

الگوریتم ژنتیک

الگوریتم ژنتیک که روش بهینه‌سازی الهام گرفته از طبیعت جاندار (موجودات زنده) است که می‌توان در طبقه‌بندی‌ها، از آن به عنوان یک روش عددی، جستجوی مستقیم و تصادفی یاد کرد. این الگوریتم، الگوریتمی مبتنی بر تکرار است و اصول اولیه آن همانطور که بیشتر اشاره شد از علم ژنتیک اقتباس گردیده است و با تقلید از تعدادی از فرآیندهای مشاهده شده در تکامل طبیعی اختراع شده است و به طور موثری از معرفت قدیمی موجود در یک جمعیت استفاده می‌کند، تا حل‌های جدید و بهبود یافته را ایجاد کند. این الگوریتم در مسائل متنوعی نظیر بهینه‌سازی، شناسایی و کنترل سیستم، پردازش تصویر و مسایل ترکیبی، تعیین توپولوژی و آموزش شبکه‌های عصبی مصنوعی و سیستم‌های مبتنی بر تصمیم و قاعده به کار می‌رود.

در اینجا لازم می‌بینم که برای ورود به موضوع اصلی این نوشتار (الگوریتم ژنتیک) مروری بر مطالبی که پیشتر در مورد مباحث «ژنتیک» و «الگوریتم» ارائه شد داشته باشیم.

چنانکه قبل‌تر اشاره شد، علم ژنتیک، علمی است که دربارهٔ چگونگی توارث و انتقال صفحات بیولوژیکی از نسلی به نسل بعد صحبت می‌کند. عامل اصلی انتقال صفحات بیولوژیکی در موجودات زنده کروموزوم‌ها^۱ و ژن‌ها^۲ می‌باشد و نحوه عملکرد آنها به گونه‌ای است که در نهایت ژن‌ها و کروموزوم‌های برتر و قوی مانده و ژن‌های ضعیف‌تر از بین می‌روند. به عبارت دیگر نتیجه عملیات متقابل ژن‌ها و کروموزوم‌ها باقی ماندن موجودات اصلح و برتر می‌باشد.

همچنین مجدداً یادآور می‌شویم که این الگوریتم برای بهینه‌سازی، جستجو و یادگیری ماشین مورد استفاده قرار می‌گیرد. اساس این الگوریتم قانون تکامل داروین (بقا بهترین) است که می‌گوید: موجودات ضعیف‌تر از بین می‌روند و موجودات قوی‌تر باقی می‌مانند. در واقع تکامل فرآیندی است که روی رشته‌ها صورت می‌گیرد، نه روی موجودات زنده‌ای که معرف موجودات رشته است. در واقع، قانون انتخاب طبیعی برای بقا می‌گوید که هر چه امکان تطبیق موجود بیشتر باشد بقای موجود امکان‌پذیرتر است و احتمال تولید مثل بیشتری، برایش وجود دارد. این قانون بر اساس پیوند بین رشته‌ها و عملکرد ساختمان‌های رمزگشایی شده آنها می‌باشد.

۱ Chromosome

۲ Gene

الگوریتم ژنتیک به دلیل تقلید نمودن از طبیعت دارای چند اختلاف اساسی با روش‌های جستجوی مرسوم می‌باشد که در زیر به تعدادی از آنها اشاره می‌کنیم.

- الگوریتم ژنتیک با رشته‌های بیتی کار می‌کند که هر کدام از این رشته‌ها کل مجموعه متغیرها را نشان می‌دهد حال آنکه بیشتر روش‌ها به طور مستقل با متغیرهای ویژه برخورد می‌کنند.

- الگوریتم ژنتیک برای راهنمایی جهت جستجو، انتخاب تصادفی انجام می‌دهد که به این ترتیب به اطلاعات مشتق نیاز ندارد.

- در الگوریتم ژنتیک روش‌های جستجو بر اساس مکانیزم انتخاب و ژنتیک طبیعی عمل می‌نمایند.

این الگوریتم‌ها مناسب‌ترین رشته‌ها را از میان اطلاعات تصادفی سازماندهی شده انتخاب می‌کنند. در هر نسل یک گروه جدید رشته‌ها با استفاده از بهترین قسمت‌های دنباله‌های قبلی و بخش جدید اتفاقی برای رسیدن به یک جواب مناسب به وجود می‌آیند. با وجود اینکه الگوریتم‌ها تصادفی هستند ولی در زمره الگوریتم‌های تصادفی ساده نیستند. آنها به طور کارآمدی به اکتشاف اطلاعات گذشته در فضای جستجو می‌پردازند تا در یک نقطه جستجوی جدیدی با پاسخ‌های بهتر به سمت بهترین جواب پیش روند. هنگام پیش‌آمدسازی^۳ الگوریتم‌های ژنتیک عمل پیش‌آمدسازی ساده را نمی‌پیمایند بلکه آنها داده‌های پیشین را با تفکر انتخاب جستجوی جدید برای رسیدن پیشرفت مورد نظر توأم می‌کنند.

- الگوریتم ژنتیک در هر تکرار چند نقطه از فضای جستجو را در نظر می‌گیرد بنابراین شانس اینکه به یک ماکزیمم محلی همگرا شود کاهش می‌یابد.

در بیشتر روش‌های جستجوی مرسوم (روش گرادیان) قاعده تصمیم حاکم به این صورت عمل می‌کند که از این یک نقطه به نقطه دیگر حرکت می‌کند. این روش‌ها می‌توانند در فضاهاى جستجو دارای چند بیشینه خطرناک باشند. زیرا ممکن است آنها به یک ماکزیمم محلی همگرا شوند. لیکن الگوریتم ژنتیک جمعیت‌های کاملی از رشته‌ها (نقاط) را تولید می‌کند سپس هر نقطه را به صورت انفرادی امتحان می‌کند و با ترکیب محتویات آنها یک جمعیت جدید را که شامل نقاط بهبود یافته است تشکیل می‌دهد. صرف نظر از انجام یک

جستجو ملاحظه هم‌زمان تعدادی نقطه در الگوریتم ژنتیک آنها را با ماشین‌های موازی تطبیق می‌سازد زیرا در اینجا تکامل هر نقطه یک فرآیند مستقل است. لذا الگوریتم ژنتیک فقط نیاز به اطلاعاتی در مورد کیفیت حل‌های ایجاد شده به وسیله هر مجموعه از متغیرها دارد، در صورتی که بعضی از روش‌های بهینه‌سازی نیاز به اطلاعات یا حتی نیاز به شناخت کامل از ساختمان مسأله و متغیرها دارند. چون الگوریتم ژنتیک نیاز به چنین اطلاعات مشخصی از مسأله ندارد بنابراین قابل انعطاف‌تر از بیشتر روش‌های جستجو است. همچنین الگوریتم ژنتیک از روش‌های جستجوی نوعی که برای راهنمایی جهت روش‌های جستجویشان از انتخاب تصادفی استفاده می‌کنند متفاوت است زیرا اگر چه برای تعریف روش‌های تصمیم‌گیری از تصادف و شانس استفاده می‌کند ولی در فضای جستجو به صورت تصادفی قدم نمی‌زند. [13]

- الگوریتم ژنتیک از قوانین احتمالی پیروی می‌کند و نه از قوانین قطعی. [5]

۲-۳- مکانیزم الگوریتم ژنتیک

الگوریتم ژنتیک به عنوان یک الگوریتم محاسباتی بهینه‌سازی با در نظر گرفتن مجموعه‌ای از نقاط فضای جواب در هر تکرار محاسباتی به نحو مؤثری نواحی مختلف فضای جواب را جستجو می‌کند. در مکانیزم جستجو گرچه مقدار تابع هدف تمام فضای جواب محاسبه نمی‌شود ولی مقدار محاسبه شده تابع هدف برای هر نقطه، در متوسط‌گیری آماری تابع هدف برای هر نقطه، در متوسط‌گیری آماری تابع هدف در کلیه زیر فضاهایی که آن نقطه به آنها وابسته بوده دخالت داده می‌شود و این زیر فضاهای به طور موازی از نظر تابع هدف متوسط‌گیری آماری می‌شوند. این مکانیزم را توازی ضمنی^۴ می‌گویند. این روند باعث می‌شود که جستجوی فضا به نواحی از آن که متوسط آماری تابع هدف در آنها زیاد بوده و امکان وجود نقطه بهینه مطلق در آنها بیشتر است سوق پیدا کند. چون در این روش برخلاف روش‌های تک‌مسیری فضای جواب به طور همه جانبه جستجو می‌شود، امکان کمتری برای همگرایی به یک نقطه بهینه محلی وجود خواهد داشت.

امتیاز دیگر این الگوریتم آن است که هیچ محدودیتی برای تابع بهینه شونده، مثل مشتق‌پذیری یا پیوستگی لازم ندارد و در روند جستجو خود تنها به تعیین مقدار تابع هدف در نقاط مختلف نیاز دارد و هیچ

اطلاعات کمکی دیگری، مثل مشتق تابع را استفاده نمی‌کند. لذا می‌توان در مسائل مختلف اعم از خطی، پیوسته یا گسسته استفاده می‌شود و به سهولت با مسائل مختلف قابل تطبیق است.

در هر تکرار هر یک از رشته‌های موجود در جمعیت رشته‌ها، رمزگشایی شده و مقدار تابع هدف برای آن به دست می‌آید. بر اساس مقادیر به دست آمده تابع هدف در جمعیت رشته‌ها، به هر رشته یک عدد برازندگی نسبت داده می‌شود. این عدد برازندگی احتمال انتخاب را برای هر رشته تعیین خواهد کرد. بر اساس این احتمال انتخاب، مجموعه‌ای از رشته‌ها انتخاب شده و با اعمال عملگرهای ژنتیکی روی آنها رشته‌های جدید جایگزین رشته‌هایی از جمعیت اولیه می‌شوند تا تعداد جمعیت رشته‌ها در تکرارهای محاسباتی مختلف ثابت باشد.

مکانیزم‌های تصادفی که روی انتخاب و حذف رشته‌ها عمل می‌کنند به گونه‌ای هستند که رشته‌هایی که عدد برازندگی بیشتری دارند، احتمال بیشتری برای ترکیب و تولید رشته‌های جدید داشته و در مرحله جایگزینی نسبت به دیگر رشته‌ها مقاوم‌تر هستند. بدین لحاظ جمعیت دنباله‌ها در یک رقابت بر اساس تابع هدف در طی نسل‌های مختلف، کامل شده و متوسط مقدار تابع هدف در جمعیت رشته‌ها افزایش می‌یابد. بطور کلی در این الگوریتم ضمن آنکه در هر تکرار محاسباتی، توسط عملگرهای ژنتیکی نقاطی جدید از فضای جواب مورد جستجو قرار می‌گیرند توسط مکانیزم انتخاب، روند جستجوی نواحی از فضا را که متوسط آماری تابع هدف در آنها بیشتر است، کنکاش می‌کند. بر اساس سیکل اجرایی فوق، در هر تکرار محاسباتی، توسط عملگرهای ژنتیکی نقاط جدیدی از فضای جواب مورد جستجو قرار می‌گیرند توسط مکانیزم انتخاب، روند جستجو نواحی از فضا را که توسط آماری تابع هدف در آنها بیشتر است، کنکاش می‌کند. که بر این اساس، در هر تکرار محاسباتی، سه عملگر اصلی روی رشته‌ها عمل می‌کند؛ این سه عملگر عبارتند از: دو عملگر ژنتیکی و عملگر انتخابی تصادفی.

«گلد برگ»^۵ الگوریتم ژنتیکی «جان هولند» را با عنوان الگوریتم ژنتیک ساده^۶ معرفی می‌کند؛ الگوریتم

ژنتیک را از الگوریتم ژنتیک طبیعی اقتباس کردند.

در فصل یک گفتیم که: بدن همه موجودات زنده از سلول‌ها تشکیل شده است و در هر سلولی دسته کروموزوم‌های یکسانی وجود دارد. کروموزوم‌ها رشته‌هایی از DNA هستند که در واقع الگویی برای تمام بدن هستند. هر کروموزومی محتوی دسته‌هایی DNA است که ژن نامیده می‌شوند و هر ژنی پروتئین خاصی را رمزگذاری می‌کند. اساساً می‌توان گفت که هر ژن، ویژگی خاصی (مثلاً رنگ چشم) را رمزگذاری می‌کند. حالت‌های مختلف یک خصیصه (آبی، قهوه‌ای) آلل^۷ نامیده می‌شود. هر ژنی موقعیت خاص خود را بر روی کروموزوم دارد که این موقعیت لوکاس^۸ نامیده می‌شود. مجموعه کاملی از مواد ژنتیکی (همه کروموزوم‌ها) ژنوم نامیده می‌شود. دسته خاصی از ژن‌های موجود در ژنوم، ژنوتیپ نامیده می‌شود. ژنوتیپ به همراه تغییرات پس از تولد، پایه و اساس فنوتیپ موجود زنده (ارگانیزم)، ویژگی‌های فیزیکی و ذهنی از قبیل رنگ چشم و هوش و غیره است.

در تولید مثل، ابتدا ترکیب (یا تغییر)^۹ اتفاق می‌افتد. ژن‌های والدین برای ایجاد کروموزوم‌های جدید ترکیب می‌شوند. سپس جنین تشکیل شده دچار تغییر می‌شود. جهش^{۱۰} به این معناست که عناصر DNA کمی تغییر پیدا می‌کنند و این تغییرات اغلب نتیجه نسخه‌برداری غلط از ژن‌های والدین است. میزان شایستگی^{۱۱} موجود زنده (جنین) به واسطه بقای آن اندازه گیری می‌شود.

در الگوریتم ژنتیک، مجموعه ای از متغیرهای طراحی را توسط رشته‌هایی با طول ثابت^{۱۲} یا متغیر^{۱۳} کدگذاری می‌کنند که در سیستم‌های بیولوژیکی آنها را کروموزوم یا فرد^{۱۴} می‌نامند. هر رشته یا کروموزوم یک نقطه پاسخ در فضای جستجو را نشان می‌دهد. به ساختمان رشته‌ها یعنی مجموعه‌ای از پارامترها که توسط یک کروموزوم خاص نمایش داده می‌شود ژنوتیپ^{۱۵} و به مقدار رمزگشایی آن فنوتیپ^{۱۶} می‌گویند.

۷ Allele

۸ Locus

۹ Crossover

۱۰ Mutation

۱۱ Fitness

۱۲ Fixed Length

۱۳ variable

۱۴ Individual

۱۵ Genotype

۱۶ Phenotype

الگوریتم‌های وراثتی فرآیندهای تکراری هستند، که هر مرحله تکراری را نسل و مجموعه‌هایی از پاسخ‌ها در هر نسل را جمعیت نامیده‌اند.

