زنده دائما در حال تکامل است. وجود خصوصیات متفاوت فراوان بین نسلهایی از جانداران و باقی ماندن تنها عده‌ای از آنها چارلز داروین را به فکر فرو داشت و او نتیجه گرفت که این خصوصیات برترند که باعث باقی ماندن برخی از انواع موجودات در روند تکامل و از بین رفتن تعداد کثیری از آنها می‌شود و این خصوصیات قابل به ارث رسیدن در نسلهای بعدی هستند.

هوگو دو واریس مشاهده کرد که گاه این تغییر در برخی از جمعیت گیاهان بسیار چشم‌گیر و غیرقابل توضیح است. او با نظریه‌ی جهش گفته‌های داروین را تکمیل نمود. البته لازم به ذکر است که نقش جهش در روند تکامل بسیار ناچیز است، اما خواهیم دید که با پیاده‌سازی آن در سیستمهای هوشمند تکاملی نتایج از به دام افتادن در برخی از نقاط نجات پیدا می‌کنند.

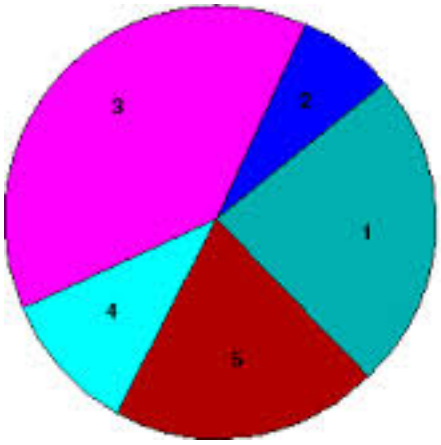
کروموزوم حاوی اطلاعات ژنتیکی ما هنگام تولید مثل است. یک کروموزوم حاوی ژنهایی است که هرکدام خصوصیتی را در خود نگه می دارند. مثلا خصوصیت مربوط به رنگ چشم را نگه‌داری می‌کنند.

در فضای مسئله نیز هرراه حل می‌بایست به کمیتهایی از متغیرها تفسیر شود. در الگوریتم ژنتیک راه حل در قالب یک کروموزوم که حاوی ژنهاست شبیه سازی می‌شود. مزیت این الگوریتم این است که مبتنی بر جمعیت عمل می‌کند نه یک راه‌حل و این باعث جست‌وجوی موازی در فضای مسئله خواهد شد. در ابتدای کار ما جمعیتی داریم از جوابهای بالقوه که هرکدام یک کروموزوم هستند. سپس تابعی داریم برای ارزیابی هریک از جوابها، این تابع در مسائل مختلف متفاوت است و نوعی تابع هیوریستیک محسوب می‌شود. مثلا در یک نمونه می تواند با استخراج ویژگیها و وزنهایی که دارند (از آنجا که ویژگیها دارای اهمیت یکسان نیستند) یک جواب بالقوه را ارزیابی کند. یعنی آنکه بگوید جواب به دست آمده چه مقدار به جواب بهینه نزدیک است. البته ممکن است با مسائلی روبه رو شویم که پیدا کردن جواب بهینه‌ی واقعی در آن غیر ممکن است. با این حال می‌بایست مکانیزمی داشته باشیم تا بتوانیم دو جواب را از یکدیگر با توجه به سازگاریشان تفکیک دهیم. و این مقداردهی نمی‌تواند مثلا دارای دو مقدار خوب و بد باشد (مانند مسائل تصمیم گیری: بلی و خیر) . باید رنجی داشته باشیم و هرچقدر این رنج بزرگتر باشد الگوریتم بهینه تر عمل خواهد کرد. چون می‌تواند دسته‌بندی بهتری بین جوابها برقرار سازد. لازم به ذکر است که می‌بایست مکانیزمی داشته باشیم تا به وسیله‌ی آن بتوانیم یک جواب را به رشته‌ای از کمیتها تفسیر کنیم. مثلا اگر جواب یک عدد در مبنای ۱۰ است می‌توانیم آن را به کد باینری تفسیر کنیم. البته این تفسیر در مسائل مختلف بسته به جواب متفاوت است. مثلا بی‌معنیست اگر بخواهیم جوابی داشته باشیم که با تبدیل آن به کروموزوم ۱۰۰ هزار کمیت برای هر جواب تولید شود. زیرا ما ازاین کروموزومها برای تولید مثل استفاده خواهیم کرد تا نسل بعدی را شکل دهیم. با توجه به مکانیزمهای مختلف تولید مثل طول خیلی زیاد یا خیلی کم کروموزوم می‌تواند تاثیر مستقیم بگذارد بر بهینگی اجرای آن، فرض کنید یک جمله به کمیتهای صفر و یکی تقلیل یافته است و هنگام تولیدمثل به روش تک نقطه‌ای ما از یک نقطه دو کروموزوم را قطع و بهم می چسبانیم. در این حالت احتمال خراب کردن یکی از متغیرهای مسئله بسیار زیادتر از موقعی است که ما جمله را به حروف آن کد کنیم، فرض کنید که هر حرف در باینری رشته‌ای به طول ۲۵۵ ایجاد می کند. انتخاب یک نقطه‌ی تصادفی در کروموزوم به احتمال ۲ به ۲۵۵ اولین اندیس رشته یا آخرین اندیس آن خواهد بود. و به احتمال ۲۵۳ به ۲۵۵ یعنی بالای ۹۹ درصد حرف مربوط تغییر می کند و این تغییر به گونه‌ای است که ممکن است کل کلمه را بی‌معنی سازد. به همین علت کمیتهای باید تا حد امکان معنا دار باشند و نباید همچنین رشته ای مثلا به طول سه برای آن در نظر بگیریم چون در آن صورت الگوریتم خیلی کند عمل خواهد کرد. به خاطر داشته باشید که مبنای این الگوریتم تصادفی است و ممکن است یک کلمه سه حرفی را بارها تکرار سازد.

