دانشگاه سیستان و بلوچستان

تحصیلات تکمیلی

پایان‌نامه کارشناسی ارشد در ریاضی کاربردی (گرایش تحقیق در عملیات)

عنوان:

حل برنامه‌ریزی خطی فازی و برنامه‌ریزی خطی دوترازه با استفاده از الگوریتم ژنتیک

استاد راهنما:

دکتر حسن میش‌مست نهی

تحقیق و نگارش:

شهربانو شرافتمند

این پایان‌نامه از حمایت مالی معاونت پژوهشی دانشگاه سیستان و بلوچستان بهره‌مند شده است

بهمن ۱۳۹۱

تقدیم به:

پدر و مادر مهربانم

که هرچه دارم از برکت وجود آن هاست

و تقدیم به

خواهران فداکار و برادران عزیزم

و همه آنان که دوستشان دارم و دوستم دارند...

سپاس‌گزاری

بعد از حمد و ستایش خداوند منان که توفیق کسب علم و دانش را به من عطا نمود. لازم می‌دانم از کلیه کسانی که مرا در امر تحقیق و نگارش این پایان نامه حمایت کردند تقدیر و سپاسگزاری نمایم. در آغاز از استاد فرهیخته و گرانقدرم جناب آقای دکتر حسن میش‌مست نهی که با هدایتگری و راهنمایی‌های ارزنده‌شان سختی‌های این مسیر را بر من آسان نمودند قدردانی می‌نمایم و نیز از خانم دکتر فرانک حسین زاده سلجوقی و آقای دکتر مهدی الله دادی که زحمت داوری این پایان نامه را بر عهده گرفتند، کمال سپاسگزاری را دارم. همچنین از زحمات بی دریغ پدر و مادر عزیزم و خواهران و برادران فداکارم که همیشه مشوق من در امر تحصیل علم و دانش بوده اند تشکر و قدردانی می‌نمایم. در پایان از تمام دوستان خوبم به ویژه خانم ها زهره فیروزی، راضیه وحیدی نیا، فروغ خزیمه، زینب بهرام نژاد، مریم باقرنژاد، سیمین دلیر، زبیده آروم و خدیجه محمدنیا به خاطر یاری و مهربانی هایشان سپاسگزارم و از خداوند منان برای تمام این عزیزان آرزوی سعادت را دارم.

شهربانو شرافتمند

بهمن ۱۳۹۱

چکیده

برنامه‌ریزی خطی فازی یک برنامه‌ریزی با کاربرد بالا در مسائل بهینه‌سازی صنعتی می‌باشد، همچنین برنامه‌ریزی خطی دوترازه یک تکنیک برای مدل کردن تصمیم‌های غیر متمرکز است، لذا برای گرفتن بهترین تصمیم با روابط سلسه مراتبی در سازمان‌ها مناسب می‌باشد. به دلیل اینکه در موقعیت‌های واقعی ابعاد مسئله بزرگ می‌باشد، ممکن است حل این مسائل با روش‌های مرسوم مشکل باشد. الگوریتم‌های فراابتکاری به خصوص الگوریتم ژنتیک (GA) روشی کارا در حل مسائل بهینه‌سازی پیچیده می‌باشند. در این پایان‌نامه، نخست به بیان مفاهیمی از الگوریتم ژنتیک، مجموعه‌های فازی و غیرفازی سازها می‌پردازیم.

در ادامه الگوریتم ژنتیک را برای بدست آوردن مقادیر غیر فازی شده ضرایب فازی در برنامه‌ریزی خطی با محدودیت‌های فازی ($FCLP$) به کار می‌بریم. به این صورت که هر عدد فازی را با تقسیم کردن به نقاط افراز شبیه‌سازی کرده، سپس با استفاده از تابع برازندگی مرکز ثقل، مقادیر بهینه درجه عضویت را در نقاط افراز بدست می‌آوریم. این الگوریتم ژنتیک بدون استفاده از اصل گسترش، حساب بازه‌ها و ... برای محاسبات فازی و بدون استفاده از روش جریمه برای تخلف‌های محدودیت‌ها یک جواب بهینه بدست می‌آورد. در فصل سوم اصلاحی از این الگوریتم ژنتیک به منظور نزدیک‌تر شدن جواب به حل مستقیم پیشنهاد می‌شود. نتایج نشان داد که الگوریتم ژنتیک اصلاح شده جواب‌های مناسب تری ارائه می‌دهد.

در فصل چهارم، الگوریتم ژنتیکی برای حل $FCLP$ ارائه می‌دهیم. این الگوریتم با تولید مقادیر تصادفی در تکیه گاه عدد فازی، یک جواب بهینه برای $FCLP$ ارائه می‌دهد. در فصل نهایی به حل برنامه‌ریزی خطی دوترازه ($BLPP$) با استفاده از GA، الگوریتم بهینه‌سازی تراکم ذرات (PSO) و ترکیب GA و PSO ($HGAPSO$) می‌پردازیم. در این الگوریتم‌ها نه تنها جمعیت آغازین بلکه عملگرهای ژنتیک برای $BLPP$ شدنی می‌باشند. از اجرای این الگوریتم‌ها مشاهده شد $HGAPSO$ عملکرد بهتری دارد.

واژگان کلیدی: الگوریتم ژنتیک، مسأله برنامه‌ریزی خطی فازی، محدودیت فازی، مسأله برنامه‌ریزی خطی دوترازه، الگوریتم بهینه‌سازی تراکم ذرات.

