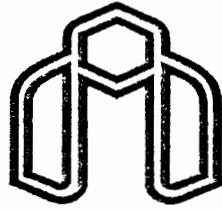


بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ



دانشگاه صنعتی شاهرود

دانشکده عمران و معماری

پایان نامه کارشناسی ارشد عمران - سازه

**بهینه یابی مقاطع، هندسه و توپولوژی خرپاها با استفاده از**

**الگوریتم ژنتیک و نرم افزار Matlab V7.0**

توسط:

احسان اله ضیغمی

استاد راهنما:

دکتر وحیدرضا کلاتجاری

استاد مشاور:

دکتر علی کیهانی

تابستان ۸۴

# چکیده

علم بهینه یابی در مهندسی سازه بدنبال کاهش هزینه های مربوط به مصالح مصرفی ضمن تأمین استحکام مورد نیاز سازه ها می باشد. در این راستا معمولاً بهینه یابی سازه های اسکلتی به چهار دسته عمده تقسیم می شود: ۱- بهینه یابی مقطع<sup>۱</sup> که در برگیرنده تعیین سطح مقطع بهینه اعضا می باشد. ۲- بهینه یابی شکل یا هندسه<sup>۲</sup> که در آن مختصات بهینه گرهی به ازای نحوه قرارگیری مفروض اعضا تعیین می شود. ۳- بهینه یابی توپولوژی<sup>۳</sup> مشتمل بر تعیین بهترین آرایش اعضا در فضای مفروض مختصات گرهی و ۴- بهینه یابی پیکر بندی<sup>۴</sup> که در برگیرنده سه مورد قبل بطور توأم می باشد.

روشهای بهینه یابی سازه ها نیز معمولاً به دو گروه اصلی دسته بندی می شود: ۱- روشهای مبتنی بر برنامه ریزی ریاضی<sup>۵</sup> که دیدگاه غیر احتمالاتی بر فضای مساله حاکم است و ۲- روشهای مبتنی بر کاوش تصادفی<sup>۶</sup> که از تئوری احتمالات در فرمول بندیهای خود بهره می برد. تحلیل سازه ها نیز به نوبه خود به دو گروه اصلی دسته بندی می شوند: ۱- روش سختی یا تغییر مکان<sup>۷</sup> و ۲- روش نرمی یا نیروها<sup>۸</sup>.

1- Sizing Optimization

2- Geometrical (Shape) Optimization

3- Topological Optimization

4- Layout Optimization

5- Mathematical Programming

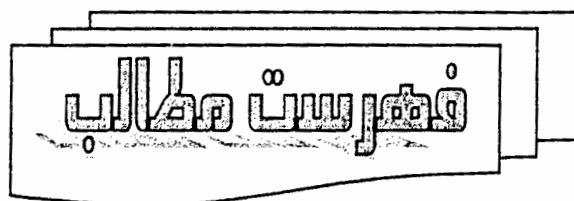
6- Random Search Method

7- Stiffness (Displacement) Method

8- Flexibility (Force) Method

در این رساله ما به بهینه یابی سازه های خرپایی با استفاده از روشهای مبتنی بر الگوریتم ژنتیک (GA) می پردازیم و ضمن ایجاد یک بسته نرم افزاری جهت بهینه سازی خرپاها که با استفاده از برنامه نویسی در محیط نرم افزار Matlab V7.0 حاصل می شود، به حل مثالهای گوناگون پرداخته و ضمن انتخاب گزینه های مختلف برای الگوریتم ژنتیک به مقایسه نتایج حاصل از اجراهای مختلف برنامه با مراجع می پردازیم. روش استفاده شده در تحلیل سازه روش سختی می باشد، که جزئیات آن در پیوست الف آورده شده است.

در فصل اول این رساله ضمن بیان مقدمه ای بر الگوریتم ژنتیک، کارهای انجام شده توسط محققان در این زمینه به طور اجمالی مورد بررسی قرار می گیرد. در فصل دوم مبانی بهینه یابی سازه ها بر اساس الگوریتم ژنتیک تشریح شده است. در این فصل با ذکر یک مثال جامع نحوه عملکرد الگوریتم ژنتیک تشریح شده است. در فصل سوم برخی از تکنیک های نوین الگوریتم ژنتیک از جمله مفاهیم الگوریتم ژنتیک توزیعی *DGA* مورد بررسی قرار گرفته است و روش هایی که در آن ها امکان استفاده از سیستم های کامپیوتری موازی وجود دارد، در این فصل بررسی می شوند و در خاتمه این فصل به حل یک مثال خرپای ۱۲۸ سیله ای با روش های مطرح شده می پردازیم و به مقایسه نتایج با مراجع می پردازیم. در فصل چهارم به معرفی قابلیت های نرم افزار Matlab V7.0 در بهینه سازی سازه ها پرداخته و با معرفی گزینه های ابزار بهینه سازی الگوریتم ژنتیک به حل یک مثال معروف با استفاده از Matlab V7.0 می پردازیم و گرافها و نمودارهای خروجی جهت بررسی نتایج مورد استفاده قرار می گیرد. در فصل پنجم، با استفاده از بسته نرم افزاری ایجاد شده در محیط Matlab V7.0 به بهینه سازی مقاطع، مقاطع و هندسه، مقاطع و توپولوژی و در نهایت به بهینه سازی همزمان مقاطع، هندسه و توپولوژی سازه های خرپایی پرداخته ایم و ضمن حل مثالهای گوناگون با پارامترهای مختلف GA همچنین بکارگیری توابع جریمه متفاوت به مقایسه این نتایج با مراجع پرداخته ایم و در نهایت در فصل ششم، ضمن بیان نتایج، پیشنهاداتی نیز برای انجام کارهای آتی عنوان شده است.



صفحه

عنوان

**فصل ۱: مقدمه و تاریخچه الگوریتم ژنتیک**

- ۱-۱ مقدمه ..... ۱
- ۲-۱ تاریخچه الگوریتم ژنتیک و مروری اجمالی بر کارهای انجام شده ..... ۳

**فصل ۲: مبانی بهینه یابی سازه ها با استفاده از الگوریتم ژنتیک**

- ۱-۲ مقدمه ..... ۶
- ۲-۲ تعاریف تابع پناستی ..... ۸
- ۱-۲-۲ تعریف پیشنهادی تابع پناستی ..... ۹
- ۳-۲ تعاریف تابع شایستگی ..... ۱۰
- ۴-۲ تعریف فرآیند تکثیر و انتخاب ..... ۱۱
- ۵-۲ اپراتورهای پیوند و جهش ..... ۱۲
- ۶-۲ تشریح عملکرد الگوریتم ژنتیک توسط یک مثال عددی ..... ۱۳

**فصل ۳: تکنیک های نوین در الگوریتم ژنتیک**

- ۱-۳ مقدمه ..... ۲۹
- ۲-۳ الگوریتم ژنتیک و الگوریتم ژنتیک توزیعی ..... ۳۳
- ۱-۲-۳ الگوریتم ژنتیک ..... ۳۳
- ۲-۲-۳ الگوریتم ژنتیک توزیعی ..... ۳۴
- ۳-۳ الگوریتم ژنتیک برای مسائل مقید ..... ۳۶

۳۶	۱-۳-۳ اصلاح تابع هدف برای مسائل مقید.....
۳۸	۲-۳-۳ الگوریتم ژنتیک توزیعی محیطی.....
۳۹	۴-۳ مثالهای عددی.....
۳۹	۱-۴-۳ طراحی یک سازه خرابایی.....
	۲-۴-۳ چرا الگوریتم ژنتیک ساده در پیدا کردن جواب مسائلی که دارای تعداد متغیرهای طراحی
۴۲	زیادی هستند به خوبی عمل نمی کند؟.....
۴۴	۳-۴-۳ قابلیت کاوش الگوریتم های ژنتیک توزیعی.....
۴۵	۴-۴-۳ کارایی الگوریتم های ژنتیک توزیعی محیطی.....
۴۷	۵-۴-۳ بهبود سرعت.....
۴۸	۵-۳ نتیجه گیری.....

**فصل ۴: توانمندیهای نرم افزار Matlab V7.0 در الگوریتم ژنتیک**

۵۰	۱-۴ مقدمه.....
۵۲	۲-۴ الگوریتم ژنتیک در Matlab V7.0.....
۵۳	۱-۲-۴ فراخوانی تابع الگوریتم ژنتیک در خط دستور.....
۵۵	۲-۲-۴ استفاده از ابزار الگوریتم ژنتیک.....
۵۷	۳-۴ مثال تابع Rastrign.....
۵۳	۱-۳-۴ تابع Rastrign.....
۵۸	۲-۳-۴ پیدا کردن مینیمم تابع.....
۶۱	۴-۴ گراف های ترسیم شده.....
۶۴	۵-۴ برخی از اصطلاحات الگوریتم ژنتیک.....
۶۵	۱-۵-۴ تابع شایستگی.....

۶۵	..... ۲-۵-۴ اعضا
۶۶	..... ۳-۵-۴ جمعیت و نسلها
۶۶	..... ۴-۵-۴ پراکندگی
۶۷	..... ۵-۵-۴ مقادیر شایستگی و بهترین مقدار شایستگی
۶۷	..... ۶-۵-۴ والدین و فرزندان
۶۸	..... ۶-۴ نمای کلی الگوریتم ژنتیک در Matlab V7.0
۶۸	..... ۱-۶-۴ جمعیت اولیه
۷۰	..... ۲-۶-۴ ایجاد نسل بعدی
۷۰	..... ۱-۲-۶-۴ فرزندان پیوند یافته
۷۰	..... ۲-۲-۶-۴ فرزندان جهش یافته
۷۲	..... ۷-۴ شکل نسلهای بعدی
۷۳	..... ۸-۴ حالت های توقف الگوریتم ژنتیک

**فصل ۵: بهینه یابی مقاطع، هندسه و توپولوژی خرپاها با استفاده از الگوریتم ژنتیک**

۷۶	..... ۱-۵ بهینه یابی مقاطع سازه های خرپایی با استفاده از الگوریتم ژنتیک
۷۷	..... ۱-۱-۵ فرمولبندی مسأله بهینه یابی مقاطع منفصل سازه های خرپایی بر اساس GA
۷۸	..... ۲-۱-۵ رمزگشایی کرموزوم ها در مسأله بهینه یابی مقاطع منفصل خرپاها بر اساس GA
۸۱	..... ۳-۱-۵ مثالهای عددی
۸۱	..... ۱-۳-۱-۵ بهینه یابی مقاطع خرپای مستوی ۱۰ میله ای
۸۴	..... ۲-۳-۱-۵ بهینه یابی مقاطع خرپای فضایی ۲۵ میله ای
۸۹	..... ۴-۱-۵ نتیجه گیری
۹۰	..... ۲-۵ بهینه یابی مقاطع و هندسه سازه های خرپایی با استفاده از الگوریتم ژنتیک

۹۱	۱-۲-۵ توابع پناستی و شایستگی
۹۱	۲-۲-۵ فرآیند پیوند و جهش
۹۱	۳-۲-۵ بهینه یابی مقاطع و هندسه پیوسته خراباها
۹۵	۴-۲-۵ بهینه یابی مقاطع منفصل و هندسه پیوسته خراباها
۹۶	۵-۲-۵ مثالهای عددی
۹۷	۱-۵-۲-۵ بهینه یابی مقاطع و هندسه خرابای مستوی ۱۸ میله ای
۱۰۱	۲-۵-۲-۵ بهینه یابی مقاطع و هندسه خرابای فضایی ۲۵ میله ای
۱۰۵	۶-۲-۵ نتیجه گیری
۱۰۷	۳-۵ بهینه یابی سطح مقطع و توپولوژی سازه های خرابایی با استفاده از الگوریتم ژنتیک
۱۰۷	۱-۳-۵ مقدمه
۱۰۹	۲-۳-۵ بهینه یابی سطح مقطع و توپولوژی با استفاده از روش سختی
۱۱۰	۳-۳-۵ الگوریتم ژنتیک برای بهینه یابی سطح مقطع و توپولوژی خراباها
۱۱۲	۴-۳-۵ آشنایی با تعاریف پایه نظریه گرافها
۱۱۵	۵-۳-۵ تولید جمعیت اولیه با توجه به گراف پایه
۱۱۵	۶-۳-۵ تعیین مقادیر متغیرهایی طراحی سطح مقطع در عملیات GA
۱۱۶	۷-۳-۵ تابع جریمه و تابع شایستگی مورد استفاده
۱۱۸	۸-۳-۵ فرآیند انتخاب، مقادیر نرخ پیوند و جهش
۱۱۹	۹-۳-۵ مثالهای عددی
۱۱۹	۱-۹-۳-۵ بهینه یابی سطح مقطع و توپولوژی خرابای مستوی ۶ گرهی
۱۲۵	۲-۹-۳-۵ بهینه یابی سطح مقطع و توپولوژی خرابای مستوی ۱۴ گرهی
۱۲۷	۳-۹-۳-۵ بهینه یابی سطح مقطع و توپولوژی شبکه $2 \times 2$ و $3 \times 3$ خرابای مستوی ۱۴



۱۲۹	۴-۹-۳-۵ بهینه یابی سطح مقطع و توپولوژی خرابای فضایی ۱۰ گرهی
۱۳۲	۱۰-۳-۵ نتیجه گیری
۱۳۳	۴-۵ بهینه یابی مقاطع و هندسه سازه های خرابایی با استفاده از الگوریتم ژنتیک
۱۳۳	۱-۴-۵ مقدمه
۱۳۴	۲-۴-۵ فرمولبندی مسأله بهینه یابی مقاطع، هندسه و توپولوژی خراباها
۱۳۷	۳-۴-۵ مثالهای عددی
۱۳۷	۱-۳-۴-۵ بهینه یابی مقاطع، هندسه و توپولوژی خرابای مستوی ۱۴ گرهی
۱۴۰	۴-۴-۵ نتیجه گیری

### فصل ۶: نتایج و پیشنهادات

۱۴۴	۱-۶ نتایج
۱۴۶	۲-۶ پیشنهادات
۱۴۸	فهرست مراجع
۱۵۳	ضمیمه الف (روش تحلیل سازه)
۱۶۰	ضمیمه ب (بسته نرم افزاری بهینه یابی خراباها با استفاده از نرم افزار مطلب)

# فهرست اشکال

<u>صفحه</u>	<u>عنوان</u>
۱۲	شکل ۱-۲ خریای ۳ میله ای
۱۴	شکل ۲-۲ رشته کروموزوم به طول ۸
۳۵	شکل ۱-۳ الگوریتم ژنتیک توزیعی
۳۷	شکل ۲-۳ الگوریتم ژنتیک توزیعی محیطی
۴۰	شکل ۳-۳ مدلی از سازه خریا
۴۲	شکل ۴-۳ روند تابع شایستگی (معادله ۳-۱)
۴۳	شکل ۵-۳ روند تابع شایستگی (معادله ۲-۳)
۴۵	شکل ۶-۳ مقایسه الگوریتم های ژنتیک ساده و توزیعی
۴۷	شکل ۷-۳ مقایسه الگوریتم های ژنتیک توزیعی و توزیعی محیطی
۴۸	شکل ۸-۳ بهبود سرعت
۵۵	شکل ۱-۴ ابزار الگوریتم ژنتیک

۵۸	شکل ۴-۲ ترسیمه تابع Rastrigin
۵۹	شکل ۴-۳ روند تغییرات شایستگی بر اساس نسلها
۶۴	شکل ۴-۴ روند تغییرات شایستگی بر اساس نسلها در حالت لگاریتمی
۶۶	شکل ۴-۵ پراکندگی اعضای جمعیت
۶۹	شکل ۴-۶ پراکندگی جمعیت اولیه
۷۱	شکل ۴-۷ سه نوع فرزند تولید شده در الگوریتم ژنتیک
۷۱	شکل ۴-۸ پراکندگی فرزندان تولید شده در الگوریتم ژنتیک
۷۲	شکل ۴-۹ پراکندگی فرزندان تولید شده در الگوریتم ژنتیک مربوط به نسلهای ۶۰، ۸۰، ۹۵، ۱۰۰
۷۵	شکل ۴-۱۰ موارد توقف الگوریتم ژنتیک
۸۳	شکل ۵-۱ خریای مستوی ۱۰ میله ای
۸۴	شکل ۵-۲ خریای فضایی ۲۵ میله ای
	شکل ۵-۳ جعبه ابزار ایجاد شده توسط برنامه نویسی در محیط Matlab V7.0
۸۹	برای کنترل و انتخاب ورودی ها و خروجی های الگوریتم ژنتیک
۹۲	شکل ۵-۴ هندسه اولیه خریای مستوی ۱۵ میله ای
۹۷	شکل ۵-۵ هندسه اولیه خریای ۱۸ میله ای
۹۹	شکل ۵-۶ هندسه بهینه خریای مستوی ۱۸ میله ای
۱۰۰	شکل ۵-۷ هیستوگرام مربوط به شایستگی اعضای جمعیت در طرح بهینه
۱۰۰	شکل ۵-۸ نمودار مربوط به نقض قیود (پارامتر Cg) در نسلهای مختلف
۱۰۱	شکل ۵-۹ هندسه اولیه خریای فضایی ۲۵ میله ای
۱۰۴	شکل ۵-۱۰ هندسه بهینه شده (خط پیوسته) بهمراه هندسه اولیه (خط چین) خریای فضایی ۲۵ میله ای
	شکل ۵-۱۰ نمای X-Z هندسه بهینه (خط پیوسته) و هندسه اولیه (خط چین) خریای ۲۵ میله ای ۱۰۴

## عنوان

## صفحه

- شکل ۵-۱۱ نمای Y-Z هندسه بهینه (خط پیوسته) و هندسه اولیه (خط چین) خرپای ۲۵ میله ای ۱۰۴
- شکل ۵-۱۲ نمایش شبکه متعامد، گراف کامل و گراف با SRT-2 ۱۱۴
- شکل ۵-۱۳ گراف پایه ستاره ای برای گراف خنثی ۱۵ گره ای ۱۱۴
- شکل ۵-۱۴ گراف پایه ستاره ای برای خرپای مستوی ۶ گره ای ۱۲۰
- شکل ۵-۱۵ مقادیر تابع هدف (وزن سازه) در نسلهای مختلف ۱۲۰
- شکل ۵-۱۶ مقادیر نقض قیود در نسلهای مختلف ۱۲۱
- شکل ۵-۱۷ توپولوژی بهینه برای گراف پایه ستاره ای ۱۰ میله ای ۱۲۱
- شکل ۵-۱۸ میانگین فواصل مابین اعضای جمعیت و روند همگرایی جوابها ۱۲۲
- شکل ۵-۱۹ میانگین فواصل مابین اعضای جمعیت و روند همگرایی جوابها ۱۲۲
- شکل ۵-۲۰ توپولوژی بهینه خرپای ۶ گره ای (اگر  $P_{3y}$  و  $P_{4y}$  بصورت دو وضعیت بار بر سازه اثر کند) ۱۲۴
- شکل ۵-۲۱ مقادیر نقض قیود در نسلهای مختلف ۱۲۴
- شکل ۵-۲۲ سازه مبنا برای خرپای ۱۴ گره ای (گراف پایه ستاره ای ۳۱ عضوی) ۱۲۵
- شکل ۵-۲۲ توپولوژی بهینه خرپای مستوی ۱۴ گره ای ۱۲۶
- شکل ۵-۲۳ شبکه های  $2 \times 2$  و  $3 \times 3$  خرپای مستوی ۱۲۷
- شکل ۵-۲۴ توپولوژی بهینه شبکه های  $2 \times 2$  و  $3 \times 3$  خرپای مستوی ۱۲۸
- شکل ۵-۲۵ توپولوژی بهینه اصلاح شده شبکه های  $2 \times 2$  و  $3 \times 3$  خرپای مستوی ۱۲۹
- شکل ۵-۲۶ خرپای فضایی ۱۰ گره ای ۱۲۹
- شکل ۵-۲۷ توپولوژی بهینه خرپای فضایی ۱۰ گره ای ۱۳۰
- شکل ۵-۲۸ سازه مبنا برای خرپای ۱۴ گره ای (گراف پایه ستاره ای ۳۱ عضوی) ۱۳۷
- شکل ۵-۲۹ طراحی بهینه خرپای مستوی ۱۴ گره ای توسط راجان (۱۹۹۵) ۱۴۲
- شکل ۵-۳۰ طراحی بهینه خرپای مستوی ۱۴ گره ای توسط روش رساله (با امکان همپوشانی اعضا) ۱۴۳
- شکل ۵-۳۱ طراحی بهینه خرپای مستوی ۱۴ گره ای توسط روش رساله (بدون امکان همپوشانی اعضا) ۱۴۳

# فهرست جداول

## صفحه

## عنوان

۱۶	جدول (۱-۲) جزئیات تعیین مقادیر تابع شایستگی - نسل ۱
۱۷	جدول (۲-۲) جزئیات عملکرد فرآیند انتخاب بر اساس نظریه چرخ گردان - نسل ۱
۱۹	جدول (۳-۲) جزئیات عملکرد اپراتورهای پیوند و جهش در روش چرخ گردان - نسل ۱
۲۰	جدول (۴-۲) جزئیات تعیین مقادیر تابع شایستگی در روش چرخ گردان - نسل ۲
۲۱	جدول (۵-۲) جزئیات عملکرد فرآیند انتخاب بر اساس نظریه چرخ گردان - نسل ۲
۲۱	جدول (۶-۲) جزئیات عملکرد اپراتورهای پیوند و جهش در روش چرخ گردان - نسل ۲
۲۲	جدول (۷-۲) جزئیات تعیین مقادیر تابع هدف اصلاح شده در روش چرخ گردان - نسل ۳
۴۱	جدول (۱-۳): پارامترهای سازه خریایی
۵۳	جدول (۱-۴) تفاوت‌های الگوریتم ژنتیک با الگوریتم‌های استاندارد

## عنوان

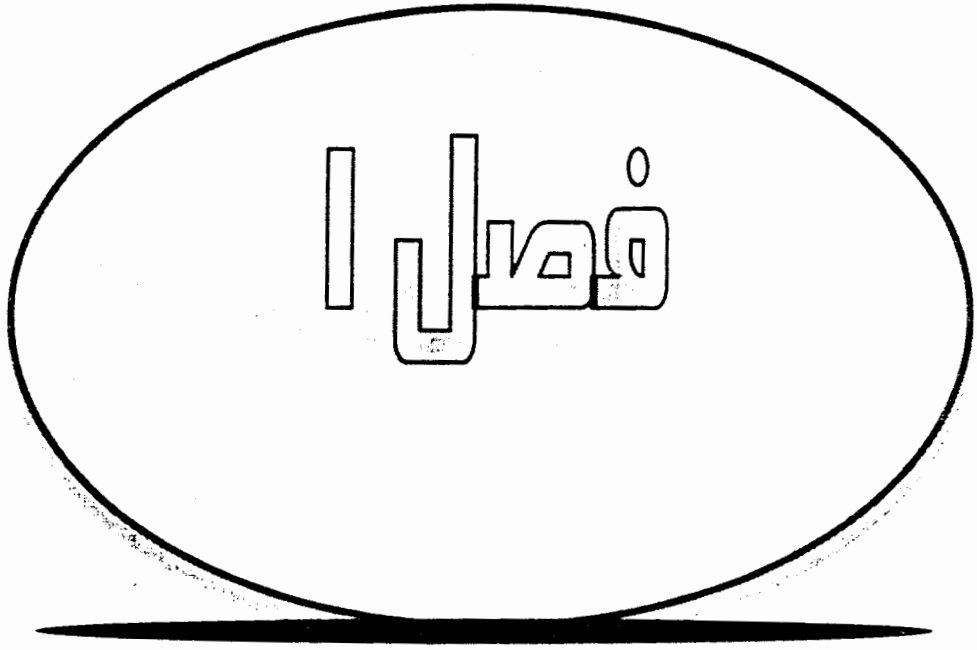
## صفحه

- جدول (۱-۵) جزئیات گروه بندی و شماره گذاری نمونه اعضای خریای مستوی ۱۰ میله ای ۸۲
- جدول (۲-۵) اطلاعات طراحی خریای دو بعدی ۱۰ میله ای ۸۳
- جدول (۳-۵) مقایسه نتایج طرح بهینه مقاطع خریای دو بعدی ۱۰ میله ای ۸۳
- جدول (۴-۵) اطلاعات مربوط به ثابتها و روابط بکار گرفته شده در این رساله مربوط به خریای ۱۰ میله ای ۸۴
- جدول (۵-۵) جزئیات گروه بندی و شماره گذاری اعضای خریای فضایی ۲۵ میله ای ۸۵
- جدول (۶-۵) مجموعه مقاطع منفصل  $S_1$  و  $S_2$  ۸۶
- جدول (۷-۵) اطلاعات طراحی خریای فضایی ۲۵ میله ای ۸۶
- جدول (۸-۵) مقایسه نتایج طرح بهینه مقاطع خریای فضایی ۲۵ میله ای ۸۷
- جدول (۹-۵) اطلاعات مربوط به ثابتها و شماره روابط مورد استفاده برای خریای ۲۵ میله ای ۸۸
- جدول (۱۰-۵) مفروضات طراحی خریای ۱۵ میله ای ۹۳
- جدول (۱۱-۵) نتایج طرح بهینه مقاطع منفصل و هندسه خریای ۱۵ میله ای ۹۴
- جدول (۱۲-۵) اطلاعات طراحی خریای ۱۸ میله ای ۹۸
- جدول (۱۳-۵) نتایج طرح بهینه مقاطع منفصل و هندسه پیوسته خریای ۱۸ میله ای بر اساس روشهای مطرح شده در این رساله و مقایسه با مرجع [37] ۹۹
- جدول (۱۴-۵) مختصات گرهی و گروه بندی اعضای خریای فضایی ۲۵ میله ای ۱۰۱
- جدول (۱۵-۵) اطلاعات طراحی خریای فضایی ۲۵ میله ای ۱۰۲
- جدول (۱۶-۵) اطلاعات مربوط به بارگذاری و خصوصیات مکانیکی مصالح خریای ۲۵ میله ای ۱۰۲

## عنوان

## صفحه

جدول (۵-۱۷)	نتایج طرح بهینه مقاطع منفصل و هندسه پیوسته خرپای فضایی ۲۵ میله ای بر اساس روش مطرح شده و مقایسه با مرجع [35]	۱۰۳
جدول (۵-۱۸)	مجموعه مقاطع منفصل مورد استفاده در مثالها	۱۱۹
جدول (۵-۱۹)	نتایج سطح مقطع و توپولوژی بهینه خرپای مستوی ۶ گرهی منتخب از لیست مقاطع $S_1$ و $S_2$	۱۲۳
جدول (۵-۲۰)	نتایج سطح مقطع و توپولوژی بهینه برای خرپای مستوی شش گرهی منتخب از لیست مقاطع $S_3$ (اگر $P_{2v}$ و $P_{4v}$ بصورت دو وضعیت بار فرض شوند)	۱۲۳
جدول (۵-۲۱)	گروه بندی اعضای خرپای ۱۴ گرهی	۱۲۵
جدول (۵-۲۲)	نتایج سطح مقطع و توپولوژی بهینه خرپای مستوی ۱۴ گرهی	۱۲۶
جدول (۵-۲۳)	نتایج سطح مقطع و توپولوژی بهینه شبکه های $2 \times 2$ و $3 \times 3$ خرپای مستوی	۱۲۸
جدول (۵-۲۴)	گروه بندی اعضا و تنش های مجاز کششی و فشاری اعضای هر گروه	۱۳۰
جدول (۵-۲۵)	بارگذاری موثر بر خرپای فضایی ۱۰ گرهی	۱۳۱
جدول (۵-۲۶)	نتایج مقاطع توپولوژی بهینه خرپای فضایی ۱۰ گرهی	۱۳۲
جدول (۵-۲۵)	اطلاعات طراحی خرپای ۱۴ گرهی	۱۳۹
جدول (۵-۲۶)	نتایج سطح مقطع، هندسه و توپولوژی بهینه خرپای مستوی ۱۴ گرهی	۱۴۳



مقدمه و تاريخچه الكورينم (تتبي)



## ۱-۱ مقدمه

از دیر باز توجه انسان بر کاهش هزینه ها و افزایش پس انداز بوده است. این خواسته در مهندسی سازه بصورت کاهش مصرف مصالح و یا کاهش هزینه های اجرایی نمود پیدا می کند. طرح بهینه سیستم های سازه ای از جمله خرپاها به طرح بهینه مقاطع<sup>۱</sup>، طرح بهینه شکل<sup>۲</sup>، طرح بهینه توپولوژی<sup>۳</sup> و یا ترکیبی از آنها دسته بندی می شود.

جهت بهینه سازی سازه ها، دو نوع فرمولبندی رایج وجود دارد که دسته اول بهینه یابی بر اساس برنامه ریزی ریاضی است که پاره ای از کارهای انجام شده را می توان در مراجع ملاحظه نمود. دسته دوم بهینه یابی مبتنی بر روشهای کاوش هوشمند تصادفی است که الگوریتم ژنتیک (GA) را می توان به عنوان معروف ترین آنها شناخت. تمایز عمده دو روش فوق الذکر در این است که روشهای برنامه ریزی ریاضی احتیاج به برآورد گرادیان توابع هدف و قیود حاکم بر مساله دارند در حالیکه، روشهای کاوش تصادفی بدون نیاز به برآورد گرادیان توابع، بطور هدفمندی بهینه عمومی را در فضای طراحی کاوش می کنند. در صورتیکه فضای طراحی غیر محدب باشد و یا تابع هدف، تابعی غیر پیوسته از متغیرهای طراحی باشد، بهینه یابی بر اساس GA کارآمدی خود را به اثبات می رساند. در عین حال عدم شناخت درست از پارامترها و روابط حاکم در GA از قبیل تعداد اعضای جمعیت، تعداد نسل ها،

<sup>۱</sup> Sizing Optimal Design

<sup>۲</sup> Shape Optimal Design

<sup>۳</sup> Topology Optimal Design

نرخ پیوند، نرخ جهش، نوع تابع جریمه و مقادیر ثابتهای دخیل در آن، نوع تابع شایستگی و چگونگی فرآیند انتخاب از جمله عواملی است که سبب می‌شود GA بهینه محلی را بدست آورد.

در این رساله، برای رفع این نقایص، یک بسته نرم افزاری جهت بهینه سازی سازه های خرپایی تهیه شده است که در آن امکان تغییر گزینه های مربوط به الگوریتم ژنتیک و یا اعمال قیود مختلف تنها با انتخاب گزینه مربوطه در ابزار الگوریتم ژنتیک امکان پذیر می‌باشد. علاوه بر این، در این بسته نرم افزاری امکان ترسیم گراف های مختلف توسط برنامه در حین اجراهای مختلف وجود دارد که این امکان را به کاربر می‌دهد که با انتخاب گزینه های مختلف مربوط به عملگرهای GA بهترین آنها را برای هر مثال انتخاب کرده و حتی با استفاده از این گراف ها همگرایی و یا عدم همگرایی الگوریتم به سمت جواب های بهینه و یا حتی افتادن در دام بهینه های محلی را به خوبی تشخیص دهد.

یکی دیگر از مشکلات الگوریتم ژنتیک، حجم بالای عملیات محاسباتی توسط آن می‌باشد که این مشکل با بهبود سرعت و توانایی کامپیوترها در سالهای اخیر به‌مراه استفاده از برخی از روشهای الگوریتم ژنتیک توسعه یافته ( $DGA$ ) که در این رساله به آنها پرداخته شده است و در آنها امکان استفاده از سیستم های کامپیوتری موازی وجود دارد، تا حدود زیادی مرتفع شده است.

یکی از موارد مهم در بهینه سازی سازه ها، امکان وجود ناپایداریهای موضعی یا کلی در طرحهای بهینه می‌باشد که ممکن است علیرغم ارضاء قید تغییر مکان در سازه مخفی باشد که در این رساله با استفاده از محاسبه دترمینان ماتریس سختی در جاییکه این دترمینان صفر می‌شود توپولوژی های ناپایدار تشخیص داده شده و به شدت جریمه می‌شوند. اینکار تا حد زیادی حجم عملیات عددی را کاهش و پایداری توپولوژی بهینه را تضمین می‌کند و به این ترتیب در نسلهای بعدی توپولوژی های ناپایدار امکان حضور کمتری پیدا می‌کنند

در این رساله در فصل پنجم ما به بهینه سازی مقاطع، مقاطع و هندسه، مقاطع و توپولوژی و در نهایت به بهینه سازی همزمان مقاطع، هندسه و توپولوژی سازه های خرابایی پرداخته ایم. شیوه انجام این کار، استفاده از بسته نرم افزاری ایجاد شده در محیط Matlab V7.0 با استفاده از مفاهیم الگوریتم ژنتیک و حل مثالهای گوناگون مطرح شده در مراجع با پارامترهای مختلف مربوط به GA همچنین بکارگیری توابع جریمه متفاوت می باشد و در پایان با بررسی نتایج بدست آمده از اجراهای مختلف برنامه به مقایسه این نتایج با مراجع پرداخته ایم.

## ۱-۲ تاریخچه الگوریتم ژنتیک و مروری اجمالی بر کارهای انجام شده

از پیشگامان معرفی مبانی الگوریتم ژنتیک (GA) میتوان از هلند نام برد. سپس گلدبرگ در این زمینه کتاب معروف خود را تالیف کرد. از نخستین مقالات در زمینه بهینه یابی سیستم های سازه ای با استفاده از GA می توان به کارهای گندبرگ و سمتانی، جنکینز، راجیو و کریشنا مورتی، هاجلا و لین و گریسون و پک اشاره کرد.

بسیاری از محققین مقالات متنوعی درباره بسط دامنه کاربرد GA، بهبود نتایج و افزایش سرعت محاسبات ارائه نمودند که علاوه بر افزایش کارایی و سرعت عملیات بهینه سازی ما را قادر به حل مسائلی می کند که با روش های کلاسیک الگوریتم ژنتیک قابل انجام نمی باشد.

به عنوان مثال لی<sup>۱</sup> و همکارانش به بهینه سازی مکان و هندسه مواد پیزموتریک برای توپولوژی بازوی کششی با استفاده از SLP و GA پرداختند و دانشمندان دیگر<sup>۲</sup> نیز به حل مسائل بهینه سازی خرپاها بوسیله الگوریتم ژنتیک پرداختند.

یکی از بزرگترین مشکلات در بهینه سازی با استفاده از الگوریتم ژنتیک در نظرگرفتن قیود می باشد. الگوریتم ژنتیک در واقع روشی است که جهت بهینه سازی مسائل نامقید به کار می رود.

<sup>۱</sup> Li et al.

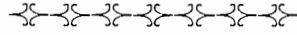
<sup>۲</sup> K.Deb and S.Gulati

بنابراین برای استفاده از الگوریتم ژنتیک توزیعی در مسائل مقید نیاز به استفاده از روش هایی جهت در نظر گرفتن قیود حاکم بر مسئله داریم. مطالعات بسیار زیادی برای در نظر گرفتن قیود در الگوریتم ژنتیک انجام گرفته است و در مقالات محققان روشهای مختلفی را برای این کار پیشنهاد کرده اند. یکی از راههای ساده برای در نظر گرفتن قیود استفاده از عبارت جریمه می باشد که بر اساس درجه نقض قیود به تابع هدف افزوده می شود. این روش به دو بخش عمده تقسیم می شود، که مربوط به قیود استاتیکی و دینامیکی است. روش تابع جریمه مرگ<sup>۱</sup> یک روش ساده می باشد که در آن هنگامی که یک نقطه جستجو قیود را نقض می کند، این نقطه حذف می شود و یک نقطه جستجو جدید بدست می آید که در آن نقطه قیود اقلانگ گردند. میخلاویچ و جانیکو<sup>۲</sup> به ارائه روش (GENOCOP) پرداختند که در آن با استفاده از یک اپراتور تصویرگر<sup>۳</sup> نقاط قابل قبول را با یک نگاشت<sup>۴</sup> بر روی نواحی قابل قبول منتقل می کند و این روش تنها در مورد مسائلی که در آنها قیود به صورت خطی می باشند قابل استفاده است.