الگوریتم‌های ژنتیک، جستجوی اصلی را در فضای پاسخ به اجرا می‌گذارند. این الگوریتم‌ها با تولید نسل^{۱۷} آغاز می‌شوند که وظیفه ایجاد مجموعه نقاط جستجوی اولیه به نام «جمعیت اولیه»^{۱۸} را بر عهده دارند و به طور انتخابی یا تصادفی تعیین می‌شوند. از آنجایی که الگوریتم‌های ژنتیک برای هدایت عملیات جستجو به طرف نقطه بهینه از روش‌های آماری استفاده می‌کنند، در فرآیندی که به انتخاب طبیعی وابسته است، جمعیت موجود به تناسب برازندگی افراد آن برای نسل بعد انتخاب می‌شود. سپس عملگرهای ژنتیکی شامل انتخاب^{۱۹}، پیوند(ترکیب)، جهش و دیگر عملگرهای احتمالی اعمال شده و جمعیت جدید به وجود می‌آید. پس از آن جمعیت جدیدی جایگزین جمعیت پیشین می‌شود و این چرخه ادامه می‌یابد.

معمولاً جمعیت جدید برازندگی بیشتری دارد این بدان معناست که از نسلی به نسل دیگر جمعیت بهبود می‌آید. هنگامی جستجو نتیجه‌بخش خواهد بود که به حداکثر نسل ممکن رسیده باشیم یا همگرایی حاصل شده باشد و یا معیارهای توقف برآورده شده باشد.[13]

۲-۴- عملگرهای الگوریتم ژنتیک

به طور خلاصه الگوریتم ژنتیک از عملگرهای زیر تشکیل شده است:

۲-۴-۱- کدگذاری^{۲۰}

این مرحله شاید مشکلترین مرحله حل مسأله به روش الگوریتم باشد. الگوریتم ژنتیک به جای اینکه بر روی پارامترها یا متغیرهای مسأله کار کند، با شکل کد شده آنها سروکار دارد. یکی از روشهای کد کردن،

۱۷ Seeding

۱۸ Initial Population

۱۹ Selection

۲۰ Encoding

کد کردن دودویی می باشد که در آن هدف تبدیل جواب مسأله به رشته‌ای از اعداد باینری (در مبنای ۲) است.

۲-۴-۲- ارزیابی^{۲۱}

تابع برازندگی را از اعمال تبدیل مناسب بر روی تابع هدف یعنی تابعی که قرار است بهینه شود به دست می‌آورند. این تابع هر رشته را با یک مقدار عددی ارزیابی می‌کند که کیفیت آن را مشخص می‌نماید. هر چه کیفیت رشته جواب بالاتر باشد مقدار برازندگی جواب بیشتر است و احتمال مشارکت برای تولید نسل بعدی نیز افزایش خواهد یافت.

۲-۴-۳- ترکیب^{۲۲}

مهمترین عملگر در الگوریتم ژنتیک، عملگر ترکیب است. ترکیب فرآیندی است که در آن نسل قدیمی کروموزوم‌ها با یکدیگر مخلوط و ترکیب می‌شوند تا نسل تازه‌ای از کروموزوم‌ها بوجود بیاید. جفت‌هایی که در قسمت انتخاب به عنوان والد در نظر گرفته شدند در این قسمت ژن‌هایشان را با هم مبادله می‌کنند و اعضای جدید بوجود می‌آورند. ترکیب در الگوریتم ژنتیک باعث از بین رفتن پراکندگی یا تنوع ژنتیکی جمعیت می‌شود زیرا اجازه می‌دهد ژن‌های خوب یکدیگر را بیابند.

^{۲۱} Evaluation

^{۲۲} Crossover

جهش نیز عملگر دیگری هست که جواب‌های ممکن دیگری را متولد می‌کند. در الگوریتم ژنتیک بعد از اینکه یک عضو در جمعیت جدید بوجود آمد هر ژن آن با احتمال جهش، جهش می‌یابد. در جهش ممکن است ژنی از مجموعه ژن‌های جمعیت حذف شود یا ژنی که تا به حال در جمعیت وجود نداشته است به آن اضافه شود. جهش یک ژن به معنای تغییر آن ژن است و وابسته به نوع کدگذاری روش‌های متفاوت جهش استفاده می‌شود.

۲-۴-۵- رمزگشایی^{۲۴}

رمزگشایی، عکس عمل رمزگذاری است. در این مرحله بعد از اینکه الگوریتم بهترین جواب را برای مسأله ارائه کرد لازم است عکس عمل رمزگذاری روی جواب‌ها یا همان عمل رمزگشایی اعمال شود تا بتوانیم نسخه واقعی جواب را به وضوح در دست داشته باشیم.

۲-۵- چارت الگوریتم به همراه شبه کد آن

در حالت کلی وقتی یک الگوریتم ژنتیکی اعمال می‌شود چرخه زیر را طی می‌کند:

ابتدا یک جمعیت اولیه از افراد به طور اتفاقی و بدون در نظر گرفتن معیار خاصی انتخاب می‌شود. برای تمامی کروموزوم‌های (افراد) نسل صفر مقدار برآزش با توجه به تابع پردازش که ممکن است بسیار ساده یا پیچیده باشد تعیین می‌شود. سپس با مکانیزم‌های مختلف تعریف شده برای عملگر انتخاب زیرمجموعه‌ای از جمعیت اولیه انتخاب خواهد شد. سپس روی این افراد انتخاب شده عملیات برش و جهش در صورت لزوم با توجه به صورت مسأله اعمال خواهد شد.

حال باید این افراد که مکانیزم الگوریتم ژنتیک در موردشان اعمال شده است با افراد جمعیت اولیه (نسل صفر) از لحاظ مقدار برازش مقایسه شوند. (قطعاً توقع داریم که افراد نسل اول با توجه به یکبار اعمال الگوریتم‌های ژنتیک روی آنان از شایستگی بیشتری برخوردار باشند، اما الزاماً چنین نخواهد بود.) به هر حال افرادی باقی خواهند ماند که بیشترین مقدار برازش را داشته باشند. چنین افرادی در مقام یک مجموعه به عنوان جمعیت اولیه برای مرحله بعدی الگوریتم عمل خواهد کرد.

هر مرحله تکرار الگوریتم یک نسل جدید را ایجاد می‌کند که با توجه به اصلاحاتی که در آن صورت پذیرفته است رو به سوی تکامل خواهد داشت. تذکر این نکته خالی از لطف نیست که هر چند الگوریتم‌های ژنتیک دارای پایه ریاضی متقن و مشخصی نیستند اما به عنوان یک مدل اجرایی و مطمئن که به خوبی نیز پیاده سازی می‌شود کارایی خود را نشان داده‌اند.

۲-۵-۱- شبه کد و توضیح آن

در اینجا الگوریتم ژنتیک به صورت شبه کد بیان شده است.

PSEUDO CODE of GA

```
t:=0; // start with an initial time
nit_population P(t); // initialize a usually random population of individuals
evaluate P(t); // evaluate fitness of all initial individuals of population
while (not done) do // test for termination criterion (time, fitness, etc.)
t:=t+1; // increase the time counter
P':=select_parents P(t); // select a sub-population for offspring production
recombine P' (t); // recombine the "genes" of selected parents
mutate P' (t); // perturb the mated population stochastically
evaluate P'(t); // evaluate it's new fitness
P:=survive P,P'(t); // select the survivors from actual fitness
```

end GA.

طرح کلی یک الگوریتم به شرح زیر می باشد:

۱- آغاز: جمعیت n کروموزومی به صورت تصادفی ایجاد کنید (راه حل های مناسب مسأله).

۲- ارزش گذاری: برازندگی $f(x)$ هر کروموزوم X در جمعیت را ارزیابی کنید.

۳- جمعیت جدید: جمعیت جدیدی را تشکیل دهید. مراحل زیر را تکرار کنید تا جمعیت جدید کامل شود:

انتخاب: دو کروموزوم (والدین) را با توجه به برازندگی آنها از میان جمعیت انتخاب کنید (هر چه برازندگی بیشتر باشد شانس انتخاب بیشتر است).

ترکیب: با توجه به احتمال ترکیب شدن^{۲۵} والدین را برای تشکیل فرزندان^{۲۶} جدید با هم ترکیب کنید.

جهش: با توجه به احتمال جهش^{۲۷} فرزندان را در هر لوکاس (موقعیت در کروموزوم)^{۲۸} مورد جهش

قرار دهید.

پذیرفتن: فرزندان جدید را در جمعیت جدید بگنجانید.

جایگزینی: جمعیت جدید ایجاد شده را برای روند الگوریتم بکار ببرید.

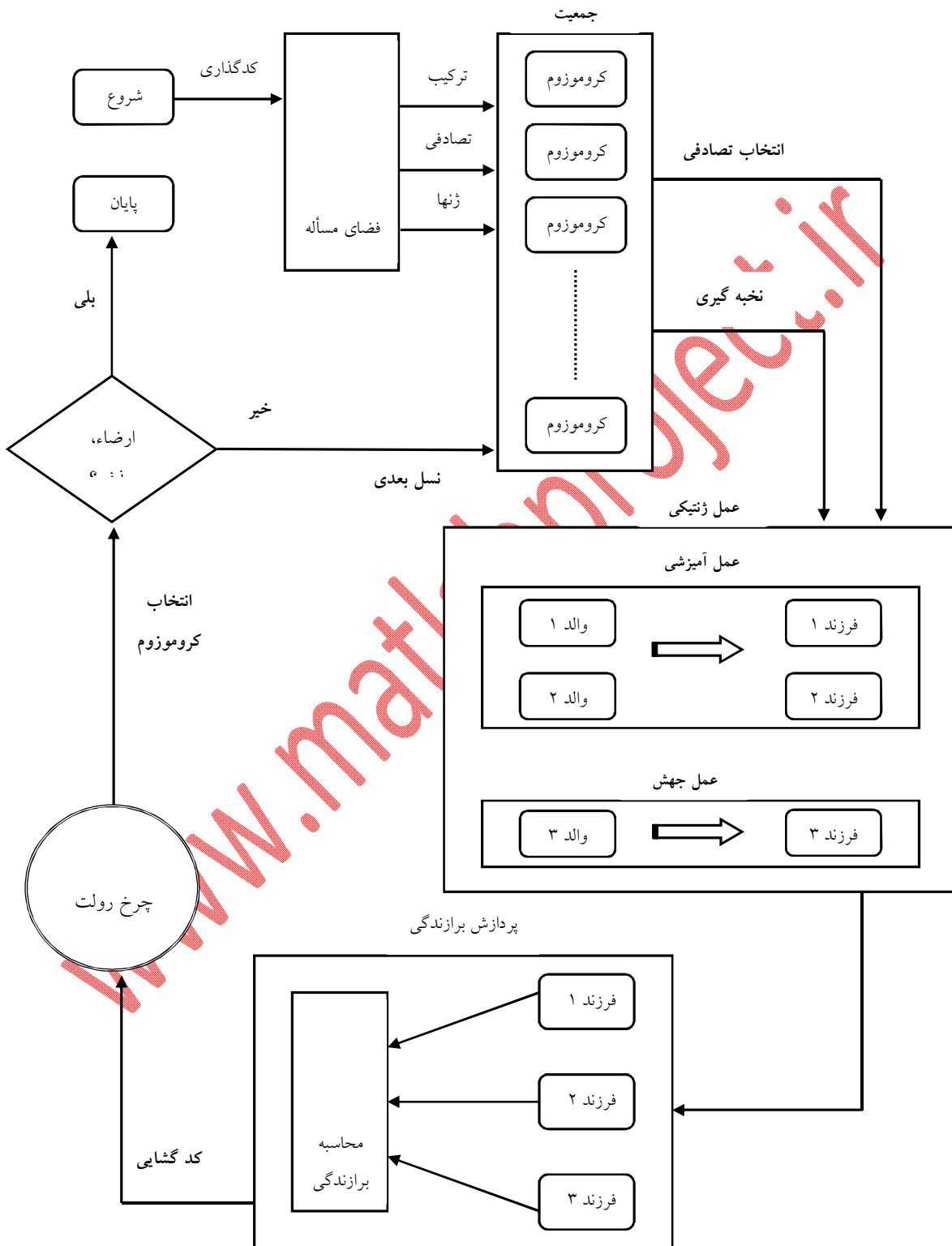
^{۲۵} Probability Crossover

^{۲۶} Offspring

^{۲۷} Probability Mutation

^{۲۸} Lockus

۲-۵-۲- چارت الگوریتم ژنتیک



شکل ۲-۱- چارت الگوریتم ژنتیک.

۲-۶- تابع هدف

تابع هدف، هدف و خواسته ما از طرح مسأله است. یعنی، تابع هدف، شاخصی از نحوه عملکرد افراد در فضای مسأله می باشد. [3]

۲-۷- روش های کد کردن

در این قسمت به بررسی کامل انواع کدینگ (کدگذاری) خواهیم پرداخت. هر چند همانطور که قبلاً هم گفته شد معمولاً از کد کردن دودویی استفاده می شود، اما در بسیاری از موارد کدینگ های دیگری به دلیل ماهیت مسأله مورد نیاز است.

انواع کدینگ (کدگذاری):

- ۱- کدینگ باینری^{۲۹}
- ۲- کدینگ جایگشتی^{۳۰}
- ۳- کدینگ مقدار^{۳۱}
- ۴- کدینگ درختی^{۳۲}

الگوریتم ژنتیک به جای این که بر روی پارامترها یا متغیرهای مسأله کار کند، با شکل کد شده آنها سروکار دارد. یکی از روش های کد کردن، کد کردن دودویی می باشد که در آن هدف تبدیل جواب مسأله به رشته ای از اعداد باینری (در مبنای ۲) است.

تعداد بیت هایی که برای کد گذاری متغیرها استفاده می شود، به دقت مورد نظر برای جواب ها، محدوده تغییر پارامترها و رابطه بین متغیرها وابسته است. رشته یا دنباله ای از بیت ها که به عنوان شکل کد شده یک جواب از مسأله مورد نظر می باشد، کروموزوم نامیده می شود. در حقیقت بیت های یک کروموزوم نقش ژن ها

^{۲۹} Binary Coding

^{۳۰} Permutation Coding

^{۳۱} Value Coding

^{۳۲} Tree Coding

در طبیعت را بازی می‌کنند. یکی از ویژگی‌های اصلی الگوریتم‌های ژنتیک آن است که به طور متناوب بر روی فضای کدینگ و فضای جواب کار می‌کنند. اعمال ژنتیکی بر روی فضای کدینگ یا کروموزوم‌ها اعمال می‌شود، در حالی که انتخاب و ارزیابی بر روی فضای جواب عمل می‌نماید.