فهرست مطالب

فهرست شکل‌ها ی

فهرست جدول‌ها ل

۱ مقدمات و مفاهیم اولیه ۱

۱-۱ تاریخچه الگوریتم ژنتیک ۲

۱-۲ ساختار الگوریتم ژنتیک ۳

۱-۳ روشهای جستجو ۵

۱-۳-۱ جستجوی تصادفی ۵

۱-۳-۲ روش گرادیان ۶

۱-۳-۳ جستجوی تسلسلی ۶

۱-۴ الگوریتم ژنتیک در مقایسه با سایر روش های جستجو ۶

۱-۵ اصطلاحات الگوریتم ژنتیک و تناظر آنها با سیستم طبیعی ۷

۱-۶ اجزاء تشکیل دهنده ساختار الگوریتم ژنتیک ۸

۱-۶-۱ رمزگشایی ۸

۱-۶-۲ جمعیت ۸

۱-۶-۳ تابع برازندگی ۹

۱-۶-۴ درجه بندی تابع برازندگی ۱۰

۱-۶-۵ انتخاب ۱۱

۱-۶-۶ عملگر ترکیب ۱۳

۱-۶-۷ عملگر جهش ۱۵

۱-۷ معیار همگرایی ۱۶

۱-۸ مزیت‌های استفاده از الگوریتم ژنتیک ۱۷

۱-۹ نواقص الگوریتم ژنتیک ۱۸
۱-۱۰ مجموعه‌های فازی ۱۸
۱-۱۰-۱ مفاهیم فازی ۱۹
۱-۱۰-۲ نمادگذاری مجموعه‌های فازی ۱۹
۱-۱۰-۳ تعاریف مقدماتی در مجموعه‌های فازی ۲۰
۱-۱۰-۴ اعداد فازی ۲۱
۱-۱۰-۵ عدد فازی $L-R$ ۲۲
۱-۱۰-۶ عدد فازی مثلثی ۲۲
۱-۱۱ غیر فازی ساز ۲۳
۱-۱۱-۱غیرفازی ساز ماکزیمم ۲۳
۱-۱۱-۲غیرفازی ساز مرکز ثقل ۲۵

۲ یک الگوریتم ژنتیک برای حل برنامه ریزی خطی با محدودیت های فازی ۲۷
۲-۱ مقدمه ۲۸
۲-۲ مسئله برنامه ریزی خطی فازی ۲۸
۲-۲-۱ حل $FCLP$ بر مبنای اعداد فازی ۲۹
۲-۳ یک الگوریتم ژنتیک برای حل $FCLP$ ۳۳
۲-۴ نتایج عددی ۳۶

۳ الگوریتم ژنتیک اصلاح شده لین ۴۴
۳-۱ مقدمه ۴۵
۳-۲ روش الگوریتم ژنتیک اصلاح شده لین ۴۵
۳-۲-۱ الگوریتم ژنتیک اصلاح شده۱ . . . . . ۴۸
۳-۲-۲ الگوریتم ژنتیک اصلاح شده۲ . . . . . ۴۹
۳-۳ نتایج عددی ۵۰

۴ حل برنامه‌ریزی خطی با محدودیت‌های فازی بوسیله یک روش الگوریتم ژنتیک تصادفی ۶۷
۴-۱ مقدمه ۶۸
۴-۲ دستگاه نامعادلات خطی با ضرایب فازی ۶۸
۴-۳ مسئله برنامه‌ریزی خطی با محدودیت‌های فازی ۶۹

۴-۴ یک روش الگوریتم ژنتیک برای حل برنامه‌ریزی خطی با محدودیت‌های فازی ۷۰
۴-۴-۱ الگوریتم ژنتیک پیشنهادی ۷۰
۴-۵ نتایج عددی ۷۳

۵ حل برنامه‌ریزی خطی دوترازه با استفاده از الگوریتم ژنتیک، الگوریتم PSO و ترکیب آن‌ها ۷۵
۵-۱ مقدمه ۷۶
۵-۲ مسأله برنامه‌ریز خطی دوترازه ۷۶
۵-۲-۱ روش kام-بهترین ۷۸
۵-۳ الگوریتم ژنتیک برای حل برنامه‌ریزی دوترازه ۸۲
۵-۴ حل برنامه‌ریزی خطی دوترازه بوسیله الگوریتم بهینه‌سازی تراکم ذرات ۸۴
۵-۴-۱ الگوریتم بهینه‌سازی تراکم ذرات PSO ۸۵
۵-۴-۲ الگوریتم PSO برای حل $BLPP$ ۸۶
۵-۵ حل برنامه‌ریزی خطی دوترازه بوسیله ترکیب الگوریتم ژنتیک و بهینه‌سازی تراکم ذرات . . ۸۷
۵-۵-۱ ترکیب GA و PSO $(HGAPSO)$ ۸۸
۵-۶ نتایج عددی ۹۰
۵-۷ حل مدل زنجیره تأمین دوترازه با استفاده از $HGAPSO$ ۹۹
۵-۷-۱ مدیریت زنجیره تأمین و مدیریت توزیع ۹۹
۵-۷-۲ تشریح مدل زنجیره تأمین ۹۹