از بین جمعیت تعدادی برای عمل تولید مثل انتخاب می‌شوند. این انتخاب براساس میزان سازگاری آنها با محیط صورت می‌گیرد.

تابع ارزیابی یا به اصطلاح fitness برای هر جواب یک میزان برازندگی در نظر می‌گیرد بهترینها را انتخاب و با عمل تولیدمثل جمعیت جدید را ایجاد می‌کند. باید توجه داشت که اگر جمعیت هر نسل نسبت به نسل قبلی بیشتر شود الگوریتم در زمانی که به سمت بی‌نهایت پیش می‌رود دچار کمبود حافظه خواهد شد. از طرفی انجام عمل تولید مثل به صورت یک به یک یعنی هر دو والد یک فرزند داشته باشند نیز باعث بهینه عمل نکردن الگوریتم خواهد بود، زیرا تنها یکی از همسایه های جواب در آن چک می‌شود. باید مکانیزمی داشت که جمعیت در هر نسل تعداد ثابتی داشته باشد. روشهای متفاوتی برای این موضوع بررسی شده است. یکی از آنها اجرای دوباره‌ی تابع برازش روی فرزندان و انتخاب تعداد مشخصی از آنها بر اساس برازندگی به نسل بعدی است. موردی که اینجا باید توجه شود این است که در این صورت نیازی به دوباره حساب کردن میزان برازندگی در نسل بعد برای انتخاب والدین نیست زیرا آنها در زمان t-1 مورد ارزیابی قرار گرفته‌اند. باید توجه داشت که اینگونه نیست که حتما اعضای برازنده اجازه ی تولید مثل داشته باشند بلکه احتمال انتخاب شدن آنها بیشتر خواهد بود. دلیل این امر آن است که نباید گوناگونی یا diversity جمعیت به طور ناگهانی کاهش یابد. این امر احتمال رسیدن به جواب بهینه‌ی محلی را بیشتر می‌کند. همچنین استفاده از مکانیزم جهش با احتمال کم هم به همین علت گذاشته شده است. تا جمعیت به طور ناگهانی حول یک دایره گرد نیایند که اگر این اتفاق بیفتد فضای بسیار بزرگی از حالتها بررسی نمی‌شود و احتمال افتادن در نقطه‌ی بهینه ی محلی زیادتر می‌شود.

