
بسم الله الرحمن الرحيم

دانشکده مهندسی عمران
رشته عمران گرایش سازه

پایان‌نامه کارشناسی ارشد

بهینه سازی توپولوژی سازه ها تحت بارهای دینامیکی با استفاده از الگوریتم GESO

نگارنده: احسان روح پرور

استاد راهنما:
دکتر علی کیهانی

شهریور ۱۳۹۶

شماره: ۲۸۵/۹۶/ت ت ۱
تاریخ: ۹۶/۷/۲۴

باسمه‌تعالی

مدیریت تحصیلات تکمیلی

فرم شماره (۳) صورتجلسه نهایی دفاع از پایان نامه دوره کارشناسی ارشد

با نام و یاد خداوند متعال، ارزیابی جلسه دفاع از پایان نامه کارشناسی ارشد آقای احسان روح‌پرور با شماره دانشجویی ۹۲۰۷۳۰۴ رشته مهندسی عمران گرایش سازه تحت عنوان بهینه‌سازی توپولوژی سازه‌ها تحت بارهای دینامیکی با استفاده از الگوریتم GESO که در تاریخ ۱۳۹۶/۰۶/۲۲ با حضور هیأت محترم داوران در دانشگاه صنعتی شاهرود برگزار گردید به شرح ذیل اعلام می گردد:

قبول (با امتیاز ۱۷٫۷ درجه خوب) ☑ **مردود** ☐

نوع تحقیق: نظری ☐ **عملی** ☐

عضو هیأت داوران	نام ونام خانوادگی	مرتبهٔ علمی	امضاء
۱ـ استادراهنمای اول	علی کیهانی	دانشیار	
۲- استادراهنمای دوم			
۳- استاد مشاور			
۴- نماینده تحصیلات تکمیلی	محسن کرامتی	استادیار	
۵- استاد ممتحن اول	وحیدرضا کلات‌جاری	دانشیار	
۶ـ استاد ممتحن دوم	جلیل شفائی	استادیار	

نام و نام خانوادگی رئیس دانشکده: احمد احمدی

تاریخ و امضاء و مهر دانشکده:

تبصره: در صورتی که کسی مردود شود حداکثر یکبار دیگر (در مدت مجاز تحصیل) می تواند از پایان نامه خود دفاع نماید (دفاع مجدد نباید زودتر از ۴ ماه برگزار شود).

تقدیم

این پایان‌نامه را ضمن تشکر و سپاس بیکران و در کمال افتخار و امتنان تقدیم می‌نمایم به

پدر و مادر دلسوز و مهربانم که زحماتشان با هیچ واژه‌ای قابل قدردانی نیست.
دستم را گرفتند و راه رفتن را در این وادی زندگی پر از فراز و نشیب آموختند.

تشکر و قدردانی

چگونه سپاس گویم مهربانی و لطف استاد گرامی‌ام را که سرشار از عشق و یقین است. چگونه سپاس گویم تأثیر علم‌آموزی شما را که چراغ روشن هدایت را بر کلبه‌ی محقر وجودم فروزان ساخته است. آری در مقابل این‌همه عظمت و شکوه شما، مرا نه توان سپاس است و نه کلام وصف.

احسان روح پرور

شهریور۱۳۹۶

چکیده

در حال حاضر بیشتر مطالعات و پژوهش‌ها در زمینه بهینه‌سازی توپولوژی تحت بارهای استاتیکی می‌باشد، اگر چه در دنیای واقعی نیروها ماهیت دینامیکی دارند. اگر این سازه‌ها تحت بارهای دینامیکی قرار بگیرند تنش‌ها و تغییر شکل‌های پیش بینی نشده‌ای در آن‌ها به‌وجود خواهد آمد. در دنیای واقعی اکثر بارها به صورت دینامیکی به سازه‌ها اعمال می شوند. بنابراین لازم است تحلیل و طراحی بر مبنای بارهای دینامیکی نیز صورت گیرد. لازم است سازه‌ها تحت بارهای دینامیکی نیز بهینه‌سازی شوند تا رفتار واقعی سازه بهتر مدل شود. بارگذاری دینامیکی اثر بسیار متفاوتی بر پاسخ سازه می‌گذارد، هرچه فرکانس بار اعمالی نسبت به فرکانس سازه بیشتر باشد، بار اعمالی اثر متفاوت‌تری نسبت به حالت استاتیکی بر سازه خواهد داشت.

روش بهینه‌سازی تکاملی سازه (ESO) بر اساس یک ایده ساده استوار است که در آن با حذف تدریجی مواد ناکارآمد از دامنه طراحی به سازه بهینه دست می‌یابد. به طور کلی نتایج بدست آمده با استفاده از این روش به احتمال زیاد محلی بوده که برخلاف طرح بهینه‌ای است که از بهینگی کلی بدست می‌آید. در این تحقیق روش ژنتیک الگوریتم (GA) با روش ESO ترکیب شده و روش جدیدی به نام (GESO) بوجود می‌آید، که از رفتار عالی روش GA درجستجوی بهینگی کلی بهره می‌برد. در این روش هر المان توسط تحلیل اجزا محدود به شکل یک فرد در نظر گرفته شده و مقدار برازندگی آن با توجه به بزرگی اعداد حساسیت بدست می‌آید. سپس تمامی المان‌ها در دامنه اولیه کل جمعیت در GA را تشکیل می‌دهد. پس از چند نسل، با همگرایی المان‌های باقی مانده به سمت بهینگی نهایی شاهد شباهت بیشتری با بهینگی کلی نسبت به روش ESO خواهیم بود. این عمل با اضافه کردن عملگرهای الگوریتم ژنتیک، نظیر Mutation، Crossover به ESO انجام می‌گیرد. در واقع با این‌کار، الگوریتم تکاملی جدیدی به نام GESO به‌وجود می‌آید که قدرت آن در جستجوی پاسخ‌های کلی، به مراتب بالاتر از ESO می‌باشد.

کلمات کلیدی: GESO، بهینه‌سازی تکاملی سازه‌ها، توپولوژی، بارهای دینامیکی، ژنتیک الگوریتم

تعهدنامه

این‌جانب **احسان روح‌پرور** دانشجوی دوره کارشناسی ارشد رشته **مهندسی عمران** گرایش **سازه** دانشکده **مهندسی عمران** دانشگاه صنعتی شاهرود نویسنده پایان‌نامه **بهینه سازی توپولوژی سازه ها تحت بارهای دینامیکی با استفاده از الگوریتم GESO** تحت راهنمایی **دکتر علی کیهانی** متعهد می‌شوم.

- تحقیقات در این پایان‌نامه توسط این‌جانب انجام‌شده است و از صحت و اصالت برخوردار است.
- در استفاده از نتایج پژوهش‌های محققان دیگر به مرجع مورداستفاده استناد شده است.
- مطالب مندرج در پایان‌نامه تاکنون توسط خود یا فرد دیگری برای دریافت هیچ نوع مدرک یا امتیازی در هیچ جا ارائه نشده است.
- کلیه حقوق معنوی این اثر متعلق به دانشگاه شاهرود است و مقالات مستخرج بانام "دانشگاه صنعتی شاهرود" و یا " Shahrood University of Technology " به چاپ خواهد رسید.
- حقوق معنوی تمام افرادی که در به دست آمدن نتایج اصلی پایان‌نامه تأثیرگذار بوده‌اند در مقالات مستخرج از پایان‌نامه رعایت می‌گردد.
- در کلیه مراحل انجام این پایان‌نامه، در مواردی که از موجود زنده (یا بافت‌های آن‌ها) استفاده‌شده است ضوابط و اصول اخلاقی رعایت شده است.
- در کلیه مراحل انجام این پایان‌نامه، در مواردی که به حوزه اطلاعات شخصی افراد دسترسی یافته یا استفاده‌شده است اصل رازداری ، ضوابط و اصول اخلاق انسانی رعایت شده است.

تاریخ

امضای دانشجو

مالکیت نتایج و حق نشر

- کلیه حقوق معنوی این اثر و محصولات آن (مقالات مستخرج، کتاب، برنامه‌های رایانه‌ای، نرم افزارها و تجهیزات ساخته شده است) متعلق به دانشگاه شاهرود است. این مطلب باید به نحو مقتضی در تولیدات علمی مربوطه ذکر شود.
- استفاده از اطلاعات و نتایج موجود در پایان‌نامه بدون ذکر مرجع مجاز نمی‌باشد.

فهرست

عنوان **صفحه**

فصل اول: مقدمه و کلیات

فصل دوم: روش‌های بهینه‌سازی

فصل سوم: بهینه سازی دینامیکی با روش‌هایESO و GESO

فصل چهارم: مثال‌های عددی

فصل پنجم: نتیجه‌گیری و پیشنهادات

فهرست اشکال

فهرست جداول

فصل اول

مقدمه و کلیات

۱-۱ مقدمه

بهینه‌سازی به مفهوم کلی آن به معنی یافتن بهترین راه حل برای یک مسأله است. اما در مسائل مهندسی لازم است به شکلی دقیق‌تر بیان شود. در اینجا عبارت بهینه مشخص کننده بهترین طرحی است که کارایی و محدودیت‌های تعریف شده توسط مهندس طراح را اغنا کند. در مسائل مهندسی، به طور خاص در علم مهندسی سازه، شاخه بهینه‌سازی کاربرد فراوانی دارد. با توجه مفهوم بهینه‌سازی و تعریف سازه می‌توان بهینه‌سازی سازه را اینگونه تعریف کرد: یافتن چیدمانی که بار را به بهترین شکل ممکن تحمل و منتقل کند.

با توجه به اینکه بهینه‌سازی سازه و تحلیل و طراحی سازه ارتباط نزدیکی با یکدیگر دارند، یک سازه را برای برآوردن ملاحظات مختلفی طراحی می‌کنند که می‌توان به ملاحظات هندسی، ساخت، مکانیکی و کارکردی اشاره کرد. این درحالی است که طرح باید از لحاظ اقتصادی نیز مقرون به صرفه باشد. درحالی که وجه اقتصادی طرح متضاد با ملاحظات مدنظر می‌باشد. در نتیجه می‌توان گفت مهم‌ترین وظیفه بهینه‌سازی سازه پیدا نمودن بهترین طرحی است که هم ملاحظات بیان شده را ارضا و هم مواد مصرفی را حداقل نماید.

در روش‌های سنتی، از یک فرآیند سعی و خطا برای رسیدن به طرح بهینه استفاده می‌شود که نیازمند حدس اولیه‌ای هست که بستگی به بینش و تجربه طراح دارد. در هر مرحله با توجه به ملاحظات مقرر عملکرد سازه مورد ارزیابی قرار گرفته و طرح در فرآیند تکراری برای نزدیک کردن طرح با ملاحظات مقرر پس از هربار تجزیه و تحلیل مجدد بهبود می‌یابد. این روند تا ارضا ملاحظات توسط طرح تکرار می‌شود. این روند دارای معایبی به شرح زیر است[۱]:

۱. با توجه به اینکه راه حل واحدی برای بیشتر طرح‌ها وجود ندارد، طرح نهایی بستگی زیادی به دانش و تجربه طراح خواهد داشت.
۲. با توجه به اینکه نمی‌توان بین رایانه و طراح ارتباط خودکاری برقرار نمود، در نتیجه باعث فرآیند تعاملی ناچیز و زمان‌بری می‌شود.

بهینه‌سازی سازه برای برطرف کردن نقات ضعف مذکور توسعه پیدا نمود، که تحلیل و طراحی را با فرآیند تکرارشونده ادغام و به شکل خودکار صورت می‌پذیرد. عواملی که باعث به وقوع پیوستن این روند شده‌اند:

۱. فرمول بندی روش‌های مختلف برای مکانیزم‌های محاسباتی عموماً به‌وسیله تحلیل اجزا محدود انجام می-گیرد. که محاسبات عددی را تسهیل و سازه را با دقت قابل توجهی تحلیل می‌کند.
۲. توان رایانه‌ها به طور قابل ملاحظه‌ای افزایش پیدا کرده که باعث کاهش هزینه محاسبات تحلیل سازه شده است.

با توجه به محدودیت‌های هندسی مسائل بهینه‌سازی سازه، این مسائل را می‌توان به سه دسته بهینه‌سازی شکل، اندازه و توپولوژی دسته‌بندی کنیم. اگرچه نمی‌توان به طور کاملاً مشخص این سه دسته را از هم متمایز کنیم اما می‌توان ادعا کرد بهترین حالت بهینه‌سازی زمانی اتفاق می‌افتد که هر سه آنها به شکل همزمان ارضا شوند که به این نوع بهینه‌سازی، اصطلاح بهینه‌سازی توپولوژی گفته می‌شود. به این دلیل که در تمام نقاط سازه انرژی کرنشی یکسانی وجود دارد و در نتیجه مصالح بی‌مصرفی وجود ندارد می‌توان گفت این نوع بهینه‌سازی ایده‌آل‌ترین شکل بهینه‌سازی می‌باشد، که این به معنی استفاده بهینه از مصالح است. به همین خاطر می‌توان گفت نتیجه طراحی توپولوژی درنتیجه طراحی شکل و اندازه دارای تأثیر می‌باشد. زمانی که یک سازه الاستیک تحت بارهای دینامیکی قرار گیرد، توپولوژی سازه تأثیر به‌سزایی در پاسخ دینامیکی سازه خواهد داشت. و طراحی بهینه شکل و اندازه سازه تحت بار دینامیکی را تحت تأثیر قرار می‌دهد. بنابراین طراحی بهینه توپولوژی سازه می تواند در ایجاد یک رفتار دینامیکی مناسب مؤثر باشد[۱].

۱-۲ ضرورت تحقیق

در حال حاضر بیشتر مطالعات و پژوهش‌ها در زمینه بهینه‌سازی توپولوژی تحت بارهای استاتیکی می‌باشد، اگر چه در دنیای واقعی نیروها ماهیت دینامیکی دارند.اگر این سازه‌ها تحت بارهای دینامیکی قرار بگیرند تنش

ها و تغییر شکل‌ها پیش‌بینی نشده‌ای در آن‌ها به‌وجود خواهد آمد. در دنیای واقعی اکثر بارها به صورت دینامیکی به سازه‌ها اعمال می‌شوند. بنابراین لازم است تحلیل و طراحی بر مبنای بارهای دینامیکی نیزصورت گیرد. لازم است سازه‌ها تحت بارهای دینامیکی نیز بهینه‌سازی شوند تا رفتار واقعی سازه بهتر مدل شود. بارگذاری دینامیکی اثر بسیار متفاوتی بر پاسخ سازه می‌گذارد. هرچه فرکانس بار اعمالی نسبت به فرکانس سازه بیشتر باشد، بار اعمالی اثر متفاوت تری نسبت به حالت استاتیکی بر سازه خواهد داشت[۲].

استفاده از روش‌های ریاضی در بهینه‌سازی سازه‌ها تحت بارهای دینامیکی در مسائل با مقیاس بزرگ تقریباً غیر ممکن است. به طور کلی بهینه‌سازی سازه به وسیله‌ی تحلیل اجزای محدود استاتیکی انجام می‌شود، که از بارهای استاتیکی به عنوان ورودی استفاده می‌کند. در مسائل مهندسی بارهای استاتیکی با ضرائب دینامیکی جایگزین بارهای دینامیکی می‌شوند. به طور معمول برای بهینه‌سازی توپولوژی سازه‌ها تحت بارهای دینامیکی، روش بارهای استاتیکی معادل (ESLs) پیشنهاد می‌شود. روش بار استاتیکی معادل اولین بار در سال ۱۹۹۹ توسط چوئی و پارک[۳] پیشنهاد شد. با این روش میدان جابجایی معادل حالت بارگذاری دینامیکی را می‌توان در هر لحظه در سازه به‌وجود آورد. در واقع روش ESLs دربرگیرنده‌ی اثر بارهای خارجی و نیروهای داخلی است. به بیان دیگر ESLs پاسخ معادل بارگذاری دینامیکی را بدون معادلات پیچیده وابسته به زمان، ایجاد می‌کند. همان طور که ذکر شد از آنجایی‌که به کاربردن بارهای دینامیکی در تحلیل و طراحی خیلی سخت می‌باشد، معمولاً بارهای استاتیکی به همراه ضرایب دینامیکی مورد استفاده قرار می‌گیرد. ضرایب دینامیکی به طور کلی توسط آیین نامه‌های طراحی یا تجربه قبلی طراح تعیین می‌شوند. در نتیجه ممکن است با استفاده از بارهای استاتیکی در تحلیل و طراحی به جواب‌های دقیقی نرسیم.

در سال ۱۹۹۳ روش جدیدی برای بهینه‌سازی سازه‌ها توسط استیون و ژی[۴] ارائه گردید که بهینه‌سازی تکاملی یا ESO نامیده شد. اساس این روش حذف تدریجی المان‌های ناکارآمد (کم تنش) از سازه اولیه در یک فرآیند تکاملی است. تاکنون دیدگاه‌ها و نظریات مختلفی در مورد روش بهینه‌سازی تکاملی سازه‌ها ابراز شده است.

در این میان بعضی از پژوهشگران به انتقاد از این روش پرداخته و آن را در تولید ساختار بهینه ناکارآ معرفی می‌کنند و ادعا می‌کنند روش بهینه سازی تکاملی سازه‌ها همیشه به جواب مطلوب منجر نمی‌شود و بعضی مواقع جواب‌های فوق العاده غیر بهینه و محلی تولید می‌کند. زیرا حذف مواد بدون در نظر گرفتن پتانسیل آنها در مراحل بعد صورت می‌گیرد به عبارتی المان‌هایی که شاید نیاز داشته باشند در تکرارهای بعدی معیارهای بهینگی برای آنها کنترل گردد، دائما حذف می‌شوند.

بهینه‌سازی تکاملی دوجهتی سازه‌ها یا BESO که توسط استیون و ژی[۵] ارائه گردید شکل توسعه یافته روش ESO است که از افزودن المان در کنار حذف المان استفاده می‌کند. به بیان دیگر این روش به‌صورت اضافه‌کردن المان‌های کارآ و حذف المان‌های ناکارآ در ساختار اولیه سازه می‌باشد. برای مسائل بهینه‌سازی توپولوژی تحت بارهای دینامیکی، روش BESO فقط برپایه نتیجه تحلیل دینامیکی انجام می‌گیرد و نیازی برای تبدیل بارهای دینامیکی به بارهای استاتیکی همراه ضرایب دینامیکی در مقایسه با روش ESLs ندارد.

۱-۳ روش انجام

روشی که در این تحقیق برای بهینه‌سازی سازه‌ها تحت بارهای دینامیکی مورد بررسی قرار می‌گیرد، در سال ۲۰۰۸ توسط لیو[۶] برای مسائل استاتیکی ارائه شد که آن را GESO نامیدند. در واقع این روش از تلفیق عملگرهای الگوریتم ژنتیک و بهینه‌سازی تکاملی سازه‌ها بدست می‌آید. در این روش مفهوم اساسی بقاء بهترین در الگوریتم ژنتیک با روش ESO ترکیب شده و عملکرد این روش را در جستجوی پاسخ‌های کلی بهبود بخشیده است. در واقع با این کار، الگوریتم تکاملی جدیدی به نام GESO به‌وجود می‌آید که قدرت و سرعت آن در جستجوی پاسخ‌های کلی، به مراتب بالاتر از ESO می‌باشد. در روش GESO تمامی المان‌ها یک سازه به‌وسیله‌ی تحلیل اجزای محدود از هم گسسته شده می‌شوند، به عبارت دیگر هر المان یک فرد از جمعیت است. در روند تکاملی، المان‌های قوی‌تر که مقادیر حساسیت بزرگتری دارند حفظ می‌شوند و المان‌هایی که مقادیر حساسیت

پایین‌تری دارند حذف خواهند شد. این ایده به‌وسیله‌ی ساخت یک رشته باینری به طول n بیت با شماره ۱ انجام می‌شود. بعد از تحلیل حساسیت مقادیر هر المان محاسبه شده و عملگرهای ژنتیک مانند selection، crossover و mutation برای کروموزوم به کار گرفته می‌شود. در هر تکرار تنها المان‌هایی که تمامی ژن‌های کروموزم‌شان مقدار صفر بگیرند، حذف خواهند شد.

۱-۴ ساختار پایان‌نامه

در فصل دوم به شرح روش‌های مختلف بهینه‌سازی و بررسی تحقیقات محققین پیشین پرداخته می‌شود.

در فصل سه بررسی روش‌های تکاملی سازه‌ها و ژنتیک الگوریتم و ترکیب دو روش مذکور برای بهینه‌سازی سازه‌ها در مسائل استاتیکی و انجام می‌گیرد.

در فصل چهار از روش‌های ESO و GESO برای بهینه‌سازی توپولوژی تعدادی از سازه‌های کلاسیک دوبعدی تحت بارهای دینامیکی، استفاده می‌شود.

مهمترین کارهای انجام شده در این فصل به شرح زیر است :

۱- بررسی اثرات طول کروموزوم در روش GESO و به دست آوردن طول مناسب آن.

۲- استفاده از تکنیک هموارسازی به منظور جلوگیری از رسیدن به الگوی شطرنجی در فرآیند بهینه‌سازی.

۳- حل چندین مثال بهینه‌سازی با استفاده از روش GESO و اثبات توانایی آن در بهینه‌سازی انواع سازه‌ها.

۱-۵ بیان یک مسأله بهینه‌سازی استاندارد

بیشتر مهندسین با ایده بکارگیری مدل‌ها برای پیش‌بینی کارکرد ابزارهای فیزیکی که آزمایش آنها ممکن است وقت‌گیر و هزینه‌بر باشند، آشنایی دارند. یک مدل باید قادر به بیان نحوه تاثیر هر جزء بر عملکرد و کارایی سیستم و کنش متقابل آن بر دیگر اجزای سیستم باشد. یک مدل ریاضی مجموعه‌ای از معادلات است که سیستمی واقعی را برحسب ویژگی‌های فیزیکی، کاری و اقتصادی آن تشریح و تعریف می‌کند. یک مدل ریاضی امکان بررسی سلسله وسیعی از پارامترهای سیستم را که برای دستیابی به کارکرد و کارایی بهینه سیستم لازم‌اند، در اختیار طراح قرار می‌دهد. با توجه به مطالب گفته شده، استفاده از مدل‌های ریاضی برای طراحی بهینه مناسب به نظر می‌رسد[۷].

به طور کلی مدل ریاضی یک مسأله بهینه‌سازی به صورت زیر قابل بیان است:

$$\text{Minimize } F(x)$$

به شرط اینکه:

قیدهای نامساوی $g_i(\mathbf{x}) \le 0 \qquad i = 1.2.\ldots.n$

قیدهای مساوی $h_j(\mathbf{x}) \le 0 \qquad j = 1.2.\ldots.m$

قیدهای جانبی $x^l \le x_k \le x^u \qquad k = 1.2.\ldots.p$

با حل روابط فوق، بردار $\boldsymbol{x} = [x_1 . x_2 . \ldots . x_n]^T$ که درایه‌های آن متغیرهای طراحی هستند، بدست می‌آید. در ادامه، بخش‌های تشکیل دهنده یک مسأله استاندارد بهینه‌سازی شرح داده می‌شوند.

۱-۵-۱ متغیرهای طراحی

متغییرهای طراحی، متغیرهایی هستند که می‌توان آزادانه آنها را کنترل کرد و در بسیاری موارد توصیف کننده فیزیک و هندسه مسأله می‌باشند. مجموعه متغیرهای طراحی باید به نحوی انتخاب شوند که به‌توان به کمک آنها تمام مقادیر مؤثر دیگر بر اجرا یا کارایی سیستم را ارزیابی کرد. معمولاً در مدل ریاضی، مجموعه متغیرهای طراحی با بردار $\boldsymbol{x} = [x\,.\,x\ \ldots..x_n\,]^T$ مشخص می‌گردند که در آن n تعداد متغییرها است. هدف اصلی، دستیابی به بهترین مجموعه مقادیر ممکن در رابطه با کارایی سیستم می‌باشد. این مجموعه بهتر را متغییر بهینه نامیده و معمولاً بصورت $\boldsymbol{x}^* = [x_1^*\,.\,x_2^*\ \ldots..x_n^*\,]^T$ نشان داده می‌شود.

۱-۵-۲ تابع هدف

معیاری است برای مقایسه بین طرح‌هایی که کلیه شرایط مسأله در آنها برآورده شده است. این تابع کمیتی است که کارایی متغییرها را اندازه می‌گیرد و آن را به صورت $\mathrm{F}\,(X)$ که تابعی از متغییرهای طراحی بیان می‌کنند.

۱-۵-۳ قیود طراحی

در بسیاری از مسائل عملی نمی‌توان متغیرهای طراحی را به دلخواه انتخاب کرد، بلکه باید مقید به قیودی به نام قیدهای طراحی باشند. این قیود متضمن برآورده شدن ویژگی‌های عملی و دیگر نیازمندی‌های طرح بوده و معمولاً به صورت تابعی از متغیرهای طراحی بیان می‌گردند. این محدودیت‌ها ممکن است به صورت‌های زیر اعمال گردند:

۱- محدودیت‌های نامساوی $g_i(\mathbf{x}) \le 0 \qquad i = 1.2.\ldots n$

۲- محدودیت‌های مساوی $h_j(\mathbf{x}) \le 0 \qquad j = 1.2.\ldots m$

۳- محدودیت‌های جانبی $x^l \le x_k \le x^u \qquad k = 1.2.\ldots p$

۱-۶ روش‌های یافتن جواب بهینه

با تعریف مسأله به صورت یک مسأله استاندارد بهینه‌سازی، روش‌های متعددی برای دستیابی به بردار متغیرهای X که تمامی شرایط را ارضا کرده و کارایی سیستم را بهینه کند، وجود دارد. روش‌های به‌دست آوردن جواب بهینه را می‌توان به دو دسته کلی روش‌های بهینه‌سازی کلاسیک و روش‌های بهینه‌سازی عددی تقسیم‌بندی کرد.

برای یافتن بیشینه یا کمینه نامقید یک تابع چندمتغیره می‌توان از روش‌های حساب دیفرانسیل استفاده کرد. در این روش فرض می‌شود که تابع هدف نسبت به متغیرهای طراحی دو بار مشتق‌پذیر بوده و مشتق‌ها پیوسته باشند. عمدتاً برای حل مسائلی که دارای قید مساوی هستند از روش ضرائب لاگرانژ استفاده می‌شود. در روشهای بهینه‌سازی عددی معمولاً از یک نقطه اولیه شروع کرده و با تغییر این نقطه در تکرارهای مختلف حل بهینه بدست می‌آید. روشهای بهینه‌سازی عددی را براساس نحوه بدست آوردن نقطه جدید در هر تکرار به گروه‌های مختلفی تقسیم‌بندی می‌کنند[۷].

۱-۶-۱ روش‌های بهینه‌سازی کلاسیک

روش‌های کلاسیک بهینه‌سازی در یافتن مقدار بهینه توابع پیوسته و مشتق‌پذیر مفید هستند. این روش‌ها تحلیلی بوده و در تعیین نقاط بهینه از روش‌های حساب دیفرانسیل استفاده می‌کنند. هر چند که این روش‌ها در مسائل ساده قابل استفاده بوده و در بیان استدلال‌ها ارزش دارند، به‌کاربردن آنها در مورد توابع پیچیده اصولاً غیرعملی است. حتی اگر تابع هدف و محدودیت‌ها نسبت به متغیرهای تصمیم‌گیری به سادگی مشتق پذیر باشند، حل مسأله غالباً به حل دستگاه‌های معادلات غیرخطی منجر می‌گردد که برای حل آنها باید از روش‌های عددی استفاده کرد.

۱-۶-۲ روش‌های بهینه‌سازی عددی

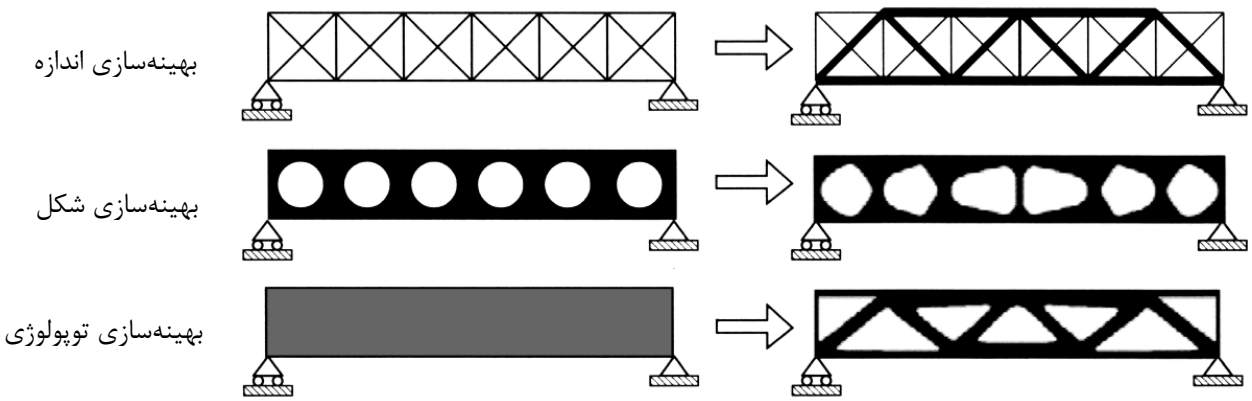
همانطور که قبلاً گفته شد چنانچه تابع هدف و قیدها توابع نسبتاً ساده‌ای از متغیرهای طراحی باشند، می‌توان در حل مسأله از روشهای بهینه‌سازی کلاسیک استفاده کرد. از طرف دیگر، چنانچه تابع هدف و یا قیدهایی وجود داشته باشند که نتوان آنها را به صورت توابع صریحی از متغیرهای طراحی بیان کرد و یا اینکه توابع به‌دست آمده بسیار پیچیده باشند، در این صورت نمی‌توان مسأله را با روشهای بهینه‌سازی کلاسیک حل نمود. بسیاری از مسائل طراحی مهندسی دارای چنین مشخصه‌هایی هستند. بنابراین در چنین حالاتی روش‌های بهینه‌سازی کلاسیک قابل استفاده نبوده و باید از روش‌های بهینه‌سازی عددی استفاده کرد.