در روش ارائه شده توسط هاجلا و یو<sup>۵</sup> جواب های بدست آمده توسط عملگر الگوریتم ژنتیک ضمانت قرارگیری در ناحیه قابل قبول را دارا هستند. در مقالات روش های متعددی از الگوریتم ژنتیک مطرح شده است که برای حل مسائل مقید از تکنیک های هدف چندگانه استفاده می کند. سوری و همکاران<sup>۶</sup> به ارائه روش COMOGA پرداختند که در آن تمام اعضای جمعیت بر اساس میزان نقض قیود رتبه بندی می شوند. همچنین روش های متعددی نیز برای در نظر گرفتن قیود مطرح شده است. از جمله، رایت و همکاران<sup>۷</sup> به ارائه روشی پرداختند که در آن نقاطی که قیود در آنها نقض می شود با در نظر گرفتن برخی احتمالات امکان حضور در جمعیت منتخب را دارا می باشند. کوزل و

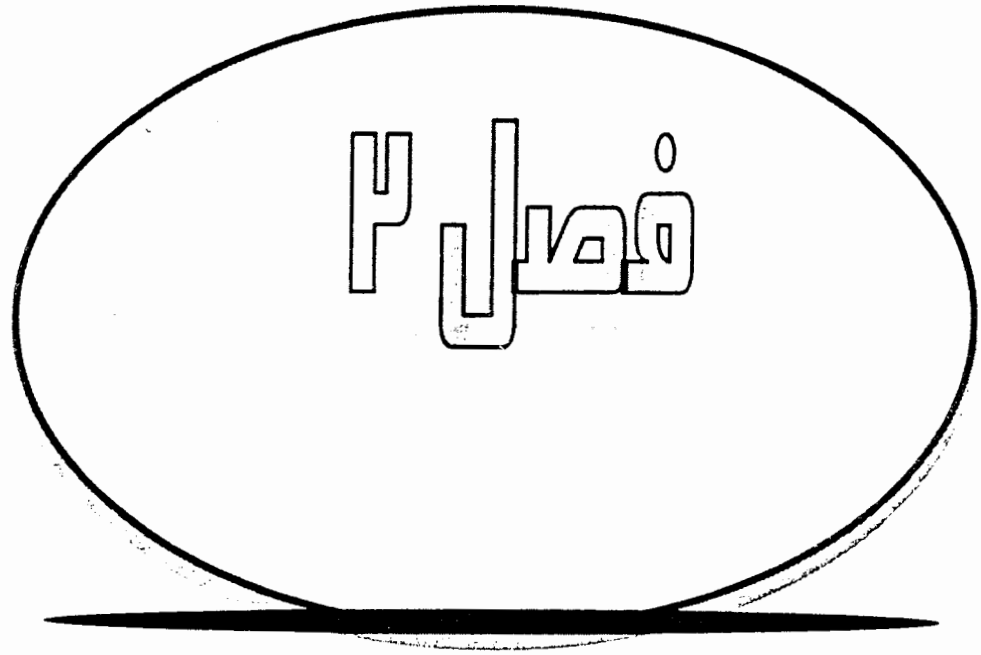
<sup>۱</sup> Death penalty method  
<sup>۲</sup> Michalewicz and Janikow  
<sup>۳</sup> Projection operators  
<sup>۴</sup> mapping  
<sup>۵</sup> Hajela and Yoo.  
<sup>۶</sup> Surry, et al.  
<sup>۷</sup> Wright et al.

میخاویچ<sup>۱</sup> به ارائه یک روش نگاشت خاص<sup>۲</sup> پرداختند. تاک و سان<sup>۳</sup> نیز با استفاده از روش لاگرانژین قیود را در نظر گرفتند.



---

<sup>۱</sup> Koziel and Michalewicz  
<sup>۲</sup> homomorphous mapping approach  
<sup>۳</sup> Tahk and Sun



مبانی بهینه‌یابی سازه‌ها با استفاده از الگوریتم ژنتیک

## ۲-۱ مقدمه

الگوریتم ژنتیک (GA) که به شبیه‌سازی تکامل مخلوقات و وراثت آنها می‌پردازد یکی از روش‌های بسیار قوی مورد استفاده در بهینه‌سازی می‌باشد که امکان استفاده از آن نه تنها در مسائل پیوسته<sup>۱</sup> وجود دارد بلکه در مسائل ناپیوسته<sup>۲</sup> نیز قابل استفاده است. این روش هوشمند بطور موفقیت آمیزی طرح بهینه عمومی<sup>۳</sup> را بدون در نظر گرفتن فرضیات محدود کننده‌ای از قبیل پیوسته بودن فضای کاوش و یا وجود مشتقات پیدا می‌کند. در این روش مساله بهینه‌یابی مقید با تعریف تابع پناالتی خاص و ترکیب آن با تابع هدف و ایجاد تابع هدف اصلاح شده<sup>۴</sup>، به یک مساله بهینه‌یابی نامقید تبدیل می‌شود.

بسیاری از محققان الگوریتم ژنتیک را برای حل مسائل واقعی به کار گرفته‌اند و به جواب‌های بسیار خوبی نیز دست یافته‌اند که از جمله این موارد، استفاده از الگوریتم ژنتیک در بهینه‌سازی سازه‌های خرپایی می‌باشد.

الگوریتم ژنتیک الهام گرفته شده از قوانین ژنتیک طبیعی<sup>۵</sup> و انتخاب طبیعی<sup>۶</sup> است که در آن بهترینها<sup>۷</sup> حق بقا دارند. GA عملیات خود را بر روی متغیرهای طراحی رمز شده<sup>۸</sup> که رشته‌ای از کاراکترها با طول محدود هستند، انجام می‌دهد.

در بهینه‌یابی بر اساس GA چهار مرحله اساسی انجام می‌پذیرد:

(۱) تولید تصادفی مجموعه‌ای از طرح‌ها که یک جمعیت اولیه<sup>۹</sup> را تشکیل می‌دهد. هر طرح در این جمعیت یک رشته منحصر بفرد<sup>۱۰</sup> نامیده شده و تعداد این رشته‌های منحصر بفرد، pop-size را تشکیل می‌دهد. هر رشته منحصر بفرد توسط زنجیره‌ای از کاراکترها تعریف می‌شود. به هر کاراکتر معمولاً یک

<sup>۱</sup> - Continious Problems

<sup>۲</sup> - Discrete Problems

<sup>۳</sup> - Global Optimum

<sup>۴</sup> - Modified Objective Function

<sup>۵</sup> - Natural Genetic

<sup>۶</sup> - Natural Selection

<sup>۷</sup> - Fittest

<sup>۸</sup> - Coded Design Variables

<sup>۹</sup> - Initial Population

<sup>۱۰</sup> - Individual

عدد تصادفی دودویی<sup>۱</sup> اختصاص یافته و تعداد کاراکترها یا تعداد بیت ها طول آنرا مشخص می کند. هر رشته<sup>۲</sup> مشتمل بر زیر رشته هایی<sup>۳</sup> است که هر کدام بیانگر یک متغیر طراحی است. تعداد زیر رشته ها مساوی با تعداد متغیرهای طراحی موجود در مساله بهینه یابی است. طول زیر رشته ها به نحوی تعیین می شود تا بتوان کلیه اطلاعات متغیر طراحی را مابین کرانه های تحتانی و فوقانی آن در مرحله رمز گشایی<sup>۴</sup> بدست آورد. در GA، رشته منحصر بفرد و کاراکتر به ترتیب هم ارز با کروموزوم<sup>۵</sup> و ژن<sup>۶</sup> در الفبای ژنتیک طبیعی است.

(۲) پس از تولید تصادفی جمعیت اولیه، با رمز گشایی که نگاشت از دستگاه دودویی به دستگاه دهگانی است، مقدار صحیح یا اعشاری متغیرهای طراحی در هر رشته ارزیابی و به ازای آنها مقادیر تابع هدف، تنش اعضا، تغییر مکانهای گرهی و میزان نقض قیود در ارتباط با پاسخ سازه<sup>۷</sup> به بارهای خارجی تعیین می گردد.

(۳) با تعریف تابع پناستی که وابسته به میزان نقض قیود است و ترکیب آن با تابع هدف، تابع هدف اصلاح شده تعیین و مساله بهینه یابی مقید به بهینه یابی نامقید تبدیل می گردد. در ادامه با تعریف تابع شایستگی<sup>۸</sup>، به مقادیر تابع هدف اصلاح شده وابسته به هر کروموزوم یک مقدار شایستگی اختصاص می یابد. بالطبع هر طرح اولیه ای که از نقض کمتری برخوردار باشد، شایستگی بیشتری به آن اختصاص می یابد. سپس با استفاده از فرآیند تکثیر<sup>۹</sup> که الهام گرفته شده از قوانین تکامل طبیعی<sup>۱۰</sup> است، سعی می شود بهترین ها<sup>۱۱</sup> انتخاب شوند. در این رساله فرآیند انتخاب<sup>۱۲</sup> با استفاده از روش چرخ گردان<sup>۱۳</sup> انجام شده است. یک کروموزوم خیلی خوب شانس آنرا میباید تا چندین بار در جمعیت منتخب تکرار شود. (۴) در این مرحله با استفاده از اپراتورهای GA که مرسوم ترین آنها پیوند<sup>۱۴</sup> و جهش<sup>۱۵</sup> است، جمعیت جدید (فرزندان<sup>۱۶</sup>) که کیفیت کروموزوم های آن نسبت به جمعیت منتخب بهبود یافته است، ایجاد می گردد. اپراتور پیوند، عمل جفت یابی<sup>۱۷</sup> بین کروموزوم های دلخواه را جهت ایجاد فرزندان بعهده دارد.

<sup>۱</sup>- Binary

<sup>۲</sup>- String

<sup>۳</sup>- Sub string

<sup>۴</sup>- Decoding

<sup>۵</sup>- Chromosome

<sup>۶</sup>- Gene

<sup>۷</sup>- Structural Response

<sup>۸</sup>- Fitness Function

<sup>۹</sup>- Reproduction

<sup>۱۰</sup>- Natural Evolution

<sup>۱۱</sup>- High Fitness

<sup>۱۲</sup>- Selection

<sup>۱۳</sup>- Roulette Wheel

<sup>۱۴</sup>- Crossover

<sup>۱۵</sup>- Mutation

<sup>۱۶</sup>- Offspring

<sup>۱۷</sup>- Mating



تعداد این عملیات به نرخ پیوند<sup>۱</sup> بستگی پیدا می‌کند. اپراتور جهش بسته به نرخ جهش<sup>۲</sup>، بطور تصادفی سبب تغییر مقدار در برخی از کاراکترها می‌شود. اگر به کاراکترهای جمعیت اعداد دودویی اختصاص داده شده باشد، اپراتور جهش در برخی از کاراکترها مقدار صفر را به یک و یا بالعکس تبدیل می‌کند. هر دو اپراتور وظیفه ایجاد پراکندگی<sup>۳</sup> در فضای جستجو را به منظور کسب بهینه واقعی<sup>۴</sup> بعهده دارند. پس از آنکه جمعیت فرزندان تولید گردید، این جمعیت برای نسل بعد<sup>۵</sup> مورد استفاده قرار می‌گیرد. این مراحل به تعداد نسل‌های پیش‌بینی شده برای کسب بهینه عمومی تکرار می‌شود.

## ۲-۲ تعاریف تابع پنالتی

تعاریف متنوعی برای تابع پنالتی در مقالات معرفی شده است. راجیو و کریشنامورتی در مقاله خود تابع پنالتی را به صورت زیر تعریف نموده‌اند:

$$f_{penalty} = f(\mathbf{X}) \cdot K \cdot C \quad ; \quad C = \sum_{q=1}^Q \max[0, g_q(\mathbf{X})] \quad (1-2)$$

در رابطه فوق  $f_{penalty}$  تابع جریمه،  $f(\mathbf{X})$  تابع هدف که معمولاً وزن یا حجم سازه است،  $g_q(\mathbf{X})$  میزان نقض سازه در ارتباط با هر یک از قيود،  $\mathbf{X}$  بردار متغیرهای طراحی و  $Q$  تعداد کل قيود حاکم بر مساله می‌باشد. ثابت  $K$  نیز برابر ۱۰ پیشنهاد شده است. همچنین راجان در مقاله خود تابع پنالتی را بشرح ذیل پیشنهاد کرده است:

$$f_{penalty} = r \cdot C \quad ; \quad r = K' \cdot F_{min} \quad (2-2)$$

در رابطه فوق  $C$  دارای تعریفی مشابه رابطه (۲-۱)،  $K'$  عددی ثابت وابسته به انتخاب طراح و  $F_{min}$  حداقل مقدار تابع شایستگی برای کثیف طرحهای قابل قبول<sup>۶</sup> نسل قبلی<sup>۷</sup> می‌باشد. برای نسل اول یک مقدار تخمینی برای  $r$  اتخاذ می‌شود. اپراتور و همکاران در مقاله خود، با استناد به رابطه پیشنهادی

<sup>۱</sup>- Crossover Rate

<sup>۲</sup>- Mutation Rate

<sup>۳</sup>- Diversity

<sup>۴</sup>- True Optimum

<sup>۵</sup>- New Generation

<sup>۶</sup>- Feasible Design

<sup>۷</sup>- Previous Generation

توسط جریزنر و هاک یک تابع پینالتی پویا<sup>۱</sup> بر حسب شماره نسل،  $g_c$ ، بصورت رابطه (۲-۳) معرفی کرده اند :

$$f_{penalty} = (r_0 \cdot g_c)^2 \sum_{q=1}^Q [g_q(\mathbf{X}) \cdot K] \quad (3-2)$$

در رابطه فوق  $K=10$  و  $r_0 = 0.5$  پیشنهاد شده است. مساله مهمی که در روابط مطرح شده برای تابع پینالتی به چشم می خورد. وابستگی همه آنها به ثابتهایی است که توسط طراح بایستی فرض شود و تاثیر بسزایی بر روی کیفیت پاسخ بهینه می گذارد. این مساله معمولاً سبب می شود تا طراح با فرض مقادیر متفاوتی برای ثابتها، تاثیر آنها را بر روی جواب بهینه بررسی کند و حجم عملیات GA را از آنچه که هست بیشتر کند. در این رساله با تعریف رابطه پیشنهادی برای تابع پینالتی سعی شده است تا مقدار زیادی مشکلات مربوط به توابع جریمه کلاسیک مرتفع گردد.

## ۲-۳ تعاریف تابع شایستگی

برای تابع شایستگی تعاریف متنوعی ارائه شده است. راجیو و کریشناامورتی در مقاله خود، رابطه

زیر را پیشنهاد نموده اند :

$$F = [\phi_{\max}(\mathbf{X}) + \phi_{\min}(\mathbf{X})] - \phi(\mathbf{X})$$

$$\phi(\mathbf{X}) = f(\mathbf{X}) + f_{penalty} \quad (4-2)$$

در رابطه فوق  $F$  و  $\phi(\mathbf{X})$  به ترتیب مقادیر شایستگی و تابع هدف اصلاح شده هر که موزوم و  $\phi_{\max}(\mathbf{X})$  و  $\phi_{\min}(\mathbf{X})$  به ترتیب حداکثر و حداقل مقدار تابع هدف اصلاح شده در جمعیت جاری می باشد. طبق رابطه بالا کروموزومی با کمترین مقدار تابع هدف اصلاح شده، بیشترین مقدار شایستگی به آن تعلق می گیرد.

<sup>۱</sup> Dynamic Penalty Function

ساکاموتا و<sup>۱</sup> دا در مقاله خود، تابع شایستگی مقیاس شده  $F$  را بصورت زیر تعریف کرده اند:

$$F = -a \Phi(\mathbf{X}) + b \quad ; \quad \Phi(\mathbf{X}) = f(\mathbf{X}) + r \sum_{q=1}^Q \max[0, g_q(\mathbf{X})]$$

$$a = \Phi_{avg}(\alpha - 1) / (\Phi_{avg} - \Phi_{min}) \quad ; \quad b = \Phi_{avg}(\alpha \Phi_{avg} - \Phi_{min}) / (\Phi_{avg} - \Phi_{min}) \quad (5-2)$$

در روابط فوق  $\Phi_{avg}$  و  $\Phi_{min}$  به ترتیب مقدار متوسط و حداقل  $\Phi(\mathbf{X})$  در کل جمعیت، ثابت  $\alpha = 1.5$  و ثابت  $r$  متناسب با انتخاب طراح می باشد. تعریف تابع شایستگی مقیاس شده سبب همگرایی بهتر<sup>۲</sup> به واسطه نزدیکتر کردن<sup>۳</sup> محدوده پراکنش تابعهای شایستگی جمعیت حول  $\Phi_{avg}$  می شود. کروموزوم هایی که شایستگی آنها کمتر  $\Phi_{avg}$  باشد به سمت بالا مقیاس شده<sup>۴</sup> و کروموزوم هایی که شایستگی آنها بیشتر از  $\Phi_{avg}$  باشد به سمت پایین مقیاس<sup>۵</sup> می شوند. به این ترتیب حاکمیت کروموزوم های غیر متعارف از بین رفته و همگرایی و کاوش بهتر در فضای طراحی انجام می پذیرد. اریاتور و همکاران نیز در مقاله خود، الگوریتمی برای تعیین مقدار تابع شایستگی مقیاس شده هر کروموزوم ارائه نموده اند.

## ۲-۴ تعریف فرآیند تکثیر و انتخاب

در فرآیند تکثیر کروموزوم های خوب از کروموزوم های بد جدا می شوند و کروموزوم هایی با شایستگی بالا شانس تکرار بیشتری را در جمعیت منتخب پیدا می کنند در حالیکه کروموزوم هایی که شایستگی کمتری دارند از شانس کمتری برای حضور در نسل های بعدی برخوردار هستند و در نسلهای بعد تمایل به حذف شدن دارند. این کار توسط فرآیند انتخاب صورت می پذیرد. بدین منظور روشهای متنوعی مطرح شده است که روش چرخ گردان یکی از معروف ترین آنهاست<sup>۶</sup>. در این روش به هر عضو جمعیت با توجه به مقدار شایستگی آن بخشی از چرخ گردان رولت اختصاص می یابد. چرخ رولت به تعداد جمعیت اولیه چرخیده می شود و جمعیت انتخاب شده برای نسل بعد را ایجاد می کند. در این روش

<sup>۱</sup> Scaled Fitness Function

<sup>۲</sup> Smooth Convergence

<sup>۳</sup> Narrowing

<sup>۴</sup> Scaled up

<sup>۵</sup> Scaled down

<sup>۶</sup> Goldberg, 2002

ممکن است برخی از کروموزوم ها که شایستگی بیشتری دارند به تعداد دفعات بیشتر از یک بار تکرار شود و برخی از کروموزوم ها که شایستگی کمتری دارند اصلاً در نسلهای بعدی حضور پیدا نکنند که این امر در کل باعث بالا رفتن شایستگی جمعیت می شود.

برای انجام این عمل به طریقه زیر عمل کرد:

۱. یک عدد تصادفی  $rand$  بین صفر و شایستگی نهایی  $F$  تولید می کنیم.
  ۲. برای کروموزوم فعلی  $c = 1$  را در نظر می گیریم.
  ۳. اگر  $rand \leq \sum_{i=1}^c f(x_i)$  بود، کروموزوم جاری تکثیر می شود و عملیات متوقف می شود. در غیر اینصورت به مقدار  $c$  یک عدد اضافه می شود و گام ۳ دوباره تکرار می شود.
- در روش چرخ گردان با جستجو به این روش در واقع شبیه به این است که ما چرخ رولت را به تعداد اعضای جمعیت چرخانده ایم و کروموزوم های نسل بعدی با توجه به قطاعی که از چرخ رولت به خود اختصاص داده است برای نسل های بعدی انتخاب شده اند.
- در این روش، کروموزوم های شایسته با قطعیت بیشتر شانس حضور در جمعیت منتخب را پیدا کرده و بجای کروموزوم های ضعیف، بهترین کروموزوم ها جایگزین می شود. این کار باعث می شود که روند بهبود کروموزوم ها در نسل های متوالی از بی نظمی کمتری برخوردار باشد.

## ۲-۵ اپراتورهای پیوند و جهش

پس از تکمیل فرآیند انتخاب، نوبت به اعمال اپراتور پیوند بر روی جمعیت منتخب<sup>۱</sup> جهت تولید جمعیت فرزندان<sup>۲</sup> می رسد. در انجام فرآیند پیوند، با انتخاب مقدار ثابت نرخ پیوند، به ترتیب برای هر کروموزوم یک عدد تصادفی تولید می شود. اگر عدد تصادفی تولید شده از نرخ پیوند کمتر باشد، این کروموزوم انتخاب می شود تا با کروموزوم بعدی که شرایط فوق را دارد، آمیزش نماید. سپس نوبت به اعمال اپراتور جهش می رسد. هدف از اینکار ایجاد پراکندگی بیشتر در محدوده کاوش فضای طراحی است

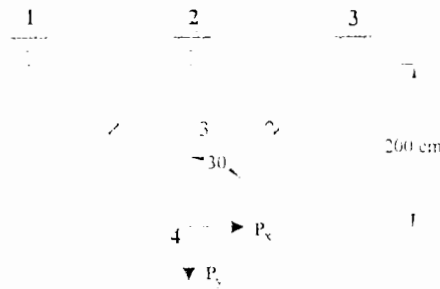
<sup>۱</sup> Mating Pool

<sup>۲</sup> Offspring

در انجام فرآیند جهش، با انتخاب مقدار ثابت نرخ جهش، برای کلیه بیت‌های کروموزوم های جمعیت عدد تصادفی تولید می‌شود. اگر عدد تصادفی تولید شده از نرخ جهش کوچکتر باشد، مقدار آن بیت عوض یعنی مقدار صفر به یک و یا بالعکس تبدیل می‌شود.

## ۲-۶ تشریح عملکرد الگوریتم ژنتیک توسط یک مثال عددی

در این بخش نحوه عملکرد الگوریتم ژنتیک برای بهینه یابی مقاطع سازه های خرابیایی توصیف می‌شود. بدین منظور همانند شکل (۱-۲) خرپای سه میله ای مورد بررسی قرار می‌گیرد.



شکل (۱-۲) خرپای ۳ میله ای

اطلاعات طراحی عبارتند از:  $E = 21008 \times 10^3 \text{ Mpa}$  ;  $\rho = 7850 \text{ kg/m}^3$  ;  $\sigma_{\max} = \pm 127.15 \text{ Mpa}$  , mm  
 هدف از این مثال یافتن سطح مقطع بهینه برای اعضای ۱، ۲ و ۳ تحت وضعیت بار مفروض است. فرمولبندی ریاضی آن به صورت مقابل بیان می‌شود: تابع هدف وزن،  $f(x)$ ، را تحت قیود  $g_i(x) \leq 0$  ,  $i = 1, 2, \dots, m$  کمینه کنید. m تعداد کل قیود می‌باشد. از آنجائیکه هدف کمینه کردن وزن می‌باشد،  $f(x)$  برای مثال خرپای ۳ میله ای به صورت رابطه (۷-۲) نمایش داده می‌شود.

$$f(x) = \sum_{j=1}^3 \rho A_j L_j \quad (۶-۲)$$

در رابطه فوق  $A_j$  و  $L_j$  به ترتیب سطح مقطع و طول عضو  $j$ ام و  $\rho$  وزن مخصوص مصالح می‌باشد.  $g_i(x)$  به صورت زیر بیان می‌شود:

$$\sigma_j \leq \sigma_a \quad ; \quad j = 1, 2, 3 \quad (۷-۲)$$

$$u_1, u_2 \leq u_a \quad (۸-۲)$$

در رابطه فوق  $\sigma$  تنش عضو زام،  $\sigma_a$  تنش مجاز،  $u_1, u_2$  تغییر مکان گره ۴ در جهات افقی و قائم و  $u_a$  تغییر مکان مجاز می باشد. در این مثال کلیه قیود نمیتوانند بطور صریح برحسب متغیرهای طراحی سطح مقطع توصیف شوند و ارزیابی آنها مستلزم تحلیل خرابا است. با توجه به آنکه الگوریتم ژنتیک برای مسائل بهینه یابی نامقید<sup>۱</sup> بسیار مناسب می باشد و مثال مطرح شده مقید می باشد، لذا ضروری است آنرا به یک مساله بهینه یابی نامقید تبدیل کنیم. این کار می تواند با استفاده از توابع پنالتی داخلی یا خارجی<sup>۲</sup> صورت پذیرد. در تحلیل این مثال قیود نرمال شده<sup>۳</sup> و میزان نقض آنها بر این مبنا ارزیابی شده است. این کار نتایج خوبی را بدنبال خواهد داشت زیرا به لحاظ کمی میزان نقض تنش بسیار متفاوت با نقض تغییر مکان است و نرمال کردن قیود سبب می شود ارزش کمی یکسانی به آنها اختصاص یابد. روابط قیود نرمال شده بصورت زیر قابل بیان است:

$$\frac{\sigma_j}{\sigma_a} - 1 \leq 0 \quad ; \quad \frac{u_j}{u_a} - 1 \leq 0 \quad (۹-۲)$$

در این مثال از تعریف تابع جریمه پیشنهادی توسط راجان مطابق روابط (۲-۲) استفاده شده است که در آن تعداد وضعیتهای بارگذاری  $nlc = 1$ ، تعداد کل قیود  $Q = 5$ ، مقدار اولیه ثابت تابع پنالتی  $k_p = 4.0$  در نظر گرفته شده است. حال با توجه به روابط (۲-۴) و (۲-۵)، تابع هدف اصلاح شده  $\phi(x)$  به نحوی که نقض قیود را در برگیرد بصورت زیر نوشته می شود:

$$\phi(x) = f(x)(1 + KC_g) \quad (۱۰-۲)$$

در رابطه فوق پارامتر  $K$  بایستی به نحو مناسبی در تناسب با میزان تاثیر گذاری آن برای حضور کروموزوم های نقض شده در نسل بعد اختیار شود. با توجه به آنکه تعداد اعضای جمعیت و تعداد نسلهای در این مثال نمونه کم است لذا  $K = 4.0$  فرض شده است. اکنون الگوریتم ژنتیک می تواند بهینه یابی نامقید را با

<sup>۱</sup> Unconstrained Optimization Problem  
<sup>۲</sup> Interior or Exterior Penalty Function

<sup>۳</sup> Normalized Constraints

توجه به تابع  $\phi(x)$  انجام دهد. چنانچه متغیرهای طراحی منفصل<sup>۱</sup> فرض شوند، مقادیرشان بایستی از لیست پروفیل‌های موجود<sup>۲</sup> انتخاب گردد. بدین منظور لیست S بصورت زیر در نظر گرفته می‌شود:

$$S = \{ 1.2, 1.4, 1.6, 1.8, 2.0, 2.2, 2.4, 2.6, 2.8, 3.0, 3.2, 3.4, 3.6, 3.8, 4.0, 4.4 \} \text{ (cm}^2\text{)}.$$

همانگونه که مشاهده می‌شود در مجموع ۱۶ مقطع منفصل در S موجود است. متغیرهای طراحی سطح مقطع سه عضو، یعنی  $A_1, A_2, A_3$  می‌باشند. در صورتیکه اعضا به نحوی گروه بندی<sup>۳</sup> شوند که سطح مقطع اعضای ۱ و ۲ یکسان باشند یعنی  $A_1 = A_2$ ، تعداد متغیرهای طراحی به دو کاهش یافته و به صورت  $A^1$  و  $A^2$  نمایش داده می‌شود. از آنجائیکه GA با متغیرهای طراحی رمز شده<sup>۴</sup> کار می‌کند لذا ضروری است متغیرهای طراحی سطح مقطع به صورت یک رشته رمز شوند. در میان شیوه های متنوع رمز کردن، در اینجا رمز کردن دودویی اتخاذ شده است زیرا کار با آن آسان است. هر متغیر طراحی میتواند یکی از ۱۶ مقطع لیست را اتخاذ کند. بنابراین یک رشته دودویی به طول ۴ برای این منظور کافیست. با توجه به آنکه دو متغیر طراحی پس از گروه بندی اعضا وجود دارد، همانند شکل (۲-۲) رشته ای با طول کل ۸ لازم می‌شود.

زیر رشته مربوط به گروه ۲      زیر رشته مربوط به گروه ۱

شکل (۲-۲) رشته کروموزوم به طول ۸

زیر رشته 0000 بیانگر اولین عنصر لیست S یعنی مقطع ۱/۲ و زیر رشته 1111 بیانگر مقطع شانزدهم یعنی ۴:۴ می‌باشد. هر مقدار مابین این دو توسط رمز گشایی بدست می‌آید. جهت شروع عملیات بهینه یابی توسط GA لازم است جمعیت اولیه ای از رشته ها (کروموزوم ها) تولید شود. فرض می‌شود تعداد کروموزوم ها یا اعضای جمعیت برابر ۸ باشد. این تعداد کم فقط به منظور تشریح نحوه عملکرد GA لحاظ

<sup>1</sup> Discrete  
<sup>4</sup> Coded

<sup>2</sup> Available Sections

<sup>3</sup> Linking

شده است. مقادیر بیهیهای رشته ها در جمعیت اولیه بطور تصادفی تولید و در ستون دوم جدول (۱-۲) نمایش داده شده اند. ستون (۳) جدول (۱-۲)، مقادیر متغیر طراحی مربوط به گروه اول را نمایش می‌دهد. این مقادیر توسط رمز گشایی زیر رشته اول هر کروموزوم با طول چهار و تعیین مقدار نظیر آن در لیست پروفیل‌های  $S$  حاصل شده است. برای مثال زیر رشته متناظر با گروه یک در اولین کروموزوم 0111 است و معادل اعشاری این عدد دودویی ۷ بدست می‌آید. اکنون مقدار متغیر طراحی متناظر با عدد ۷ در لیست پروفیل‌ها،  $S_8$  (هشتمین مقدار لیست  $S$ ) یعنی  $2/6$  می‌باشد. از آنجاییکه معادل دهگانی اعداد دودویی از صفر شروع می‌شود نه از یک، تابع نگاشت به صورت  $A_j = S_{i+1}$  بیان می‌شود که در آن  $i$  برابر عدد اعشاری رمزگشایی شده متناظر با زیر رشته گروه  $j$  می‌باشد. ستون (۵) مقادیر تابع هدف یعنی وزن خریا را به ازای کروموزوم مربوطه نشان می‌دهد. اکنون نوبت به کنترل نقض احتمالی قیود توسط طرح های متناظر با رشته مربوطه می‌رسد. بدین منظور خریای ۳ میله ای به ازای بارهای مؤثر تحلیل و مقادیر تنشها و تغییر مکانها مطابق ستونهای (۶) الی (۱۰) جدول (۲-۱) محاسبه می‌شوند. همانگونه که مشاهده می‌شود قیود تنش و تغییر مکان در کروموزوم های ۶، ۵، ۳، ۱ و ۸ نقض شده اند. اکنون ضریب نقض  $C_j$  با توجه به روابط (۱-۲) و (۲-۱۰) محاسبه و در ستون (۱۱) نمایش داده شده است. جدول (۱-۲) نشان می‌دهد که در جمعیت اولیه (نسل-۱) کروموزوم ۷ بهترین طرح می‌باشد زیرا در میان ۸ کروموزوم دارای کمترین وزن بوده و هیچ قیدی را نقض نکرده است. مقادیر تابع هدف اصلاح شده  $\phi(x)$  با استفاده از رابطه (۲-۱۱) محاسبه و در ستون (۱۲) درج شده است. در ادامه بابتی شایستگی نظیر مقادیر  $\phi(x)$  بدست آید. این کار به نحوی انجام می‌پذیرد که بهترین کروموزوم دارای حداکثر شایستگی شود. گلد برگ در کتاب خود [24] پیشنهاد می‌کند  $\phi(x)$  از یک عدد ثابت بزرگ بنحوی کسر گردد که مقادیر شایستگی غیر منفی و متناسب با ارزش واقعی هر کروموزوم بدست آید. در این رساله عدد ثابت نامبرده با توجه به رابطه (۲-۵)، بصورت مجموع حداکثر و حداقل مقدار  $\phi(x)$  جمعیت در نظر گرفته شده است. مقادیر شایستگی که به این ترتیب حاصل می‌شود در ستون (۱۳) جدول (۱-۲) نمایش داده شده است و جمع کل شایستگی جمعیت به تعداد  $nc=8$  برابر  $309/254$  می‌باشد.



جدول (۱-۲) جزئیات تعیین مقادیر تابع شایستگی - نسل ۱

شماره کروموزوم	جمعیت اولیه	A <sup>1</sup>	A <sup>2</sup>	f(x) (Kg)	σ <sub>1</sub> (Mpa)	σ <sub>2</sub> (Mpa)	σ <sub>3</sub> (Mpa)	u <sub>1</sub> (mm)	u <sub>2</sub> (mm)	C <sub>g</sub>	φ(x)	F
(۱)	(۲)	(۳)	(۴)	(۵)	(۶)	(۷)	(۸)	(۹)	(۱۰)	(۱۱)	(۱۲)	(۱۳)
۱	01110111	2.6	2.6	13.509	160.82	85.36	164.12	0.87	-1.63	0.208	24.758	43.630
۲	10011111	4.4	3.0	17.785	121.38	55.98	118.23	0.75	-1.18	0.000	17.785	50.603
۳	01000111	2.0	2.0	11.333	190.59	92.49	188.72	1.13	-1.88	0.578	37.525	30.862
۴	11110011	4.4	1.8	18.779	120.19	75.60	130.53	0.51	-1.30	0.000	18.779	49.608
۵	00110110	1.8	2.4	10.294	209.78	100.78	207.04	1.25	-2.06	0.833	44.578	23.809
۶	01100000	2.4	1.2	10.586	211.28	129.53	227.20	0.94	-2.26	0.980	52.075	16.312
۷	10101001	3.2	3.0	16.312	133.46	72.15	137.07	0.71	-1.36	0.000	16.312	52.075
۸	00111101	1.8	3.8	12.492	174.36	65.36	159.82	1.25	-1.59	0.271	26.035	42.353

در ادامه نوبت به استفاده از عملیات تکثیر، پیوند و جهش جهت تولید جمعیت نسل بعدی برای استفاده در نسل بعد می‌رسد. عملکرد فرآیند تکثیر در تطابق با نظریه داروین در مورد بقای بهترین هاست<sup>۱</sup>. در این فرآیند کروموزوم‌های با شایستگی بالا شانس بیشتری برای حضور در جمعیت منتخب پیدا کرده و جهت پیوند کاندید می‌شوند. از آنجاییکه تعداد کروموزوم‌های جمعیت در نسل بعد باید ثابت بماند، کروموزوم‌های با شایستگی کم کنار گذاشته می‌شوند. در این مثال جهت تولید جمعیت منتخب از دو روش چرخ گردان و روش پیشنهادی استفاده شده است.

در روش چرخ گردان ضریب  $\frac{F_i}{\sum_{i=1}^{nc} F_i}$  همانند ستون (۱) جدول (۲-۲) برای کلیه کروموزوم‌ها

محاسبه می‌شود. سپس همانند ستون (۲) مقدار  $\frac{F_i}{\sum_{i=1}^{nc} F_i}$  جمعیتی تعیین می‌گردد. در ادامه همانند ستون

(۳) به تعداد اعضای جمعیت اعداد تصادفی یکنواخت در محدوده صفر تا یک تولید می‌شود. اگر عدد

تصادفی تولید شده کوچکتر یا مساوی با  $\frac{F_i}{\sum_{i=1}^{nc} F_i}$  جمعیتی نظیر هر کروموزوم باشد آن کروموزوم برای

تکثیر انتخاب می‌شود. طبیعتاً کروموزومی با شایستگی بیشتر قطاع بزرگتری را در چرخ گردان به خود

<sup>۱</sup> Darwinian Survival of the Fittest

اختصاص داده و شانس بیشتری برای حضور در جمعیت منتخب ( با تعداد بیشتر از یک ) پیدا می کند. ستون (۴) شماره کروموزوم های منتخب و ستون (۵) جمعیت منتخب برای پیوند را نمایش می دهد. همانگونه که مشاهده می شود در روش چرخ گردان کروموزوم های ۴،۲ و ۷ یک بار کپی، کروموزوم ۸ دو بار کپی، کروموزوم ۱ سه بار کپی و کروموزوم های ۵،۳ و ۶ انتخاب نشده اند. البته به سبب آنکه سه عدد تصادفی تولید شده اول غیر یکنواخت می باشند در جمعیت منتخب اشکالاتی ایجاد شده است که در جمعیت های بعدی مرتفع می شود. در مثالهای عملی که تعداد اعضای جمعیت زیاد می باشد، چنین بی نظمی هایی مشکل آفرین نمی شود.

کنون نوبت به عملکرد اپراتور پیوند می رسد. در این مثال پیوند دو نقطه ای<sup>۱</sup> برای هر دو روش انتخاب بکار گرفته شده است. بدین منظور بایستی مجموعه ای از پارامترها بطور تصادفی تولید شوند. این پارامترها به یافتن زوج ها<sup>۲</sup> و دو نقطه قطع<sup>۳</sup> مربوط می شوند. جهت زوج یابی با تخصیص نرخ پیوند،  $P^c$ ، همانند ستون (۱) جدول (۲-۲) و (۴-۲) یک عدد تصادفی برای هر کروموزوم تولید و با آن مقایسه می شود.