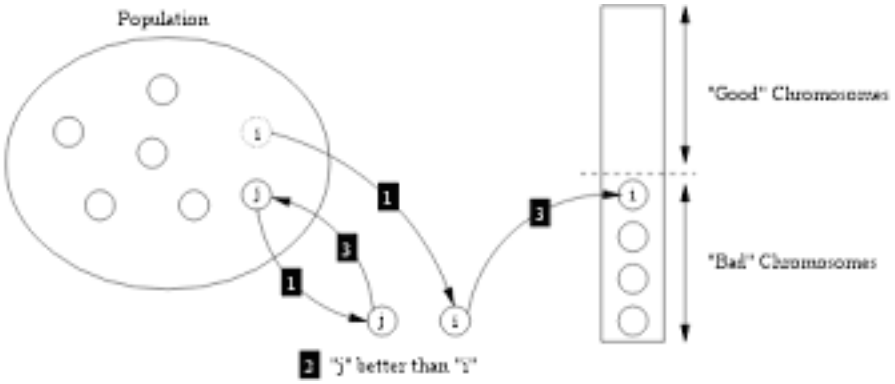
معمول ترین روش انتخاب روش چرخ گردان یا Roulette wheel selection می‌باشد. در این روش چرخی را در نظر گیرید که هر کدام از اعضای جمعیت قسمتی از آن را اشغال کرده‌اند و ما این چرخ را می گردانیم تا نشانه روی یکی از آنها قرار گیرد. احتما قرارگیری نشانه روی تیکه‌های بزرگتر بیشتر است. پس تیکه‌هایی را بزرگ می‌گیریم که تابع برازش آنها برازندگی بیشتری را برایشان ارزیابی کرده است. برای پیاده‌سازی این روش کافیست در یک ساختمان داده مانند آرایه اندیسهای جمعیتمان را قرار دهیم و برای نشان دادن بزرگی یک تکه می بایست تعداد تکرار شدن آن در آرایه افزایش یابد، مثلا اگر ۵ بالاترین میزان برازندگی را دارد می بایست بیشتر از همه در آرایه تکرار شود، مثلا ۱۰ خانه از آرایه به تصادف برای ۵ در نظر گرفته شود. سپس یک عدد تصادفی بین ۱ تا طول آرایه تولید می‌کنیم و خانه‌ی مربوط به آن به عنوان یکی از والدها انتخاب می شود. احتمال انتخاب شدن ۵ به علت بالاتر بودن فراوانی آن از بقیه بیشتر خواهد بود.



Population	Fitness
1	25.0
2	5.0
3	40.0
4	10.0
5	20.0

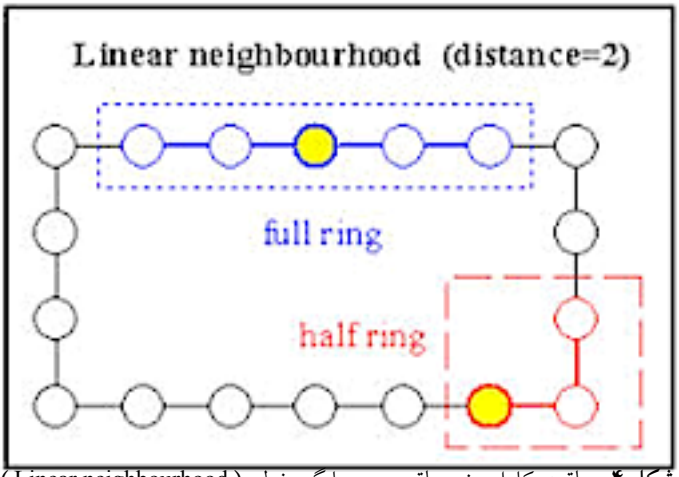
شکل ۳. انتخاب به روش چرخ گردان (roulette wheel selection)

روش انتخاب دیگری به نام انتخاب مسابقه‌ای وجود دارد که در آن k کروموزوم به احتمال برابر انتخاب می‌شوند. سپس از میان آنها تعدادی از بهترینها انتخاب و بقیه حذف می شوند و این عمل تا آنجا که جمعیت والدها به اندازه‌ی مورد نظر برسد تکرار می‌شود.