۱-۷ شاخه‌های بهینه‌سازی سازه

بهینه‌سازی سازه‌ها را می‌توان به سه شاخه کلی دسته‌بندی کرد[۸]:

- بهینه‌سازی اندازه (سایز)[۱]
- بهینه‌سازی شکل[۲]
- بهینه‌سازی توپولوژی[۳]

این دسته‌بندی بر مبنای انتخاب متغیرهای طراحی به‌وجود آمده است. همانطور که در شکل (۲-۱) پیداست پارامتر معرف سطح مقطع یک عضو خرپا و یا تیر، یک متغیر طراحی در شاخه بهینه‌سازی اندازه است. کمیت‌های هندسی که شکل سازه را تعریف می‌کنند متغیر طراحی در شاخه بهینه‌سازی شکل هستند. و ارتباط بین المان‌ها و نیز مشخص کردن محل سوراخ‌ها (محل‌هایی که لازم نیست مصالح قرار بگیرد) در قالب متغیرهای بهینه‌سازی توپولوژی، این شاخه از بهینه‌سازی را به‌وجود می‌آورند.

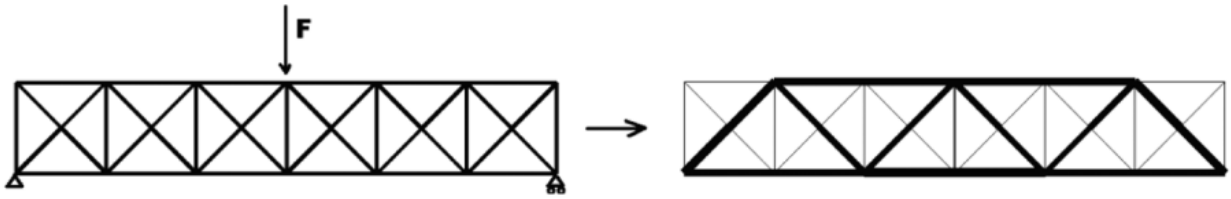


شکل (۱-۱) شاخه‌های مختلف بهینه‌سازی سازه‌ها.

[۱] Size Optimization
[۲] Shape Optimization
[۳] Topology Optimization

۱-۷-۱ بهینه‌سازی اندازه

در این شاخه از بهینه‌سازی هدف یافتن اندازه سطح مقطع یک صفحه و یا اعضای تشـکیل دهنده خرپا و یا کاربردهای مشابه می‌باشد. در این نوع بهینه‌سازی، شکل و ساختار سازه در حـین بهینه‌سازی ثابت است. بدین معنی که مدل مکانیکی سازه از ابتدا تا انتهای حل تغییر نمی‌کند.



شکل (۱-۲) بهینه‌سازی اندازه.

بنابراین مدل گسسته شـده سـازه در روش المان محـدود، اخـتلاف محـدود و یـا روش المان مرزی برای حل معادلات دیفرانسیل حاکم، نیاز به تغییر ندارد.

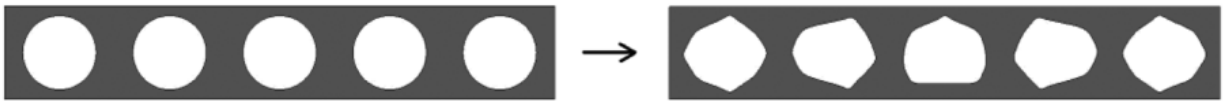
موضوع بهینه‌سازی اندازه برای سازه‌های گسسته مثل سازه‌هـای شـبکه‌ای یـا خرپایی به اندازه کافی رشد کرده و برنامه‌های مناسبی با مبنای روش المان محدود جهت حـل آنها در دسترس است. در حالیکه در مورد سیستم‌های بهینه‌سازی سازه‌های پیوسته مثل سازه‌های پوسته‌ای، پایه‌ریزی شده بر مبنای طبیعت واقعی سازه، کمتر رشد داشته است. به‌دلیل اینکه تعداد اعضای تشکیل‌دهنده سازه‌های گسسـته محـدود اسـت بنـابراین تعـداد متغیرهـای طراحی در این سازه‌ها نامنتاهی نیست. در نتیجه می‌توان به راحتی وجود جواب بهینـه را در

مـورد این سازه‌ها با معیارهای فشردگی[1] بررسی کرد و می‌توان ثابـت کـرد تخمین‌هـای خطـی متـوالی می‌تواند جواب را به یک مقدار ثابت همگرا کند.

اما در مورد سازه‌های پیوسته مثل صفحه‌هـا و پوسته‌ها، متغییر طراحی، مثلاً ضخامت صفحه، خود تابعی پیوسته از مکان است. مشـکل اساسـی در برخورد با این نوع مسائل انتخاب یک معیار همگرایی مناسب جهـت حصـول بـه جـواب بهینـه مسأله می‌باشد. یکی از برخوردهای رایج با این مشکل گسسته‌سازی سـازه جهـت محـدود کـردن متغیرهای طراحی به تعداد متناهی متغیر است. هر چند این عمل تا حدی کار را ساده می‌کنـد اما انتخاب تعداد متغیرهای طراحی خود محل سؤال است.

۱-۷-۲ بهینه‌سازی شکل

از زمانی که روش المان محدود توسط بسیاری از محققین به‌عنوان ابزاری مناسب پذیرفته شده است، شاخه بهینه‌سازی شکل رشـد به‌سـزایی داشـته اسـت. بـارزترین ویژگـی ایـن شـاخه از بهینه‌سازی این است که محدوده‌هایی که مدل مکانیکی سازه را دربر می‌گیرد متغیر طراحـی اسـت و متناوباً در روند بهینه‌سازی تغییر می‌کند. برای به‌کارگیری این شـاخه از بهینـه‌سـازی لازم اسـت توانایی تولید مدل گسسته سازه جهت انجام تحلیل در روند بهینه‌سـازی را داشـته باشـیم. شـکل (۱-۳) یک مدل ساده از این موضوع را نشان می‌دهد.



شکل (۱-۳) بهینه‌سازی شکل.

[1] Campactness

۱-۷-۳ بهینه‌سازی توپولوژی

یکی از اهداف اولیه مهندسین طراح کاهش وزن سازه‌ها بوده است. به این منظور مطالعـات و تحقیقات زیادی هم در مورد ابعاد اعضا تشکیل دهنده سازه و هم در مورد شـکل مـرز خـارجی سازه و هم شکل فضاهای خالی داخلی سازه صورت گرفته است. اما تجربه نشان داده است که اگـر چیدمان و ساختار اولیه سازه در ابتدا معلوم باشد تنها با این روش‌ها، بـه مقـدار کمـی (در حالـت خوش‌بینانه ٪۱۵) از وزن سازه کاسته می‌شود.

یکی از روش‌های پیشنهادی جهت کاهش وزن سازه، معرفی فضاهای خالی به سازه است که این فضاهای خالی قابلیت افـزایش و یـا کـاهش در رونـد بهینـه‌سـازی را دارنـد. ایـن ایـده اولیـه شکل‌گیری شاخه بهینه‌سازی توپولوژی بوده است. بهینه‌سازی توپولوژی در واقع انتخـاب همزمـان توپولـوژی (نحـوه تـوالی ارتبـاط المان‌هـای تشکیل دهنده سازه)، شکل (شکل هندسی) و اندازه اعضا تشکیل دهنده سازه می‌باشد. بهینه‌سازی توپولوژی به‌دلیل اینکه در مقایسه با دیگر روش‌های بهینه‌سازی حجم بیشتری از مصـالح را کـاهش می‌دهد از درجه اهمیت بالاتری برخوردار است.

برتری این شاخه از بهینه‌سازی نسبت به دیگر شاخه‌ها از آنجا حاصل می‌شـود کـه در ایـن مورد بسیاری از پارامترها از جمله ارتبـاط و تـوالی المان‌هـا در ابتـدای حـل مجهول‌انـد و در رونـد بهینه‌سازی مشخص می‌شوند، به عبارت دیگر تنها لازم است محدوده طراحی، شرایط مرزی و بارگذاری مشخص شود تا به‌توان توپولوژی بهینه را به‌دست آورد.

شکل (۱-۴) بهینه‌سازی توپولوژی.

فصل دوم

روش‌های بهینه‌سازی

۲-۱ مقدمه

توسعه و گسترش بهینه‌سازی سازه‌ها در سالیان گذشته منجر به معرفی روش‌ها و تئوری‌های مختلفی برای یافتن پاسخ بهینه شده است. اصول اساسی تئوری بهینه‌سازی که در قرن ۱۷ و ۱۸ میلادی پایه‌گذاری شد، طی سال‌های متمادی مورد آزمایش و بررسی قرار گرفته است. محققان بزرگی مانند اولر[۱]، گالیله[۲]، آیزاک نیوتن[۳]، ژاکوب برنولی[۴]، ویلیام روان همیلتون[۵] و لاگرانژ[۶] در این زمینه تحقیقات فراوانی انجام داده و به یافته‌های ارزشمندی رسیده‌اند. این دانشمندان هم به دیدگاه ریاضی و هم دیدگاه تحلیلی بهینه‌سازی علاقمند بوده اند. به عنوان مثال لاگرانژ اصول اساسی بهینه‌سازی عددی را پایه‌گذاری کرد که امروزه بسیار مورد استفاده قرار می‌گیرد[۸].

۲-۲ روش‌های بهینه‌سازی

برای بهینه‌سازی سازه‌ها می‌توان از روش‌های گوناگونی استفاده کرد و مجموعه بهینه متغیرهای طراحی را برای رسیدن به ساختار بهینه معین نمود. این روش‌ها به دو گروه عمده تقسیم بندی می‌شوند[۹].

۱. روش‌های گرادیان

گروه اول از مشتقات تابع هدف و قیود برای جستجو مقدار بهینه استفاده می‌کنند. در این گونه مسائل همواره فرض می‌شود که مسائل محدب هستند، مقدار مینیمم یافت می‌شود و پاسخ بهینه وجود دارد. ولی در مکانیک

---

۱ Euler
۲ Galileo Galilei
۳ Isaac Newton
۴ Jacob Bernouli
۵ William Rowar Hamilton
۶ Lagrange

سازه‌ها مسائلی وجود دارند که محدب نیستند، چون ممکن است که مسأله ناپیوسته باشد. به همین دلیل باید از روش‌های دیگری مانند روش‌های عددی استفاده نمود که مستقل از مشتقات تابع باشند.

۲. روش‌های ابتکاری

روش‌های بهینه‌سازی ابتکاری بر مبنای درک مستقیم مسأله و مشاهدات طبیعی توسعه پیدا کرده‌اند، که برپایه قوانین نسبتاً ساده و دریافت‌های مستقیم انسان ایجاد شده می‌باشند. اگر چه اینگونه روش‌ها به دلیل عدم نیاز به محاسبات پیچیده ریاضی توانایی خود را جستجوی پاسخ بهینه مناسب به اثبات رسانده‌اند، ولی نمی‌توان ادعا کرد که به پاسخ بهینه مطلق (فراگیر) خواهند رسید.

در ادامه به بررسی روش‌هایی از بهینه‌سازی سازه‌ها می‌پردازیم.

۲-۲-۱ روش همگن‌سازی

روش همگن‌سازی اجازه بهینه‌سازی همزمان ساختار، شکل و اندازه را می‌دهد. در این روش سازه به‌وسیله یک تابع فراگیر چگالی[۱] که به همه بخش‌های سازه ماده اختصاص می‌دهد، توصیف می‌گردد. سازه به وسیله المان‌های محدود نمی‌شود که هر یک از آن‌ها شامل یک ماده کامپوزیتی به همراه ریز حفره‌هایی[۲] با چگالی متغیر در بازه $\{۱و۰\}$ می‌باشند.

در عمومی‌ترین روش شکل، روش همگن‌سازی عبارت است از مینیمم کردن انرژی کرنشی سازه‌ای با حجم ثابت، که در آن متغیره طراحی چگالی ماده می‌باشد. حجم با استفاده از المان‌های محدود گسسته‌سازی شده و چگالی

[۱] Global density function
[۲] Micro voids

هر یک از المان‌ها به وسیله تعدادی متغیر مهندسی که بر ماده و ریز ساختارهای[۱] آن حاکم‌اند، کنترل می‌شود. در این روش چگالی ماده دقیقاً با خاصیت مؤثر آن مرتبط می‌گردد[۱۰].

رابطه‌سازی ریاضی روش همگن‌سازی به شکل زیر می‌باشد:

$$minimise\ \ell(u)$$

$$subject\ to\ a_e(u\,.v) = \ell(u)\ for\ all\ v \in u \qquad (۱-۲)$$

$$\ell(u) = \int_\Omega fud\Omega + \int_\Gamma tud\Gamma \qquad \text{شکل خطی نیرو} \qquad (۲-۲)$$

$a_e(u\,.v)$ شکل دو سویه[۱] انرژی است که عبارت است از کار مجازی داخلی یک سازه الاستیک در حالت تعادل u برای یک جابه‌جایی دلخواه v و برابر است با:

$$a_e(u.v) = \int_\Omega E_{ijk\ell}(D)\varepsilon_{ij}(u)\varepsilon_{k\ell}(u)d\Omega \qquad (۳-۲)$$

که در آن:

f : نیروهای حجمی

t : ترکشن‌های[۲] سطحی

ε_{ij} : کرنش‌های خطی‌سازی شده

$$\varepsilon_{ij} = \frac{1}{2}\left(\frac{\partial u_i}{\partial x_i} + \frac{\partial u_j}{\partial x_j}\right) \qquad (۴-۲)$$

U : فضای سینماتیک قابل قبول میدان‌های جابه‌جایی.

$E_{ijk\ell}$: تانسور صلبیت[۳] که به متغیرهای طراحی وابسته بوده، می‌تواند چگالی، لایه بندی و یا دوران لایه بندی هر المان باشد.

[۲] Microstructures
[۱] Bilinear

۲-۲-۲ نظریه چیدمان بهینه[۱]

اصول اساسی این تئوری هنگام بهینه‌سازی خرپاها پایه‌گذاری شد، تئوری چیدمان بهینه بر چهار اصل اساسی استوار است[۱۱]:

۱) فضای سازه‌ای که شامل مجموعه تمام اعضای ممکن و در تمام فضای در دسترس می‌باشد.

۲) لزوم وجود یک معیار بهینگی محیط پیوسته[۲] (COC) که شرط مینیمم کردن هزینه (وزن) بوده، می‌تواند بر پایه شرایط کان-تاکر شکل بگیرد.

۳) مفهوم سازه همواره[۳]، که سازه‌ای مجازی می‌باشد معیار مناسبی برای بررسی مقایسه شباهت مکانیکی کمیات خاص در معیار بهینگی می‌باشد.

۴) تابع معیار چیدمان[۴] ϕ^e که به وسیله معیارهای بهینگی تعریف شده، می‌تواند از رابطه زیر بدست آید:

$$\phi^e = 1\ (for\ A^e \neq 0)\,.\quad \phi^e \leq 1\ (for\ A^e = 0) \qquad (۵-۲)$$

که در آن:

A^e : مساحت سطح جانبی المان e ام.

عبارت بیان کننده تابع معیار ϕ معمولاً شامل کرنش ε^0 در المان e ام سازه حقیقی و کرنش $\bar{\varepsilon}^e$ در المان e سازه همراه می‌باشد. ولی ممکن است سازه بهینه‌ای که از این روش بدست می‌آید به سیستم گسسته‌ای از تعداد محدودی از اعضای خرپا محدود می‌گردد. میدان‌های کرنش حقیقی و همراه در کل فضای در دسترس روش بهینه‌سازی چیدمان شامل مراحل زیر می‌باشد:

---

[۲] Tractions
[۳] Rigidity

[۱] Optimal Layout Theory
[۲] Continuum-type Optimality Criterion
[۳] Adjoint Structure
[۴] Layout Criterion Function

۱) میدان همراه را به‌گونه‌ای پیدا کنید که:

الف) شرایط مرزی سینماتیک (تکیه‌گاه‌ها) و شرایط پیوستگی سازگاری سینماتیک برقرار باشند.

ب) تابع معیار چیدمان ϕ^e باید حداقل در یک جهت در همه نقاط فضای موجود یک مقدار واحد را اختیار کند.

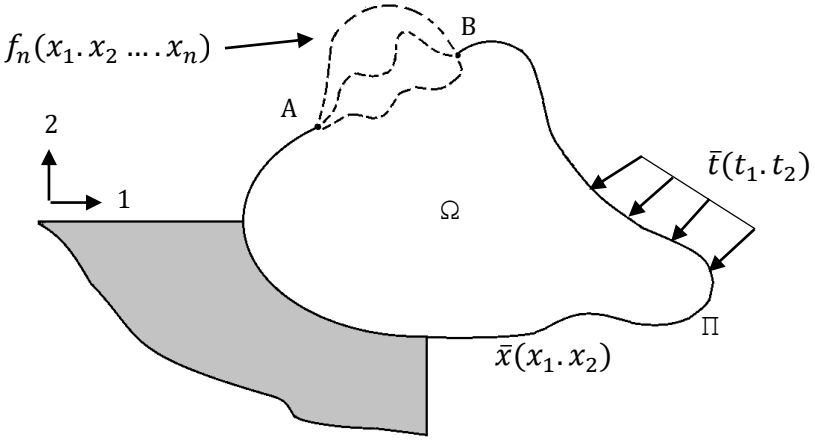
۲) سطوح جانبی اعضا در طول خطوط با $\phi = 1$ منطبق گردند، به‌گونه‌ای که سیستم حاصل قابلیت انتقال پایدار نیروهای خارجی را داشته باشد و کرنش‌های حقیقی و همراه حاصل در اعضای حذف نشده با میدان‌های کرنش تطبیق داده شده پیشین هم‌خوانی داشته باشند.

نکته قابل ذکر در مورد این روش آن است که برای برآورد نمودن همزمان شرایط فوق معمولاً نیاز به ابتکار و خلاقیت انسان به‌عنوان مغز متفکر می‌باشد.

۲-۲-۳ بهینه‌سازی شکل

در روش بهینه‌سازی شکل، یک دامنه سازه‌ای معمولاً به‌وسیله کانتورهایی تعریف می‌گردد که بصورت اسپلاین-های هندسی گسسته می‌باشند. مختصات گره‌های کنترلی این اسپلاین‌ها به‌عنوان متغیرهای طراحی در نظر گرفته می‌شوند. در این روش با تعداد کمی متغیر می‌توان شکل بهینه سازه را معین نمود. برای یک دامنه سازه‌ای مفروض، شکل (۲-۴) بخشی از مرز را می‌توان با یک تابع شکل f_S نمایش داد. اگر شکل بهینه این ناحیه از دامنه براساس یک معیار انتخاب شده باشد، مسأله را می‌توان با تابع J_S توصیف نمود:

$J_s : f_s \rightarrow C \in R$



شکل (۲-۱) ساختار سازه در محلی که قسمت $A\,B$ باید بهینه گردد.

C را می‌توان به‌عنوان مثال تنش فون میسز و یا انرژی کرنشی سازه در نظر گرفت. برای حل مسأله مینیمم‌سازی J_s و f_s بصورت تابعی از تعدادی پارامتر نامعلوم، به‌عنوان مثال متغیرهای طراحی $\bar{X} = (x_1 . x_2 . x_3 \ldots . x_n)$ تعریف می‌گردد. سپس مسأله طراحی شکل را می‌توان به‌صوت زیر به شکل یک مسأله برنامه‌ریزی غیرخطی برای تابع $F = \bar{X} \rightarrow C$ نوشت:

$$minimise\ F(\bar{X})$$

$$subject\ to\ geometric\ constraints \quad b_j = 0 .\ \ j = 1.2 \ldots . k$$

$$g_i \geq 0 .\ \ i = 1.2 \ldots . m$$

۲-۲-۴ بهینه‌سازی شکل به کمک کامپیوتر CAO

این روش برپایه مشاهده رشد طبیعی درختان شکل گرفته است. در درختان تنها خارجی‌ترین حلقه رشد خود را با بارگذاری خارجی تطبیق می‌دهد. این تطبیق به‌صورتی انجام می‌شود که توزیع تنش فون میسز ثابتی در

سطح درخت به‌وجود آید. این روش تلاش می‌کند تا از ایجاد قله‌های تنش محلی جلوگیری کرده و طراحی با وزن حداقل ایجاد کند. این مکانیزم رشد تبدیل به یک روش بهینه‌سازی شده است که از افزایش رشد بیولوژیک پیروی کرده و به کمک نرم افزارهای اجزای محدود پیاده‌سازی می‌شود[۱۲].

روش CAO شامل مراحل زیر است:

۱) تولید شکل اجزا محدود اولیه سازه.

۲) ایجاد پوششی از یک لایه نازک اجزای محدود روی تقسیم بندی اولیه، که این لایه نازک ثانویه دارای مدول الاستیک کوچکتری نسبت به بقیه ماده تشکیل دهنده‌ی سازه داشته باشد. این لایه معادل حلقه رشد درختان می‌باشد.

۳) اعمال نیروها روی سازه باعث ایجاد نواحی با تنش بالا و نواحی با تنش پایین می‌گردد. با استفاده از معادله زیر و فرض لایه سطحی به‌عنوان لایه قابل جابه‌جایی با قابلیت رشد یا کوچک شدن خواهیم داشت:

$$\varepsilon_v = k(\sigma_{vonmises} - \sigma_{ref}) \qquad (۶-۲)$$

که در آن:

$\sigma_{vonmises}$: توزیع تنش فون میسز محاسبه شده با FEA

σ_{ref}: مقدار مرجع تنش

رشد یا کوچک شدن لایه سطحی مطابق معادله (۲-۶) باعث ایجاد توزیعی از جابه‌جایی در سطح می‌گردد. همچنین می‌توان به لایه سطحی اجازه کوچک شدن ($\varepsilon_v < 0$) داد یا از ایجاد آن جلوگیری نمود.

۴) مرز داخلی لایه سطحی به بیرون منتقل می‌گردد، به گونه‌ای که ضخامت آن در تمام مدت ثابت بماند.

۵) مراحل ۳ تا ۵ روی شکل اصلاح شده مرحله ۴ تکرار می‌شوند.

۶) تحلیل اجزای محدود با یک مدول الاستیک در تمام سازه انجام می‌شود تا وجود یا عدم وجود قله‌های تنش در سازه را مشخص کند.

۷) مراحل ۳ تا ۶ تا زمانی که تمامی قله‌های تنش بطور کامل حذف شوند و یا محدودیت‌های طراحی از افزایش بیشتر ابعاد سازه جلوگیری کنند، ادامه می‌یابند.

بطور خلاصه این روش باعث افزودن ماده به نواحی پر تنش و حذف آن از نواحی کم تنش می‌گردد.

۲-۲-۵ بهینه‌سازی ساختاری[۱]

بهینه‌سازی ساختاری ابزاری است که به اطلاع در انتخاب ساختارهای اولیه سازه کمک می‌کند. هدف بهینه-سازی ساختاری حذف یا باز توزیع ماده در دامنه اولیه سازه و رسیدن به یک ساختار بهینه می‌باشد.

این روش، یک روش تکرارشدنی بوده، معمولاً همراه با تاریخچه‌ای از ساختارهای میانی می‌باشد که آن‌ها نیز قابل استفاده در فرآیند طراحی می‌باشند[۱۳].

تا کنون چهار الگوریتم اصلی برای بهینه‌سازی ساختاری سازه‌ها ارائه شده‌اند:

- روش حذف سخت/ نرم[۲]
- روش همگن‌سازی
- دیدگاه سازه مبنا[۳]
- روش حباب[۴]

---

۱ Topology Optimization
۲ Soft Kill/Hard Kill Method
۳ Ground Structure Approach
۴ Bubble Method

سه روش آخر همگی دارای تابع هدف، متغیرهای طراحی و قیود بوده و مسأله بهینه‌سازی را با استفاده از روش‌های مانند برنامه‌ریزی ترتیبی درجه دو حل می‌کنند. این سه روش نسبتاً پیچیده می‌باشند. برخلاف آنها روش حذف نرم/ حذف سخت بر اصول بسیار ساده‌ای استوار است. این روش در حالت عمومی یک الگوریتم بهینه‌سازی نمی‌باشد، ولی از حذف ماده ناکارآ برای بهینه‌سازی ساختار باقیمانده سازه استفاده می‌کند.

روش بهینه‌سازی تکاملی سازه‌ها[۱] (ESO) و مشتقات آن یعنی بهینه‌سازی تکاملی با راهبرد افزونی[۲] (AESO) و بهینه‌سازی تکامل دوجهتی[۳] (BESO) در این گروه از روش‌های بهینه‌سازی قرار می‌گیرند.

۲-۲-۶ بهینه‌سازی تکاملی سازه‌ها

روش ESO و BESO که از مشتقات آن می‌باشد بین روش‌های ابتکاری و روش‌های مبتنی بر گرادیان قرار می‌گیرد. این بدان معناست که ESO می‌تواند با جستجو در فضای طراحی هم به مینیمم‌های محلی و هم به مینیمم‌های مطلق دست می‌یابد. ولی به‌دلیل ویژگی تکاملی آن، در برخورد با یک نقطه مینیمم محلی توقف نخواهد کرد، بلکه به جستجوی خود برای رسیدن به پاسخ بهتر ادامه می‌دهد. روش بهینه‌سازی تکاملی سازه‌ها (ESO) برای اولین بار توسط ژی و استیون ارائه شد[۴]. ایده اساسی این روش بر مفهوم ساده و تجربی استوار است که بیان می‌کند یک سازه با حذف تدریجی (hard-killing) المان‌های با تنش پایین، به سمت بهینگی تکامل پیدا کند. برای ماکزیمم کردن سختی سازه، معیار انرژی کرنشی عناصر به‌جای معیار تنش جایگزین می‌شود، که توسط چو، ژی و استیون در سال ۱۹۹۶ ارائه شد[۱۴]. بهینه‌سازی تکاملی سازه‌ها با رویکرد دوجهته (BESO) با این ایده که اجازه اضافه شدن المان‌های جدید در اطراف المان‌هایی که حساسیت بالایی دارند، توسعه پیدا نمود.

---

۱ Evolutionary Structural Optimization
۲ Additive Evolutionary Structural Optimization
۳ Bidirectional Evolutionary Structural Optimization

برای مسائل بهینه‌سازی سختی که از معیار انرژی کرنشی استفاده می‌کنند، انرژی کرنشی المان‌های خالی به‌وسیله برونیابی خطی میدان‌های جابه‌جایی تخمین زده می‌شوند. از روش‌های ESO و BESO در نشریه‌ها و کاربردهای متعددی توسط محققین در سراسر جهان به‌کار گرفته شده است[۱۵].

۲-۲-۶-۱ ساختار مسأله

انرژی پتانسیل برای یک حجم الاستیک خطی در مکانیک محیط‌های پیوسته با میدان جابجایی مجاز به شکل زیر بیان می‌شود:

$$\Pi = \int \frac{1}{2}\varepsilon^T \boldsymbol{E}\varepsilon \, dV - \int \boldsymbol{u}^T \boldsymbol{F} dV - \int \boldsymbol{u}^T \boldsymbol{T} dS - \boldsymbol{D}^T \boldsymbol{Q} \qquad (۷-۲)$$

که در آن

ϵ : میدان کرنش

$\boldsymbol{E}$: ماتریس مشخصات مواد

$\boldsymbol{u}$: میدان جابجایی

$\boldsymbol{F}$: نیروهای حجم

$\boldsymbol{T}$: ترکشن‌های سطحی

$\boldsymbol{D}$: جابجایی گسسته سازه

$\boldsymbol{Q}$: بارهای خارجی گسسته

S . V : ناحیه‌ای از سطح و حجم سازه

با استفاده از روند المان محدود، معادله (۲-۷) به شکل زیر تغییر می‌یابد:

$$\prod = \frac{1}{2}\boldsymbol{D}^T\boldsymbol{K}\boldsymbol{D} - \boldsymbol{D}^T\boldsymbol{P} \qquad (۲-۸)$$

که $\boldsymbol{K}$ ماتریس سختی کل، $\boldsymbol{P}$ ماتریس بار، که شامل بارهای گرهی و نیروهای حجمی و ترکشن‌های سطحی:

$$\boldsymbol{P} = \boldsymbol{Q} + \boldsymbol{R} \qquad (۲-۹)$$

که $\boldsymbol{R}$ برابر است با:

$$\boldsymbol{R} = \sum_{i=1}^{i=n} \boldsymbol{r}_e^i \qquad (۲-۱۰)$$

$$\mathbf{r}_e = \int_{V_e} \boldsymbol{N}^T \boldsymbol{F} dV - \int_{S_e} \boldsymbol{N}^T \boldsymbol{T} dS \qquad (۲-۱۱)$$

که در آن $\boldsymbol{N}$ ماتریس تابع شکل، V_e و S_e حجم و محیط المان و n تمام المان‌های سازه می‌باشند.

معادله تعادل استاتیکی برابر است با:

$$\boldsymbol{K}\boldsymbol{u} = \boldsymbol{P} \quad \text{یا} \quad \boldsymbol{K}\boldsymbol{D} = \boldsymbol{P} \qquad (۲-۱۲)$$

سختی کلی یک سازه را می‌توان به‌وسیله انرژی کرنشی بدست آورد. ماکزیمم کردن سختی سازه برابر است با مینیمم کردن انرژی کرنشی. رابطه‌سازی ریاضی مسأله بهینه‌سازی سختی را می‌توان به شکل زیر ارائه کرد:

$$Minimise \quad f = C(x) = \boldsymbol{P}^T\boldsymbol{u} \qquad (۲-۱۳)$$

$$Subject\ to \quad g = W^* - \sum_{i=1}^{n} W_i\, x_i = 0 \qquad (۲-۱۴)$$

$$x_i \in \{0,1\}$$

که در آن W_i وزن یک المان و W^* وزن کل سازه را نشان می‌دهد. متغیر طراحی باینری $\{0,1\}$ عدم حضور یا حضور یک المان را بیان می‌کند.

توجه شود که:

$$\boldsymbol{P^T u = u^T K u} \qquad (۱۵-۲)$$

انرژی کرنشی را می‌توان به‌وسیله مشارکت هر المان به شکل زیر بیان کرد:

$$c = \boldsymbol{u^T K u} = \sum_{i=1}^{n} \left(\boldsymbol{u_i^T K_i u_i}\right) = \sum_{i=1}^{n} 2C_i \qquad (۱۶-۲)$$

که $\boldsymbol{K_i}$ و $\boldsymbol{u_i}$ ماتریس سختی و بردار جابجایی المان i ام هستند. و $C_i = \frac{1}{2}\left(\boldsymbol{u_i^T K_i u_i}\right)$ انرژی کرنشی المان می‌باشد.