جدول (۲-۲) جزئیات عملکرد فرآیند انتخاب بر اساس نظریه چرخ گردان - نسل ۱

فرآیند انتخاب بر اساس نظریه چرخ گردان (۱)				
$\frac{F_i}{\sum_{i=1}^n F_i}$	مقدار تصادفی ستون <sup>۱</sup>	عدد تصادفی تولید شده	شماره کروموزوم های منتخب برای تکثیر	جمعیت منتخب
(۱)	(۲)	(۳)	(۴)	(۵)
0.1411	0.1411	9.537 E-7	1	01110111
0.1636	0.3047	9.727 E-5	1	01110111
0.0998	0.4045	9.826 E-3	1	01110111
0.1604	0.5649	9.924 E-1	8	00111101
0.0770	0.6419	2.321 E-1	2	10011111
0.0527	0.6947	4.450 E-1	4	11110011
0.1684	0.8630	9.490 E-1	8	00111101
0.1369	1.0000	8.506 E-1	7	10101001

در صورتیکه این عدد از نرخ پیوند کوچکتر باشد، این کروموزوم برای پیوند انتخاب می شود. در این مثال در روش چرخ گردان  $P^c = 0.19$  و در روش پیشنهادی  $P^c = 1/10$  منظور شده است. پس از آنکه همانند

<sup>۱</sup> Two-site Crossover

<sup>۲</sup> Match

<sup>۳</sup> Cross Site

ستون (۲) جداول (۲-۳) و (۲-۴) زوجهای پیوند دو به دو شناسایی شدند، برای این زوج ها دو نقطه قطع تصادفی همانند ستونهای (۳) و (۴) جداول تولید می شود. پیوند دو نقطه ای به شرح ذیل برای زوج ها صورت می پذیرد. بعنوان مثال دو رشته A و B بصورت زیر در نظر گرفته می شود:

$$A = x_1, x_2, x_3, |x_4, x_5, x_6|, x_7, x_8$$

$$B = y_1, y_2, y_3, |y_4, y_5, y_6|, y_7, y_8$$

فرض می شود نقاط قطع تولید شده اعداد ۳ و ۶ باشد. این نقاط بطور تصادفی تولید شده و در رشته ها بصورت خطوط قائم نمایش داده شده است. پس از عمل پیوند، رشته A به A' و رشته B به B' تبدیل می شود.

$$A' = x_1, x_2, x_3, y_4, y_5, y_6, x_7, x_8$$

$$B' = y_1, y_2, y_3, x_4, x_5, x_6, y_7, y_8$$

همانگونه که مشاهده می شود پس از عمل پیوند بیتهای مابین دو نقطه قطع بین دو کروموزوم A و B جابجا شده و کروموزوم های فرزند A' و B' پدید آمده اند. با این روال جمعیت فرزندان متنی بر روشهای چرخ گردان در ستون (۵) جداول (۲-۴) نمایش داده شده اند. از دیگر مراحل رایج در عملیات GA که تکامل جمعیت برای نسل بعد را به دنبال خواهد داشت، استفاده از اپراتور جهش است. هدف از بکارگیری اپراتور جهش، کاوش بهتر فضای طراحی است. در غیاب عملکرد اپراتور جهش در صورتیکه فقط از اپراتور پیوند استفاده شود، پس از طی چندین نسل، کروموزوم برتر به تعداد اعضای جمعیت تکثیر یافته و فرآیند پیوند دیگر نمی تواند هیچگونه تغییری در کیفیت جمعیت فرزندان یا به عبارت دیگر نتایج طرح بهینه ایجاد کند. در فرآیند جهش با تخصیص نرخ جهش،  $P^m$ ، برای تک تک ژن ها (بیتهای) کروموزوم های فرزند، اعداد تصادفی یکنواخت تولید می شود. در صورتیکه این اعداد از نرخ جهش کوچکتر باشد، مقدار بیت مربوطه از صفر به یک یا بالعکس تبدیل می شود. مقدار مناسب نرخ جهش در کیفیت نتایج تاثیر بسزایی می گذارد. نرخ جهش بالا سبب می شود که نظم جمعیت فرزندان بهم خورده و عملیات GA به سمت روند کاوش تصادفی سوق داده شود در حالیکه نرخ جهش پایین سبب می شود که فضای طراحی بخوبی کاوش نشود.

در این مثال  $P^m = 0.5$  برای روش چرخ گردان منظور شده است. در مثالهای واقعی ارائه شده در بخشهای بعد، با استفاده از نرخهای جهش متعارف، نتایج خوبی بدست آمده است. ستون (۶) جداول (۲-۳) جمعیت فرزندان پس از اعمال فرآیند جهش را نمایش می دهد. این جمعیت در نسل ۲ استفاده شده و عملیاتی مشابه نسل ۱ تکرار می شود.

جدول (۲-۳) جزئیات عملکرد اپراتورهای پیوند و جهش در روش چرخ گردان - نسل ۱

عدد تصادفی تولید شده	کروموزوم های منتخب برای پیوند	نقاط قطع در پیوند دو نقطه ای		جمعیت تولید شده پس از پیوند	جمعیت تولید شده پس از جهش
		$CS_1$	$CS_2$	مشترک بر نظریه چرخ گردان	مشترک بر نظریه چرخ گردان
(۱)	(۲)	(۳)	(۴)	(۵)	(۶)
-	-	-	-	01110111	01110111
9.684 E-1	✓	5	2	01110111	01110111
7.494 E-1	✓	5	2	01110111	01110111
6.912 E-1	✓	7	5	00111111	00111111
8.157 E-1	✓	7	5	10011101	10011101
3.911 E-1	✓	4	3	11110011	11110011
5.660 E-1	✓	4	3	00101111	00101111
1.006 E-1	✓	-	-	10101001	10101001
7.698 E-1					

جداول (۲-۴) و (۲-۵) نشان می دهند که کیفیت کروموزوم های نسل دوم در قیاس با نسل اول بهتر و مقادیر نقض بطور محسوسی کاهش یافته است. این مطلب با مقدار عددی شایستگی تشخیص داده نمی شود زیرا مقدار ثابتی که  $\phi_i(x)$  از آن کسر می شود تا مقدار شایستگی بدست آید، در هر نسل متفاوت می باشد. چنانچه این مقدار ثابت در کلیه نسل ها یکسان در نظر گرفته شود، شخص می تواند افزایش در مقدار عددی شایستگی را طی نسل های متوالی مشاهده کند. همچنین در صورتیکه مقدار ثابت  $K$  در کلیه نسل ها یکسان باشد، در یک مساله کمینه سازی مقدار میانگین  $\phi_i(x)$  در خلال نسل های متوالی کاهش می یابد. چنین بهبودی بطور ریاضی با استفاده از قضیه شمای هلند<sup>۱</sup> قابل بیان می باشد.

<sup>۱</sup> Holland's theory of schemata

<sup>۴</sup> Template

قضیه بنیادی الگوریتم ژنتیک<sup>۱</sup> میزان رشد یا کاهش مورد انتظار یک شمای<sup>۲</sup> خاص طی نسل‌های متوالی و تحت اثر اپراتورهای تکثیر، پیوند و جهش را تعیین می‌کند. قضیه شما به اختصار به این شرح می‌باشد: شما مشابه یک شابلن<sup>۴</sup> عمل می‌کنید. به عنوان مثال شمایی نظیر 10\*\*\* توصیف‌کننده کلیه رشته‌هایی است که با 10 شروع و دارای طول ۵ است. شمای 11\*00 توصیف‌کننده مجموعه دو رشته (11000, 11100) می‌باشد. یک شما توسط دو مشخصه اش یعنی رتبه<sup>۵</sup> و طول<sup>۶</sup> توصیف می‌شود. رتبه یک شما،  $O(H)$ . تعداد بیت‌های با موقعیت و مقدار ثابت در آن می‌باشد. به عنوان مثال رتبه شمای 10\*\*\* برابر ۲ و رتبه شمای 11\*00 برابر ۴ است. طول شمای  $H$  یعنی  $\delta(H)$ . فاصله مابین دورترین بیت‌های با موقعیت و مقدار ثابت می‌باشد. طول شمای 10\*\*\* برابر ۱ و طول شمای 11\*00 برابر ۴ است. با استفاده از این تعاریف، قضیه بنیادی الگوریتم ژنتیک به صورت زیر بیان می‌شود [24].

$$m(H, t+1) \geq m(H, t) \frac{F(H)}{\bar{F}} \left[ 1 - P^c \frac{\delta(H)}{l-1} - O(H)P^m \right] \quad (2-11)$$

در رابطه فوق  $m$  تعداد شمای  $H$  موجود در جمعیت،  $t$  شماره نسل،  $F(H)$  میانگین شایستگی رشته‌هایی حاوی شمای  $H$ ،  $\bar{F}$  میانگین شایستگی کل جمعیت،  $P^c$  نرخ پیوند،  $P^m$  نرخ جهش،  $l$  طول کل رشته،  $O(H)$  رتبه و  $\delta(H)$  طول شمای  $H$  می‌باشد.

جدول (۲-۴) جزئیات تعیین مقادیر تابع شایستگی در روش چرخ‌گردان - نسل ۲

F	$\phi(x)$	$C_g$	$u_2$ (mm)	$u_1$ (mm)	$\sigma_3$ (Mpa)	$\sigma_2$ (Mpa)	$\sigma_1$ (Mpa)	$f(x)$ (Kg)	$A^2$	$A^1$	جمعیت حاصل از نسل ۱	شماره کروموزوم
(۱۳)	(۱۲)	(۱۱)	(۱۰)	(۹)	(۸)	(۷)	(۶)	(۵)	(۴)	(۳)	(۲)	(۱)
16.312	26.707	0.208	-1.63	0.87	164.12	85.36	160.82	13.509	2.6	2.6	01110111	1
16.312	26.707	0.208	-1.63	0.87	164.12	85.36	160.82	13.509	2.6	2.6	01110111	2
16.312	26.707	0.208	-1.63	0.87	164.12	85.36	160.82	13.509	2.6	2.6	01110111	3
17.337	25.683	0.218	-1.51	1.41	151.43	52.26	174.88	12.709	4.4	1.6	00101111	4
26.176	16.843	0.000	-1.27	0.75	127.45	62.89	128.29	16.843	3.8	3.0	10011101	5
24.240	18.779	0.000	-1.30	0.51	130.53	73.60	120.19	18.779	1.8	4.4	11110011	6
17.337	25.683	0.218	-1.51	1.41	151.43	52.26	174.88	12.709	4.4	1.6	00101111	7
26.707	16.312	0.000	-1.36	0.71	137.10	72.15	133.46	16.312	3.0	3.2	10101001	8

<sup>2</sup> The Fundamental Theorem of Genetic Algorithms  
<sup>3</sup> Schema

<sup>5</sup> Order  
<sup>6</sup> Defining Length

جدول (۲-۵) جزئیات عملکرد فرآیند انتخاب بر اساس نظریه چرخ گردان - نسل ۲

فرآیند انتخاب بر اساس نظریه چرخ گردان				
$\frac{F_i}{\sum_{i=1}^{nc} F_i}$	مقادیر حجمی ستون (۱)	عدد تصادفی تولید شده	شماره کروموزوم های منتخب برای تکثیر	جمعیت منتخب
(۱)	(۲)	(۳)	(۴)	(۵)
0.1015	0.1015	4.551 E-1	5	10011101
0.1015	0.2030	9.655 E-1	8	10101001
0.1015	0.3045	5.147 E-1	5	10011101
0.1079	0.4123	9.865 E-1	8	10101001
0.1629	0.5752	6.390 E-1	6	11110011
0.1508	0.7260	5.374 E-1	5	10011101
0.1079	0.8338	2.731 E-1	3	01110111
0.1662	1.0000	5.822 E-1	6	11110011

قفسه شما به اختصار بیان می دارد که شماهای نسبتاً کوتاه با رتبه پایین و میانگین شایستگی بالا، بطور نمایی شانس بیشتری برای حضور در جمعیت های متوالی پیدا می کنند. زیرا اپراتور تکثیر شماهای شایسته را بیشتر کپی کرده و اپراتور پیوند شماهایی با طول کوتاه را با احتمال کمتری بی نظم می کند.

جدول (۲-۶) جزئیات عملکرد اپراتورهای پیوند و جهش در روش چرخ گردان - نسل ۲

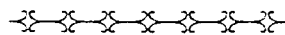
عدد تصادفی تولید شده	زوج کروموزوم های منتخب برای پیوند	نقاط قطع در پیوند دو نقطه ای		جمعیت تولید شده پس از پیوند	جمعیت تولید شده پس از جهش
		$CS_1$	$CS_2$	مشی بر نظریه چرخ گردان	مشی بر نظریه چرخ گردان
(۱)	(۲)	(۳)	(۴)	(۵)	(۶)
8.046 E-1	✓	3	5	10110101	10110111
2.660 E-1	✓	3	5	10001001	10001001
8.682 E-1	✓	1	5	10110101	10110111
6.877 E-1	✓	1	5	10001001	10001001
4.627 E-1	✓	2	5	11110111	11110111
7.364 E-1	✓	2	5	10110101	10110111
3.811 E-1	✓	2	7	01111011	01111011
4.896 E-1	✓	2	7	11110111	11110111

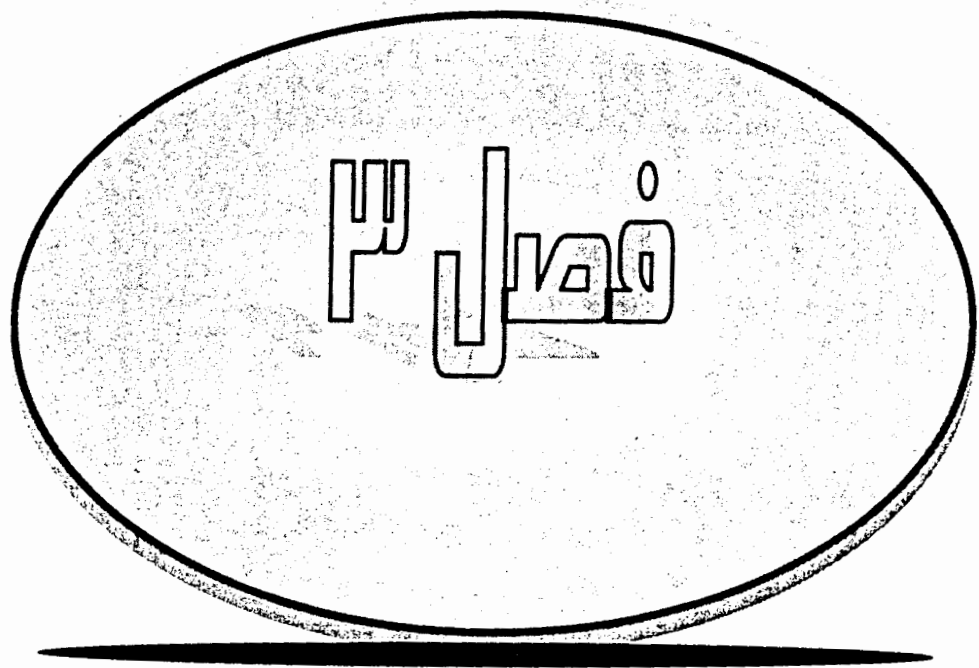
مجدداً به مثال مورد بحث بازگشته و نتایج نسل ۳ را مورد بررسی قرار می‌دهیم. جدول (۲-۷) بر اساس روش چرخ گردان نشان می‌دهد که بهترین طرح در کروموزوم هفتم به ازای  $C=0.0$ ، سازه ای با وزن ۱۴/۷۶۵ کیلوگرم می‌باشد.

جدول (۲-۷) جزئیات تعیین مقادیر تابع هدف اصلاح شده در روش چرخ گردان - نسل ۳

شماره کروموزوم	جمعیت حاصل از نسل ۲	$A^1$	$A^2$	$f(x)$ (Kg)	$\sigma_1$ (Mpa)	$\sigma_2$ (Mpa)	$\sigma_3$ (Mpa)	$u_1$ (mm)	$u_2$ (mm)	$C_g$	$\phi(x)$
(۱)	(۲)	(۳)	(۴)	(۵)	(۶)	(۷)	(۸)	(۹)	(۱۰)	(۱۱)	(۱۲)
1	10110111	3.4	2.6	16.410	133.71	76.00	139.81	66	-1.39	0.000	16.410
2	10001001	2.8	3.0	14.862	145.89	75.82	147.80	81	-1.47	0.044	15.197
3	10110111	3.4	2.6	16.410	133.71	76.00	139.81	66	-1.39	0.000	16.410
4	10001001	2.8	3.0	14.862	145.89	75.82	147.80	81	-1.47	0.044	15.197
5	11110111	4.4	2.6	20.035	110.77	66.18	117.97	51	-1.17	0.000	20.035
6	10110111	3.4	2.6	16.410	133.71	76.00	139.81	66	-1.39	0.000	16.410
7	01111011	2.6	3.4	14.765	146.29	70.83	144.74	87	-1.44	0.000	14.765
8	11110111	4.4	2.6	20.035	110.77	66.18	117.97	51	-1.17	0.000	20.035

الگوریتم ژنتیک مراحل فوق را با تولید جمعیت جدید و ارزیابی شایستگی کروموزوم های آن تا تکمیل تعداد نسل های پیش بینی شده تکرار می‌کند. اگرچه عملیات تصادفی زیادی در GA صورت می‌پذیرد، اما طرح بهینه توسط یک روش کاوش صرفاً تصادفی تعیین نمی‌شود. با بررسی کیفیت نسل های متوالی مشاهده می‌شود که GA قضیه بنیادی بیان شده در رابطه (۲-۱۱) را تحقق می‌بخشد. چنانچه در این مثال تعداد نسل سازینا ادامه یابد هیچگونه بهبودی در طرح بهینه حاصل نمی‌شود و این جواب را می‌توان به عنوان پاسخ نهایی پذیرفت. تعریف یک معیار برای تصمیم گیری در مورد خاتمه عملیات GA ضرورت دارد. یکی از این معیارها به این صورت تعریف می‌شود که در صورت یکسان شدن درصد مشخصی از ستونیا و سطرهای ماتریس جمعیت، همگرایی حاصل شده است. این مقدار می‌تواند بین ۸۰ تا ۸۵ درصد در نظر گرفته شود.





# تکنیک های نوین در الگوریتم ژنتیک



## ۳-۱ مقدمه

بسیاری از محققین مقالات متنوعی درباره بسط دامنه کاربرد GA، بهبود نتایج و افزایش سرعت محاسبات ارائه نمودند که در این فصل این روش ها مورد بررسی قرار می گیرند و چند روش جدید پیشنهادی نیز بررسی می گردد که علاوه بر افزایش کارایی و سرعت عملیات بهینه سازی ما را قادر به حل مسائلی می کند که با روش های کلاسیک الگوریتم ژنتیک قابل انجام نمی باشد. کارایی این روشهای پیشنهادی در این فصل با حل یک مثال سازه ای مربوط به خرپایی که دارای ۱۲۸ المان می باشد مورد بررسی قرار گرفته است و نتایج بدست آمده با این روش با نتایج موجود در مقالات که غالباً با استفاده از روشهای کلاسیک الگوریتم ژنتیک انجام گرفته است، مقایسه گردیده است.

بررسی نتایج و نمودارهای بدست آمده از این روش ها علاوه بر بهبود نتایج افزایش چشمگیری در سرعت عملیات بهینه سازی را نیز به همراه دارد و ما را قادر به حل مسائلی که فضای طراحی در آن ها بسیار بزرگ می باشد و با روش های معمول قابل حل نمی باشد، می کند.

همانطور که گفته شد، الگوریتم ژنتیک در پیدا کردن جواب در فضای عمومی به خوبی عمل می کند. در حقیقت، الگوریتم ژنتیک در تابع های آزمایشی جواب های خوبی را بدست می دهد. با این وجود تعداد کمی از کاربران، الگوریتم ژنتیک را در حل مسائلی که دارای تعداد زیادی از متغیرهای طراحی هستند به کار گرفته اند. اما همواره این سوال برای کاربران مطرح بوده است که چرا الگوریتم ژنتیک در پیدا کردن جواب مسائلی که دارای تعداد متغیرهای طراحی زیادی هستند به خوبی عمل نمی کند؟

یک از دلایل این مطلب این است که فضای قابل قبول<sup>۱</sup> در مقایسه با فضای عمومی طراحی بسیار کوچک می باشد و علت این امر نیز قیود هستند و قیود باعث می شوند که این فضای قابل قبول طراحی محدود بشود.

<sup>۱</sup> Feasible domain

یکی از روش های بسیار جدید و کارا در بهینه سازی بوسیله الگوریتم ژنتیک که در آن قابلیت استفاده از توانایی های کامپیوترهای موازی نیز به میزان خوبی وجود دارد روش الگوریتم ژنتیک توزیعی<sup>۱</sup> (DGA) می باشد. این روش بهینه سازی برای محاسباتی که نیاز به دقت بالایی<sup>۲</sup> دارند ابزار بسیار مناسبی می باشد و ما از آن برای طراحی و بهینه سازی سازه های خرابی استفاده می کنیم.

همانطور که در مبانی الگوریتم ژنتیک گفته شد ، اغلب مسائل بهینه سازی در عالم واقعیت به صورت مسائل مقید هستند و از آنجائیکه روش بهینه سازی الگوریتم ژنتیک توزیعی برای مسائل نامقید کاربرد دارد بنابراین برای اینکه بتوان از این روش در بهینه سازی اینگونه مسائل استفاده کرد باید تابع شایستگی در مسائل مقید را مورد اصلاح قرار داد. در این بخش ما به معرفی روشی جهت در نظر گرفتن قیود و تبدیل روش بهینه سازی الگوریتم ژنتیک توزیعی مقید به نامقید می پردازیم و به طور همزمان به معرفی مدل گسترش یافته الگوریتم ژنتیک توزیعی محیطی<sup>۳</sup> (EDGA) نیز می پردازیم. در مدل گسترش یافته الگوریتم ژنتیک توزیعی فضای طراحی به بخش های مختلفی تقسیم می شود که هر کدام از این بخشها<sup>۴</sup> دارای محیطهای<sup>۵</sup> متفاوتی می باشند و با قیود متفاوتی نیز سرکار دارند. این طریقه عمل باعث می شود که جواب های بدست آمده تنوع خوبی داشته باشند و در پایان فرآیند بهینه سازی نیز الگوریتم به جوابهای بسیار خوبی برسد.

در جاهایی که به دلیل زیاد بودن تعداد متغیرهای طراحی و بزرگ بودن فضای کاوش و همچنین محدود بودن ناحیه قابل قبول<sup>۶</sup> امکان استفاده از روشهای متداول الگوریتم ژنتیک وجود ندارد، مدل پیشنهادی به طور مؤثری فضای طراحی را مورد کاوش قرار می دهد و جواب های بسیارخوبی نیز در مقایسه با روش های متداول به دست می دهد.

<sup>۱</sup> Distributed Genetic Algorithm (DGA)

<sup>۲</sup> high performance computing

<sup>۳</sup> Environment Distributed Genetic Algorithm (EDGA)

<sup>۴</sup> island

<sup>۵</sup> environment

<sup>۶</sup> Feasible domain

امروزه، شاهد پیشرفت بسیار سریع کامپیوترها در زمینه نرم افزار و سخت افزار به خصوص در زمینه محاسبات عددی هستیم و از آنجائیکه اغلب کامپیوترهای بزرگ<sup>۱</sup> به صورت کامپیوترهای موازی یکپارچه<sup>۲</sup> هستند، بنابراین اگر الگوریتم های مورد استفاده در بهینه سازی، قابلیت استفاده از این توان بالای کامپیوترهای موازی را دارا نباشد، ارزش زیادی ندارند و امکان استفاده از این قدرت بالای محاسباتی از دست می رود.

مسائلی که در آن ها ما به دنبال پیدا کردن یک مینیمم یا ماکزیمم برای یک تابع هدف<sup>۳</sup> هستیم، به مسائل بهینه سازی معروف هستند. به منظور حل مسائل بهینه سازی یک فرآیند تکرار<sup>۴</sup>، مابین بهینه کننده که وظیفه<sup>۵</sup> تصمیم گیری در مورد نقطه<sup>۶</sup> بعدی جستجو را دارد و همچنین تحلیل کننده که وظیفه<sup>۷</sup> مشخص کردن مقدار تابع هدف را دارد، انجام می گردد. این فرآیند تکرار ممکن است هزینه بالایی داشته باشد. بنابراین استفاده از این روش ها می تواند کمک مؤثری را به ما در این زمینه بکند. برای بهینه سازی روش های متعددی وجود دارد که در اغلب آن ها علیرغم استفاده از فرآیند تکرار، امکان استفاده از روش های موازی وجود ندارد و امکان استفاده از این قابلیت مفید کامپیوترهای بزرگ فراهم نمی باشد. بنابراین به منظور استفاده از توان محاسباتی بالای این کامپیوترها نیاز به استفاده از روشهای بهینه سازی می باشد که قابلیت موازی سازی<sup>۸</sup> را داشته باشند.

"الگوریتم ژنتیک (GA) که به شبیه سازی تکامل مخلوقات و وراثت آنها می پردازد یکی از روشهای بسیار قوی مورد استفاده در بهینه سازی می باشد که امکان استفاده از آن نه تنها در مسائل پیوسته<sup>۹</sup> وجود دارد بلکه در مسائل ناپیوسته<sup>۱۰</sup> نیز قابل استفاده است. بسیاری از محققان الگوریتم ژنتیک را برای حل مسائل واقعی به کار گرفته اند و به جوابهای بسیار خوبی نیز دست یافته اند. به عنوان مثال

<sup>۱</sup> Super Computer  
<sup>۲</sup> Massive parallel Computers  
<sup>۳</sup> Objective function  
<sup>۴</sup> iteration  
<sup>۵</sup> parallelism  
<sup>۶</sup> Continues problems  
<sup>۷</sup> Discrete problems

لی<sup>۱</sup> و همکارانش به بهینه سازی مکان و هندسه مواد پیزموتریک برای توپولوژی بازوی کششی با استفاده از SLP و GA پرداختند و دانشمندان دیگر<sup>۲</sup> نیز به حل مسائل بهینه سازی خرپاها بوسیله الگوریتم ژنتیک پرداختند. به طور کلی دانشمندان بسیاری الگوریتم ژنتیک را بعنوان یک روش بسیار مناسب جهت پیدا کردن جوابهای بهینه عمومی در بهینه سازی سازه ها بکار گرفتند.

بدلیل اینکه الگوریتم ژنتیک یکی از روش های جستجو با نقاط مختلف می باشد بنابراین نیاز به تعداد دفعات تکرار بسیار زیادی جهت رسیدن به جواب بهینه دارد. در حالت های واقعی محاسبه تابع هدف کار بسیار وقتگیری می باشد و اغلب به آسانی انجام نمی گیرد. این موضوع باعث می شود که بسیاری از استفاده کنندگان این روش را برای مسائل واقعی به کار نبرند. امروزه الگوریتم ژنتیک به عنوان یکی از روش های بسیار قوی جهت استفاده در کامپیوترهای موازی مطرح می باشد. الگوریتم های ژنتیک مختلفی جهت استفاده در کامپیوترهای موازی وجود دارد و به طور کلی مدل های الگوریتم ژنتیک به سه دسته تقسیم می شود:

۱- مدل های پایه پیرو<sup>۳</sup>

۲- مدل های مجزا<sup>۴</sup> (الگوریتم ژنتیک توزیعی)

۳- مدل های سلولی<sup>۵</sup>

در این رساله توجه ما بیشتر به روشهایی است که در آن الگوریتم ژنتیک توزیعی مورد استفاده قرار می گیرد.

یکی از بزرگترین مشکلات در بهینه سازی با استفاده از الگوریتم ژنتیک در نظر گرفتن قیود می باشد. الگوریتم ژنتیک در واقع روشی است که جهت بهینه سازی مسائل نامقید به کار می رود.

<sup>۱</sup> Li et al.

<sup>۲</sup> K. Deb and S. Gulati

<sup>۳</sup> Master-slave models

<sup>۴</sup> Island models

<sup>۵</sup> Cellular model

بنابراین برای استفاده از الگوریتم ژنتیک توزیعی در مسائل مقید نیاز به استفاده از روش هایی جهت در نظر گرفتن قیود حاکم بر مسئله داریم. مطالعات بسیار زیادی برای در نظر گرفتن قیود در الگوریتم ژنتیک انجام گرفته است و در مقالات محققان روشهای مختلفی را برای این کار پیشنهاد کرده اند. یکی از راههای ساده برای در نظر گرفتن قیود استفاده از عبارت جریمه می باشد که بر اساس درجه نقض قیود به تابع هدف افزوده می شود. این روش به دو بخش عمده تقسیم می شود، که مربوط به قیود استاتیکی و دینامیکی است. روش تابع جریمه مرگ<sup>۱</sup> یک روش ساده می باشد که در آن هنگامی که یک نقطه جستجو قیود را نقض می کند، این نقطه حذف می شود و یک نقطه جستجو جدید بدست می آید که در آن نقطه قیود اقلان گردند. میخلاویچ و جانیکو<sup>۲</sup> به ارائه روش (GENOCOP) پرداختند که در آن با استفاده از یک اپراتور تصویرگر<sup>۳</sup> نقاط قابل قبول را با یک نگاشت<sup>۴</sup> بر روی نواحی قابل قبول منتقل می کند و این روش تنها در مورد مسائلی که در آنها قیود به صورت خطی می باشند قابل استفاده است. در روش ارائه شده توسط هاجلا و یو<sup>۵</sup> جواب های بدست آمده توسط عملگر الگوریتم ژنتیک ضمانت قرارگیری در ناحیه قابل قبول را دارا هستند. در حال حاضر روش های متعددی از الگوریتم ژنتیک وجود دارد که برای حل مسائل مقید از تکنیک های هدف چندگانه استفاده می کند. سوری و همکاران<sup>۶</sup> به ارائه روش COMOGA پرداختند که در آن تمام اعضای جمعیت بر اساس میزان نقض قیود رتبه بندی می شوند. همچنین روش های متعدد جدیدی نیز برای در نظر گرفتن قیود وجود دارد. رایت و همکاران<sup>۷</sup> به ارائه روشی پرداختند که در آن نقاطی که قیود در آنها نقض می شود با در نظر گرفتن برخی احتمالات امکان حضور در جمعیت منتخب را دارا می باشند. کوزل و

<sup>۱</sup> Death penalty method

<sup>۲</sup> Michalewicz and Janikow

<sup>۳</sup> Projection operators

<sup>۴</sup> mapping

<sup>۵</sup> Hajela and Yoo.

<sup>۶</sup> Surry. et al.

<sup>۷</sup> Wright et al.

میخلاقچ<sup>۱</sup> به ارائه یک روش نگاشت خاص<sup>۲</sup> پرداختند. تاک و سان<sup>۳</sup> نیز با استفاده از روش لاگرانژین قیود را در نظر گرفتند.

روش های مطرح شده در بالا برای کار با مسائل بهینه سازی مقید روش های خوبی می باشد، با این وجود این روش ها قادر به حل برخی از مسائل نیستند بخصوص در جائیکه تعداد متغیرهای طراحی زیاد و فضای طراحی بزرگ باشد این روش ها کارایی چندانی ندارند. یکی از دلایلی که باعث می شود این روش ها برای مسائلی که در آن ها فضای طراحی بزرگ می باشد به درستی کار نکند این است که ناحیه قابل قبول<sup>۴</sup> در مقایسه با فضای طراحی<sup>۵</sup> عمومی بسیار کوچک می باشد. بنابراین بسیاری از جواب ها قیود را اقلان نمی کنند. در این حالت وقتی در جمعیت اولیه هیچ نقطه جستجوی قابل قبولی وجود نداشته باشد، الگوریتم ژنتیک نیاز به مکانیزمی برای یافتن نقاط جستجویی که قیود را اقلان کند دارد.

در این قسمت ما به معرفی دو روش برای حل مسائل مقید اینگونه با استفاده از الگوریتم ژنتیک می پردازیم که توسط تانیمورا<sup>۶</sup> ارائه شده است. در ابتدا، یک تابع پناستی جدید معرفی می شود که می تواند برای مسائلی که تمام جمعیت اولیه آن در ناحیه غیرقابل قبول قرار دارد به کار رود. در این روش در ابتدا سعی می شود که شرایط نقض قیود طوری اصلاح شود که قیود اقلان گردد. پس از آن، بعد از اینکه تمام قیود اقلان گردید، مقدار تابع هدف کاهش پیدا می یابد. در روش پیشنهادی دوم با بکارگیری روش الگوریتم ژنتیک توزیعی محیطی<sup>۷</sup> (EDGAs) به حل مسائل بهینه سازی مقید پرداخته می شود. این روش نیز مانند الگوریتم ژنتیک توزیعی، دارای کارایی بالایی برای کار با کامپیوترهای

<sup>1</sup> Koziel and Michalewicz

<sup>2</sup> homomorphous mapping approach

<sup>3</sup> Tahk and Sun

<sup>4</sup> feasible domain

<sup>5</sup> design field

<sup>6</sup> Yusuke Tanimura

<sup>7</sup> Environment Distributed Genetic Algorithms(EDGAs)

موازی می‌باشد. در این روش، در یک زمان، هر بخش از فضای طراحی با قیود متفاوتی در ارتباط می‌باشد و این شیوه باعث انجام کاوش بسیار خوب فضای طراحی می‌شود.

بمنظور روشن شدن کارایی و خصوصیات این روش در طراحی بهینه‌سازها، در ابتدا به طراحی و بهینه‌سازی و مقایسه نتایج بدست آمده با نتایج موجود در مراجع مربوط به یک سازه خریابی ۱۲۸ میله ای می‌پردازیم. پس از بکارگیری این روش پیشنهادی در مورد خریابی داده شده می‌توان این روش را به عنوان یک روش عمومی در طراحی سایر سازه ها نیز به کار برد.

### ۲-۳ الگوریتم ژنتیک و الگوریتم ژنتیک توزیعی

#### ۱-۲-۳ الگوریتم ژنتیک

الگوریتم ژنتیک یک الگوریتم جستجوی تصادفی می‌باشد که بر پایه قوانین انتخاب طبیعی<sup>۱</sup> و جفت یابی<sup>۲</sup> بنا شده است. [4] در الگوریتم ژنتیک ساده<sup>۳</sup> (SGA)، متغیرهای طراحی به رشته های عددی<sup>۴</sup> تبدیل می‌گردند و جستجو در فضای رشته ها انجام می‌شود. فرآیند تبدیل فضای حقیقی به رشته های عددی توسط رمزبندی<sup>۵</sup> صورت می‌پذیرد و عملیات معکوس آن رمزگشایی<sup>۶</sup> نامیده می‌شود. عملیات تبدیل به رمز و رمزگشایی باعث می‌شود که کار کردن با مسائل گسسته آسان تر گردد.

الگوریتم ژنتیک یکی از روش های دارای نقاط جستجو چندگانه می‌باشد که به مجموعه ای از این نقاط جستجو یک جمعیت<sup>۷</sup> گفته می‌شود و به هریک از آن ها نیز یک عضو<sup>۸</sup> گفته می‌شود. در الگوریتم ژنتیک عملگرهای متفاوتی وجود دارد و اعضای جمعیت در هر مرحله مورد ارزیابی قرار

<sup>1</sup> Natural selection

<sup>2</sup> recombination

<sup>3</sup> simple GA

<sup>4</sup> Bit string

<sup>5</sup> Coding

<sup>6</sup> Decoding

<sup>7</sup> Population

<sup>8</sup> Individual

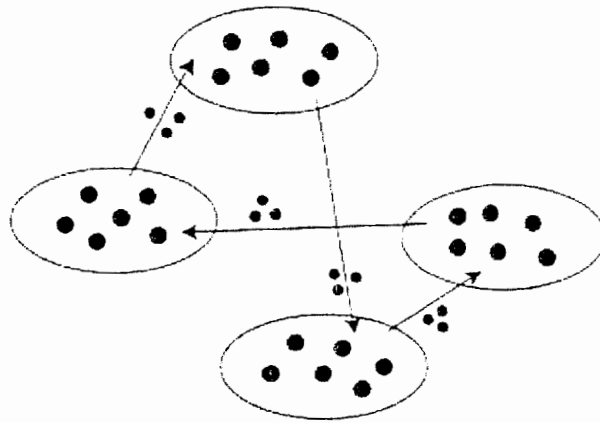
می گیرند و بهترین جوابها انتخاب می گردند. این عملگرها بترتیب ارزیابی<sup>۱</sup> و انتخاب<sup>۲</sup> نامیده می شود. بعد از فرآیند انتخاب، دو عدد از اعضا به طور تصادفی انتخاب می شوند و این اعضا دو عضو جدید را تولید می کنند. اعضای قدیمی والدین<sup>۳</sup> و اعضای جدید بدست آمده فرزندان<sup>۴</sup> نامیده می شوند. در روش الگوریتم ژنتیک فرزندان با والدین جایگزین می گردند که این عملگر پیوند<sup>۵</sup> نامیده می شود. در فرآیند پیوند، فرزندان برخی از بیت های والدین را به ارث می برند. بعد از عملیات پیوند، با استفاده از مقدار مشخصی از احتمال مقدار برخی از بیتها عوض می شود که این عملگر جهش<sup>۶</sup> نامیده می شود. عملیات ارزیابی، انتخاب، پیوند و جهش تا پیدا کردن جواب بهینه ادامه پیدا می کند. روندی که شامل تمامی این عملگرها باشد یک نسل<sup>۷</sup> نامیده می شود. در خلال نسلهای بعدی، صفت های خوب در اعضای جمعیت غالب می شود و این امر باعث افزایش و بهبود جوابهای قابل قبول می شود.