نتیجه‌گیری و پیشنهادات ۱۰۳

مراجع ۱۰۵

آ کد الگوریتم‌ها ۱۰۷
آ-۱ کد الگوریتم ژنتیک اصلاح شده۱ . . . . . ۱۰۷
آ-۲ کد الگوریتم کلی ۱۰۹
آ-۳ کد الگوریتم ژنتیک ونگ برای حل $BLPP$ ۱۱۰
آ-۴ کد الگوریتم PSO کیو و هوانگ برای حل $BLPP$ ۱۱۲
آ-۵ کد الگوریتم $HGAPSO$ کیو وهان برای حل $BLPP$ ۱۱۴

واژه‌نامه انگلیسی به فارسی ۱۱۷

فهرست شکل‌ها

۱-۱ یک نمایش برای غیرفازی ساز ماکزیمم ۲۴
۱-۲ یک نمایش برای غیرفازی ساز مرکز ثقل ۲۵

۲-۱ یک $\alpha-$برش عدد فازی $\tilde{A}$ ۳۰
۲-۲ ناحیه ی شدنی مثال ۲-۱ در صفحه دکارتی ۳۷
۲-۳ نمودار برازندگی برای عدد فازی ۱۲ . . . . . ۳۸
۲-۴ نمودار برازندگی برای عدد فازی ۲۵ . . . . . ۳۸
۲-۵ نمودار برازندگی برای عدد فازی ۴۶۰۰۰ . . . . . ۳۹
۲-۶ نمودار برازندگی برای عدد فازی ۲۰ . . . . . ۳۹
۲-۷ نمودار برازندگی برای عدد فازی ۵ . . . . . ۳۹
۲-۸ نمودار برازندگی برای عدد فازی ۲۵۰۰۰ . . . . . ۴۰
۲-۹ نمودار برازندگی برای عدد فازی ۲۲ . . . . . ۴۰
۲-۱۰ نمودار برازندگی برای عدد فازی ۱۲ . . . . . ۴۰
۲-۱۱ نمودار برازندگی برای عدد فازی ۳۰۰۰۰ . . . . . ۴۱

۳-۱ افراز بازه عدد فازی $\tilde{W}$ ۴۶
۳-۲ یک $\alpha-$برش عدد فازی $\tilde{W}$ ۴۷
۳-۳ نمایش مراحل الگوریتم ژنتیک اصلاح شده ۵۱
۳-۴ نمودار همگرایی برازندگی برای عدد فازی ۱۲ در حالت بیشینه‌سازی برازندگی ۵۳
۳-۵ نمودار همگرایی برازندگی برای عدد فازی ۱۲ در حالت کمینه‌سازی برازندگی ۵۳
۳-۶ نمودار همگرایی برازندگی برای عدد فازی ۵ در حالت بیشینه‌سازی برازندگی ۵۴
۳-۷ نمودار همگرایی برازندگی برای عدد فازی ۵ در حالت کمینه‌سازی برازندگی ۵۴
۳-۸ نمودار همگرایی برازندگی برای عدد فازی ۲۵۰۰۰ در حالت بیشینه‌سازی برازندگی . . ۵۵

۳-۹ نمودار همگرایی برازندگی برای عدد فازی ۲۵۰۰۰ در حالت کمینه‌سازی برازندگی . . . ۵۵
۳-۱۰ نمودار همگرایی برازندگی برای عدد فازی ۱۲ . . . . . ۵۷
۳-۱۱ نمودارهمگرایی برازندگی برای عدد فازی ۵ . . . . . ۵۷
۳-۱۲ نمودار همگرایی برازندگی برای عدد فازی ۱۲ . . . . . ۵۸
۳-۱۳ نمودار همگرایی برازندگی برای عدد فازی ۵ . . . . . ۵۹
۳-۱۴ نمودار همگرایی برازندگی برای عدد فازی ۲۵ . . . . . ۶۱
۳-۱۵ نمودار همگرایی برازندگی برای عدد فازی ۵ . . . . . ۶۱
۳-۱۶ نمودار همگرایی برازندگی برای عدد فازی ۲۲ . . . . . ۶۲
۳-۱۷ نمودار همگرایی برازندگی برای عدد فازی ۲ . . . . . ۶۵
۳-۱۸ نمودار همگرایی برازندگی برای عدد فازی ۱۵ . . . . . ۶۵

۴-۱ نمودار برازندگی بر حسب نسل برای اجرا ۱ . . . . . ۷۴
۴-۲ نمودار برازندگی بر حسب نسل برای اجرا ۲ . . . . . ۷۴

۵-۱ ناحیه شدنی مثال ۵-۷ . . . . . ۸۱
۵-۲ نمودارهمگرایی GA برای مثال ۵-۸ . . . . . ۹۲
۵-۳ نمودارهمگرایی PSO برای مثال ۵-۸ . . . . . ۹۲
۵-۴ نمودارهمگرایی $HGAPSO$ برای مثال ۵-۸ . . . . . ۹۲
۵-۵ نمودارهمگرایی GA برای مثال ۵-۹ . . . . . ۹۴
۵-۶ نمودارهمگرایی PSO برای مثال ۵-۹ . . . . . ۹۴
۵-۷ نمودارهمگرایی $HGAPSO$ برای مثال ۵-۹ . . . . . ۹۴
۵-۸ نمودارهمگرایی GA برای مثال ۵-۱۰ . . . . . ۹۶
۵-۹ نمودارهمگرایی PSO برای مثال ۵-۱۰ . . . . . ۹۶
۵-۱۰ نمودارهمگرایی $HGAPSO$ برای مثال ۵-۱۰ . . . . . ۹۶
۵-۱۱ نمودارهمگرایی GA برای مثال ۵-۱۱ . . . . . ۹۸
۵-۱۲ نمودارهمگرایی PSO برای مثال ۵-۱۱ . . . . . ۹۸
۵-۱۳ نمودارهمگرایی $HGAPSO$ برای مثال ۵-۱۱ . . . . . ۹۸