در طبیعت نیز به همین شکل است یعنی افراد (کروموزوم‌ها) در یک فضای حقیقی غیر کد شده در حالت فنوتیپ حضور دارند. در صورت کد شدن با هر مکانیزمی حالت ژنوتیپ خود را بروز می‌دهند. ذکر این نکته ضروری است که هر زمان از کدینگ صحبت به میان می‌آید بطور پیش فرض منظور کد کردن از نوع باینری می‌باشد (رشته‌های دودویی). همچنین ساختمان داده مورد استفاده بطور پیش فرض، رشته می‌باشد. [13]

۲-۷-۱- کدینگ باینری

این تبدیل، تبدیل استاندارد در الگوریتم‌های ژنتیک می‌باشد. کد گذاری باینری ساده‌ترین کد گذاری و بهترین تبدیل برای عملگرهای ژنتیک است اما در مسائل پیچیده این نوع تبدیل چندان مناسب نیست چون معمولاً باعث می‌شود طول کروموزوم‌ها برای نگهداری اطلاعات پاسخ، بسیار بزرگ شود. در تبدیل باینری، اعضای جمعیت به رشته‌هایی از ۰ها و ۱ها تبدیل می‌شوند.

به عنوان مثال فرض کنید الگوریتم می‌خواهد ماکزیمم تابع $F(x,y,z)$ را پیدا کند. در نظر بگیرید جستجو باید در اعداد صحیح مثبت و در محدوده ۰ تا ۲۵۵ انجام شود هر پاسخ ممکن شامل سه عدد X و Y و Z می‌باشد.

طول هر عدد در محدوده مورد نظر مسأله در تبدیل باینری حداکثر ۸ بیت می‌باشد. اگر هر کروموزوم را به صورت XYZ در نظر بگیریم بنابراین برای پوشش دادن به تمام پاسخ‌های ممکن لازم است طول کروموزوم $3 \times 8 = 24$ بیت باشد. برای این مسأله کروموزوم C می‌تواند به شکل زیر باشد.

$$C=11010010 \ 11100011 \ 00110111$$

در همین مسأله اگر لازم باشد اعداد منفی نیز جستجو شود می‌توان یک بیت به ابتدای هر رشته اضافه کرد که مثلاً اگر ۰ باشد، عدد، مثبت و اگر ۱ باشد عدد، منفی در نظر گرفته شود.

$$000000001=1$$

$$100000001=-1$$

تبدیل اعداد اعشاری نیز می‌تواند با استفاده از چنین تمهیداتی انجام شود.

البته برای این نوع کدینگ دو روش در قسمت «نمایش رشته‌ها» به طور کامل بیان شده است. [13]

۲-۷-۲- کدینگ جایگشتی

در این روش، کروموزوم‌ها به صورت رشته‌ای از اعداد طبیعی نشان داده می‌شوند که هرکدام از این اعداد، مربوط به پارامتر ویژه‌ای در فضای حل مسئله است. ترتیب قرارگیری این اعداد مهم بوده و طول رشته دقیقاً با تعداد پارامترهای تعریف شده در مسئله برابر است. کاربرد این نوع کدگذاری در حل مسئله فروشنده دوره‌گرد است که تعریف آن در زیر آمده است. [3]

در بسیاری از مسایل مانند مسئله «فروشنده دوره‌گرد» با جایگشت‌های مختلفی از مجموعه راه‌حل‌ها رو به رو هستیم. در این مسئله تعدادی شهر داریم که فاصله میان آنها معلوم است و با شروع از یک شهر و ختم به همان شهر می‌بایست:

۱- از تمام شهرها فقط و فقط یکبار عبور نماییم.

۲- کمترین مسافت ممکنه را طی نماییم.

نکته‌ای که در اینجا مهم است و باعث شده تا کدینگ باینری روش مناسبی برای این مسئله نباشد، این نکته است که حتماً باید برش میان دو والد به نحوی صورت بگیرد که هیچ عنصری تکراری وجود نداشته باشد.

روش تک نقطه (در قسمت ترکیب که جلوتر بیان خواهد شد به طور کامل شرح داده خواهد شد) به این شکل اصلاح می‌شود که تمام قسمت قبل از نقطه برش در والد اول عیناً در فرزند کپی می‌گردد. بقیه ژن‌های والد اول که مطمئناً هنوز در فرزند تکرار نشده‌اند، مطابق با ترتیب قرار گرفتنشان در والد دوم در فرزند کپی می‌شوند.

$$(123456789) + (453689712) = (123456897)$$

شکل ۲-۲- ترکیب تک نقطه.

البته در حالت جایگشتی می توان از PMX هم استفاده نمود.

| After Crossover | Before Crossover |
|------------------|------------------|
| IHD BCJ AGEF | IHD EFG ACBJ |
| HCA EFG IBDJ | HGA BCJ IEDF |

شکل ۲-۳- ترکیب جایگشتی.

برای جهش نیز از مکانیزم زیر استفاده می شود:
 دو موقعیت کاملاً دلخواه انتخاب شده و جای آنها با یکدیگر تعویض می شود.

$$(123456789) = (183456729)$$

شکل ۲-۴- جهش: کدینگ جایگشتی.

[13]

۲-۷-۳- کد گذاری مقدار

در این نوع روش کدگذاری، کروموزومها می توانند هر نوع داده مرتبط با مسأله را در رشته خود اختیار نمایند. این دادهها می توانند از نوع اعداد حقیقی، عبارات منطقی، دستورات جهت یابی، دادههای کد شده به صورت رشتههای حرفی و... باشند. [3]

در این نوع کدگذاری تمامی مکانیزمهای عملگر برش مانند حالت باینری، استفاده می شود. برای عملگر جهش نیز مانند زیر عمل می شود.

$$(1.29 \ 5.68 \ 2.86 \ 4.11 \ 5.55) = (1.29 \ 5.68 \ 2.73 \ 4.22 \ 5.55)$$

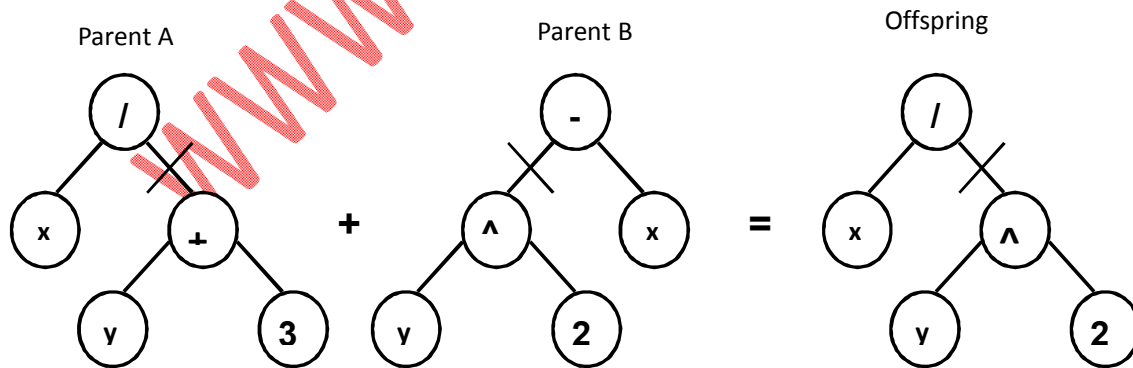
شکل ۲-۵- جهش: کدینگ مقدار.

یعنی به بعضی ژن‌ها بطور تصادفی عددی اضافه شده یا از آنها کم می‌گردد.[13]

۲-۷-۴- کدینگ درخت

این روش که توسط «جان کوزا»^{۳۳} توسعه یافت، بیشتر برای برنامه‌نویسی ژنتیک^{۳۴} توسط نرم‌افزار «لیسپ»^{۳۵} مورد استفاده قرار می‌گیرد، [5] که برنامه‌ها را به عنوان شاخه‌های داده در ساختار درخت نشان می‌دهد. در این روش تغییرات تصادفی می‌توانند با عوض کردن عملگرها یا تغییر دادن ارزش یک گره داده شده در درخت، یا عوض کردن یک زیردرخت با دیگری به وجود آیند.[10] یا به زبان ساده‌تر این طور می‌توان گفت که: در این نوع کد گذاری یک درخت دودویی برای عبارات (کروموزوم) تشکیل می‌دهیم که معادل درخت پارس است و تمام اعمال مربوط به درخت پارس بر روی درخت قابل انجام است.[13]

این روش بیشتر در نرم‌افزار کامپایلر فوق بکار برده می‌شود البته در نرم افزارهایی که تحت عنوان «کدهای الگوریتم ژنتیک» برای یک مسأله خاص نوشته می‌شوند از این نوع روش رمزگذاری و آدرس‌دهی کروموزوم استفاده می‌شود.[5]



شکل ۲-۶- کدینگ درختی.

برای جهش هم یک نود دلخواه تغییر می‌کند.

۲-۸- نمایش رشته‌ها

نمایش مناسب رشته‌ها به ویژگی‌های فضای جستجو بستگی دارد ولی معمولاً به صورت رشته‌های دودوئی نمایش داده می‌شوند. در حل با الگوریتم ژنتیک متغیرها عموماً به صورت دودوئی و با طول رشته ثابت کد گذاری می‌شوند.

برای حل یک مسأله بهینه‌سازی به کمک الگوریتم ژنتیک ابتدا متغیرهای مستقل مسأله تشخیص داده می‌شود و سپس دامنه تغییرات معین می‌شود.

بنابر اینکه هر متغیر پیوسته یا گسسته باشد یکی از روش‌های زیر را انتخاب می‌شود:

الف) متغیرهای پیوسته:

با فرض اینکه متغیر مورد نظر از X_{min} تا X_{max} در حال تغییر باشد و برای نشان دادن آن از یک رشته بیتی استفاده شده باشد.

ب) متغیرهای گسسته:

معمولاً در مسائل مربوط به طراحی استفاده از متغیرهای گسسته اجتناب‌ناپذیر است. برای مثال می‌توان از انتخاب جنس مواد و یا استفاده از جداول استاندارد نام برد. در هر یک از موارد با فرض اینکه متغیر مورد نظر دارای n مقدار قابل قبول باشد. طول زیر رشته معادل آن از رابطه زیر به دست می‌آید.

برای مثال اگر متغیری دارای ۸ مقدار باشد، برای تمام حالات به یک زیر رشته با سه خانه نیاز است.

حال اگر متغیر مورد نظر دارای ۱۰ مقدار متفاوت باشد به یک زیر رشته حداقل با ۴ خانه نیاز است. ولی زیر

رشته‌ای با ۴ خانه توانایی ۲۴ مقدار متفاوت را دارد. برای حل این مشکل می‌توان به تعداد مورد نظر (در

اینجا ۱۴ مورد) از مقادیری که در اختیار است به طور تصادفی انتخاب کرده و به صورت تکراری جایگزین

کرد.

برای مثال فرض کنید در یک مسأله طراحی جنس قطعه مورد بررسی به عنوان متغیر بهینه‌سازی از میان جنس‌های آهن، چدن و برنج قابل انتخاب باشد برای بیان این متغیر $L_i = 2$ خواهد بود که به صورت زیر کد گذاری می‌شود:

$$\text{آهن} = 00 \quad \text{برنج} = 01 \quad \text{چدن} = 10 \quad \text{آهن} = 11$$

همانگونه که نشان داده شده است برای فضای چهارم به صورت تصادفی یکی از سه جنس قابل قبول قرار داده شده است پس از اینکه تمام n متغیر طراحی بصورت زیر دسته‌هایی در نظر گرفته شدند با کنار هم قرار دادن این زیر رشته‌ها به یک رشته دودویی از اعداد به طول L_c خواهیم رسید. این رشته از اعداد که معرف یک طرح خواهد بود همان کروموزوم است. فرمول ریاضی که برای محاسبه طول کروموزوم‌ها بکار می‌رود به شکل زیر می‌باشد:

$$L_c = \sum L_i$$

رابطه ۲-۱- محاسبه طول کروموزوم.

حال با داشتن مکان ابتدا و انتهای هر ژن می‌توان از روی کروموزوم مقدار هر ژن را محاسبه کرد. هر کروموزوم باید اطلاعاتی درباره راه‌حلی که نشان می‌دهد داشته باشد. یکی از روش‌های رمز گذاری که بیشتر مورد استفاده قرار می‌گیرد، رشته دوتایی^{۳۶} است. نمونه‌ای از کروموزوم‌ها در این حالت، در شکل زیر نشان داده شده است.

| | |
|--------------|------------------|
| Chromosome 1 | ۱۱۰۱۱۰۰۱۰۰۱۱۰۱۱۰ |
| Chromosome 2 | ۱۰۱۰۰۱۱۱۰۱۱۱۰۱۰۱ |

شکل ۲-۷- نمونه کروموزوم الگوریتم ژنتیکی.

هر کروموزوم به صورت رشته‌ای باینری نشان داده می‌شود. هر بخش موجود در رشته، ویژگی‌هایی از راه‌حل را نشان می‌دهد. احتمال دیگر آهن است که تمام رشته نمایان گر یک عدد باشد. البته راه‌های دیگر

بسیاری برای رمز گذاری وجود دارد. رمز گذاری، عمدتاً به مسأله حل شده بستگی دارد. به طور مثال می توان مستقیماً اعداد صحیح و یا حقیقی را رمز گذاری کرد.