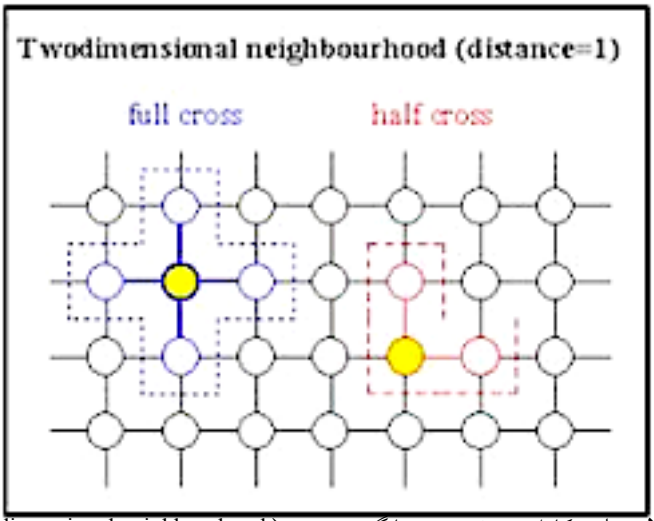


شکل ۳. انتخاب به روش مسابقه‌ای (Tournament selection)

در روشی به نام انتخاب محلی یا local selection ابتدا نیمی از جمعیت با استفاده از یکی از روشهای انتخاب جدا شده سپس برای هر عضو همسایگی در نظر گرفته می‌شود و جفت آن عضو نیز از میان همسایگی آن انتخاب خواهد شد. مثلا بهترین همسایه به عنوان جفت آن انتخاب می شود. در این روش گوناگونی تا حدبیشتری نسبت به روشهای پیشین حفظ می‌شود. همسایگیهای مختلفی برای این روش تعریف شده اند مانند همسایگی خطی یا همسایگی صلیبی کامل دوبعدی یا نیم صلیبی دو بعدی یا ستاره‌ای و ...

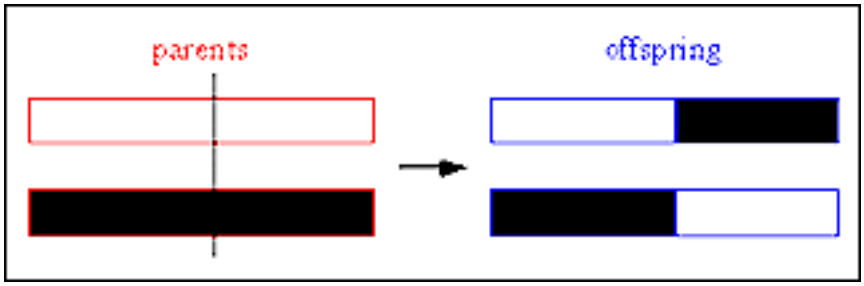


شکل ۴. حلقه‌ی کامل و نیم‌حلقه در همسایگی خطی (Linear neighbourhood)



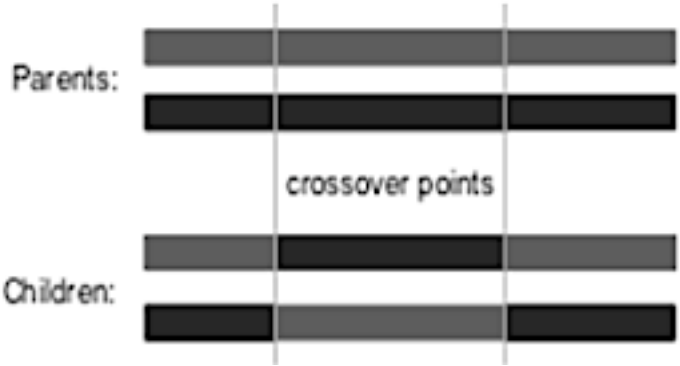
شکل ۵. صلیب کامل و نصفه در همسایگی دو بعدی (Twodimensional neighbourhood)

روشهای متفاوتی نیز برای تولید مثل کردن مطرح شده است. ساده‌ترین آنها جفت گیری تک نقطه‌ای است. در آن یک نقطه به تصادف روی کروموزوم انتخاب شده و نیمی از کروموزوم اول تا رسیدن به نقطه مربوط به یک والد و نیمی دیگر مربوط به والد دیگر می باشد.

