دانشگاه تبریز
دانشکده عمران
گروه سازه

پایان‌نامه
برای دریافت درجه کارشناسی ارشد در رشته مهندسی عمران -سازه

عنوان

بهینه‌سازی سازه‌ها با استفاده از تابع جریمه خودانطباقی در الگوریتم ژنتیک

استاد راهنما
دکتر ناصر تقی‌زادیه

استاد مشاور
دکتر کامبیز کوهستانی

پژوهشگر
احسان محمودی کوچکسرایی

دیماه 1389

تقدیر و تشکر

این تحقیق را تقدیم می‌نمایم به:

پدر و مادر عزیز و مهربانم که هیچگونه کمکی را در تمام مراحل زندگی از بنده دریغ ننمودند و همیشه کمک حال بنده بودند و می‌باشند.

همچنین تقدیم به تمام اساتید دانشکده عمران بویژه دو استاد عزیز و گرانقدرم جناب دکتر تقی‌زادیه و جناب دکتر کوهستانی که در تمام مراحل تهیه این اثر، بنده را یاری و راهنمایی نمودند.

در انتها از تمام کسانی که در تهیه و تنظیم این اثر بنده را یاری نموده‌اند کمال تشکر و قدردانی را دارم

نام خانوادگی دانشجو: محمودی کوچکسرایی **نام**: احسان

عنوان پایان‌نامه: بهینه‌سازی سازه‌ها با استفاده از تابع جریمه خودانطباقی در الگوریتم ژنتیک

استاد راهنما: دکتر ناصر تقی زادیه **استاد مشاور**: دکتر کامبیز کوهستانی

مقطع تحصیلی: کارشناسی ارشد **رشته**: عمران **گرایش**: سازه **دانشگاه**: تبریز

دانشکده: مهندسی عمران **تاریخ فارغ التحصیلی**: دیماه1389 **تعداد صفحه**: 74

کلید واژه‌ها: بهینه‌سازی سازه‌ها، تابع جریمه خودانطباقی، الگوریتم ژنتیک، تابع هدف، بهینه‌سازی مقید، سازه‌های خرپایی.

چکیده:

امروزه با پیشرفت سریع دانش، بهینه‌سازی از اهمیت بالایی در علوم مختلف مهندسی برخوردار شده است. بهینه‌سازی یکی از مفاهیمی است که به خوبی از پل ارتباطی بین تئوری و عمل عبور کرده و دارای کاربرد گسترده‌ای می‌باشد.

با توجه به اینکه برای حل مناسب همهٔ مسائل موجود در بهینه‌سازی روش واحدی وجود ندارد، روش‌های متعددی از بهینه‌سازی برای حل مسائل مختلف پدید آمده‌اند. یکی از این روش‌ها، استفاده از الگوریتم ژنتیک است که یکی از اعضای خانوادهٔ مدل‌های محاسباتی الهام گرفته شده از روند تکامل انسان است. این الگوریتم‌ها راه‌های بالقوهٔ یک مسئله را در قالب کروموزوم‌های ساده کد می‌کنند و سپس عملگرهای ترکیبی را بر روی این ساختارها اعمال می‌نمایند. در الگوریتم ژنتیک، قیدها معمولاً با استفاده از مفهوم تابع جریمه مورد بررسی قرار می‌گیرند. بدین صورت که با جریمه افراد غیرممکن، از مقدار شایستگی آنها متناسب با درجه تخطی قیدها در جمعیت می‌کاهد. در بیشتر شکل‌های جریمه، باید بعضی از ضرایب را در ابتدای محاسبات تعیین نمود که معمولاً این ضرایب دارای مفهوم فیزیکی روشنی نمی‌باشند. به همین علت، تعیین مقادیر تقریبی این ضرایب حتی با آزمایش نیز تقریباً غیرممکن می‌باشد. با این وجود، بیشتر شکل‌های جریمه از ضرایب ثابت در سراسر محاسبات استفاده می‌کنند که ممکن است منتج به جریمه خیلی قوی و یا خیلی ضعیف در مدت مراحل مختلف تکامل گردد.

در این تحقیق، یک شکل جریمه جدید که مجزا از مشکلات فوق‌الذکر باشد، توسعه داده می‌شود. تابع جریمه ارائه شده قابلیت تعدیل خود را در مدت تکامل دارد. همچنین ضرایب استفاده شده در شکل ارائه شده مفهوم فیزیکی روشنی دارند. بنابراین، به راحتی می‌توان مقادیر تقریبی این ضرایب را با آزمایش تعیین نمود.

فهرست مطالب:

بخش اول: مقدمه

فصل اول: بهینه‌سازی ---------- 2

1-1- آشنایی با مفهوم بهینه‌سازی ---------- 2

فصل دوم: روشهای بهینه‌سازی ---------- 3

2-1- اصول حل یک مسئله بهینه‌سازی ---------- 3

2-1-1. بردارهای طراحی ---------- 3

2-1-2. قیدهای طراحی ---------- 3

2-1-3. تابع هدف ---------- 3

2-2- سیستم‌های تطبیقی ---------- 4

2-2-1. هوش مصنوعی ---------- 4

2-2-2. سیستم‌های تکاملی ---------- 5

2-3- تکنیک‌ها و روش‌های بهینه‌سازی سراسری ---------- 5

2-3-1. روش کلاسترینگ ---------- 5

2-3-2. شبیه‌سازی گرم و سرد کردن ---------- 5

2-3-3. الگوریتم‌های آماری بهینه‌سازی ---------- 6

2-3-4. جستجوی ممنوعه ---------- 6

2-3-5. انشعاب و تحدید ---------- 6

2-3-6. الگوریتم جامعه مورچگان ---------- 7

2-3-7. الگوریتم گروه ذرات ---------- 7

2-3-8. الگوریتم ژنتیک ---------- 8

2-4- انواع الگوریتم‌های ژنتیک ---------- 8

2-4-1. الگوریتم‌های ژنتیک ترتیبی ---------- 8

2-4-2. الگوریتم‌های ژنتیک موازی ---------- 9

2-4-3. الگوریتم‌های ژنتیک هیبرید ---------- 9

2-4-4. الگوریتم‌های ژنتیک خودسازمان ---------- 9

2-4-5. الگوریتم‌های ژنتیک آشفته ---------- 10
2-4-6. الگوریتم‌های ژنتیک زایشی ---------- 10
2-4-7. الگوریتم‌های ژنتیک حالت دائمی ---------- 11

بخش دوم: موادها و روش‌ها

فصل سوم: الگوریتم ژنتیک و مسائل مقید ---------- 13
3-1- کلیات الگوریتم ژنتیک ---------- 13
3-2- الگوریتم ژنتیک در MATLAB ---------- 14
3-3- تابع جریمه ---------- 17
3-3-1. تابع جریمه مرده ---------- 19
3-3-2. توابع جریمه استاتیکی ---------- 19
3-3-3. توابع جریمه دینامیکی ---------- 21
3-3-4. توابع جریمه سرد و گرم ---------- 22
3-3-5. توابع جریمه خودانطباقی ---------- 24
3-3-6. الگوریتم ژنتیک مجزا ---------- 25
3-3-7. توابع جریمه گروه تکاملی ---------- 26
3-4- روش جریمه پیشنهاد شده ---------- 26

فصل چهارم: مثال‌های کاربردی ---------- 28
4-1- خرپای صفحه‌ای 10 عضوی ---------- 28
4-2- خرپای صفحه‌ای 17 عضوی ---------- 29
4-3- خرپای صفحه‌ای 18 عضوی ---------- 31
4-4- خرپای فضایی 22 عضوی ---------- 32
4-5- خرپای فضایی 25 عضوی ---------- 34
4-6- خرپای فضایی 72 عضوی ---------- 37

فصل پنجم: تجزیه و تحلیل مثال‌های کاربردی ---------- 39
5-1- نتایج بهینه‌سازی خرپای صفحه‌ای 10 عضوی ---------- 40
5-2- نتایج بهینه‌سازی خرپای صفحه‌ای 17 عضوی ---------- 45

5-3- نتایج بهینه‌سازی خرپای صفحه‌ای 18 عضوی ----------47
5-4- نتایج بهینه‌سازی خرپای فضایی 22 عضوی----------49
5-5- نتایج بهینه‌سازی خرپای فضایی 25 عضوی----------52
5-6- نتایج بهینه‌سازی خرپای فضایی 72 عضوی----------54

بخش سوم: نتایج و پیشنهادات

فصل ششم: نتایج و پیشنهادات ----------62
6-1- نتیجه‌گیری ----------62
6-2- پیشنهادات----------63

پیوست ----------64

مراجع و مآخذ ----------71

فهرست جداول:

جدول 1. مشخصات کلی خرپای صفحه‌ای 10 عضوی ---------- 29

جدول 2. مشخصات کلی خرپای صفحه‌ای 17 عضوی ---------- 30

جدول 3. مشخصات کلی خرپای صفحه‌ای 18 عضوی ---------- 32

جدول 4. مشخصات کلی خرپای فضایی 22 عضوی ---------- 33

جدول 5. محدودهٔ تنش برای خرپای فضایی 22 عضوی ---------- 34

جدول 6. شرایط بارگذاری خرپای فضایی 22 عضوی ---------- 34

جدول 7. مشخصات کلی خرپای فضایی 25 عضوی ---------- 35

جدول 8. محدودهٔ تنش برای خرپای فضایی 25 عضوی ---------- 36

جدول 9. شرایط بارگذاری خرپای فضایی 25 عضوی ---------- 36

جدول 10. شرایط بارگذاری خرپای فضایی 72 عضوی ---------- 38

جدول 11. گروه‌بندی سطح مقطع اعضا خرپای فضایی 72 عضوی ---------- 38

جدول 12. اطلاعت مربوط به جعبه ابزار الگوریتم ژنتیک در MATLAB ---------- 39

جدول 13. مقایسه نتایج تحقیقات گذشته با تحقیق حاضر مربوط به حالت اول خرپای صفحه‌ای 10 عضوی -- 40

جدول 14. مقادیر تنش و تغییرمکان حالت اول خرپای صفحه‌ای 10 عضوی ---------- 41

جدول 15. شش مرحله از اجرای برنامه برای حالت اول خرپای صفحه‌ای 10 عضوی ---------- 41

جدول 16. مقایسه نتایج تحقیقات گذشته با تحقیق حاضر مربوط به حالت دوم خرپای صفحه‌ای 10 عضوی -- 43

جدول 17. مقادیر تنش و تغییرمکان حالت دوم خرپای صفحه‌ای 10 عضوی ---------- 44

جدول 18. شش مرحله از اجرای برنامه برای حالت دوم خرپای صفحه‌ای 10 عضوی ---------- 44

جدول 19. مقایسه نتایج تحقیقات گذشته با تحقیق حاضر مربوط به خرپای صفحه‌ای 17 عضوی---------- 45