فهرست جدول‌ها

۱-۱ مقایسه غیرفازی سازهای ماکزیمم و مرکز ثقل ۲۶

۱-۲ پهنه‌های چپ و راست برای ضرایب محدودیت‌ها در حالت ۱ . . . . . ۳۸
۲-۲ مقادیر غیرفازی شده ضرایب در حالت ۱ . . . . . ۳۸
۳-۲ پهنه‌های چپ و راست برای ضرایب محدودیت‌ها در حالت ۲ . . . . . ۴۱
۴-۲ مقادیر غیرفازی شده ضرایب در حالت ۲ . . . . . ۴۲
۵-۲ پهنه‌های چپ و راست برای ضرایب محدودیت‌ها در حالت ۳ . . . . . ۴۲
۶-۲ مقادیر غیرفازی شده ضرایب در حالت ۳ . . . . . ۴۲

۱-۳ $\mu_i{}^*,\ i = ۰, ۱, ..., ۲t$ بدست آمده در تکرار Kام ۵۲
۲-۳ مقادیر غیرفازی شده ضرایب در حالت بیشینه‌سازی برازندگی ۵۶
۳-۳ مقادیر غیرفازی شده ضرایب در حالت کمینه‌سازی برازندگی ۵۶
۴-۳ مقادیر غیرفازی شده ضرایب برای حالت ۲ مثال۲-۱ . . . . . ۵۶
۵-۳ مقادیر غیرفازی شده ضرایب برای حالت ۳ مثال۲-۱ . . . . . ۵۸
۶-۳ مقادیر غیرفازی شده ضرایب ۵۹
۷-۳ $\mu_i{}^*,\ i = ۰, ۱, ..., ۲t$ بدست آمده در تکرار Kام ۶۰
۸-۳ مقایسه نتایج ۶۳
۹-۳ پهنه های چپ و راست برای ضرایب محدودیت ها ۶۴
۱۰-۳ مقادیر غیرفازی شده ضرایب ۶۵

۱-۴ نتایج بدست آمده از اجرای الگوریتم پیشنهادی ۷۳
۲-۴ کروموزوم با بیشترین مقدار برازندگی در نسل Kام در اجرای اول ۷۳

۱-۵ پارامترهای اولیه ۹۱

۵-۲ نتایج مثال ۵-۸ . . . . . ۹۱
۵-۳ نتایج مثال ۵-۹ . . . . . ۹۳
۵-۴ نتایج مثال ۵-۱۰ . . . . . ۹۵
۵-۵ نتایج مثال ۵-۱۱ . . . . . ۹۷
۵-۶ تشریح محدودیت‌ها ۱۰۱
۵-۷ پارامترهای اولیه مدل ۱۰۱
۵-۸ پارامترهای اولیه ۱۰۲
۵-۹ نتایج مثال زنجیره تأمین ۱۰۲

فصل ۱
مقدمات و مفاهیم اولیه

در این فصل مفاهیم اولیه‌ای از الگوریتم ژنتیک، مجموعه‌های فازی و غیرفازی ساز می‌آوریم.

۱-۱ تاریخچه الگوریتم ژنتیک

الگوریتم‌های ژنتیک [۱] با اقتباس از فرآیندهای مشاهده شده در تکامل طبیعی، اولین بار توسط جان هلند [۲] در دهه ۷۰ پیشنهاد شد و به طور کلی متعلق به دسته بالاتری تحت عنوان الگوریتم‌های تکاملی هستند. این الگوریتم بر اساس اصل «حیات مناسب‌ترین» نظریه داروین عمل می‌کند. به عبارت دیگر مناسب‌ترین کروموزوم‌ها از طریق جابجایی تصادفی ژن‌های نسل قبل به وجود می‌آیند و به حیات خود ادامه می‌دهند. به این ترتیب الگوریتم ژنتیک با شبیه سازی فرایندهای ژنتیک طبیعی سعی می‌کند مسائل پیچیده دنیای واقعی را که با روش‌های بهینه سازی مرسوم قابل حل نیستند را حل نماید. امروزه الگوریتم ژنتیک جای خود را در میان روش های بهینه سازی مسائل پیچیده به خوبی باز کرده است. بسیاری از مسائل بهینه سازی در زمینه مهندسی به ویژه سیستم‌های کارخانجات، بطور ذاتی خیلی مشکل هستند. اگر بخواهند توسط روش های معمولی حل شوند، پیچیده می‌شوند. از سال ۱۹۹۰ رشد محسوسی در علایق برای تقلید از زندگی طبیعی به وجود آمده است. شبیه سازی فرآیندهای تکاملی توسط بشر به تکنیک‌های بهینه سازی احتمالی منجر شده که الگوریتم‌های تکاملی نامیده می‌شوند که اغلب می‌توانند به شیوه ای متفاوت از تکنیک‌های بهینه سازی معمولی عمل کنند. سه شاخه از این تحقیقات وجود دارد: الگوریتم ژنتیک، برنامه ریزی تکاملی، استراتژی تکاملی. از میان آن‌ها الگوریتم ژنتیک شاید معروف‌ترین باشد. اخیراً الگوریتم ژنتیک به عنوان یک تکنیک بهینه‌سازی برای مسائل پیچیده توجه زیادی را به خود جلب کرده است و به طور موفقیت آمیزی در ناحیه مهندسی صنعتی به کار برده می‌شود. در میان روش‌های بهینه‌سازی الهام گرفته از طبیعت جاندار، الگوریتم ژنتیک از همه تکامل یافته تر می‌باشد. الگوریتم ژنتیک بر اساس اصول تکامل پایه ریزی شده است. در طبیعت افرادی که در رقابت برای دستیابی به منابع محدودی مانند غذا و سرپناه پیروز می‌شوند باقی می‌مانند و تولید مثل می‌کنند. برتری این افراد مدیون ویژگی‌های فردی آن‌هاست که تا حد زیادی تحت تأثیر ژن‌های آنها قرار دارد. تولید مثل افراد پیروز موجب تکثیر ژن‌ها و در نتیجه فرزندانی بهتر می‌گردد. با انجام متوالی انتخاب بهترین افراد جمعیت و تولید مثل آنها کل جمعیت به سود سازش بیشتر با محیط خود یعنی دستیابی به منابع بهتر و بیشتر سوق می‌دهد. اصول اولیه الگوریتم ژنتیک توسط جان هلند در سال ۱۹۶۲ در دانشگاه میشیگان ایالات متحده به وضوح بیان گردید. پیشرفت و شهرت این الگوریتم نتیجه تلاش شاگردان وی از جمله داویس [۳] و خصوصاً گلدبرگ [۴] می‌باشد. آنها در تحقیقات خود دو هدف اصلی زیر را دنبال می‌کردند:

[۱] Genetic Algorithms
[۲] John Holand
[۳] Davis
[۴] Goldberg

۱. ارائه شرح دقیق و خلاصه ای از عملکرد قابل قبولی از سامانه‌های طبیعی
۲. طراحی نرم افزارهای سامانه‌های ساختگی که ساز و کارهای اصلی سامانه‌های طبیعی را در برداشته باشد.

پس از این گزارش مقدماتی، تحقیق در جهت توسعه چارچوب ریاضی این الگوریتم و به طور همزمان در زمینه کاربردهای آن ادامه یافت. در نهایت این تحقیقات منجر به انتشار کتاب سازش در سامانه های طبیعی و مصنوعی در سال ۱۹۷۵ گردید که در حال حاضر مرجع بسیاری از منابع در این مبحث به شمار می‌رود[۳۱].

۱-۲ ساختار الگوریتم ژنتیک

شکل معمول الگوریتم ژنتیک توسط گلدبرگ معرفی شده است [۱۲]. الگوریتم ژنتیک یک تکنیک جستجوی احتمالی بر اساس مکانیزم انتخاب طبیعی در ژنتیک طبیعی است. الگوریتم ژنتیک با تکنیک‌های جستجوی مرسوم تفاوت دارد. با یک مجموعه اولیه از جواب های تصادفی شروع می‌کند که جمعیت اولیه نام دارد. هر عضو در جمعیت کروموزوم نام دارد، که یک جواب را برای مسئله موجود ارائه می‌کند. یک کروموزوم یک رشته از نمادها است که معمولاً و نه لزوماً به صورت یک رشته دودویی می‌باشد طی هر بار تکرار الگوریتم ژنتیک، مجموعه‌ای جدید از کروموزوم‌ها تشکیل می شود. جمعیت ایجاد شده در هر زمان نسل نامیده می‌شود. طی هر نسل، قابلیت کروموزوم‌ها توسط تابع برازندگی که با توجه به تابع هدف تعیین می‌شود، برآورده می‌شود.

کروموزوم‌ها از طریق تکرارهای پی در پی که زاد و ولد نام دارد تکامل می‌یابند و در حین هر زاد و ولد از طریق مقدار برازندگیشان ارزیابی می‌شوند. برای تولید نسل بعدی کروموزوم جدید (فرزند) به یکی از صورت‌های زیر تولید می‌شود:

۱) ترکیب دو کروموزوم از نسل حاضر با استفاده از عملگر ترکیب
۲) اصلاح کردن یک کروموزوم با استفاده از عملگر جهش

نسل جدید با توجه به مقادیر برازندگی کروموزوم‌ها (از بین والدین و فرزندان) تشکیل می‌گردد. کروموزوم‌های سازگارتر شانس بیشتری برای انتخاب شدن دارند. بعد از چندین نسل الگوریتم به یکی از بهترین جواب ها همگرا می‌شود که امیدوارانه جواب بهین یا زیر بهینی از مسئله است. فرض کنید $p(t)$ و$c(t)$ والدین و فرزندان در نسل tام باشند. ساختار کلی الگوریتم ژنتیک به صورت زیر است:

۱- شروع
۲- قرار بده $t = ۰$
۳- جمعیت اولیه $p(t)$ را تولید کن.
۴- $p(t)$ را ارزیابی کن.

۵- تا زمانی که شرط خاتمه برقرار نیست فرزندان $c(t)$ را در ترکیب والدین $p(t)$ تولید کن.

۶- $c(t)$ را ارزیابی کن.

۷- نسل جدید $p(t+1)$ را از $p(t)$ و $c(t)$ تولید کن.

۸- قرار بده $t = t + 1$ و برو به ۴

۹- پایان

این الگوریتم یک نسخه اصلاح یافته گریفنستت [۵] و بکر [۶] است.