### ۳-۲-۲ الگوریتم ژنتیک توزیعی

در الگوریتم ژنتیک توزیعی (DGA) یک جمعیت بزرگ به تعدادی زیر جمعیت<sup>۸</sup> کوچکتر تقسیم می شود و الگوریتم ژنتیک مرسوم برای هر کدام از زیر جمعیت ها به طور جداگانه به کار می رود. هر کدام از این زیر جمعیت ها به عنوان یک بخش<sup>۹</sup> نامیده می شوند. تعدادی عضو از هر کدام از زیر جمعیت ها انتخاب می شود و به طور تناوبی این اعضا به زیر بخشهای دیگر مهاجرت<sup>۱۰</sup> می کنند که این مطلب در شکل ۱-۳ نشان داده شده است.

<sup>1</sup> Evaluated  
<sup>2</sup> Selection  
<sup>3</sup> parents  
<sup>4</sup> children  
<sup>5</sup> Crossover  
<sup>6</sup> Mutation  
<sup>7</sup> Generation  
<sup>8</sup> Subpopulation  
<sup>9</sup> Island  
<sup>10</sup> migration





شکل ۳-۱ الگوریتم ژنتیک توزیعی

در فرآیند مهاجرت دو پارامتر تعریف می‌شود: وقفه مهاجرت<sup>۱</sup> که مشخص کننده تعداد نسل هایی موجود بین هر مهاجرت می‌باشد و نرخ مهاجرت<sup>۲</sup> که مشخص کننده درصد اعضای انتخاب شده برای انتقال از هر زیرجمعیت در زمان مهاجرت می‌باشد. هر کدام از زیر جمعیت ها را می‌توان به یکی از پردازشگرهای کامپیوترهای موازی نسبت داد و ارتباط بین این پردازنده ها فقط در زمان مهاجرت رخ می‌دهد. آرایش (نحوه) مهاجرت انتخاب شده به صورت یک حلقه با مقصدهای تصادفی<sup>۳</sup> می‌باشد که هر کدام از زیر جمعیت ها دارای یک مقصد می‌باشند و این مقصدها به طور تصادفی در هر دوره تناوب مهاجرت مشخص می‌شود. که این مطلب در شکل ۳-۱ نشان داده شده است. بیت های مهاجرت کننده به صورت تصادفی از هر زیر جمعیت انتخاب می‌شوند. عملگر مهاجرت، اعضا را از یک زیر جمعیت انتخاب کرده و به زیر جمعیت دیگری می‌فرستد، که ممکن است شامل اعضای با ساختارهای بسیار متفاوت باشد. بعد از فرآیند مهاجرت، الگوریتم ژنتیک جمعیت های انتقال یافته را با باقیمانده اعضای جمعیت ترکیب می‌کند. بنابراین در الگوریتم ژنتیک توزیعی، از آنجائیکه یک جمعیت به تعدادی زیر جمعیت تبدیل می‌شود بنابراین تعداد اعضای هر زیرجمعیت کمتر است و

<sup>۱</sup> Migration interval

<sup>۲</sup> Migration rate

<sup>۳</sup> Ranom Destination

همگرایی<sup>۱</sup> سریعتر حاصل می شود. لذا الگوریتم ژنتیک توزیعی دارای تنوع جواب بسیار زیادی می باشد و در مقایسه با الگوریتم ژنتیک معمولی دارای قابلیت جستجوی بسیار بهتری در فضای طراحی می باشد.

### ۳-۳ الگوریتم ژنتیک برای مسائل مقید

#### ۳-۳-۱ اصلاح تابع هدف برای مسائل مقید

همانطور که قبلاً گفته شد، الگوریتم ژنتیک، روشی برای حل مسائل نامقید می باشد. بنابراین برای حل مسائل مقید تابع هدف یا متغیرهای طراحی باید مورد اصلاح قرار گیرند. روشهای متعددی برای در نظر گرفتن قیود در الگوریتم ژنتیک وجود دارد که در مراجع مختلف مورد بررسی قرار گرفته است. بارزترین این روشها در رابطه ۳-۱ نشان داده شده است:

$$\text{eval}(x) = \begin{cases} f(x), & \text{if } x \text{ is in feasible domain} \\ f(x) + \text{penalty}(x), & \text{otherwise} \end{cases} \quad (۳-۱)$$

که در آن تابع شایستگی از مقادیر تابع هدف و تابع جریمه تشکیل شده است. هنگامی که یک جواب قیود را اقماع نکند، مقدار تابع شایستگی بدتر می شود و این جواب نیز بدتر می شود و هنگامی که فضای قابل قبول نسبت به فضای طراحی بسیار کوچک باشد تعداد زیادی از جوابها که در الگوریتم ژنتیک تولید می شود خارج از ناحیه قابل قبول قرار می گیرد و کاوش مؤثر فضای طراحی نمی تواند به خوبی انجام گیرد. این مشکلی است که در تمام مسائل بهینه سازی هنگامی که تعداد متغیرهای طراحی زیاد است یا فضای طراحی بزرگ است بوجود می آید.

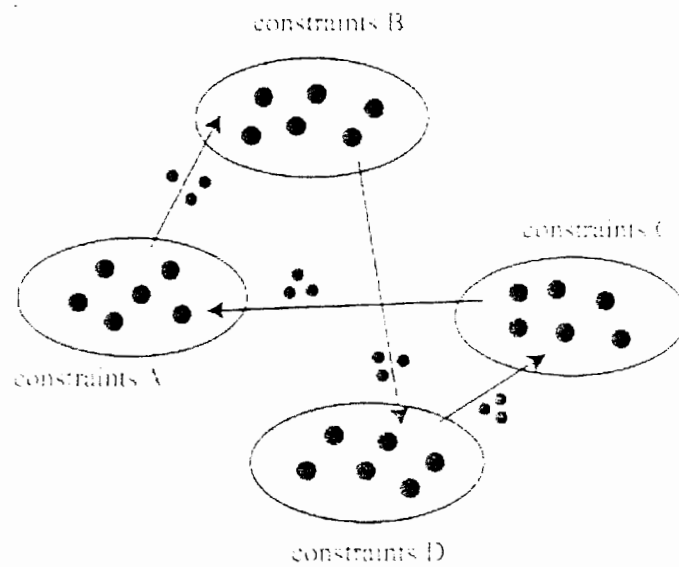
<sup>۱</sup> Convergence

برای رفع این مشکل تانیمرا تابع شایستگی زیر را معرفی کرده است:

$$eval(x) = \begin{cases} f(x), & \text{if } x \text{ is in feasible domain} \\ \text{Constant} + \text{Max}(\text{Constraint}(x)_i), & \text{otherwise} \end{cases}$$

(۲-۳)

در این تابع شایستگی، هنگامی که هیچ عضوی در اعضای جمعیت وجود ندارد که قیود را اقلان کند، در ابتدا، مقدار قیود نقض شده کاهش داده می شود. بعد از چند نسل، جواب هایی که قیود را اقلان کنند پیدا می شوند. بعد از پیدا کردن جواب هایی که قیود را اقلان کردند مقدار تابع هدف مینیمم می شود.



شکل ۲-۳ الگوریتم ژنتیک توزیعی محیطی

### ۳-۳-۲ الگوریتم ژنتیک توزیعی محیطی

اغلب در مسائل دنیای واقعی، قیود بسیار زیادی وجود دارد که در این حالت از روش الگوریتم ژنتیک توزیعی محیطی (EDGA) می توان استفاده کرد. در روش الگوریتم ژنتیک توزیعی محیطی در هر یک از بخش ها پارامترهایی مانند تعداد اعضای جمعیت و غیره یکسان است. از سوی دیگر، محیط هر کدام از بخش ها در الگوریتم ژنتیک توزیعی محیطی متفاوت است. در این رساله روش الگوریتم ژنتیک توزیعی محیطی برای انواع مختلفی از قیود به کار گرفته شده است. در این مورد، تابع شایستگی هر کدام از قسمت ها نیز متفاوت در نظر گرفته می شود و نتایج بدست آمده در هر قسمت با هم مقایسه می گردد. در ابتدا هر کدام از بخش ها<sup>۱</sup> به تعدادی زیرگروه تقسیم می شود که تعداد گروهها به تعداد قیود می باشد. یکی از موارد بسیار مهم در جستجوی الگوریتم ژنتیک توزیعی محیطی می باشد و از آنجائیکه در روش الگوریتم ژنتیک توزیعی محیطی هر کدام از بخشها عهده دار یکی از قیود می باشد، بنابراین تنوع جوابها برای ما حاصل می شود و به طور کلی با کاربرد الگوریتم ژنتیک توزیعی محیطی در مورد مسائل مقید، همگرایی سریع الگوریتم بهمراه جواب های بهینه خوب بدست می آیند. در ادامه به ارائه یک مثال می پردازیم که دارای تعداد متغیرهای طراحی بسیار زیادی است و با روش های مطرح شده در این بخش مورد بررسی قرار گرفته است و نتایج بدست آمده با نتایج موجود در مراجع مقایسه و مشاهده می شود که استفاده از این روش ها علاوه بر بهبود کیفی نتایج تأثیر قابل ملاحظه ای در سرعت همگرایی الگوریتم به سمت جوابهای بهینه از خود نشان می دهد.

<sup>۱</sup> Island

## ۳-۴ مثالهای عددی

## ۳-۴-۱ طراحی یک سازه خرپایی

در این بخش، روش های مطرح شده با استفاده از یک مثال سازه ای مورد بررسی قرار می گیرد و طراحی یک سازه خرپایی به عنوان یک مثال عددی انجام می شود. مدلی از سازه خرپایی در شکل ۳-۳ و پارامترهای سازه خرپایی نیز در جدول ۳-۱ نشان داده شده است. این سازه خرپایی دارای ۱۲۸ المان با مقطع دایره ای می باشد. نیرو در قسمت بالایی خرپا وارد می شود. تابع هدف ما حجم کلی این سازه می باشد که با استفاده از روش توضیح داده شده در قسمت الگوریتم ژنتیک توزیعی مینیمم می شود. متغیرهای طراحی سطح مقطع هر یک از اعضای سازه خرپایی می باشد. که این متغیرهای طراحی پیوسته نبوده و دارای ۸ مقدار ناپیوسته<sup>۱</sup> می باشند.

مقدار تغییر مکان گرهی که نیرو بر آن اعمال می شود همچنین تنش بوجود آید در هر المان باید بین مقادیر مطلوب باشد که در واقع این ها قيود مسئله می باشند. مثال مطرح شده در این قسمت یک حالت ساده بهینه سازی سطح مقطع سازه خرپایی می باشد و این روش در بهینه سازی همزمان سطح مقطع، هندسه و توپولوژی خرپاها در فصل پنجم بکار گرفته شده که نتایج خوبی نیز حاصل شده است. بهینه سازی این مثال در مراجع با استفاده از یک کامپیوتر چند شاخه<sup>۲</sup> انجام گرفته است اما با توجه به عدم دسترسی ما به این سیستم های کامپیوتری بهینه سازی این مثال با استفاده از یک سیستم کامپیوتری PC و به کارگیری برنامه Matlab 7.0 انجام شده است و جوابهای قابل قبولی نیز بدست آمده است.

<sup>۱</sup> Discrete values

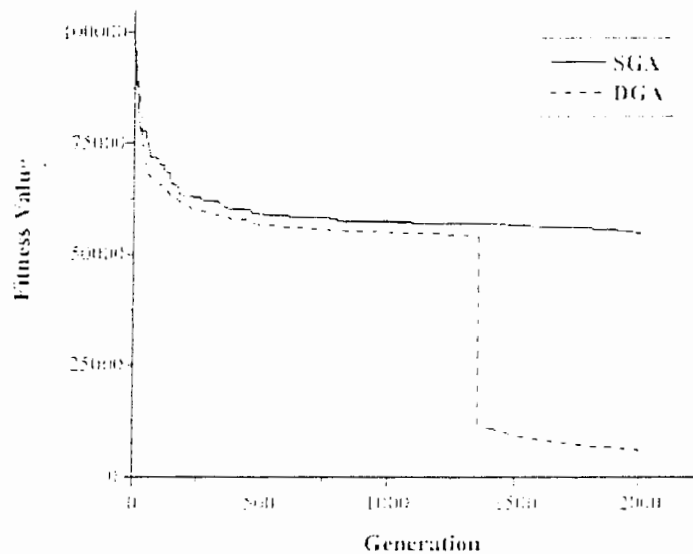
<sup>۲</sup> PC Cluster

جدول ۳-۱: پارامترهای سازه خرابیایی

متغیرهای طراحی	
$A_i ; i = 1, \dots, 640$	متغیرهای سطح مقطع :
اطلاعات قيود	
<p><i>Island 0-3</i></p> $\text{eval}(x) = \begin{cases} f(x), & \text{if } x \text{ is in feasible domain} \\ \text{constant} + \text{Max}(\text{constraint}_1(x)_i) + \text{Max}(\text{constraint}_2(x)_i), & \text{otherwise} \end{cases}$	
<p><i>Island 4-5</i></p> $\text{eval}(x) = \begin{cases} \text{constant} + \text{Max}(\text{constraint}_2(x)_i), & \text{if } x \text{ satisfy constraint}_1 \\ \text{constant}_2 + \text{Max}(\text{constraint}_1(x)_i), & \text{otherwise} \end{cases}$	
<p><i>Island 6-7</i></p> $\text{eval}(x) = \begin{cases} \text{constant} + \text{Max}(\text{constraint}_2(x)_i), & \text{if } x \text{ satisfy constraint}_2 \\ \text{constant}_2 + \text{Max}(\text{constraint}_2(x)_i), & \text{otherwise} \end{cases}$	
لیست پروفیل‌های موجود	
$A_i \in S = \{ 1.0, 5.0, 10.0, 20.0, 40.0, 60.0, 80.0, 100.0 \} \text{ (mm}^2 \text{)} ; i = 1, \dots, 640$	
اطلاعات بارگذاری	
$F_y$ +1000 N	$F_x$ 0
مرد	وضعیت
A(x)	1
خصوصیات الگوریتم	
$1.0 \times 10^4 \text{ Mpa}$	مدول الاستیسیته
256	تعداد نژادها
640	تعداد اعضاها
پیوند تک نقطه ای	روش پیوند
0.8	نرخ پیوند
0.001	نرخ جهش
500	تعداد نسلها
حلقه ای	توپولوژی مهاجرت
5 نسل	وقفه مهاجرت
0.2	نرخ مهاجرت

در این حالت تنها تنش به عنوان قید در نظر گرفته می‌شود و جواب‌ها با استفاده از الگوریتم ژنتیک ساده بدست آمده است.

مقادیر تابع شایستگی در مقایسه با تابع هدف بسیار بزرگ می‌باشد. با توجه به این شکل، مقادیر شایستگی در خلال نسل‌های مختلف تغییر نمی‌کند. این نتیجه به این معنی است که هیچ کدام از اعضا نمی‌تواند قیود مربوط به الگوریتم ژنتیک را اقناع کند. بنابراین، جوابها را با هیچ کدام از پارامترهای تابع جریمه هنگامی که معادله ۱-۳ را مورد استفاده قرار می‌دهیم نمی‌توان بدست آورد. این نتایج با جواب های بدست آمده بوسیله الگوریتم ژنتیک توزیعی تقریباً یکسان است.



شکل ۳-۵ روند تابع شایستگی (معادله ۳-۲)

در شکل ۳-۵، روند جواب های الگوریتم ژنتیک ساده و الگوریتم ژنتیک توزیعی هنگامی که روش پیشنهادی تانیمرا (معادله ۲-۳) مورد استفاده قرار می‌گیرد را نشان می‌دهد. مقدار جریمه ۵۰۰۰۰ در این مورد استفاده می‌شود که این مقدار از مقدار تابع هدف و قیود بیشتر است. الگوریتم ژنتیک توزیعی شایستگی را به مقدار زیادی در حول و حوش نسل ۱۳۰۰م کاهش می‌دهد. در این نقطه، الگوریتم ژنتیک به نقطه ای را که در آن قیود اقناع می‌شود و مقدار تابع شایستگی جابجا می‌شود

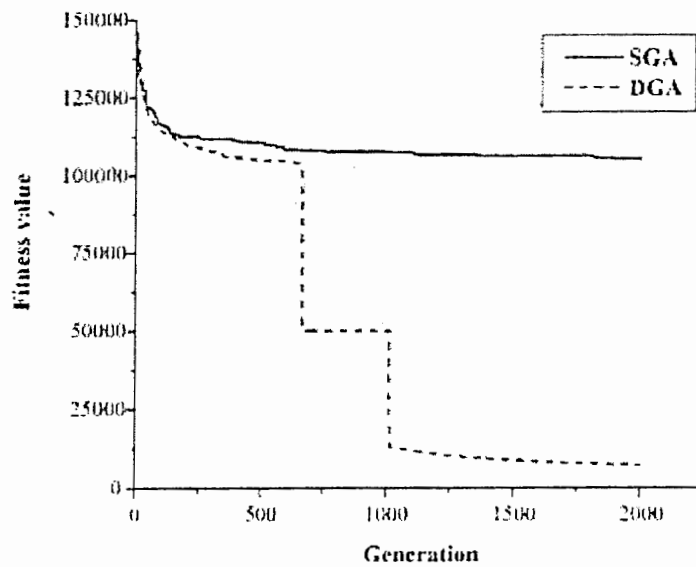
دست پیدا می کند. الگوریتم ژنتیک ساده نیز مقدار تابع شایستگی را کاهش می دهد و این بدان معنی است که الگوریتم ژنتیک ساده نیز سعی در پیدا کردن نقطه ای که قیود را اقلان کند می باشد اما در انجام این امر موفق نمی باشد. با توجه به نتایج بدست آمده، مشخص است که روش پیشنهادی برای طراحی سازه های خرپایی در الگوریتم ژنتیک توزیعی هنگامی که فضای طراحی بزرگ است، بسیار راهگشا و مؤثر می باشد.

### ۳-۴-۳ قابلیت کاوش الگوریتم های ژنتیک توزیعی

از آنجائیکه در الگوریتم ژنتیک توزیعی، جمعیت به تعدادی زیر جمعیت تبدیل می شود، تعداد در زیر جمعیت ها کم می باشد و این امر باعث افزایش سرعت همگرایی می شود و همچنین بدلیل مکانیزم مهاجرت، تنوع جوابها در خلال فرآیند کاوش حاصل می شود. بدلیل ویژگی های الگوریتم ژنتیک توزیعی در مقایسه با الگوریتم های ژنتیک متداول، این روش دارای قابلیت کاوش بسیار بالاتر فضای طراحی می باشد. در شکل ۳-۶ روند تغییرات شایستگی الگوریتم ژنتیک ساده و الگوریتم ژنتیک توزیعی نشان داده شده است. در این مورد، تغییر مکان و تنش هر دو به عنوان قیود در نظر گرفته می شوند. بنابراین در این مورد مسئله در مقایسه با حالتی که تنها تنش به عنوان قید در نظر گرفته می شود پیچیده تر می شود. در این بهینه سازی مقدار ثابت جریمه برابر ۵۰۰۰۰ در نظر گرفته می شود.

الگوریتم ژنتیک ساده قادر به پیدا کردن نقطه که در آن قیود اقلان شوند نمی باشد. در صورتیکه الگوریتم ژنتیک توزیعی قادر به کاهش مقدار شایستگی می باشد.





شکل ۳-۶ مقایسه الگوریتم های ژنتیک ساده و توزیعی

الگوریتم ژنتیک توزیعی مقدار شایستگی را در بین نسلهای ۷۰۰ و ۱۰۰۰ به میزان قابل توجهی کاهش می‌دهد و این دو حالت مربوط به نقاطی است که الگوریتم ژنتیک توزیعی نقاطی را که در این دو نقطه دو قید به ترتیب اقلان می‌شوند را پیدا می‌کند. با استفاده از این مشاهدات می‌توان به این نتیجه رسید که الگوریتم ژنتیک توزیعی در مقایسه با الگوریتم های ژنتیک ساده قابلیت کاوش بهتر فضای طراحی را دارا می‌باشد. این نتایج نه تنها در طراحی سازه های خرابایی بدست می‌آید بلکه در بکارگیری هر تابع عددی آزمایشی نیز نتایج به همین منوال بدست می‌آید.

### ۳-۴-۴ کارایی روش الگوریتم های ژنتیک توزیعی محیطی

در قسمت ۳-۳-۲ الگوریتم ژنتیک توزیعی محیطی معرفی گردید. در این بخش کارایی الگوریتم ژنتیک توزیعی محیطی مورد بررسی قرار می‌گیرد. در شکل ۳-۷ روند شایستگی الگوریتم ژنتیک

توزیعی و الگوریتم ژنتیک توزیعی محیطی نشان داده شده است. در این حالت هر دو قید تغییر مکان و تنش در نظر گرفته شده است و مقدار جریمه برابر ۵۰۰۰۰ در نظر گرفته شده است

تابع شایستگی هر کدام از بخش ها به شکل زیر می باشد:

#### Island 0-3

$$\text{eval}(x) = \begin{cases} f(x), & \text{if } x \text{ is in feasible domain} \\ \text{constant} + \text{Max}(\text{constraint}_1(x)), + \text{Max}(\text{constraint}_2(x)), & \text{otherwise} \end{cases}$$

#### Island 4-5

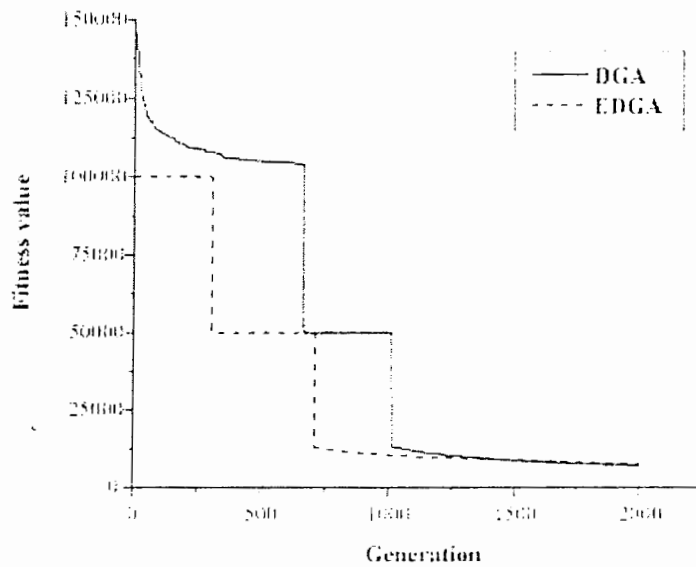
$$\text{eval}(x) = \begin{cases} \text{constant} + \text{Max}(\text{constraint}_2(x)), & \text{if } x \text{ satisfy constraint}_1 \\ \text{constant}_2 + \text{Max}(\text{constraint}_1(x)), & \text{otherwise} \end{cases}$$

#### Island 6-7

$$\text{eval}(x) = \begin{cases} \text{constant} + \text{Max}(\text{constraint}_1(x)), & \text{if } x \text{ satisfy constraint}_2 \\ \text{constant}_2 + \text{Max}(\text{constraint}_2(x)), & \text{otherwise} \end{cases}$$

(۳-۳)

با توجه نمودار صفحه بعد، مشخص است که الگوریتم ژنتیک توزیعی محیطی در مقایسه با الگوریتم ژنتیک توزیعی جواب ها را با سرعت بیشتری کاوش می کند. در ضمن الگوریتم ژنتیک توزیعی محیطی دارای مکانیزمی برای حفظ تنوع جواب ها می باشد که این مکانیزم باعث رسیدن به نتایج بسیار خوبی می شود.

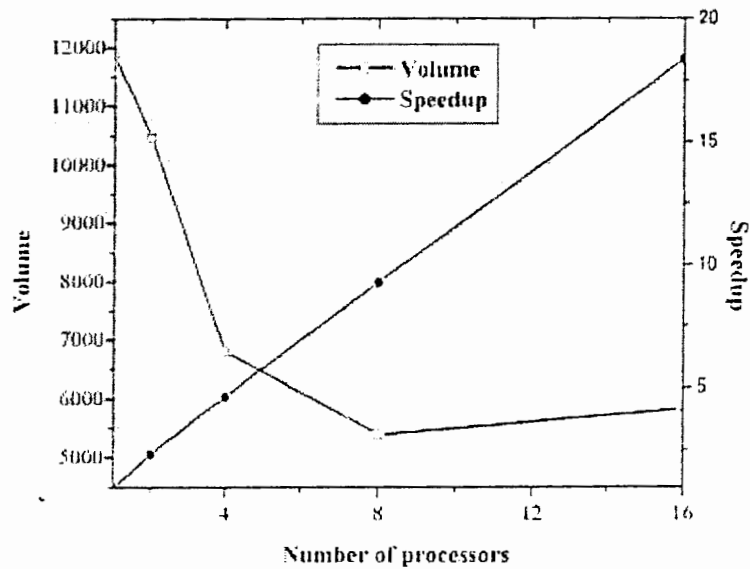


شکل ۳-۷ مقایسه الگوریتم های ژنتیک توزیعی و توزیعی محیطی

### ۳-۴-۵ بهبود سرعت

در الگوریتم ژنتیک توزیعی، هرکدام از زیرجمعیت ها به یکی از پردازشگرها اختصاص می یابد. هنگامیکه تعداد پردازشگرها متفاوت باشد، مدل الگوریتم ژنتیک توزیعی نیز متفاوت می شود. هنگامی که ارتباطات شبکه تنها در زمان مهاجرت رخ دهد، هزینه شبکه بسیار کاهش می یابد. بنابراین الگوریتم ژنتیک توزیعی برای مدل سازی موازی<sup>۱</sup> روشی بسیار مناسب می باشد. در شکل ۳-۸ بهبود سرعت در مقابل افزایش تعداد پردازشگرها نشان داده شده است. افزایش سرعت بدست آمده تقریباً به صورت خطی یا بیشتر از آن با افزایش تعداد پردازشگرها افزوده می شود.

<sup>۱</sup> parallel model



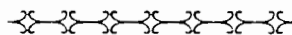
شکل ۳-۸ بهبود سرعت

### ۳-۵ نتیجه گیری

در این بخش از رساله در ابتدا ما به بررسی روش الگوریتم ژنتیک توزیعی (DGAs) برای طراحی سازه های خرپایی پرداختیم و در ادامه به طور خاص ما به بررسی و بحث در مورد قیود مختلف پرداختیم.

اغلب فضای قابل قبول طراحی در مسائلی که دارای تعداد زیادی از متغیرهای طراحی هستند و قیود زیادی نیز دارند بسیار محدود می باشد. بنابراین بیشتر کروموزوم های الگوریتم ژنتیک توزیعی کروموزوم های مرده می باشند و قیود را اقناع نمی کنند. در این موارد، الگوریتم ژنتیک توزیعی نمی تواند فضای طراحی را به طور مؤثری مورد کاوش قرار دهد. بنابراین وجود مکانیزمی که در آن کروموزوم های مرده تولید نشود بسیار مهم می باشد. بدین منظور یک مجموعه از تابع های شایستگی برای الگوریتم ژنتیک توزیعی بکار گرفته شد که، در ابتدا الگوریتم ژنتیک توزیعی مقادیر نقض قیود را کاهش می دهد و به پیدا کردن جوابهایی می پردازد که قیود را اقناع کنند. بعد از پیدا کردن جوابهایی

که قیود را اقلع می کنند، مقادیر تابع هدف کاهش می یابد. با این آهنگ الگوریتم ژنتیک توزیعی می تواند به سمت جواب های بهینه برود. در یک زمان، الگوریتم ژنتیک توزیعی محیطی نیز بکار گرفته شد. در الگوریتم ژنتیک توزیعی محیطی قیود متفاوتی در هر بخش<sup>۱</sup> اعمال می شود و این روش باعث افزایش قابلیت کاوشگری الگوریتم می شود. در مثال سازه خرابایی، مراجع از سیستم های کامپیوتری موازی برای بکارگیری *DGA* و الگوریتم ژنتیک توزیعی محیطی استفاده کرده اند. که در اینجا به دلیل عدم دسترسی به این سیستم های کامپیوتری به مدل سازی این روش با استفاده از یک کامپیوتر PC خانگی و بکارگیری نرم افزار Matlab 7.0 پرداخته شده است. لازم بذکر است که با بکارگیری روش الگوریتم ژنتیک توزیعی محیطی می توان اقدام به طراحی و بهینه سازی طیف وسیعی از مسائل سازه ای کرده و از این روش به عنوان یک شیوه کاربردی، در طراحی سازه های بزرگ واقعی بهره جست.



---

<sup>۱</sup> Island

## ۴-۱ مقدمه

بسته نرم افزاری Matlab v7.0 که در تاریخ ششم ماه می سال ۲۰۰۴ به بازار آمد یکی از قویترین برنامه هایی است که قابلیت استفاده در فرآیند بهینه سازی با استفاده الگوریتم ژنتیک را دارا می باشد. این نرم افزار علاوه بر توان بالای محاسبات ریاضی، بدلیل سازگاری با ماتریس ها و عملیات مربوط به آن ها این امکان را به برنامه نویس می دهد که علاوه بر استفاده از منوهای مربوط به بهینه سازی که در نرم افزار موجود می باشد به ایجاد اپراتورهای جدید مربوط به بهینه سازی بپردازد.

در بهینه سازی با استفاده از الگوریتم ژنتیک برای اینکه کار کردن با مسائل گسسته آسان تر شود معمولاً متغیرهای طراحی را با عملیات کدگذاری به رشته های عددی تبدیل می کنند بنابراین یکی از روشهای مناسب برای کار با این رشته های عددی تبدیل آن ها به ماتریسها می باشد و چون نرم افزار Matlab v7.0 ماتریس ها و عملیات آن ها را به خوبی می شناسد بنابراین انجام این کار بسیار آسان می باشد. علاوه بر این ها یکی از مزایای نرم افزار Matlab v7.0 این است که در آن جعبه ابزاری برای بهینه سازی بوسیله الگوریتم ژنتیک می باشد که بسیاری از اپراتور های الگوریتم ژنتیک به طور پیش فرض در آن وجود دارد و در تمامی قسمت ها در جائیکه نیاز به تعریف یک اپراتور جدید مثلاً یک عملگر پیوند خاص یا یک اپراتور تولید کننده جمعیت اولیه خاص باشد، این کار به راحتی امکان پذیر است.

یکی دیگر از مزایای بسیار مهم استفاده از نرم افزار Matlab v7.0 در بهینه سازی بوسیله الگوریتم ژنتیک امکان استفاده از نمودارهای مختلفی می باشد که این نرم افزار در حین بهینه سازی برای ما ترسیم می کند. به طور مثال در هنگام اجرای برنامه در هر قسمت می توان مقدار تابع شایستگی را با پیشرفت نسلیها به صورت یک گراف مورد ارزیابی قرار داد و روند بهینه سازی را با دیدی باز مشاهده کرد و در صورتی که بهینه سازی به درستی پیش نمی رود مثلاً مقدار تابع هدف با پیشرفت نسلیها تغییر مطلوب چندانی نمی کند و یا اینکه الگوریتم در دام بهینه های محلی افتاده است

با تغییر پارامترهای الگوریتم ژنتیک این مشکلات را حل کرد یا در جهت کارا تر کردن هرچه بیشتر الگوریتم گام برداشت.

در این رساله ما علاوه بر استفاده از کارایی های بالای نرم افزار Matlab v7.0 در امر بهینه سازی سازها در بسیاری از مثالها و قسمتها با استفاده از قابلیت های برنامه نویسی نرم افزار به ایجاد اپراتورهای مختلف مانند: توابع جریمه خاص یا نمودارهای مختلف پرداخته ایم و با تلفیق قابلیت های نرم افزار با تواناییهای برنامه نویسی به بهترین شکل مثال ها را مورد مدل سازی و بهینه سازی قرار داده ایم و ضمن بکارگیری موارد پیشنهادی در قسمت های مختلف به مقایسه نتایج و نمودارها با مراجع مختلف پرداخته ایم و با بررسی جداول خروجی و نمودارها در جاهائیکه بهبود در نتایج و یا افزایش سرعت همگرایی الگوریتم حاصل شده است به خوبی قابل تشخیص می باشد.

در این فصل ما ابتدا توضیح مختصری در مورد الگوریتم ژنتیک در Matlab v7.0 می دهیم و به معرفی قابلیت های جعبه ابزار الگوریتم ژنتیک پرداخته و قسمت های مختلف آن را مورد بررسی قرار می دهیم. در ادامه به منظور آشنایی با نحوه عملکرد فرآیند بهینه سازی با استفاده از نرم افزار Matlab v7.0 به حل یک مثال معروف بهینه سازی که دارای مقادیر بهینه محلی بسیار زیادی نیز می باشد با این نرم افزار می پردازیم. در کارایی های هرچه بیشتر این نرم افزار را در حل مثال های پیچیده تر سازه ای در فصل بهینه سازی سازه ها در فصول بعدی مشاهده خواهیم کرد.

## ۴-۲ الگوریتم ژنتیک در Matlab v7.0

الگوریتم ژنتیک یکی از روشهای بسیار مناسب حل مسائل بهینه سازی می باشد که بر پایه فرآیند انتخاب طبیعی<sup>۱</sup> که براساس تکامل بیولوژیکی بنا شده است عمل می کند. الگوریتم ژنتیک در هر قسمت با استفاده از فرآیند تکرار به اصلاح اعضای جمعیت جوابها می پردازد. به این صورت که در هر مرحله، الگوریتم ژنتیک به صورت تصادفی به انتخاب اعضا<sup>۲</sup> از میان جمعیت<sup>۳</sup> موجود جهت والدین<sup>۴</sup> می پردازد که این والدین در نسل بعدی برای تولید فرزندان<sup>۵</sup> مورد استفاده قرار می گیرند. بعد از ایجاد نسلهای متوالی، سرانجام جمعیت به سمت یک جواب بهینه<sup>۶</sup> میل می کند.

الگوریتم ژنتیک را برای حل مسائل بهینه سازی گوناگونی که در آنها تابع هدف<sup>۷</sup> ناپیوسته، دیفرانسیل ناپذیر، تصادفی و یا غیرخطی می باشد و حل آنها با استفاده از الگوریتم های استاندارد قابل انجام نیست، می توان به خوبی به کار گرفت.

الگوریتم ژنتیک در هر گام با استفاده از سه فرآیند اصلی به ایجاد نسل بعدی از جمعیت جاری می پردازد :

- **فرآیند انتخاب** به انتخاب اعضای که والدین نامیده می شوند و در تولید اعضای جمعیت نسل بعدی شرکت می کنند، می پردازد.
- **فرآیند پیوند** در ترکیب والدین جهت ایجاد فرزندان برای نسل بعدی ایفای نقش می کند.
- **فرآیند جهش** به ایجاد تغییرات تصادفی در والدین جهت ایجاد فرزندان می پردازد.

<sup>1</sup> Natural selection  
<sup>2</sup> individuals  
<sup>3</sup> population  
<sup>4</sup> parents  
<sup>5</sup> childs  
<sup>6</sup> optimal solution  
<sup>7</sup> objective function



الگوریتم ژنتیک در مقایسه با سایر الگوریتم های استاندارد از دو جهت دارای تفاوت هایی می باشد

که در جدول زیر به طور خلاصه این تفاوت ها نشان داده شده است.

جدول ۴-۱ تفاوت های الگوریتم ژنتیک با الگوریتم های استاندارد

الگوریتم های استاندارد	الگوریتم ژنتیک
در هر فرآیند تکرار، یک نقطه را تولید می کند و نقاط پی در پی در پی در نهایت به جواب بهینه می رسد.	در هر فرآیند تکرار، جمعیتی از نقاط را تولید می کند و این جمعیت در نهایت به جواب بهینه می رسد.
نقطه پی در پی در پی بعدی را با استفاده از محاسبات قطعی انتخاب می کند	جمعیت بعدی را با استفاده از محاسباتی که مربوط به انتخاب های تصادفی می باشد انتخاب می کند.

#### ۴-۲ بکارگیری الگوریتم ژنتیک

برای استفاده از الگوریتم ژنتیک در جعبه ابزار نرم افزار Matlab v7.0 دو روش وجود دارد:

- فراخوانی تابع الگوریتم ژنتیک GA در خط دستور<sup>۱</sup>.
- استفاده از ابزار الگوریتم ژنتیک<sup>۲</sup> که یک رابط گرافیکی<sup>۳</sup> مربوط ژنتیک الگوریتم می باشد.

در این قسمت توضیح مختصری در ارتباط با این روش ها داده می شود.

#### ۴-۲-۱ فراخوانی تابع الگوریتم ژنتیک GA در خط دستور

برای استفاده از الگوریتم ژنتیک در خط دستور، فراخوانی تابع الگوریتم ژنتیک ga با استفاده از

ساختار زیر انجام می گیرد:

```
[x fval] = ga(@fitnessfun, nvars, options)
```

<sup>۱</sup> command line

<sup>۲</sup> Genetic Algorithm Tool

<sup>۳</sup> graphical interface

که در آن

- @fitnessfun ابزار برای تابع شایستگی می باشد.
- nvars برابر تعداد متغیرهای مستقل تابع شایستگی می باشد.
- options ساختاری شامل گزینه های مربوط به الگوریتم ژنتیک می باشد. که اگر این گزینه ها تغییر داده نشود الگوریتم ژنتیک از مقادیر پیش فرض خود استفاده می کند.