جدول 20. مقادیر تنش و تغییرمکان خرپای صفحه‌ای 17 عضوی---------- 46

جدول 21. شش مرحله از اجرای برنامه خرپای صفحه‌ای 17 عضوی ---------- 47

جدول 22. مقایسه نتایج تحقیقات گذشته با تحقیق حاضر مربوط به خرپای صفحه‌ای 18 عضوی---------- 48

جدول 23. مقادیر تنش خرپای صفحه‌ای 18 عضوی ---------- 49

جدول 24. شش مرحله از اجرای برنامه خرپای صفحه‌ای 18 عضوی ---------- 49

جدول 25. مقایسه نتایج تحقیقات گذشته با تحقیق حاضر مربوط به خرپای فضایی 22 عضوی---------- 50

جدول 26. مقادیر تنش و تغییرمکان خرپای فضایی 22 عضوی ---------- 51

جدول 27. شش مرحله از اجرای برنامه خرپای فضایی 22 عضوی---------- 51

جدول 28. مقایسه نتایج تحقیقات گذشته با تحقیق حاضر مربوط به خرپای فضایی 25 عضوی---------- 52

جدول 29. مقادیر تنش و تغییرمکان خرپای فضایی 25 عضوی ---------- 53

جدول 30. شش مرحله از اجرای برنامه خرپای فضایی 25 عضوی---------- 54

جدول 31. مقایسه نتایج تحقیقات گذشته با تحقیق حاضر مربوط به حالت اول خرپای فضایی 72 عضوی ---- 55

جدول 32. مقادیر تنش و تغییرمکان حالت اول خرپای فضایی 72 عضوی ---------- 56

جدول 33. شش مرحله از اجرای برنامه برای حالت اول خرپای فضایی 72 عضوی ---------- 57

جدول 34. مقایسه نتایج تحقیقات گذشته با تحقیق حاضر مربوط به حالت دوم خرپای فضایی 72 عضوی ---- 58

جدول 35. مقادیر تنش و تغییرمکان حالت دوم خرپای فضایی 72 عضوی ---------- 59

جدول 36. شش مرحله از اجرای برنامه برای حالت دوم خرپای فضایی 72 عضوی ---------- 60

جدول 37. مشخصات کلی خرپای فضایی 72 عضوی ---------- 65

فهرست اشکال و نمودارها:

شکل 1. خرپای صفحه‌ای 10 عضوی ---------- 29

شکل 2. خرپای صفحه‌ای 17 عضوی ---------- 30

شکل 3. خرپای صفحه‌ای 18 عضوی ---------- 31

شکل 4. خرپای فضایی 22 عضوی ---------- 33

شکل 5. خرپای فضایی 25 عضوی ---------- 36

شکل 6. خرپای فضایی 72 عضوی---------- 37

نمودار 1. نمودار ستونی نتایج مربوط به حالت اول خرپای صفحه‌ای 10 عضوی ---------- 41

نمودار 2. نمودار ستونی نتایج مربوط به حالت دوم خرپای صفحه‌ای 10 عضوی---------- 43

نمودار 3. نمودار ستونی نتایج مربوط به خرپای صفحه‌ای 17 عضوی ---------- 46

نمودار 4. نمودار ستونی نتایج مربوط به خرپای صفحه‌ای 18 عضوی ---------- 48

نمودار 5. نمودار ستونی نتایج مربوط به خرپای فضایی 22 عضوی---------- 50

نمودار 6. نمودار ستونی نتایج مربوط به خرپای فضایی 25 عضوی---------- 53

نمودار 7. نمودار ستونی نتایج مربوط به حالت اول خرپای فضایی 72 عضوی ---------- 55

نمودار 8. نمودار ستونی نتایج مربوط به حالت دوم خرپای فضایی 72 عضوی ---------- 58

بخش اول

مقدمه

فصل اول: بهینه‌سازی

1-1- آشنایی با مفهوم بهینه‌سازی

امروزه با پیشرفت سریع دانش، بهینه‌سازی از اهمیت بالایی در علوم مختلف مهندسی برخوردار شده است. بهینه‌سازی یکی از مفاهیمی است که به خوبی از پل ارتباطی بین تئوری و عمل عبور کرده و دارای کاربرد گسترده‌ای می‌باشد. مهندسان عمران، مکانیک، برق، کامپیوتر، شیمی، نفت، متالوژی و ... همگی به دنبال بهینه‌سازی ابزار آلات، فرمول‌ها، معادلات و مدل‌های مربوط به شاخه تخصصی خود در راستای استفاده هر چه بهتر از زمان و منابع هستند.

بدست آوردن بهترین نتیجه ممکن برای یک مسئله با توجه به شرایط حاکم بر آن را بهینه‌سازی گویند. از همان سال‌های اولیه پیدایش بشر، تمایل به انجام کارها و فعالیت‌ها با کمترین زحمت و نائل شدن به بیشترین سود و منفعت، از مشخصه‌های ذاتی انسان‌ها و مهمترین دغدغه‌های فکری آنها بود.