۲-۲-۶-۲ تحلیل حساسیت

در تحلیل حساسیت، حساسیت المان با توجه به گرادیان تابع هدف بدست می‌آید. همان طور که ذکر شد انرژی کرنشی بصورت معادله زیر بدست آمد:

$$C = \boldsymbol{P^T u} \qquad (۱۷-۲)$$

با فرض اینکه تغییراتی در سازه ایجاد شود، مقدار تغییرات انرژی کرنشی برابر است با:

$$\Delta C = \boldsymbol{P^T \Delta u + \Delta P^T u} \qquad (۱۸-۲)$$

چنانچه فرض کنیم که تغییرات سازه تأثیری در شرایط بارگذاری ندارد یا به عبارت دیگر $\boldsymbol{\Delta P = \{0\} z}$ ، خواهیم داشت:

$$\Delta C = \boldsymbol{P^T \Delta u} \qquad (۱۹-۲)$$

با فرض $\Delta \boldsymbol{P} = \{\boldsymbol{0}\}$ آنگاه معادله تعادل به شکل زیر نمایش داده می‌شود:

$$\Delta \boldsymbol{K}\boldsymbol{u} + \boldsymbol{K}\Delta \boldsymbol{u} = \boldsymbol{0} \qquad (۲۰-۲)$$

با ضرب معادله بالا در $\boldsymbol{P}^T \boldsymbol{K}^{-1}$ (یا $\boldsymbol{u}^T$) خواهیم داشت:

$$\boldsymbol{P}^T \boldsymbol{K}^{-1} \boldsymbol{K} \Delta \boldsymbol{u} = -\boldsymbol{u}^T \Delta \boldsymbol{K} \boldsymbol{u} \qquad (۲۱-۲)$$

در نتیجه مشتق رابطه برابر می‌شود با:

$$\Delta C = \boldsymbol{P}^{\boldsymbol{T}} \Delta \boldsymbol{u} = -\boldsymbol{u}^T \Delta \boldsymbol{K} \boldsymbol{u} \qquad (۲۲-۲)$$

تغییرات ناشی از حذف یک المان یا به عبارتی $\Delta \boldsymbol{K} = -\boldsymbol{K_i}$ ، باعث تغییر معادله به شکل زیر می‌شود:

$$\Delta C = \boldsymbol{u_i^T K_i u_i} = 2C_i \qquad (۲۳-۲)$$

در نتیجه می‌توان حساسیت هر المان را از رابطه $C_i = \frac{1}{2} \boldsymbol{u_i^T K_i u_i}$ محاسبه نمود.

۲-۲-۶-۳ روند بهینه‌سازی با روش ESO

به طور کلی روند تکامل در این روش شامل دو مرحله اصلی، که عبارت است از تحلیل سازه و اصلاح سازه است که به شیوه‌ی تکراری انجام می‌گیرد. اصلاح سازه به وسیله تحلیل حساسیت و حذف المان‌ها صورت می‌پذیرد.

۱- برای شرایط مرزی و بارگذاری مشخص، دامنه طراحی را مشخص می‌کنیم.

۲- تحلیل المان محدود برای بدست آوردن تغییر مکان‌ها را انجام می‌دهیم.

۳- عدد حساسیت را برای تمامی المان‌ها محاسبه می‌کنیم.

۴- حذف المان‌هایی که کمترین حساسیت را دارا می‌باشند.

۵- مراحل ۲ تا ۴ را تا زمانی که سازه به وزن تعیین شده برسد، تکرار می‌کنیم.

در این روند از یک طرح اولیه که شامل تمامی المان‌ها است، برای شروع بهینه‌سازی استفاده می‌شود. هیچ نود جدید در این روند تکاملی اضافه نمی‌شود و تنها نیاز به یک مش‌بندی اولیه می‌باشد. در تحلیل حساسیت فرض بر این بوده است که تغییرات سازه و در نتیجه جواب کوچک است، بنابراین تنها درصد اندکی از المان‌ها در هر تکرار باید حذف شوند، این میزان به وسیله پارامتری به نام " نسبت تغییر " (MR) تعیین می‌شود. برای مثال چنانچه ٪۱ = MR باشد و طرح جاری دربردارنده‌ی ۲۰۰۰ المان باشد، آنگاه ۲۰ = ۰,۰۱×۲۰۰۰ المان در این تکرار حذف خواهند شد[۱۶].

۲-۲-۶-۴ بهینه‌سازی تکاملی سازه با راهبردهای افزودنی و دوجهتی BESO

یک از ایراداتی که به روش ESO کلاسیک وارد می‌شود این است که در این روش اگر المانی حذف شود، قابلیت بازگشت مجدد به ساختار سازه را نخواهد داشت. در حالی که ممکن است المان حذف شده در مراحل بعد دوباره مورد نیاز باشد. به منظور برطرف نمودن این اشکال، روش جدیدی به نام ESO دوجهتی ارائه گردید. در این روش اجازه داده می‌شود که المان‌های حذف شده در صورت نیاز بازگردند. ESO دو جهتی یا BESO همچنین می‌تواند به منظور بررسی اینکه آیا ESO کلاسیک المان‌ها را پیش از موعد حذف می‌کند یا نه به کار گرفته می‌شود. قانونی در روش ESO کلاسیک وجود دارد که باید از آن پیروی شود، این قانون بیان می‌کند که همواره دامنه، فضا یا حجم اولیه را بیش از ابعاد مورد نیاز انتخاب کرد، تا به ESO اجازه داده شود که المان‌هایی را که باید حذف شوند خود مشخص کند[۱۷].

اثبات شده است که اگر طراح حداکثر فضای طراحی ممکن را مشخص نکند، ESO به ساختار بهینه نخواهد رسید. پرسشی که اینجا مطرح می‌شود این است که آیا می‌توان از اصل تکاملی ESO برای بازگرداندن المان‌ها به سازه استفاده نمود. در این حالت به‌جای شروع فرآیند تکاملی با حداکثر ممکن تعداد المان‌ها، از حداقل مقدار ماده برای اتصال تکیه‌گاه‌ها به نیروها استفاده می‌شود که خود المان‌های مورد نیاز را ایجاد کند. به عبارت دیگر این روش عبارت است از افزودن ماده به نقاط مورد نیاز و حذف آن از نقاطی که به ماده نیاز ندارد. در نتیجه به روش ESO اصلاح شده‌ای خواهیم رسید که مستقل از تقسیم بندی اولیه بوده و به سوی پاسخ بهینه میل می‌کند و به حدس اولیه طراح بستگی ندارد[۱۸].

۲-۲-۶-۵ روند بهینه‌سازی با روش BESO

در این روند علاوه بر حذف المان‌ها امکان اضافه شدن المان‌ها نیز میسر شده است. که باعث بازگشت المان‌های کارآمدی که در تکرارهای قبل حذف شده‌اند و کاهش زمان محاسبات شده است.

۱- تعیین دامنه فیزیکی اولیه برای آغاز فرآیند، که حداقل تعداد المان‌هایی است که نیروها را به تکیه‌گاه‌ها متصل می‌کند.
۲- ویژگی مشخصه همه المان‌های سازه که در دامنه فیزیکی سازه قرار ندارند، صفر منظور گردد.
۳- تحلیل اجزای محدود سازه.
۴- محاسبه عدد حساسیت.
۵- حذف المان‌هایی که دارای کمترین مقدار حساسیت هستند، و همزمان اضافه کردن المان‌هایی که دارای بیشترین مقدار حساسیت در اطراف المان‌های موجود در دامنه فیزیکی سازه.
۶- تکرار گام‌های ۲ تا ۵ تا زمانی که به وزن تعیین شده برای سازه برسیم.

در این روش دو پارامتر نسبت افزودن (AR) و نسبت پایه (SR) مورد استفاده قرار می‌گیرد. به طور مثال چنانچه قرار باشد که در هر تکرار ۲۰ المان مورد اصلاح قرار بگیرند و $AR = ۰,۶$ باشد، ۶ = ۰,۶×۲۰ تا المان اضاف خواهد شد و ۱۴ المان از دامنه طراحی حذف خواهد شد. بدیهی هست که اگر $AR < ۰,۵$ باشد از حجم سازه کاسته می‌شود و اگر $AR > ۰,۵$ به حجم سازه افزوده خواد شد[۱۵].

۲-۲-۷ الگوریتم ژنتیک

الگوریتم ژنتیک روش یادگیری بر پایه تکامل بیولوژیک است و در سال ۱۹۷۵ توسط جان هلند[۱۹] معرفی گردید. الگوریتم ژنتیک برای حل مسأله، مجموعه بسیار وسیعی از حل‌های ممکن را تولید می‌کند. هر یک از این حل‌ها با استفاده از یک تابع برازش مورد ارزیابی قرار می‌گیرد. آنگاه تعدادی از بهترین پاسخ‌ها باعث تولید پاسخ‌های جدیدی می‌شوند که این کار باعث تکامل پاسخ‌ها می‌گردد. بدین ترتیب فضای جستجو در جهتی تکامل پیدا می‌کند که به حل مطلوب برسد. در صورت انتخاب صحیح پارامترها، این روش می‌تواند بسیار مؤثر عمل کند.

۲-۲-۷-۱ بررسی الگوریتم ژنتیک

در بین روش‌های بهینه‌سازی، روش الگوریتم ژنتیک یا بهینه‌سازی الهام گرفته از طبیعت، از تکامل یافته‌ترین روش‌ها با متغیرهای گسسته به شمار می‌رود. این الگوریتم براساس اصول تکامل طبیعی پایه‌گذاری شده است. هرجا انجام کاری با قدرت لازم است، طبیعت آن را بهتر انجام می‌دهد و رازهای سازش‌پذیری و بقا، بهترین آموزنده‌ای است که می‌توان از مطالعه دقیق مثال‌های زیست شناسی به آن رسید. در طبیعت افرادی هستند که در رقابت برای دستیابی به منابع محدودی مانند غذا و سرپناه پیروز شده و باقی می‌مانند، برتری این افراد مدیون ویژگی‌های فردی آن‌هاست. تولید مثل افراد پیروز موجوب تولید فرزندان بهتر می‌شود. با انجام متوالی انتخاب بهترین افراد جمعیت و تولید مثل آن‌ها، کل جمعیت به سوی سازش بیشتر با محیط خود، یعنی دستیابی به

منابع بهتر و بیشتر سوق پیدا می‌کند. الگوریتم ژنتیک، از الگوریتم‌های جستجویی هستند که بر روند طبیعی و ژنتیک‌های طبیعی استوارند، این الگوریتم ها مناسب‌ترین رشته‌ها را از میان اطلاعات تصادفی سازمان‌دهی شده با روش جستجوی انسانی انتخاب می‌کنند. در هر نسل یک مجموعه جدید از رشته‌های مصنوعی با استفاده از بیت‌ها آفریده می‌شوند، این قسمت جدید تصادفی انتخاب می‌شود و میزان قدرت یا برازندگی آن به دست می‌آید.. در واقع در یک الگوریتم ژنتیک برای ایجاد نسل جدید یک روند اتفاقی ساده وجود ندارد، بلکه آن‌ها داده‌های قبلی را همراه با نقاط جستجوی جدید، برای رسیدن به پیشرفت مورد نظر توأم می‌کنند[۱۹].

اصول اولیه این الگوریتم توسط هلند[۱۹] در دانشگاه میشیگان ایالات متحده در سال ۱۹۷۵ ارائه شد. از اوایل دهه‌ی ۱۹۸۰ به بعد مقاله‌ها و رساله‌های بسیاری در تأیید بهینه‌سازی توابع، توسط این الگوریتم انتشار یافته است. در ادامه به معرفی بیشتر ژنتیک و همچنین ساختار آن، پارامترهای الگوریتم و چگونگی انتخاب آن‌ها پرداخته می‌شود.

۲-۲-۷-۲ ژنتیک در طبیعت

فرض کنید در ساختار طبیعی، حالت مطلوب رسیدن به فردی با خصوصیات مطلوب می‌باشد. وظیفه ما رسیدن به شخص مورد نظر می‌باشد، برای دستیابی به این منظور کارهای زیر طبق روند طبیعی انجام می‌شود. ابتدا یک جمعیت اولیه در نظر می‌گیریم، مثلاً ۱۰۰ نفر، حال در جمعیت اولیه حالت مطلوب یعنی شخصی با خصوصیات ذکر شده جستجو می‌شود، اگر پیدا شود که کار پایان می‌یابد، در غیر این صورت روند کار ادامه داده می‌شود. برای این کار باید جمعیت تغییر داده شود و یک جمعیت جدید تولید شود و دوباره حالت مطلوب جستجو شود، برای تولید جمعیت، سه عمل انتخاب، تزویج و جهش را انجام می‌دهیم. تمامی این اعمال در طبیعت هم وجود دارند. برای انتخاب سعی می‌شود که به تعداد جمعیت اولیه یک جمعیت ثانویه تولید شود، خصوصیات این جمعیت

ثانویه به گونه‌ای است که افراد برازنده‌تر به تعداد بیشتری وجود دارند، و برعکس آن افرادی که برازندگی کمتری دارند به تعداد کمتر و افراد دارای برازندگی کم انتخاب نمی‌شوند[۲۰].

البته در طبیعت نیز همین‌گونه است یعنی موجودات نخبه‌تر یا برازنده‌تر احتمال زنده ماندن و تولید نسل جدید بیشتری نسبت به موجودات دیگر دارند، در اصل به افراد برازنده‌تر، احتمال بیشتری برای انتخاب داده می‌شود نه اینکه حتماً آن‌ها انتخاب می‌شوند و همین‌گونه برای افراد با برازندگی کم احتمال کمتر برای انتخاب و شرکت در نسل بعد داده می‌شود نه اینکه حتماً انتخاب نمی‌شوند.

سوالی که پیش می‌آید این است که برازنده‌تر بودن به چه معناست. برای پاسخ به این سوال باید به خصوصیات کروموزم هر فرد توجه شود، هر فردی دارای چند ژن می‌باشد و به مجموع ژن‌های یک فرد کروموزم گفته می‌شود. برای پی بردن به خصوصیات کروموزمی هر فرد باید ژن‌های آن شناخته شود، چنانچه در یک مسأله تمامی اشخاص دارای سه ژن باشند، به جستجوی شخصی که سه ژن آن مانند شخص مطلوب است، پرداخته می‌شود. حال با دانستن کروموزوم و خصوصیات هر ژن، برازندگی و نخبه بودن هر شخص تعریف می‌شود؛ با فرض اینکه هیچ کدام از سه ژن نسبت به یکدیگر برازندگی ندارند. در نتیجه شخصی که سه ژن آن مانند شخص ایده‌آل باشد از همه برازنده‌تر است و شخصی که دو ژن آن مانند شخص ایده‌آل باشد، دارای برازندگی کمتر و به همین ترتیب. در جمعیت جدید، توزیع تعداد افراد برازنده‌تر در آن نسبت به جمعیت قبل بیشتر است.

در حالت ریاضی، طبق معلومات مسأله تابعی تعریف می‌شود و تمامی ژن‌ها روی آن اثر داده می‌شود و مقدار این تابع معیاری برای برازنده‌تر بودن است. حال با دو عملکرد تزویج وجهش که در طبیعت هم وجود دارند می‌توان به جمعیت جدید رسید، عملکرد تزویج به این صورت است که جمعیت دو به دو با یکدیگر تزویج داده می‌شود تا به ازای هر پدر و مادر، دو فرزند جدید به وجود آید، به طور مثال چنانچه تعداد کل جمعیت ۱۰۰۰ نفر باشد،

۵۰۰ زوج را شامل می‌شوند که بعد از تزویج این ۵۰۰ زوج با هم، ۱۰۰۰ فرزند جدید به‌وجود می‌آورند، و این فرزندان در نسل بعد نقش پدر و مادر را دارند.

دلیل تزویج افراد به زبان ساده این است که زوج‌ها بتوانند کمبودهایشان را از طریق جایگزین کردن قسمتی از کروموزوم‌هایشان با یکدیگر جبران کنند و به حالت مطلوب نزدیکتر شوند. که البته این هم یک احتمال است و ممکن است اصلاً هر دو شخص از حالت اولیه خود، بدتر هم بشوند، ولی چیزی که مشخص است این است که روند طبیعت هیچ‌گاه اشتباه نمی‌کند زیرا این الگوریتم بر مبنای اصول طبیعی پایه‌ریزی شده است. برای توافق بیشتر این الگوریتم با طبیعت سعی می‌شود که درصدی از جمعیت تزویج داده نشود، مثلاً ۸۰ درصد جمعیت با یکدیگر تزویج داده می‌شود، همان طور که در طبیعت هم تقریباً همین‌گونه است.

بعد از تولید چند نسل جمعیت به سوی یک مقدار همگرا پیش می‌رود، یعنی اکثر کروموزوم‌های افراد مثل هم می‌شوند، در اینجا یک مشکل پیش می‌آید زیرا وقتی دو شخص کروموزوم‌های شبیه به هم داشته باشند، در نتیجه با تزویج با یکدیگر به دو شخص جدید تبدیل نمی‌شوند، زیرا کمبود یک شخص در شخص دیگر وجود ندارد و به نوعی کمبودها و دارایی‌های هر زوج یکسان است. در این حالت جمعیت به اصطلاح در یک مقدار بهینه محلی گیر می‌کند، یعنی در تمامی جمعیت، تعدادی از ژن‌ها کاملاً بهینه شده‌اند ولی در بهترین حالت خود قرار ندارند. البته توجه داریم که رسیدن به این حالت نیز یک احتمال است و ممکن است اصلاً تمامی جمعیت بدون گیر کردن در بهینه‌های محلی به سوی مقدار ایده‌ال و مطلوب پیش رود، ولی در هر صورت برای توافق بیشتر با طبیعت و کاهش احتمال گیر کردن در مقادیر بهینه محلی از عملگری به‌عنوان جهش استفاده می‌شود. عملگر جهش بر روی بعضی از کروموزوم‌ها عمل می‌کند و خصوصیات آن‌ها را تغییر می‌دهد و بدین وسیله مشکل احتمالی ذکر شده برطرف می‌شود. بعد از عمل جهش جمعیت جدید به‌وجود می‌آید که به احتمال زیاد از جمعیت پیشین خود یعنی از پدر و مادرهای خود بهینه‌تر هستند و این روند ادامه پیدا می‌کند تا به شخص مطلوب و بهینه رسیده شود[۲۰].

۲-۲-۷-۳ شبیه‌سازی پارامترهای ژنتیک به شکل ریاضی

برای حل مسائل، باید روند ژنتیک طبیعی را به شکل ریاضی نوشت. در ادامه به شبیه‌سازی عواملی چون جمعیت اولیه، تزویج و جهش پرداخته می‌شود. جمعیت اولیه به صورت یک ماتریس دوبعدی تعریف می‌شود که بیت‌های آن از صفر و یک‌های تصادفی پر شده است.

به هر سطر آن یک کروموزوم گفته می‌شود و هر کروموزوم از چند ژن درست شده است، در واقع ژن‌ها همان متغیرها می‌باشند. برای بدست آوردن طول هر ژن یا متغیر مورد نیاز به‌صورت تعدادی بیت، در ماتریس جمعیت به صورت زیر عمل می‌شود:

فرض کنید متغیرها X باشد، و $a < X <$ b ، در نتیجه:

$$X = \text{a} + (\text{عدد کدگشایی شده در } n \text{ بیت}) \times \left(\frac{b-a}{2^n-1}\right) \quad (۲۴-۲)$$

در نگاشت بالا n تعداد بیت‌های مورد نیاز برای هر ژن می‌باشد که از فرمول زیر بدست می‌آید:

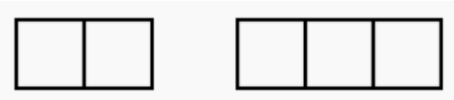
$$(b-a) \times \left(\frac{1}{p}\right) \leq 2^n \quad (۲۵-۲)$$

که در آن p دقت اعداد بین دو عدد صحیح است. حال با داشتن تعداد متغیر و طول هر کدام از آن‌ها طول هر کروموزوم به دست آورده می‌شود اگر فرض شود که طول این ماتریس جمعیت m و عرض آن n باشد، در نتیجه m طول هز کروموزوم و n تعداد کروموزوم‌ها می‌باشد.

حال هر کروموزوم به دیده یک شخص است و با این دید به تزویج اشخاص پرداخته می‌شود، تزویج به این صورت است که دو به دو بیت‌های کروموزوم‌ها از نقطه‌ای دلخواه به بعد با هم عوض می‌شوند، آن نقطه دلخواه به صورت تصادفی تعیین می‌شود. عملگر جهش به این صورت است که، در نقاطی تصادفی در ماتریس جمعیت، یک را به صفر تبدیل می‌کند.

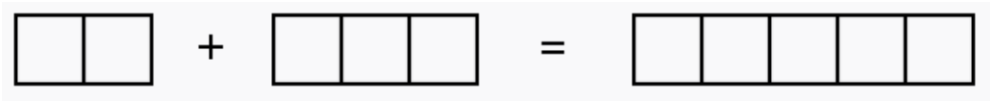
۲-۲-۷-۴ اصطلاحات به کار گرفته شده در ژنتیک

۱- ژن: به چند بیت اختصاصی برای هر متغیر ژن گفته می‌شود. در شکل (۲-۲) چند ژن با طول‌های متفاوتی قرار دارند، گفتنی است که هر ژن مربوط به یک متغیر است و طول این ژن ها به بازه تغییرات متغیر و به دقت اعداد موجود در آن بستگی دارد[۲۱].



شکل (۲-۲) چند ژن با طول‌های متفاوت

۲- کروموزوم: به چند ژن که خصوصیات یک چیز را می‌رساند، کروموزوم گفته می‌شود، در واقع از کنار هم قرار گرفتن ژن‌های یک مسأله، یک کروموزوم به‌وجود می‌آید که طول آن با مجموع طول‌های ژن‌ها برابر است.



شکل (۲-۳) تشکیل کروموزم‌ها از ژن‌ها

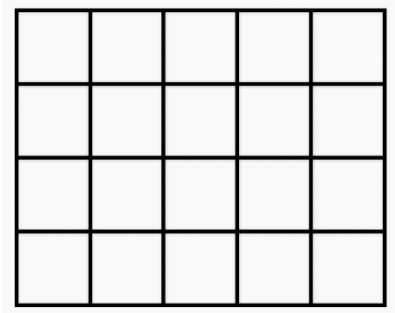
باید توجه داشت که در یک مسأله خاص طول تمامی کروموزوم‌ها باید برابر باشد ولی در مورد ژن‌ها اینگونه نیست.

۳- طول هر ژن: که به تعداد متغیر و بازه‌ی تغییرات متغیر و دقت آن بستگی دارد.

۴- طول هر کروموزوم: که به تعداد متغیر و بازه‌ی تغییرات متغیر و دقت آن بستگی دارد.

۵- تعداد کروموزوم: تعداد کروموزوم که به مسأله بستگی دارد.

۶- جمعیت: به تعدادی کروموزوم که در کنار هم قرار می‌گیرند جمعیت گفته می‌شود.



شکل (۲-۴) جمعیت به‌وجود آمده از چند کروموزوم

۷- جهش: تغییر یک کروموزوم با تغییر یک یا چند بیت از آن از صفر به یک یا از یک به صفر.

۸- احتمال جهش: احتمال جهش که معمولاً $۰,۰۰۱ < P_m < ۰,۰۱$ است.

۹- تزویج: تبدیل دو کروموزوم به دو کروموزوم جدید با جایگزینی قسمتی از بیت‌هایشان با یکدیگر.

۱۰- P_c احتمال تزویج، که معمولاً ۰,۹ در نظر گرفته می‌شود.

۱۱-تابع برازش: تابعی برای تشخیص شایستگی یک کروموزوم.

۱۲-انتخاب: به انتخاب کروموزوم‌ها گفته می‌شود.

۲-۲-۷-۵ روند کلی الگوریتم

۱- عملگر انتخاب، جمعیت اولیه را به صورت تصادفی ایجاد می‌کند.

۲- جمعیت جاری کدگشایی می‌گردد.

۳- مقادیر تابع هدف برای اعضای جمعیت جاری محاسبه می‌شوند.

۴- تابع برازندگی اعضا با استفاده از مقادیر تابع هدف نظیر محاسبه می‌شوند.

۵- جمعیت مؤثر از جمعیت جاری استخراج می‌شود.

۶- عملگرهای تزویج هر دو عضو جمعیت مؤثر را به هم منتسب نموده و عمل پیوند بین آن دو انجام می‌شود.

۷- عملگر جهش به اعضای جمعیت جدید اعمال می‌شود.

۸- اعضای جمعیت جدید کدگشایی می‌شوند.

۹- معیار همگرایی برحسب متغیرهای جدید کنترل می‌شوند.

۱۰- درصورت برقراری شرط همگرایی، محاسبات متوقف می‌شود.

۱۱- درصورت عدم برقراری شرط همگرایی، جمعیت جدید جایگزین جمعیت جاری شده و محاسبات از گام ۲ تکرار می‌شود.

۲-۲-۷-۶ تنظیم پارامترهای الگوریتم وراثتی (ژنتیک)

این پارامترها عبارتند از[۲۱]:

L = طول هر کروموزوم که عبارت است از مجموع طول متغیرها یا ژن‌های موجود در آن.

L_i = طول هر ژن (متغیر)

P_c = احتمال تزویج

P_m = احتمال جهش

F = تابع برازش

N = تابع برازش

۲-۲-۷-۶-۱ توضیح مختصری در ارتباط با هر پارامتر

L_i = طول هر ژن یا متغیر که به محدوده متغیرها و دقت بازه متغیر بستگی دارد، مثلاً اگر $0 < x < 20$ باشد، در نتیجه متناسب با طول بازه که ۲۰ باشد و همچنین دقت اعداد آن مثلاً بین صفر و یک، ۰,۰۰۱ در نظر بگیریم در نتیجه طول متغیر یا ژن در نظر گرفته می‌شود، که طول آن به صورت چند بیت است و با کد باینری، اعداد صفر و یک در آن جایگزین می‌شود. محدوده‌ی متغیر باید مقداری منطقی باشد، یعنی نه خیلی بیشتر از انتظار باشد که باعث شود طول متغیر زیاد گردد و زمان اجرای برنامه را افزایش دهد و همچنین نباید بیش از حد کوچک باشد، زیرا ممکن است مقادیر بهینه خارج از محدود قرار بگیرد.

P_c = احتمال تزویج که عددی بین صفر و یک است، بدین صورت که به کروموزوم‌ها احتمالی برای تزویج و تولید نسل جدید داده می‌شود، یعنی اگر ۱۰۰ کروموزوم داشته و احتمال P_c = ۰,۸ باشد در نتیجه با این احتمال کروموزوم‌ها با یکدیگر تزویج می‌کنند، یعنی ۰۸ تا از ۱۰۰ تا کروموزوم، دو به دو با یکدیگر تزویج می‌کنند، البته عمل تزویج در اینجا نوعی تبدیل است تا جمعیت کل ثابت باقی بماند. انتخاب P_c بستگی به تجربه و نوع مسأله دارد. انتخاب P_c مناسب بر جواب‌دهدی و مدت زمان جواب‌دهی و دقت جواب بسیار تأثیر گذار می‌باشد، اگر آن را کم در نظر بگیریم متلاً P_c = ۰,۵ در نتیجه تعداد زیادی از کروموزوم‌ها (نصف کروموزوم‌ها) بدون تزویج باقی می‌مانند و به مرحله بعد نمی‌روند و کار خاصی در قبالشان انجام نمی‌شود. به همین علت انتخاب P_c های کم موجب نقص در جواب می‌شود، از طرفی نباید P_c را خیلی زیاد در نظر گرفت، مثلاً ۱، زیرا با این کار یعنی تمامی جمعیت دو به دو با یکدیگر تزویج می‌کنند و نتیجه تزویج می‌تواند هر چیز باشد در صورتی که در حالت P_c = ۰,۹، در حدود ۱۰ درصد جمعیت نخبه بدون تزویج به مرحله بعد منتقل می‌شوند، ولی در این حالت (P_c = ۱) ممکن است که آن ۱۰ درصد ارزش خود را از دست داده باشند و خراب شوند. البته تمامی این روندها تصادفی هستند، ولی در هر صورت تجربه می‌گوید که P_c در حدود ۰,۹ در نظر گرفته شود. مقادیر مطلوب دیگر می‌تواند ۰,۸۵ یا ۰,۸ باشند.

P_m = احتمال جهش، هر چند که این احتمال نیز می‌تواند عددی بین صفر و یک باشد ولی برای همگرایی باید مقادیر کوچک P_m را منظور کنیم، انتخاب مقدار مناسب برای P_m نیز بستگی به تجربه و نوع مسأله دارد. اگر P_m=۰,۰۰۵ باشد، در نتیجه $n \times$ ۰,۰۰۵ تا از بیت‌ها عوض می‌شوند. یعنی اگرصفر بود به یک و اگر یک بود به صفر تغییر می‌یابد. همانگونه که گفته شد انتخاب P_m بزرگ، مثلاً ۰,۱ جواب را دچار مشکل می‌سازد بدین صورت که به احتمال زیاد بسیاری از کروموزوم‌ها و ژن‌ها تغییر خصوصیت می‌دهند، مثلاً یک ژن نخبه ممکن است با جهش به کل از نخبگی درآید. حال هر چه P_m بزرگتر شود احتمال خراب شدن ژن‌ها و شامل شدن جهش به روی کروموزوم‌های برتر بیشتر می‌شود که احتمال اینکه برنامه در مقادیر بهینه محلی گیرکند و نتواند به مقدار بهینه کلی دست یابد، (قطعاً P_mهای بزرگتر از ۰,۱ کارایی الگوریتم را به کلی از بین می‌برد و ارزش آن را سقط می‌کند) معمولاً ۰,۰۱ > P_m > ۰,۰۰۱ مناسب می‌باشد.

F = تابع برازش یا تابع هدف، تابع F در اصل تابعی است که متغیرها یا ژن‌ها را به هم ربط می‌دهد، به‌دست آوردن این تابع یا به بیان دیگر، ربط دادن متغیرها به یکدیگر از مهمترین کارهای یک مهندس می‌باشد. مقدار برازش یا شایستگی می‌تواند هرچیزی باشد، مثلاً ماکزیمم چیزی باشد (مثل ماکزیمم نیرویی که یک خرپا می‌تواند تحمل کند)، و یا مینیمم باشد (مثل مینیمم وزن یک سازه) و یا یک عدد دلخواه باشد، و تمامی این‌ها بستگی به مسأله دارد.

N = تعداد کروموزوم‌ها: برای تولید جمعیت اولیه که مقدار آن ثابت می‌ماند، در ابتدا احتیاج به تولید جمعیت اولیه است، تا با این جمعیت کار شروع و ادامه داده شود، تعداد اولیه این جمعیت هر مقدار دلخواه می‌تواند باشد و این مقدار را برنامه به صورت ورودی از ما می‌خواهد. البته دادن N به ورودی برنامه مانند P_m و P_c معیارهایی دارد، (هرچند که زیاد و کم دادن این مقدار نسبی است یعنی باز هم به مسأله بستگی دارد مثلاً ممکن است تعداد ۱۰۰۰ کروموزوم برای مسأله‌ای بسیار زیاد باشد ولی برای مسأله دیگر مطلوب باشد).