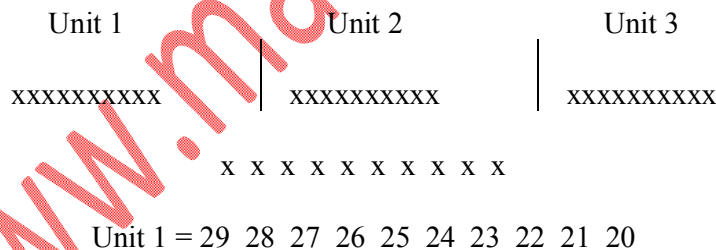
روش معمول برای رمز گشائی متغیرهای گسسته به این صورت است که جدولی از تمام حالات زیر رشته مربوط تشکیل می شود و هنگام رمز گشائی، معادل هر مقدار زیر رشته از جدول انتخاب می شود. [13]

۲-۹- انواع روش های تشکیل رشته

برای تشکیل رشته دو روش وجود دارد:

الف) روش سری

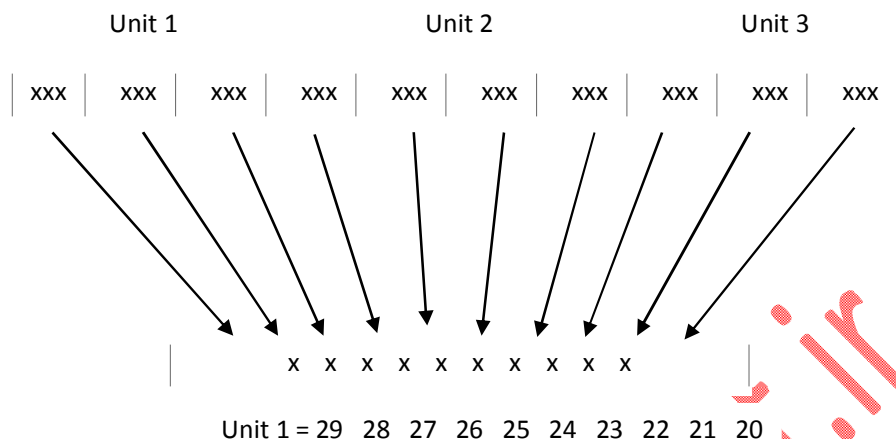
در روش سری بیت های عددی متناظر هر واحد مطابق شکل (۲-۸) کنار هم قرار می گیرند. لازم به توضیح است که تا آخر برنامه هم محل هر واحد ثابت است و هیچ تغییری نمی کند، البته در ابتدای تشکیل رشته هیچ محدودیتی در انتخاب جایگاه وجود ندارد ولی بعد از تشکیل رشته تحت هیچ شرایطی نباید جابجا شوند.



شکل ۲-۸- روش سری.

ب) روش محاطی

در روش محاطی بیت های عددی متناظر واحدها مطابق شکل (۲-۹) طوری کنار هم قرار می گیرند که در NP (تعداد واحدها) بیتی در کنار هم قرار می گیرند و هر بیت متعلق به یک واحد می باشد. به بیان ساده تر اینکه، جایگاه های بیتی در هر NP بیت به طور متناوب متعلق به یک واحد خاص می باشد.



شکل ۲-۹- روش محاطی.

[13]

۲-۱۰- باز گرداندن رشته‌ها به مجموعه متغیرها

در روند اجرای الگوریتم ژنتیک لازم است رشته‌ها به متغیرها تبدیل شوند و به عبارت دیگر نمادهای مربوط به هر رشته (نماد ژنی) به دست آیند، تا از آنجا با قرار دادن متغیرها در تابع هدف، مقدار تابع هدف و در نتیجه ارزیابی آن رشته به دست آید. بنابراین یکی از قسمت‌های مهم الگوریتم ژنتیک قسمت رمز گشایی است که در مرحله ارزیابی انجام می‌شود.

برای باز گرداندن هر رشته به فضای جستجوی واقعی (فضای متغیرها) باید تعداد بیت(های) مربوط به تک تک متغیرها، نوع متغیرها (پیوسته یا گسسته) و محل هر متغیر در رشته مشخص باشند. به هر حال با دانستن اطلاعات مربوط به هر متغیر، زیر رشته متناظر با این متغیر را استخراج و با توجه به محتویات آن مقدار واقعی متغیر را به دست می‌آوریم. در صورتی که متغیر X پیوسته باشد، این فرآیند به صورت زیر انجام می‌شود.

ابتدا زیررشته مربوط به این متغیر مشخص می‌شود. سپس مقدار واقعی آن در مبنای ۱۰ به دست می‌آید. اکنون برای به دست آوردن مقدار متغیر X با توجه داشته باشیم که صورتی که تعداد بیت مربوط به این متغیر

برابر n باشد، در واقع ما متغیر X را در یک فاصله $[0, 2^n - 1]$ نگاشت کرده‌ایم که عدد به دست آمده در مبنای ۱۰ در واقع عدد مربوط به این فاصله می‌باشد. حال مقدار متغیر X با توجه به رابطه (۲-۲) محاسبه می‌شود.

$$X = \left[\frac{X_{\max} - X_{\min}}{2^N - 1} \right] m + X_{\min}$$

رابطه ۲-۲- بازگرداندن رشته‌های متناظر به مجموعه متغیرها.

در صورتی که متغیر مورد نظر ناپیوسته باشد، مقدار اعشاری حاصل از زیر رشته متناظر با متغیر ناپیوسته y در واقع نمایانگر اندیس درایه از یک بردار است که مقدار آن درایه همان مقدار متغیر y می‌باشد. در واقع در حالت‌هایی که متغیرها ناپیوسته باشند، بجای رمز نمودن خود متغیرها اندیس‌های مربوط به آن متغیرها رمز می‌شوند و فرآیندهای ژنی نیز روی اندیس‌ها صورت می‌گیرد نه روی متغیرها. [13]

۲-۱۰-۱- تعداد بیت‌های متناظر با هر متغیر

تعداد بیت‌های متناظر با هر متغیر در صورتی که از رشته‌های بیتی استفاده شود، به صورت زیر به دست می‌آید.

$$2^n = \frac{X_{\max} - X_{\min}}{d}$$

رابطه ۲-۳- محاسبه تعداد بیت‌های متناظر با هر متغیر.

که در رابطه فوق X_{\max} بیست‌ترین مقدار مجاز متغیر و X_{\min} کمترین مقدار مجاز متغیر X است و d نیز دقت مورد نظر برای این متغیر می‌باشد. اکنون با استفاده از رابطه (۲-۴) مقدار n محاسبه می‌کنیم. اگر در محاسبه مقدار n اعشاری شود، کوچکترین عدد صحیح بزرگتر یا مساوی آن را در نظر می‌گیریم.

$$n = \log_2 \left[\frac{X_{\max} - X_{\min}}{d} \right]$$

رابطه ۲-۴- محاسبه مقدار n برای استفاده در رابطه ۲-۳.

همانطور که گفته شد، تعداد بیت مورد نیاز برای تک تک متغیرهای مسأله به دست می‌آید و در نهایت در صورتی که k متغیر داشته باشیم طول هر رشته برابر خواهد بود با:

$$N_s = N_1 + N_2 + \dots + N_k$$

رابطه ۲-۵- محاسبه طول رشته برای k متغیر.

[13]

۲-۱۱- جمعیت

مفهوم جمعیت در الگوریتم ژنتیک شبیه به چیزی است که در زندگی طبیعی وجود دارد. برای مسأله گزاره‌هایی وجود دارند که می‌توانند به عنوان پاسخ، چه درست، چه غلط در نظر گرفته شوند. به این گزاره‌ها پاسخ‌های ممکن یا شدنی می‌گوییم. مثلاً اگر مسأله یافتن ماکزیمم یک تابع در مجموعه اعداد صحیح باشد، تمام اعداد صحیح می‌توانند به عنوان پاسخ شدنی مسأله در نظر گرفته شوند.

در الگوریتم ژنتیک به عنوان اولین مرحله لازم است مجموعه‌ای از جواب‌های شدنی به عنوان جمعیت اولیه ایجاد شود. اعضای این مجموعه معمولاً به صورت تصادفی انتخاب می‌شوند اما در الگوریتم‌های بهینه، از قیدهایی استفاده می‌شود تا جمعیت پراکندگی بیش از حد نداشته باشد. تعداد اعضای جمعیت به نوع مسأله بستگی دارد.

در واقع تعداد اعضا پارامتری است که با تغییر آن می‌توان دقت جواب‌ها و سرعت همگرایی جستجو را بهبود بخشید. در برخی مسائل یک جمعیت ۸ عضوی کاملاً مناسب است در حالی که در برخی یک جمعیت ۱۰۰ عضوی نیز کافی نیست. براساس تجربه بهتر است تعداد اعضای جمعیت عددی بین ۱۰ تا ۱۶۰

باشد. [13]

۲-۱۱-۱- ایجاد جمعیت اولیه

پس از تعیین سیستم کدینگ و مشخص شدن روش تبدیل هر جواب به کروموزوم، باید جمعیت اولیه‌ای از کروموزوم‌ها تولید نمود. در اکثر موارد، جمعیت اولیه به صورت تصادفی تولید می‌شود. اما گاهی اوقات برای بالا بردن سرعت و کیفیت الگوریتم از روش‌های ابتکاری نیز برای تولید جمعیت اولیه استفاده می‌گردد. در هر صورت عمومی‌ترین و راحت‌ترین روش، استفاده از یک رویکرد تصادفی می‌باشد.

اندازه جمعیت اولیه معمولاً به سائز رشته کد شده وابسته است. به عنوان مثال اگر کروموزوم‌ها در یک مسأله ۳۲ بیتی هستند، قطعاً باید جمعیت انتخابی اولیه بیشتر از حالتی باشد که کروموزوم‌ها به عنوان مثال ۱۶ بیتی هستند.

معمولاً احتمال برش بین ۸۰ تا ۹۵ درصد، احتمال جهش بین نیم تا ۱ درصد و اندازه جمعیت بین ۲۰ تا ۳۰ در نظر گرفته می‌شود. آنگاه به کروموزوم‌های انتخاب شده با توجه به یک تابع برازش، مقداری حقیقی که نشان دهنده ارزش آنها است تخصیص داده می‌شود و مراحل الگوریتم‌های ژنتیک ادامه می‌یابد. [13]

۲-۱۱-۲- اندازه جمعیت

«گلدبرگ» برای محاسبه بهترین اندازه جمعیت برای کدهای دودویی متغیرهای پیوسته تا طول حداکثر ۶۰ رشته مقدار زیر را پیشنهاد می‌کند.

$$N_{Pop} = 1.65 \times 2^{(0.21 \times Lc)}$$

رابطه ۲-۶- محاسبه اندازه جمعیت.

[13]

به طور مثال اگر طول هر کروموزوم برابر با ۲۵ باشد، آنگاه داریم: $N_{Pop} = 1.65 \times 2^{(0.21 \times 25)} = 62$.

و یا اگر طول هر کروموزوم برابر با ۳۵ باشد آنگاه خواهیم داشت: $N_{Pop} = 1.65 \times 2^{(0.21 \times 35)} = 270$.

اگر تعداد کروموزوم‌ها بسیار کم باشد، الگوریتم ژنتیک امکان انجام عمل ترکیب کمتری خواهد داشت و تنها بخشی کوچک از فضای جستجو کشف خواهد شد. از طرفی دیگر، اگر تعداد کروموزوم‌ها بسیار زیاد باشد، روند الگوریتم ژنتیک کند خواهد بود، بررسی نشان داده است که در نتیجه برخی محدودیت‌ها (که

عمدتاً به کد گذاری و خود مسأله بستگی دارد) استفاده از جمعیت زیاد، ثمربخش نخواهد بود، زیرا این کار، مسأله را سریعتر از حالتی که جمعیتی متوسط استفاده می‌شود، حل نخواهد کرد.

در صورتی که تعداد اعضای جمعیت بسیار زیاد باشد، اگرچه وضعیت جستجو ممکن است به صورت بهتری نمایش داده شود زیرا با افزایش تعداد رشته‌ها، انتخاب رشته بهتر امکان‌پذیرتر می‌شود، ولی از طرفی نیازمندی‌های حافظه‌های کامپیوتر و زمان اجرای برنامه زیاد می‌شود. اگر تعداد اعضای جمعیت نیز کوچکتر از حد مشخصی باشد، جمعیت مورد نظر فقط قسمت کوچکی از فضای جستجو را نشان می‌دهد و ممکن است جستجو برای رسیدن به حل بهینه در این جمعیت موفقیت‌آمیز نباشد یا مستلزم صرف زمان زیادی باشد، در عمل تعداد اعضای جمعیت مقداری است که به صورت تجربی به دست آمده و نشان داده شده است. با این تعداد رشته در جمعیت، می‌توان به حل‌های مناسبی دست یافت این تعداد ۲ الی ۲/۵ برابر طول هر رشته می‌باشد. [13]

۲-۱۲- محاسبه برازندگی (تابع ارزش)

تابع برازندگی از اعمال تبدیل مناسب بر روی تابع هدف یعنی تابعی که قرار است بهینه شود به دست می‌آید. این تابع هر رشته را با یک مقدار عددی ارزیابی می‌کند که کیفیت آن را مشخص می‌نماید. هر چه کیفیت رشته جواب بالاتر باشد مقدار برازندگی جواب بیشتر است و احتمال مشارکت برای تولید نسل بعدی نیز افزایش خواهد یافت. بسته به اینکه مسأله مورد نظر بیشینه‌سازی یا کمینه‌سازی باشد برازندگی بیشتر مترادف با بیشینه یا کمینه بودن تابع هدف خواهد بود، از آنجایی که الگوریتم ژنتیک طبیعتاً به دنبال بیشینه تابع است باید مسائل کمینه‌سازی به بیشینه‌سازی تبدیل شود.

چندین روش برای تبدیل تابع هدف برازندگی وجود دارد. ساده‌ترین حالت مساوی قرار دادن تابع برازندگی با تابع هدف است. این روش در مسائلی که تابع هدف بایستی بیشینه شود مناسب است.

برای تبدیل مسائل کمینه‌سازی به بیشینه‌سازی روش‌های مختلفی وجود دارد، با فرض اینکه مقدار ϕ_i تابع هدف معادل فرد I ام باشد، ساده‌ترین راه، کم کردن ϕ_i از یک مقدار ثابت C است بطوریکه به ازای تمام نسل‌ها $\phi_i < C$ باشد.

$$f_i = C - \phi_i$$

رابطه ۷-۲- تبدیل مسائل کمینه‌سازی به بیشینه‌سازی.

در صورتی که نتوانیم بزرگترین مقدار تابع هدف را حدس بزنیم می‌توانیم در هر نسل ϕ_{\min} و ϕ_{\max} یافته و برازندگی را به صورت زیر محاسبه کنیم.

$$f_i = (\phi_{\max} + \phi_{\min}) - \phi_i$$

رابطه ۸-۲- تبدیل مسائل کمینه‌سازی به بیشینه‌سازی.

روش دیگر، استفاده از تابع نمائی زیر است:

$$f_i = e^{-\phi_i}$$

رابطه ۹-۲- تبدیل مسائل کمینه‌سازی به بیشینه‌سازی.