امروزه در طراحی، ساخت و نگهداری هر سیستم مهندسی، مهندسان باید تصمیمات مدیریتی متعددی را در مراحل مختلف اتخاذ نمایند. می‌توان بهینه‌سازی را به عنوان فرآیند یافتن شرایطی که مقدار بیشینه و یا کمینه یک تابع را بدست می‌دهد، تعریف نمود. با توجه به اینکه برای حل مناسب همهٔ مسائل موجود در بهینه‌سازی روش واحدی وجود ندارد، روش‌های متعددی از بهینه‌سازی برای حل مسائل مختلف پدید آمده‌اند. اهمیت موضوع بهینه‌سازی اولین بار در طراحی سازه های هوافضا با وزن کمینه مورد توجه قرار گرفت. در این سازه‌ها، با توجه به حساسیت فوق‌العاده کاربرد آنها، به جای اینکه مبنای طراحی هزینه آن باشد، وزن سازه هدف بهینه‌سازی خواهد بود. اما در دیگر صنایع مربوط به سیستم‌های مهندسی همچون عمران، مکانیک و صنایع خودرو ممکن است هزینه در درجه اول اهمیت باشد، هر چند که وزن سیستم، عملکرد و حتی هزینه سازه را تحت تاثیر قرار خواهد داد. افزایش روزافزون کاربرد سازه-های مهندسی و محدود بودن مواد خام و کمبود منابع انرژی از جملع عواملی است که طراحان را به سوی طراحی سازه‌های سبک، ارزان قیمت و در عین حال کارا، وادار می‌سازد. [1]، [4]

در ادامه این تحقیق به بررسی روش‌های بهینه سازی در فصل دوم، الگوریتم ژنتیک در مسائل مقید و بیان روش پیشنهادی در فصل سوم، بیان چند مورد مثال کاربردی در فصل چهارم، نتایج حاصل از روش پیشنهادی در این تحقیق در فصل پنجم و نتیجه‌گیری و پیشنهادات در فصل ششم می‌پردازیم.

فصل دوم: روش‌های بهینه‌سازی

2-1- اصول حل یک مسئله بهینه‌سازی

بطور کلی در بهینه‌سازی، یک سری اصول و راه‌های مشخصی در بهینه‌کردن مسئله وجود دارد. در بهینه‌کردن مسائل باید موارد زیر را در نظر گرفت

2-1-1. بردارهای طراحی

هر سیستم مهندسی با مجموعه‌ای از کمیت‌ها بیان می‌شود که برخی از آنها بصورت متغیرهایی در فرآیند تصمیم‌گیری ظاهر می‌شوند. کمیت‌های معینی که در خارج از مسئله دارای مقادیر ثابتی هستند، متغیرهای معلوم نامیده می‌شوند. سایر کمیت‌ها به صورت متغیرهایی در فرآیند طراحی رفتار می‌کنند که در اصطلاح بهینه‌سازی معمولاً بردارها یا متغیرهای طراحی نامیده می‌شوند. [1]

2-1-2. قیدهای طراحی

در بسیاری از مسائل عملی، نمی‌توان متغیرها را به دلخواه انتخاب کرد، بلکه این متغیرها باید ویژگی‌های عملی مشخص و دیگر نیازمندی‌ها را برآورده کنند. قیدهایی که باید به منظور تهیه یک طرح قابل قبول برآورده شوند، قیدهای طراحی گویند. [1]

2-1-3. تابع هدف

روش‌های طراحی معمول، ما را در یافتن یک طرح قابل قبول یاری می‌دهند. این نوع طراحی، تنها نیازمندی‌های عملی و دیگر نیازمندی‌های مسئله را برآورد می‌سازد. اما در اینجا، تنها قابل قبول بودن یک طرح مورد نظر نیست بلکه هدف از بهینه‌سازی انتخاب بهترین طرح از میان طراحی‌های قابل قبول موجود می‌باشد. بنابراین باید معیاری برای طرح‌های قابل قبول و انتخاب بهترین آنها تعیین شود. چنین معیاری که طرح نسبت به آن بهینه می‌شود را بصورت تابعی از متغیرهای طراحی بیان می‌کنند و آن را تابع معیار و یا تابع هدف می‌نامند. انتخاب تابع هدف در بیشتر مسائل طراحی ساده بنظر می‌رسد ولی با این وجود، ممکن است در برخی مواقع، بهینه‌سازی نسبت به یک معیار مشخص، به نتایجی بیانجامد که نسبت به یک معیار دیگر رضایت‌بخش نباشد. در نتیجه باید در انتخاب تابع هدف دقت لازم را مد نظر قرار داد تا به یک معیار بهینه مناسب برسیم. [1]

2-2- سیستم‌های تطبیقی

در دهه نود قرن نوزدهم، تحقیقات دو تن از پیشگامان علم ژنتیک، وجود نوعی دستور یا کد وراثتی را بر همگان اثبات نمود. چارلز داروین[1] در سال 1859 نظریه تکامل خود را مطرح کرده بود و گرگور مندل[2] نیز در سال 1865 موفق شده بود قوانین اساسی وراثت را کشف کند. اما هیچ یک از آنها نتوانستند دریابند که چه عواملی باعث کنترل و هدایت سیستم‌های مورد مطالعه آنها می‌شود. جان هالند[3] توانست برای اولین بار این مشاهدات را به صورت فرمول‌بندی ریاضی و منطق ژنتیک در آورد و شرح دهد که در سیستم‌های طبیعی چگونه انتخاب‌ها صورت می‌گیرد و چگونه می‌توان یک سیستم مصنوعی نرم‌افزاری طراحی نمود که بتواند با استفاده از مکانیزم انتخاب در طبیعت، به حل مسائل مشکل کمک نماید. به همین منظور نیاز به یک سری تکنیک‌های عددی می‌باشد که وقتی به یک مسئله اعمال می‌شود بصورت خودکار و مستقل اهمیت نسبی هر یک از پارامترهای ورودی را برای حل مسئله تنظیم می‌کند. این حوزه از محاسبات هدف اصلی سیستم‌های تطبیقی[4] می‌باشد. حوزه سیستم‌های تطبیقی شامل دو شاخه اصلی به نام هوش مصنوعی[5] و سیستم‌های تکاملی[6] می‌باشد. [1]