برای تولید نسل جدید از عملگرهای ژنتیک استفاده می‌کنیم که عبارتند از:

۱) ترکیب و جهش

۲) انتخاب

عملگر ژنتیک فرآیند ارثی ژن‌ها را برای خلق فرزند در هر نسل تقلید می‌کند. عملگر تکامل از فرآیند تکاملی داروین برای خلق جمعیتی از یک نسل به نسل بعد تقلید می‌کند.

ترکیب عملگر اصلی ژنتیک است این عملگر بر روی دو کروموزوم برای تولید فرزند در یک زمان عمل می‌کند. یک راه ساده برای به دست آوردن عمل ترکیب می‌تواند انتخاب یک نقطه برش تصادفی و جا به جا کردن سمت چپ آنها بر اساس نقطه انتخاب شده با هم است. این روش با نمایش کروموزوم‌ها به صورت رشته بیتی خوب کار می‌کند. نرخ ترکیب (P_c) به عنوان نسبت تعداد فرزندان در هر نسل به اندازه جمعیت تعریف می‌شود. این نسبت امید ریاضی تعداد کروموزم‌ها را برای استفاده از عملگر ترکیب مشخص می‌کند. یک نرخ ترکیب بالا اجازه اکتشاف بیشتر فضای جواب را می‌دهد و احتمال افتادن در بهینه های محلی را می‌کاهد. ولی اگر نرخ ترکیب خیلی بالا باشد منجر به اتلاف زمان محاسباتی زیادی در کشف فضای جواب می‌شود. جهش یک عملگر پشتیبان است که تغییرات تصادفی در کروموزم های متفاوت انجام می‌دهد یک راه ساده برای به دست آوردن جهش می‌تواند تغییر دادن یک یا چند ژن باشد. در الگوریتم ژنتیک جهش نقش اساسی دارد:

۱)جایگزین کردن ژنهای از دست رفته از جمعیت در طول فرآیند انتخاب به طوری که آنها می‌توانند در جمعیت جدید آزمایش شوند.

۲) فراهم کردن ژن‌هایی که در جمعیت اولیه نبودند.

نرخ جهش (P_m) به صورت درصدی از کل ژن‌های جمعیت تعریف می‌شود. نرخ جهش میزان ژن‌های جدیدی را که به جمعیت وارد می‌شوند کنترل می‌کند. اگر نرخ جهش خیلی پایین باشد بسیاری از ژن‌هایی که می‌توانند مفید باشند هرگز آزمایش نمی‌شوند و اگر نرخ جهش خیلی بالا باشد، آشفتگی تصادفی بسیاری

[۵] Grefenstette
[۶] Baker

به وجود می‌آید و فرزندان شباهت کمتری به والدین خواهند داشت. همچنین الگوریتم توانایی آموختن از جستجوهای قبلی را از دست خواهد داد. عملگر انتخاب بهترین کروموزم ها را از یک نسل به نسل دیگر منتقل می‌کند. در واقع کروموزوم‌های با مقدار تابع صلاحیت بالا را، برای نسل جدید انتخاب می‌کند[۱۱].

۱-۳ روشهای جستجو

جستجو یکی از روش‌های رایج حل برای مسائلی است که شخص نمی‌تواند یک دنباله از مراحل که منجر به جواب می‌شود را مشخص کند. جستجو می‌تواند به دو طریق انجام شود. استراتژی اول از اطلاعات در حوزه مسئله استفاده نمی‌کند، دومین استراتژی از اطلاعات اضافی برای راهنمایی جستجو همراه با بهترین جهت جستجو استفاده می‌کند. دو نکته مهم در استراتژی‌های جستجو وجود دارد:

۱) بهره برداری از بهترین جواب ها

۲) کشف کردن فضای جواب

مقایسه جستجوی تصادفی و الگوریتم ژنتیک نشان می‌دهد که جستجوی تصادفی یک مثال از استراتژی است که فضای جستجو را کشف می‌کند در حالی که از بهره برداری از فضای جواب صرف نظر می‌کند. الگوریتم ژنتیک یک دسته از روش‌های جستجو است که عناصر جهت و جستجوی تصادفی را که می‌تواند یک توازن قابل توجه بین کشف و بهره برداری از فضای جستجو برقرار کند، ترکیب می‌کند. معمولاً الگوریتم برای حل کردن مسائل بهینه سازی یک دنباله از مراحل محاسباتی است که به طور مجانبی به جواب بهینه همگراست. مهمترین دسته از روش های بهینه سازی دنباله ای از محاسبات روی گرادیان یا مشتق مراحل بالا از تابع هدف تولید می‌کنند. این روش ها اکثراً برای نقاط تنها به کار می‌روند و جستجو به صورت نقطه به نقطه انجام می‌گیرد. این نگرش نقطه به نقطه، خطر افتادن در بهینه‌های محلی را دارد. الگوریتم ژنتیک یک جستجوی چند جهته را به وسیله حفظ کردن جمعیتی از جواب ها در هر مرحله انجام می‌دهد. نگرش جمعیت به جمعیت برای ساختن یک جمعیت به دور از بهینه محلی تلاش می‌کند. در هر زاد و ولد جواب‌های نسبتاً خوب دوباره تولید می‌شوند در حالی که جواب‌های نسبتاً بد می‌میرند. الگوریتم ژنتیک به منظور راهنمایی جستجو به سمت ناحیه ای از فضای جستجو که احتمال بهبود بهتری وجود دارد از قوانین انتقال احتمالی برای انتخاب جوابی که باید تولید شود و جوابی که باید بمیرد استفاده می‌کند. روش‌های مختلفی به منظور جستجو و بهینه سازی ابداع شده است. از بین روش‌های مختلف جستجو به شرح سه روش زیر بسنده می‌کنیم [۴] :