نتایج با استفاده از پارامترهای زیر نشان داده می شود:

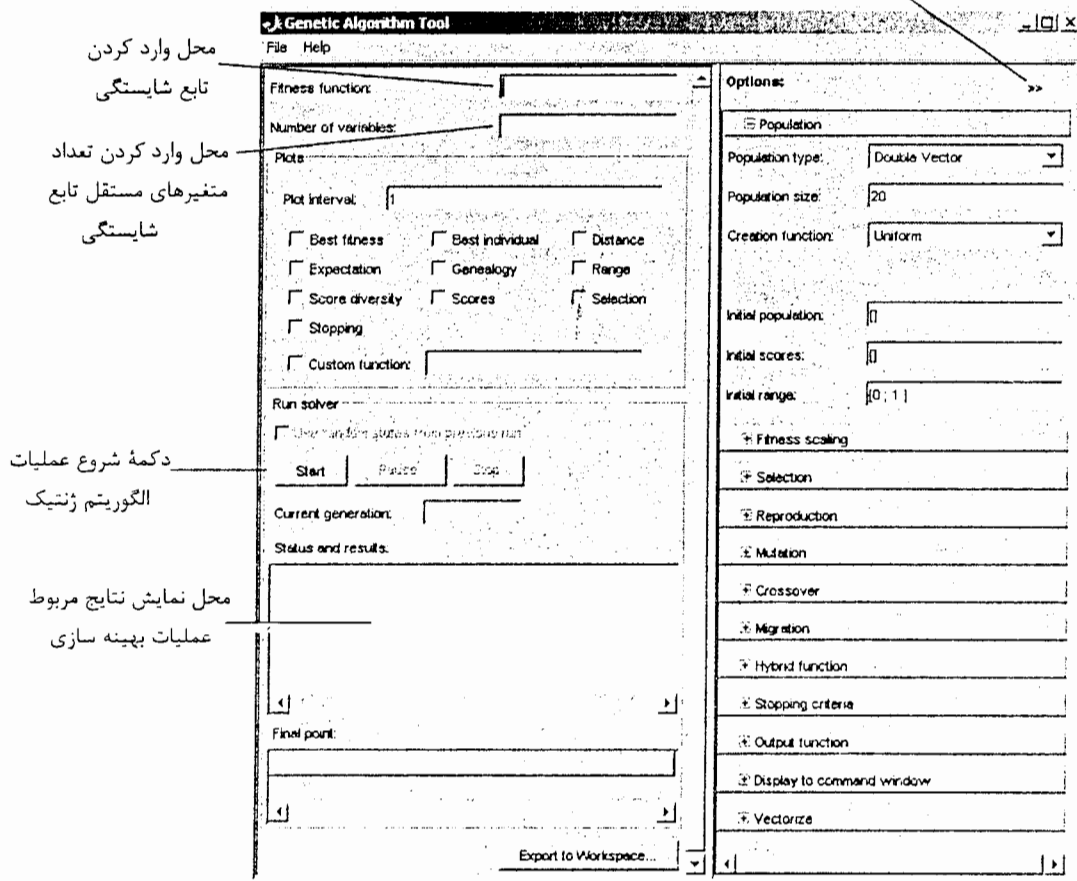
- fval مقدار نهایی تابع شایستگی
- x نقطه ای که در آن مقدار نهایی تابع شایستگی حاصل می شود.

استفاده از تابع الگوریتم ژنتیک GA در جایی مناسب است که بخواهیم :

- نتایج را مستقیماً به محیط کاری<sup>۱</sup> نرم افزار MATLAB برگشت دهیم.
- الگوریتم ژنتیک را به تعداد دفعات زیاد و با مقادیر مختلف با استفاده از فراخوانی تابع ga از یک M-file اجرا کنیم.

<sup>۱</sup> workspace

با کلیک در این قسمت توضیحات مربوط به گزینه ها نشان داده می شود.



شکل ۴-۱ ابزار الگوریتم ژنتیک

#### ۴-۲-۲ استفاده از ابزار الگوریتم ژنتیک

ابزار الگوریتم ژنتیک<sup>۱</sup>، یک رابط کاربری گرافیکی<sup>۲</sup> می باشد که به کاربر این امکان را می دهد که بدون کار با خط دستور الگوریتم ژنتیک را مورد استفاده قرار دهد. برای اجرای ابزار الگوریتم ژنتیک لازم است که در خط دستور عبارت زیر را تایپ کنیم :

gatool

<sup>1</sup> Genetic Algorithm Tool

<sup>2</sup> graphical user interface

اجرای این دستور باعث می شود ابزار الگوریتم ژنتیک مانند آن چه در شکل ۴-۱ نمایش داده شده است. ظاهر گردد. برای استفاده از ابزار الگوریتم ژنتیک ابتدا باید برخی از اطلاعات را بصورت داده های ورودی مشخص کنیم. که در ذیل به آن ها اشاره می کنیم :

- تابع شایستگی همان تابع هدفی است که می خواهیم مینیمم کنیم. تابع شایستگی را بصورت عبارت `@fitnessfun` در قسمت مربوطه در ابزار الگوریتم ژنتیک وارد می کنیم و `fitnessfun.m` یک کد برنامه نویسی M-file می باشد که کار آن محاسبه تابع شایستگی می باشد. نوشتن یک کد برنامه نویسی M-file برای تابعی که ما می خواهیم مینیمم کنیم در صفحات بعدی توضیح داده شده است.

- تعداد متغیرها نشان دهنده طول بردار ورودی مربوط به تابع شایستگی می باشد. برای تابع `my_fun` که در قسمت «نوشتن یک M-file برای تابعی که می خواهیم مینیمم کنیم» آورده شده است باید عدد ۲ را وارد کنیم و برای اجرای الگوریتم ژنتیک دکمه شروع<sup>۱</sup> را فشار می دهیم. ابزار نتایج حاصل از بهینه سازی را در مربع مربوط به وضعیت و نتایج<sup>۲</sup> نشان داده می شود. برای ایجاد تغییرات در گزینه های مربوط به الگوریتم ژنتیک می توان از قسمت گزینه ها<sup>۳</sup> استفاده کرد. برای دیدن زیرمجموعه های مربوط به قسمت گزینه ها باید بر روی علامت + کنار آن کلیک کنیم تا زیر مجموعه های آن نشان داده شود.

<sup>۱</sup> Start

<sup>۲</sup> Status and Results

<sup>۳</sup> Options

## ۴-۳ مثال: تابع Rastrigin

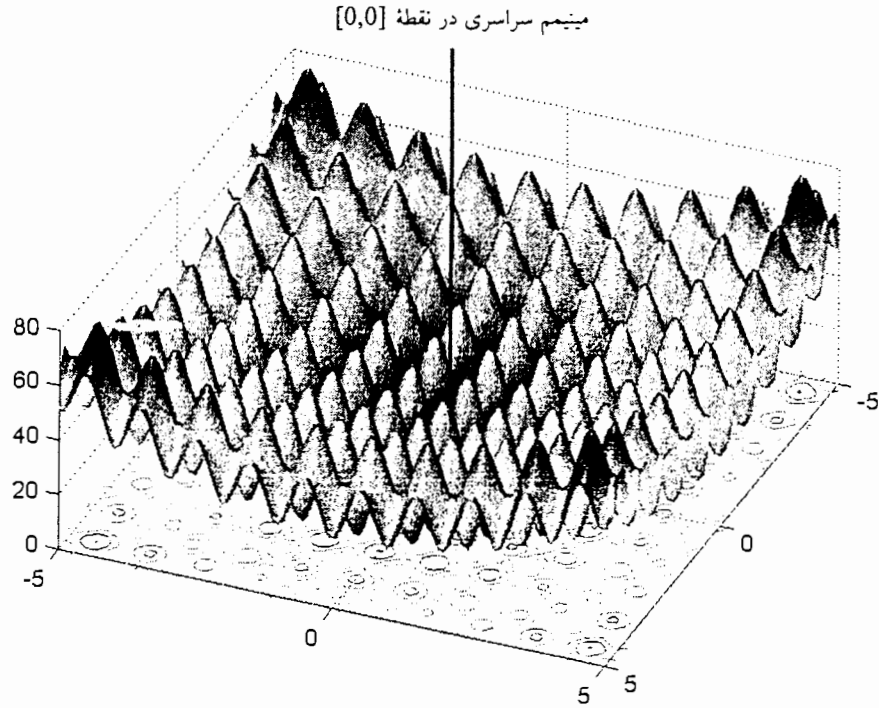
در این قسمت مثالی مورد بررسی قرار می گیرد که نشان می دهد که چگونه می توان مقدار مینیمم یک تابع مانند Rastrigin را که اغلب به عنوان یک تابع آزمایشی برای بررسی عملکرد الگوریتم ژنتیک مورد استفاده قرار می گیرد را بدست آورد.

## ۴-۳-۱ تابع Rastrigin

برای دو متغیر مستقل، تابع Rastrigin بصورت زیر تعریف شده است :

$$Ras(x) = 20 + x_1^2 + x_2^2 - 10(\cos 2\pi x_1 + \cos 2\pi x_2)$$

که در جعبه ابزار به صورت یک M-file با نام rastriginsfcn.m نشان داده می شود که به محاسبه مقدار این تابع می پردازد. شکل ۴-۲ ترسیمه ای از این تابع را نشان می دهد. همانطور که شکل نشان می دهد، تابع Rastrigin دارای مینیمم های محلی بسیار زیادی می باشد که در شکل بصورت فرورفتگی ها نشان داده شده است. با این حال، این تابع تنها دارای یک مقدار مینیمم عمومی می باشد که در نقطه  $[0,0]$  در صفحه  $x-y$  قرار دارد و همانطور که در شکل نشان داده شده است مقدار تابع در این نقطه برابر صفر می باشد و در سایر نقاط مینیمم محلی مقدار تابع بزرگتر از این مقدار می باشد و هرچه فاصله نقاط مینیمم محلی از نقطه مینیمم سراسری بیشتر باشد مقدار تابع نیز بیشتر می شود. تابع Rastrigin معمولاً به صورت یک تابع آزمایشی جهت بررسی الگوریتم ژنتیک به کار می رود. زیرا دلیل داشتن نقاط مینیمم محلی بسیار زیاد پیدا کردن نقطه مینیمم تابع با استفاده از روش های استاندارد معمولاً امکان پذیر نمی باشد. پستی و بلندیهای ترسیمه این تابع ماکزیمم و مینیمم های متناوب را نشان می دهد.



شکل ۴-۲ ترسیم تابع Rastrigin

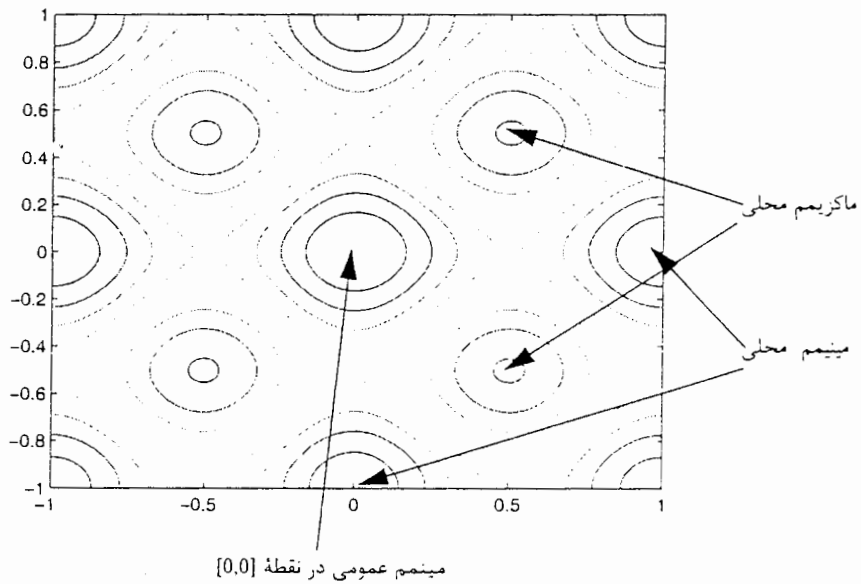
### ۴-۳-۲ پیدا کردن مینیمم تابع Rastrigin

در این قسمت نشان داده می شود که چگونه می توان با استفاده از ابزار الگوریتم ژنتیک به محاسبه مینیمم تابع Rastrigin پرداخت. باید توجه کرد که از آن جاییکه الگوریتم ژنتیک برای انجام جستجو از مقادیر تصادفی استفاده می کند بنابراین مقادیر جوابها ممکن است در اجراهای مختلف برنامه با همدیگر تفاوت های اندکی داشته باشد.

برای پیدا کردن مقدار مینیمم تابع لازم است بترتیب زیر عمل کنیم:

۱. برای اجرای ابزار الگوریتم ژنتیک در خط دستور عبارت gatool را تایپ می کنیم.
۲. مقادیر زیر را در ابزار الگوریتم ژنتیک وارد می کنیم :

- در قسمت تابع شایستگی عبارت `@rastriginsfcn` را وارد می کنیم.
- در قسمت تعداد متغیرها عدد ۲ را وارد می کنیم که تعداد متغیرهای مستقل مربوط به تابع Rastrigin می باشد.

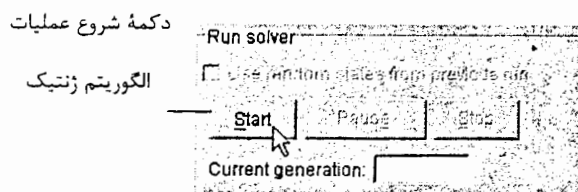


شکل ۳-۴ ترسیمه تابع Rastrigin

مقدار تابع شایستگی و تعداد متغیرها همانطور که در شکل زیر نشان داده شده است در ابزار الگوریتم ژنتیک وارد می گردند.

Fitness function:	@rastriginsfcn
Number of variables:	2

۳. دکمه شروع را در قسمت اجرای تحلیلگر همان طور که در شکل دیده می شود فشار می دهیم.



در زمانی که الگوریتم ژنتیک در حال اجرا شدن می باشد در قسمت نسل جاری<sup>۱</sup> شماره نسل در حال اجرا نشان داده می شود. با فشردن دکمه توقف<sup>۲</sup> الگوریتم به صورت موقتی متوقف می شود و این دکمه تبدیل به دکمه شروع مجدد<sup>۳</sup> می شود که با فشردن این دکمه الگوریتم از همان نقطه ای که در آن متوقف شده است شروع به ادامه می کند. هنگامی که الگوریتم پایان پیدا می کند در قسمت حالت و نتایج اطلاعات خروجی مانند آنچه در ذیل نشان داده شده است، نمایش داده می شود.

Status and results:	
GA running.	
GA terminated.	
Fitness function value: 0.0067749206244585025	
Optimization terminated:	
maximum number of generations exceeded.	
Final point	
1	2
0.00274	-0.00516

در قسمت حالت و نتایج اطلاعات زیر نمایش داده می شود :

- مقدار نهایی تابع شایستگی هنگامی که الگوریتم پایان می یابد :

Function value: 0.0067749206244585025

با دقت در مقدار نشان داده شده در این قسمت درمی یابیم که این عدد به مقدار مینیمم واقعی

تابع Rastrigin بسیار نزدیک می باشد.

- دلیل پایان یافتن عملیات.

Exit: Optimization terminated:

maximum number of generations exceeded.

<sup>1</sup> Current generation

<sup>2</sup> Pause

<sup>3</sup> resume



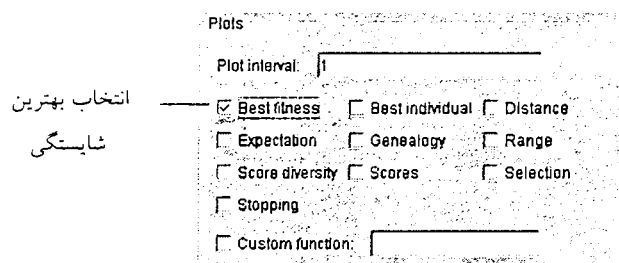
در این مثال، الگوریتم ژنتیک بعد از ۱۰۰ نسل متوقف می شود این عدد برابر مقدار پیش فرض گزینه نسل ها<sup>۱</sup> می باشد که مشخص کننده ماکزیمم تعداد نسل های است که الگوریتم مورد محاسبه قرار می دهد.

- که نقطه نهایی، در این مثال برابر عدد زیر است :

[0.00274 , -0.00516]

#### ۴-۴ گراف های ترسیم شده

در قسمت ترسیم این امکان فراهم شده است که منحنی های مختلف که اطلاعات مربوط به الگوریتم ژنتیک را در هنگام اجرای آن در اختیار ما قرار می دهد مورد مشاهده قرار گیرند. این اطلاعات به ما کمک می کند که با تغییر دادن گزینه های الگوریتم ژنتیک عملکرد آن را بهبود دهیم. بطور مثال، برای ترسیم منحنی بهترین مقدار و مقدار مینیمم تابع شایستگی در هر نسل لازم است که مربع کناری عبارت بهترین مقدار شایستگی<sup>۲</sup> را انتخاب کنیم. این مطلب در شکل ۴-۳ نشان داده شده است.

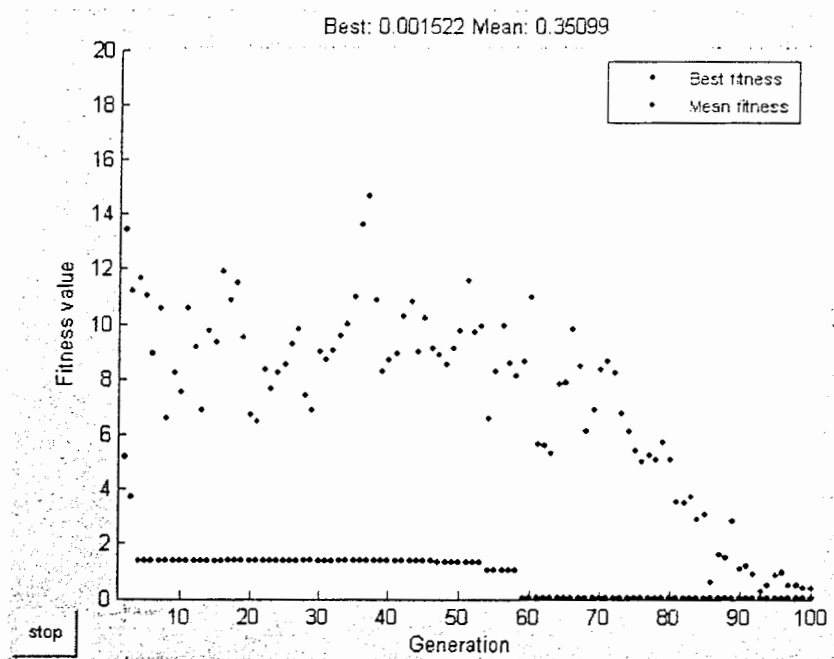


<sup>1</sup> Generations

<sup>2</sup> Best Fitness Value

هنگامی که دکمه شروع فشرده می شود، ابزار الگوریتم ژنتیک ترسیمه ای از مقادیر مینیمم و بهترین مقادیر تابع شایستگی در هر نسل برای ما ترسیم می کند. هنگامی که الگوریتم متوقف می شود نموداری مطابق شکل ۳-۴ نمایش داده می شود.

نقاط موجود در قسمت پائین نمودار مشخص کننده مقادیر بهترین شایستگی هستند در حالیکه نقاط بالایی نمودار مشخص کننده مقادیر متوسط تابع شایستگی در هر نسل می باشند. همچنین نمودار مقادیر متوسط و بهترین مقدار تابع شایستگی مربوط به نسل جاری را به صورت عددی در بالای نمودار نشان می دهد. به منظور داشتن یک نمایش بهتر از کاهش مقدار تابع شایستگی می توان مقیاس محور y را به مقیاس لگاریتمی تبدیل کرد.



شکل ۳-۴ روند تغییرات شایستگی بر اساس نسلها

برای این منظور

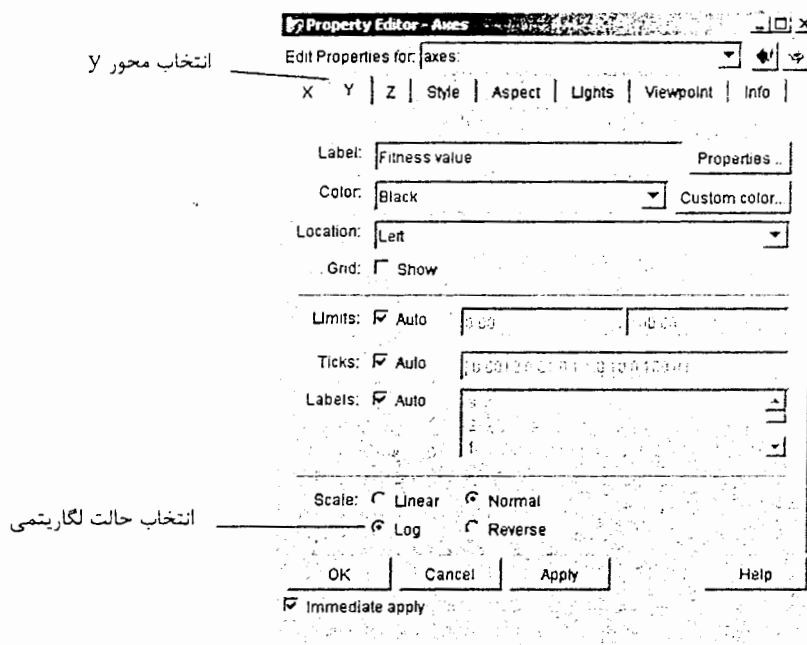
۱. از ابزار Edit در پنجره ترسیم برای بازکردن ویرایشگر خصوصیات Axes Properties را

مانند آنچه در شکل نشان داده شده است عمل می‌کنیم.

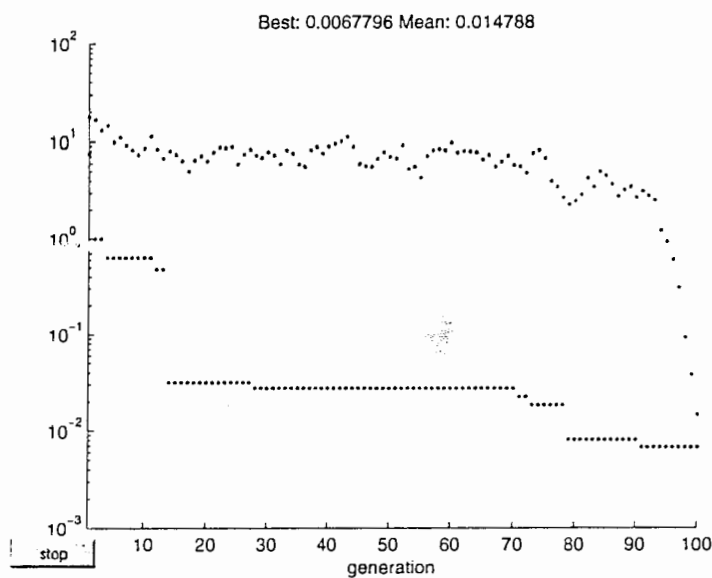
۲. کلید  $y$  را انتخاب می‌کنیم.

۳. در مربع مربوط به Scale گزینه Log را انتخاب می‌کنیم.

در این حالت شکل نمودار بصورت شکل ۴-۴ نمایش داده می‌شود.



به طور نمونه بهترین مقدار تابع شایستگی در نسلهای اولیه به طور بسیار سریعی بهبود می‌یابد هنگامی که اعضا از جواب بهینه دورتر هستند و در جمعیت هایی که به جواب بهینه نزدیکتر هستند بهبود بهترین مقدار تابع شایستگی بسیار آرامتر می‌باشد.



شکل ۴-۴ روند تغییرات شایستگی بر اساس نسلها  
در حالت لگاریتمی

#### ۴-۵ برخی از اصطلاحات الگوریتم ژنتیک

این بخش به معرفی برخی از اصطلاحات بکار رفته در الگوریتم ژنتیک شامل موارد زیر می پردازد :

- تابع شایستگی<sup>۱</sup>
- اعضا<sup>۲</sup>
- جمعیت ها و نسل ها<sup>۳</sup>
- مقادیر شایستگی و بهترین مقدار شایستگی
- والدین و فرزندان<sup>۴</sup>

<sup>۱</sup> Fitness Functions

<sup>۲</sup> Individuals

<sup>۳</sup> Populations and Generations

<sup>۴</sup> Parents and Children

#### ۴-۵-۱ تابع شایستگی

تابع شایستگی تابعی است که می‌خواهیم مینیمم کنیم و این تابع در روش های بهینه سازی استاندارد به نام تابع هدف<sup>۱</sup> معرفی می‌شود. ابزار الگوریتم ژنتیک در جهت پیدا کردن مینیمم تابع شایستگی گام برمی‌دارد. تابع شایستگی را می‌توان به صورت یک M-file نوشت و آن را بعنوان یک عبارت ورودی برای تابع اصلی الگوریتم ژنتیک در نظر گرفت.

#### ۴-۵-۲ اعضا

یک عضو شامل هر نقطه ای است که می‌توانیم تابع شایستگی را در مورد آن به کار بگیریم. مقدار تابع شایستگی برای یک عضو امتیاز<sup>۲</sup> آن نامیده می‌شود. برای مثال، اگر تابع شایستگی بصورت زیر تعریف شده باشد :

$$f(x_1, x_2, x_3) = (2x_1 + 1)^2 + (3x_2 + 4)^2 + (x_3 - 2)^2$$

بردار (1, -3, 2) که طول آن برابر تعداد متغیرهای مسئله می‌باشد، یک عضو می‌باشد که امتیاز آن بصورت  $f(2, -3, 1) = 51$  می‌باشد.

هر عضو معمولاً به صورت یک genome و هر بردار شامل اعضا نیز به عنوان genes تعبیر می‌شود.

<sup>1</sup> objective function

<sup>2</sup> Score

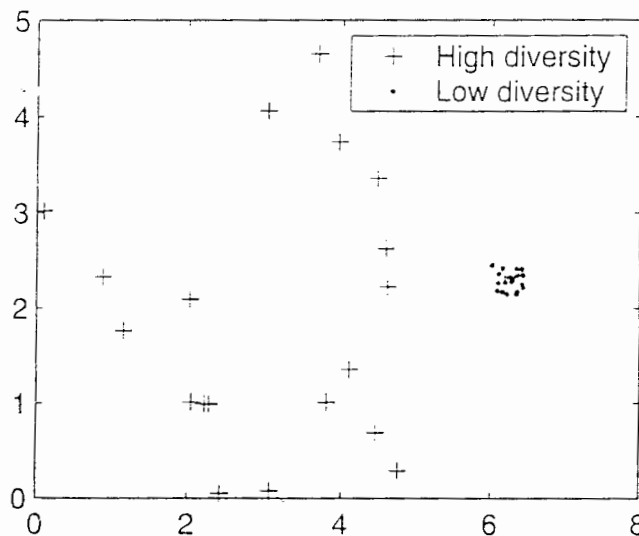
### ۴-۵-۳ جمعیت ها و نسل ها

یک جمعیت آرایه ای از اعضا می باشد. برای مثال، اگر سایز جمعیت برابر ۱۰۰ باشد و تعداد متغیرهای تابع شایستگی برابر ۳ باشد اعضای جمعیت به صورت یک ماتریس 100-by-3 معرفی می شود. یک عضو مشابه ممکن است بیشتر از یک بار در جمعیت حضور پیدا کند. برای مثال، عضو (1, 3, 2) می تواند بیشتر از یک سطر آرایه وجود داشته باشد.

در هر فرآیند تکرار، الگوریتم ژنتیک مجموعه ای از محاسبات را بر روی جمعیت موجود برای ایجاد یک جمعیت جدید انجام می دهد. هر کدام از این جمعیت های متوالی یک نسل نامیده می شود.

### ۴-۵-۴ پراکندگی<sup>۱</sup>

پراکندگی برابر میانگین فاصله بین اعضای یک جمعیت می باشد. یک جمعیت دارای پراکندگی بالایی می باشد اگر میانگین فاصله ها بزرگ باشد. در شکل ۴-۵ جمعیت سمت چپ دارای پراکندگی بالایی می باشد در حالیکه جمعیت سمت راست دارای پراکندگی پائینی می باشد. پراکندگی برای



شکل ۴-۵ پراکندگی اعضای جمعیت

<sup>۱</sup> Diversity

الگوریتم ژنتیک یک امر ضروری و لازم می باشد و باعث می شود که الگوریتم ژنتیک فضای بزرگتری را مورد کاوش قرار دهد.

#### ۴-۵-۵ مقادیر شایستگی و بهترین مقادیر شایستگی

مقدار شایستگی یک عضو جمعیت برابر مقدار تابع شایستگی برای آن عضو می باشد. ابزار الگوریتم ژنتیک مقدار مینیمم تابع شایستگی را پیدا می کند و بهترین مقدار شایستگی برای یک جمعیت برابر کمترین مقدار شایستگی برای هر عضو در آن جمعیت می باشد. لازم بذکر است که در تئوری های مربوط به الگوریتم ژنتیک معمولاً بهترین مقدار شایستگی برای یک جمعیت برابر بیشترین مقدار شایستگی برای آن عضو در آن جمعیت می باشد. که در برنامه نویسی های انجام شده در این رساله ما به شیوه مرسوم در تئوری های الگوریتم ژنتیک عمل کرده ایم.

#### ۴-۵-۶ والدین و فرزندان

برای ایجاد نسل بعدی، الگوریتم ژنتیک اعضای مشخصی را از نسل جاری انتخاب می کند که والدین نامیده می شود و از آن ها در تولید اعضای نسل بعدی که فرزندان نامیده می شود استفاده می کند. به طور مشخص الگوریتم ژنتیک بیشتر به انتخاب والدینی که دارای شایستگی بیشتری هستند تمایل دارد.

## ۴-۶ نمای کلی الگوریتم ژنتیک در Matlab V7.0

در این قسمت به ارائه خلاصه ای از عملکرد الگوریتم ژنتیک می پردازیم:

۱. الگوریتم با ایجاد جمعیت اولیه تصادفی آغاز می گردد.
۲. الگوریتم به ایجاد رشته ای از جمعیت ها یا نسل ها می پردازد. در هر گام، الگوریتم برای ایجاد نسل بعدی، اعضای نسل جاری را به کار می گیرد. به منظور ایجاد نسل جدید، الگوریتم گام های زیر را طی می کند:
  - I. ابتدا با محاسبه مقدار شایستگی اعضای جمعیت جاری امتیاز هر کدام از اعضا را مشخص می کند.
  - II. با مقیاس بندی مقادیر خام تابع شایستگی امکان تبدیل آن ها را به یک محدوده مقادیر قابل استفاده تر فراهم می کند.
  - III. والدین را بر اساس مقدار شایستگی آن ها انتخاب می کند.
  - IV. فرزندان را با استفاده از والدین ایجاد می کند. فرزندان با استفاده از ایجاد تغییرات تصادفی در یکی از والدین منفرد تحت عنوان جهش<sup>۱</sup> یا با ایجاد پیوند<sup>۲</sup> بین یک جفت از والدین انجام می پذیرد.
  - V. جایگزینی جمعیت جاری با فرزندان تولید شده از آن ها برای ایجاد نسل بعدی.
۳. الگوریتم در جاییکه با یکی از معیارهای توقف<sup>۳</sup> برخورد کند متوقف می شود.

### ۴-۶-۱ جمعیت اولیه

الگوریتم ژنتیک با ایجاد یک نسل اولیه تصادفی همانند آنچه در شکل ۴-۶ نشان داده شده است.

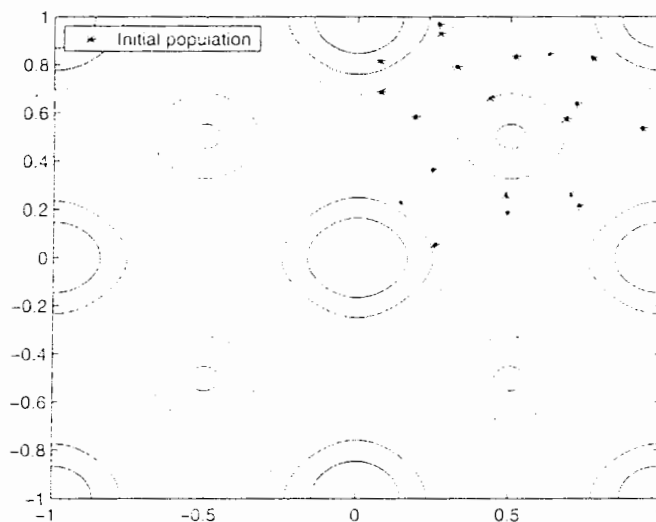
<sup>1</sup> Mutation  
<sup>2</sup> Crossover  
<sup>3</sup> stopping criteria



آغاز می‌گردد. در این مثال، جمعیت اولیه شامل ۲۰ عضو می‌باشد که این عدد برابر مقدار پیش فرض اندازه جمعیت<sup>۱</sup> در گزینه جمعیت ابزار الگوریتم ژنتیک می‌باشد. توجه کنید که تمام اعضای جمعیت اولیه در ربع اول دستگام مختصاتی قرار می‌گیرند و مختصات آن‌ها بین ۰ و ۱ می‌باشد که این امر به این دلیل است که در قسمت دامنه اولیه<sup>۲</sup> در گزینه جمعیت مقدار [0;1] وارد شده است.

اگر ما بدانیم که نقطه مینیمم تابع به صورت تقریبی در کدام ناحیه قرار می‌گیرد باید مقدار دامنه

اولیه در گزینه جمعیت طوری تنظیم شود که این مقدار در نقطه میانی این ناحیه قرار بگیرد. برای



شکل ۴-۶ پراکندگی جمعیت اولیه

مثال، در مورد تابع Rastrigin اگر تصور می‌شود نقطه مینیمم حول و حوش [0 0] می‌باشد باید دامنه اولیه را برابر [-1;1] در نظر گرفت. با این حال همچنان که مثال‌ها نشان می‌دهد، حتی اگر مقادیر کمتری برای دامنه اولیه در نظر گرفته شود، الگوریتم ژنتیک قادر به پیدا کردن مقدار مینیمم می‌باشد.

<sup>1</sup> Population size  
<sup>2</sup> initial population

#### ۴-۶-۲ ایجاد نسل بعدی

در هرگام، الگوریتم ژنتیک برای ایجاد فرزندان و تکمیل نسل بعدی جمعیت جاری را مورد استفاده قرار می دهد. الگوریتم گروهی از اعضای جمعیت جاری را که والدین نامیده می شوند انتخاب می کند که این اعضا ژن های<sup>۱</sup> خود برای ایجاد فرزندان در اختیار آن ها قرار می دهند.

الگوریتم معمولاً اعضای را که دارای شایستگی بیشتری هستند بعنوان والدین انتخاب می کند. در ابزار الگوریتم ژنتیک کاربر می تواند تابعی را که الگوریتم با استفاده از آن به انتخاب والدین می پردازد را در قسمت تابع انتخاب<sup>۲</sup> در گزینه Selection انتخاب کند و یا به ایجاد تابع انتخاب دلخواه پردازد.

الگوریتم ژنتیک سه نوع فرزند برای نسل بعدی ایجاد می کند :

- فرزندان نخبه<sup>۳</sup> که اعضای از نسل جاری هستند که بهترین مقدار تابع شایستگی را دارا هستند. این اعضا به طور خودکار در نسل بعدی حضور پیدا می کنند.
- فرزندان پیوند یافته<sup>۴</sup> که از پیوند یافتن بین رشته های والدین بوجود می یابند.
- فرزندان جهش یافته<sup>۵</sup> که از ایجاد تغییرات تصادفی در یکی از والدین منفرد ایجاد می گردد.

#### ۴-۶-۲-۱ فرزندان پیوند یافته

الگوریتم ژنتیک با استفاده از ترکیب یک جفت از والدین جمعیت جاری به ایجاد فرزندان پیوند یافته می پردازد. در هر مرتبه بردار فرزند، تابع پیوند پیش فرض یک جز یا ژن را به طور همزمان از یک جفت والدین انتخاب می کند و آن ها را به فرزندان نسبت می دهد.