2-2-1. هوش مصنوعی

واژه هوش مصنوعی اولین بار در سال 1950 بکار برده شد. فلسفه وجودی هوش مصنوعی، در دو مفهوم اساسی هوش بشری، یعنی "چگونه" و "چرا" خلاصه می‌شود. در این حوزه، تکنیک‌های حساب گزاره‌های سیمون، منطق گزاره‌ای[7] و منطق مسندی[8] در شبیه‌سازی شیوه‌های استنتاج راه‌حل، یعنی همان بخش "چگونه" از هوش انسان بکار برده شده است. این روش‌ها در زمینه سیستم‌های خبره و تصمیم‌گیری بسیار مفید بوده است. موضوع مورد بررسی دیگر محققان از حوزه هوش مصنوعی، تقلید مکانیزم اندیشه است یعنی بخش "چرا" هوش بشری. این سیستم‌های مکانیکی شامل شبکه‌های معنوی[9] و شبکه‌های عصبی می‌باشد که شبکه‌های عصبی شناخته‌ترین سیستم در کاربرد عملی می‌باشد. [1]، [3]

[1] Charles Darwin
[2] Gregor Mendel
[3] John Holland
[4] Adaptive Systems
[5] Artificial Intelligence
[6] Evolutionary Systems
[7] Propositional Logic
[8] Predicate Logic
[9] Semantic Networks

2-2-2. سیستم‌های تکاملی

یک شاخه اصلی از سیستم‌های تطبیقی، سیستم تکاملی می‌باشد که گاهی از آن به یک حیات مصنوعی تعبیر می‌شود. واژه تکاملی استعاره از یک سیستم رایانه-مبنا[1] است که بر اساس رویکرد قانون بقای بهترین‌ها[2] به حل مسئله می‌پردازد بعبارت دیگر سیستم تکاملی، یک رقابت قانونمند میان راه‌حل‌های موجود در محیط مسئله است. تناسب و برازش هر عنصر از محیط بر مبنای ارزش نسبی آن، که با توجه به تابع هدف مسئله تعیین می‌شود، می‌باشد. جذابیت این سیستم بعنوان یک استراتژی حل مسئله، در توانایی طبیعت در یافتن راه‌حل‌های متفاوت برای مسائل دینامیکی و پیچیده است. بنابراین اگر تکامل بعنوان یک الگوریتم حل مسئله مورد توجه قرار گیرد، می‌توان از میان راه‌حل‌های مختلف موجود برای یک مسئله، راه‌حل بهینه را اختیار نمود. [1]، [8]

2-3- تکنیک‌ها و روش‌های بهینه‌سازی سراسری

منظور از بهینه‌سازی سراسری یافتن مجموعه‌ای از بهترین پارامترها برای بهینه‌سازی تابع هدف می‌باشد. این نوع مسائل بهینه‌سازی سراسری در کلاس برنامه‌نویسی غیر خطی[3] قرار می‌گیرند. از جمله این روش‌ها می‌توان به موارد زیر اشاره نمود

2-3-1. روش کلاسترینگ

در این روش یک جستجوی محلی از چند نقطه که در دامنهٔ جستجو توزیع شده‌اند، انجام می‌شود. در این روش برای جلوگیری از تکرار نقطه مینیمم محلی و افزایش کارایی الگوریتم سعی شده است تا با تعیین دقیق نقاط آغازین از این نقیصه جلوگیری شود. روش کلاسترینگ برای بهینه‌سازی توابع بدون محدودیت توسعه داده شده‌اند. فرض اولیه در استفاده از این روش این است که تابع هدف نسبتاً ساده و کم هزینه باشد. [3]

2-3-2. شبیه‌سازی گرم و سرد کردن

این روش از فرآیند گرم و سرد کردن فلز الهام گرفته است. در کاربرد این روش در سیستم‌های ترمودینامیک، حالت آغازین سیستم با توجه به دو پارامتر، انرژی E و دما T تعیین می‌شود. در این سیستم با ثابت نگه داشتن T، تغییرات انرژی dE محاسبه می‌شود. اگر dE منفی باشد پیکربندی جدید پذیرفته می‌شود اگر dE مثبت باشد پیکربندی جدید با احتمالی برابر با فاکتور بولتزمان $e^{-\frac{dE}{T}}$ پذیرفته

[1] Computer Based
[2] Survival of the Fitness
[3] Non Linear Programming

می‌گردد. این حالت تا زمانی ادامه می‌یابد تا با کاهش دما سیستم به حالت انجماد یعنی T=0 برسد. این روش به راحتی برای حل مسائل ترکیبی قابل استفاده است. در این روش باید در انتخاب دمای اولیه، تعداد تکرارها در هر دما و میزان کاهش دما در هر گام دقت لازم صورت گیرد. [3]

2-3-3. الگوریتم‌های آماری بهینه‌سازی

این روش، از مدل‌های آماری تابع هدف برای تحت تاثیر قرار دادن نقاط نمونه جدید استفاده می‌کند. این روش‌ها با آرگومان‌های احتمالی بیز[1] سر وکار دارند. از این روش‌ها می‌توان برای پوشش شرایط معمولی در فرآیند بهینه‌سازی استفاده نمود. الگوریتم‌های بهینه‌سازی آماری عموماً درباره مسائل رقابتی مورد استفاده قرار می‌گیرند البته کارایی آنها با توجه به پیچیدگی‌های ریاضی برای پیاده‌سازی محدود است. [3]