اگر تعداد کروموزوم زیاد باشد در نتیجه سرعت همگرایی را پایین می‌آورد، ولی احتمال گیر کردن در بهینه-های محلی را کاهش می‌دهد. اگر تعداد کروموزوم‌ها کم باشد در نتیجه سرعت همگرایی بالا می‌شود، ولی عیب بزرگ در این حالت این است که احتمال گیر کردن در بهینه‌های محلی را را زیاد است، در نتیجه انتخاب تعداد کروموزوم ها هم بستگی به تجربه و مسأله دارد و در کل انتخاب کروموزوم بیشتر، بهتر از انتخاب کم آن است، (این مطلب کلی نمی‌باشد زیرا در مسائل بسیار سنگین استفاده از میکروژن‌ها رایج می‌باشد که در آن تعداد کروموزوم‌ها کم می‌باشد). در مورد انتخاب تعداد کروموزوم‌ها در حالت کلی می‌توان بر این نکته اشاره کرد که معمولاً در مهندسی دقت و سرعت در ملازمه با یکدیگرند، یعنی در هر جا دقت خوب خواسته شده، سرعت پایین است و بلاعکس.

۲-۲-۷-۶-۲ ایجاد جمعیت اولیه

حال که کروموزوم‌ها مشخص می‌باشد نوبت به ایجاد جمعیت اولیه می‌رسد، این عمل به صورت تصادفی صورت می‌پذیرد و تمامی کروموزوم‌ها به صورت باینری (صفر و یک) و تصادفی پر می‌شوند.

۲-۲-۷-۶-۳ کدگشایی

حال مقادیر مربوط به متغیرهای هر کروموزوم به ترتیب کدگشایی می‌شود و تبدیل به عدد می‌گردنند، مثلاً فرض کنید هر کروموزوم شامل دو متغیر که یکی ۶ بیتی و دیگری ۴ بیتی است، در نتیجه دو عدد به‌وجود می‌آید که کاملاً تصافی هستند ولی محدوده‌ای مشخص دارند.

۲-۲-۷-۶-۴ ارزیابی هر یک از کروموزوم‌ها

بعد از کدگشایی متغیرهای موجود در هر کروموزوم، نوبت به شایستگی یا برازش هر کروموزوم می‌رسد، و همان طور که گفته شد این شایستگی بستگی به مسأله دارد و ممکن است Max یا Min و یا یک عدد خاصی باشد. مثلاً اگر نوع شایستگی Max باشد در نتیجه مقادیری از تابع برازش که بیشتر هستند شایستگی بیشتری دارند، و این شایستگی در مراحل بعد بسیار مفید است، (مثلاً احتمال انتخاب کروموزوم‌های آن‌ها از بین کروموزوم-های دیگر بیشتر می‌شود).

۲-۲-۷-۶-۵ انتخاب

روش‌های انتخاب:

۱- انتخاب متناسب با برازندگی (روش چرخ گردان)

۲- انتخاب به صورت تورنومنت

۳- انتخاب نخبگان

۴- انتخاب بولتزمن

۵- انتخاب مرتبه‌ای

۶- Sigma scaling

در اینجا به توضیح دو روش انتخاب شامل انتخاب با برازندگی و انتخاب به صورت تورنومنت پرداخته می‌شود.

الف) روش چرخ‌گردان

در این روش، چرخ‌گردان و یک نشانگر وجود دارد. بر روی چرخ گردان قطاع‌هایی متناسب با مقدار برازندگی هر کروموزوم انتخاب می‌شود و سپس چرخ چرخانده می‌شود و هرجا که ایستاد قطاع مربوط به نشانگر انتخاب

می‌شود و این کار به تعداد کل قطاع‌ها ادامه می‌یابد. منطقی است که هر قطاعی بزرگتر باشد، احتمال انتخاب آن بیشتر می‌باشد و بلاعکس.

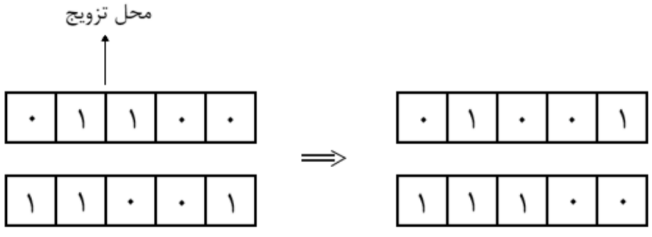
ب) انتخاب به روش تورنومنت

در این روش کروموزوم‌ها با یکدیگر مقایسه و هر کدام که برازنده‌تر بود به نسل بعد برده می‌شود. همان طور که گفته شد برازندگی می‌تواند Max یا Min باشد (یعنی نباید تصور کرد که برازنده‌تر بودن همیشه به معنای ماکزیمم‌تر بودن است). در این روش به ترتیب برازندگی هر کروموزوم را با برازندگی کروموزومی تصادفی مقایسه می‌شود و هر کدام را که برازنده‌تر بود انتخاب می‌شود و به نسل بعد برده می‌شود. این کار را به تعداد تمامی کروموزوم‌ها انجام می‌دهیم.

روش تورنومنت هم برای یافتن ماکزیمم و هم مینیمم کاربرد دارد و در کل کارایی ان بهتر از روش چرخ گردان است.

۲-۲-۷-۶-۶ تزویج (Crossover)

بعد از انتخاب کروموزوم‌ها نوبت به تزویج می‌رسد، در نتیجه با در نظر گرفتن احتمال تزویج (P_c) عمل تزویج بر روی کروموزوم‌ها انجام می‌گیرد. انتخاب محل تزویج به صورت تصادفی انجام می‌پذیرد. در شکل ۲-۵ کروموزوم-ها از محل نشان داده شده تزویج داده می‌شوند و دو کروموزوم جدید به‌وجود می‌آید.



شکل (۲-۵) تزویج دو کروموزوم

عوض کردن محل نشان داده شده به معنای عوض کردن بیت‌های شامل صفر و یک هر کروموزوم با هم می‌باشد. روش‌های دیگر تزویج نیز وجود دارد، مثلاً تزویج دو نقطه‌ای که در آن به جای تزویج از یک نقطه، از دو نقطه تزویج انجام می‌شود. علت تزویج این است که ممکن است اطلاعات موجود در بیت‌های آخر یا اول موجود در کوروموزوم‌ها مفید نباشند و با عمل تزویج به مقدار مطلوب خود برسند و کروموزوم بهینه شود، یعنی کمبود خود را از کروموزوم‌های دیگر جبران کند.

۲-۲-۷-۶-۷ جهش (Mutation)

جهش به دو منظور به کار برده می‌شود:

۱- احتمال گیر کردن در بهینه‌های محلی.

۲- اگر بیت‌های تمامی کروموزوم‌ها از یک جا به بعد مثل هم باشند و جای دو بیت یا چند بیت عوض شود، در نتیجه کروموزوم جدیدی به‌وجود نمی‌آید.

البته توجه داریم که این دو مورد احتمال هست و اطلاعی نداریم که این دو مورد به وجود می‌آیند یا خیر ولی در هر صورت ما باید این دو احتمال را در نظر بگیریم و از P_m استفاده کنیم.

جهش به این صورت است که اگر N کروموزوم داشته باشیم که طول هر کدام L باشد (L تعداد بیت‌های هر کروموزوم). در نتیجه به تعداد $N \times L \times P_m$ تا از بیت‌ها را به صورت تصادفی تغییر می‌دهیم، یعنی همان طور که قبلاً اشاره شد اگر بیت مورد نظر صفر بود یک و اگر یک بود صفر گذاشته می‌شود. به طور کلی استفاده از جهش برای سازگاری بیشتر با طبیعت مفید است.

بعد از جهش نسل جدیدی خواهیم داشت، یعنی نسل جدید جایگزین جمعیت اولیه می‌شود و مراحل تکرار می‌شود تا شرط توقف الگوریتم ارضا گردد.

۲-۲-۷-۶-۸ شرط توقف الگوریتم

۱- چنانچه تعداد نسل به اندازه معینی برسد، مثلاً شرط توقف الگوریتم در تولید ۱۰۰۰ نسل گذاشته شود.

۲- حد همگرایی کروموزوم‌ها به مقدار مشخصی برسد، یعنی مثلاً شرط توقف وقتی که ۹۰ درصد کروموزوم‌ها شبیه به هم می‌شوند.

۳- مقدار بهینه‌ی ژن در نسل $t+1$ نسبت به نسل t از مقدار ε کوچکتر می‌شد.

۴- مقدار متوسط توابع برازش در چند نسل از یک مقدار تلرانس بیشتر تغییر نکند.

۵- زمان اجرای برنامه به مقدار مشخصی برسد.

۶- شرط توقف در مسائل جستجو وقتی است که تمامی قیود ارضا شوند.

شروط توقف اول و دوم نسبت به هم ارجحیت خاصی ندارند و می‌توانند مورد استفاده قرار بگیرند، شرط سوم شرط خوبی نمی‌باشد، زیرا ممکن است در چند بار تولید نسل، مقدار بهتری به‌دست نیاید و بعد از چند نسل به یکباره به مقدار خوبی برسیم. شرط چهارم نیز روش قابل قبولی است، همچنین قابل ذکر است که شرط پنجم کلی نمی‌باشد.

فصل سوم

بهینه سازی دینامیکی با روش‌هایESO و GESO

۳-۱ مقدمه

در این فصل ابتدا به شرح و بررسی ایده اصلی و نحوع عملکردی روش‌های ESO و BESO پرداخته می‌شود. و نهایتاً روش GESO در بهینه‌سازی توپولوژی سازه‌ها تحت بارهای دینامیکی مورد بررسی قرار می‌گیرد که شامل بحث درباره ساختار مسأله، معیار همگرایی و تحلیل حسایت می‌باشد. همان‌طور که اشاره شد این فصل متمرکز بر بهینه‌سازی دینامیکی سازه‌ها می‌باشد که به راحتی برای دامنه‌ی زیادی از مسائل می‌توان به‌کار برد.

۳-۲ بهینه سازی دینامیکی با روشESO

بهینه‌سازی تحت بارهای دینامیکی دارای اهمیت زیادی در اکثر رشته‌های مهندسی می‌باشد. مقالاتی زیادی در زمینه بهینه‌سازی تحت بارهای استاتیک در مقایسه با مقالات بهینه‌سازی تحت بارهای دینامیکی متمرکز بوده است. از این بین بیشتر مطالعات و تحقیقاتی انجام گرفته در این زمینه محدود به تغییر اندازه قاب‌ها و ضخامت ورق‌ها بوده است. علاوه بر این موارد، در زمینه‌ی بهینه‌سازی توپولوژی مسائل دینامکی کارهایی صورت گرفته است، که به موفقیت‌هایی نیز نائل شده است. که می‌توان به کارهای انجام گرفته توسط دیاز و کیکوچی[۲۲] و تنک وهاجیوارا[۲۳] که از روش همگن‌سازی بهره برده‌اند، اشاره کرد. همان طور که می دانیم پاسخ یک سازه به تحریک دینامیکی به چند فرکانس طبیعی اول سازه وابستگی زیادی دارد. همچنین زمانی که فرکانس تحریک دینامیکی نزدیک به یکی از فرکانس های طبیعی سازه شود، ارتعاش بیش از حد سازه اتفاق می افتد. اغلب نیاز است که طراحی بیشتر سازه ها، بهینه‌سازی سازه در دامنه تعیین شده به‌صورتی که از ارتعاش بیش از حد سازه جلوگیری نماید، صورت پذیرد.

در این بخش به روش‌های بهینه‌سازی تکاملی سازه (ESO) برای بهینه‌سازی توپولوژی سازه تحت بارهای دینامیکی پرداخته می‌شود. ایده ساده‌ی روش ESO که برای اولین بار توسط ژی و استیون[۴] در سال ۱۹۹۳

ارائه شد، بر پایه حذف تدریجی مواد ناکارآمد و تکامل تدریجی سازه به سمت بهینگی می‌باشد. از مزیت‌های مهم روش مذکور می‌توان به ایده قابل فهم و آسان و عملیات ریاضی ساده در مقایسه با روش‌های متداول دیگر اشاره کرد. از این روش برای گستره وسیعی از مسائل مختلف بهینه‌سازی سازه مانند بهینه‌سازی برپایه تنش، سختی، فرکانس و بارهای کمانشی استفاده شده است. این پوشش نشان دهنده‌ی توانایی این روش در حل مسائل مختلف بهینه‌سازی سازه می‌باشد. در گام بعدی روش BESO که بر پایه ESO می باشد، ارائه شد که می‌تواند علاوه بر حذف مواد ناکارآمد، به شکل همزمان مواد کارآمد را نیز اضافه نماید. در ادامه به بررسی روش ESO جهت بهینه‌سازی در مسائل دینامیکی که توسط ژی و استیون[۲۴] ارائه شده است، پرداخته می‌شود.

۳-۲-۱ رابطه‌سازی ریاضی روش و محاسبه‌ی حساسیت

معمولاً برای محاسبه‌ی حساسیت نیاز به داشتن گرادیان‌های تابع هدف نسبت به متغیرهای طراحی است. اما در روش‌های تصادفی مانند ESO نیاز به هیچ اطلاعاتی از گرادیان تابع هدف نمی‌باشد. و جستجو برای رسیدن به نقطه بهینه با مقایسه مقادیر تابع هدف در نقاط مختلف دامنه طراحی صورت می‌پذیرد. حساسیت در این روش در واقع برآوردی از تغییر تابع هدف در نتیجه‌ی حذف یک المان ارائه می‌کند. به طور معمول تحلیل حساسیت بخصوص برای مسائل دینامیکی پیچیده و زمان‌بر است. اما همان طور که نشان داده خواهد شد، عدد حساسیت را می‌توان به راحتی برای هر المان محاسبه کرد.

معادله دیفرانسیل حاکم بر مسائل دینامیکی به صورت زیر می‌باشد:

$$M\ddot{U} + C\dot{U} + KU = F(x,t) \qquad x \epsilon \Omega^s, t > 0 \qquad (۱-۳)$$

که $[K]$ ماتریس سختی کل، $[M]$ ماتریس جرم کل، $[C]$ ماتریس میرایی کل ، $F(x\ ,t)$ بار هارمونیک با فرکانسی مشخص و Ω^s دامنه طراحی سازه می باشد.

اگر بار $F(x\ ,\ t)$ یک بار هارمونیک وابسته به زمان در نظر بگیریم، آنگاه می‌توانیم داشته باشیم:

$$F(x,t) = f(\omega)e^{i\omega t} \qquad (۲-۳)$$

$f(\omega)$ اندازه بار و ω فرکانس زاویه‌ای می‌باشد.

بردار جابجایی گرهی می‌تواند به‌صورت $U(x\ ,\ t) = u(\omega)e^{i\omega t}$ در نظر گرفته شود. با جایگذاری معادله جابه‌جایی و معادله بار هارمونیک در معادله (۵-۱) بردار جابجایی به‌صورت زیر بدست خواهد آمد:

$$\{-\omega^2[M] + i\omega[C] + [K]\}[u(\omega)] = f(\omega) \qquad (۳-۳)$$

برای یک سازه الاستیک نامیرا خواهیم داشت:

$$\{-\omega^2[M] + [K]\}[u(\omega)] = f(\omega) \qquad (۴-۳)$$

در یک مسأله بهینه‌سازی به طور معمول هدف، ماکزیم کردن سختی سازه است. ماکزیمم کردن سختی سازه معادل با مینیمم کردن انرژی کرنشی سازه می باشد. مسأله بهینه‌سازی برای این دسته از مسائل به شکل زیر تعریف می‌شود:

$$Maximise \quad C = f^T u \qquad (۵-۳)$$

$$Subject\ to \quad g = V^* - \sum_{i=1}^{N} V_i x_i = 0\ . \quad x_i \in \{0\ .1\}\ . \qquad (۶-۳)$$

که C عبارت است از انرژی کرنشی، V^* وزن کل ، V_i وزن i امین المان، x_i متغیر باینری طراحی که نشان دهنده‌ی این است که المان وجود دارد $(x_i = 1)$ یا ندارد $(x_i = 0)$ و n تعداد کل المان‌های سازه می‌باشد. این فرمول بیان کننده‌ی یک مسأله بهینه‌سازی تحت قید وزن است.

چنانچه i امین المان از یک سازه شامل n المان حذف شود، ماتریس سختی به شکل زیر تغییرخواهد کرد:

$$\Delta \boldsymbol{K} = \boldsymbol{K}^* - \boldsymbol{K} = -\boldsymbol{K}_i \qquad (۷-۳)$$

که $\boldsymbol{K}^*$ ماتریس سختی بدست آمده از حذف i امین المان و $\boldsymbol{K_i}$ ماتریس سختی i امین المان می‌باشد.

فرض می‌شود که حذف المان تأثیری بر بردار بارگذاری ندارد تغییر در بردار جابه‌جایی به شکل زیر می تواند تعریف شود:

$$\Delta \boldsymbol{u} = -\boldsymbol{K}^{-1}\Delta \boldsymbol{K}\boldsymbol{u} \qquad (۸-۳)$$

بعد از حذف i امین المان خواهیم داشت:

$$\Delta C = \frac{1}{2}\boldsymbol{P}^T\Delta \boldsymbol{u} = \frac{1}{2}\boldsymbol{P}^T\boldsymbol{K}^{-1}\Delta \boldsymbol{K}\boldsymbol{u} = \frac{1}{2}\boldsymbol{u}_i^T\boldsymbol{K}_i\boldsymbol{u}_i \qquad (۹-۳)$$

که $\boldsymbol{u}_i$ بردار جابجایی i امین المان می‌باشد.

برای مشخص شدن اینکه چه المان‌هایی باید از سازه حذف شوند تا به سازه‌ای با حداقل انرژی کرنشی دست یابیم، باید برای هر المان عدد حساسیت به طریق زیر محاسبه شود:

$$\alpha_i = \frac{1}{2}\boldsymbol{u}_i^T\boldsymbol{K}_i\boldsymbol{u}_i \qquad (۱۰-۳)$$

معادله بالا نشان دهنده‌ی تغییر در انرژی کرنشی ناشی از حذف i امین المان است. حذف المان‌هایی که مقادیر α_i آن‌ها کمترین تأثیر را در مسیر مینیمم کردن انرژی کرنشی یک سازه دارند. اگر از المان ۴ گرهی استفاده شود $\boldsymbol{K}_i$ ماتریس ۸ × ۸ و $\boldsymbol{u}_i$ یک بردار ۸ × ۱ خواهند بود.

۳-۲-۲ روند بهینه‌سازی برای مسائل دینامیکی با استفاده از روش ESO

بهینه‌سازی روندی تکاملی است، به عبارت دیگر تنها مقادیر کمی از مواد باید در هر تکرار از سازه حذف شوند. بهتر است حداکثر یک درصد از کل مواد در هر مرحله از سازه حذف شوند.

گام اول- اعمال تقسیم بندی اجزا محدود در تمام دامنه طراحی با توجه به مرزها.

گام دوم- انجام تحلیل دینامیکی برای بدست آوردن مقادیر جابه‌جایی.

گام سوم- محاسبه‌ی حساسیت المان‌ها.

گام چهارم- حذف المان‌هایی که کمترین حساسیت را دارا می‌باشند.

گام پنجم- تکرار گام‌های ۲ تا ۴ تا زمانی که وزن به میزان تعیین شده برسد.

همان‌گونه که ملاحظه می‌شود روند حل بسیار ساده است. فقط به یک شبکه بندی اجزا محدود در کل روند تکاملی نیاز است و لازم به باز تولید آن نمی‌باشد. در عوض المان‌هایی که در هر تکرار حذف می‌شوند را در تشکیل ماتریس سختی و جرم کل نادیده می‌گیریم. بنابراین در طول روند تکامل هر چه المان‌های بیشتری حذف می‌شوند، تعداد معادلات کمتری خواهیم داشت.

می‌توان راه‌های مختلفی برای پایان دادن به روند تکامل در نظر گرفت. یک استراتژی این است که حداکثر مقدار موادی را که اجازه داده می‌شود تا از سازه حذف شوند، برای پایان یافتن روند بهینه‌سازی در نظر بگیریم یا می‌توان زمانی که انرژی کنشی سازه تغییر محسوسی در جهت مطلوب ندارد روند را متوقف نماییم.

۳-۲-۳ روند بهینه‌سازی برای مسائل دینامیکی با استفاده از روش BESO

گام اول- اعمال تقسیم بندی اجزا محدود در دامنه طراحی که همه دامنه طراحی را پوشش دهد.

گام دوم- مشخص کردن طرح اولیه که می‌تواند قسمتی از تمام دامنه طراحی باشد. ترجیحاً حداقل المان‌هایی که تکیه گاه‌ها و بار را به هم متصل می‌کند.

گام سوم- انجام تحلیل دینامیکی.

گام چهارم- محاسبه‌ی حساسیت المان‌های موجود و المان‌هایی که پتانسیل اضافه شدن را دارا می‌باشند.

گام پنجم- مرتب سازی المان‌ها براساس میزان حساسیت‌شان. حذف المان‌هایی که کمترین حساسیت را دارا می‌باشند و اضافه کردن المان‌هایی که بیشترین حساسیت را در اطراف المان‌های موجود دارند.

گام ششم- تکرار گام‌های ۳ تا ۵ تا زمانی که وزن به میزان تعیین شده برسد. پارامترهای MR و AR برای مشخص شدن اینکه چه المان‌هایی حذف یا اضافه می‌شوند مورد استفاده قرار می‌گیرد. پارامتر MR با توجه به مقدار AR به دو قسمت $MR \times AR$ و $(1 - AR) \times MR$ تقسیم می‌شود.

اگر AR کمتر از ۰,۵ باشد تعداد المان‌های حذف شده بیشتر از المان‌های اضافه شده می‌شود و وزن سازه کاهش می‌یابد، و بلاعکس اگر AR بیشتر از ۰,۵ باشد تعداد المان‌های حذف شده کمتر از المان‌های اضافه شده می‌شود و وزن سازه افزایش می‌یابد. این روند تا رسیدن به وزن تعیین شده ادامه می‌یابد.

۳-۲-۴ جمع‌بندی

همان طور که ذکر شد از روش‌های ESO و BESO می‌توان به راحتی برای بهینه‌سازی سختی یک سازه استفاده شود. روند بهینه‌سازی تقریباً مشابه با آنچه در مسائل استاتیکی می‌باشد صورت می‌گیرد، و تنها تفاوت موجود در نحوه‌ی محاسبه مقادیر حساسیت است. بزرگترین مزیت این دو روش در مقایسه با روش‌های ریاضی و همگن‌سازی سادگی آنها است که باعث شده به راحتی قابل فهم باشد. همچنین بعد از گذشت چند دهه روش اجزا محدود توانسته محبوبیت زیادی در علوم مهندسی پیدا کند، ولی متاسفانه در بهینه سازی آنچنان که باید عمومیت پیدا نکرده است. حال آنکه این روش ها با توجه به پتانسیلی که دارند توانسته‌اند پلی بین روش اجزا محدود و بهینه‌سازی سازه برقرار نمایند. در واقع به راحتی می‌توان کدهای روش اجزا محدود را برای روش‌های مذکور در مسائل دینامیکی وفق داد، که تنها با اضافه کردن تعدادی سطر برای محاسبه و مقایسه مقادیر حساسیت المان‌ها صورت می‌پذیرد.

۳-۳ بهینه سازی دینامیکی با روش Genetic ESO (GESO)

روش بهینه‌سازی تکامل سازه‌ها (ESO) بر مبنای یک مفهوم ساده بنا شده است، که به کمک حذف تدریجی مواد ناکارآمد از سازه می‌توان آن را بهینه نمود، اما بهینه‌سازی فقط بر مبنای حذف، روش را به سمت محلی شدن سوق می‌دهد. روش بهینه‌سازی تکاملی دوجهتی سازه‌ها (BESO) شکل بهبودیافته روش ESO شامل حذف و اضافه نمودن المان‌ها است. این روش هم نمی‌تواند از محلی شدن جلوگیری نماید زیرا اضافه کردن المان‌ها تنها در اطراف المان‌های موجود بدون در نظر گرفتن شایستگی واقعی آن‌ها صورت می‌پذیرد[۲۵].

با تلفیق عملگرهای الگوریتم ژنتیک با بهینه‌سازی تکاملی سازه‌ها که توسط لیو و همکارانش[۶] انجام گرفت، شکل جدیدی از الگوریتم بدست آمده که GESO نامیده می‌شود. این الگوریتم از مزیت الگوریتم ژنتیک در پیدا کردن جواب‌های کلی سود برده است. در این رویکرد رفتار مؤثرتری از افراد در جمعیت برای مدل اجزا محدود

ارائه می‌شود که متفاوت با آنچه پیش از این در کاربرد روش GA در طراحی سازه بوده است، می‌باشد. پیش از این استفاده از عملگرهای تزویج و جهش برای مسائل بهینه‌سازی، اثرات مناسبی بر روند بهینه‌سازی داشته است. در این بخش اثرات مختلف روش پیشنهادی در سرعت و عملکرد مسائل بهینه سازی مورد بررسی قرار گرفته می‌شود. همچنین به معرفی این الگوریتم جدید و پارامترهای مؤثر بر عملکرد آن برای بهینه‌سازی مسائل دینامیکی پرداخته می‌شود.

۳-۳-۱ درباره بهینه‌سازی تکاملی سازه‌ها

در طی سه دهه گذشته، کارهای زیادی در زمینه بهینه‌سازی توپولوژی انجام شده و تکنیک‌های مختلفی گسترش پیدا نموده‌اند. که از این میان روش‌های مانند روش SIMP بنزو [۲۶]، زو و رزوینی[۲۷] و ریتز[۲۸] و روش ESO استیون و ژی[۴] مورد استقبال بیشتری قرار گرفته‌اند.

در سال‌های اخیر، ژنتیک الگوریتم کاربرد زیادی در دامنه‌ی وسیعی از مسائل بهینه‌سازی توپولوژی داشته است هولند[۱۹] گلدبرگ و سامتانی[۲۹]. ژنتیک الگوریتم (GA) به علت استفاده از جستجوی تصادفی و همچنین استفاده از مقدار تابع هدف در نقاط مختلف فضای جستجو بدون نیاز به داشتن گرادیان تابع هدف، روشی بسیار کارآمد و مناسبی برای مسائل بهینه‌سازی توپولوژی می‌باشد. اساس این الگوریتم قانون تکامل داروین (بقا بهترین) است که اولین بار توسط هولند ارائه شد[۱۹]. در روش GA به‌جای اینکه از یک نقطه به نقطه دیگر برای رسیدن به جواب حرکت شود، در هر تکرار چند نقطه از فضای جستجو در نظر گرفته می‌شود، که تکامل هر نقطه یک فرآیند کاملاً مستقل است.

در روشی که توسط لیو[۶] در سال ۲۰۰۸ ارائه شده است، از عملگرهای ژنتیک الگوریتم در روش ESO برای بهینه‌سازی توپولوژی مسائل استاتیکی استفاده شد. در این روش به‌جای اینکه ترکیب‌های متعددی از المان‌ها در نظر گرفته شود، هر المان در دامنه‌ی طراحی توسط اجزا محدود از هم گسسته شده و بصورت منفرد رفتار می‌کند. این کار باعث کاهش چشمگیری در اندازه مسأله برای GA می‌شود.

زیرا جمعیت در این حالت معطوف به یک سازه واحد هست و یک طرح تصادفی برای سازه انتخاب نمی‌شود. با این حال همانند روش ESO تنها امکان حذف المان‌ها وجود دارد و المان‌های مهم که حذف شده‌اند امکان بازگشت در تکرارهای بعدی را ندارند. بهینه‌سازی تکاملی سازه یا ESO یک روش نیمه نظری، نیمه تجربی می‌باشد که در این روش، حرکت تدریجی از سازه اولیه به سمت سازه بهینه صورت می‌گیرد. بطور کلی ESO بر اساس روند ساده ای استوار است، که با در نظر گرفتن کلیه شرایط حاکم بر سازه، از قبیل شرایط ساختگاهی، تکیه‌گاهی، بارگذاری و ... با حذف تدریجی مصالح غیر لازم از سازه، شکل نهایی آن را با توجه به نیازها و انتظارات از تابع هدف، ارائه و سازه را به سمت بهترین نوع توزیع سوق می‌دهد. این توزیع به نحوی انجام می‌گیرد که مصالح از نقاط قوی‌تر با کارآیی کمتر به نقاط ضعیف تر با کارایی بیشتر انتقال یافته، و سازه با توجه به عملکرد مورد نظر، قابلیت بیشتری از خود نشان می‌دهد.

در این تحقیق از ترکیب روش‌های ESO و GA برای بهینه‌سازی سختی مسائل مختلف دینامیکی استفاده می‌شود. در این روش هر المان، به صورت منفرد که دارای رشته ژنتیکی باینری با طول مشخصی است، رفتار می‌کند. عملگرهای GA به‌جای کل سازه در ناحیه‌های مجزا بر روی المان‌ها اجرا می‌شوند. و باعث افزایش سرعت و دقت همگرایی می‌شود. علاوه بر این از مکانیزم filter scheme که توسط هانگ و ژی[۳۰] ارائه شده است برای برون‌یابی حساسیت المان‌های خالی در جهت بهبود عملکرد روش BESO استفاده می‌شود.

۳-۳-۲ درباره الگوریتم ژنتیک

ایده الگوریتم ژنتیک یا GA از دو اصل انتخاب و تولید نسل در طبیعت بهره برده است. با گذشت زمان ساختار ژنتیکی موجودات تغییرکرده و نسل‌های جدیدتر با محیط سازگاری بیشتری دارند. بدین طریق که امکان زنده ماندن و تولید مثل موجودات قوی‌تر بیشتر از موجودات ضعیف می‌باشد. در نسل‌های جدیدتر ساختار ژنتیکی موجودات قوی تکرار می‌شود و موجودات ضعیف از بین می‌روند. در بعضی موارد نیز جهش به‌وجود می‌آید بدین معنی که از آمیزش دو موجود، موجودی متولد می‌شود که بر اثر جهش ژنتیکی خیلی بهتر یا بدتر از والدین خود می‌باشد و به تعبیری یک نابغه یا یک عقب‌مانده متولد می‌شود و در نسل‌های بعدی تأثیر می‌گذارد. طبیعت تضمین می‌کند که این نوع زاد و ولد موجب تولید موجودات بهتر شود.