معمولاً مسائل بهینه‌سازی دارای قیدهایی هستند که هنگام حل مسئله باید ارضاء شوند در صورتیکه قیدها کم یا ساده باشند و احتمال ارضاء شدن آنها به خودی خود زیاد باشد می‌توان در هر نسل افرادی را که قیدها را ارضاء نمی‌کنند با افراد جدید به طور تصادفی جایگزین کرد ولی در شرایط پیچیده قیدها به صورت تابع جریمه در تابع هدف منظور می‌شوند.

$$\phi_{inew} = P_i + \phi_i$$

رابطه ۱۰-۲- جایگزینی افراد جدید با افرادی که قیدهای مسئله را ارضاء نمی‌کنند.

علامت و مقدار P_i باید به گونه‌ای باشد که موجب شود افرادی از هر نسل که کمتر قیدها را ارضاء می‌کنند در نهایت از برازندگی کمتری برخوردار باشند. [13]

۲-۱۳- انواع روش‌های انتخاب^{۳۸}

در مرحله انتخاب، یک جفت از کروموزوم‌ها برگزیده می‌شوند تا با هم ترکیب شوند، عملگر انتخاب رابط بین دو نسل است و بعضی از اعضای نسل کنونی را به نسل آینده منتقل می‌کند، بعد از انتخاب، عملگرهای ژنتیک روی دو عضو برگزیده اعمال می‌شوند، معیار در انتخاب اعضاء ارزش تطابق آنها می‌باشد اما روند انتخاب حالتی تصادفی دارد.

شاید انتخاب مستقیم و ترتیبی به این شکل که بهترین اعضاء دوبه‌دو انتخاب شوند در نگاه اول روش مناسبی به نظر برسد اما باید به نکته‌ای توجه داشت؛ در الگوریتم ژنتیک ما با ژن‌ها روبه‌رو هستیم، یک عضو با تطابق پایین اگرچه در نسل خودش عضو مناسبی نمی‌باشد اما ممکن است شامل ژن‌های خوب باشد و اگر شانس انتخاب شدنش * باشد، این ژن‌های خوب نمی‌توانند به نسل‌های بعد منتقل شوند. پس روش انتخاب باید به گونه‌ای باشد که به این عضو نیز شانس انتخاب شدن داده شود. راه‌حل مناسب، طراحی روش انتخاب به گونه‌ای است که احتمال انتخاب شدن اعضاء با تطابق بالاتر بیشتر باشد. انتخاب باید به گونه‌ای صورت بگیرد که تا جایی که ممکن است هر نسل جدید نسبت به نسل قبلی‌اش تطابق میانگین بهتری داشته باشد. [13]

روش‌های متداول انتخاب عبارتند از:

- انتخاب چرخ رولت

- انتخاب ترتیبی

- انتخاب بولتزمن

- انتخاب حالت پایدار

- نخبه سالاری

- انتخاب رقابتی

- انتخاب قطع سر

-انتخاب قطعی بریندل

-انتخاب جایگزینی نسلی اصلاح شده

-انتخاب مسابقه

-انتخاب مسابقه تصادفی

[13]

۲-۱۳-۱- انتخاب چرخ رولت

انتخاب چرخ رولت که اولین بار توسط «هولند» پیشنهاد شد یکی از مناسب‌ترین انتخاب‌های تصادفی بوده که ایده آن، احتمال انتخاب می‌باشد. احتمال انتخاب متناظر با هر کروموزوم، براساس برازندگی آن محاسبه شده که اگر f_k مقدار برازندگی کروموزوم k ام باشد، احتمال بقای متناظر با آن کروموزوم عبارت است از:

$$P_k = \frac{f_k}{\sum_{i=1}^n f_i}$$

رابطه ۲-۱۱- احتمال انتخاب در روش چرخ رولت.

حال کروموزوم‌ها را براساس P_k مرتب کرده و q_k که همان مقادیر تجمعی P_k می‌باشد که به صورت

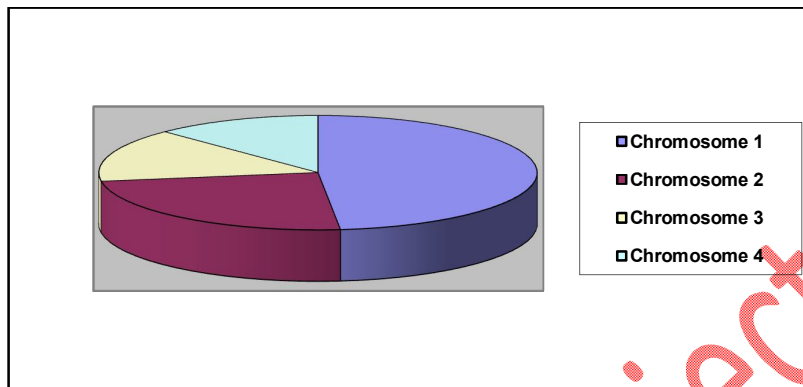
زیر به دست می‌آید:

$$q_k = \sum_i^k P_i$$

رابطه ۲-۱۲- محاسبه مقادیر تجمعی در چرخ رولت.

چرخ رولت به این صورت عمل می‌کند که برای انتخاب هر کروموزوم یک عدد تصادفی بین یک و صفر تولید کرده و عدد مذکور در هر بازه‌ای که قرار گرفت، کروموزوم متناظر با آن انتخاب می‌شود. البته روش پیاده‌سازی چرخ رولت به این شکل است که ما یک دایره را در نظر گرفته و آن را به تعداد کروموزوم‌ها

طوری تقسیم می‌کنیم که هر بخش متناظر با مقدار برازندگی کروموزوم مربوط باشد، حال چرخ را چرخانده و هر کجا که چرخ متوقف شد به شاخص چرخ نگاه کرده، کروموزوم مربوط به آن بخش انتخاب می‌گردد.



شکل ۲-۱۰- چرخ رولت.

انتخاب چرخ رولت، روشی است که نسبت مقدار تطابق، اعضاء را انتخاب می‌کند. این روش یک چرخ رولت را شبیه‌سازی می‌کند تا تعیین کند کدام اعضاء شانس باز تولید را دارند. هر عضو به نسبت تطابقش، تعدادی از بخش‌های چرخ رولت را به خود اختصاص می‌دهد. سپس در هر مرحله انتخاب یک عضو و برگزیده می‌شود و روند آنقدر تکرار می‌شود تا به اندازه کافی، جفت برای تشکیل نسل بعد انتخاب گردد. این روش انتخاب را می‌توان به صورت زیر بیان کرد:

برای مانند V در نظر بگیرید:

$$v = [1, \dots, M]$$

M تعداد عناصر بردار است اگر تعداد اعضاء مجموعه N باشد، هر عضو $i \in 1, \dots, N$ دارای تطابقی مانند f_i می‌باشد. هر عضو i به نسبت f_i ، P_i بار در v تکرار می‌شود. هر چه f_i بیشتر باشد، عضو مکان‌های بیشتری را به خود اختصاص می‌دهد. عدد از تشکیل بردار v یک مقدار تصادفی $1 \leq r \leq M$ انتخاب می‌شود. این مقدار به مکانی در بردار اشاره می‌کند که آن مکان خود معرف عضوی از اعضاء جمعیت است، به عنوان مثال اگر جمعیتی با $N=4$ داشته باشیم و تطابق اعضا عبارت باشد از:

$$f_1 = 10, f_2 = 10, f_3 = 15, f_4 = 25$$

مقدار مجموع تطابق‌ها عبارت است از:

$$\sum_{i=1}^N f_i = 60$$

بردار v را برداری با ۶۰ عنصر در نظر می‌گیریم. این بردار به صورت زیر پر می‌شود. به عضو ۱، ۱۰ مکان، به عضو ۲، ۱۰ مکان، به عضو ۳، ۱۵ مکان و به عضو ۴، ۲۵ مکان اختصاص می‌یابد.

$$v = \{1,1,\dots,1,2,2,\dots,2,3,3,\dots,3,4,4,\dots,4\}$$

حالا r بین ۱ تا ۶۰ به تصادف انتخاب می‌شود. فرض کنید $r=32$ نتیجه می‌شود:

$$v = [32] = 32$$

پس عضو ۳ انتخاب می‌شود. [13]

۲-۱۳-۲- انتخاب حالت پایدار^{۳۹}

در اکثر الگوریتم‌های ژنتیک که در مقالات ارائه شده‌اند، جمعیت جدید به طور کامل توسط فرزندان بوجود می‌آید و این فرزندان جایگزین والدین خود می‌شوند. در بعضی روش‌ها به برخی از اعضای جمعیت قدیمی، اجازه حضور در جمعیت جدید، داده می‌شود. انتخاب حالت پایدار یکی از این روش‌هاست. در این روش، فقط تعداد اندکی از اعضای جمعیت کنونی، با اعضای جدید جایگزین می‌شوند، به عبارت دیگر، بدترین اعضا با فرزندان که از بهترین اعضا بوجود آمده‌اند تعویض می‌شوند اما بافت کلی جمعیت، چندان تغییر نمی‌کند. [13]

۲-۱۳-۳- انتخاب نخبه‌گرایی^{۴۰}

ایده نخبه‌سالاری یا گرابی، ویژگی تازه‌ای به پروسه انتخاب اضافه می‌کند، در نخبه‌سالاری، بهترین عضو هر جمعیت، زنده می‌ماند و در جمعیت بعد حضور دارد، به عبارت دیگر عضوی که بالاترین تطابق را

دارد به طور خودکار به جمعیت جدید منتقل می‌شود. این روش ابتدا در سال ۱۹۷۵ توسط «کندی جونز» معرفی شد. اعمال نخبه سالاری در الگوریتم ژنتیک، معمولاً باعث بهبود کارایی آن می‌شود. [13]

۲-۱۳-۴- انتخاب رقابتی^{۴۱}

این روش تعدادی از اعضای جمعیت را به تصادف انتخاب می‌کند و سپس اگر شرطی خاص برقرار باشد، بهترین یا تعدادی از بهترین‌های آنها را به عنوان والد بر می‌گزیند، اگر شرط برقرار نشود، بدترین عضو یا تعدادی از بدترین‌ها، در تشکیل جمعیت آینده به عنوان والد در نظر گرفته می‌شوند. شکل استاندارد این روش، رقابت دوتایی یا باینری است و به شکل زیر می‌باشد:

۱- ۲ عضو به تصادف انتخاب می‌شوند.

۲- مقدار r بین ۰ و ۱ به تصادف تعیین می‌شود.

۳- پارامتر $1 \leq K \leq 0$ توسط کاربر تعیین می‌شود. (مثلاً $K=0,75$)

۴- اگر $r < k$ عضو برتر و اگر $r \leq K$ عضو بدتر بین این دو عضو، به عنوان والد انتخاب می‌شود.

۵- دو عضو انتخاب شده برای رقابت، به جمعیت بر می‌گردند و می‌توانند دوباره در رقابت شرکت کنند.

روش انتخاب رقابتی می‌تواند به صورت رقابت n تایی نیز انجام شود. [13]

۲-۱۳-۵- انتخاب قطع سر

در این روش که توسط «گلدنبرگ» معرفی و ارائه شده، ابتدا یک عدد T که کوچکتر از صد می‌باشد، تعریف شده سپس کروموزوم‌ها را بر مبنای مقادیر برازندگی مرتب کرده و T درصد برتر را انتخاب می‌کنیم. حال از هر یک از آنها $\frac{100}{T}$ کپی به نسل بعد انتقال می‌دهیم مثلاً فرض کنید $T=20$ و تعداد کروموزوم‌ها نیز ۲۰ عدد باشد، بنابراین ۲۰٪ کروموزوم‌های اول یعنی $\frac{100}{5} = 4$ تای اول را از لیست مرتب شده انتخاب کرده

و از هر کدام ۵ کپی در نظر می‌گیریم. [13]

۲-۱۳-۶- انتخاب قطعی بریندل

این روش که توسط «بریندل» معرفی و ارائه شده، به این صورت است که احتمال انتخاب برای هر کروموزوم طبق فرمول زیر محاسبه می‌شود:

$$(P_k = \frac{f_k}{\sum f_i})$$

رابطه ۲-۱۳-۶- احتمال انتخاب در روش «قطعی بریندل».

و تعداد مورد انتظار برای هر کروموزوم نیز از رابطه به دست می‌آید.

$$e_k = P_k \cdot (Pop_Size)$$

رابطه ۲-۱۴- محاسبه مقادیر تجمعی در روش «قطعی بریندل».

حال هر کروموزوم بر طبق قسمت صحیح مقدار انتظار، به نمونه تخصیص داده می‌شود و سپس جمعیت بر حسب قسمت اعشار تعداد مورد انتظار، مرتب شده و به تعداد مورد نیاز جهت تکمیل جمعیت، به ترتیب از بالای لیست برداشته می‌شود. [13]

۲-۱۳-۷- انتخاب جایگزینی نسلی اصلاح شده

در این روش که توسط «وایتلی»^{۴۲} ارائه شده، ابتدا کروموزوم‌ها براساس مقدار برآزش منظم شده، سپس به تعداد نوزادان تولید شده از انتهای لیست کروموزوم‌ها حذف می‌گردند، آنگاه نوزادان جایگزین کروموزوم‌های حذف شده می‌شوند. مثلاً اگر ۱۰ کروموزوم وجود داشته باشد، ابتدا آنها را مرتب کرده و بعد اگر قرار باشد ۴ نوزاد نیز تولید شود پس از تولید نوزادان، آنها جایگزین ۴ کروموزوم آخر لیست می‌شوند.

[13]

۲-۱۳-۸- انتخاب مسابقه^{۴۳}

در این روش که توسط «گلدبرگ» ارائه شده، به تعداد Pop-Siz^{۴۴}، مجموعه شامل چند عضو که از قبل مشخص شده، تولید کرده و در هر مجموعه بهترین عضو انتخاب می‌گردد. اندازه مجموعه فوق که به آن اندازه مسابقه گفته می‌شود معمولاً برابر ۲ فرض می‌گردد. [13]

۲-۱۳-۹- انتخاب مسابقه تصادفی

این روش که توسط «وزل» ارائه شده، مانند حالت قبل بوده، با این تفاوت که به جای این که مجموعه به صورت تصادفی انتخاب شود با کمک چرخ رولت انتخاب شده و بهترین آن به عنوان یک عضو از نسل جدید در نظر گرفته می‌شود. [13]

۲-۱۴- انواع روش‌های ترکیب

در طبیعت بقای نسل یکی از مهمترین فاکتورهاست و تنها عملگر ممکن برای این امر آمیزش است. در الگوریتم‌های ژنتیکی نیز آمیزش وجود دارد. آمیزش با تعویض ژن‌ها، بین دو کروموزوم انجام می‌گردد و

۴۲ Whitley

۴۳ Tournament Selection

۴۴ تعداد کل جمعیت در هر نسل.