2-3-4. جستجوی ممنوعه[2]

ایده اصلی جستجوی ممنوعه در سال 1986 توسط گلوور[3] مطرح گشت. در این روش از یک تابع خوداکتشاف در کنار تابع اکتشافی استفاده می‌شود. راهبرد کلی در جستجوی ممنوعه، دوری کردن از قرار گرفتن در یک چرخه تاخیری می‌باشد. این کار با ممنوع کردن و یا جریمه کردن حرکات تکراری انجام می‌شود. در واقع این روش اجازهٔ جستجوی نقاطی که قبلاً مورد بررسی قرار گرفته‌اند را سلب می‌کند. این روش تا حدی از مشاهدات رفتار انسان الهام گرفته شده است. در این روش فرض اصلی بر این است امکان ورود به یک مسیر وارسی شده در فضای جستجو وجود ندارد، برای جلوگیری از تکرار، یک راه‌حل ضعیف‌تر را که در مسیری جدید وجود دارد، انتخاب می‌کند. با این کار از افتادن در مینیمم‌های محلی جلوگیری می‌شود و در نهایت به راه‌حل مطلوب منتهی می‌گردد. [3]، [4]

2-3-5. انشعاب و تحدید

انشعاب و تحدید یکی از روش‌های معمول جستجو می‌باشد. فرض کنید هدف مینیمم‌سازی تابع $f(x)$ باشد و متغیر x در ناحیهٔ مشخصی محدود باشد. با استفاده از انشعاب و تحدید باید ابزاری برای محاسبه کران پایین مسئله، ابزاری برای تقسیم مسئله به زیر مسائل کوچکتر (فرزندان) و راهی برای محاسبه کران بالای راه‌حل حداقل برای برخی از نمونه‌های مسئله وجود داشته باشد. این روش از حالتی تحت عنوان حالت ریشه آغاز می‌شود، اگر محدودهٔ مسئله با جواب تطبیق داشته باشد که حل مسئله

[1] Bayesian
[2] Tabu
[3] Glover

پایان می‌پذیرد. در غیر این صورت این حالت به نواحی جدیدی تقسیم می‌شود. این روند تا آنجا ادامه می‌یابد تا به یک راه‌حل مناسب دست یابد. [3]

2-3-6. الگوریتم جامعه مورچگان[1]

الگوریتم بهینه‌سازی جامعه مورچگان (Aco) اولین بار توسط دوریگو[2] و همکارانش در سال 1991 مطرح گشت و در این دهه تعدادی از محققان جهت توسعه آن فعالیت نمودند. الگوریتم جامعه مورچگان یک روش نسبتاً جدید برای حل مسائل بهینه‌سازی است که بر اساس شبیه‌سازی رفتار جامعه مورچگان واقعی، پایه‌ریزی شده است. مورچگان برای یافتن غذا کوتاهترین مسیر را از محل زندگی خود تا منبع غذا پیدا می‌کنند. برای این منظور، مورچگان یک ماده به نام فرومون[3] با حرکت بر روی مسیرها قرار می‌دهند. هر چه میزان فرومون قرار گرفته بر روی مسیر بیشتر باشد شانس انتخاب آن مسیر توسط مورچگان بیشتر می‌شود. مراحل اصلی الگوریتم جامعه مورچگان عبارتند از

- تعریف مقادیر پارامترها و میزان فرومون اولیه
- پروسه ساخت حل توسط هر مورچه
- به هنگام نمودن میزان فرومون

که پس از تعریف مقادیر اولیه پارامترها و میزان فورمون اولیه، وارد حلقه تکرار الگوریتم شده و پروسه تکامل شامل ساخت حل و به هنگام نمودن فرومون، تا رسیدن به شرایط مطلوب ادامه خواهد داشت. هدف از به هنگام نمودن فرومون، افزایش میزان فرومون در جواب‌های خوب و کاهش آن در جواب‌های ضعیف است. [5]، [6]، [7]

2-3-7. الگوریتم گروه ذرات[4]

الگوریتم گروه ذرات (Pso) اولین بار توسط راسل ابرهارت[5] و جیمز کندی[6] در سال 1995 ابداع گشت. ایده اولیه این روش از رفتار اجتماعی جانوران مثل پرواز دسته جمعی پرندگان یا حرکت دسته جمعی ماهی‌ها الهام گرفته شده است. الگوریتم گروه ذرات با یک ماتریس جمعیت بصورت تصادفی شروع شده که هر یک از سطرهای ماتریس یک پرنده در نظر گرفته شده که ذره نامیده می‌شود. هر پرنده یک

[1] Ant Colony Optimization
[2] Dorigo
[3] Pheromone
[4] Particle Swarm Optimization
[5] Russell Eberhart
[6] James Kennedy

موقعیت داشته و یک شایستگی که بر اساس موقعیتش بدست می‌آید که با یک سرعت حرکت می‌کند. پرندگان سرعتشان را بر اساس بهترین پاسخی که تاکنون بدست آمده و بهترین پاسخ در نسل فعلی تغییر می‌دهند. این سرعت با موقعیت پرنده جمع شده و موقعیت جدید پرنده را بدست می‌آورد. اگر بهترین پاسخ در نسل فعلی از بهترین پاسخی که تاکنون بدست آمده مناسب‌تر بود، جایگزینی صورت می‌گیرد و این کار تا رسیدن به یک جواب مطلوب ادامه می‌یابد. [9]، [10]