به طور خلاصه می‌توان گفت GA یک روش بهینه‌سازی است که روند تکامل و وراثت موجودات زنده را شبیه‌سازی می‌کند. در GA جمعیتی از افراد برای جستجوی اولیه تولید می‌شود و به افراد مقادیر برازندگی اختصاص داده می‌شود. سپس عملگرهای انتخاب، تزویج و جهش بر روی افراد اعمال می‌شود. با انتخاب افرادی که دارای برازندگی بالایی هستند و انجام عمل تزویج که باعث می‌شود تا مشخصات بهتری از نتیجه وراثت به‌دست بیاید و در ادامه انجام جهش که باعث حفظ تنوع می‌شود، که از طریق ایجاد مشخصات جدیدی به‌وسیله تغییر در بخش‌هایی از کروموزوم افراد می‌شود.

۳-۳-۲-۱ تزویج

کل جمعت براساس حساسیت‌هایشان رتبه‌بندی می‌شوند. حدود ٪۱۰ از کل جمعیت کنار گذاشته می‌شود و عملگر تزویج روی مابقی جمعیت اعمال می‌شود. برخلاف GA معمول هر فرد تنها یکبار می‌تواند برای عمل تزویج انتخاب شود. احتمال انتخاب شریک بر اساس مقدار P_c است، و بهتر است مقدار P_c برای این کلاس بزرگتر از

۰,۵ انتخاب شود. بعد از اینکه هر فرد برای تزویج انتخاب می‌شود، ترکیب به صورت حالت تک نقطه‌ای یا دو نقطه‌ای یا یکنواخت برای جابه‌جایی بخش‌هایی از رشته‌های ژنتیکی انجام می‌گیرید.

.

۳-۳-۲-۲ جهش

به طور کلی عملگر جهش به شکل تصادفی ممکن است هر ژن از هر کروموزوم را از ۱ به ۰ ممکن تغییر دهد. احتمال جهش با P_m که مقداری بین ۰ تا ۱ دارد نشان داده می‌شود. قبل از اینکه جهش در هر تکرار انجام گیرد. عملگر جهش تمایل به تغییر ۱ به ۰ در المان‌ها دارد. در این فرآیند یک سری از اعداد تصادفی در بازه‌ی بین ۰ تا ۱ ایجاد شده، و هر عدد مرتبط با یک ژن از کروموزوم می‌باشد. اگر این عدد تصادفی کمتر از P_m باشد و آن ژن ۱ باشد، به ۰ تغییر پیدا می‌کند.

۳-۳-۲-۳ انتخاب المان

عملیات زاد و ولد در GA بر پایه قانون تکامل داروین (بقا بهترین) است. اساساً زاد و ولد یک فرآیند انتخاب است، که در آن افراد با ژن‌های بهتر برای تولید نسل بعدی و انجام تزویج و جهش انتخاب می‌شوند. و افراد با ژن‌های بدتر در نسل بعدی جایی نخواهند داشت. در این روش اندازه جمعیت در روند بهینه‌سازی ثابت باقی می‌ماند.

اما در رویکرد این قسمت, هر فرد نشان دهنده‌ی المانی است که یک موقعیت خاص از دامنه‌ی طراحی را اشغال نموده است و هیچ معنایی ندارد که یک المان بیشتر از یکبار در نسل بعدی حضور داشته باشد. انتخاب المان بر اساس کروموزوم مربوط به آن می‌باشد، به عبارت دیگر المان‌هایی با کروموزوم‌های کاملاً ۰ حذف شده و آنهایی که باقی می‌مانند نسل جدیدی را می‌سازند.

بعلاوه مستقل از مش بودن یکی از مهمترین ویژگی‌هایی است که از تکنیک‌های بهینه‌سازی متداول انتظار می‌رود. وابستگی به مش باعث ایجاد بی‌ثباتی عددی و در نتیجه ایجاد حفره‌هایی در طرح نهایی می‌شود، و به الگو شطرنجی معروف است که می‌توان به کارهای صورت گرفنه توسط زیگموند و پترسون[۳۱] و بندز و زیگموند[۳۲] اشاره کرد. الگوهای شطرنجی برای طراحی بیشتر جزئیات تقریبا غیر عملی می‌باشند. الگوریتم‌هایی مانند کنترل محیط یانگ[۳۳ و الگوریتم هموارسازی لی[۳۴] برای مقابله با این مشکل ارائه شدند. همچنین می‌توان از مکانیزم filter scheme که توسط ژانگ و ژی[۳۰] ارائه شده است، استفاده شود. این مکانیزم در جلوگیری از ایجاد الگوی شطرنجی و ایجاد طرحی وابسته به مش، بسیار مؤثر می‌باشد.

در مقایسه با روش GA مرسوم، در این روش هر المان به شکل منفرد رفتار می‌کند. این روش نشان داد که حتی اگر جستجو طرح به شکل نیمه تصادفی هم باشد، جواب‌ها به نتایج مشابهی همگرا می‌شوند. و نیاز به انجام اجراهای متعدد برای اطمینان از رسیدن به بهینگی نمی‌باشد. رفتار المان‌ها به شکل منفرد از یک سو، و تحت تأثیر بودن جهت جستجو با توجه به میزان حساسیت از سوی دیگر باعث کاهش حجم محاسبات و رسیدن به نتایج دقیق شده است. با این حال، روش تصادفی ارائه شده دارای اشکلاتی در مواجه با مسائلی که حساس به المان‌های غیرمتصل می‌باشد. به‌خصوص برای مسائل بهینه‌سازی دینامیکی، که این المان‌های باعث آسیب به روند همگرایی می‌شود. در این موارد لازم است که در هر تکرار المان‌هایی که به این شکل می‌باشند از روند تکاملی حذف شوند[۶].

۳-۳-۳ ساختار مسأله و محاسبه حساسیت

همه روش‌های بهینه‌سازی نیاز به یک شاخص ریاضی دارند تا با مقایسه مقدار آن در بین طراحی‌های موجود بتوان ساختار بهینه را شناسایی نمود. مصالح کارآمد و یا در مقابل آن مصالح ناکارآمد توسط اعدادی به نام اعداد

حساسیت تعریف می‌شوند. در واقع با توجه به تابع برازش, اعداد حساسیت مختلفی نیز تعریف می‌شود و با اضافه یا حذف یک المان، عدد حساسیت نیز تغییر خواهد کرد.

معادله دیفرانسیل حاکم بر مسائل دینامیکی بدون در نظر گرفتن میرایی با معادله زیر مشخص می‌شود:

$$M\ddot{U} + C\dot{U} + KU = F(x,t) \qquad x \epsilon \Omega^s, t > 0 \qquad (۱۱-۳)$$

که $[K]$ ماتریس سختی کل،$[M]$ ماتریس جرم کل، $[C]$ ماتریس میرایی کل ، $F(x,t)$ بار هارمونیک با فرکانسی مشخص و Ω^s دامنه طراحی سازه می باشد.

برای یک سازه الاستیک نامیرا بردار جابه‌جایی به شکل زیر محاسبه می‌شود:

$$\{-\omega^2[M] + [K]\}[u(\omega)] = f(\omega) \qquad (۱۲-۳)$$

همان‌طور که به تفضیل در فصول گذشته توضیح داده شد عدد حساسیت المان‌ها برای مسائل بهینه‌سازی سختی به شکل زیر به دست می‌آید:

$$\alpha_i = \frac{1}{2}\boldsymbol{u}_i^T \boldsymbol{K}_i \boldsymbol{u}_i \qquad (۱۳-۳)$$

۳-۳-۴ نحوه عملکرد GESO

مطابق با پژوهش‌های صورت گرفته معیار حذفی که در ESO معرفی شده است منجر به حذف المان‌هایی می‌شود که در تکرارهای بعدی برای رسیدن به طرح بهینه کلی، لازم بوده برگشت داده شوند. در روش ESO ممکن است بعضی از المان‌هایی که در انتقال نیرو مهم هستند در ابتدای طراحی به دلیل داشتن عدد حساسیت پایین حذف شوند؛ و این باعث تغییر در مسیر انتقال نیرو می‌شود. احتملاً مسیر درست و کارآمد نیرو مفقود می‌شود و این دلیل محلی بودن روش ESO می‌باشد. اگر چه BESO اجازه اضافه کردن المان‌ها را داده است اما این روش رسیدن به یک طرح بهینه کلی را نتوانسته ضمانت نماید. در روش BESO المان‌های اضافه شده فقط به پیرامون المان‌های موجود افزوده می‌شوند.

پس از آن لیو[۶] و همکاران توانایی الگوریتم ژنتیک را در رسیدن به جواب‌های کلی در بهینه‌سازی سازه‌ها نشان دادند. در روش GESO طراحی از یک دامنه کامل شروع می‌شود، یعنی تمام المان‌ها در ابتدا وجود دارند و تمامی المان‌های موجود یک جمعیت را مشخص می‌نمایند.

در ابتدا سازه به‌وسیله اجزا محدود تحلیل می‌شود و عدد حساسیت المان‌ها مشخص می‌شود. پس از آن اعداد حساسیت رتبه‌بندی می‌شوند. بدیهی است هر المانی که بیشترین عدد حساسیت را دارا باشد رتبه بالاتری را به خود اختصاص می‌دهد. المان‌هایی که دارنده بالاترین رتبه هستند باقی می‌مانند و المان‌هایی که پایین‌ترین رتبه را دارند مستعد حذف شدن می‌باشند. به این ترتیب که به هر المان کروموزومی به طول اختیاری داده می‌شود و به تمامی ژن‌های هر کروموزوم در ابتدا عدد یک اختصاص داده است. پس از تحلیل اعداد حساسیت رتبه‌بندی می‌شوند و عملگرهای انتخاب، جهش و تزویج بر روی کروموزوم‌های هر المان اعمال می‌شود. تنها زمانی که تمامی ژن‌های هر المان صفر شود آن المان حذف خواهد شد. به این ترتیب که المان‌های برازنده‌تر با شانس بالایی باقی خواهند ماند و المان‌هایی که برازندگی کمتری دارا می‌باشند با احتمال قوی‌تری حذف خواهند شد.

به طور کلی روند بهینه‌سازی را می‌توان در سه گام کلی بیان نمود: (۱) محاسبه برازندگی هر فرد، (۲) اعمال عملگرهای تزویج و جهش برای تولید نسل آینده، و (۳) حذف المان‌ها، به عبارت دیگر انتخاب افراد برای ماندن در دامنه طراحی، و تکرار این روند تا رسیدن به بهینگی. برای رسیدن به یک جواب کلی شاید ناگریز به بارها حل مسأله باشیم، درنتیجه می‌توان در کنار GESO از یک ضخامت موقتی برای المان‌ها استفاده شود. هر المانی که تنها یک صفر در کروموزوم خود داشته باشد ضخامتش به K برابر ضخامت اولیه خود می‌شود. بنابراین اگر المانی در انتقال نیرو مؤثر باشد عدد حساسیت آن در مراحل بعدی افزوده می‌شود و شانس دوباره‌ای برا حضور خواهد داشت. در این روش جدید با کدگذاری متغیرهای طراحی شروع می‌شود، یعنی اختصاص یک رشته باینری با طولی مشخص به هر المان در دامنه طراحی[۳۵].

در روش ترکیبی GA-ESO هر المان به شکل یک فرد و دارای کروموزومی شامل مقادیر باینری در نظر گرفته می‌شود، و عملگرهای تزویج و جهش روی آن اعمال می‌شود. المان‌هایی با رشته‌های کاملاً ۰ در پایان هر تکرار از طرح حذف، و المان‌هایی با حداقل یه عدد ۱ در رشته ژنتیکی حفظ خواهند شد. هرچه میزان عدد ۱ در کروموزوم یک المان بیشتر باشد، احتمال بیشتری برای باقی ماندن در تکرارهای بعدی دارد. برای بهینه سازی با طرح اولیه که تمام دامنه را پوشش می‌دهد، رشته‌های ژنتیکی کاملاً ۱ به هر المان اختصاص پیدا می‌کند. توجه به این نکته لازم است که مشخص کردن طول رشته ژنتیکی به سرعت انجام برنامه وابسته است. به طور معمول این طول نباید از ۴ کمتر باشد[۳۵].

در هر تکرار، ابتدا تحلیل اجزا محدود انجام و حساسیت هر المان محاسبه می‌شود. سپس عملگرهای تزویج و جهش بر روی تمام کروموزوها که براساس برازندگی‌شان رتبه‌بندی شده‌اند، اعمال می‌شود. تزویج روندی است که در آن رشته‌های والدین به بخش‌هایی شکسته شده و تعدادی از آنها با بخش‌های مشابه با والد دیگر عوض می‌شود. سپس عملگر جهش روی رشته‌های ژنتیکی اعمال و باعث تغییر در اعداد باینری می‌شود. به این صورت که در المان‌هایی که کمترین حساسیت را دارا می‌باشند تغییر ۱ به ۰ رخ می‌دهد. پس از اعمال عملگرهای تزویج و جهش، جمعیت جدیدی از المان‌ها تولید می‌شود. در این روش برخلاف GA هیچ یک از المان‌ها بیشتر از یکبار نمی‌تواند انتخاب شود.

۳-۳-۵ روند بهینه‌سازی برای روش GESO

۱- ساخت مدل المان محدود و تعیین شرایط مرزی و بارگذاری.

۲- اختصاص کروموزوم با طول دلخواه و ژن‌های پر از یک به هر المان.

۳- انجام تحلیل دینامیکی برای بدست آوردن مقادیر جابجایی.

۴- بدست آوردن عدد حساسیت‌شان برای هر المان.

۵- رتبه‌بندی اعداد حساسیت.

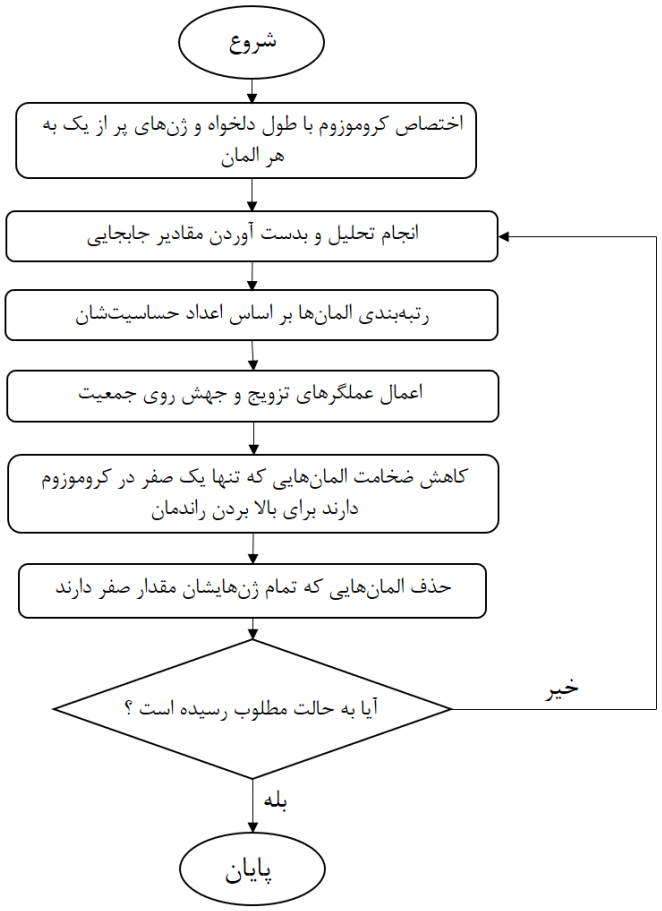
۶- انجام فرآیند تزویج بر روی کل جمعیت.

۷- انجام فرآیند جهش بروی المان‌هایی که دارای حساسیت کم می‌باشند.

۸- کاهش ضخامت المان‌هایی که تنها یک صفر در کروموزوم دارند برای بالا بردن راندمان.

۹- حذف کروموزوم‌هایی که تمام ژن‌هایشان مقدار صفر دارند.

۱۰-تکرار گام‌های ۳ تا ۸ تا رسیدن به بهینگی.



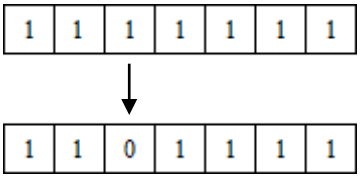
شکل (۳-۱) فلوچارت روش GESO

در این روش در ابتدای روند بهینه‌سازی به ازای هر المان یک کروموزوم با طول دلخواه که تمامی ژن‌ها حاوی عدد یک می‌باشند اختصاص داده می‌شود.

1	1	1	1	1	1	1
1	1	1	1	1	1	1
1	1	1	1	1	1	1
1	1	1	1	1	1	1
1	1	1	1	1	1	1
1	1	1	1	1	1	1
1	1	1	1	1	1	1

شکل (۳-۲) ایجاد جمعیت اولیه در روش GESO

پس از آن مقادیر حساسیت رتبه‌بندی شده، و برازندگی هر یک از کروموزوم‌ها مشخص می‌شود. پس از اعمال عملگر تزویج روی کل جمعیت در ادامه عملگر جهش بر روی المان‌هایی که بدترین رتبه را دارند اعمال می‌شود، به این شکل که هر کروموزومی که مورد جهش قرار می‌گیرد مقدار صفر به آن ژن اختصاص داده می‌شود.



شکل (۳-۳) انجام جهش در کروموزوم

در این روش می‌توان برای افزایش راندمان روش GESO بعد از تزویج و جهش ضخامت المان‌هایی که تنها یک صفر در کروموزوم دارند را کاهش داد. سپس در یک فرآیند تکراری هر بار تعدادی از المان‌ها حذف شده و طرح به سمت بهینگی سوق داده می‌شود[۶].

فصل چهارم

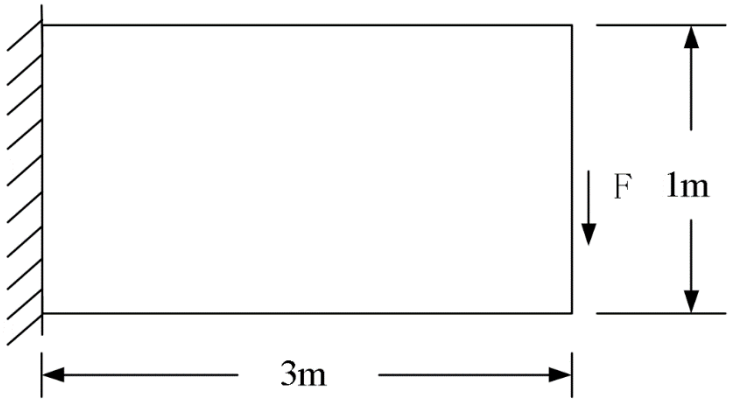
مثال‌های عددی

۴-۱ مقدمه

در این فصل ابتدا به بررسی طول بهینه کروموزوم برای مسائل مورد بررسی توسط روش پیشنهادی GESO پرداخته می‌شود. سپس با ارائه مکانیزمی از ایجاد حالت شطرنجی (Checkerboard) که مانع از رسیدن به بهینگی در مسائل بهینه‌سازی می‌شود، جلوگیری به‌عمل می‌آید. در پایان به صحت‌سنجی روش پیشنهادی و بررسی چند مثال در زمینه بهینه‌سازی دینامیکی و تحلیل نتایج به‌دست آمده پرداخته می‌شود.

۴-۲ انتخاب طول کروموزوم

همان طور که مشخص است هر چه طول کروموزوم بیشتر باشد، زمان اجرای برنامه بیشتر افزایش پیدا خواهد کرد. اما اگر طول کروموزوم از حد نرمال کمتر انتخاب شود ممکن است باعث اختلال در انجام وظایف عملگرهای جهش و تزویج شود. به همین دلیل یافتن طول مناسب برای کروموزوم اهمیت زیادی دارد. در این قسمت سعی شده است طول بهینه کروموزوم برای مسائل مورد بررسی قرار گیرد. بدین جهت به بررسی تیر کنسولی به ابعاد ۱ m×۲m می‌پردازیم. ضخامت صفحه برابر ۰,۰۵ m است. مدول الاستیک و نسبت پواسن به ترتیب برابر با ۲۰۰ GPa و ۰,۳ است. از یک مش‌بندی با تعداد ۲۰×۴۰ المان چهار گرهی استفاده شده است.



شکل (۴-۱) دامنه طراحی تیر کنسول

در ادامه به بررسی روش GESO به منظور کمینه کردن وزن سازه پرداخته شده است، بدین منظور پارامتر W_{obj} معرفی شده است، که برابر با نسبت وزن تغییر یافته به وزن اولی است، به عبارت دیگر:

$$W_{obj} = \frac{W}{W_0} \qquad (۱-۴)$$

طول کروموزوم را به ترتیب برابر ۳، ۴، ۵ و ۶ در نظر گرفته می‌گیریم و عدد حساسیت را برای هر طول هنگامی که وزن جسم به ۵۰٪ وزن ابتدایی خود می‌رسد محاسبه می‌کنیم. نتایج بدست آمده در جدول ۴-۱ آورده شده است.

جدول (۴-۱) محاسبه عدد حساسیت شکل (۴-۱) با طول کروموزوم‌های مختلف با W_{obj}=۰,۵

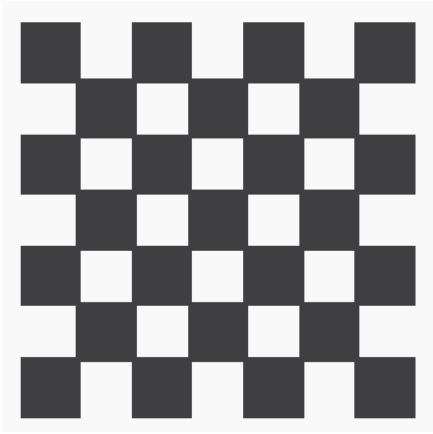
تعداد تکرارها	عدد حساسیت	طول کروزوم
۴۵	$۲,۳۱۰\times۱۰^{-۶}$	۳
۵۸	$۲,۲۸۶\times۱۰^{-۶}$	۴
۸۶	$۲,۳۳۶\times۱۰^{-۶}$	۵
۱۰۱	$۲,۳۵۱\times۱۰^{-۶}$	۶

لازم است اشاره شود عدد حساسیت تعریف شده در این مثال به صورت میانگین عدد حساسیت کل المان‌ها در نظر گرفته شده است.

همان طور که از جدول بالا مشاهده می‌شود با افزایش طول المان تعداد تکرارها افزایش می‌یابد. در حالتی که طول کروموزوم برابر با چهار است، میانگین حساسیت کمتری به ازای تکرار کمتر بدست می‌آید که می‌توان گفت طول بهینه کروموزوم برای روش GESO برابر چهار است.

۴-۳ ارائه مکانیزمی برای جلوگیری از ایجاد حالت شطرنجی (Checkerboard)

المان‌هایی که فقط که از گوشه به یکدیگر متصل شده‌اند مانند شکل (۴-۲) مشکلی است که تحلیل‌گران در بهینه‌سازی سازه به طور معمول با آن مواجه می‌شوند. به این الگو در بهینه‌سازی سازه اصطلاح حالت شطرنجی اطلاق می‌شود. برای ممانعت از رسیدن به حالت شطرنجی از یک روش ساده اما مؤثر که توسط استیون و ژی [۳۸] ارائه شده است، در این تحقیق مورد استفاده قرار می‌گیرد.



شکل (۴-۲) یک الگوی شطرنجی شدن

این روش شامل دو مرحله اصلی می‌باشد:

مرحله اول محاسبه عدد حساسیت برای هر نود می‌باشد که برابر است با میانگین عدد حساسیت المان‌های متصل به هر نود، که به شکل زیر تعریف می‌شود:

$$\alpha_k = \frac{\sum_{i=1}^{n_e} V_i \alpha_i}{\sum_{i=1}^{n_e} V_i} \qquad (۴-۲)$$

که در آن n_e برابر تمام المان‌هایی که متصل به نود k ام می‌باشد و V_i و α_i به ترتیب برابر با حجم و عدد حساسیت المان‌های متصل به نود k ام است.

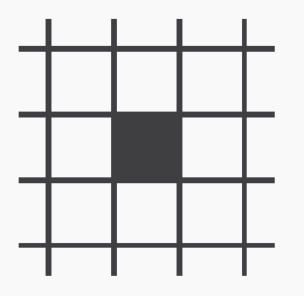
مرحله دو شامل محاسبه فاکتور هموارسازی برای هر المان می‌باشد که این فاکتور برابر است با:

$$\alpha_k = \frac{1}{V_e} \int N_k \, d_k dV \qquad (۳-۴)$$

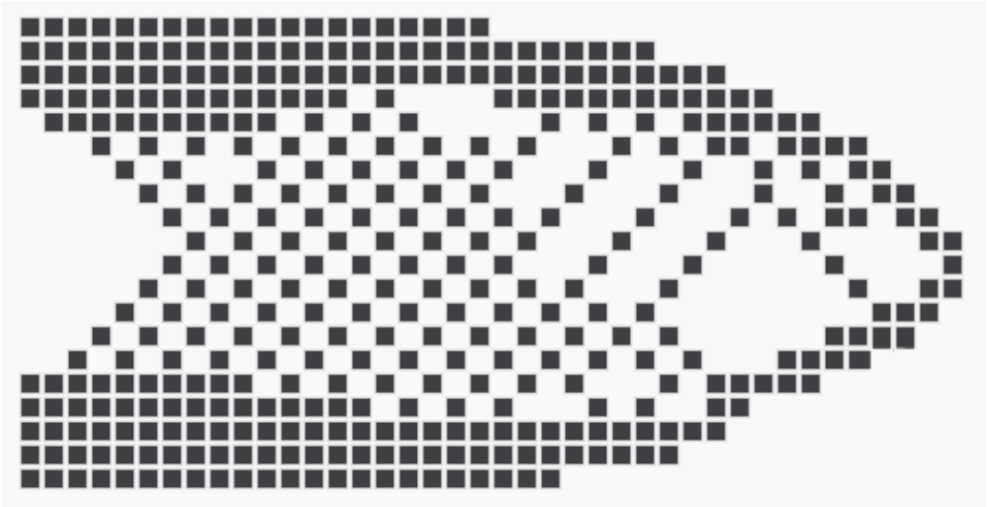
که در آن N_k بردار تابع شکل و V_e حجم المان e امرا مشخص می‌کند. به عنوان مثال برای یک المان مربعی با چهار گره، فاکتور هموارسازی برابر است با:

$$\alpha_e = \frac{1}{n} \sum_{i=1}^{n} \alpha_k \qquad (۴-۴)$$

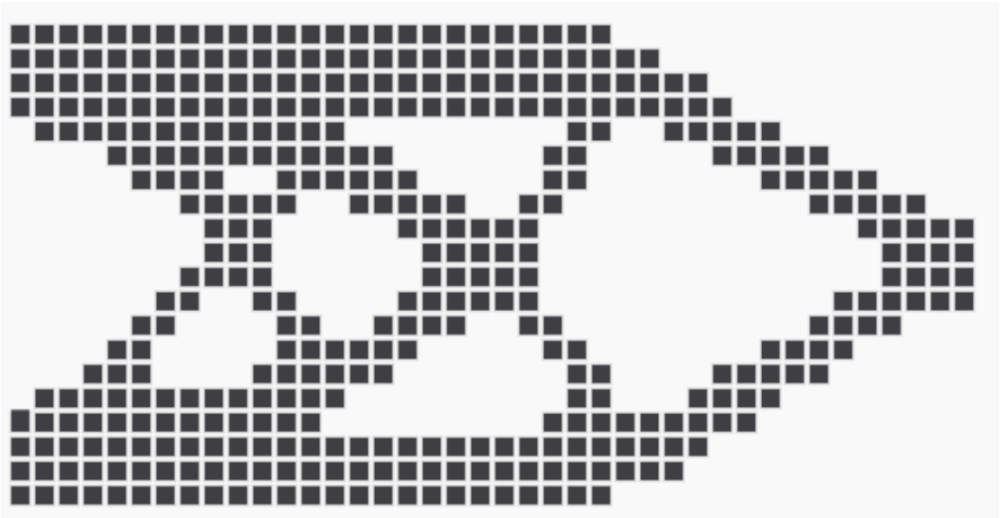
روشی که بالا شرح داده شد در واقع تکنیک هموارسازی مرتبه یک است و عدد حساسیت هر یک از المان‌ها با تأثیرگیری از المان‌های متصل به آن المان محاسبه می‌شود که درشکل (۴-۳) نشان داده شده است.