هر کدام از کروموزوم‌ها خصوصیتی از خود را به فرزندان انتقال می‌دهند. بدیهی است کروموزوم‌هایی که دارای برزندگی بیشتری هستند شانس بیشتری برای آمیزش دارند.

مهمترین عملگر در الگوریتم ژنتیک، عملگر ترکیب است. ترکیب، فرآیندی است که در آن نسل قدیمی

کروموزوم‌ها با یکدیگر مخلوط و ترکیب می‌شوند تا نسل تازه‌ای از کروموزوم‌ها بوجود بیاید.

جفت‌هایی که در قسمت انتخاب، به عنوان والد در نظر گرفته شدند، در این قسمت ژن‌هایشان را با هم مبادله می‌کنند و اعضای جدید بوجود می‌آورند.

بر اساس مباحث تئوری (که در ادامه بررسی خواهد شد) ترکیب اعضا با تطابق بالا باعث بوجود آمدن

اعضایی می‌شود که از تطابق میانگین، تطابق بیشتری دارند. ترکیب در الگوریتم ژنتیک باعث از بین رفتن

پراکندگی یا تنوع ژنتیکی جمعیت می‌شود، زیرا اجازه می‌دهد ژن‌های خوب یکدیگر را بیابند. [13]

در ادامه متداولترین روشهای ترکیب شرح داده خواهد شد.

۲-۱۴-۱- جابه‌جایی دودویی^{۴۵}

روش‌های معمول جابه‌جایی تک نقطه^{۴۶}، دو نقطه^{۴۷} و جابه‌جایی یکنواخت^{۴۸} می‌باشد. ساده‌ترین جابجا

کردن، جابه‌جایی تک نقطه‌ای است. در جابه‌جایی تک نقطه‌ای، ابتدا جفت کروموزوم والد (رشته دودویی) در

نقطه مناسبی در طول رشته بریده شده و سپس قسمت‌هایی از نقطه برش، با هم عوض می‌شوند، بدین ترتیب

دو کروموزوم جدید به دست می‌آید که هر نقطه از آن ژن‌هایی را از کروموزوم‌های والد به ارث می‌برند.

برای جابه‌جایی چند نقطه‌ای^{۴۹}، m موقعیت جابجا شدن، $k_i \in \{1, 2, \dots, l-1\}$ که k_i نقطه جابه‌جایی و

l طول کروموزوم می‌باشد را به صورت تصادفی و بدون تکرار انتخاب می‌کنیم، سپس جهت ایجاد فرزندی

جدید بیت‌های بین نقاط مشخص شده در والدین با هم عوض می‌شوند. این عملیات در شکل (۲-۱۱) نشان

۴۵ Binary Crossover

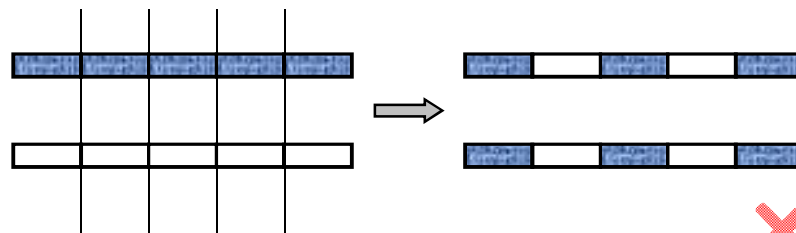
۴۶ Single Point Crossover

۴۷ Two Point Crossover

۴۸ Monotonous Crossover

۴۹ Multi Crossover

داده شده است. فلسفه انجام جابجایی این است که قسمت‌هایی از کروموزوم که بیان‌کننده سهم بسزایی در عملکرد بهتر یک عضو خاص هستند ممکن است در زیر رشته‌های همسایه یافت نشوند.



شکل ۲-۱۱- جابجایی چند نقطه.

به نظر می‌رسد نحوه عملکرد جابجایی در چند نقطه نسبت به روش همگرایی به مقادیر بالاتر برازندگی به پیشرفت و توسعه جستجو در فضای داده‌های مربوطه بیشتر کمک می‌کند، لذا جستجو در دامنه جواب قوی‌تر می‌شود.

«یانگ»^{۵۰} عملکرد جابجایی چند نقطه را مورد بررسی قرار داده و ثابت کرد که عملگر جابه‌جایی بیشتر، عملکرد الگوریتم ژنتیک را کاهش می‌دهد.

عملگرهای جابه‌جایی یک نقطه‌ای و چند نقطه‌ای در جایی اثر می‌کنند که کروموزوم دقیقاً در آن نقاط فقط می‌تواند جدا شود اما عملگر جابه‌جایی یکنواخت پتانسیل جابجا شوندگی را به تمام نقاط یک کروموزوم به صورت یکنواخت نسبت می‌دهد. به این معنی که احتمال جابجا شدن کروموزوم در هر نقطه برابر خواهد بود. یک الگوی بیان‌کننده عمل جابه‌جایی (به همان طولی که کروموزوم‌ها دارند) به صورت تصادفی ایجاد می‌شود و مقدار تعیین شده در هر بیت از این نمونه نشان می‌دهد که کدام یک از والدین به عنوان مرجع مقداردهی برای آن بین از فرزند خواهد بود. تعداد نقاط برش ثابت نیست ولی به طور متوسط برابر $\frac{l}{2}$ است.

$$P_1 = 1011000111$$

$$P_2 = 0001111000$$

$$Mask = 0011001100$$

$$O_1 = 0011110100$$

$$O_2 = 1001001011$$

دو کروموزوم اولیه (والدین) و الگوی عملگر جابه‌جایی و فرزندان حاصل را در نظر بگیرید.

در اینجا مشاهده می‌شود که فرزند اول O_1 بدین صورت ایجاد شده است که اگر بیت مربوط در الگوی جابه‌جایی ($Mask$) ۱ باشد آن بیت در O_1 برابر با مقدار بیت متناظر در P_1 و همچنین اگر بیت مربوط در الگوی جابه‌جایی ۰ باشد مقدار آن بیت در O_1 برابر با مقدار بیت متناظر در P_2 است. رشته O_2 با جابجا کردن P_1 و P_2 و در نظر گرفتن همین شیوه ایجاد شده است یعنی مقدار ۱ در رشته $Mask$ به معنی مقدار بیت متناظر در P_2 برای همان بیت در O_2 است.

مشخصاً عملگر جابه‌جایی یکنواخت همانند عملگر جابه‌جایی چند نقطه‌ای باعث کاهش خطای همگرایی ناشی از طول باینری استفاده شده و نوع کد کردن سری پارامترهای داده شده می‌شود. این مسأله کمک می‌کند که بر خطای همگرایی موجود در حالت جابه‌جایی تک نقطه‌ای در زیررشته‌های کوتاه غلبه کنیم بدون آنکه نیاز به دانستن مقادیر بیت‌های اعضا در کروموزوم‌های ارائه شده داشته باشیم.

«اسپرز»^{۵۱} و «دیجانگ»^{۵۲} نشان داده‌اند که چگونه جابه‌جایی یکنواخت به وسیله احتمال عوض شدن و جابجا شدن بیت‌ها پارامتری می‌شود. این پارامتر فوق‌العاده می‌تواند بدون توصیف یک همگرایی مربوط به طول رشته‌های استفاده شده در کنترل مقدار تغییر یافته در طول ترکیب‌بندی مجدد استفاده شود هنگامی که از عملگر جابه‌جایی یکنواخت در مقادیر حقیقی استفاده شود به آن «ترکیب‌بندی منفصل» گفته می‌شود. در مقایسه‌هایی که بین عملگرهای دودوئی به هر دو صورت تئوری و تجربی انجام شده و نتایج به دست آمده نشان می‌دهد که هیچ یک از این عملگرها نمی‌تواند به طور مطلق بهترین بوده و اختلاف در سرعت این روش‌ها هم نمی‌تواند بیش از ۲٪ باشد.

عملگر جابه‌جایی دیگری که مطرح می‌شود به اسم «مخلوط» (شافل)^{۵۳} است. یک نقطه قطع مجزا انتخاب می‌شود اما قبل از اینکه بیت‌ها تعویض شوند در هر دو والد بیت‌ها به صورت تصادفی جابجا می‌شوند. بعد از ترکیب‌بندی مجدد بیت‌ها در رشته فرزند جایگذاری می‌شوند. این عملگر نیز خطای همگرایی رشته‌ها را با جابه‌جایی تصادفی بیت‌ها در هر جایی که عملگر جابه‌جایی انجام می‌شود حذف می‌کند.

۵۱ Spears

۵۲ Dejong

۵۳ Shuffle

عملگر دیگری نیز عمل جابه‌جایی را مقید می‌کند که همیشه اعضای جدیدی ایجاد کند در هر جایی که ممکن باشد. معمولاً این عملگر بدین صورت عمل می‌کند که مکان نقاط قطع را محدود می‌کند به گونه‌ای که نقاط قطع تنها جایی اتفاق می‌افتد که مقادیر ژن در دو کروموزوم متفاوت است. [13]

۲-۱۴-۲- جابه‌جایی حقیقی^۵

در کد گذاری حقیقی که کروموزوم‌ها به صورت برداری از اعداد حقیقی می‌باشند روش‌های زیادی برای عملگر جابه‌جایی حقیقی ارائه شده که اکثر آنها در دو دسته زیر خلاصه می‌شود:

۱- جابه‌جایی عمومی

۲- جابه‌جایی محاسباتی

عملگرهای جابه‌جایی عمومی با توسعه روش‌های جابه‌جایی دودوئی برای کد گذاری حقیقی تهیه می‌شود که مثال ساده از آن عملگر جابه‌جایی ساده می‌باشد که شامل جابه‌جایی تک نقطه، دو نقطه و چند نقطه است که مشابه همان حالت دودوئی می‌باشند با این تفاوت که در این جابه‌جایی یک بیت دودوئی (1,0) یک عدد حقیقی در رشته است، با فرض اینکه $C = (C_1^1, \dots, C_n^1)$ و $C = (C_1^2, \dots, C_n^2)$ دو کروموزومی می‌باشند که تحت عمل جابه‌جایی قرار می‌گیرند و نقطه $i \in \{1, 2, \dots, n-1\}$ نقطه جابه‌جایی باشد دو کروموزوم جدید که از اعمال عملگر جابه‌جایی ساده حاصل می‌شود و به صورت روابط زیر خواهد بود:

$$H1 = (C_1^1, C_2^1, \dots, C_i^1, C_{i+1}^2, \dots, C_n^2)$$

$$H2 = (C_1^2, C_2^2, \dots, C_i^2, C_{i+1}^1, \dots, C_n^1)$$

رابطه ۲-۱۵- جابه‌جایی ساده.

$Hk = (h_1^k, \dots, h_i^k, \dots, h_n^k)$ جابجایی محاسباتی بر اساس مفهوم ترکیب خطی بردارها تهیه شده است. با این فرض که دو فرزند $k = \frac{1}{2}$ است بر اساس این عملگر تولید شده باشد، در این صورت تحت شرایط مختلف برای λ_1 و λ_2 ضرایب در روابط مذکور، سه نوع مختلف از این عملگر ایجاد می‌شود؛

۱- جابجایی خطی^{۵۵}؛ که در آن λ_1 و λ_2 هر دو حقیقی هستند.

۲- جابجایی سلبی^{۵۶}؛ که در آن $\lambda_1 + \lambda_2 = 1$ است.

۳- جابجایی برجسته (محدب)^{۵۷}؛ که در آن $\lambda_1 + \lambda_2 = 1$ بوده و λ_1 و λ_2 هر دو حقیقی مثبت هستند.

اسامی «جابجایی خطی»، «جابجایی سلبی» و «جابجایی برجسته» از تئوری مجموعه‌های محدب وام گرفته شده است.

عملگر «جابجایی برجسته» معمولاً بیشتر از بقیه کاربرد دارد. این عملگرها به طور شماتیکی در شکل --- نشان داده شده است.

$$h_i^1 = \lambda_1 C_i^2 + \lambda_2 C_i^1 \quad h_i^2 = \lambda_1 C_i^1 + \lambda_2 C_i^2$$

رابطه ۲-۱۶- جابجایی محاسباتی.

[13]

۲-۱۴-۳- ترکیب تک نقطه‌ای

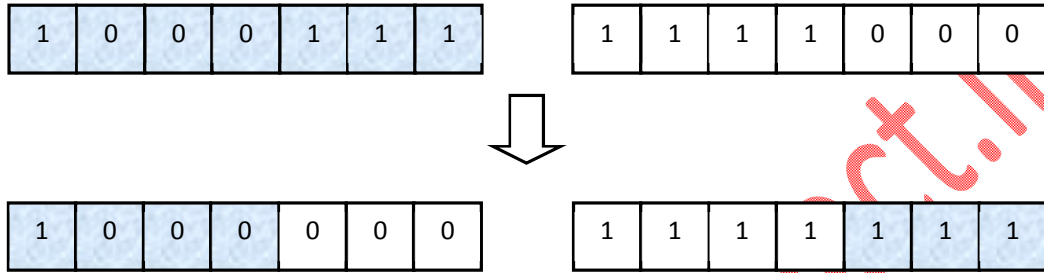
ترکیب تک نقطه‌ای دو کروموزوم را با انتخاب تصادف موقعیتی مانند P ، ترکیب می‌کند P مقداری کمتر یا مساوی طول کروموزومها است. اگر تعداد (طول) ژن‌ها در کروموزومها N باشد از دو کروموزوم والد، دو فرزند به صورت زیر بوجود می‌آید.

۵۵ Linear Crossover

۵۶ Affine Crossover

۵۷ Convex Crossover

یک فرزند با کپی کردن ژن‌های $1 \dots (P_1 - 1)$ از کروموزوم والد اول و ژن‌های $P \dots N$ از کروموزوم دوم، ساخته می‌شود و فرزند دیگر به طور مشابه، این بار با کپی کردن ژن‌های $1 \dots (P_1 - 1)$ از والد دوم و ژن‌های $P \dots N$ از والد اول، بوجود می‌آید. در این نوع ترکیب از دو والد، دو فرزند بوجود می‌آید. به عنوان مثال، این نوع ترکیب در شکل نشان داده شده است. در این مثال $P = 4$ می‌باشد.



شکل ۲-۱۲- ترکیب تک نقطه‌ای.