2-3-8. الگوریتم ژنتیک

اصطلاح الگوریتم ژنتیک دارای دو معنای کلی می‌باشد. معنای اول که یک تفسیر محدود از این ابزار است، اولین بار توسط جان هالند در سال 1975 مطرح شد. بسیاری از نظریه‌های فعلی الگوریتم‌های ژنتیک نیز بر اصول اولیه مطرح شده توسط هالند استوار هستند. در معنای دوم الگوریتم‌های ژنتیک بر هر مدل جمعیت‌گرا که از عملگرهای انتخاب و ترکیب برای تولید نمونه‌های جدید در فضای جستجو استفاده می‌کند، دلالت دارد. الگوریتم‌های ژنتیک یکی از اعضای خانوادهٔ مدل‌های محاسباتی الهام گرفته شده از روند تکامل انسان است. این الگوریتم‌ها راه‌های بالقوهٔ یک مسئله را در قالب کروموزوم‌های ساده کد می‌کنند و سپس عملگرهای ترکیبی را بر روی این ساختارها اعمال می‌نمایند. الگوریتم‌های ژنتیک اغلب به عنوان روشی برای بهینه‌سازی توابع شناخته می‌شود که البته دامنهٔ استفاده از این روش بسیار گسترده‌تر از این است. [1]، [2]، [8]

2-4- انواع الگوریتم‌های ژنتیک

الگوریتم‌های ژنتیک نسل جدیدی از روش‌های جستجو و بهینه‌سازی می‌باشند. این الگوریتم‌ها به دلیل کارایی، قدرت، سودمندی و سادگی‌شان بسیار پرطرفدار هستند. الگوریتم‌های ژنتیک دارای انواع مختلفی می‌باشند که هر یک در مورد محدودهٔ خاصی از مسائل ساده و دشوار دارای کاربرد می‌باشند که از جمله آنها می‌توان به موارد زیر اشاره نمود

2-4-1. الگوریتم‌های ژنتیک ترتیبی[1]

الگوریتم‌های ژنتیک ترتیبی کاربرد موفقی در بسیاری از زمینه‌ها دارند اما به هر حال دارای نقایصی می‌باشند. از جمله نقایص این الگوریتم‌ها می‌توان به موارد زیر اشاره نمود [3]، [4]، [8]

- این روش معمولاً غیر بهینه است.
- عملیات ارزیابی در این روش بسیار زمان‌بر می‌باشد.

[1] Sequential Genetic Algorithms

- برای حل مسائل به این روش اندازهٔ جمعیت باید بسیار بزرگ باشد.

2-4-2. الگوریتم‌های ژنتیک موازی

در راستای برطرف کردن نقایص یاد شده برای الگوریتم‌های ژنتیک ترتیبی و همچنین مطالعهٔ مدل‌هایی با کارایی و بهره‌وری بالاتر مورد استفاده قرار می‌گیرند. در هنگام استفاده از الگوریتم‌های ژنتیک موازی باید بین دو مفهوم زیر تمایز قائل شد.

- الگوریتم ژنتیک موازی یک مدل خاصی از الگوریتم‌های ژنتیک است.
- الگوریتم ژنتیک موازی ابزاری برای پیاده‌سازی الگوریتم‌های ژنتیک است.

در الگوریتم‌های ژنتیک موازی کل جمعیت در قالب یک فرم توزیع شده ارائه می‌شوند. به این معنی که یا جمعیت از چند زیر جمعیت مستقل از هم تشکیل می‌شود و یا اینکه تنها یک جمعیت وجود دارد ولی هر یک از اعضای جمعیت تنها با مجموعهٔ محدودی از همسایگانش تعامل دارند. یکی از مزایای این روش عدم توقف در همگرایی زودرس می‌باشد. علت آن را می‌توان در توانایی برقراری گوناگونی مناسب بین زیر جمعیت‌ها از طریق مبادلهٔ خصوصیات بین زیر مجموعه‌ها دانست. [3]، [4]، [8]

2-4-3. الگوریتم‌های ژنتیک هیبرید

هدف از طراحی الگوریتم‌های ژنتیک هیبرید استفاده از توابع اکتشافی در جهت بهبود کیفیت فرزندان تولید شده طی عملیات تلفیق می‌باشد. روال کلی کار به این صورت است که پس از تهیه جمعیت تصادفی، عملیات تلفیق بین دو والدین که بصورت تصادفی انتخاب شده‌اند، انجام می‌شود. سپس از توابع اکتشافی Remove Sharp و Local Opt برای نزدیک کردن فرزندان به حداکثر محلی استفاده می‌شود. در صورت برتری شایستگی فرزندان نسبت به والدین، فرزندان جایگزین والدین می‌شوند و در نهایت مابقی عملیات بر روی جمعیت انجام می‌شود. [3]، [4]، [8]

2-4-4. الگوریتم‌های ژنتیک خودسازمان[1]

الگوریتم‌های ژنتیک خودسازمانی نوعی الگوریتم‌های ژنتیک با پارامترهای انطباقی می‌باشند. به این معنی که پارامترهای الگوریتم ژنتیک نظیر اندازهٔ جمعیت، احتمال تلفیق یا احتمال جهش در حین اجرای الگوریتم ژنتیک تغییر می‌کنند. در این روش تعادل بین اکتشاف و استخراج به گونهٔ متفاوتی از الگوریتم ژنتیک برقرار می‌شود. این کار با تغییر مقادیر احتمال جهش و احتمال تلفیق با توجه به مقادیر شایستگی راه‌حل‌ها انجام می‌پذیرد. مقادیر احتمال جهش و احتمال تلفیق زمانی که جمعیت در یک حداکثر محلی