<sup>۱</sup> genes

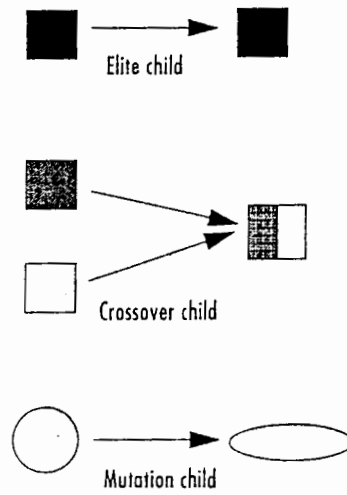
<sup>۲</sup> Selection function

<sup>۳</sup> Elite children

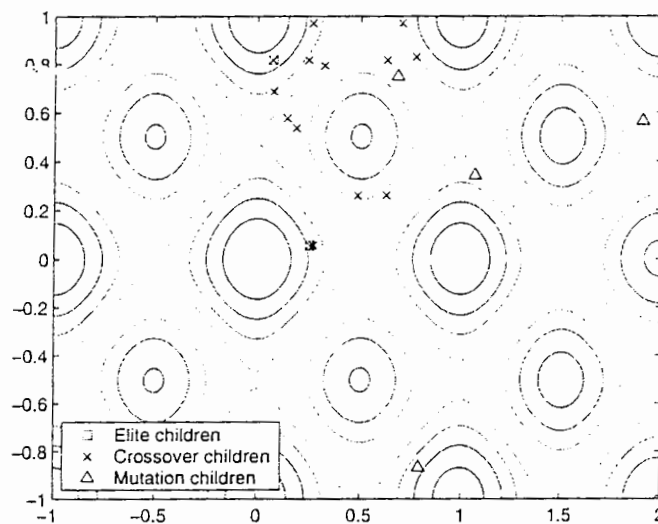
<sup>۴</sup> Crossover children

<sup>۵</sup> Mutation children

دیاگرام های زیر شماتیکی از این سه نوع فرزند را نشان می دهد:



شکل ۴-۷ سه نوع فرزند تولید شده در الگوریتم ژنتیک



شکل ۴-۸ پراکندگی فرزندان تولید شده در الگوریتم ژنتیک

#### ۴-۶-۲-۲ فرزندان جهش یافته

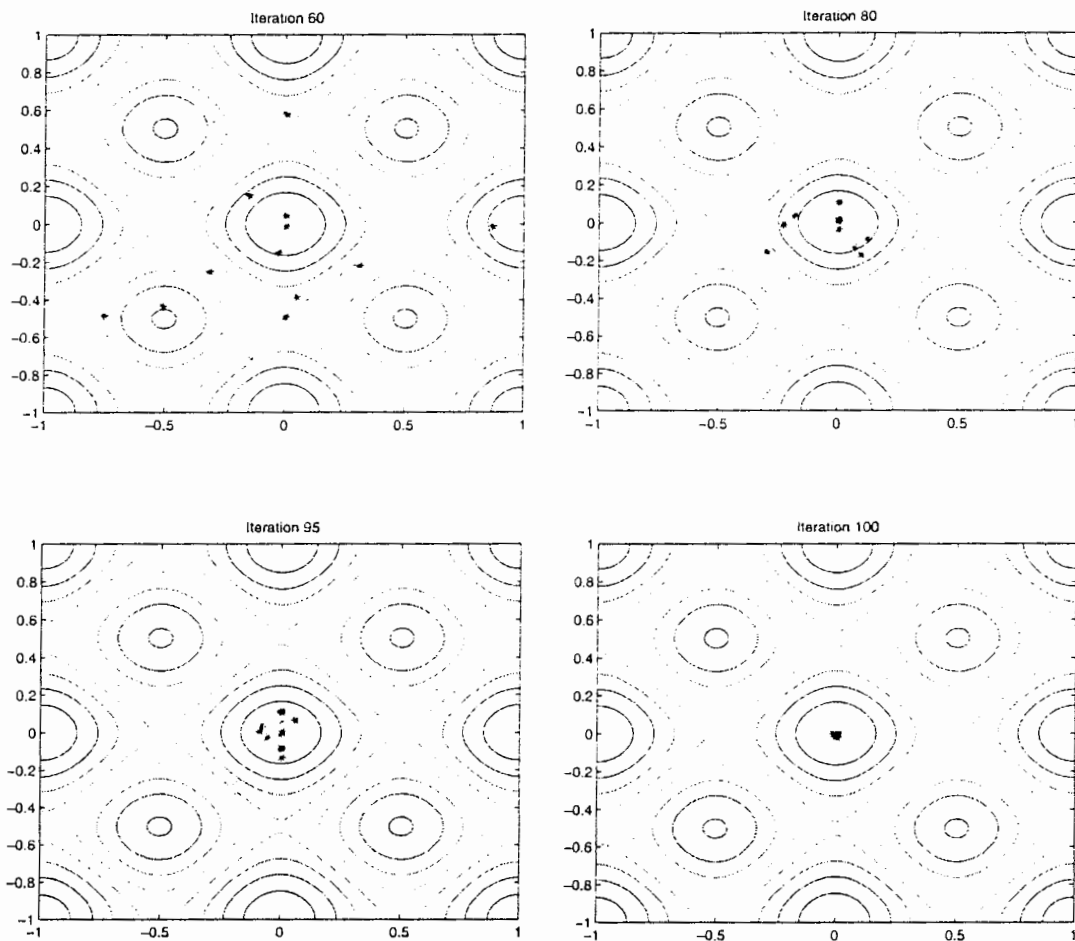
الگوریتم ژنتیک با استفاده از ایجاد تغییرات تصادفی در ژن های والدین به ایجاد فرزندان جهش

یافته می پردازد. شکل ۴-۸ فرزندان بوجود آمده از نسل اولیه را نشان می دهد که جمعیت نسل دوم ر

تشکیل می دهند و مشخص می کند که کدام یک از آن ها نخبه، پیوند یافته و یا جهش یافته می باشد.

#### ۷-۴ شکل نسل های بعدی

شکل ۴-۹ جمعیت های مربوط به الگوریتم را در تکرارهای ۶۰، ۸۰، ۹۵ و ۱۰۰ نشان می دهد. هرچه تعداد نسلها بیشتر می شود اعضای جمعیت به همدیگر نزدیکتر می شوند و به نقطه مینیمم [0 0] دست پیدا می کند.



شکل ۴-۹ پراکندگی فرزندان تولید شده در الگوریتم ژنتیک مربوط به نسلهای ۶۰، ۸۰، ۹۵، ۱۰۰

## ۸-۴ حالت های توقف الگوریتم ژنتیک

الگوریتم ژنتیک با استفاده از پنج شرط مشخص می کند که چه زمانی الگوریتم متوقف شود:

- **نسلها<sup>۱</sup>** - در الگوریتم هنگامی که تعداد نسلها به مقدار نسل های مشخص شده در جعبه ابزار الگوریتم ژنتیک برسد الگوریتم به صورت خودکار متوقف می شود.
- **حد زمانی<sup>۲</sup>** - در الگوریتم ژنتیک هنگامی که از زمان اجرای الگوریتم به میزان حد زمانی مشخص شده در ورودی های الگوریتم ژنتیک بگذرد الگوریتم متوقف می شود.
- **حد شایستگی<sup>۳</sup>** - در الگوریتم ژنتیک هنگامی که مقدار تابع شایستگی برای بهترین نقطه در جمعیت جاری کمتر یا برابر حد شایستگی شود الگوریتم متوقف می شود.
- **توقف نسلها<sup>۴</sup>** - در الگوریتم ژنتیک هنگامی که هیچ بهبودی در تابع هدف در طول نسلهای متوالی به میزان مشخص توقف نسل بوجود نیاید الگوریتم متوقف می شود.
- **توقف حد زمانی<sup>۵</sup>** - در الگوریتم ژنتیک هنگامی که هیچ بهبودی در تابع هدف در طول یک زمان مشخص به اندازه زمان حد توقف بوجود نیاید الگوریتم متوقف می شود.

الگوریتم ژنتیک هنگامی که با هر کدام از پنج حالت شرح داده شده در بالا برخورد کند به طور خودکار متوقف می شود. مقدار هر کدام از این معیارها را در قسمت Stopping criteria options در ابزار الگوریتم ژنتیک می توان مشخص نمود. مقادیر پیش فرض در شکل ۴-۱۰ نشان داده شده است.

هنگامی که الگوریتم ژنتیک اجرا می گردد در قسمت موارد توقف الگوریتم ژنتیک می شود نشان داده می شود. گزینه های Stall time limit و Time limit از طولانی شدن الگوریتم جلوگیری

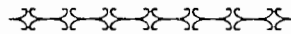
---

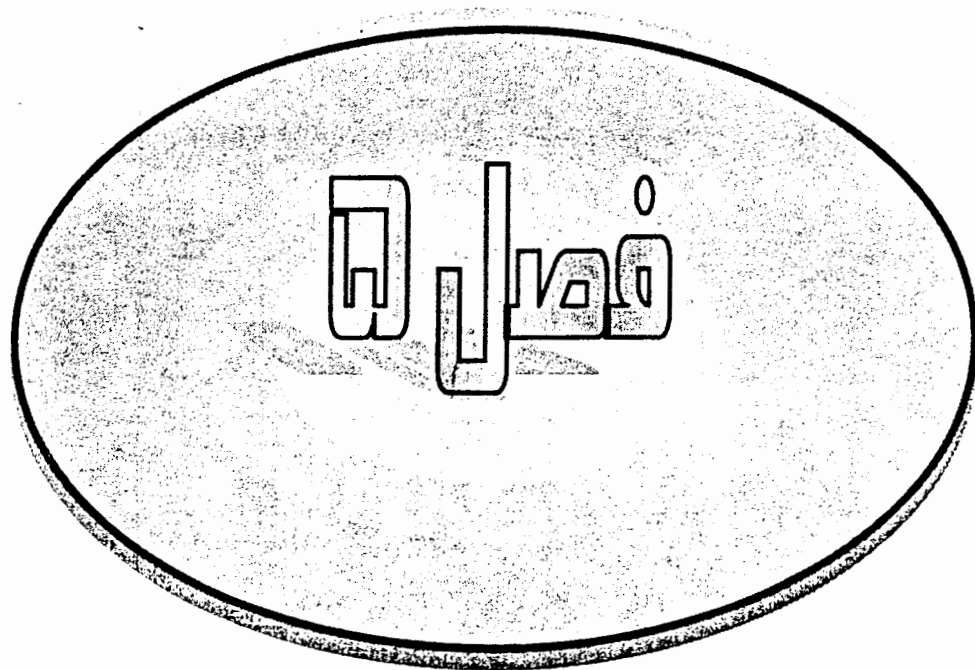
<sup>1</sup> Generations  
<sup>2</sup> Time limit  
<sup>3</sup> Fitness limit  
<sup>4</sup> Stall generations  
<sup>5</sup> Stall time limit

Stopping criteria	
Generations:	100
Time limit:	Inf
Fitness limit:	Inf
Stall generations:	50
Stall time limit:	20

شکل ۴-۱۰ موارد توقف الگوریتم ژنتیک

می کند. هنگامی که الگوریتم در این شرایط متوقف می شود ممکن است با افزایش مقدار این پارامترها بتوان جواب ها را تا حدی بهبود بخشید.





بهینه سازی سطح مقطع، هندسه  
و توپولوژی

از دیر باز توجه انسان بر کاهش هزینه ها و افزایش پس انداز بوده است. این خواسته در مهندسی سازه بصورت کاهش مصرف مصالح و یا کاهش هزینه های اجرایی نمود پیدا می کند. طرح بهینه سیستم های سازه ای از جمله خراباها به طرح بهینه مقاطع<sup>۱</sup>، طرح بهینه شکل<sup>۲</sup>، طرح بهینه توپولوژی<sup>۳</sup> و یا ترکیبی از آنها دسته بندی می شود. در این فصل از رساله ما در قسمت اول به بهینه سازی مقاطع، در قسمت دوم به بهینه سازی مقاطع و هندسه، در قسمت سوم به بهینه سازی مقاطع و توپولوژی و در قسمت آخر به بهینه سازی همزمان مقاطع، هندسه و توپولوژی. سازه های خرابایی می پردازیم و در هرکدام از این قسمتها، ضمن حل مثالهایی در حالت های مختلف به مقایسه نتایج و نمودارهای مربوطه با مراجع مختلف می پردازیم.

همانطور که می دانیم، در مسائل واقعی طراحی سازه معمولاً ما با فضای طراحی بزرگ، تعداد متغیرهای زیاد و قیود متفاوتی دست به گریبان هستیم. استفاده از الگوریتم های جستجوی تصادفی از جمله الگوریتم ژنتیک در حل این مسائل بدلیل عدم کاوش مؤثر فضای طراحی، تولید کرموزوم های مرده به علت وجود قیود متفاوت و افتادن در دام بهینه های محلی و معمولاً کارایی چندانی نداشته و تاکنون کمتر طراحی به حل این گونه مسائل با روش الگوریتم ژنتیک پرداخته است.

<sup>1</sup> Sizing Optimal Design    <sup>2</sup> Shape Optimal Design

<sup>3</sup> Topology Optimal Design



## ۵-۱ بهینه سازی مقاطع سازه های خرپاهایی با استفاده از الگوریتم ژنتیک

در بسیاری از موارد به لحاظ مسایل معماری و یا اجرایی (ساخت) لازم می شود که طرح بهینه مقاطع به ازای توپولوژی ثابت و یا پیکربندی ثابت انجام پذیرد. همچنین معمولاً در مسایل کاربردی مهندسی، مقاطع استاندارد تولید کارخانه ای استفاده می شود. به این ترتیب<sup>۱</sup> در طرح بهینه لازم شود متغیرهای طراحی سطح مقطع از لیست پروفیل‌های موجود<sup>۱</sup> انتخاب شوند، باید بهینه یابی مقاطع منفصل<sup>۲</sup> صورت پذیرد. یکی از روشهای بهینه یابی سازه ها که می تواند متغیرهای طراحی منفصل و یا پیوسته را وارد فرمولبندی خود کند، بهینه یابی مبتنی بر الگوریتم ژنتیک است. این روش هوشمند بطور موفقیت آمیزی طرح بهینه عمومی را بدون در نظر گرفتن فرضیات محدود کننده ای از قبیل پیوسته بودن فضای کاوش و یا وجود مشتقات پیدا می کند.

در بهینه یابی سازه ها بر اساس GA<sup>۳</sup>، بطور متعارف تحلیل سازه جهت کسب نیروهای داخلی به میزان تعداد نسلی ضربدر تعداد کروموزوم های جمعیت (pop-size) لازم می شود. علاوه بر آن کیفیت نتایج به انتخاب تعداد نسل ها، تعداد اعضای جمعیت (pop-size)، طول هر کروموزوم یا طول هریک از زیر رشته های آن (مربوط به هر متغیر طراحی)، نرخ پیوند، نرخ جهش، نوع تابع پنالتی و مقدار ثابتهای آن، تعریف تابع شایستگی و غیره بستگی دارد.

در قسمت اول این فصل بهینه یابی مقاطع خرپاها بر اساس GA بخصوص هنگامیکه متغیرهای طراحی منفصل هستند و سطح مقطع اعضای خرپا بایستی از لیست مقاطع موجود انتخاب شوند، مورد توجه قرار می گیرد.

<sup>۱</sup> Available List

<sup>۲</sup> Discrete Variable Optimization

<sup>۳</sup> GA Based Structural Optimization

در تمام روش های بیان شده در این فصل دو فاکتور کاهش حجم عملیات عددی و افزایش سرعت محاسبات و همگرایی به سمت بهینه عمومی مد نظر بوده است. همچنین در روشهای مطرح شده سعی شده است تا نتایج به بهینه واقعی<sup>۱</sup> نزدیکتر شده و حتی الامکان وابستگی به ضرایب ثابت تابع پنالتهی، نرخ پیوند و نرخ جهش کاهش یابد. در خاتمه هر قسمت مثالهای متداول مطرح شده در مقالات با روش های انجام شده مورد بررسی قرار گرفته و نتایج حاصله با مقالات مقایسه گردیده است.

### ۵-۱-۱ فرمولبندی مساله بهینه یابی مقاطع منفصل خرپاها بر اساس GA

فرمولبندی مساله بهینه یابی مقاطع منفصل سازه های میله ای (خرپاها) با استفاده از GA

بصورت زیر تعریف می شود :

پیدا کنید :

$$\mathbf{X} = [x_1, x_2, \dots, x_{ns}]^T ; x_s \in S ; s = 1, \dots, ns \quad (1-5)$$

تا کمینه شود :

$$\phi(\mathbf{X}) = f(\mathbf{X}) + f_{penalty} \quad (2-5)$$

در معرض قیود زیر :

$$g_i(\mathbf{X}) = \left| \frac{\sigma_i}{\sigma_a} \right| - 1 \leq 0 ; i = 1, \dots, ne \quad (3-5)$$

$$g_j(\mathbf{X}) = \left| \frac{\Delta_j}{\Delta_a} \right| - 1 \leq 0 ; j = 1, \dots, ndof \quad (4-5)$$

$$\begin{cases} g_k(\mathbf{X}) = \left| \frac{\sigma_k}{\sigma_{cr}} \right| - 1 \leq 0 ; k = 1, \dots, mc \\ \sigma_{cr} = \frac{\pi^2 E \alpha_k X_k}{l_k^2} \end{cases} \quad (5-5)$$

در روابط فوق  $\mathbf{X}$  بردار متغیرهای طراحی،  $X_s$  سطح مقطع عرضی گروه  $S$  ام متعلق به لیست پروفیلپای موجود<sup>۲</sup>  $S$ ،  $ns$  تعداد متغیرهای طراحی سطح مقطع به تعداد گروه بندی اعضا<sup>۳</sup>،  $f(\mathbf{X})$  تابع

<sup>۱</sup> True Optimum

<sup>۲</sup> Available Profile List

<sup>۳</sup> Member Linking ( Grouping )

هدف که معمولاً وزن یا حجم سازه است،  $f_{penalty}$  تابع جریمه طرح های غیر قابل قبول<sup>۱</sup> مربوط به نقض قيود پاسخ سازه،  $\phi(\mathbf{X})$  تابع هدف اصلاح شده،  $\sigma_i$  تنش محوری عضو  $i$  ام،  $\sigma_a$  تنش مجاز محوری،  $ne$  تعداد اعضای خرپا،  $\Delta_j$  تغییر مکان درجه آزادی  $j$  ام،  $\Delta_a$  تغییر مکان مجاز،  $ndof$  تعداد درجات آزادی فعال گره های خرپا،  $\sigma_k$  تنش فشاری عضو  $k$  ام،  $X_k$  سطح مقطع عرضی عضو  $k$  ام متعلق به گروه  $S$  ام،  $l_k$  طول عضو  $k$  ام،  $mc$  تعداد اعضای فشاری،  $\alpha_k$  ضریب کمانش وابسته به شکل سطح مقطع عرضی عضو  $k$  ام ( $I_k = \alpha_k \cdot X_k^2$ )، و  $g_i(\mathbf{X})$ ،  $g_j(\mathbf{X})$  و  $g_k(\mathbf{X})$  به ترتیب قيود تنش محوری، تغییر مکان و کمانش اولیه می باشند.

در تمام مثالهای بخش های ۱-۵، ۲-۵ و ۳-۵، تابع هدف وزن سازه می باشد که بصورت زیر تعریف

می شود:

$$f(\mathbf{X}) = \sum_{i=1}^{nc} X_i l_i \rho_i \quad (۶-۵)$$

که در آن  $\rho_i$  وزن مخصوص مصالح عضو  $i$  ام متعلق به گروه  $S$  ام است.

### ۵-۱-۲ رمز گشایی کروموزوم ها در مساله بهینه یابی مقاطع منفصل خرپاها

همانگونه که اشاره شد هر کروموزوم مشتمل بر زیر رشته هایی است که متغیرهای طراحی توسط آنها رمز می شوند. در بهینه یابی مقاطع منفصل خرپاها، متغیرهای طراحی سطح مقطع عرضی اعضای گروه بندی شده می باشند که بایستی از لیست پروفیل های موجود یعنی از مجموعه  $S = \{A_1, A_2, \dots, A_{nos}\}$  انتخاب گردند. اگر طول زیر رشته  $i$  ام یعنی  $i = 1, \dots, ns$  بیانگر تعداد بیتها یا کاراکترهای آن باشد و در مرحله رمز کردن<sup>۲</sup> به آن بطور تصادفی اعداد دودویی اختصاص یابد، در صورت رمزگشایی<sup>۳</sup>  $2^{L_i}$  مقدار میتواند به خود اختصاص دهد که کوچکترین مقدار، عدد صحیح صفر و بزرگترین مقدار عدد صحیح  $2^{L_i} - 1$  می باشد. به این ترتیب برای آنکه بتوان عدد صحیح

<sup>۱</sup>Unacceptable Design

<sup>۲</sup> Encoding

<sup>۳</sup> Decoding

تصادفی زیر رشته  $i$  ام یعنی  $IR_i$  را به شماره مقطع قرار گرفته در لیست پروفیل های  $S$  یعنی  $IS_i$  ارتباط دهیم، از رابطه کلی زیر می‌توان استفاده نمود:

$$\begin{aligned} IS_i &= IND_i + IR_i ; \quad i = 1, 2, \dots, ns \\ 1 &\leq IS_i \leq nos \\ 1 &\leq IND_i \leq nos - 2^{L_i} + 1 \\ 0 &\leq IR_i \leq 2^{L_i} - 1 \end{aligned} \quad (7-5)$$

در رابطه بالا  $ns$  تعداد زیر رشته‌ها برابر با تعداد متغیرهای طراحی (تعداد گروه بندی های اعضا) و  $nos$  تعداد کل مقاطع موجود در لیست  $S$  می‌باشد. شاخص  $IND_i$  مربوط به موقعیت قرار گیری مقطع متغیر طراحی  $i$  ام در لیست پروفیل‌ها می‌باشد. مقدار تصادفی  $IS_i$  اندیسی است که سطح مقطع  $A_{IS_i}$  را برای زیر رشته  $i$  ام در لیست  $S$  شناسایی می‌کند. در صورتیکه همواره  $IND_i = 1$  اختیار شود. مقدار  $nos$  باید به نحوی باشد که بر اساس رابطه  $2^{L_i} = nos$  مقدار صحیحی برای  $L_i$  بدست آید. در غیر این صورت باید  $L_i$  به سمت بالاترین عدد صحیح گرد شده و برای انتخاب سطح مقطع از لیست پروفیل‌ها از تناسب خطی استفاده شود. اینکار ممکن است سبب شود برخی از مقاطع لیست با وزن بیشتری برای متغیرهای طراحی انتخاب شوند و طرح بهینه را دچار اشکال کند. بعلاوه در صورت بزرگ بودن  $nos$  اشکال بزرگتری که ایجاد می‌شود این است که  $L_i$  ها و در نتیجه طول کروموزوم بلند، حجم عملیات عددی زیاد، همگرایی کند و عموماً بهینه محلی حاصل می‌شود. بنابراین باید مکانیزمی اندیشید که  $IND_i$  مناسب برای هر متغیر طراحی تعیین و حتی الامکان طول زیر رشته‌ها،  $L_i$ ، کاهش یابد. به این ترتیب طرح بهینه در محدوده کوچکتری از فضای طراحی<sup>۱</sup> جستجو خواهد شد.

برای رفع این مشکل راجیو و کریشنا مورتی در مقاله خود [34]، روشی دو فازی ارائه کرده‌اند. در فاز یک پس از معرفی رابطه‌ای برای  $L_i$ ، لیست  $S$  به تعداد  $2^{L_i}$  واحد تقسیم بندی می‌شود. سپس با تولید تصادفی  $IND_i$  و  $IR_i$  برای کلیه زیر رشته‌ها، توسط روابطی مقادیر سطح مقطع نظیر در

<sup>۱</sup> Design Space

لیست پروفیل ها شناسایی می شود. با طی مراحل فوق به تعداد pop-size، جمعیت اولیه ای تشکیل و عملیات GA آغاز می شود. اینکار آنقدر ادامه می یابد تا بالاخره در یکی از کروموزوم ها جواب قابل قبول با  $C = 0.0$  بدست آید. مقادیر  $IND_i$  این کروموزوم شناسایی و برای استفاده در عملیات فاز ۲ ذخیره می شود. در فاز ۲ عملیات GA بطور معمول انجام و در صورت عدم بهبود نتایج در ۵ سیکل متوالی، با اتخاذ روشهای ابداعی<sup>۲</sup>، سعی شده است از حصول بهینه محلی ممانعت شود.

با ملاحظه روابط مربوط به قیود رفتاری خرپاها در می یابیم که در صورت نقض قیود تنش محوری و کمانش اولیری، سهم هر عضو را می توان با استفاده از روابط (۳-۵) و (۵-۵) بطور مستقیم تعیین کرد در حالیکه در صورت نقض قید تغییر مکان، رابطه (۴-۵) متاثر از اندرکنش کلیه اعضای خرپا بوده و سهم هر عضو بطور مستقیم قابل ارزیابی نیست.

در رابطه بالا  $\Delta_i$  تغییر مکان درجه آزادی  $i$ ام،  $u_i$  تغییر شکل محوری عضو  $i$ ام خرپا،  $P_i$ ،  $A_i$  و  $E_i$  به ترتیب نیروی محوری، طول، مدول الاستیسیته و سطح مقطع عضو  $i$ ام خرپا می باشد. اگر عضو  $i$ ام متعلق به گروه  $s$ ام باشد،  $A_i$  بصورت متغیر طراحی  $X_s$  در روند بهینه یابی ظاهر می شود. در صورتیکه قید تغییر مکان نقض شود داریم:

$$V_{\Delta_j} = \left| \frac{\Delta_j}{\Delta_a} \right| - 1 \quad ; \quad V_{\Delta_j} > 0 \quad ; \quad j = 1, \dots, \text{ndof} \quad (۸-۵)$$

بالاخره سهم عضو  $i$ ام از نقض کلیه قیود بصورت زیر بیان می شود:

$$V_i = v_{i,\Delta} + v_{i,\sigma} + v_{i,bk} \quad ; \quad i = 1, \dots, ne \quad (۹-۵)$$

$$v_{i,\Delta} = \sum_{j=1}^{\text{ndof}} v_{i,\Delta_j} \quad ; \quad V_{\Delta_j} > 0 \quad (۱۰-۵)$$

$$v_{i,\sigma} = \left| \frac{\sigma_i}{\sigma_a} \right| - 1 > 0 \quad (۱۱-۵)$$

$$v_{i,bk} = \left| \frac{\sigma_k}{\sigma_{cr}} \right| - 1 > 0 \quad (۱۲-۵)$$

در روابط فوق  $V_{i,\Delta}$ ،  $V_{i,\sigma}$  و  $V_{i,bk}$  بترتیب سهم عضو  $i$  ام از نقض قید تغییر مکان درجات آزادی، نقض قید تنش محوری و نقض قید تنش کمانش اوپلری است. برای لحاظ عملکرد کلی عضو در سازه ممکن است مقادیر  $V_{i,\Delta}$  و یا  $V_i$  منفی نیز بدست آید. بعبارت دیگر عضو ممکن است برای بعضی از قیود مناسب و برای بعضی از قیود ضعیف باشد.

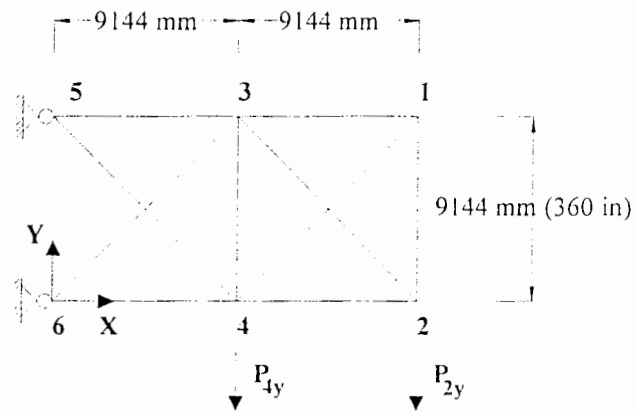
### ۵-۱-۳ مثالهای عددی

در این بخش ضمن بررسی عملکرد روشهای مطرح شده در این رساله با استفاده از تابع هدف ۶-۵ که همان وزن سازه می باشد و استفاده از توابع جریمه ۱-۲ و تابع شایستگی ۲-۴، چند مثال رایج مربوط به حالت های مختلف بهینه سازی مربوط به سطح مقطع خرپاها که در مقالات مطرح شده است مورد بررسی قرار می گیرد و نتایج موجود در مقالات با نتایج بدست آمده با یکدیگر مقایسه می شوند.

### مثال (۵-۱) بهینه یابی مقاطع خرپای مستوی ۱۰ میله ای

در این مثال نیز با استفاده از روش های مطرح شده الگوریتم ژنتیک، حداقل وزن خرپای رایج ۱۰ میله ای همانند شکل (۵-۱) مورد بررسی قرار گرفته است.

در جدول (۵-۱) تعداد متغیرهای طراحی (ns=10)، شماره گره دو انتهای اعضای هر گروه نشان داده شده است. طرح بهینه خرپا بر اساس سطح مقطع های موجود در مجموعه  $S_2$  جدول (۶-۵)، بدست آمده است. در جدول (۵-۲) کلیه اطلاعات طراحی درج شده است. مدول الاستیسیته و وزن مخصوص مصالح همانند مثال قبل در نظر گرفته می شود.



شکل (۱-۵) خرپای مستوی ۱۰ میله ای

در جداول (۳-۵) و (۴-۵) حالت‌های متفاوت بهینه یابی بررسی و نتایج با برخی از مراجع مقایسه شده است. با ملاحظه نتایج مشاهده می‌شود که استفاده از تابع جریمه مطرح شده، وابستگی طرح بهینه به مقادیر ثابت تابع پنالتی و تابع شایستگی را تا حد زیادی مرتفع می‌سازد.

جدول (۱-۵) جزئیات گروه بندی و شماره گذاری نمونه اعضای خرپای مستوی ۱۰ میله ای

شماره گذاری اعضا	شماره گره های دو انتهای اعضای هر گروه	متغیرهای طراحی	شماره گروه
1	3-5	$X_1$	1
2	1-3	$X_2$	2
3	4-6	$X_3$	3
4	2-4	$X_4$	4
9	3-4	$X_5$	5
6	1-2	$X_6$	6
7	4-5	$X_7$	7
8	3-6	$X_8$	8
5	2-3	$X_9$	9
10	1-4	$X_{10}$	10

جدول (۲-۵) اطلاعات طراحی خرابی دو بعدی ۱۰ میله ای

اطلاعات مربوط به قیود		
قیود تغییر مکان در راستای محورهای x و y :		
$ \Delta_j  \leq 50.8 \text{ mm (2 in)}, j = 1,2,3,4$ ; $ \Delta_a  = 50.8 \text{ mm (2 in)}$		
قیود تنش :		
$ \sigma_i  \leq 172.25 \text{ Mpa (25 Ksi)}, i=1,2,\dots,10$ ; $ \sigma_a  = 172.25 \text{ Mpa (25 Ksi)}$		
اطلاعات مربوط به بارگذاری		
شماره نکره	$P_x$	$P_y$
2	0	-445.374 KN (-100 Kips)
4	0	-445.374 KN (-100 Kips)

جدول (۳-۵) مقایسه نتایج طرح بهینه مقاطع خرابی دو بعدی ۱۰ میله ای

متغیرهای طراحی $\ast$ ، $(\text{in}^2) \text{ cm}^2$										وزن N. (lb)	روش
$X_{10}$	$X_9$	$X_8$	$X_7$	$X_6$	$X_5$	$X_4$	$X_3$	$X_2$	$X_1$		
0.645 (0.1)	138.709 (21.5)	138.709 (21.5)	48.387 (7.5)	3.226 (0.5)	0.645 (0.1)	112.903 (17.5)	158.064 (24.5)	0.645 (0.1)	177.42 (27.5)	22728.8 (5103.3)	روش این رسانه
0.645 (0.1)	145.161 (22.5)	135.484 (21.0)	48.387 (7.5)	3.226 (0.5)	0.645 (0.1)	100.0 (15.5)	154.838 (24.0)	0.645 (0.1)	183.871 (28.5)	22601.8 (5074.8)	روش این رسانه
0.645 (0.1)	138.709 (21.5)	138.709 (21.5)	51.613 (8.0)	3.226 (0.5)	0.645 (0.1)	100.0 (15.5)	141.935 (22.0)	0.645 (0.1)	196.774 (30.5)	22601.8 (5074.8)	روش این رسانه
0.645 (0.1)	148.387 (23.0)	135.484 (21.0)	48.387 (7.5)	3.226 (0.5)	0.645 (0.1)	100.0 (15.5)	148.387 (23.0)	0.645 (0.1)	187.096 (29.0)	22634.8 (5082.2)	روش در این رسانه
0.645 (0.1)	138.709 (21.5)	138.709 (21.5)	48.387 (7.5)	3.226 (0.5)	0.645 (0.1)	96.774 (15.0)	151.613 (23.5)	0.645 (0.1)	193.548 (30.0)	22568.4 (5067.3)	روش این رسانه
0.645 (0.1)	132.258 (20.5)	129.032 (20.5)	54.839 (8.5)	0.645 (0.1)	0.645 (0.1)	103.226 (16.0)	151.613 (23.5)	0.645 (0.1)	206.451 (32.0)	22645.3 (5084.6)	راچیو و همکاران [34]



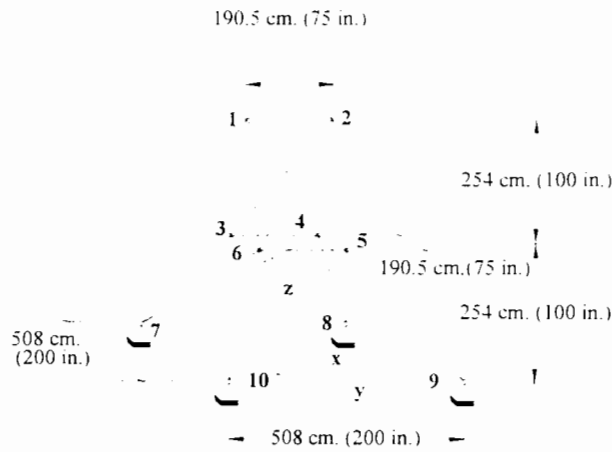
جدول (۴-۵) اطلاعات مربوط به ثابتها و روابط بکار گرفته شده در روش این رساله مربوط به خرابای ۱۰ میله ای

شماره رابطه	شماره رابطه	ثابت تابع	اعضای	تعداد نسلها	احتمال	احتمال	روش
تابع شایستگی	تابع پناهی	پناهی	جمعیت	در هر سیکل	جهش	پیوند	
(۶-۲)	(۲-۲)	15.0	50	200	0.15	0.90	روش این رساله
اریاتور و همکاران [31]	(۳-۲)	2.8	50	200	0.15	0.90	روش این رساله
(۵-۲)	(۳-۲)	2.8	50	200	0.15	0.90	روش این رساله
(۵-۲)	(۱-۲)	2.4	50	50	0.15	0.90	روش این رساله
(۵-۲)	(۱-۲)	3.0	50	200	0.15	0.90	روش این رساله

### مثال ۲-۵ بهینه یابی مقاطع خرابای فضایی ۲۵ میله ای

در این مثال بهینه یابی مقاطع خرابای فضایی ۲۵ میله ای همانند شکل (۱-۳) که در اکثر

مراجع مطرح می شود، برای کسب حداقل وزن مورد بررسی قرار گرفته است.



شکل (۲-۵) خرابای فضایی ۲۵ میله ای

در جدول (۵-۵) گروه بندی ۸ گانه اعضا ( $ns=8$ ) و شماره گذاری های دلخواه اعضای سازه درج شده است. در جدول (۶-۵) مجموعه پروفیل‌های منفصل  $S_1$  با ۳۰ سطح مقطع و  $S_2$  با ۷۹ سطح مقطع مورد استفاده در مثالها نشان داده شده است. در اکثر مقالات جواب بهینه (حداقل وزن) به ازای مجموعه مقاطع  $S_1$  بدست آمده است. جدول (۸-۵) کلیه اطلاعات طراحی خرابی فضایی ۲۵ میله ای را در بر دارد. این مثال اثر کمانش اعضا در قالب رابطه (۹-۵) را در نظر نمی گیرد. در جدول (۸-۵) نتایج طرح بهینه خرابی فضایی ۲۵ میله ای در صورتیکه متغیرهای طراحی از لیست های  $S_1$  و یا  $S_2$  انتخاب شوند، نمایش داده شده است.