لازم به ذکر است که اگر P برابر ۱ شود یا برابر طول کروموزوم‌ها، آنگاه دو والد بدون تغییر وارد جمعیت بعدی می‌شوند. [13]

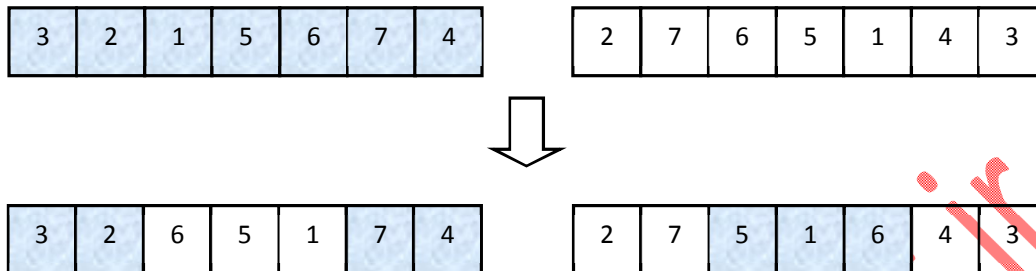
۲-۱۴-۴- ترکیب دو نقطه‌ای

در ترکیب دو نقطه‌ای، دو موقعیت P_1 و P_2 به عنوان موقعیت‌های ترکیب، به طور تصادفی بین ۱ و طول کروموزوم‌ها (N) انتخاب می‌شود. روش ایجاد فرزندان مانند ترکیب تک نقطه‌ای است.

فرزند اول، ژن‌های $1 \dots (P_1 - 1)$ را از والد اول، ژن‌های $P_1 \dots (P_2 - 1)$ را از والد دوم و ژن‌های $P_2 \dots N$ را مجدداً از والد اول، به ارث می‌برد.

فرزند دوم، ژن‌های $1 \dots (P_1 - 1)$ را از والد دوم، ژن‌های $P_1 \dots (P_2 - 1)$ را از والد اول و ژن‌های $P_2 \dots N$ را مجدداً از والد دوم، به دست می‌آورد.

در این روش ترکیب نیز، از یک جفت، دو فرزند بوجود می‌آید، در این روش احتمال اینکه والدین بدون تغییر به جمعیت بعد منتقل شوند، کمتر است. در شکل نمونه‌هایی از این ترکیب با موقعیت‌های ترکیب $P_1 = 2$ و $P_2 = 5$ نشان داده شده است.



شکل ۲-۱۳- ترکیب دو نقطه‌ای.

[13]

۲-۱۴-۵- ترکیب n نقطه‌ای

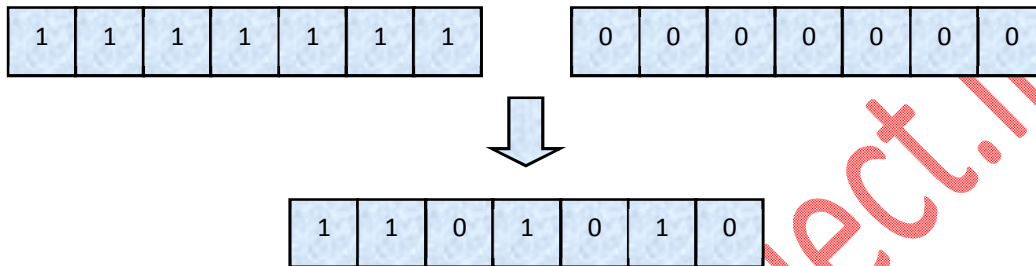
با انتخاب n موقعیت ترکیب و چیدن ژن‌ها مشابه آنچه در ترکیب تک نقطه‌ای و دو نقطه‌ای گفته شد، ترکیب n نقطه‌ای خواهیم داشت. [13]

۲-۱۴-۶- ترکیب یکنواخت

در ترکیب یکنواخت، هر ژن کروموزوم جدید به صورت جداگانه انتخاب می‌شود. هر ژن وابسته به موقعیتش به صورت تصادفی از یکی از دو والد انتخاب می‌شود، مثلاً ژن اول از والد اول، ژن دوم از والد دوم، ژن سوم از والد اول تا ژن آخر، برخلاف ترکیب‌هایی که قبلاً ذکر شد، این نوع ترکیب، یک فرزند بوجود می‌آورد، در واقع در این حالت از یک ماسک استفاده می‌شود.

جمعیت جدیدی که با ترکیب یکنواخت بوجود می‌آید، دارای تنوع ژنتیکی بیشتری نسبت به ترکیب‌های تک نقطه‌ای و دو نقطه‌ای می‌باشد به همین دلیل این نوع ترکیب در جمعیت‌هایی که اعضای کمی دارند اثر بهتری دارد تا جمعیت‌هایی که تعداد اعضای زیادی دارند.

در جمعیت‌های کوچک، ممکن است به تنوع ژنتیکی نیاز باشد تا روش، سریعتر همگرا شود، اما در جمعیت‌های بزرگ، معمولاً تنوع ژنتیکی لازم فراهم است. در شکل (۲-۱۴) نمونه‌ای از ترکیب یکنواخت مشاهده می‌شود.



شکل ۲-۱۴- ترکیب یکنواخت

[13]

۲-۱۴-۷- ترکیب حسابی

ترکیب حسابی به صورت زیر بیان می‌شود.

اگر A و B دو عضو از جمعیت فعلی باشند که به عنوان والد انتخاب شده‌اند از آنها دو فرزند a و

b به صورت زیر بوجود می‌آید:

$$a = \delta A + (1 - \delta)B$$

$$b = \delta B + (1 - \delta)A$$

رابطه ۲-۱۷- ترکیب حسابی.

پارامتر δ مقداری در بازه $[0,1]$ می‌باشد که در هر ترکیب می‌تواند مقدار مختلفی داشته باشد. [13]

۲-۱۴-۸- ترتیب

این روش توسط «دیویس» معرفی گردیده و به روش OX معروف می‌باشد در این روش دو عدد را به صورت تصادفی به عنوان نقاط به دست آورده سپس قسمت مابین را در دو طرف کروموزوم ثابت نگه داشته ولی قسمت‌های دو طرف به این صورت به دست می‌آید که برای نوزاد اول در والد دوم از ابتدای کروموزوم شروع کرده و آنهایی را که در قسمت ما بین نوزاد وجود ندارد در جای خالی قرار می‌گیرند. با مثال زیر بحث روشن می‌شود:

والد اول: ۴۷۶/۳۵۹۸/۱۲

والد دوم: ۸۳۵/۶۴۱/۲۹

برای نوزاد اول قسمت ما بین والد یک، بدون تغییر جابجا می‌شود.

نوزاد اول: ---/۳۵۹۸/---

حال قسمت خالی از روی والد دوم پر می‌گردد.

نوزاد اول: ۷۴۶/۳۵۹۷/۱۲

برای نوزاد دوم نیز به همان صورت عمل می‌شود:

نوزاد دوم: ۳۵۹/۷۶۴۱/۸۲

[13]

۲-۱۴-۹- چرخه

این روش توسط «الیور اسمیت»^{۵۸} و «هولند» معرفی شده و به نام عملگر CX معروف می‌باشد و به اینگونه عمل می‌شود که ابتدا اولین ژن را عیناً از والد اول به نوزاد اول کپی کرده، سپس یک چرخه بین ژن‌هایی که در دو کروموزوم والد اول و دوم وجود دارد ایجاد می‌گردد، نقطه شروع همان ژن فوق می‌باشد. برای ایجاد چرخه باید ابتدا همان ژن را در کروموزوم دوم یافته و مکان آن را در نظر بگیریم. سپس ژن

موجود در همان مکان از کروموزوم اول تثبیت کنیم. این عمل تا جایی که چرخه کامل شود ادامه می‌یابد (تا جایی که به نقطه شروع برسد). در این لحظه برای تکمیل نوزاد اول باید ژنهای باقیمانده از کروموزوم دوم را به همان ترتیب نوزاد اول جایگذاری نمود.

که مثال زیر بحث را روشن می‌کند.

والد اول: ۳۴۶۵۷۸۲۹۱

والد دوم: ۷۹۴۳۵۶۲۸۱

برای تولید نوزاد اول، ابتدا ژن از کروموزوم والد اول را منتقل کرده و عمل ادامه می‌یابد:

نوزاد اول: ۳--۵۷----

حال که چرخه کامل شد عددهای به دست آمده را در والد دوم حذف کرده باقیمانده به ترتیب عبارت

است از ۹۴۶۲۸۱ که به همین حالت در جاهای خالی نوزاد اول قرار می‌گیرد:

نوزاد اول: ۳۹۴۵۷۶۲۸۱

برای نوزاد دوم نیز به همان صورت عمل می‌شود.

نوزاد دوم: ۷--۳۵----

که با روند فوق رشته زیر حاصل می‌شود.

نوزاد دوم: ۷۴۶۳۵۸۲۹۱

[13]

۲-۱۴-۱۰- محدب

در این عملگر اگر والد اول P_1 و والد دوم P_2 باشد نوزاد اول و دوم به صورت زیر حاصل می‌شود:

$$c_1 = \lambda_1 P_1 + \lambda_2 P_2 \quad \text{نوزاد اول:}$$

$$c_2 = \lambda_1 P_2 + \lambda_2 P_1 \quad \text{نوزاد دوم:}$$

اگر $\lambda_1 = \lambda_2 = 0.5$ باشد به آن عملگر «تقاطع می متوسط» می گویند. اگر $\lambda_2 = -0.5$ و $\lambda_1 = 1.5$ باشد به آن «نسبت سلبی» گفته و در صورتی که λ_1 و λ_2 به صورت تصادفی از بازه $[-d, 1=d]$ انتخاب شود به آن «تقاطع میانه توسعه یافته» می گویند.

«تقاطع می متوسط» توسط «دیویس»، «تقاطع میانه توسعه یافته» توسط «مولن بین» و «نسبت سلبی» توسط «رایت» ارائه شده اند، البته «چنگ» و «جن» حالتی را که λ_1 و λ_2 دو عدد تصادفی بوده و دارای شرط $\lambda_1 > 1$ و $\lambda_2 > 0$ و $\lambda_1 + \lambda_2 \leq 2$ باشد را تحت عنوان «عملگر خطی» ارائه نمودند. [13]

۲-۱۴-۱۱- بخش نگاهشسته

این روش که توسط «گلدبرگ» و «لینگل» معرفی شده به روش PMX معروف بوده و در حقیقت همان عملگر دو نقطه برش می باشد که برای حالت خاصی ارائه شده است. در این روش دو عدد به صورت تصادفی به عنوان نقاط برش به دست آورده، سپس قسمت مابین دو نقطه برش را در دو کروموزوم تعویض کرد و آنگاه قسمت های دو طرف طوری مقدار گذاری می شوند که در هیچکدام از دو کروموزوم، تکراری صورت نگیرد. روش کار با یک مثال روشن شده است.

والد اول: ۴۳/۵۶۲۸/۷۹۱

والد دوم: ۶۵/۸۳۴۹/۲۱۷

حال برای تولید نوزاد به این صورت عمل می شود که قسمت مابین را عوض کرده بعد در والد اول از ابتدای کروموزوم شروع نموده هر عددی را که در قسمت مابین کروموزوم جدید نباشد عیناً نوشته و برای تکراری ها جای خالی قرار داده می شود سپس در والد دوم از ابتدای کروموزوم شروع کرده و هر عددی که در نوزاد جدید نباشد به جای محل خالی، گذاشته می شود تا کلیه جاهای خالی پر گردد. برای نوزاد دوم نیز به همین صورت عمل می شود. بنابراین قسمت های میانی را جابجا کرده دو کروموزوم زیر حاصل می شود.

نوزاد اول: ---/۸۳۴۹/---

نوزاد دوم: ---/۵۶۲۸/---

سپس جاهای خالی نوزاد در صورت تکراری نبودن ژن مورد نظر پر می شود.

نوزاد اول: ۱-۸۳۴۹/۷--

حال جاهای خالی باقیمانده نوزاد اول پر می شود.

نوزاد اول: ۶۵/۸۳۴۹/۷۲۱

باتوجه به مطالب گفته شده، نوزاد دوم به دست زیر به دست می آید.

نوزاد اول: ۱۷-۵۶۲۸/--

نوزاد دوم: ۴۳/۵۶۲۸/۹۱۷

[13]

۲-۱۵- احتمال ترکیب

ترکیب لازم نیست در هر نسل اتفاق بیفتد، در واقع ممکن است نسل‌هایی بدون عملگر ترکیب به نسل‌های جدید تبدیل شوند. برای تعیین رخ دادن یا ندادن ترکیب از پارامتری به نام احتمال ترکیب P_c استفاده می شود که این پارامتر مقدار بین ۰ و ۱ است. از آنجاکه ترکیب نقشی اساسی در رشد مقدار میانگین تطابق جمعیت دارد، مقدار P_c بین ۰,۵ تا ۰,۸ و بیشتر بین ۰,۷ تا ۰,۸ در نظر گرفته می شود.