[1] Adaptive Genetic Algorithms

قرار می‌گیرد افزایش یافته و زمانی که جمعیت در فضای جستجو پراکنده می‌شود کاهش می‌یابند. [3]، [4]، [8]

2-4-5. الگوریتم‌های ژنتیک آشفته[1]

هدف از توسعه الگوریتم‌های ژنتیک آشفته ایجاد تابع شایستگی $f(x)$ بصورت مجموع m زیر تابع مستقل $f_i(x_i)$ می‌باشد. هر یک از $f_i(x_i)$ بر روی k مکان هندسی تعریف شده‌اند که k سطح فریب را در فریبنده‌ترین زیر تابع تخمین می‌زند. الگوریتم‌های ژنتیک آشفته شامل مراحل زیر است

- فاز initialization (مقداردهی آغازین)
- فاز primordial
- فاز juxtapositional

الگوریتم‌های ژنتیک آشفته در مرحلهٔ شروع الگوریتم از روش مقداردهی آغازین شمارشی جزیی[2] برای تعریف جمعیت اولیه استفاده می‌کند. به این ترتیب جمعیت شامل همهٔ راه‌حل‌های جزیی ممکن روی k مکان هندسی می‌باشد. بنابراین هر بلوک دقیقاً یک بار در یکی از راه‌حل‌ها ارائه می‌شود و در بین راه‌حل‌ها بلوک تکراری وجود ندارد. پس از تعریف جمعیت اولیه، از الگوریتم انتخاب مسابقه‌ای برای انجام فاز primordial استفاده می‌شود. در این مرحله اندازهٔ جمعیت با حذف راه‌حل‌های نامناسب کوچک‌تر می‌گردد. در فاز juxtapositional مانند یک الگوریتم‌های ژنتیک ساده از عملگرهای ترکیبی استفاده می‌شود. در این مرحله عملگر تلفیق استاندارد با عملگر برش و پیوند[3] جایگزین می‌شود. احتمال برش و پیوند معمولاً طوری تعیین می‌گردد که امکان رشد رشته‌ها را از k تا 1 به سرعت فراهم آورد. [3]، [4]، [8]

2-4-6. الگوریتم‌های ژنتیک زایشی[4]

الگوریتم‌های ژنتیک زایشی، فرزندان جدید را با توجه به جمعیت حاضر و با استفاده از عملگرهای ژنتیک تولید می‌کند و به این ترتیب جمعیت جدید ایجاد شده جایگزین جمعیت فعلی می‌شود. معمولاً در عملیات جایگزینی مربوط به روش زایشی در هر تکرار کل جمعیت حاضر با جمعیت جدید جایگزین می‌شود. عموماً انتظار می‌رود که فرزندان بلوک‌های مهم را از بهترین رشته‌های جمعیت به نسل بعد

[1] Messy Genetic Algorithms
[2] Partially Enumrative Initialization
[3] Cut and Splice
[4] Generational Genetic Algorithms

منتقل نمایند. در واقع در این روش بهترین رشته‌های جمعیت حاضر به نسل بعد منتقل نمی‌شود. به طبع، این روش دارای مشکلاتی به خاطر نحوهٔ عملکرد خود می‌باشد. یکی از مشکلات این روش این است که رشته‌های مرغوب قابلیت تولید مثل مجدد در نسل‌های بعدی را از دست می‌دهند. همچنین امکان دارد در صورت وقوع تلفیق و یا جهش مقادیر بیت‌های مهم یک رشته تخریب شده و از انتشار این مقادیر به نسل بعدی جلوگیری شود. برای رفع این مشکل در پیاده‌سازی الگوریتم‌های ژنتیک زایشی از مفهوم نخبه‌گرایی استفاده می‌شود. به این معنی که بهترین رشته‌های جمعیت فعلی بدون قید و شرط به نسل بعدی منتقل می‌شوند و به این ترتیب از نابودی رشته‌های مرغوب جلوگیری می‌گردد. [3]، [8]

2-4-7. الگوریتم‌های ژنتیک حالت دائمی[1]

الگوریتم‌های ژنتیک حالت دائمی افزایشی (گام به گام) با مدل زایشی متفاوت می‌باشند. در این روش عموماً در هر گام زمانی فقط یک عضو جدید به جمعیت جدید اضافه می‌شود. استراتژی جایگزینی تعیین می‌کند که کدامیک از اعضای جمعیت باید با عضو جدید جایگزین گردد. بعبارت دیگر، در الگوریتم‌های ژنتیک حالت دائمی تنها تعداد کمی از افراد در هر تکرار جایگزین می‌گردند. الگوریتم‌های ژنتیک حالت دائمی کم جمعیت به سرعت گوناگونی خود را از دست می‌دهند. الگوریتم‌های ژنتیک حالت دائمی با جمعیت زیاد هزینه محاسباتی بالایی داشته و سرعت همگرایی آنها بسیار کم است. البته این روش دارای مزایایی نیز می‌باشد. از جمله این مزایا می‌توان گستردگی و توزیع مناسب جمعیت در فضای جستجو اشاره نمود. الگوریتم‌های ژنتیک حالت دائمی خودبه‌خود از استراتژی نخبه‌گرایی پیروی می‌کند زیرا در این روش فقط بدترین رشته‌ها حذف می‌شوند و رشته‌های مناسب بصورت خودکار در جمعیت باقی می‌مانند. [3]، [8]

[1] Steady State Genetic Algorithms