شکل (۴-۳) المان درگیر در هموارسازی از مرتبه یک با المان منتخب



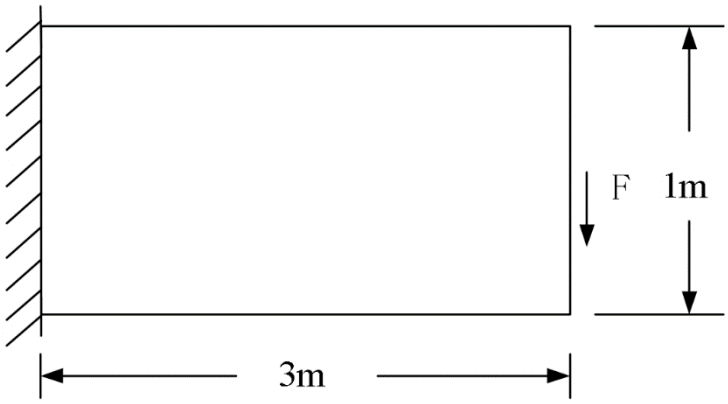
شکل (۴-۴) تیر کنسول بهینه‌سازی شده بدون استفاده از تکنیک هموارسازی



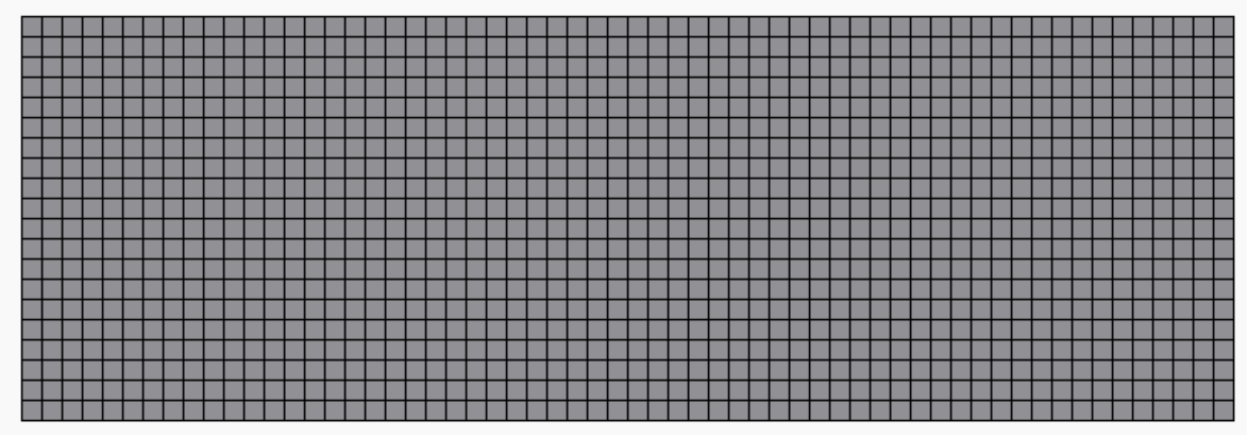
شکل (۴-۵) تیر کنسول بهینه‌سازی شده با استفاده از تکنیک هموارسازی

۴-۴ صحت‌سنجی روش

در مطالعه حاضر به منظور صحت‌سنجی نتایج به بررسی بهینه سازی سختی یک تیر کنسول با روش GESO و مقایسه نتایج با روش BESO که توسط تنگ ژیاو-یان تنگ و تاو جا-وی [۲] انجام گرفته است، پرداخته می‌شود. دامنه طراحی به طول ۳ متر و ارتفاع ۱ متر می باشد. ضخامت برابر ۰,۰۵ متر در نظر گرفته شده است. از یک مش‌بندی با تعداد ۲۰×۶۰ المان چهار گرهی استفاده شده است. مدول الاستیسیته $۲۰۰\ GPa$ و نسبت پواسون ۰,۳ می‌باشد. یک بار دینامیکی به مقدار $۱۰sin\pi t\ N$ در مرکز سر آزاد تیر وارد می‌شود. روند بهینه‌سازی تا رسیدن به ۵۰ درصد حجم اولیه ادامه می‌یابد. ($P_c = ۰,۹$ و $P_m = ۰,۱$)

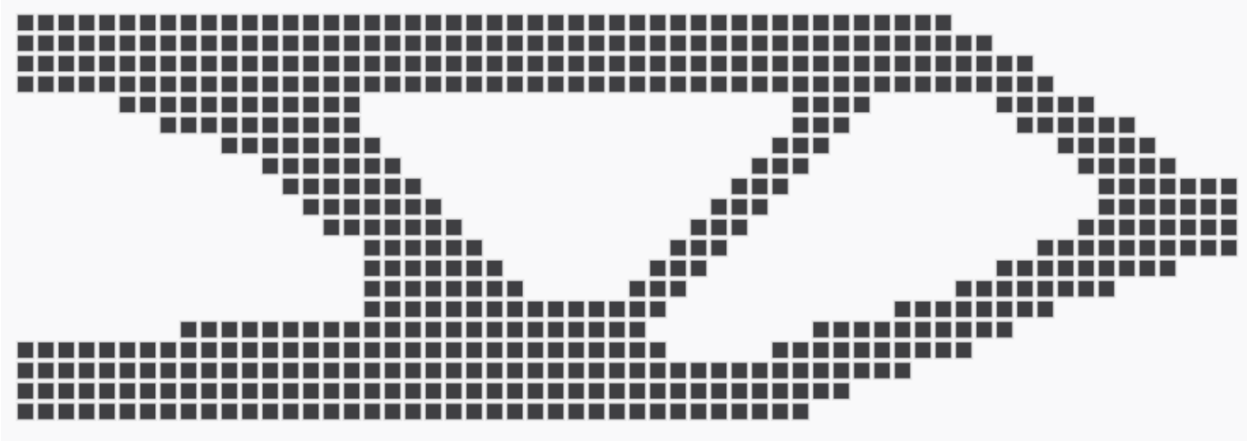


شکل (۴-۶) شرایط تکیه‌گاهی و بارگذاری تیر کنسول

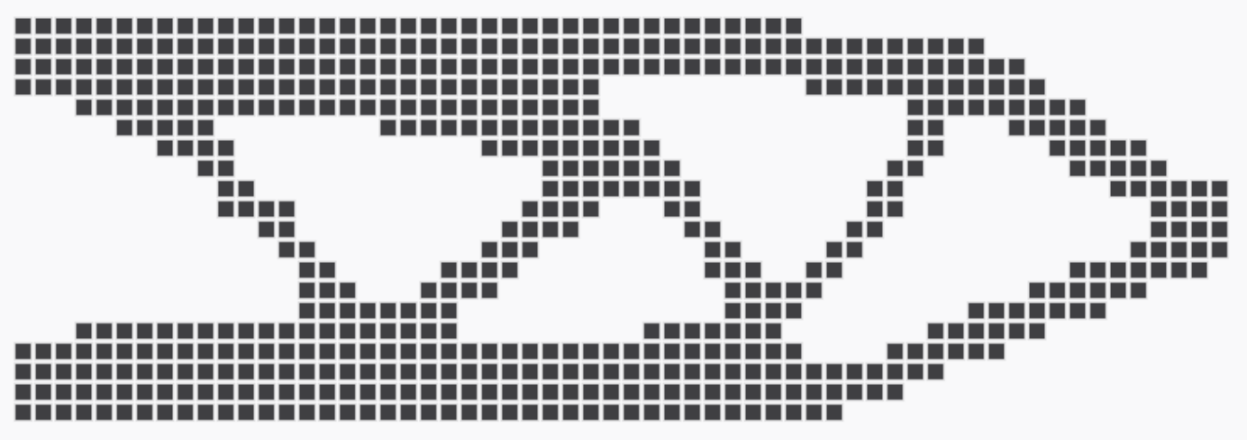


شکل (۴-۷) مدل اجزا محدود تیر کنسول

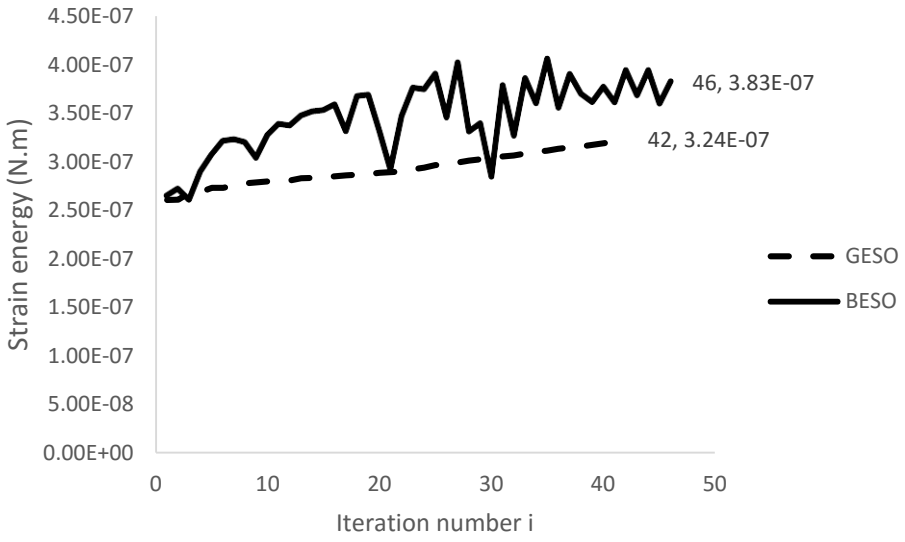
همان‌طور که در بالا مشاهده می‌شود از یک مدل اجزا محدود شامل ۲۰×۶۰ المان چهار گرهی استفاده شده است.



شکل (۴-۸) طرح بهینه تیر کنسول با استفاده از روش BESO



شکل (۴-۹) طرح بهینه تیر کنسول با استفاده از روش GESO



شکل (۴-۱۰) نمودار انرژی کرنشی تیر کنسول

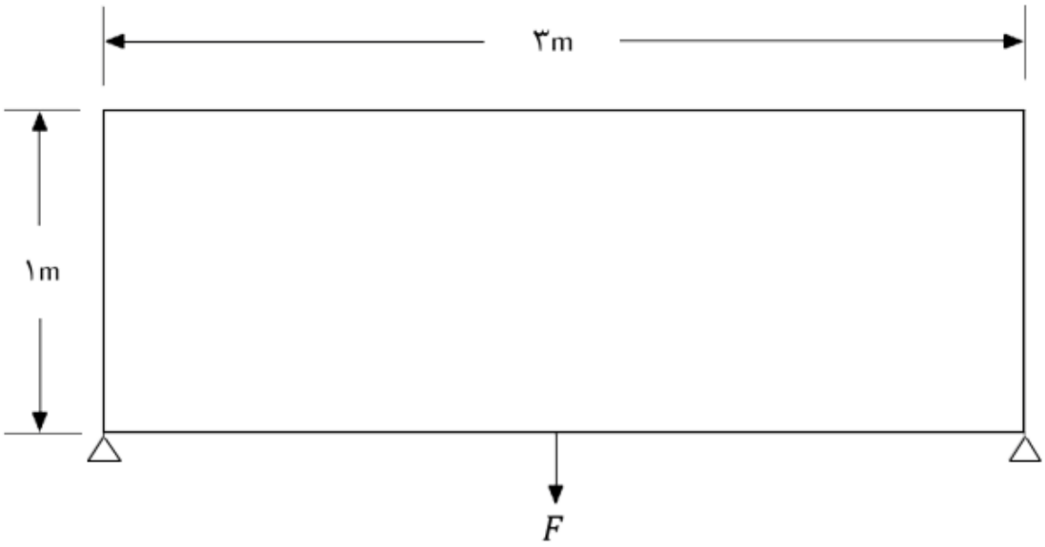
مقدار انرژی کرنشی سازه از مقدار $۳,۸۳\times۱۰^{-۷}$ N.m در روش BESO به مقدار $۳,۲۴\times۱۰^{-۷}$ N.m در روش GESO کاهش پیدا کرده است. حداکثر مقدار جابجایی از $۲,۰۸\times۱۰^{-۷}$ m به $۱,۸۲۵\times۱۰^{-۷}$ m مقدار کاهش داشته است، که نشان دهنده‌ی کاهش در انرژی کرنشی و افزایش سختی سازه می‌باشد. همچنین در مقایسه با سازه‌ی بهینه‌سازی نشده، سازه‌ی بهینه‌سازی شده با ۵۰ درصد مصالح کمتر دارای انرژی کرنشی پایین‌تری می‌باشد، که دلالت بر افزایش سختی سازه دارد. با توجه به نمودار می‌توان مشاهده کرد که روش GESO سیر یکنواختی در روند بهینه‌سازی دارا می‌باشد.

با توجه به نتایج بدست آمده، مشاهده می‌شود که مقدار انرژی کرنشی سازه برای حجم مشخص شده‌ای از مصالح در روش GESO نسبت به روش BESO کاهش پیدا کرده است. در نتیجه با استفاده از روش GESO به توپولوژی سازه‌ای با سختی بیشتر دست پیدا کرده‌ایم.

۴-۵ حل چند مثال با روش GESO و مقایسه نتایج با روش ESO

۴-۵-۱ تیر میشل

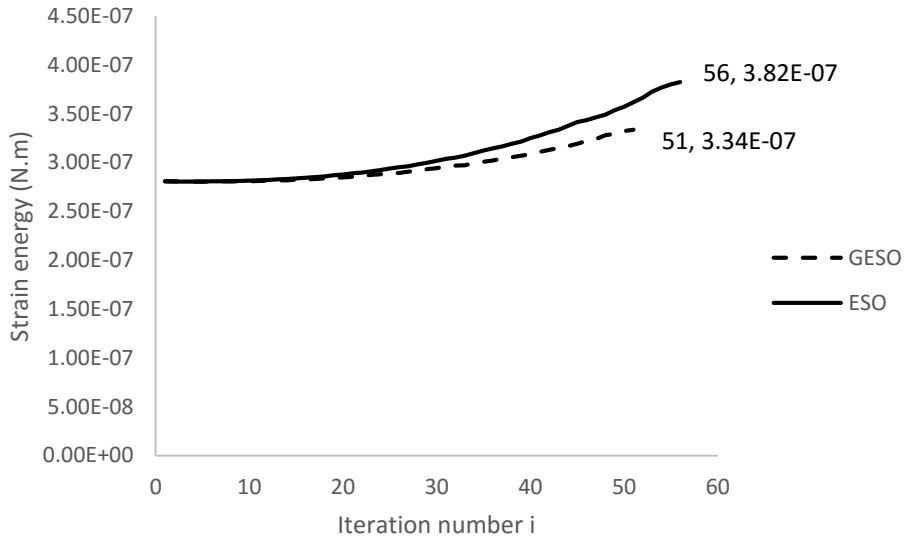
در شکل (۷-۱۲) تیر دو سر مفصلی با ابعاد ۳ m×۱ m و ضخامت ۰,۰۵ m نشان داده شده است. در این مثال از یک مش‌بندی با تعداد ۲۰×۶۰ المان چهار گرهی استفاده شده است. مدول الاستیسیته ۷۰ GPa، نسبت پواسون ۰,۳ و چگالی سازه برابر ۲۷۰۰ Kg/m^3 می‌باشد. یک بار دینامیکی به مقدار $10\sin 50t\, N$ در مرکز فاصله‌ی بین دو تکیه‌گاه بر سازه اعمال می‌شود. روند بهینه‌سازی تا رسیدن به ۵۰ درصد حجم اولیه ادامه می‌یابد. ($P_c = ۰,۹$ و $P_m = ۰,۱$)



شکل (۴-۱۱) شرایط تکیه‌گاهی و بارگذاری تیر میشل

شکل (۴-۱۲) طرح بهینه تیر میشل با استفاده از روش ESO

شکل (۴-۱۳) طرح بهینه تیر میشل با استفاده از روش GESO

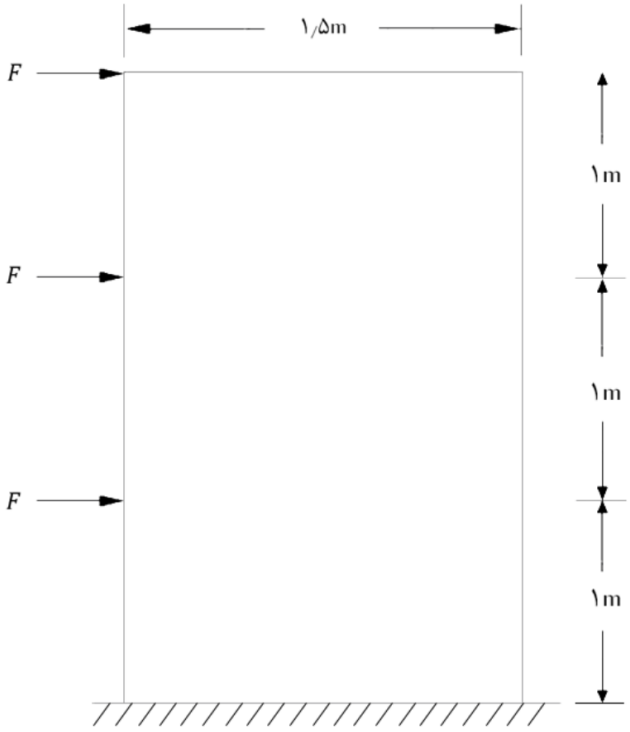


شکل (۴-۱۴) نمودار انرژی کرنشی تیر میشل

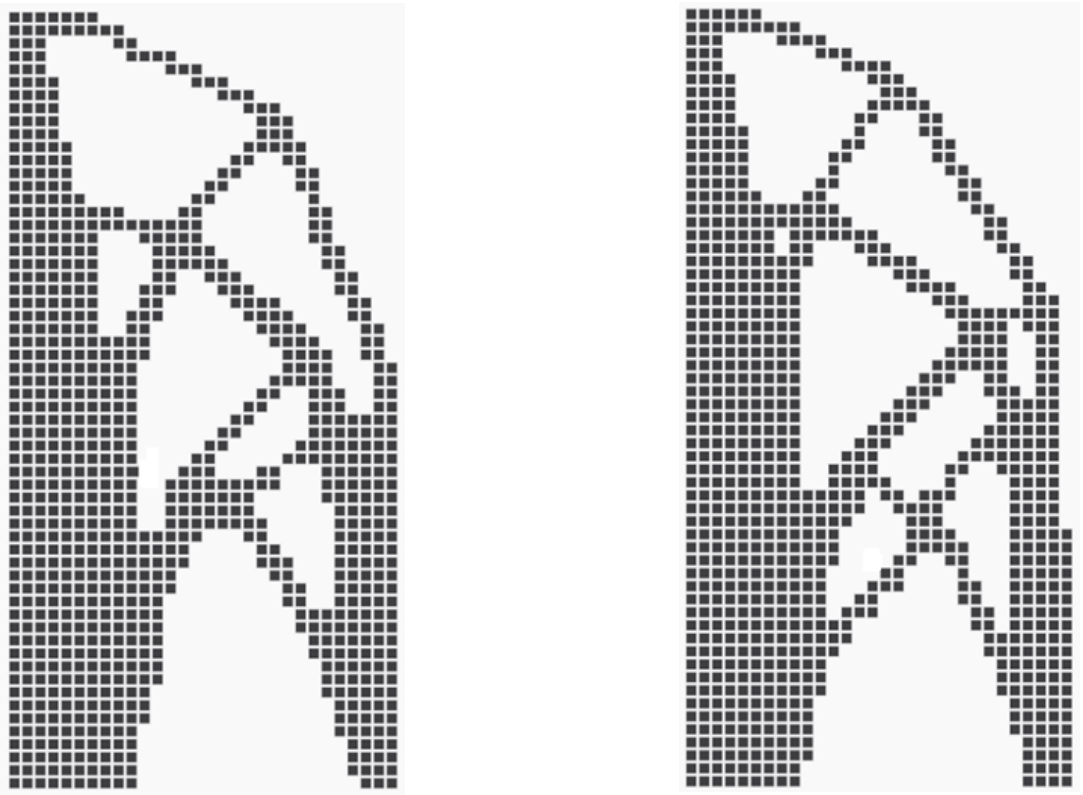
مقدار انرژی کرنشی سازه از مقدار $۳,۰۰\times۱۰^{-۷}$ N.m در روش ESO به مقدار $۲,۷۹\times۱۰^{-۷}$ N.m در روشGESO کاهش پیدا کرده است. همچنین حداکثر مقدار جابجایی از $۳,۸۲۴\times۱۰^{-۸}$ m به $۳,۷۱\times۱۰^{-۸}$ m مقدار کاهش داشته است. مقدار فرکانس طبیعی سازه در حالت بهینه‌سازی شده از مقدار ۱۳۶,۴ rad/s در روش ESO در روش به ۱۴۲,۴ rad/s در روش GESO افزایش پیدا کرده است، و اختلاف بیشتری را نسبت به فرکانس بارگذاری ایجاد کرده است. با توجه به نتایج بالا مشاهده می‌شود که مقدار انرژی کرنشی سازه برای حجم مشخص شده‌ای از مصالح در روش GESO نسبت به ESO کاهش پیدا کرده است و سازه دارای سختی بیشتری می‌باشد.

۴-۵-۲ مثالی از بهینه‌سازی با قید وزن

در این مثال به بررسی سازه‌ای که دارای دامنه‌ی طراحی به طول ۱,۵ m، ارتفاع ۳ m و ضخامت۰,۰۵ m می باشد، می‌پردازیم. از یک مش‌بندی با تعداد ۳۰×۶۰ المان چهار گرهی استفاده شده است. مدول الاستیسیته $۲۰۰GPa$، نسبت پواسون ۰,۳ و چگالی سازه برابر $۷۰۰۰\ Kg/m^3$ می‌باشد. سه بار دینامیکی به مقدار $۱۰۰\sin۱۰۰t\ N$ در ترازهای مشخص شده به سازه اعمال می‌شود. روند بهینه‌سازی تا رسیدن به ۵۰ درصد حجم اولیه ادامه می‌یابد. ($P_c = ۰,۹$ و $P_m = ۰,۱$)

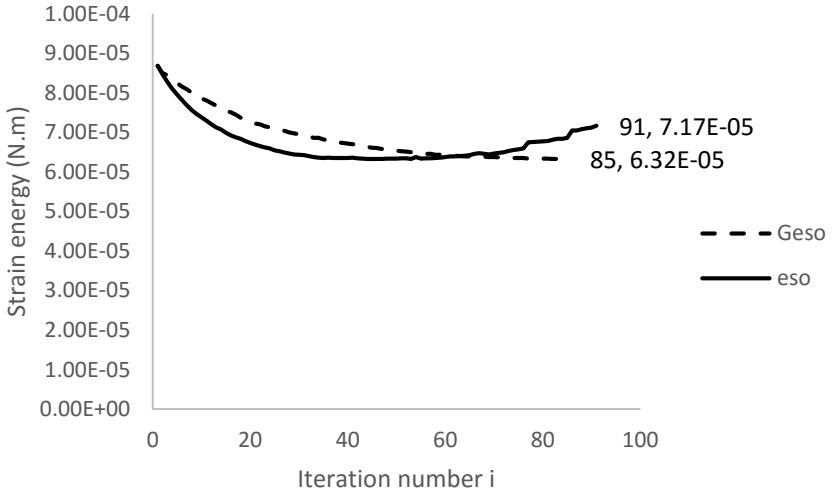


شکل (۴-۱۵) شرایط تکیه‌گاهی و بارگذاری مثال ۴-۵-۲

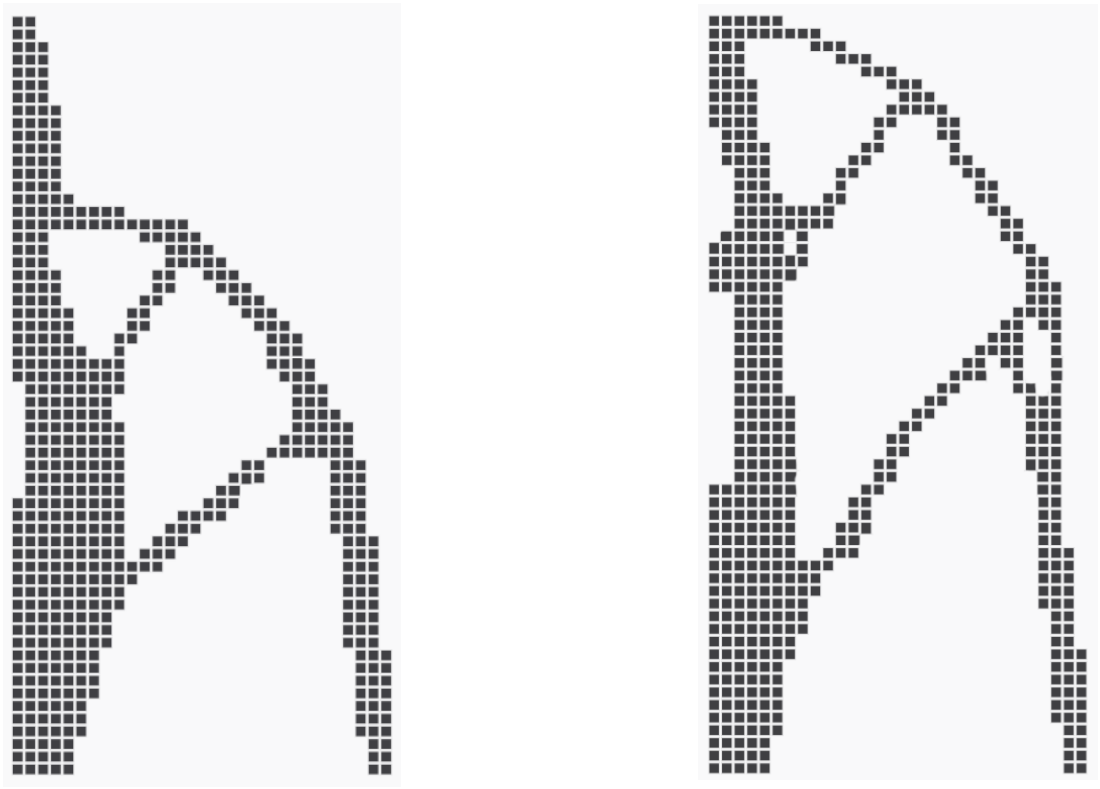


(ب) با استفاده از روش GESO (الف) با استفاده از روش ESO

شکل (۴-۱۶) طرح بهینه سازه تحت قید وزن ٪۵۰.

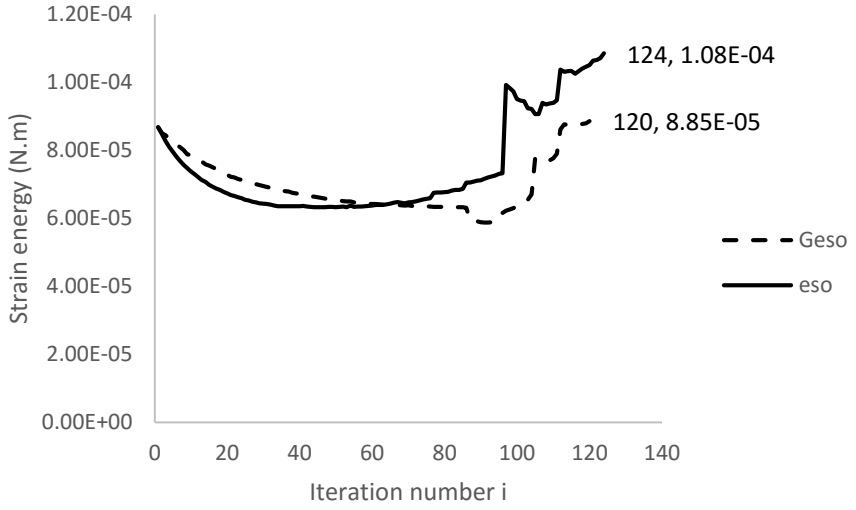


شکل (۴-۱۷) نمودار انرژی کرنشی سازه تحت قید وزن ٪۵۰



(ب) با استفاده از روش GESO (الف) با استفاده از روش ESO

شکل (۴-۱۶) طرح بهینه سازه تحت قید وزن ٪۳۰.



شکل (۴-۱۹) نمودار انرژی کرنشی سازه تحت قید وزن ٪۳۰

همان طور که مشاهده می‌شود مقدار انرژی کرنشی سازه تحت قید ٪۵۰ از مقدار $۷,۱۷\times۱۰^{-۵}$ N.m در روش ESO به مقدار $۶,۳۲\times۱۰^{-۵}$ N.m در روش GESO کاهش پیدا کرده است. همچنین حداکثر مقدار جابجایی از $۱,۷۴۸\times۱۰^{-۷}$ m به $۱,۵۴۱\times۱۰^{-۷}$ m مقدار کاهش داشته است. مقدار فرکانس طبیعی سازه در حالت بهینه‌سازی شده از مقدار ۱۳۴ rad/s در روش ESO در روش به ۱۳۹ rad/s در روش GESO افزایش پیدا کرده است که نسبت به فرکانس بارگذاری که برابر ۱۰۰ rad/s فاصله‌ی بیشتری را ایجاد کرده است. نتایج نشان دهنده‌ی این موضوع است که برای حجم مشخص شده‌ای از مصالح، انرژی کرنشی سازه در روش GESO نسبت به ESO کاهش داشته است.

فصل پنجم

نتیجه‌گیری و پیشنهادات

۵-۱ نتیجه‌گیری

هدف اصلی این پایان‌نامه بررسی روش نوین GESO به منظور بررسی نقاط قوت و ضعف آن، اعمال آن روی تعدادی از مسائل بهینه‌سازی و بررسی و مقایسه نتایج حاصل از این روش با روش‌های گذشته می‌باشد. به منظور اعمال روش‌های مورد استفاده در این تحقیق در مسائل کلاسیک و مهندسی بهینه‌سازی مجموعاً حدود ۴۰۰۰ خط برنامه کامپیوتری به زبان فرترن نوشته شده است. با توجه به نتایج حاصل از مسائل مطرح شده و مطالب ارائه شده و مقایسه جواب‌های بهینه روش حاضر با روش‌های گذشته، این روش، روشی کارا، مطلوب و بسیار مناسب برای یافتن بهینه سراسری است.

در روش GESO مفهوم اساسی بقاء بهترین در الگوریتم ژنتیک با روش ESO ترکیب شد تا عملکرد این روش را در جستجوی پاسخ‌های کلی بهبود ببخشد. این عمل با اضافه کردن عملگرهای الگوریتم ژنتیک، نظیر Mutation، Crossover به ESO انجام می‌گیرد.

مهمترین نتایج حاصل در این تحقیق عبارت است از:

۱- مسائل مورد بررسی قرار گرفته شده نشان دادن که انرژی کرنشی سازه برای حجم مشخصی از مصالح کاهش می‌یابد، در نتیجه با استفاده از روش GESO برای مسائل بهینه‌سازی توپولوژی سازه‌ها تحت بارهای دینامیکی به سازه‌های سخت‌تر دست پیدا خواهیم کرد.

۲- در این تحقیق با بررسی نتایج حاصل و مقایسه با نتایج روش ESO، شاهد ظرفیت بالای روش GESO در جستجوی نتایج بهینه بیشتر می‌باشیم که احتمال رسیدن به بهینگی سراسری را افزایش می‌دهد.

۳- فرکانس طبیعی سازه بهینه سازی با روش GESO نسبت به فرکانس بارگذاری اختلاف بیشتری پیدا می‌کند.

۴- با بررسی نتایج حاصل شاهد آن هستیم که سیر یکنواختی در روند بهینه سازی در مقایسه با روش‌های دیگر حاکم است.

۵-۲ پیشنهادات برای ادامه کار

در زمان انجام این‌کار، زمینه هایی برای ادامه تحقیق در آینده به منظور بهبود نتایج حاصل نمایان شد. پیشنهادات زیر برای ادامه کار ارائه می‌شود :

۱- امکان بازگشت المان‌های کار آمد که از سازه حذف شده‌اند.

۲- تقسم جمعیت به چندین کلاس به گونه‌ای که عملگر تزویج بروی کلاس‌های مشابه انجام شود بدین ترتیب که عملگر تزویج بر روی المان‌هایی که دارای یک رتبه می‌باشند انجام شود.

۳- بررسی چگونگی عملکرد میزان کارآیی روش GESO با استفاده از معیار تنش. روش GESO نیز همانند ESO و BESO قابلیت استفاده از معیار تنش را دارد. در همه مثال‌های این پایان‌نامه از معیار جابه‌جایی استفاده شده است.

منابع

[1] P. W. Christensen and A. Klarbring, *An introduction to structural optimization*. Springer Science & Business Media, 2008.

[2] X. Y. Teng, J. W. Tao, and J. S. Han, "Structural Stiffness Optimization under Dynamic Loads," in *Applied Mechanics and Materials*, 2014, vol. 668, pp. 264-267: Trans Tech Publ.

[3] W. Choi and G. Park, "Transformation of dynamic loads into equivalent static loads based on modal analysis," *International Journal for Numerical Methods in Engineering*, vol. 46, no. 1, pp. 29-43, 1999.

[4] Y. M. Xie and G. P. Steven, "A simple evolutionary procedure for structural optimization," *Computers & structures*, vol. 49, no. 5, pp. 885-896, 1993.

[5] X. Yang, Y. Xie, G. Steven, and O. Querin, "Bidirectional evolutionary method for stiffness optimization," *AIAA journal*, vol. 37, no. 11, 1999.