اگر ترکیبی صورت نگیرد، فرزندان دقیقاً همانند والدین خواهند بود. اگر ترکیب صورت بگیرد، فرزندان از بخش‌هایی از کروموزوم‌های والدین به وجود می آیند. اگر احتمال ترکیب ۱۰۰٪ باشد، در این صورت همه فرزندان در نتیجه ترکیب به وجود آمده‌اند. اگر این احتمال ۰٪ باشد، کل نسل جدید در اثر نسخه برداری عینی کروموزوم‌های نسل قدیم بوجود آمده است (این به این معنی نیست که نسل جدید همانند نسل قدیم است). ترکیب با این امید صورت می گیرد که کروموزوم‌های جدید، حاوی بخش‌های مناسب کروموزوم‌های قدیمی است و در نتیجه کروموزوم‌های جدید بهتر خواهد بود. با این حال خوب است که برخی از قسمت‌های

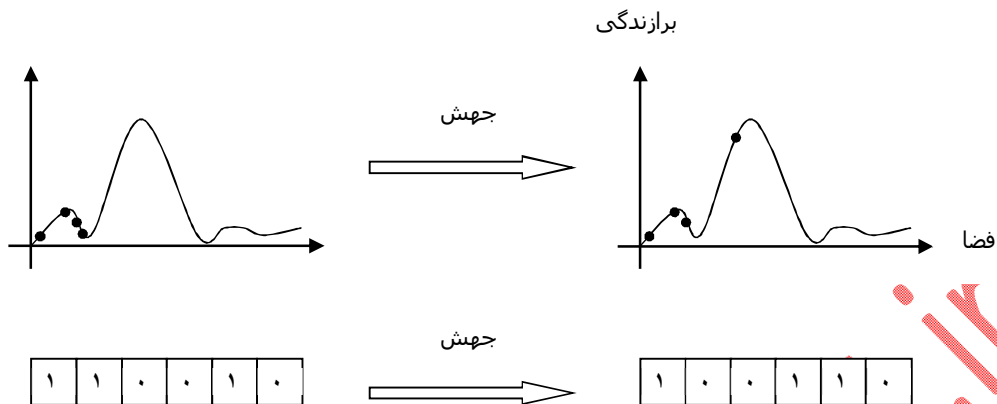
نسل قدیم برای نسل بعدی باقی بمانند. [13]

۲-۱۶- تحلیل مکانیزم جابجایی

به دلیل اهمیت مرحله ترکیب، بسیاری از تحقیقات درباره الگوریتم‌های ژنتیکی به روش‌های ترکیب و تحلیل آنها اختصاص یافته است. در این قسمت تعدادی از مهمترین این روش‌ها مورد بحث قرار می‌گیرد. معمولاً کاربرهای الگوریتم‌های ژنتیکی تعداد نقاط شکست کروموزوم در عمل ترکیب را یک یا دو نقطه انتخاب می‌کنند. اگر روش ترکیب دو نقطه‌ای را گسترش دهیم، ترکیب چند نقطه‌ای خواهیم داشت که هر رشته به صورت حلقه‌ای از بیت‌ها در نظر گرفته می‌شود که به K قسمت تعریف شده است؛ K تعداد نقاط شکست می‌باشد. در ترکیب چند نقطه‌ای، قسمت‌های دو کروموزوم یکی در میان باهم تعویض می‌شوند. بین اندازه جمعیت و نوع ترکیب ارتباط مستقیم وجود دارد؛ تجربیات نشان می‌دهد که ترکیب تک‌نقطه‌ای در جمعیت‌های کوچک بسیار مناسب است، اما برای جمعیت‌های بزرگتر ترکیب دو نقطه‌ای مناسبتر است. ترکیب‌های با تعداد نقاط شکست ثابت در جمعیت‌های کوچک امکان جستجوی بیشتری به ما می‌دهد. این واگرائی ذاتی در جمعیت‌های بزرگ لزوم جستجوی بیشتر را کاهش می‌دهد و بنابراین ترکیب دو نقطه‌ای جواب مناسب را به دست می‌آورد. [13]

۲-۱۷- جهش

در طبیعت برخی عوامل مانند تابش اشعه ماوراء بنفش باعث به وجود آمدن تغییرات غیرقابل پیش‌بینی در کروموزوم‌ها می‌شوند. از آنجایی که الگوریتم‌های ژنتیکی از قانون تکامل پیروی می‌کنند در این الگوریتم‌ها نیز عملگر جهش با احتمال کم اعمال می‌شود. جهش باعث جستجو در فضاهاى دست نخورده مسأله می‌شود می‌توان استنباط کرد که مهمترین وظیفه جهش اجتناب از همگرایی به بهینه محلی است. در آشکال زیر نحوه جهش و کارکرد آن نمایش داده شده است:



شکل ۲-۱۵- شبیه سازی جهش به کمک نمودار.

به تعبیری دیگر می توان جهش را مشابه شروع مجدد تصادفی الگوریتم «تپه نوردی»^{۵۹} هنگام گیر افتادن در فلات دانست.

در الگوریتم ژنتیک نیز بعد از اینکه یک عضو در جمعیت جدید به وجود آمد، هر ژن آن با احتمال جهش^{۶۰}، جهش می یابد. در جهش ممکن است ژنی از مجموعه ژن های جمعیت حذف شود یا ژنی که تا حال در جمعیت وجود نداشته است به آن اضافه شود. جهش یک ژن به معنای تغییر آن ژن است و وابسته به نوع کدگذاری، روش های متفاوت جهش استفاده می شود.

اگر مرحله جهش صورت نگیرد، فرزندان بلافاصله بعد از ترکیب و بدون هیچ تغییری بوجود می آیند (یا مستقیماً نسخه برداری می شوند که در نتیجه عمل ترکیب هم صورت نگرفته است). اگر تغییر صورت بگیرد، یک یا بیش از یک قسمت از کروموزوم تغییر می کند. اگر احتمال تغییر ۱۰۰٪ باشد، یعنی همه کروموزوم های تغییر کرده اند و اگر ۰٪ باشد، هیچ تغییر نکرده است.

به طور کلی جهش از قرار گرفتن $GA^{۶۱}$ در اکستریم های محلی جلوگیری می کند. جهش نباید زیاد صورت بگیرد زیرا در این صورت الگوریتم ژنتیک به جستوی کاملاً تصادفی تبدیل خواهد شد.

^{۵۹} Hill Climbing

^{۶۰} Presumption Mutation - P_m

^{۶۱} GA=الگوریتم ژنتیک.

همانطور که گفته شد، هر عضو، وابسته به احتمال جهش، جهش می‌یابد. احتمال جهش (P_m) مقداری است که توسط کاربر تعیین می‌شود. در الگوریتم استاندارد ژنتیک بنا به دلایلی مقدار این پارامتر، بسیار کوچک، مثل $P_m = 0.01$ یا حتی $P_m = 0.001$ در نظر گرفته می‌شود. اما از آنجا که کمتر از شکل استاندارد این الگوریتم استفاده می‌شود، به عنوان یک پیشنهاد می‌توان P_m را به صورت زیر تخمین زد:

$$P_m = \frac{1}{N}$$

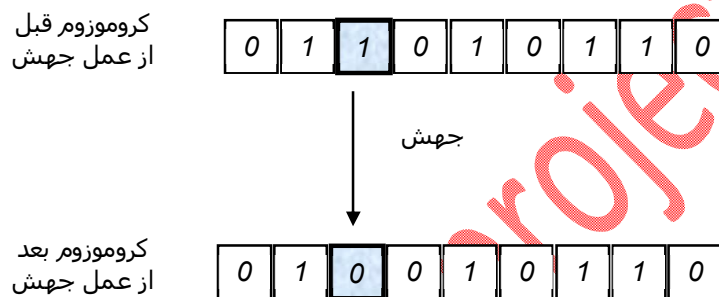
رابطه ۲-۱۸-احتمال جهش.

که N تعداد ژن‌های کروموزوم است. در هر حال P_m بین ۰ و ۱ است و معمولاً عددی کوچک انتخاب می‌شود در فرزندی که بوجود آمده است (توسط ترکیب)، به ترتیب مقداری تصادفی بین ۰ و ۱ به هر ژن اختصاص می‌یابد. اگر این مقدار اختصاص داده شده از P_m کمتر باشد، ژن جهش می‌یابد و اگر بیشتر باشد ژن تغییر نمی‌کند. نرخ بالای جهش، باعث تنوع و پراکندگی ژنتیکی در جمعیت می‌شود که این پراکندگی ممکن است همگرایی را به تأخیر بیندازد. به همین دلیل پیشنهاد می‌شود برای جمعیت‌های بزرگ یا در نسل‌های آخر از P_m های کوچکتر و برای جمعیت‌های کوچک یا در نسل‌های ابتدایی از P_m های بزرگتر استفاده شود. GA پارامترهای دیگری نیز دارد که از قرار گرفتن GA در بهینه‌محلی جلوگیری می‌کند، از پارامترهای مهم دیگر: تعداد جمعیت^{۶۲} و احتمال ترکیب می‌باشد. [13]

در ادامه چند روش جهت پیاده‌سازی جهش بیان می‌شود.

۲-۱۷-۱- جهش باینری^{۶۳}

در الگوریتم ژنتیک با کد گذاری باینری، این عملگر اغلب با تولید تصادفی یکی از اعداد ۰ و ۱ جایگزینی آن به جای بیت مورد نظر صورت می‌گیرد، اما در برخی کاربردهای ژنتیک عمل جهش دودویی در یک بیت با متمم ساختن آن بیت انجام می‌شود. به این صورت که اگر بیت مورد نظر ۰ بوده به بیت ۱ و بر عکس تبدیل خواهد شد، که آزمایش‌ها نشان داده شده است که روش دوم مناسبتر است.



شکل ۲-۱۶-۲- جهش باینری.

[13]

۲-۱۷-۲- جهش حقیقی^{۶۴}

در کدگذاری حقیقی، عملگر جهش باعث تولید تصادفی یک مقدار جدید در یک موقعیت خاص در کروموزوم می‌شود، در نتیجه این تغییرات تصادفی در جمعیت کروموزوم‌ها، نواحی بیشتری از فضای کاوش بررسی شده و از همگرایی بی‌موقع (ناگهانی محلی) الگوریتم جلوگیری می‌شود.

یک مثال از عملگر جهش حقیقی، جهش تصادفی یا یکنواخت می‌باشد. با این فرض که $C = (c_1, \dots, c_i, \dots, c_n)$ یک کروموزوم و C_i یک ژنی باشد که تحت عمل جهش قرار می‌گیرد آنگاه C_i یک مقدار انتخابی تصادفی جدید از محدوده C_i می‌باشد که بجای ژن C_i در کروموزوم جدید جایگذاری

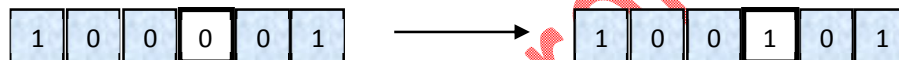
^{۶۳} Binary Mutation

^{۶۴} Real Mutation

خواهد شد. مثال دیگری برای این روش عملگر جهش مرزی که در آن یکی از ژن‌های کروموزوم به طور تصادفی با حد پایین یا بالای محدوده آن ژن جایگزین می‌شود $C_i = a_i$ یا $C_i = b_i$ [13].

۲-۱۷-۳- وارونه سازی بیت

از این نوع جهش هنگامی استفاده می‌شود که کد گذاری، کد گذاری باینری باشد. در اینجا بیتی که شرایط جهش را دارد اگر ۰ باشد به ۱ و اگر ۱ باشد به ۰ تغییر می‌دهد. به عنوان نمونه اگر در شکل ژن چهارم شرایط جهش را داشته باشد به صورت نشان داده شده، جهش می‌یابد.

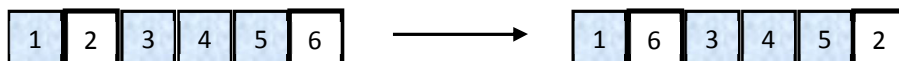


شکل ۲-۱۷-۳- جهش: وارونه سازی بیت.

[13]

۲-۱۷-۴- تغییر ترتیب قرارگیری

از این نوع جهش مخصوصاً در الگوریتم‌هایی استفاده می‌شود که کد گذاری بر اساس مقدار باشد البته در دیگر کد گذاری‌ها مثل کد گذاری باینری هم می‌توان این جهش را بکار برد. در این جهش، محل قرارگیری دو ژنی که می‌خواهد جهش بیابند در کروموزوم تعویض می‌شود. در شکل نمونه‌ای از این جهش نشان داده شده است.

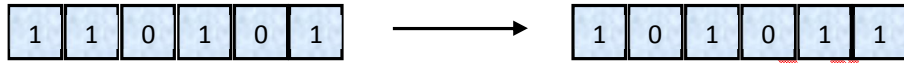


شکل ۲-۱۸-۴- جهش: تغییر ترتیب قرارگیری.

[13]

۲-۱۷-۵- وارون سازی

این عمل در طبیعت بسیار رخ می دهد ولی در الگوریتم های ژنتیکی به ندرت استفاده می شود و دلیل آن ایجاد تخریب زیاد است. این عملگر کروموزوم را معکوس می کند.



شکل ۲-۱۹- جهش: وارون سازی.

[13]

۲-۱۷-۶- تغییر مقدار

این نوع جهش را نمی توان برای کدگذاری باینری یا کدگذاری های مشابه که امکان تغییر ژن ها وجود ندارد، به کار برد. در این جهش به ژنی که شرایط جهش را دارد مقداری اضافه یا کم می شود. اضافه شدن و کم شدن می تواند به تصادف انتخاب شود یا الگوریتم مقید به استفاده از یکی از این دو عمل باشد. مقداری که به ژن افزوده یا از آن کاسته می شود، وابسته به محدوده مقدار ژن است و باز می تواند به تصادف انتخاب شود یا برای الگوریتم تعریف شود. بدیهی است مقدارهای بزرگ پراکندگی ژنتیکی را افزایش می دهند. در شکل --- نمونه ای از این جهش نشان داده شده است. این جهش خصوصاً برای کدگذاری هایی که در آنها، ژن ها به صورت اعداد حقیقی هستند مناسب است.

(1.29, 5.68, 2.86, 4.11, 5.55) → (1.29, 5.68, 2.73, 4.22, 5.55)

شکل ۲-۲۰- جهش: تغییر مقدار.

همانطور که در شکل دیده می شود، ژن های دوم و سوم کروموزوم جهش یافته اند. از شکل پیداست

که مقدار انتخابی برای افزودن یا کاستن می تواند از ژنی به ژن دیگر متفاوت باشد. [13]

۲-۱۸- محک اختتام اجرای الگوریتم ژنتیک

برای اینکه تشخیص دهیم چه موقع الگوریتم از اجرا متوقف شود، از شیوه‌های مختلفی می‌توان استفاده کرد. به عنوان نمونه می‌توان همگرا شدن کل جمعیت را در نظر گرفت و یا اینکه فاصله ارزیابی (برازندگی) بهترین فرد جمعیت از متوسط ارزیابی‌ها (برازندگی‌ها) را در نظر گرفت که در این حالت باید از حد مشخصی کوچکتر باشد، یا مقدار تابع هدف از حد مشخصی بیشتر باشد یا می‌توان تعداد نسل‌های مشخصی را به عنوان محک اختتام در نظر گرفت:

- ۱- به دست آوردن جواب نهایی مورد نظر بعد از چند تکرار کم و قابل قبول بودن جواب به ازای خطای خاص.
- ۲- اگر با پیشروی الگوریتم هیچ نوع بهبودی مشاهده نشد خواه الگوریتم جواب دلخواه را پیدا کرده باشد و یا اینکه در مینیمم محلی گیر کرده باشد.
- ۳- اگر مقدار میانگین تابع هدف به ازای تعدادی تکرار به مقدار خاصی رسیده باشد.
- ۴- الگوریتم به تعداد ثابتی از نسل‌ها رسیده باشد.
- ۵- بیشترین درجه برازش فرزندان حاصل شود یا دیگر نتایج بهتری حاصل نشود.
- ۶- بازرسی دستی.
- ۷- ترکیب‌های بالا.