۱-۳-۱ جستجوی تصادفی

جستجوی تصادفی یک روش غیرهوشمندانه جستجو است که برای توابع مختلف قابل استفاده می‌باشد. در این روش نقاط در فضای جستجو به صورت تصادفی و یا یک روش منظم انتخاب و میزان انطباق آنها

ارزیابی می‌شود. به دلیل ویژگی غیرهوشمندانه بودن، این روش معمولاً به تنهایی به کار گرفته نمی‌شود.

۱-۳-۲ روش گرادیان

روش گرادیان برای اعمال روی توابع پیوسته خوش رفتار به کار گرفته می‌شود. این روش‌ها از اطلاعات مربوط به گرادیان تابع برای هدایت مسیر جستجو استفاده می‌کنند. این روش به دو صورت مستقیم و غیرمستقیم می‌باشد. در روش مستقیم با استفاده از تابع هدف و حرکت به سوی نقطه بهینه (بر اساس گرادیان) جستجو صورت می‌پذیرد. در روش غیرمستقیم نقطه بهینه محلی را با حل دستگاه معادلات غیرخطی به دست می‌آورند. این روش به دلایل زیر دارای ضعف می‌باشد:

الف) این روش روی توابعی که صرفاً یک ماکزیمم نسبی (مینیمم نسبی) دارند به خوبی اجرا می‌شود، ولی در مورد مسائلی که چندین ماکزیمم نسبی (مینیمم نسبی) دارند ممکن است به یک جواب نه چندان قابل قبول منجر گردد.

ب) این روش بر اساس وجود مشتق های جزئی پایه گذاری شده است، لذا اگر به هر دلیل نتوان مشتق تابع را به دست آورد (مثلاً در مورد توابع ناپیوسته) استفاده از این روش منتفی است.

۱-۳-۳ جستجوی تسلسلی

با ترکیب روش‌های جستجوی تصادفی و گرادیان، روش جستجوی جدیدی تحت عنوان جستجوی تسلسلی ایجاد شده است. بدین صورت الگوریتم هرگاه به یک نقطه بهینه موضعی رسید، جستجو را از یک نقطه دیگر که به تصادف انتخاب شده است، شروع نماید. این روش در مواردی که فضای جواب محدود و تعداد عناصر آن کم است، قابل استفاده است. بنابراین کارایی آن ضعیف است.

۱-۴ الگوریتم ژنتیک در مقایسه با سایر روش های جستجو

الگوریتم ژنتیک همانند سایر روش‌هایی که در قسمت قبل بیان شد با یک تابع برازندگی که بر حسب مورد بیشینه‌سازی یا کمینه‌سازی می باشد سر و کار دارد. در الگوریتم ژنتیک لازم است که مجموعه پارامترهای طبیعی مسئله بهینه سازی به صورت یک رشته با طول محدود و با استفاده از حروفی معین و مشخص کد می‌شوند. به عنوان مثال در بیشینه‌سازی تابع $y = x^۲$ در فاصله $[۰, ۳۱]$ ، در روش‌های معمولی ابتدا مقدار x را به تدریج افزایش می دهند تا به مقدار حداکثر تابع در این بازه دست یابند. ولی در الگوریتم ژنتیک پس از کد کردن مجموعه پارامترها، تشابه بین کدها بررسی می‌شود و لذا محدودیت‌هایی از قبیل: عدم پیوستگی تابع هدف، نداشتن مشتق و ... از بین می‌روند.

در اکثر روش‌های بهینه سازی از یک نقطه در فضای جواب و با استفاده از تعدادی قواعد انتقال، به یک نقطه بعدی حرکت صورت می‌گیرد. این روش خطرناک است زیرا وجود ماکزیمم‌های محلی باعث بروز خطا می‌شود. در مقابل، الگوریتم‌های ژنتیک از یک پایگاه اطلاعاتی قوی که در برگیرنده نقاط بسیاری است استفاده می‌نمایند و آنها را به طور همزمان مورد بررسی قرار داده و حرکت به سوی بهترین نقطه را به طور موازی انجام می‌دهند. لذا احتمال وقوع خطا کاهش می‌یابد. به طور کلی می‌توان مزایای الگوریتم ژنتیک را نسبت به دیگر روش های جستجو در چهار مورد زیر بیان نمود:

۱) الگوریتم ژنتیک با کدهای مجموعه جواب ها سر و کار دارد نه با خود جواب‌ها

۲) الگوریتم به دنبال یک جامعه جواب می‌گردد نه یک جواب تنها

۳) الگوریتم از اطلاعات ارزیابی شده استفاده می‌کند نه مشتق و دیگر دانش‌های معین

۴) الگوریتم ژنتیک از قوانین احتمالات جهت انتقال از یک مرحله به مرحله‌ی دیگر استفاده می‌کند نه از قوانین ثابت.