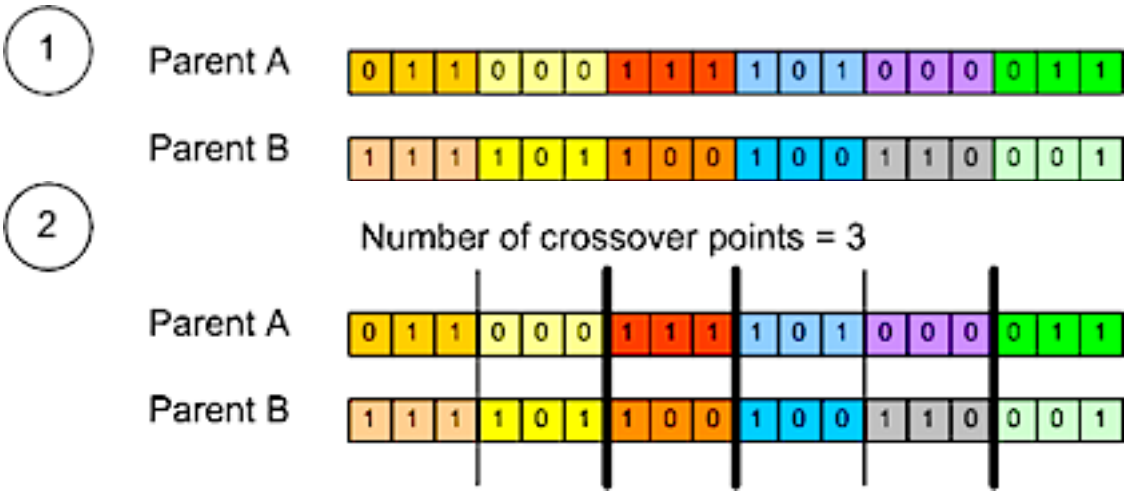


شکل ۴. تقاطع تک نقطه‌ای (single point crossover)

در روشهای دیگر تعداد نقاط تقاطع را افزایش می‌دهند، مثلا فرض کنید کمیتهای استخراج شده برای جواب یک مسئله با ارزشهای به طور غیرمساوی پراکنده شده اند، در این صورت در روش تک نقطه‌ای به احتمال ۵۰ درصد نیمه‌ی پر ارزش هریک از والدین را از دست می‌دهیم. برای کاهش این احتمال روشهای تلفیق دونقطه ای و چندنقطه ای مطرح شدند.

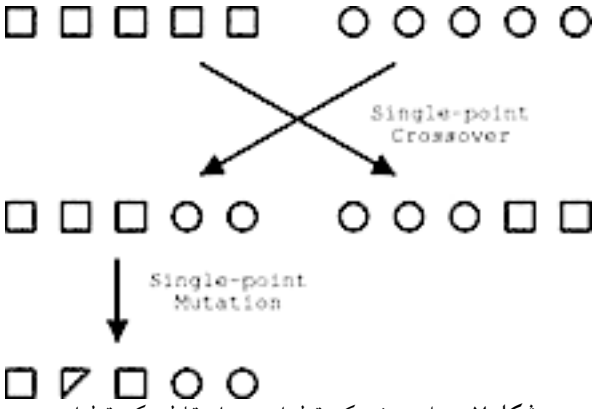


شکل ۵. تقاطع دو نقطه‌ای (binary single-point crossover)