جدول (۵-۵) جزئیات گروه بندی و شماره گذاری اعضای خرابی فضایی ۲۵ میله ای

شماره گذاری اعضا	شماره گره های انتهایی اعضای هر گروه	متغیرهای طراحی	شماره گروه
1	1-2	$X_1$	1
2,3,4,5	1-4, 2-3, 1-5, 2-6	$X_2$	2
6,7,8,9	2-5, 2-4, 1-3, 1-6	$X_3$	3
10,11	3-6, 4-5	$X_4$	4
12,13	3-4, 5-6	$X_5$	5
14,15,16,17	3-10, 6-7, 4-9, 5-8	$X_6$	6
18,19,20,21	3-8, 4-7, 6-9, 5-10	$X_7$	7
22,23,24,25	3-7, 4-8, 5-9, 6-10	$X_8$	8

جدول (۵-۶) مجموعه مقاطع منفصل  $S_1$  و  $S_2$

$$S_1 = \{ 0.645 \times I \ (I=1,2,\dots,26), 18.064, 19.355, 20.645, 21.935 \} \text{ (cm}^2\text{)}$$

$$S_1 = \{ 0.1 \times I \ (I=1,2,\dots,26), 2.8, 3.0, 3.2, 3.4 \} \text{ (in}^2\text{)}$$

$$S_2 = \{ 0.645, 3.226 \times I \ (I=1,2,\dots,76), 256.854, 258.08 \} \text{ (cm}^2\text{)}$$

$$S_2 = \{ 0.1, 0.5 \times I \ (I=1,2,\dots,76), 39.81, 40.0 \} \text{ (in}^2\text{)}$$

جدول (۵-۷) اطلاعات طراحی خرابای فضایی ۲۵ میله ای

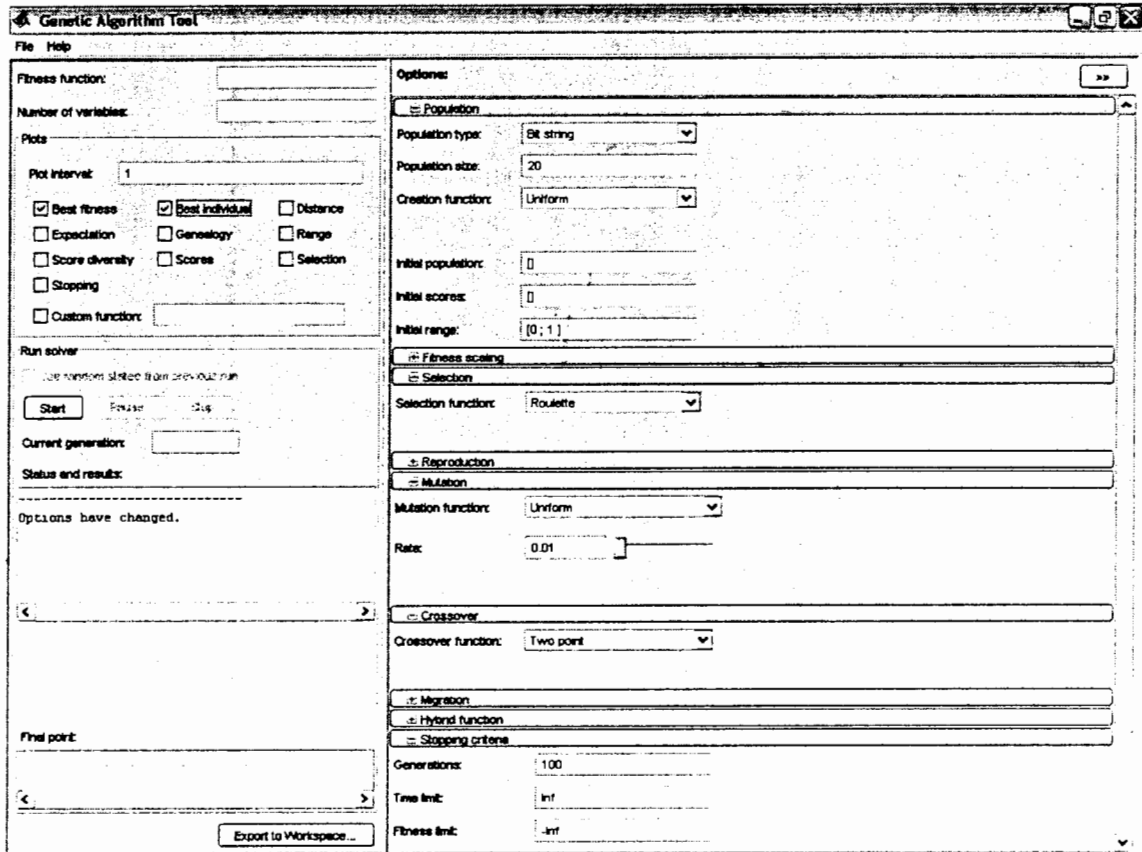
اطلاعات مربوط به قيود <sup>۱</sup>			
قيود تغيير مكان در راستای محوره های X و Y:			
$ \Delta_j  \leq 0.889 \text{ cm. (0.35 in)}, j = 1, 2; \Delta_a = \pm 0.889 \text{ cm. } (\pm 0.35 \text{ in.})$			
قيود تنش:			
$ \sigma_i  \leq 275.6 \text{ Mpa (40 Ksi)}, i = 1, 2, \dots, 25, \sigma_a = \pm 275.6 \text{ Mpa } (\pm 40 \text{ Ksi})$			
اطلاعات مربوط به بارگذاري			
$P_z$ (KN)	$P_y$ (KN)	$P_x$ (KN)	شماره گره
-44.537 (-10 Kips)	-44.537 (-10 Kips)	4.454 (1 Kips)	1
-44.537 (-10 Kips)	-44.537 (-10 Kips)	0	2
0	0	2.227 (0.5 Kips)	3
0	0	2.672 (0.6 Kips)	6
خواص مصالح			
$E = 6.895 \times 10^4 \text{ Mpa } (10^4 \text{ Ksi})$			مدول الاستیسیته:
$\rho = 0.0272 \text{ N/cm}^3 \text{ (0.1 lb/in}^3\text{)}$			وزن مخصوص مصالح:

جدول (۵-۸) مقایسه نتایج طرح بهینه مقاطع خرپای فضایی ۲۵ میله ای

متغیرهای طراحی، $(in^2) cm^2$								وزن، N (lb)	روش
$X_8$	$X_7$	$X_6$	$X_5$	$X_4$	$X_3$	$X_2$	$X_1$		
21.935 (3.4)	6.452 (1.0)	5.806 (0.9)	8.387 (1.3)	2.581 (0.4)	21.935 (3.4)	0.645 (0.1)	2.581 (0.4)	2218.63 (498.15)	روش بین رساله
25.806 (4.0)	0.645 (0.1)	6.452 (1.0)	12.903 (2.0)	0.645 (0.1)	19.355 (3.0)	3.226 (0.5)	3.226 (0.5)	2150.04 (482.75)	روش بین رساله
21.935 (3.4)	3.226 (0.5)	5.806 (0.9)	12.258 (1.9)	0.645 (0.1)	21.935 (3.4)	3.226 (0.5)	0.645 (0.1)	2160.29 (485.05)	روش بین رساله
21.935 (3.4)	4.516 (0.7)	6.452 (1.0)	8.387 (1.3)	0.645 (0.1)	21.935 (3.4)	1.935 (0.3)	0.645 (0.1)	2170.53 (487.35)	روش بین رساله
21.935 (3.4)	5.161 (0.8)	5.806 (0.9)	8.387 (1.3)	0.645 (0.1)	21.935 (3.4)	1.935 (0.3)	0.645 (0.1)	2170.53 (487.35)	روش بین رساله
21.935 (3.4)	2.581 (0.4)	6.452 (1.0)	14.194 (2.2)	0.645 (0.1)	21.935 (3.4)	2.581 (0.4)	0.645 (0.1)	2157.08 (484.35)	روش بین رساله
19.355 (3.0)	11.613 (1.8)	55.161 (0.8)	0.645 (0.1)	1.29 (0.2)	14.839 (2.3)	11.613 (1.8)	0.645 (0.1)	2431.79 (546.01)	راجیو و همکاران [27]
21.935 (3.4)	2.581 (0.4)	5.806 (0.9)	7.097 (1.1)	0.645 (0.1)	20.645 (3.2)	7.742 (1.2)	0.645 (0.1)	2199.26 (493.80)	راجیو و همکاران [31]
25.806 (4.0)	0.645 (0.1)	6.452 (1.0)	12.903 (2.0)	0.645 (0.1)	22.581 (3.5)	0.645 (0.1)	0.645 (0.1)	2138.82 (480.25)	روش بین رساله

جدول (۵-۹) اطلاعات مربوط به ثابتها و شماره روابط مورد استفاده برای خرپای ۲۵ میله ای

انتخاب از لیست مقاطع منفصل	شماره رابطه تابع شایستگی	شماره رابطه تابع پتانسی	ثابت تابع پتانسی	تعداد اعضای جمعیت	تعداد نسل های هر سیکل	احتمال جهش	احتمال پیوند	روش بین رسانه
$S_1$	(۴-۲)	(۱-۲)	10.0	20	200	0.005	0.55	روش بین رسانه
$S_2$	(۴-۲)	(۲-۲)	2.05	50	200	0.15	0.90	روش بین رسانه
$S_1$	(۴-۲)	(۱-۲)	3.4	50	200	0.15	0.90	روش بین رسانه
$S_1$	(۴-۲)	(۱-۲)	3.4	50	50	0.05	0.50	روش بین رسانه
$S_1$	(۴-۲)	(۱-۲)	3.0	50	50	0.15	0.90	روش بین رسانه
$S_1$	(۴-۲)	(۱-۲)	3.6	50	200	0.15	0.90	روش بین رسانه
$S_2$	(۴-۲)	(۱-۲)	2.6	50	200	0.15	0.90	روش بین رسانه



شکل (۳-۵) جمعه ابزار ایجاد شده توسط برنامه نویسی در محیط Matlab V7.0

برای کنترل و انتخاب ورودی ها و خروجی های الگوریتم ژنتیک

### ۴-۱-۵ نتیجه گیری

در این فصل بهینه یابی مقاطع عرضی منفصل خرپاها با استفاده از الگوریتم ژنتیک و روش تغییر مکانها (سختی) مورد بررسی قرار گرفت (روش مورد استفاده در تحلیل سازه مفروض). استفاده از محیط نرم افزار Matlab V7.0 برای برنامه نویسی و ایجاد یک بسته نرم افزاری جهت بهینه سازی سازه ها بدلیل سازگاری این نرم افزار با عملیات ماتریسی بسیار مناسب بوده و بدلیل استفاده از m فایل های جداگانه برای فایل ورودی سازه و فایل ایجاد اپراتورهای مختلف الگوریتم ژنتیک مانند

انواع مختلف عملگرهای پیوند، انتخاب و یا حتی جهش و فایل های خروجی جداگانه مانند انواع مختلف نمودارهای خروجی کارکردن با این مثالها را هرچه بیشتر آسانتر می کند. همچنین ایجاد یک جعبه ابزار مانند آنچه در شکل ۳-۵ نشان داده شده است، انتخاب پارامترهای مختلف الگوریتم ژنتیک و نمودارهای خروجی را برای ما بیش از پیش آسان می کند.

## ۲-۵ بهینه سازی مقاطع و هندسه سازه های خریایی با استفاده از الگوریتم

### ژنتیک

به هنگام طراحی سازه ها مواردی پیش می آید که باید با اتخاذ توپولوژی ثابت، سطح مقطع و هندسه بهینه سازه را تعیین نمود. بدین منظور به سبب زیاد بودن تعداد متغیرهای طراحی مشتمل بر سطح مقطع اعضا و مختصات قابل تغییر گرهی معمولاً با فضای طراحی بسیار بزرگ مواجه هستیم. انتخاب سطح مقطع اعضا از لیست پروفیل های موجود سبب گسستگی فضای طراحی شده و به علاوه قیود حاکم بر تنش اعضا، کمانش و تغییر مکانهای گرهی سبب می شود که امکان در دام بهینه محلی افتادن افزایش یابد. به هنگام بهینه یابی هندسه خریابها با GA در صورتیکه تابع هدف وزن در نظر گرفته شود، ممکن است مواردی پیش آید که با ایجاد تغییرات در هندسه اولیه و بزرگ نمودن فضای طراحی، بهینه محلی حاصل شود حال آنکه اگر هندسه اولیه ثابت در نظر گرفته می شد و فقط بهینه یابی سطح مقطع انجام می پذیرفت، وزن کمتری حاصل می شد. این مساله بدلیل ترکیبات بسیار زیادی است که مقاطع منفصل و هندسه پیوسته سازه می توانند از خود نشان دهند.

در بهینه یابی سازه ها مبتنی بر روش سختی (تغییر مکان) ماتریس سختی هر بار تشکیل و یا احیا می شود و دستگاه معادلات خطی از مرتبه  $n$  (به تعداد درجات آزادی گرهی) حل می گردد. جزئیات این روش تحلیل سازه در پیوست الف بطور کامل توضیح داده شده است.

### ۵-۲-۱ توابع پنالتی، شایستگی

در این رساله برای بهینه یابی مقاطع و هندسه خرپاها از تعریف تابع پنالتی مطابق رابطه (۱-۲) فصل دوم و برای ارزیابی شایستگی کروموزوم ها از رابطه (۲-۴) استفاده شده است.

### ۵-۲-۲ فرآیند پیوند و جهش

همانند قسمت قبل جهت بهینه یابی مقاطع و هندسه خرپاها از فرآیندهای پیوند و جهش جهت ایجاد پراکندگی در محدوده کاوش فضای طراحی استفاده شده است. نوع اپراتور پیوند، دو نقطه ای و در انتخاب نقاط قطع تصادفی در طول رشته کروموزوم هیچ تفاوتی بین متغیرهای طراحی سطح مقطع و متغیرهای طراحی هندسی قائل نشده است. طی بررسی هایی که بر روی مثالهای کاربردی به عمل آمد نرخ پیوند ۰/۹ مناسب تشخیص داده شد.

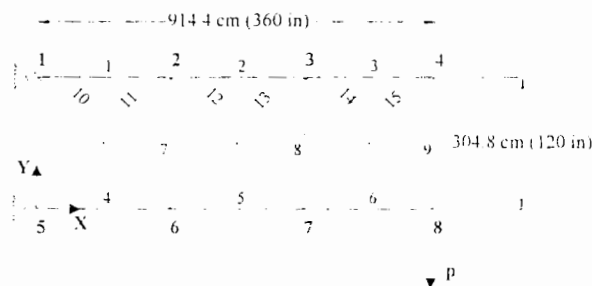
در این رساله با توجه به آنکه در روند بهینه یابی در برخی از مثال ها فضای طراحی بسیار بزرگ و وسیع می باشد لذا بکار گیری مقادیر نرخ جهش بیشتر از حد متعارف باعث بهبود نتایج می شود.

### ۵-۲-۳ بهینه یابی مقاطع منفصل و هندسه پیوسته خرپاها

در روند بهینه یابی، هندسه اولیه را می توان بعنوان یکی از هندسه های متفاوتی که توسط GA تولید می شود در نظر گرفت. لذا یافتن سطح مقطع های نزدیک به بهینه برای هندسه اولیه می تواند



یک نقطه شروع خوب در فضای طراحی محسوب شود. همچنین گاهی شرایطی پیش می‌آید که به سبب ترکیبات بسیار زیاد سطح مقطع اعضا و مختصات قابل تغییر گرهی، به تصور یافتن خرابایی بهینه با هندسه جدید در دام بهینه محلی افتاده ایم حال آنکه اگر همان خرابا تنها برای سطح مقطع بهینه یابی می‌شد، وزن کمتری بدست می‌آمد. بعنوان مثال خرابای ۱۵ میله ای مطرح شده در مرجع [35] مورد بررسی قرار می‌گیرد. شکل (۴-۵) خرابای مورد بحث و جدول (۵-۱۰) مفروضات طراحی را نشان می‌دهد.



شکل (۴-۵) هندسه اولیه خرابای مستوی ۱۵ میله ای

همانگونه که در مفروضات طراحی مشاهده می‌شود ۲۳ متغیر طراحی مستقل وجود دارد که ۱۵ تای آن مربوط به متغیرهای طراحی سطح مقطع اعضا و ۸ تای دیگر مربوط به متغیرهای هندسی یا همان مختصات قابل تغییر گرهی با دقت مفروض (۰/۰۱ in) (۰/۰۲۵۴ cm) است. همانگونه که در جدول (۵-۱۱) مشاهده می‌شود در صورتیکه هندسه اولیه خرابا ثابت فرض شود و بهینه یابی توسط برنامه مشروح در مرجع [36] تنها برای سطح مقطع اعضا انجام پذیرد وزن  $372/65 \text{ N}$  ( $83/67 \text{ lb}$ ) بدست می‌آید حال آنکه وو و چاو در مرجع [35] با بهینه یابی مقاطع منفصل و هندسه خرابا، بهینه محلی با وزن  $536/80 \text{ N}$  ( $120/528 \text{ lb}$ ) را بدست آورده اند.

جدول (۵-۱۰) مفروضات طراحی خرابای ۱۵ میله ای

متغیرهای طراحی			
$A_i ; i = 1, \dots, 15$	متغیرهای سطح مقطع :		
$x_2 = x_6 ; x_3 = x_7 ; y_2 ; y_3 ; y_4 ; y_6 ; y_7 ; y_8$	متغیرهای هندسی :		
اطلاعات قيود			
قيود تنش محوری :			
$(\sigma_t)_i \leq 172.4 \text{ Mpa (25 Ksi)} ; i = 1, \dots, 15$ $ (\sigma_c)_i  \leq 172.4 \text{ Mpa (25 Ksi)} ; i = 1, \dots, 15$			
قيود متغیرهای هندسی <sup>۱</sup> :			
$254 \text{ cm (100 in)} \leq x_2 \leq 355.6 \text{ cm (140 in)} ; 558.8 \text{ cm (220 in)} \leq x_3 \leq 660.4 \text{ cm (260 in)} ;$ $254 \text{ cm (100 in)} \leq y_2 \leq 355.6 \text{ cm (140 in)} ; 254 \text{ cm (100 in)} \leq y_3 \leq 355.6 \text{ cm (140 in)} ;$ $127 \text{ cm (50 in)} \leq y_4 \leq 228.6 \text{ cm (90 in)} ; -50.8 \text{ cm (-20 in)} \leq y_6 \leq 50.8 \text{ cm (20 in)} ;$ $-50.8 \text{ cm (-20 in)} \leq y_7 \leq 50.8 \text{ cm (20 in)} ; 50.8 \text{ cm (20 in)} \leq y_8 \leq 152.4 \text{ cm (60 in)} ;$ $P = 0.0254 \text{ cm (0.01 in)}$			
لیست پروفیل‌های موجود			
$A_i \in S = \{ 0.716, 0.910, 1.123, 1.419, 1.742, 1.852, 2.239, 2.839, 3.477, 6.155, 6.974, 7.574, 8.600, 9.600, 11.381, 13.819, 17.400, 18.064, 20.200, 23.00, 24.6, 31.0, 38.4, 42.4, 46.4, 55.0, 60.0, 70.0, 86.0, 92.193, 110.774, 123.742 \} \text{ (cm}^2\text{)}$ $A_i \in S = \{ 0.111, 0.141, 0.174, 0.22, 0.27, 0.287, 0.347, 0.44, 0.539, 0.954, 1.081, 1.174, 1.333, 1.488, 1.764, 2.142, 2.697, 2.8, 3.131, 3.565, 3.813, 4.805, 5.952, 6.572, 7.192, 8.525, 9.3, 10.85, 13.33, 14.29, 17.17, 19.18 \} \text{ (in}^2\text{)} ; i = 1, \dots, 15$			
اطلاعات بارگذاری			
$F_y$	$F_x$	گره	وضعیت
-44.537 KN (-10.0 Kips)	0	8	1
خصوصیات مصالح			
$6.895 \times 10^4 \text{ Mpa (1.0} \times 10^4 \text{ Ksi)}$	مدول الاستیسیته ، E		
$0.0272 \text{ N/cm}^3 \text{ (0.1 lb/in}^3\text{)}$	وزن مخصوص ، ρ		

میزان نقض احتمالی هر عضو برای تنش مجاز محوری ، تنش کمانش اوپلری و تغییر مکان

مجاز گرهی توسط روابط (۴-۱) تعیین می‌شود :

$$V_{i,\sigma} = \left| \frac{\sigma_i}{\sigma_a} \right| - 1 > 0$$

جدول (۵-۱۱) نتایج طرح بهینه مقاطع منفصل و هندسه خرابای ۱۵ میله ای

متغیرهای طراحی	بهینه یابی مقاطع و هندسه [35]	بهینه یابی مقاطع برای هندسه اولیه [36]
متغیرهای طراحی سطح مقطع، $\text{cm}^2$ (in <sup>2</sup> )		
A <sub>1</sub>	7.574 (1.174)	6.155 (0.954)
A <sub>2</sub>	6.155 (0.954)	6.155 (0.954)
A <sub>3</sub>	2.839 (0.440)	0.910 (0.141)
A <sub>4</sub>	8.600 (1.333)	6.974 (1.081)
A <sub>5</sub>	6.155 (0.954)	3.477 (0.539)
A <sub>6</sub>	1.123 (0.174)	2.239 (0.347)
A <sub>7</sub>	2.839 (0.440)	0.716 (0.111)
A <sub>8</sub>	2.839 (0.440)	0.716 (0.111)
A <sub>9</sub>	6.974 (1.081)	0.716 (0.111)
A <sub>10</sub>	8.600 (1.333)	2.839 (0.440)
A <sub>11</sub>	1.123 (0.174)	1.419 (0.220)
A <sub>12</sub>	1.123 (0.174)	0.910 (0.141)
A <sub>13</sub>	2.239 (0.347)	2.839 (0.440)
A <sub>14</sub>	2.239 (0.347)	2.839 (0.440)
A <sub>15</sub>	2.839 (0.440)	1.123 (0.174)
متغیرهای طراحی هندسی، (in) cm		
x <sub>2</sub>	312.9 (123.189)	-
x <sub>3</sub>	588.251 (231.595)	-
y <sub>2</sub>	272.26 (107.189)	-
y <sub>3</sub>	302.705 (119.175)	-
y <sub>4</sub>	153.573 (60.462)	-
y <sub>6</sub>	-42.489 (-16.728)	-
y <sub>7</sub>	39.535 (15.565)	-
y <sub>8</sub>	93.078 (36.645)	-
وزن، N (lb)	536.80 (120.528)	372.65 (83.67)

$$v_{i,bk} = \left| \frac{\sigma_{i,c}}{\sigma_{cr}} \right| - 1 > 0$$

$$V_{\Delta_j} = \left| \frac{\Delta_j}{\Delta_a} \right| - 1 > 0 \quad ; \quad i=1, \dots, ne ; j=1, \dots, ndof$$

(۱۳-۵)

در صورتیکه قیدی نقض نشود مقدار  $v_i=0$  منظور می شود. در روابط فرقی  $\sigma_i$  تنش محوری عضو  $i$  ام (کششی یا فشاری)،  $\sigma_{i,c}$  تنش محوری فشاری عضو  $i$  ام،  $\sigma_a$  تنش مجاز محوری،  $\sigma_{cr}$  تنش بحرانی کمانش اوپلری،  $\Delta$  تغییر مکان درجه آزادی زام،  $\Delta_a$  تغییر مکان مجاز گرهی،  $ne$  تعداد اعضای خرپا،  $ndof$  تعداد درجات آزادی گرهی خرپا،  $v_{i,\sigma}$ ،  $v_{i,bk}$  و  $v_{i,\Delta}$  به ترتیب میزان نقض عضو  $i$  ام از تنش مجاز محوری، تنش کمانش اوپلری و تغییر مکان مجاز گرهی است. به این ترتیب به ازای جمیع عوامل میزان نقض کلی عضو  $i$  ام از رابطه  $V_i = v_{i,\sigma} + v_{i,bk} + v_{i,\Delta}$  حاصل می شود.  $V_i > 0$  بیانگر ضعف عضو  $i$  ام در پاسخ سازه نسبت به اعمال بارهای خارجی است.

#### ۵-۲-۴ عملیات بهینه یابی مقاطع منفصل و هندسه پیوسته خرپاها

در این مرحله نخست طول زیر رشته متغیرهای طراحی سطح مقطع مانند قسمت قبل محاسبه می شود و طول زیر رشته هر متغیر طراحی هندسی نیز (مختصات قابل تغییر گرهی) توسط رابطه زیر تعیین می گردد:

$$l_c = \text{INT} \left[ 1 + \frac{\log \left( \frac{c^u - c^l}{\varepsilon} \right)}{\log 2} \right] \quad (14-5)$$

در رابطه بالا  $l_c$  طول زیر رشته متغیر طراحی هندسی،  $c^l$  و  $c^u$  به ترتیب کرانه های فوقانی و تحتانی مختصات قابل تغییر گرهی و  $\varepsilon$  دقت مورد نیاز برای مختصات گرهی است. در ادامه عملیات، اعداد تصادفی دودویی به زیر رشته ها و به تعداد کروموزوم های واقع در جمعیت (pop-size) اختصاص می یابد.

در مرحله رمزگشایی مقدار نظیر سطح مقطع برای هر متغیر طراحی از رابطه زیر حاصل می شود:

$$A_i = S_{(l_i+1)} \quad ; \quad i = 1, \dots, n \quad (15-5)$$

در رابطه بالا  $I_i$  مقدار صحیح<sup>۱</sup> زیر رشته  $i$ ام پس از نگاشت به سیستم دهگانی<sup>۱</sup> سطح مقطع متغیر طراحی  $i$ ام در موقعیت  $I_i + 1$  لیست پروفیل‌های موجود  $S$  و  $n$  تعداد متغیرهای طراحی سطح مقطع می‌باشد. مقدار فیزیکی هر متغیر طراحی هندسی پس از رمزگشایی بصورت زیر بدست می‌آید:

$$C_i = C_i^l + I_i \frac{C_i^u - C_i^l}{2^{I_{c,i}} - 1} \quad ; \quad i = 1, \dots, n \quad (۱۶-۵)$$

در رابطه بالا  $C_i^u$ ،  $C_i^l$  و  $C_i$  به ترتیب مقادیر فیزیکی، کرانه فوقانی و کرانه تحتانی متغیر طراحی هندسی  $i$ ام،  $I_i$  و  $I_{c,i}$  به ترتیب مقدار صحیح و طول زیر رشته متغیر طراحی هندسی  $i$ ام و  $n$  تعداد کل متغیرهای طراحی هندسی (مختصات قابل تغییر گرهی) است. پس از رمزگشایی، هر کروموزوم معرف یک طرح هندسی با مختصات گرهی و سطح مقطع‌های جدید می‌باشد. لذا با تحلیل سازه به تعداد pop-size و تعیین شایستگی هر کروموزوم، فرآیندهای انتخاب، پیوند و حذف همانند مطالب مطرح شده در فصل دوم بکار گرفته می‌شود.

## ۵-۲-۵ مثالهای عددی

در این بخش دو مثال کاربردی جهت بررسی روشهای مطرح شده بهینه‌یابی مقاطع منفصل و هندسه پیوسته خرابها همراه با مقایسه نتایج بدست آمده از این روش‌ها با برخی از مراجع می‌باشد در این بخش در قسمت اول به بهینه‌یابی مقاطع و هندسه خرابی ۱۸ میله‌ای که توسط ارباتور و همکارانش<sup>۲</sup> مورد بررسی قرار گرفت، اختصاص دارد و در قسمت دوم به بهینه‌یابی مقاطع و هندسه خرابی ۲۵ میله‌ای فضایی که توسط چاوو و همکارانش<sup>۳</sup> مورد بررسی قرار گرفت، می‌پردازیم.

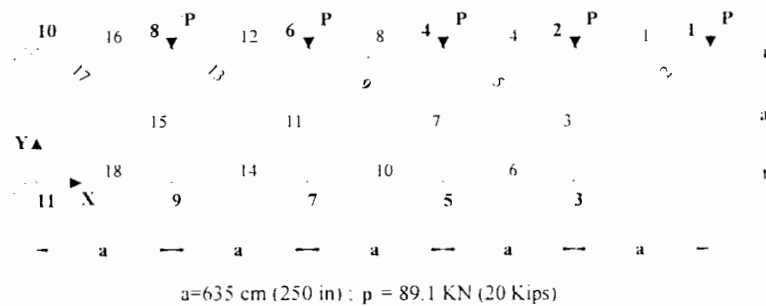
<sup>۱</sup> Decimal Integer

<sup>۲</sup> Hasançebi & Erbatur

<sup>۳</sup> Wu & Chow

### مثال ۳-۵ خریای مستوی ۱۸ میله ای

این مثال به بهینه یابی مقاطع و هندسه خریای ۱۸ میله ای اختصاص دارد که شکل (۵-۵) هندسه اولیه آن همراه با بارهای موثر و جدول (۵-۱۲) اطلاعات طراحی آن را نمایش می دهد.



شکل (۵-۵) هندسه اولیه خریای ۱۸ میله ای

همانگونه که ملاحظه می شود سطح مقطع اعضای خریا به چهار گروه دسته بندی شده و مختصات  $X$  و  $Y$  گره های ۳، ۵، ۷ و ۹ می توانند تغییر کنند. لذا ۴ متغیر طراحی سطح مقطع و ۸ متغیر طراحی هندسی وجود دارد. مقاطع از لیست پروفیل های  $S$  مشتمل بر مقاطع  $(2/0 \text{ in}^2)$   $12/90.3 \text{ cm}^2$  تا  $(21/75 \text{ in}^2)$   $140/322 \text{ cm}^2$  با نمو افزایشی  $(0/25 \text{ in}^2)$   $1/613 \text{ cm}^2$  به تعداد ۸۰ عدد انتخاب شده اند. دقت  $(1/0 \text{ in})$   $2/54 \text{ cm}$  نیز برای متغیرهای طراحی هندسی منظور شده است.

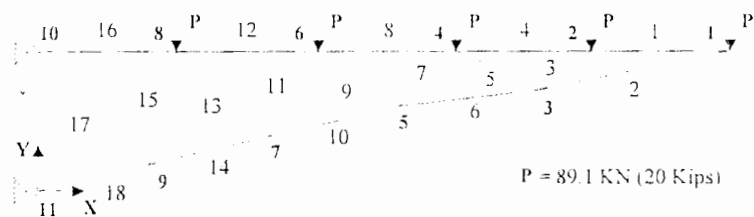
در جدول (۵-۱۳) نتایج طرح بهینه مقاطع منفصل و هندسه پیوسته خریا بر اساس روش انجام شده بهمراه نتایج بدست آمده مراجع جهت مقایسه نمایش داده شده است. شکل (۵-۶) نیز هندسه تغییر شکل یافته خریا پس از بهینه یابی را نمایش می دهد.

جدول (۵-۱۲) اطلاعات طراحی خرابای ۱۸ میله ای

متغیرهای طراحی			
متغیرهای طراحی سطح مقطع :			
$A_1=A_4=A_8=A_{12}=A_{16}$ ; $A_2=A_6=A_{10}=A_{14}=A_{18}$ ; $A_3=A_7=A_{11}=A_{15}$ ; $A_5=A_9=A_{13}=A_{17}$			
متغیرهای طراحی هندسی :			
$x_3, y_3, x_5, y_5, x_7, y_7, x_9, y_9$			
اطلاعات مربوط به قیود			
قیود تنش محوری :		$(\sigma_t)_i \leq 137.9 \text{ Mpa ( 20 Ksi ) } ; i = 1, \dots, 18$	
قیود تنش کمانش اولبری :		$ (\sigma_c)_i  \leq 137.9 \text{ Mpa ( 20 Ksi ) } ; i = 1, \dots, 18$	
قیود متغیرهای طراحی هندسی :		$ (\sigma_c)_i  \leq \alpha EA_i / L_i^2 ; i = 1, \dots, 18$	
$-571.5 \text{ cm ( - 225 in ) } \leq y_3, y_5, y_7, y_9 \leq 622.3 \text{ cm ( 245 in ) } ;$			
$1968.5 \text{ cm ( 775 in ) } \leq x_3 \leq 3111.5 \text{ cm ( 1225 in ) } ;$			
$1333.5 \text{ cm ( 525 in ) } \leq x_5 \leq 2476.5 \text{ cm ( 975 in ) } ;$			
$698.5 \text{ cm ( 275 in ) } \leq x_7 \leq 1841.5 \text{ cm ( 725 in ) } ;$			
$63.5 \text{ cm ( 25 in ) } \leq x_9 \leq 1206.5 \text{ cm ( 475 in ) } ; p = 2.54 \text{ cm ( 1.0 in )}$			
لیست پروفیل‌های موجود			
$A_i \in S = \{ 12.903, 14.516, \dots, 138.709, 140.322 \} \text{ ( cm}^2 \text{ )}$			
$A_i \in S = \{ 2.0, 2.25, \dots, 21.5, 21.75 \} \text{ ( in}^2 \text{ ) } ; i = 1, \dots, 18$			
اطلاعات بارگذاری			
	$F_y$	$F_x$	گره‌ها
	$-89.075 \text{ KN ( - 20 Kips )}$	0	1,2,4,6,8
			وضعیت بار 1
خصوصیات مکانیکی مصالح			
$6.895 \times 10^4 \text{ Mpa ( 1.0} \times 10^4 \text{ Ksi )}$	مدول الاستیسیته، E		
4.0	ضریب کمانش، $\alpha$		
$0.0272 \text{ N/cm}^3 \text{ ( 0.1 lb/in}^3 \text{ )}$	وزن مخصوص، p		

جدول (۵-۱۳) نتایج طرح بهینه مقاطع منفصل و هندسه پیوسته خرابای ۱۸ میله ای بر اساس روشهای مطرح شده در این رساله و مقایسه با مرجع [37]

متغیرهای طراحی	روش پیشنهادی در این رساله	[37] Erbatur & Hasançebi
متغیرهای طراحی سطح مقطع، $(\text{in}^2) \text{Cm}^2$		
$A_1$	79.032 ( 12.25 )	80.645 ( 12.50 )
$A_2$	116.129 ( 18.0 )	117.742 ( 18.25 )
$A_3$	33.871 ( 5.25 )	35.484 ( 5.5 )
$A_5$	27.419 ( 4.25 )	24.194 ( 3.75 )
متغیرهای طراحی هندسی. $(\text{in}) \text{cm}$		
$X_3$	2319.02 (913.0)	2369.82 (933.0)
$Y_3$	474.472 (186.8)	477.52 (188.0)
$X_5$	1651.0 (650.0)	1671.32 (658.0)
$Y_5$	382.27 (150.5)	375.92 (148.0)
$X_7$	1063.752 (418.8)	1071.88 (422.0)
$Y_7$	247.396 (97.4)	254.0 (100.0)
$X_v$	520.192 (204.8)	520.7 (205.0)
$Y_v$	67.818 (26.7)	81.28 (32.0)
وزن. $(\text{lb}) \text{N}$	20255.17 (4547.9)	20372.658(4574.28)

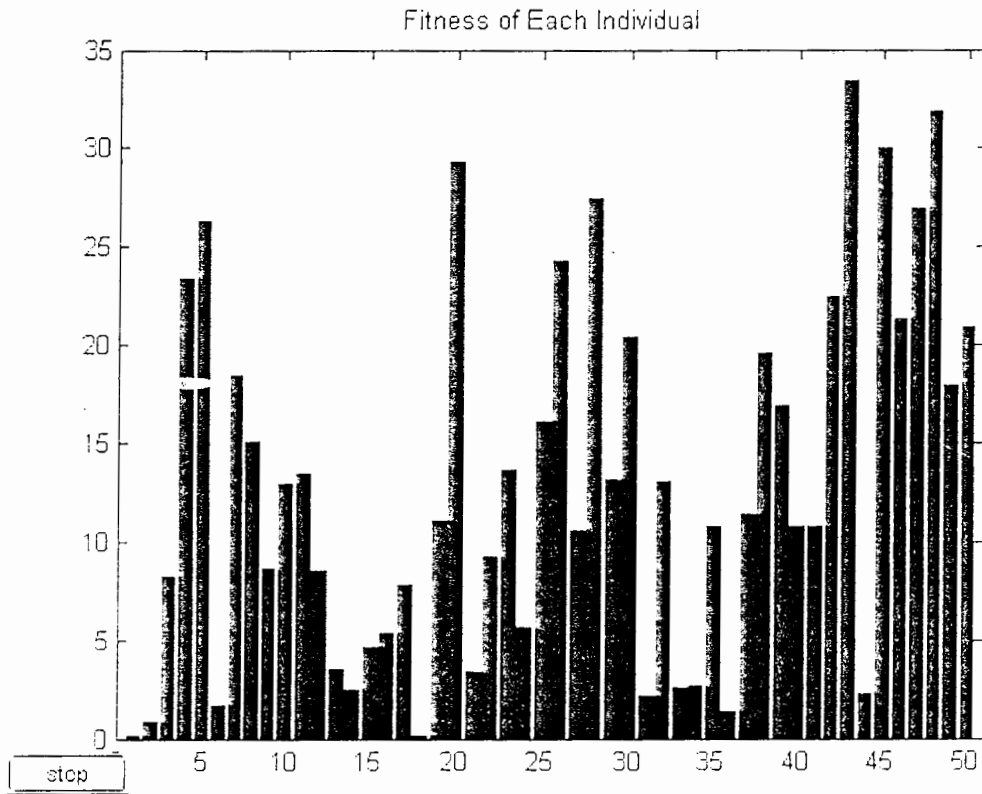


شکل (۵-۶) هندسه بهینه خرابای مستوی ۱۸ میله ای

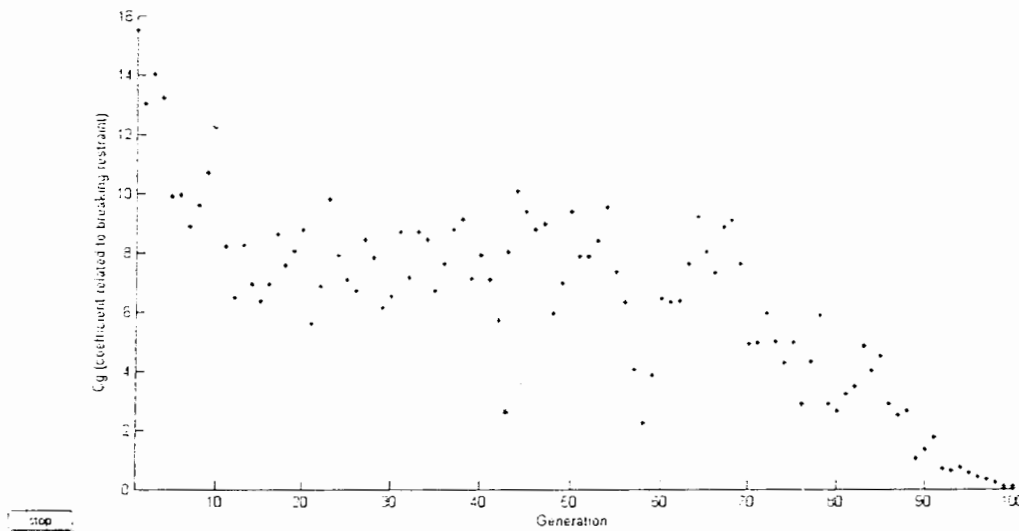
در این مثال نرخ پیوند ۰/۹، نرخ جهش ۰/۰۵، تعداد اعضای جمعیت برابر ۵۰، تعداد نسل سازی ۱۰۰ و طول زیر رشته متغیرهای طراحی سطح مقطع برابر ۱۶ منظور شده است. جهت مقابله با هندسه های ناپایدار کننده سازه با تشکیل دترمینان ماتریس سختی و کنترل آن در جاهائیکه این دترمینان نزدیک به صفر می باشد سازه ناپایدار بوده و این کرموزوم ها به شدت جریمه می شوند و



شایستگی کروموزوم مربوطه صفر لحاظ شود. به این ترتیب در فرآیند انتخاب این کروموزوم از جمعیت بعدی حذف خواهد شد.