۱-۵ اصطلاحات الگوریتم ژنتیک و تناظر آنها با سیستم طبیعی

برای فهم مناسب‌تر و بهتر تحقیقات انجام شده و همچنین مطالبی که در خصوص الگوریتم ژنتیک منتشر شده‌اند، لازم است که با این اصطلاحات تا حد نیاز آشنایی پیدا کنیم. به دلیل اینکه الگوریتم ژنتیک از ژنتیک طبیعی و علم کامپیوتر برای رسیدن به اهداف بهینه سازی در سیستم های پیچیده سود می‌برد لذا در این قسمت مروری به اصطلاحات رایج در آن و تناظر آن با همتای طبیعی شان انجام خواهد شد.

به طور کلی، رشته‌ها در الگوریتم‌های ژنتیک متناظر با کروموزوم‌ها در سیستم‌های بیولوژیکی هستند. در طبیعت یک یا چندین کروموزوم برای تشکیل یک ژنتیک کلی و توصیف یک ارگانیسم با یکدیگر ترکیب می‌گردند. به این ژنتیک کلی، ژنوتایپ می‌گویند. متناظر ژنوتایپ [۷] در سیستم های مصنوعی ساختار رشته‌ها است. در سیستم‌های طبیعی در نتیجه تقابل ژنوتایپ بامحیط، ارگانیسم رفتاری یا ظاهری از خود بروز می دهد که به آن فنوتایپ [۸]می‌گویند. متناظر آن در الگوریتم های ژنتیک رمزخوانی ساختارها به صورت مجموعه پارامترها، آلترناتیوهای جواب و نقاط فضای جواب است. در طبیعت، کروموزوم از ژن ها تشکیل شده است که هر کدام دارای مقادیری هستند به نام آلل [۹] در ژنتیک طبیعی محل هر ژن لوکاس نام دارد و به صورت جدایی از ژن‌ها تعریف می‌شود. یعنی تابعی از ژن نیست. به عنوان مثال ژن رنگ چشم، محل آن در موقعیت شماره ۱۰ و مقدار آن (فنوتایپ) آبی است.

در سیستم های طبیعی (الگوریتم های ژنتیک) رشته ها ترکیبی از ویژگی ها هستند که هر یک مقادیر خاص

[۷] Genotype
[۸] Phenotype
[۹] Allel

خود را دارند.

۱-۶ اجزاء تشکیل دهنده ساختار الگوریتم ژنتیک

در این قسمت به معرفی اجزاء اصلی تشکیل دهنده الگوریتم ژنتیک می‌پردازیم[۳۱]:

۱-۶-۱ رمزگشایی

متغیرهای طراحی را متغیرهای گسسته و پیوسته (شناور) و یا ترکیبی از این دو تشکیل می‌دهند. به این دلیل ماهیت هر یک از این متغیرها و شیوه رمزگشایی آنها متفاوت خواهد بود. در الگوریتم ژنتیک جواب در قالب رشته‌ای از ژن‌ها که کروموزوم نامیده می‌شود، نمو پیدا می‌کند. به این دلیل انتخاب نوع رمزگشایی و طول رشته کروموزوم اهمیت ویژه‌ای می‌یابد. ژن ها می‌توانند به صورت اعداد بر مبنای دو، ۰, ۱ اعداد در مبنای ده، ۰, ۱, ۲, ۳, ۴, ۵, ۶, ۷, ۸, ۹ ، حرف انگلیسی و یا هر نماد دیگری استفاده شود. به دلیل این که اکثر این الگوریتم ها توسط برنامه های کامپیوتری مورد استفاده قرار می‌گیرند و معرفی ژن‌ها به صورت دودویی هم از نظر کاربرد هم از نظر پیاده سازی با کامپیوتر آسان‌تر است، عموماً در حل مسائل از این نوع کدگذاری استفاده می‌شود. ولی ممکن است به گونه‌های دیگر در نظر گرفته شود. مثلاً بعضی مسائل مانند فروشنده دوره گرد استفاده از رمزگشایی با نوع دیگری را می‌طلبند.

۱-۶-۲ جمعیت

چون در طبیعت همواره جمعیتی از موجودات وجود دارد، لذا خصوصیات موروثی تک تک موجودات و ارتباط بین آن‌ها می‌تواند رفته رفته موجب به وجود آمدن نسل‌های با قدرت انطباق بیشتر و کارایی بهتر در پاسخگویی به محیط گردد. هنگام استفاده از الگوریتم ژنتیک به عنوان یک عنصر بهینه کننده باید در خصوص تعیین اندازه جمعیت تصمیم گرفته شود. از جمله عواملی که در این کار موثر هستند می‌توان به شیوه نمایش فضای جواب، نوع مسئله‌ای که باید حل گردد و اپراتورهای به کار رفته اشاره کرد. اما آنچه بهترین جواب را می‌دهد، تجربه است. اندازه جمعیت تاثیر زیادی در کارایی الگوریتم‌های ژنتیک دارد. اگر تعداد جمعیت خیلی کم باشد این الگوریتم با کارایی بسیار پایین عمل خواهد کرد. کار کردن با جمعیت های زیاد اگر چه می‌توانند جستجوی آگاهانه‌تری داشته باشند، ولی این تعداد زیاد باعث می‌شوند که یک همگرایی زودهنگام به سمت نقاط زیر بهینه به دست آید. به علاوه تعداد ارزیابی‌ها بالاتر رود در نتیجه سرعت همگرایی کند و آهسته خواهد بود. معمولاً اندازه جمعیت را (از یک بزرگتر و از کل فضای جواب خیلی کوچکتر می‌گیرند) ثابت در نظر می‌گیرند.