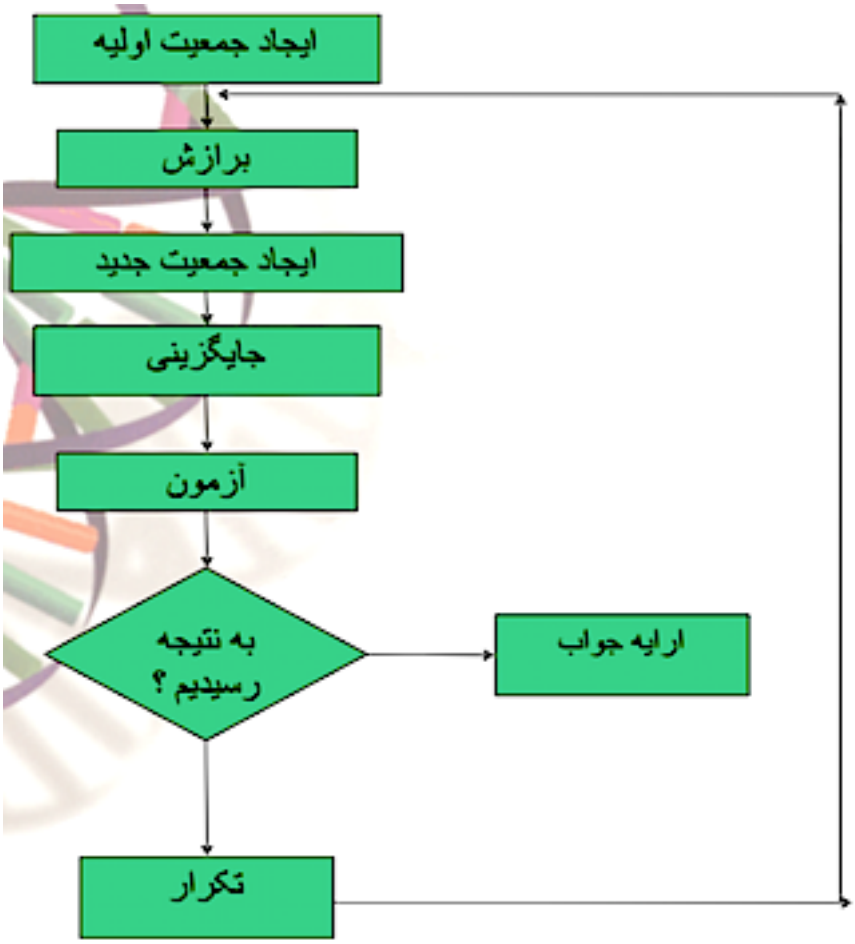
شکل ۶. تقاطع چند نقطه‌ای (binary multi-point crossover)

همچنین روشهای متفاوتی برای جهش وجود دارد. احتمال جهش نیز در مسائل گوناگون متفاوت است ولی معمولا احتمال آن کم در نظر گرفته می‌شود. روشهای مختلفی نیز برای اعمال جهش وجود دارد. مثلا تغییر تک نقطه‌ای یکی از خانه‌های کروموزوم را به تصادف تغییر می‌دهد. همچنین جهش چندنقطه‌ای و انواع دیگری از جهش را نیز داریم:



شکل ۷. عمل جهش تک نقطه ای پس از تقاطع تک نقطه‌ای

شرط پایانی الگوریتم نیز می‌تواند رسیدن به یک زمان خاص یا به یک برازندگی مشخص مدنظر قرار گرفته شود. کل الگوریتم ژنتیک به شکل زیر است.



شکل ۸. فلوچارت الگوریتم ژنتیک

مراجع

[1] Kenneth De Jong, David B Fogel and Hans-Paul Schwefel. A history of evolutionary compution

[2] محمد البرزی. الگوریتم ژنتیک
[3] دکتر هادی شهریار شاه حسینی، دکتر محمدرضا موسوی میرکلائی و مهندس مرتضی ملاجعفری. الگوریتمهای تکاملی
[4] محمد شمس جاوی. پیاده سازی و حل مسائل کاربردی با الگوریتم ژنتیک
[5] چارلز داروین، مترجم دکترنورالدین فرهیخته . منشاء انواع