[6] X. Liu, W.-J. Yi, Q. Li, and P.-S. Shen, "Genetic evolutionary structural optimization," *Journal of constructional steel research*, vol. 64, no. 3, pp. 305-311, 2008.

[7] S. S. Rao and S. S. Rao, *Engineering optimization: theory and practice*. John Wiley & Sons, 2009.

[8] W. R. Spillers and K. M. MacBain, *Structural optimization*. Springer Science & Business Media, 2009.

[9] L. R. Foulds, *Optimization techniques: an introduction*. Springer Science & Business Media, 2012.

[10] M. P. Bendsøe and N. Kikuchi, "Generating optimal topologies in structural design using a homogenization method," *Computer methods in applied mechanics and engineering*, vol. 71, no. 2, pp. 197-224, 1988.

[11] G. Rozvany, "Optimal layout theory," in *Shape and Layout Optimization of Structural Systems and Optimality Criteria Methods*: Springer, 1992, pp. 75-87.

[12] C. Mattheck and S. Burkhardt, "A new method of structural shape optimization based on biological growth," *International Journal of Fatigue*, vol. 12, no. 3, pp. 185-190, 1990.

[13] H. A. Eschenauer, V. V. Kobelev, and A. Schumacher, "Bubble method for topology and shape optimization of structures," *Structural and Multidisciplinary Optimization*, vol. 8, no. 1, pp. 42-51, 1994.

[14] Y. Xie and G. Steven, "Evolutionary structural optimization for dynamic problems," *Computers & Structures*, vol. 58, no. 6, pp. 1067-1073, 1996.

[15] X. Yang, Y. Xie, G. Steven, and O. Querin, "Bidirectional evolutionary method for stiffness optimization," *AIAA journal*, vol. 37, no. 11, 1999.

[16] X. Huang and Y.-M. Xie, "A further review of ESO type methods for topology optimization," *Structural and Multidisciplinary Optimization*, vol. 41, no. 5, pp. 671-683, 2010.

[17] O. Querin, G. Steven, and Y. Xie, "Evolutionary structural optimisation (ESO) using a bidirectional algorithm," *Engineering computations*, vol. 15, no. 8, pp. 1031-1048, 1998.

[18] Y. M. Xie and X. Huang, "Recent developments in evolutionary structural optimization (ESO) for continuum structures," in *IOP Conference Series: Materials Science and Engineering*, 2010, vol. 10, no. 1, p. 012196: IOP Publishing.

[19] J. H. Holland, "Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence," 1975.
[20] M. J. Jakiela, C. Chapman, J. Duda, A. Adewuya, and K. Saitou, "Continuum structural topology design with genetic algorithms," *Computer Methods in Applied Mechanics and Engineering*, vol. 186, no. 2, pp. 339-356, 2000.
[21] M. Mitchell, *An introduction to genetic algorithms*. MIT press, 1998.
[22] M. P. Bendsøe, A. Díaz, and N. Kikuchi, "Topology and generalized layout optimization of elastic structures," in *Topology design of structures*: Springer, 1993, pp. 159-205.
[23] L. H. Tenek and I. Hagiwara, "Static and vibrational shape and topology optimization using homogenization and mathematical programming," *Computer methods in applied mechanics and engineering*, vol. 109, no. 1-2, pp. 143-154, 1993.
[24] Y. Xie and G. Steven, "Evolutionary structural optimization for dynamic problems," *Computers & Structures*, vol. 58, no. 6, pp. 1067-1073, 1996.
[25] D. J. Munk, G. A. Vio, and G. P. Steven, "Topology and shape optimization methods using evolutionary algorithms: a review," *Structural and Multidisciplinary Optimization*, vol. 52, no. 3, pp. 613-631, 2015.
[26] M. P. Bendsøe, "Optimal shape design as a material distribution problem," *Structural and multidisciplinary optimization*, vol. 1, no. 4, pp. 193-202, 1989.
[27] M. Zhou and G. Rozvany, "The COC algorithm, Part II: topological, geometrical and generalized shape optimization," *Computer Methods in Applied Mechanics and Engineering*, vol. 89, no. 1-3, pp. 309-336, 1991.
[28] A. Rietz, "Sufficiency of a finite exponent in SIMP (power law) methods," *Structural and Multidisciplinary Optimization*, vol. 21, no. 2, pp. 159-163, 2001.
[29] D. E. Goldberg and M. P. Samtani, "Engineering optimization via genetic algorithm," in *Electronic computation*, 1986, pp. 471-482: ASCE.
[30] X. Huang and Y. Xie, "Convergent and mesh-independent solutions for the bi-directional evolutionary structural optimization method," *Finite Elements in Analysis and Design*, vol. 43, no. 14, pp. 1039-1049, 2007.
[31] O. Sigmund and J. Petersson, "Numerical instabilities in topology optimization: a survey on procedures dealing with checkerboards, mesh-dependencies and local minima," *Structural and Multidisciplinary Optimization*, vol. 16, no. 1, pp. 68-75, 1998.
[32] M. Bendsøe, "e Sigmund, O.(2003). Topology Optimization-Theory, Methods and Applications," ed: Springer, Berlin.
[33] G. W. Jang, J. H. Jeong, Y. Y. Kim, D. Sheen, C. Park, and M. N. Kim, "Checkerboard-free topology optimization using non- conforming finite elements," *International Journal for Numerical Methods in Engineering*, vol. 57, no. 12, pp. 1717-1735, 2003.
[34] Q. Li, G. Steven, and Y. Xie, "A simple checkerboard suppression algorithm for evolutionary structural optimization," *Structural and Multidisciplinary Optimization*, vol. 22, no. 3, pp. 230-239, 2001.
[35] H. Kajiwara, T. Hiroyasu, M. Miki, and A. Hashimoto, "Topology structural optimization using a hybrid of GA and ESO methods," in *Computational Intelligence*, 2006, pp. 84-89.
[36] X. Liu and W.-J. Yi, "Michell-like 2D layouts generated by genetic ESO," *Structural and Multidisciplinary Optimization*, vol. 42, no. 1, pp. 111-123, 2010.
[37] Z. Zuo, Y. Xie, and X. Huang, "Combining genetic algorithms with BESO for topology optimization," *Structural and multidisciplinary optimization*, vol. 38, no. 5, pp. 511-523, 2009.

[38] Q. Li, G. Steven, and Y. Xie, "A simple checkerboard suppression algorithm for evolutionary structural optimization," *Structural and Multidisciplinary Optimization*, vol. 22, no. 3, pp. 230-239, 2001.
[39] S. Min, N. Kikuchi, Y. Park, S. Kim, and S. Chang, "Optimal topology design of structures under dynamic loads," *Structural optimization*, vol. 17, no. 2-3, pp. 208-218, 1999.
[40] Y. Xie and G. Steven, "Evolutionary structural optimization for dynamic problems," *Computers & Structures*, vol. 58, no. 6, pp. 1067-1073, 1996.
[41] T. T. Feng, J. Arora, and E. Haug, "Optimal structural design under dynamic loads," *International Journal for Numerical Methods in Engineering*, vol. 11, no. 1, pp. 39-52, 1977.
[42] Y. Xie and G. Steven, "A simple approach to structural frequency optimization," *Computers & structures*, vol. 53, no. 6, pp. 1487-1491, 1994.
[43] X. Yang, Y. Xie, G. Steven, and O. Querin, "Topology optimization for frequencies using an evolutionary method," *Journal of Structural Engineering*, vol. 125, no. 12, pp. 1432-1438, 1999.
[44] Z. Zuo, Y. Xie, and X. Huang, "An improved bi-directional evolutionary topology optimization method for frequencies," *International Journal of Structural Stability and Dynamics*, vol. 10, no. 01, pp. 55-75, 2010.
[45] Z. H. Zuo, Y. M. Xie, and X. Huang, "Optimal topological design of periodic structures for natural frequencies," *Journal of Structural Engineering*, vol. 137, no. 10, pp. 1229-1240, 2011.

پیوست

کد فرترن مثال ۴-۵-۲

```
implicit none
integer  n,m,i,j,jj,ii,k,kk,zz,yy,ca,jn,jm,jl,jo,jp,jq,nr,rr,rr1,rr2,ijk,dl,dn,ti,ma,ijk3,jq2,n2,m2,jq3
integer  h,mn,nr2,nr3,arl,bn,bn1,nbn,ai1,ai3,ai2,hs2,ma1,ma2,ma3,mp(7,1800),maa(60,30),aa2
real*8 x,y,lx(4,2),w(4),ll(8),j_1(2,2),deter,df(2,4),b(3,8),ke(8,8),kg,c(3,3),e,v,n1,m1,s,s2,kek,mr,ar,xx
real*8   z1,z2,z3,du(8),dd(8),hs,ai,aii,nen(8,8),ng,nn(2,8),da,d2,ww,ww2,t,p,zh(2,5),q,aa,mm
real*8   kh(2,1800),kh2(8),tt,pqp,cc,cmk,ww2,d,c2,b3,d3,b2,bn2,pqp2
allocatable  ca(:,:),kg(:,:),rr(:,:),rr1(:),rr2(:),s(:,:),s2(:,:),kek(:,:,:),ma(:,:),hs2(:),ma2(:,:),ma3(:,:)
allocatable  dn(:),dl(:),bn(:),bn1(:),h(:,:),hs(:,:),ai(:,:),ai2(:,:),ai3(:,:),aii(:,:),ng(:,:),d2(:,:),ai1(:,:),cc(:,:)
allocatable  d(:),c2(:,:),b3(:,:),d3(:),b2(:,:),bn2(:),ma1(:,:,:)
allocatable  cmk(:,:,:)
data ll /-1,-1,1,-1,1,1,-1,1/
ll=ll*0.5773502691896258
data w /1,1,1,1/

open(unit=1,file='in.txt',action="read")
open(unit=2,file='out.txt',action="write")
open(unit=3,file='out2.txt',action="write")
open(unit=4,file='out3.txt',action="write")
open(unit=5,file='out4.txt',action="write")
open(unit=6,file='out5.txt',action="write")
open(unit=7,file='out6.txt',action="write")
open(unit=8,file='out7.txt',action="write")
open(unit=9,file='out8.txt',action="write")
read(1,*)mr,ar
read(1,*)
read(1,*)m,n
ti=2*(m+1)*(n+1);mn=m*n;jm=(mr*mn*(1-ar))
allocate(ca(m+1,n+1),ma(m,n),kg(ti,ti),h((m+1)*(n+1),5),s2(m*n,2),ma1(m,n,2),ma2(mn,5),ma3(3,4))
allocate(kek(8,m*n,8),s(m*n,12),ng(ti,ti),d2(4,ti),hs2(mn*4),bn(mn*4),bn1(mn*8),cc(ti,ti),cmk(ti,5,ti))
allocate(hs((m+1)*(n+1),2),ai(m*n,25),ai2(m*n,5),ai3(m*n,6),aii(m*n,25),ai1(m*n,13),rr2(ti),dl(mn)
allocate (d(ti),c2(2,ti),b3(ti,ti),d3(ti),b2(ti,ti),bn2(ti),dn(mn*4)))
read(1,*)
read(1,*)x,y,t
read(1,*)
read(1,*)e,v,da
   c=0.
       c(1,1)=e/(1-(v**2))
       c(2,2)=e/(1-(v**2))
       c(1,2)=(e*v)/(1-(v**2))
       c(2,1)=(e*v)/(1-(v**2))
       c(3,3)=e/(2*(1+v))
read(1,*)
read(1,*) nr,nr2
allocate(rr(nr,17),rr1(nr*4))
rr=0;rr1=0
```

```
read(1,*) (rr(1,j),j=2,16,2)
read(1,*)  rr(1,1)
do i=2,nr
   rr(i,1)=rr(i-1,1)+1
   do j=2,16,2
     rr(i,j)=rr(1,j)
   enddo
      enddo
n1=0
jl=1
       do i=1,m
  m1=0
  do j=1,n
      ai(jl,16)=m1          ;ai(jl,17)=n1
      ai(jl,18)=m1+x        ;ai(jl,19)=n1
      ai(jl,20)=m1+x        ;ai(jl,21)=n1+y
      ai(jl,22)=m1          ;ai(jl,23)=n1+y
      m1=m1+x
      jl=jl+1
  enddo
  n1=n1+y
enddo
ma2=1
jm=1
do i=1,mn
     ai(i,24)=1
                  ai(i,1)=i
     ma2(i,5)=i
     jm=jm+1
enddo
jm=m+1;jl=1
       do i=1,m+1
  do j=1,n+1
                  ca(jm,j)=jl
     jl=jl+1
  enddo
  jm=jm-1
enddo
jm=m;jl=1
do i=1,m
  jn=1
  do j=1,n
      ai(jl,2)=jm
      ai(jl,3)=jn
      jl=jl+1
      jn=jn+1
  enddo
  jm=jm-1
enddo
```

```
jm=m;jn=1
do i=1,m
  do j=1,n
       ai(jn,4)=ca(jm+1,j)
       ai(jn,5)=ca(jm+1,j+1)
       ai(jn,6)=ca(jm,j+1)
       ai(jn,7)=ca(jm,j)
       jn=jn+1
  enddo
  jm=jm-1
enddo
do i=1,mn
                do j=4,7
                        ai(i,2*j)=2*ai(i,j)-1
                        ai(i,2*j+1)=2*ai(i,j)
                enddo
enddo
nen=0
jl=1
do jj=1,4
   call mass (nn,ll(jl:jl+1))
   do ii=1,8
     do kk=1,8
        do j=1,2
                                nen(ii,kk)=nen(ii,kk)+nn(j,ii)*nn(j,kk)
        enddo
     enddo
      enddo
   jl=jl+2
enddo
nen=nen*x*y*da*t
ma=1;ijk3=0
ww=143;d2=0;ijk=1;xx=87;aa=16807;mm=2147483647;arl=1800

while (ijk3==0) do
kg=0;kek=0;ng=0;lx=0;h=0;hs=0;s=0;rr1=0;rr2=0;d2=0;ai2=0;ai3=0;dl=0;dn=0;s2=0;mp=0
call vava2 (ai,aii,ai1,bn,bn1,ti,nbn,nr3,rr,rr2,m,n,mn,arl,nr2,nr)

if (arl<=900) then
   ijk3=1
endif
do i=1,arl
       jm=16
            do j=1,4
                    lx(j,1)=ai(i,jm)
               lx(j,2)=ai(i,jm+1)
          jm=jm+2
            enddo
       jl=1
          do jj=1,4
```

```
                call jacobian_matrix (lx,j_1,df,deter,ll(jl:jl+1))
                    call bmatrix (b,j_1,df)
                    call kelemet (ke,b,c,deter,w(jj),t)
          do ii=1,8
                                do kk=1,8
                                   kek(ii,i,kk)=kek(ii,i,kk)+ke(ii,kk)
                                enddo
                 enddo
        jl=jl+2
            enddo
        do ii=1,8
                            do kk=1,8
                                    kg(ai1(i,ii),ai1(i,kk))=kg(ai1(i,ii),ai1(i,kk))+kek(ii,i,kk)
                            enddo
        enddo
        do ii=1,8
                                do kk=1,8
                                    ng(ai1(i,ii),ai1(i,kk))=ng(ai1(i,ii),ai1(i,kk))+nen(ii,kk)
                                enddo
        enddo
        enddo
call vava (kg,ti,nbn,nr3,rr2,ng,ww,ai1,m,n,ijk,arl)

ww2=100;tt=1
do i=1,nbn*2
   do j=1,nbn*2
       b2(i,j)=kg(i,j)-ng(i,j)*((ww2)**2)
   enddo
enddo
jm=1
do i=1,nbn*2
       jn=1
       if (rr2(i)/=0) then
          do j=1,nbn*2
             if (rr2(j)/=0) then
                  b3(jm,jn)=b2(i,j)
                  jn=jn+1
             endif
          enddo
          jm=jm+1
       endif
enddo
jm=571
do kk=1,3
aa2=nbn*2-nr3
jl=0
do j=1,arl
do i=1,arl
  if (ai1(i,13)==jm.and.jl==0) then
     ii=ai1(i,8);jl=1
```

```
   endif
 enddo
 enddo
 jj=0
 do i=1,ii
   if (rr2(i)==0) then
      jj=jj+1
   endif
 enddo
 b3(ii-jj,aa2+1)=(100)*sin(ww2*tt)
 jm=jm+600
 enddo
 do i=1,aa2-1
      if (b3(i,i)==0) then
         jm=0;jn=0;c2=0
         do j=i+1,aa2
            if (b3(j,i)/=0.and.jm/=1) then
                  jn=j
                  jm=1
             endif
         enddo
         if (jn/=0) then
           do j=1,aa2+1
                                  c2(1,j)=b3(i,j)
                 c2(2,j)=b3(jn,j)
           enddo
           do j=1,aa2+1
                                  b3(jn,j)=c2(1,j)
                 b3(i,j)=c2(2,j)
           enddo
         endif
      endif
      if (b3(i,i)/=0) then
         p=b3(i,i)
         do j=i+1,aa2
            if (b3(j,i)/=0) then
                 q=b3(j,i)/p
                 do k=i,aa2+1
                                     b3(j,k)=b3(j,k)-b3(i,k)*q
                 enddo
             endif
         enddo
      endif
 enddo
 if (b3(aa2,aa2)/=0) then
     d(aa2)=b3(aa2,aa2+1)/b3(aa2,aa2)
     elseif (b3(aa2,aa2)==0) then
     d(aa2)=0
 endif
 jm=aa2-1
```

```
do i=1,aa2-1
    p=0;jn=aa2
    do j=1,aa2-jm
       if (b3(jm,jn)/=0) then
                        p=p+b3(jm,jn)*d(jn)
       endif
       jn=jn-1
    enddo
    if (b3(jm,jm)/=0) then
         d(jm)=(b3(jm,aa2+1)-p)/b3(jm,jm)
      elseif (b3(jm,jm)==0) then
         d(jm)=0
    endif
    jm=jm-1
enddo
jn=1
do i=1,nbn*2
       if (rr2(i)/=0) then
                                d2(1,i)=d(jn)
           jn=jn+1
       endif
enddo

do i=1,nbn*2
       if (d2(1,i)>0) then
                                d2(4,i)=d2(1,i)
       endif
       if (d2(1,i)<=0) then
                                d2(4,i)=d2(1,i)*(-1)
       endif
enddo
do i=1,arl-1
           do j=i+1,arl
              z1=0;z2=0
              if (d2(4,i)<d2(4,j)) then
                                   z1=d2(4,i);z2=d2(4,j)
                          d2(4,i)=z2;d2(4,j)=z1
              endif
            enddo
       enddo
write(3,*) "   dddd    ",d2(4,1)
write(3,*) "**************************************************************************",ijk
      cmk=0
      do i=1,nbn*2
                 do j=1,nbn*2
                                cmk(i,1,j)=kg(i,j)-((ww2)**2)*ng(i,j)
              enddo
         enddo
      p=0
      do i=1,nbn*2
```

```
                do j=1,nbn*2
                        d2(2,i)=d2(2,i)+d2(1,j)*cmk(j,1,i)
            enddo
         enddo
        do i=1,nbn*2
                        p=p+d2(1,i)*d2(2,i)
           enddo
        if (ijk==1) then
                pqp=p
        endif
      write(3,*)'   cobj  ',arl,ww,(arl*p)/(mn*pqp),p
      write(8,*) p
      cmk=0
      do i=1,nbn*2
                do j=1,nbn*2
                        cmk(i,1,j)=kg(i,j)
            enddo
         enddo
       p=0
        do i=1,nbn*2
                do j=1,nbn*2
                        d2(3,i)=d2(3,i)+d2(1,j)*cmk(j,1,i)
            enddo
           enddo
        do i=1,nbn*2
                        p=p+d2(1,i)*d2(3,i)*0.5
            enddo
        if (ijk==1) then
                pqp2=p
        endif
      write(3,*)'   cobj2  ',arl,ww,(arl*p)/(mn*pqp2),p
      write(9,*) p
write(3,*)''ma''
do i=1,m
       write(3,'(30I3)')  (ma(i,j),j=1,n)
enddo
if (ijk3==0) then
s=0
do k=1,arl
      dd=0
      du=0
      do i=1,8
                        du(i)=d2(1,ai1(k,i))
           enddo
      do i=1,8
                  do j=1,8
                        dd(i)=dd(i)+du(j)*kek(j,k,i)*ai(k,24)
            enddo
           enddo
      do i=1,8
```

```
                                s(k,1)=s(k,1)+dd(i)*du(i)*0.5
      enddo
      s(k,2)=ai(k,1)
      s(k,12)=ai(k,1)
      s(k,11)=s(k,1)
enddo

do i=1,nbn
    h(i,1)=bn(i)
    jn=2
    do j=1,arl
                 do k=4,7
                          if (ai(j,k)==bn(i)) then
                    h(i,jn)=ai(j,1)
                    jn=jn+1
            endif
              enddo
    enddo
enddo
do i=1,arl-1
                     do j=i+1,arl
                z1=0;z2=0
                if (s(i,1)<s(j,1)) then
                                        z1=s(i,1);z2=s(j,1)
                   s(i,1)=z2;s(j,1)=z1
                   z1=s(i,2);z2=s(j,2)
                   s(i,2)=z2;s(j,2)=z1
                endif
             enddo
enddo

do i=1,arl
            s(i,3)=s(i,1)
            s(i,4)=s(i,2)
enddo
do i=1,nbn
       p=0
       do j=2,5
         if(h(i,j)/=0) then
           do k=1,arl
            if(s(k,4)==h(i,j)) then
              hs(i,1)=hs(i,1)+s(k,3)/4
              p=p+1
            endif
           enddo
         endif
       enddo
       hs(i,2)=p
enddo
do i=1,arl
```

```
        s(i,6)=ai1(i,13)
        do j=9,12
            jm=ai1(i,j)
            s(i,5)=s(i,5)+hs(jm,1)*hs(jm,2)
        enddo
    enddo
    do i=1,arl-1
                    do j=i+1,arl
                z1=0;z2=0
                if (s(i,5)<s(j,5)) then
                                    z1=s(i,5);z2=s(j,5)
                   s(i,5)=z2;s(j,5)=z1
                   z1=s(i,6);z2=s(j,6)
                   s(i,6)=z2;s(j,6)=z1
                endif
             enddo
    enddo
    write(7,*)
"**************************************************************************************",ijk
    maa=0
    do i=1,arl
        jm=s(i,2)
        do j=1,arl
          if (ai(j,1)==jm) then
                jo=ai(j,2);jp=ai(j,3)
             if (i<(arl*1)/4) then
                maa(jo,jp)=1
              elseif (i<(arl*2)/4) then
                maa(jo,jp)=2
              elseif (i<(arl*3)/4) then
                maa(jo,jp)=3
              elseif (i<=(arl*4)/4) then
                maa(jo,jp)=4
             endif
          endif
        enddo
    enddo
    do i=1,m
    write(7,'(30I3)')  (maa(i,j),j=1,n)
    enddo
    write(7,*)
"**************************************************************************************",ijk
    maa=0
    do i=1,arl
        jm=s(i,4)
        do j=1,arl
          if (ai(j,1)==jm) then
                jo=ai(j,2);jp=ai(j,3)
             if (i<(arl*1)/4) then
                maa(jo,jp)=1
```

```
                elseif (i<(arl*2)/4) then
                   maa(jo,jp)=2
                elseif (i<(arl*3)/4) then
                   maa(jo,jp)=3
                elseif (i<=(arl*4)/4) then
                   maa(jo,jp)=4
              endif
         endif
       enddo
   enddo
   do i=1,m
   write(7,'(30I3)')  (maa(i,j),j=1,n)
   enddo
   write(7,*)
"*************************************************************************************",ijk
   maa=0
   do i=1,arl
       jm=s(i,6)
       do j=1,arl
         if (ai(j,1)==jm) then
                jo=ai(j,2);jp=ai(j,3)
            if (i<(arl*1)/4) then
                maa(jo,jp)=1
              elseif (i<(arl*2)/4) then
                maa(jo,jp)=2
              elseif (i<(arl*3)/4) then
                maa(jo,jp)=3
              elseif (i<=(arl*4)/4) then
                maa(jo,jp)=4
            endif
         endif
       enddo
   enddo
   do i=1,m
   write(7,'(30I3)')  (maa(i,j),j=1,n)
   enddo

   if (arl-900>10) then
   z3=10
   elseif (arl-900<=10) then
          z3=arl-900
   endif

   jq2=0;jq3=0
   while (jq3==0) do
   if (arl-z3*5+1>900) then
         jp=arl-z3*5+1
       elseif (arl-z3*5+1<=900) then
         jp=901
   endif
```

```
do i=jp,arl
      xx=mod((aa*xx),mm)
      q=(xx/mm)
      s(i,7)=q
      s(i,8)=s(i,6)
enddo
do i=jp,arl-1
        do j=i+1,arl
               z1=0;z2=0
               if (s(i,7)<s(j,7)) then
                  z1=s(i,7);z2=s(j,7)
                  s(i,7)=z2;s(j,7)=z1
                  z1=s(i,8);z2=s(j,8)
                  s(i,8)=z2;s(j,8)=z1
               endif
           enddo
       enddo
jq=1
do i=jp,arl,2
    if (i+1<=arl) then
     xx=mod((aa*xx),mm)
     q=(xx/mm);ma3=0
        ma3(1,1)=1
        ma3(1,2)=0
        ma3(1,3)=1
        ma3(1,4)=0
        ma3(1,5)=1
        ma3(1,6)=0
       jm=s(i,8);jn=s(i+1,8)
       do j=1,4
          if (ma3(1,j)==1) then
              ma3(2,j)=ma2(jm,j)
              ma3(3,j)=ma2(jn,j)
           elseif (ma3(1,j)==0) then
              ma3(3,j)=ma2(jm,j)
              ma3(2,j)=ma2(jn,j)
          endif
       enddo
       do j=1,4
            ma2(jm,j)=ma3(2,j)
            ma2(jn,j)=ma3(3,j)
       enddo
       mp(1,jq)=jm
       mp(2,jq)=jn
       jq=jq+1
    endif
enddo

do i=arl-z3+1,arl
      xx=mod((aa*xx),mm)
```

```
      q=(xx/mm)
      s(i,7)=q
      s(i,8)=s(i,6)
enddo
do i=arl-z3+1,arl-1
            do j=i+1,arl
               z1=0;z2=0
               if (s(i,7)<s(j,7)) then
                  z1=s(i,7);z2=s(j,7)
                  s(i,7)=z2;s(j,7)=z1
                  z1=s(i,8);z2=s(j,8)
                  s(i,8)=z2;s(j,8)=z1
               endif
            enddo
enddo
jm=arl
do i=1,z3
  jo=s(jm,8);jl=0
  if (ma2(jo,5)/=0) then
    xx=mod((aa*xx),mm)
    q=(xx/mm)
    zh=0;p=0
       do j=1,4
          if (ma2(jo,j)==0) then
             zh(1,j)=0
           elseif (ma2(jo,j)==1) then
             p=p+1
             zh(1,j)=p
          endif
       enddo
       do j=1,4
          if (zh(1,j)==0) then
             zh(2,j)=0
           elseif (zh(1,j)>0) then
             zh(2,j)=(1/p)*zh(1,j)
          endif
       enddo
       xx=mod((aa*xx),mm)
       q=(xx/mm)
       do j=1,4
          if (q<=zh(2,j).and.jl==0) then
             ma2(jo,j)=0
             jl=1
             mp(3,jp)=jo
             jp=jp+1
          endif
       enddo
  endif
  jm=jm-1
enddo
```

```
jl=1;jm=arl
do i=1,arl
   if (arl-jq2>900) then
   q=0;p=0;jo=s(i,6)
   if (ma2(jo,5)/=0) then
     do j=1,4
       if (ma2(jo,j)==0) then
         q=q+1
       endif
     enddo
   endif
   if (q==4) then
       p=ma2(jo,5)
       ii=ma2(jo,5)
       ma2(jo,5)=0
       mp(4,jl)=p
       jl=jl+1
       do j=1,arl
         if (ai(j,1)>0) then
           if (ai(j,1)==p) then
              jo=ai(j,2);jp=ai(j,3)
                  ma(jo,jp)=0
              write(3,*) "  1   "
              write(3,*) ai(j,1)
              ai(j,1)=ai(j,1)*(-1)
              jq2=jq2+1

           endif
         endif
       enddo
    endif
    endif
enddo
if (jq2>0) then
      jq3=1
endif
if (jq2==20) then
      jq3=1
   elseif (jq2==40) then
      jq3=1
   elseif (jq2==60) then
      jq3=1
   elseif (jq2==80) then
      jq3=1
endif
do i=1,nr
   do j=1,jl-1
     if (rr(i,1)/=0.and.rr(i,1)==mp(4,j)) then
                 rr(i,1)=0
      endif
```

```
   enddo
enddo
endwhile
ma1=0
jn=m;jl=1
do i=1,m
  do j=1,n
                ma1(jn,j,1)=jl
     jl=jl+1
  enddo
  jn=jn-1
enddo

jl=1
ma1=0
do i=1,m
     do j=1,n
       jm=0;jn=0
       if (ma(i,j)==1) then
            if (i>1.and.j>1) then
               if (ma(i-1,j-1)==1) then
                          jm=jm+1
               endif
            endif
            if (i>1) then
              if (ma(i-1,j)==1) then
                          jm=jm+2
              endif
            endif
            if (i>1.and.j<n) then
              if (ma(i-1,j+1)==1) then
                          jm=jm+4
              endif
            endif
            if (j>1) then
              if (ma(i,j-1)==1) then
                          jm=jm+8
              endif
            endif
            /*****************************************************
            if (j<n) then
               if (ma(i,j+1)==1) then
                          jn=jn+1
               endif
            endif
            if (i<m.and.j>1) then
              if (ma(i+1,j-1)==1) then
                          jn=jn+2
              endif
            endif
```

```
if (i<m) then
  if (ma(i+1,j)==1) then
                jn=jn+4
  endif
endif
if (i<m.and.j<n) then
  if (ma(i+1,j+1)==1) then
                jn=jn+8
  endif
endif
/*************************************      1
if (jm==1.and.jn==0) then
                ma1(i,j,2)=1
    elseif (jm==2.and.jn==0) then
                ma1(i,j,2)=1
    elseif (jm==4.and.jn==0) then
                ma1(i,j,2)=1
    elseif (jm==8.and.jn==0) then
                ma1(i,j,2)=1

    elseif (jm==0.and.jn==1) then
                ma1(i,j,2)=1
    elseif (jm==0.and.jn==2) then
                ma1(i,j,2)=1
    elseif (jm==0.and.jn==4) then
                ma1(i,j,2)=1
    elseif (jm==0.and.jn==8) then
                ma1(i,j,2)=1

    elseif (jm==0.and.jn==0) then
                ma1(i,j,2)=1

    elseif (jm==3.and.jn==0) then
                ma1(i,j,2)=1
    elseif (jm==6.and.jn==0) then
                ma1(i,j,2)=1
    elseif (jm==9.and.jn==0) then
                ma1(i,j,2)=1
    elseif (jm==7.and.jn==0) then
                ma1(i,j,2)=1

    elseif (jm==0.and.jn==9) then
                ma1(i,j,2)=1
    elseif (jm==0.and.jn==6) then
                ma1(i,j,2)=1
    elseif (jm==0.and.jn==12) then
                ma1(i,j,2)=1
    elseif (jm==0.and.jn==14) then
                ma1(i,j,2)=1
```

```
elseif (jm==4.and.jn==1) then
            ma1(i,j,2)=1
elseif (jm==4.and.jn==9) then
            ma1(i,j,2)=1
elseif (jm==8.and.jn==2) then
            ma1(i,j,2)=1
elseif (jm==9.and.jn==2) then
            ma1(i,j,2)=1

/*******************************************    2

elseif (jm==1.and.jn==8) then
            ma1(i,j,2)=4
elseif (jm==4.and.jn==2) then
            ma1(i,j,2)=4
 elseif (jm==2.and.jn==4) then
            ma1(i,j,2)=4
elseif (jm==8.and.jn==1) then
            ma1(i,j,2)=4

elseif (jm==1.and.jn==1) then
            ma1(i,j,2)=4
elseif (jm==1.and.jn==4) then
            ma1(i,j,2)=4
elseif (jm==12.and.jn==0) then
            ma1(i,j,2)=4
elseif (jm==4.and.jn==4) then
            ma1(i,j,2)=4
elseif (jm==2.and.jn==2) then
            ma1(i,j,2)=4
elseif (jm==0.and.jn==3) then
            ma1(i,j,2)=4
elseif (jm==2.and.jn==8) then
            ma1(i,j,2)=4
elseif (jm==8.and.jn==8) then
            ma1(i,j,2)=4
/*******************************************    3


elseif (jm==10.and.jn==0) then
            ma1(i,j,2)=4
elseif (jm==2.and.jn==1) then
            ma1(i,j,2)=4
elseif (jm==8.and.jn==4) then
            ma1(i,j,2)=4
elseif (jm==0.and.jn==5) then
            ma1(i,j,2)=4
/*******************************************    4
```

```
elseif (jm==3.and.jn==1) then
            ma1(i,j,2)=4
elseif (jm==3.and.jn==4) then
            ma1(i,j,2)=4
elseif (jm==3.and.jn==8) then
            ma1(i,j,2)=4

elseif (jm==14.and.jn==0) then
            ma1(i,j,2)=4
elseif (jm==6.and.jn==4) then
            ma1(i,j,2)=4
elseif (jm==6.and.jn==2) then
            ma1(i,j,2)=4

elseif (jm==9.and.jn==1) then
            ma1(i,j,2)=4
elseif (jm==9.and.jn==4) then
            ma1(i,j,2)=4
elseif (jm==9.and.jn==8) then
            ma1(i,j,2)=4

elseif (jm==12.and.jn==1) then
            ma1(i,j,2)=4
elseif (jm==4.and.jn==5) then
            ma1(i,j,2)=4
elseif (jm==4.and.jn==3) then
            ma1(i,j,2)=4
/************************************     5

elseif (jm==2.and.jn==6) then
            ma1(i,j,2)=4
elseif (jm==0.and.jn==7) then
            ma1(i,j,2)=4
elseif (jm==4.and.jn==6) then
            ma1(i,j,2)=4

elseif (jm==1.and.jn==12) then
            ma1(i,j,2)=4
elseif (jm==2.and.jn==12) then
            ma1(i,j,2)=4
elseif (jm==8.and.jn==12) then
            ma1(i,j,2)=4

elseif (jm==10.and.jn==2) then
            ma1(i,j,2)=4
elseif (jm==12.and.jn==2) then
            ma1(i,j,2)=4
elseif (jm==8.and.jn==3) then
            ma1(i,j,2)=4
```

```
elseif (jm==1.and.jn==9) then
            ma1(i,j,2)=4
elseif (jm==8.and.jn==9) then
            ma1(i,j,2)=4
elseif (jm==2.and.jn==9) then
            ma1(i,j,2)=4

/*************************************     6

elseif (jm==3.and.jn==12) then
            ma1(i,j,2)=4
elseif (jm==6.and.jn==6) then
            ma1(i,j,2)=4
elseif (jm==9.and.jn==9) then
            ma1(i,j,2)=4
elseif (jm==12.and.jn==3) then
            ma1(i,j,2)=4

/*************************************     7
elseif (jm==7.and.jn==2) then
            ma1(i,j,2)=4
elseif (jm==7.and.jn==4) then
            ma1(i,j,2)=4
elseif (jm==7.and.jn==8) then
            ma1(i,j,2)=4
elseif (jm==7.and.jn==6) then
            ma1(i,j,2)=4
elseif (jm==7.and.jn==12) then
            ma1(i,j,2)=4
elseif (jm==7.and.jn==14) then
            ma1(i,j,2)=4

elseif (jm==1.and.jn==14) then
            ma1(i,j,2)=4
elseif (jm==2.and.jn==14) then
            ma1(i,j,2)=4
elseif (jm==4.and.jn==14) then
            ma1(i,j,2)=4
elseif (jm==3.and.jn==14) then
            ma1(i,j,2)=4
elseif (jm==6.and.jn==14) then
            ma1(i,j,2)=4

elseif (jm==13.and.jn==2) then
            ma1(i,j,2)=4
elseif (jm==9.and.jn==3) then
            ma1(i,j,2)=4
elseif (jm==9.and.jn==10) then
            ma1(i,j,2)=4
elseif (jm==13.and.jn==3) then
```

```
                    ma1(i,j,2)=4
                elseif (jm==9.and.jn==11) then
                    ma1(i,j,2)=4
                elseif (jm==13.and.jn==11) then
                    ma1(i,j,2)=4

                elseif (jm==5.and.jn==9) then
                    ma1(i,j,2)=4
                elseif (jm==12.and.jn==9) then
                    ma1(i,j,2)=4
                elseif (jm==4.and.jn==11) then
                    ma1(i,j,2)=4
                elseif (jm==13.and.jn==9) then
                    ma1(i,j,2)=4
                elseif (jm==12.and.jn==11) then
                    ma1(i,j,2)=4
            endif
        endif
   enddo
enddo
write(4,*)"ma1"
do i=1,m
        write(4,'(20I3)')  (ma1(i,j,2),j=1,n)
enddo
jl=0
do i=1,m
     do j=1,n
         if (ma(i,j)==1.and.ma1(i,j,2)==1) then
          do k=1,arl
             if (ai(k,2)==i.and.ai(k,3)==j) then
                jq2=jq2+1
               if (arl-jq2>900) then
                jo=ai(k,1)
                ai(k,1)=ai(k,1)*(-1)
               endif
             endif
          enddo
          if (arl-jq2>900) then
          write(3,*) "  2   "
          write(3,*) jo
          do k=1,5
             ma2(jo,k)=0
          enddo
          jl=jl+1
          mp(5,jl)=jo
          ma(i,j)=0
          endif
         endif

         if (ma(i,j)==1.and.ma1(i,j,2)==4) then
```

```
          do k=1,arl
             if (ai(k,2)==i.and.ai(k,3)==j) then
                ai(k,24)=2
                write(3,*) "  ee   ",ai(k,1)
             endif
          enddo
         endif

     enddo
enddo
do k=1,arl
             if (ai(k,2)==10.and.ai(k,3)==571) then
               /*  ai(k,24)=4
                write(3,*) "  ee   ",ai(k,1)
             endif
             if (ai(k,2)==11.and.ai(k,3)==1171) then
               /* ai(k,24)=4
                write(3,*) "  ee   ",ai(k,1)
             endif
enddo
write(3,'(50I4)')  (mp(5,j),j=1,jl-1)
do i=1,nr
   do j=1,jl-1
     if (rr(i,1)/=0.and.rr(i,1)==mp(5,j)) then
                 rr(i,1)=0
      endif
    enddo
enddo
write(3,'(50I4)')  (mp(5,j),j=1,jl-1)
do i=1,nr
   do j=1,jl-1
     if (rr(i,1)/=0.and.rr(i,1)==mp(5,j)) then
                 rr(i,1)=0
      endif
    enddo
enddo

jq2=0;ma1=0
do kk=1,4
if (kk==1) then
     ii=1;m2=m;n2=n
  elseif (kk==2) then
     ii=m;m2=m;n2=n
  elseif (kk==3) then
     jj=n;m2=n;n2=m
  elseif (kk==4) then
     jj=1;m2=n;n2=m
endif
do i=1,m2
     if (kk==1) then
```

```
   jj=1
elseif (kk==2) then
   jj=n
elseif (kk==3) then
   ii=1
elseif (kk==4) then
   ii=m
endif
do j=1,n2
  kh2=0;jm=0
  if (ma(ii,jj)==1) then
       if (ii>1.and.jj>1) then
          if (ma(ii-1,jj-1)==1) then
                       if (ma1(ii-1,jj-1,2)/=0) then
                       jm=jm+1
                       kh2(jm)=ma1(ii-1,jj-1,2)
                       endif
           endif
       endif
       if (ii>1) then
         if (ma(ii-1,jj)==1) then
                       if (ma1(ii-1,jj,2)/=0) then
                       jm=jm+1
                       kh2(jm)=ma1(ii-1,jj,2)
                       endif
         endif
       endif
       if (ii>1.and.jj<n) then
         if (ma(ii-1,jj+1)==1) then
                       if (ma1(ii-1,jj+1,2)/=0) then
                       jm=jm+1
                       kh2(jm)=ma1(ii-1,jj+1,2)
                       endif
         endif
       endif
       if (jj>1) then
         if (ma(ii,jj-1)==1) then
                       if (ma1(ii,jj-1,2)/=0) then
                       jm=jm+1
                       kh2(jm)=ma1(ii,jj-1,2)
                       endif
         endif
       endif
       /*****************************************************
       if (jj<n) then
          if (ma(ii,jj+1)==1) then
                       if (ma1(ii,jj+1,2)/=0) then
                       jm=jm+1
                       kh2(jm)=ma1(ii,jj+1,2)
                       endif
```

```
            endif
          endif
          if (ii<m.and.jj>1) then
            if (ma(ii+1,jj-1)==1) then
                        if (ma1(ii+1,jj-1,2)/=0) then
                        jm=jm+1
                        kh2(jm)=ma1(ii+1,jj-1,2)
                        endif
            endif
          endif
          if (ii<m) then
            if (ma(ii+1,jj)==1) then
                        if (ma1(ii+1,jj,2)/=0) then
                        jm=jm+1
                        kh2(jm)=ma1(ii+1,jj,2)
                        endif
            endif
          endif
          if (ii<m.and.jj<n) then
            if (ma(ii+1,jj+1)==1) then
                        if (ma1(ii+1,jj+1,2)/=0) then
                        jm=jm+1
                        kh2(jm)=ma1(ii+1,jj+1,2)
                        endif
            endif
          endif
          /*****************************************
          if (jm==0) then
                          jq2=jq2+1
                          ma1(ii,jj,2)=jq2
              elseif (jm/=0) then
                if (jm/=0) then
                do zz=1,jm-1
                              do yy=zz+1,jm
                    z1=0;z2=0
                    if (kh2(zz)>kh2(yy)) then
                                          z1=kh2(zz);z2=kh2(yy)
                       kh2(zz)=z2;kh2(yy)=z1
                    endif
                     enddo
                      enddo
                endif
                          ma1(ii,jj,2)=kh2(1)
          endif
        endif
if (kk==1) then
  jj=jj+1
elseif (kk==2) then
  jj=jj-1
elseif (kk==3) then
```

```
      ii=ii+1
    elseif (kk==4) then
      ii=ii-1
    endif
    enddo
    if (kk==1) then
      ii=ii+1
    elseif (kk==2) then
      ii=ii-1
    elseif (kk==3) then
      jj=jj-1
    elseif (kk==4) then
      jj=jj+1
    endif
  enddo
  write(4,*)"ma1"
  do i=1,m
        write(4,'(30I3)')  (ma1(i,j,2),j=1,n)
  enddo
  enddo

  jl=0;kh=0
  do ii=1,jq2
     jm=0
         do i=1,m
        do j=1,n
          if (ma1(i,j,2)==ii) then
             jm=jm+1
          endif
        enddo
     enddo

     if (jm/=0) then
              jl=jl+1
              kh(1,jl)=ii
              kh(2,jl)=jm
              write(2,*)ii,jm,jl
              write(2,*)kh(1,jl),kh(2,jl)
     endif

  enddo
  do i=1,jl-1
             do j=i+1,jl
                z1=0;z2=0
                if (kh(2,i)<kh(2,j)) then
                                    z1=kh(2,i);z2=kh(2,j)
                   kh(2,i)=z2;kh(2,j)=z1
                   z1=kh(1,i);z2=kh(1,j)
                   kh(1,i)=z2;kh(1,j)=z1
                endif
```

```
            enddo
enddo
write(2,*)kh(1,1),kh(2,1),kh(1,2),kh(2,2)
jl=0
do i=1,m
     do j=1,n
         if (ma(i,j)==1.and.ma1(i,j,2)/=kh(1,1)) then
          do k=1,arl
             if (ai(k,2)==i.and.ai(k,3)==j) then
                jq2=jq2+1
                if (arl-jq2>900) then
                jo=ai(k,1)
                ai(k,1)=ai(k,1)*(-1)
               endif
             endif
          enddo
          if (arl-jq2>900) then
          write(3,*) "  3   "
          write(3,*) jo
          do k=1,5
             ma2(jo,k)=0
          enddo
          jl=jl+1
          mp(5,jl)=jo
          ma(i,j)=0
         endif
         endif
     enddo
enddo
write(3,'(50I4)')  (mp(5,j),j=1,jl-1)
do i=1,nr
  do j=1,jl-1
    if (rr(i,1)/=0.and.rr(i,1)==mp(5,j)) then
                rr(i,1)=0
     endif
   enddo
enddo
/*************************************************************************************
call vava2 (ai,aii,ai1,bn,bn1,ti,nbn,nr3,rr,rr2,m,n,mn,arl,nr2,nr)
do i=1,arl
   q=0;jl=ai(i,1)
   if (ma2(jl,5)/=0) then
       do j=1,4
         if (ma2(jl,j)==0) then
            q=q+1
         endif
       enddo
              if (q==0) then
                 ai(i,24)=1
              elseif (q==1) then
```

```
                ai(i,24)=1
              elseif (q==2) then
                ai(i,24)=1
              elseif (q==3) then
                ai(i,24)=1
              endif
      endif
   enddo

   write(4,*) "**************************************************************************",ijk
   write(4,*)"ma"
   do i=1,m
   write(4,'(30I3)')  (ma(i,j),j=1,n)
   enddo
   write(5,*) "**************************************************************************",ijk
   write(5,*)"rr2"
   write(5,'(20I5)')  (rr2(j),j=1,nbn*2)
   write(5,*)"bn"
   write(5,'(100I4)')  (bn(i),i=1,nbn)
   write(5,*)"bn1"
   write(5,'(100I4)')  (bn1(i),i=1,nbn*2)
   write(5,*)"mp"
   do i=1,7
   write(5,'(50I4)')  (mp(i,j),j=1,1000)
   enddo
   write(6,*) "**************************************************************",ijk,ww
   write(6,*) "ai1"
   do i=1,arl
   write(6,*)  (ai1(i,j),j=1,13)
   enddo
   write(6,*)"ai2"
   do i=1,arl
   write(6,'(12I4)')  (ai2(i,j),j=1,5)
   enddo
   write(6,*)"hs2"
   write(6,*)"ai3"
   do i=1,zz
   write(6,'(12I4)')  (ai3(i,j),j=1,6)
   enddo
   ijk=ijk+1
   endif
   ijk=ijk+1
endwhile

end

subroutine jacobian_matrix (lx,j_1,df,deter,ll)
implicit none
real*8 :: j_1(2,2),deter,lx(4,2),j(2,2),ll(2),df(2,4)
j_1=0;df=0
```

```
        df(1,1)=0.25*(ll(2)-1)
        df(1,2)=0.25*(1-ll(2))
        df(1,3)=0.25*(1+ll(2))
        df(1,4)=-0.25*(1+ll(2))
        df(2,1)=0.25*(ll(1)-1)
        df(2,2)=-0.25*(1+ll(1))
        df(2,3)=0.25*(1+ll(1))
        df(2,4)=0.25*(1-ll(1))
        j(1,1)=0.25*(lx(1,1)*(ll(2)-1)+lx(2,1)*(1-ll(2))+lx(3,1)*(1+ll(2))+lx(4,1)*(-1-ll(2)))
        j(1,2)=0.25*(lx(1,2)*(ll(2)-1)+lx(2,2)*(1-ll(2))+lx(3,2)*(1+ll(2))+lx(4,2)*(-1-ll(2)))
        j(2,1)=0.25*(lx(1,1)*(ll(1)-1)+lx(2,1)*(-1-ll(1))+lx(3,1)*(1+ll(1))+lx(4,1)*(1-ll(1)))
        j(2,2)=0.25*(lx(1,2)*(ll(1)-1)+lx(2,2)*(-1-ll(1))+lx(3,2)*(1+ll(1))+lx(4,2)*(1-ll(1)))
        deter=j(1,1)*j(2,2)-j(2,1)*j(1,2)
        j_1(1,1)=j(2,2)/deter
        j_1(2,2)=j(1,1)/deter
        j_1(1,2)=-1*j(1,2)/deter
        j_1(2,1)=-1*j(2,1)/deter
        endsubroutine jacobian_matrix
        subroutine bmatrix (b,j_1,df)
        integer::k,l,m,j
        real*8::j_1(2,2),df(2,4),b(3,8),bb(2,4)
        bb=0;b=0
        do k=1,2
                do l=1,4
                        do m=1,2
                                bb(k,l)=bb(k,l)+j_1(k,m)*df(m,l)
                        enddo
                enddo
        enddo
        do j=1,4
                b(1,2*j-1)=bb(1,j)
                b(3,2*j-1)=bb(2,j)
                b(2,2*j)=bb(2,j)
                b(3,2*j)=bb(1,j)
        enddo
endsubroutine bmatrix

subroutine kelemet (ke,b,c,deter,w,t)
        real*8::b(3,8),c(3,3),ke(8,8),kep(8,3),deter,w,t
        kep=0;ke=0
        do k=1,8
                do l=1,3
                        do m=1,3
                                kep(k,l)=kep(k,l)+b(m,k)*c(m,l)
                        enddo
                enddo
        enddo
        do k=1,8
                do l=1,8
                        do m=1,3
```

```
                                    ke(k,l)=ke(k,l)+kep(k,m)*b(m,l)*deter*w*t
                                enddo
                            enddo
                    enddo
endsubroutine kelemet

subroutine vava (kg,ti,nbn,nr3,rr2,ng,ww,ai1,m,n,ijk,arl)
        integer::i,j,k,aa,ti,rr2(ti),nbn,nr3,jn,jm,m,n,ijk,endd,ai1(m*n,13),arl,ww4
        real*8::c,p,q,d,kg(ti,ti),ng(ti,ti),b,ww,d3,b2,bn2,ww3,q3,ww5(3)
  allocatable  d(:),c(:,:),b(:,:),d3(:),b2(:,:),bn2(:)
  aa=nbn*2-nr3
  allocate (d(aa),c(2,aa+1),b(aa,aa+1),d3(aa),b2(aa+nr3,aa+nr3),bn2(aa))
  q=0;p=0;ww3=0;q3=0;endd=0;ww4=0;ww5=0;ww4=0
  write(2,*) "**************************          ", ijk ,"          **************************"
  rt=0
  if (arl==1800) then
  ww=50;rt=1
  elseif (arl==900) then
      rt=1;ww=50
  endif
  if (rt==1) then
  OO=1
  while (endd==0) do
  d=0;b=0;c=0;b2=0;bn2=0;d3=0
  do i=1,nbn*2
     do j=1,nbn*2
                    b2(i,j)=kg(i,j)-ng(i,j)*(ww**2)
     enddo
  enddo
  jm=1
  do i=1,nbn*2
       jn=1
       if (rr2(i)/=0) then
          do j=1,nbn*2
             if (rr2(j)/=0) then
                b(jm,jn)=b2(i,j)
                jn=jn+1
             endif
          enddo
          jm=jm+1
       endif
  enddo

  do i=1,aa-1
      if (b(i,i)==0) then
         jm=0;jn=0
         do j=i+1,aa
            if (b(j,i)/=0.and.jm/=1) then
                  jn=j
                  jm=1
```

```
          endif
        enddo
        if (jn/=0) then
          do j=1,aa
                                c(1,j)=b(i,j)
              c(2,j)=b(jn,j)
          enddo
          do j=1,aa
                                b(jn,j)=c(1,j)
              b(i,j)=c(2,j)
          enddo
        endif
      endif
      if (b(i,i)/=0) then
        p=b(i,i)
        do j=i+1,aa
          if (b(j,i)/=0) then
              q=b(j,i)/p
              do k=i,aa
                                  b(j,k)=b(j,k)-b(i,k)*q
              enddo
           endif
        enddo
      endif
enddo
jm=0
do i=1,aa
if (b(i,i)>0) then
                jm=jm+1
                d3(jm)=b(i,i)
endif
enddo
do i=1,jm-1
                    do j=i+1,jm
             z1=0;z2=0
             if (d3(i)>d3(j)) then
                                  z1=d3(i);z2=d3(j)
                d3(i)=z2;d3(j)=z1
             endif
           enddo
        enddo
ww5(3)=ww5(2)
ww5(2)=ww5(1)
ww5(1)=d3(1)
        write(2,*)d3(1)
        q=d3(1)
    if (q3==0.and.ww4==0) then
        if (q>200000) then
          ww=ww+7;ww3=7
        elseif (100000<q.and.q<=200000) then
```

```
          ww=ww+6;ww3=6
        elseif (60000<q.and.q<=100000) then
          ww=ww+5;ww3=5
        elseif (30000<q.and.q<=60000) then
          ww=ww+4;ww3=4
        elseif (15000<q.and.q<=30000) then
          ww=ww+3;ww3=3
        elseif (7500<q.and.q<=15000) then
          ww=ww+2;ww3=2
        elseif (q<=7500) then
          ww=ww+1;ww3=1
        endif
        write(2,*) 1
  elseif (q3/=0) then
  if (q<q3.and.ww4==0) then
      if (q>200000) then
          ww=ww+7;ww3=7
        elseif (100000<q.and.q<=200000) then
          ww=ww+6;ww3=6
        elseif (60000<q.and.q<=100000) then
          ww=ww+5;ww3=5
        elseif (30000<q.and.q<=60000) then
          ww=ww+4;ww3=4
        elseif (15000<q.and.q<=30000) then
          ww=ww+3;ww3=3
        elseif (7500<q.and.q<=15000) then
          ww=ww+2;ww3=2
        elseif (q<=7500) then
          ww=ww+1;ww3=1
        endif
        write(2,*) 2
   elseif (q>q3.and.ww4==0) then
          ww=ww-ww3
          ww4=1
        write(2,*) 3
   endif
   endif
   if (ww4==1) then
     write(2,*) 4
          ww=ww+0.2
          ww4=2
   elseif (ww4==2) then
     if (ww5(3)>=ww5(1)) then
       write(2,*) 5
          ww=ww+0.2
          ww4=3
     elseif (ww5(3)<ww5(1)) then
          write(2,*) 6
          ww4=4
          ww=ww-0.4
```

```
            ww5(1)=ww5(3)
          endif
        elseif (ww4==3) then
          if (ww5(2)>=ww5(1)) then
           write(2,*) 7
            ww=ww+0.2
          elseif (ww5(2)<ww5(1)) then
            write(2,*) 8
            endd=1
            ww=ww-0.2
          endif
         elseif (ww4==4) then
          if (ww5(2)>=ww5(1)) then
           write(2,*) 9
            ww=ww-0.2
          elseif (ww5(2)<ww5(1)) then
            write(2,*) 10
            endd=1
            ww=ww+0.2
          endif
        endif
        write(2,*)  OO,ww
    q3=q
    OO=OO+1
    endwhile
    endif
endsubroutine vava

subroutine vava2 (ai,aii,ai1,bn,bn1,ti,nbn,nr3,rr,rr2,m,n,mn,arl,nr2,nr)
integer::i,j,k,ti,arl,nbn,nr3,jn,jm,jl,m,n,ai1(m*n,13),rr2(ti),bn(mn*4),bn1(mn*8),rr(nr,17),rr1(nr*4),z1,z2
real*8::ai(m*n,25),aii(m*n,25)
 rr1=0;rr2=0;aii=0;ai1=0;arl=0

    do i=1,mn
      if (ai(i,1)>0) then
          arl=arl+1
          do j=1,25
               aii(arl,j)=ai(i,j)
          enddo
        endif
    enddo
    jm=arl
    do i=1,mn
      if (ai(i,1)<0) then
          jm=jm+1
          do j=1,25
               aii(jm,j)=ai(i,j)
          enddo
        endif
    enddo
```

```
ai=0
do i=1,mn
      do j=1,25
          ai(i,j)=aii(i,j)
      enddo
enddo
aii=0
jm=0;bn=0
do i=1,arl
      do j=4,7
          jm=jm+1
          bn(jm)=ai(i,j)
      enddo
enddo
do i=1,jm-1
      do j=i+1,jm
        if (bn(i)==bn(j)) then
            bn(j)=0
        endif
      enddo
enddo
do i=1,jm-1
   if (bn(i)==0) then
       jl=0
       do j=i+1,jm
         if (bn(j)/=0.and.jl==0) then
            bn(i)=bn(j)
            bn(j)=0
            jl=1
         endif
       enddo
   endif
enddo
nbn=0
do i=1,jm
  if (bn(i)/=0) then
        nbn=nbn+1
  endif
enddo
do i=1,nbn-1
     do j=i+1,nbn
         if (bn(i)>bn(j)) then
            z1=bn(i);z2=bn(j)
            bn(i)=z2;bn(j)=z1
         endif
     enddo
enddo
bn1=0
jm=1
do i=1,nbn
```

```
            bn1(jm)=2*bn(i)-1
            bn1(jm+1)=2*bn(i)
            jm=jm+2
enddo
ai1=0
do i=1,arl
   jn=9
   do j=4,7
      jm=1
      do k=1,nbn
         if (bn(k)==ai(i,j)) then
             ai1(i,jn)=jm
             jn=jn+1
         endif
         jm=jm+1
      enddo
   enddo
enddo
do i=1,arl
   jm=1
       do j=9,12
                          ai1(i,jm)=2*ai1(i,j)-1
                          ai1(i,jm+1)=2*ai1(i,j)
      jm=jm+2
       enddo
enddo
do i=1,arl
    ai1(i,13)=ai(i,1)
enddo
do i=1,nbn*2
      rr2(i)=bn1(i)
enddo
jm=0
do i=1,nr
 if (rr(i,1)/=0) then
   do j=1,arl
      if (ai(j,1)==rr(i,1)) then
          do k=2,16,2
             if (rr(i,k)/=0) then
                rr(i,k+1)=ai(j,rr(i,k))
             endif
          enddo
      endif
   enddo
   do k=2,16,2
         if (rr(i,k)/=0) then
            jm=jm+1
                              rr1(jm)=rr(i,k+1)
         endif
   enddo
```

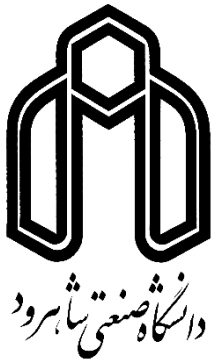
```
      endif
    enddo
    do j=1,jm-1
          do k=j+1,jm
                if (rr1(j)==rr1(k)) then
                                                rr1(k)=0
                endif
          enddo
          enddo
    nr3=0
    do i=1,jm
       if (rr1(i)/=0) then
          do j=1,nbn*2
             if (rr1(i)==rr2(j)) then
                                    rr2(j)=0
                nr3=nr3+1
             endif
          enddo
       endif
    enddo
endsubroutine vava2

subroutine mass (nn,ll)
real*8::nn(2,8),ll(2)
    nn=0
    nn(1,1)=0.25*(1-ll(1))*(1-ll(2));nn(2,2)=0.25*(1-ll(1))*(1-ll(2))
    nn(1,3)=0.25*(1+ll(1))*(1-ll(2));nn(2,4)=0.25*(1+ll(1))*(1-ll(2))
    nn(1,5)=0.25*(1+ll(1))*(1+ll(2));nn(2,6)=0.25*(1+ll(1))*(1+ll(2))
    nn(1,7)=0.25*(1-ll(1))*(1+ll(2));nn(2,8)=0.25*(1-ll(1))*(1+ll(2))
    endsubroutine mass
```

Abstract

Recently most of the studies and researches in optimization of topology are influenced by static loads, though in the real world the forces have dynamic identities. If these structures are influenced by dynamic loads, unpredictable tensions and deformations will be occurred in them. In the real world, most of the loads are applied as best structure optimization. Therefore, it is necessary that analysis and design should be performed based on dynamic loads. It is necessary that structures also be optimized under dynamic loads in order that the real behavior of the constituent is modeled. Dynamic loading has a different effect on the structure response as the more the frequency of the applied load in relation to the structure frequency; the applied load will have a different effect on static state on the structure. Evolutionary structural optimization (ESO) is based on a simple idea that an optimal structure (with maximum stiffness but minimum weight) can be achieved by gradually removing ineffectively used materials from design domain. In general, the results from ESO are likely to be local optimums other than the global optimum desired. In this paper, the genetic algorithm (GA) is integrated with ESO to form a new algorithm called Genetic Evolutionary Structural Optimization (GESO), which takes the advantage of the excellent behavior of the GA in searching for global optimums. For the developed GESO method, each element in finite element analysis is an individual and has its own fitness value according to the magnitude of its sensitivity number. Then, all elements in an initial domain constitute a whole population in GA. After a number of generations, undeleted elements will converge to the optimal result that will be more likely to be a global optimum than that of ESO

This task is performed by adding genetic algorithm operators like Mutation, Crossover in ESO. In fact, by this task, a new evolutionary algorithm called GESO is created that its power in searching the total responses is partially higher than that of ESO.

Keywords: GESO, Structure Evolutionary Optimization, Topology, Dynamic Loads, Genetic Algorithms.



Shahrood University of Technology
Faculty of Civil Engineering

M.SC.Thesis in Structural Engineering

Structural Topology Optimization under Dynamic Loads by GESO Method

By: E. Roohparvar

Supervisor:

Dr. A. Keyhani

September 2017