شکل (۷-۵) هیستوگرام مربوط به شایستگی اعضای جمعیت در طرح بهینه

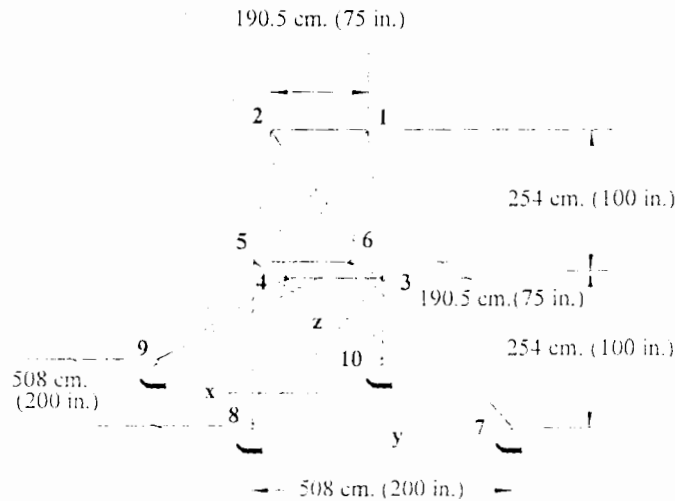


شکل (۸-۵) نمودار مربوط به نقض قیود (پارامتر  $C_g$ ) در نسلهای مختلف

### مثال ۴-۵ خرابای فضایی ۲۵ میله ای

در این مثال بهینه یابی مقاطع و هندسه یک خرابای فضایی ۲۵ میله ای شرح داده شده است.

شکل (۴-۴) هندسه اولیه خرابای مزبور همراه با شماره گذاری گره های آن و جدول (۴-۵) مختصات گره ها، گروه بندی ۸ گانه اعضا همراه با شماره گره های انتهایی اعضا را نمایش می دهد.



شکل (۴-۵) هندسه اولیه خرابای فضایی ۲۵ میله ای

جدول (۴-۵) مختصات گرهی و گروه بندی اعضای خرابای فضایی ۲۵ میله ای

Z, cm (in)	Y, cm (in)	X, cm (in)	گره	شماره عضو. (شماره گره های انتهایی)	گروه
508 (200)	0.0	-95.25 (-37.5)	1	1(1,2)	X <sub>1</sub>
508 (200)	0.0	95.25 (37.5)	2	2 (1,4), 3 (2,3), 4 (1,5), 5 (2,6)	X <sub>2</sub>
254 (100)	95.25 (37.5)	-95.25 (-37.5)	3	6 (2,5), 7 (2,4), 8 (1,3), 9 (1,6)	X <sub>3</sub>
254 (100)	95.25 (37.5)	95.25 (37.5)	4	10 (3,6), 11 (4,5)	X <sub>4</sub>
254 (100)	-95.25 (-37.5)	95.25 (37.5)	5	12 (3,4), 13 (5,6)	X <sub>5</sub>
254 (100)	-95.25 (-37.5)	-95.25 (-37.5)	6	14(3,10), 15 (6,7), 16 (4,9), 17 (5,8)	X <sub>6</sub>
0.0	254 (100)	-254 (-100)	7	18(3,8), 19 (4,7), 20(6,9), 21(5,10)	X <sub>7</sub>
0.0	254 (100)	254 (100)	8	22(3,7), 23(4,8), 24(5,9), 25(6,10)	X <sub>8</sub>
0.0	-254 (-100)	254 (100)	9		
0.0	-254 (-100)	-254 (-100)	10		

جدول (۵-۱۵) اطلاعات طراحی خرپای فضایی ۲۵ میله ای

متغیرهای طراحی	
متغیرهای طراحی سطح مقطع :	
$X_1; X_2; X_3; X_4; X_5; X_6; X_7; X_8$	
متغیرهای طراحی هندسی :	
$X_4 = X_5 = -X_3 = -X_6; X_8 = X_9 = -X_7 = -X_{10}; y_3 = y_4 = -y_5 = -y_6; y_7 = y_8 = -y_9 = -y_{10}; Z_3 = Z_4 = Z_5 = Z_6$	
اطلاعات مربوط به قیود	
قیود تنش محوری :	
$(\sigma_t)_i \leq 275.8 \text{ Mpa (40 Ksi)} ; i = 1, \dots, 25$	
$ (\sigma_c)_i  \leq 275.8 \text{ Mpa (40 Ksi)} ; i = 1, \dots, 25$	
بند تغییر مکان (در کلیه جهات محورهای مختصات) :	
$ \Delta_i  \leq 0.89 \text{ cm (0.35 in)} ; i = 1, \dots, 6$	
قیود جانبی <sup>۱</sup> متغیرهای طراحی هندسی :	
$50.8 \text{ cm (20 in)} \leq x_4 \leq 152.4 \text{ cm (60 in)} ; 101.6 \text{ cm (40 in)} \leq x_8 \leq 203.2 \text{ cm (80 in)}$	
$101.6 \text{ cm (40 in)} \leq y_4 \leq 203.2 \text{ cm (80 in)} ; 254 \text{ cm (100 in)} \leq y_8 \leq 355.6 \text{ cm (140 in)}$	
$228.6 \text{ cm (90 in)} \leq z_4 \leq 330.2 \text{ cm (130 in)} ; P = 0.0254 \text{ cm (0.01 in)}$	
لیست پروفیل‌های موجود	
$A_i \in S = \{ 0.645 \times I (I = 1, \dots, 26), 18.064, 19.355, 20.645, 21.935 \} (\text{cm}^2) ; i = 1, \dots, 25$	
$A_i \in S = \{ 0.1:1 (I = 1, \dots, 26), 2.8, 3.0, 3.2, 3.4 \} (\text{in}^2) ; i = 1, \dots, 25$	

جدول (۵-۱۶) اطلاعات مربوط به بارگذاری و خصوصیات مکانیکی مصالح خرپای ۲۵ میله ای

اطلاعات بارگذاری				
$F_z, \text{ KN (Kips)}$	$F_y, \text{ KN (Kips)}$	$F_x, \text{ KN (Kips)}$	کرد	وضعیت بار
-44.537 (-10)	-44.537 (-10)	4.454 (1.0)	1	1
-44.537 (-10)	-44.537 (-10)	0.0	2	
0.0	0.0	2.227 (0.5)	3	
0.0	0.0	2.672 (0.6)	6	
خصوصیات مکانیکی مصالح				
$6.895 \times 10^4 \text{ Mpa (1.0} \times 10^4 \text{ Ksi)}$				مدول الاستیسیته، E
$0.0272 \text{ N/cm}^3 (0.1 \text{ lb/in}^3)$				وزن مخصوص، p

در جداول (۵-۱۵) و (۵-۱۶) نیز اطلاعات طراحی درج شده است. در جدول (۵-۱۷) نتایج

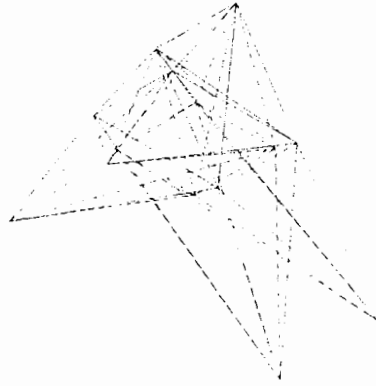
طرح بهینه مقاطع منفصل و هندسه پیوسته خرپای فضایی ۲۵ میله ای بر اساس روش مطرح شده و

مرجع [35] جهت مقایسه درج شده است. در اشکال (۵-۱۰)، (۵-۱۱) و (۵-۱۲) نیز هندسه بهینه خرابا بهمراه هندسه اولیه آن جهت مقایسه نمایش داده شده است.

جدول (۵-۱۷) نتایج طرح بهینه مقاطع منفصل و هندسه پیوسته خرابای فضایی ۲۵ میله ای بر اساس روش مطرح شده و مقایسه با مرجع [35]

متغیرهای طراحی	روش این رساله	[35].Chow & Wu
متغیرهای طراحی سطح مقطع. $(in^2) cm^2$		
$X_1$	0.645 (0.1)	0.645 (0.1)
$X_2$	0.645 (0.1)	1.290 (0.2)
$X_3$	7.097 (1.1)	7.097 (1.1)
$X_4$	0.645 (0.1)	1.290 (0.2)
$X_5$	0.645 (0.1)	1.935 (0.3)
$X_6$	0.645 (0.1)	0.645 (0.1)
$X_7$	0.645 (0.1)	1.290 (0.2)
$X_8$	6.452 (1.0)	5.806 (0.9)
متغیرهای طراحی هندسی. (in) cm		
$x_4$	92.024 (36.23)	104.318 (41.07)
$y_4$	148.742 (58.56)	135.814 (53.47)
$z_4$	293.599 (115.59)	316.484 (124.60)
$x_8$	118.008 (46.46)	129.032 (50.80)
$y_8$	324.993 (127.95)	333.959 (131.48)
وزن. (lb)N	552.264 (124.0)	606.6 (136.2)

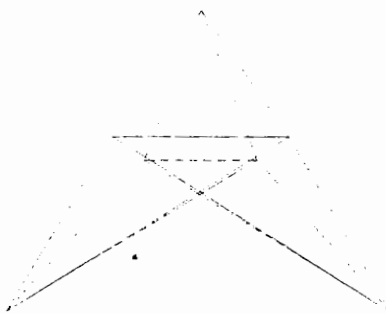
لازم بذکر است که در این مثال کلیه پارامترهای مربوط به GA همانند مفروضات مثال (۳-۵) می باشد.



شکل (۱۰-۵) هندسه بهینه شده (خط پیوسته) به همراه هندسه اولیه (خط چین) خرپای فضایی ۲۵ میله ای



شکل (۱۱-۵) نمای X-Z هندسه بهینه (خط پیوسته) و هندسه اولیه (خط چین) خرپای ۲۵ میله ای



شکل (۱۲-۵) نمای Y-Z هندسه بهینه (خط پیوسته) و هندسه اولیه (خط چین) خرپای ۲۵ میله ای

## ۵-۲-۶ نتیجه گیری

در این قسمت بهینه یابی مقاطع منفصل و هندسه پیوسته خرابها مورد بررسی قرار گرفت. ملاحظه گردید گاهی بهینه یابی سطح مقطع و هندسه وزن بیشتری نسبت به حالتی که فقط بهینه یابی مقاطع برای هندسه اولیه انجام شود، نتیجه می دهد. این مساله بدلیل بزرگ شدن دامنه فضای طراحی ناشی از طولیل شدن طول کروموزوم ها و ترکیبات بسیار زیادی است که مقاطع منفصل و مختصات قابل تغییر گرهی می توانند از خود نشان دهند.

در مسائلی که دارای تعداد زیادی از متغیرهای طراحی هستند و قیود متفاوتی نیز بر مسئله حاکم است، اغلب فضای قابل قبول طراحی بسیار محدود می باشد. بنابراین بیشتر کروموزوم های تولیدی در الگوریتم ژنتیک کروموزوم های مرده می باشند و قیود را اقلان نمی کنند و در این موارد، الگوریتم ژنتیک نمی تواند فضای طراحی را به طور مؤثری مورد کاوش قرار دهد. بنابراین وجود مکانیزمی که در آن کروموزوم های مرده تولید نشود بسیار مهم می باشد.

در برخی از روشها، با بکارگیری یک مجموعه از تابع های شایستگی برای الگوریتم ژنتیک توزیعی، در ابتدا الگوریتم ژنتیک توزیعی مقادیر نقض قیود را کاهش می دهد و به پیدا کردن جوابهایی می پردازد که قیود را اقلان کنند. بعد از پیدا کردن جوابهایی که قیود را اقلان می کنند، مقادیر تابع هدف کاهش می یابد. با این آهنگ الگوریتم ژنتیک می تواند به سمت جواب های بهینه برود.

در این روشها بطور همزمان، الگوریتم ژنتیک توزیعی محیطی نیز به قیود مسئله معرفی می شود که در الگوریتم ژنتیک توزیعی محیطی قیود متفاوتی در هر بخش<sup>۱</sup> اعمال می شود و این روش باعث افزایش قابلیت کاوشگری الگوریتم می شود. در خلال طراحی سازه خرابایی با استفاده از سیستم های کامپیوتری چند شاخه، این روش و الگوریتم ژنتیک توزیعی محیطی در برخی مراجع مورد بررسی قرار

<sup>۱</sup> Island

بررسی قرار گرفت که در اینجا ما به دلیل عدم دسترسی به این سیستم‌های کامپیوتری چند شاخه به انجام این روش با استفاده از یک کامپیوتر PC خانگی و بکارگیری نرم افزار Matlab 7.0 پرداختیم.

بر اساس نتایج بدست آمده، با بکارگیری روش مطرح شده می‌توان اقدام به طراحی و بهینه‌سازی طیف وسیعی از مسائل سازه‌ای کرد و از این روش به عنوان یک شیوه کاربردی، در طراحی سازه‌های بزرگ واقعی بهره بگیریم.

## ۳-۵ بهینه یابی سطح مقطع و توپولوژی سازه های خرپایی با استفاده از

### الگوریتم ژنتیک

#### ۵-۳-۱ مقدمه

در این قسمت توپولوژی بهینه خرپاها تحت اثر چند وضعیت بارگذاری استاتیکی<sup>۱</sup> با اعمال قیود تنش و تغییر مکان مورد مطالعه قرار گرفته است. در این زمینه دو نوع فرمولبندی رایج بکار گرفته می شود. دسته اول بهینه یابی بر اساس برنامه ریزی ریاضی است. دسته دوم بهینه یابی مبتنی بر روشهای کاوش هوشمند تصادفی است که الگوریتم ژنتیک (GA) را می توان به عنوان معروف ترین آنها شناخت. تمایز عمده دو روش فوق الذکر در این است که روشهای برنامه ریزی ریاضی احتیاج به برآورد گرادیان توابع هدف و قیود حاکم بر مساله دارند در حالیکه روشهای کاوش تصادفی بدون نیاز به برآورد گرادیان توابع، بطور هدفمندی بهینه عمومی را در فضای طراحی کاوش می کنند. در صورتیکه فضای طراحی غیر محدب<sup>۲</sup> باشد و یا تابع هدف، تابعی غیر پیوسته<sup>۳</sup> از متغیرهای طراحی باشد، بهینه یابی بر اساس GA کارآمدی خود را به اثبات می رساند. در عین حال عدم شناخت درست از پارامترها و روابط حاکم در GA از قبیل تعداد اعضای جمعیت، تعداد نسل ها<sup>۴</sup>، نرخ پیوند، نرخ جهش، نوع تابع جریمه و مقادیر ثابتهای دخیل در آن، نوع تابع شایستگی و چگونگی فرآیند انتخاب از جمله عواملی است که سبب می شود GA بهینه محلی را بدست آورد. حجم بالای عملیات محاسباتی توسط GA نیز از دیگر مشکلات این روش محسوب می شود که این مشکل با بهبود سرعت و توانایی کامپیوترها در سالهای اخیر به همراه استفاده از برخی از روش های الگوریتم ژنتیک توسعه یافته (DGA) که در آن امکان استفاده از سیستم های کامپیوتری موازی وجود دارد تا حدود زیادی مرتفع شده است.

<sup>۱</sup> Static Loading Conditions

<sup>۲</sup> Non Convex

<sup>۳</sup> Discontinuous

<sup>۴</sup> No of Generation



در بهینه یابی توپولوژی خرپاها، بطور مرسوم از سازه مینا<sup>۱</sup> با موقعیت ثابت نقاط گرهی استفاده می شود. اعضای توپولوژی بهینه بصورت زیر مجموعه ای از اعضای سازه مینا که معمولاً پر عضو انتخاب می شود، تعیین می گردند. در این راستا معمولاً دو مساله پیش می آید : نخست اینکه سازه مینای پر عضو سبب می شود احتمال کسب توپولوژی بهینه محلی افزایش یابد. دیگر اینکه چون قید تنش در مورد اعضای حذف شده<sup>۲</sup> که به آنها معمولاً سطح مقطع کوچک مجازی اختصاص می یابد اعمال نمی شود، توپولوژی بهینه ممکن است منفصل یا ناپایدار<sup>۳</sup> حاصل شود.

در این رساله با استفاده از تئوری گرافها، مفاهیم روش تحلیل سختی و GA سعی شده است با معرفی گراف پایه مناسب برای سازه مینا تعداد نسل سازی برای کسب بهینه عمومی کاهش یابد و حتی الامکان سازه ای بهینه با آرایش میله ای اجرایی حاصل شود. ضمناً همانگونه که اشاره شد چون اختصاص سطح مقطع کوچک مجازی به اعضای منتخب برای حذف ممکن است وجود ناپایداریهای موضعی یا کلی در توپولوژی بهینه را مخفی سازد (حتی اگر قید تغییر مکان ارضاء شده باشد)، لذا در این رساله با استفاده از محاسبه دترمینان ماتریس سختی در جائیکه این دترمینان صفر می شود توپولوژی های ناپایدار تشخیص داده شده و به شدت جریمه می شوند. اینکار تا حد زیادی حجم عملیات عددی را کاهش و پایداری توپولوژی بهینه را تضمین می کند و به این ترتیب در نسخه های بعدی توپولوژی های ناپایدار امکان حضور کمتری پیدا می کنند

<sup>1</sup> Ground Structure<sup>2</sup> Removed (Vanished) Members<sup>3</sup> Isolated or Singular Solution

### ۵-۳-۲ بهینه یابی مقاطع و توپولوژی با استفاده از روش سختی

همانگونه که قبلاً اشاره شد، در بهینه یابی سازه ها مبتنی بر روش سختی (تغییر مکان) ماتریس سختی از مرتبه  $n$  (درجات نامعینی سینماتیکی) هر بار تشکیل و یا احیا می شود و دستگاه معادلات خطی از مرتبه  $n$  نیز حل می گردد. در صورتیکه توپولوژی سازه تغییر کند حضور یا عدم حضور اعضای آن تغییر می یابد. به این ترتیب چنانچه به اعضای منتخب برای حذف سطح مقطع کوچک مجازی اختصاص یابد، ماتریس سختی  $K$ ، هر بار باید تعیین گردد و تحت این شرایط نیز برای هر کروموزوم مرتبط با سازه پایدار، دستگاه معادلات خطی از مرتبه  $n$  بایستی حل شود. ولی چنانچه اشاره شد، در صورتیکه در توپولوژی بهینه اعضای با سطح مقطع کوچک مجازی موجود باشد در صورت کنار گذاشتن این اعضا ممکن است سازه باقیمانده منفصل یا ناپایدار شود. تصمیم در مورد اینکه کدامیک از اینگونه اعضا باقی بمانند و سطح مقطع حداقل اجرایی به آنها اختصاص یابد، به عهده طراح گذاشته می شود و این مساله علاوه بر آنکه در سازه های بزرگ دشوار می باشد وزن سازه بهینه را نیز افزایش می دهد. در این فصل برای جلوگیری از این اشکال اعضای منتخب برای حذف از سازه مبنا کنار گذاشته شده و پس از اصلاح ماتریس  $K$ ، دترمینان ماتریس سختی مجدداً محاسبه می شود و اگر دترمینان ماتریس سختی صفر گردید سازه مربوطه دارای ناپایداری هندسی بوده و در نتیجه عملیات ماتریسی متوقف و کروموزوم مربوطه بشدت جریمه می شود.

از کنترل های بسیار مفیدی که در روند بهینه یابی مقاطع و توپولوژی انجام پذیرفته است تعیین درجه نامعینی استاتیکی سازه قبل از شروع تحلیل آن است. در صورتیکه درجه نامعینی استاتیکی منفی باشد، سازه مربوطه دارای ناپایداری توپولوژیکی است و کروموزوم مربوطه به شدت جریمه می شود. توسط رابطه (۵-۱۷) می توان درجه نامعینی استاتیکی سازه ها را ارزیابی نمود [20].

$$v(S) = aM(S) + bN(S) + cv_0(S) \quad (۵-۱۷)$$

در رابطه فوق  $v(S)$ ،  $M(S)$ ،  $N(S)$  و  $v_0(S)$  به ترتیب درجه نامعینی استاتیکی، تعداد اعضا، تعداد گره ها و تعداد مولفه های سازه  $S$  است. ضرایب  $a$ ،  $b$  و  $c$  برای خرپاهای مستوی به ترتیب برابر  $1$ ،  $2$ - و  $3$  و برای خرپاهای فضایی به ترتیب برابر  $1$ ،  $3$ - و  $6$  است. برای سازه های یکپارچه  $v_0(S) = 1$  می باشد. در صورتیکه کلیه اعضای متصل به یک گره برای حذف انتخاب شده باشند، گره مفروض در کلیه جهات بسته و در رابطه (۵-۱) شرکت داده نمی شود. همچنین اگر کلیه اعضای متصل به گره ای تحت بار و یا تکیه گاه برای حذف انتخاب شده باشند، بدون هر گونه عملیات ماتریسی کروموزوم مربوطه به شدت جریمه خواهد شد. با بررسی هایی که به عمل آمد حتی اگر احتمال حضور اعضا به نحو مناسبی تعیین شده باشد و کروموزوم های معادل با سازه ناپایدار در جمعیت اولیه کاهش یافته باشد، باز با کنترل های ساده فوق الذکر می توان بین  $10$  تا  $30$  درصد حجم عملیات عددی را کاهش داد و کروموزوم هایی با ناپایداری توپولوژیکی را از چرخه عملیات ماتریسی خارج نمود.

### ۵-۳-۳ الگوریتم ژنتیک برای بهینه یابی مقاطع و توپولوژی خرپاها

در بهینه یابی توپولوژیکی خرپاها، متغیرهای طراحی، سطح مقطع اعضا و معمولاً بیت های بولی توپولوژیکی مربوط به حضور یا عدم حضور اعضا در نظر گرفته می شود. آزادی در مقاله خود مکان بیت توپولوژیکی را در سمت چپ زیر رشته متغیر طراحی سطح مقطع معرفی می کند تا سرعت همگرایی افزایش یابد [51]. وی عدم حضور عضو را در قالب صفر شدن مقدار بیت بولی توپولوژیکی و یا مقدار صفر برای متغیر طراحی سطح مقطع (در صورتی که کلیه بیتها صفر شوند) تفسیر می کند. وازکوئز در مقاله خود [52]، که به بحث بر روی مقاله راجیو و کریشنامورتی اختصاص دارد اضافه

کردن سطح مقطع صفر به لیست مقاطع منفصل و حذف متغیر توپولوژیکی را پیشنهاد می‌کند. به این ترتیب حصول سطح مقطع صفر برای هر عضو به منزله حذف آن تلقی خواهد شد.

حذف متغیر توپولوژیکی یا تلقی حذف عضو به ازای حصول سطح مقطع صفر عنوان شده در مقالات فوق، این اشکال را ایجاد می‌کند که در صورت گروه بندی اعضا، هر متغیر طراحی سطح مقطع نماینده سطح مقطع اعضای گروه مربوطه است و سطح مقطع صفر باعث حذف کلیه اعضای آن گروه می‌شود. این مساله ممکن است مورد نظر طراح نباشد و در عین حال سبب ناپایداری یا حصول بهینه محلی<sup>۱</sup> شود. بعنوان مثال اگر کلیه اعضای قطری یک خرپای مستوی، بعنوان یک گروه در نظر گرفته شوند سطح مقطع صفر باعث حذف همگی آنها خواهد شد. در حالیکه ممکن است در توپولوژی بهینه حضور برخی از آنها لازم باشد. در ضمن همانگونه که راجح توضیح داده است هنگام تولید جمعیت اولیه با تعریف متغیر بولی توپولوژیکی احتمال حضور یا عدم حضور عضو توسط طراح تعیین می‌شود در حالیکه اگر عضو بخواهد فقط به ازای سطح مقطع صفر حذف شود، بسته به طول زیر رشته احتمال آن پایین و سازه های تولید شده در جمعیت اولیه پر عضو و این مسأله بر روی کیفیت نتایج تاثیر منفی می‌گذارد.

از دیگر پیشنهاداتی که برای بهینه یابی توپولوژی خرپاها در مراجع عنوان شده است، اختصاص اشکال معینی از آرایش میله ها بعنوان متغیرهای توپولوژیکی در جمعیت اولیه و تولید کروموزوم هایی با طولیای متفاوت (VGA) است. هدف از اینکار کاهش طول کروموزوم ها در سازه های بزرگ و پیچیده جهت افزایش سرعت همگرایی و ممانعت از حصول بهینه محلی است. از معایب این روش محدود شدن دامنه کاوش فضای طراحی و احتمالاً از دست دادن بهینه واقعی است [52]. از روشهای توپولوژیکی رایج، اختصاص احتمال حضور اعضا  $p^e$  به بیت های بولی توپولوژیکی در تولید جمعیت اولیه و اختصاص سطح مقطع کوچک به اعضای منتخب برای حذف می‌باشد. اختصاص مقدار

<sup>۱</sup> Local Optimum

مناسب به  $p^c$  سبب می شود که کروموزوم های معادل با سازه ناپایدار در جمعیت اولیه کاهش و نتایج توپولوژی بهینه بهبود چشمگیر یابد. اما تعیین مقدار مناسب برای آن (همانند سایر پارامترهای دخیل در GA) مستلزم طی مراحل سعی و خطا است و عملیات GA را از آنچه که هست حجیم تر می کند. در اینصورت قضاوت در مورد آنکه کدامیک از اینگونه اعضا برای حفظ پایداری سازه باقی بمانند به طراح واگذار می شود و ناچاراً بایستی به آنها سطح مقطع حداقل اجرایی اختصاص یابد. این مساله علاوه بر آنکه در سازه های بزرگ مشکل آفرین می شود ممکن است سازه حاصل را از حالت بهینه بودن نیز خارج سازد.

#### ۵-۳-۴ آشنایی با تعاریف پایه نظریه گرافها

برای سازه های اسکلتی، توپولوژی بهینه معمولاً بر اساس روش سازه مبنا بدست می آید. در این روش یک مجموعه از حالات مجاز نقاط گرهی ایجاد می شود و نحوه اتصال گره ها به یکدیگر توسط اعضا، توپولوژی سازه را پدید می آورد. در یک نگاه ریاضی به مساله مزبور، علی الاصول تعداد گره های ممکن برای سازه مبنا بینهایت است و این امر کار محاسبات را طولانی و پیچیده می کند. برای رفع این مشکل یکی از راه حلها این است که مکان گره های سازه مبنا توسط یک شبکه متعامد مشخص گردد که در آن فاصله جزء به جزء گره های شبکه با توجه به نوع بارگذاری و تکیه گاه ها، قبل از شروع حل مساله مقادیر ثابتی در نظر گرفته شود. به این ترتیب بهینه عمومی<sup>۱</sup> در مجموعه گره های تعریف شده در شبکه متعامد کاوش می شود که ممکن است بهینه واقعی<sup>۲</sup> در فضای طراحی نباشد. مرجع [54] مطالعاتی را برای بهبود روش سازه مبنا ارائه داده است.

<sup>۱</sup> Global Optimum

<sup>۲</sup> True Optimum

در ادامه به معرفی تعاریف مقدماتی نظریه گرافها که می تواند به عنوان یک ابزار قوی در اختیار بهینه یابی سازه ها قرار گیرد، پرداخته می شود. این نظریه کاربردهای مختلفی در شاخه های گوناگون علوم و تکنولوژی و حتی در ریاضیات پیدا کرده است زیرا وسیله بسیار مناسبی برای مدل سازی است [20].

تعاریف اساسی تئوری گرافها به شرح ذیل می باشد :

یک گراف  $S$  عبارت است از مجموعه راسها  $N(S)$ ، مجموعه اعضوها  $M(S)$  و یک تابع انطباق که به هر یک از اعضوهای گراف، یک زوج از رئوس گراف را وابسته می سازد. تابع انطباق به هر یک از اعضای گراف زوج گره های  $\{n_1, n_2\}$  را وابسته می سازد، که در این صورت رئوس  $n_1$  و  $n_2$  همسایه نامیده می شوند.

هرگاه بین دو راس بیش از یک عضو قرار داشته باشد، به آن اعضوها، اعضای چندگانه گویند.

اگر در یک گراف عضوی داشته باشیم که فقط با یک گره مشخص شود بدان حلقه گویند.

گراف متناهی که فاقد حلقه و عضو چندگانه باشد گراف ساده نامیده می شود.

گراف خنثی عبارتست از گرافی که در آن  $N \neq \phi$  ولی  $M = \phi$  باشد. برای معرفی بارهای گرهی وارد بر یک سازه می توان از این گراف استفاده نمود.

گراف کامل گرافی است که تمام گره هایش به یکدیگر متصل شده باشند. تعداد اعضای این گراف با

$$N \text{ گره برابر } \frac{N(N-1)}{2} \text{ است.}$$

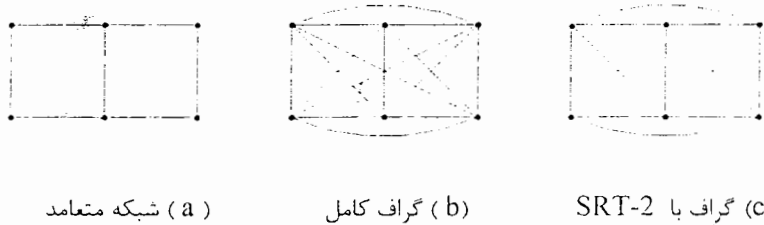
برای رسیدن به توپولوژی بهینه سراسری از گرافهای پایه مختلف می توان استفاده نمود. بعنوان مثال

سازه مبنا می تواند نخست گرافی کامل با درجه نامعینی استاتیکی داخلی  $\frac{(N-3)(N-2)}{2}$  باشد و در

مراحل بعدی با حذف اعضای با طول بلندتر از  $L_{max}$ ، گراف پایه را محدودتر کرده و مراحل بهینه یابی را

تکرار نمود. یکی دیگر از گرافهای پایه، گراف با SRT-2 است. در این گراف گره هایی که فاصله آنها از

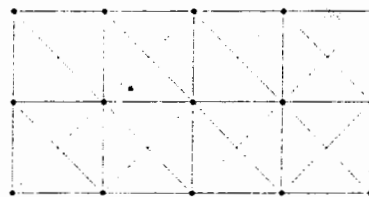
یکدیگر در گراف متعامد (شبکه متعامد) بزرگتر از ۲ باشد، به یکدیگر وصل نمی شوند. منظور از فاصله بین دو گره، حداقل تعداد عضوهای بین آن دو می باشد. برای نمونه در شکل (۵-۱۳) شبکه متعامد، گراف کامل و گرافی با SRT-2 برای گراف خنثی با ۶ گره نمایش داده شده است.



شکل (۵-۱۳) نمایش شبکه متعامد، گراف کامل و گراف با SRT-2

توپولوژی گراف پایه تاثیر بسزایی در وزن بهینه دارد. از نقطه نظر طراحی نیز، اعضای با طول بلند غیر اقتصادی هستند. ملاحظه می شود در حالاتی که اعضای با طول بلند از گراف کامل حذف می شوند، وزن کمتری برای سازه بهینه حاصل می شود. انتخاب گراف پایه با اعضای بیشتر، علاوه بر آنکه زمان محاسبات کامپیوتری را افزایش می دهد و در پروسه بهینه یابی اشکال پیچیده و غیر عملی بدست می آورد، طراح را از توپولوژی بهینه دور می کند. طی بررسی هایی که به عمل آمد، مشاهده شد انتخاب گراف پایه ستاره ای معمولاً نتایج خوبی را بدست می دهد. در این گراف پایه، هر گره فقط به گره های اطراف خود متصل می باشد.

به عنوان مثال در شکل (۵-۱۴) گراف پایه ستاره ای برای شبکه متعامد ۱۵ گرهی نشان داده شده است.



شکل (۵-۱۴) گراف پایه ستاره ای برای گراف خنثی ۱۵ گره ای

در این شکل تعداد اعضا برابر ۳۸ است که نسبت به حالت گراف کامل (۱۰۵ عضو) خیلی کمتر می باشد. این مساله سبب می شود که طول هر کروموزوم کاهش و بالطبع سرعت همگرایی افزایش و احتمال حصول بهینه محلی کاهش یابد.

### ۵-۳-۵ تولید جمعیت اولیه با توجه به گراف پایه

در تولید جمعیت اولیه بمنظور بهینه یابی توپولوژیکی فریابها، متغیرهای طراحی، سطح مقطع اعضا و معمولاً بیت های بولی توپولوژیکی مربوط به حضور یا عدم حضور اعضا در نظر گرفته می شود. در این رساله مکان بیت های توپولوژیکی را در سمت راست زیر رشته متغیر طراحی سطح مقطع در نظر گرفته می شوند و عدم حضور عضو را در قالب صفر شدن مقدار بیت بولی توپولوژیکی و یا مقدار صفر برای متغیر طراحی سطح مقطع ( در صورتیکه کلیه بیتها صفر شوند ) تفسیر می شود.

طبیعتاً در خلال نسل سازیها، اپراتورهای پیوند و جهش سبب می شوند که با احتمال کمتر برخی از اعضای حذفی مجدداً در داخل سازه شرکت داده شده و برخی از اعضای موجود نیز کنار گذاشته شوند. این مساله سبب خواهد شد تا فضای وسیع طراحی در محدوده ای منطقی کاوش و در عین حال کاستی های فرضیات انجام شده در تولید جمعیت اولیه مرتفع شود.

### ۵-۳-۶ تعیین مقادیر متغیرهای طراحی سطح مقطع در عملیات GA

همانگونه که در بخش (۳-۴) فصل ۳ عنوان گردید با استفاده از روابط (۵-۱) الی (۵-۵) می توان میزان نقض هر عضو در پاسخ به بارهای خارجی را تعیین نمود. به این ترتیب برای سازه ای تحت اثر چند وضعیت بارگذاری استاتیکی، nlc ، میزان نقض کلی عضو ام از رابطه (۵-۱۸) حاصل می شود :



$$V_i = \sum^{nlc} (v_{i,\sigma} + v_{i,bk} + v_{i,\Delta}) \quad ; \quad i = 1, \dots, ne \quad (5-18)$$

$V_i > 0$  بیانگر ضعف عضو  $i$  ام در پاسخ به بارهای خارجی است. در صورتیکه اعضای سازه گروه بندی شده باشند  $V_i$  برای کلیه اعضای هر گروه با یکدیگر جمع می شود.

در حالتی که مقاطع پیوسته باشند، با استفاده از رابطه (۵-۱۹) طول زیر رشته متغیر طراحی سطح مقطع پیوسته  $i$  ام،  $l_{bi}$ ، بدست می آید.

$$l_{bi} = \text{INT} \left[ 1 + \frac{\log \left( \frac{A_i^u - A_i^l}{\epsilon_i} \right)}{\log 2} \right] \quad ; \quad i = 1, \dots, ns \quad (5-19)$$

در رابطه فوق  $A_i^u$ ،  $A_i^l$  و  $\epsilon_i$  به ترتیب کرانه فوقانی، کرانه تحتانی و دقت مه، د نیاز برای متغیر طراحی سطح مقطع پیوسته  $i$  ام می باشد. اینک با استفاده از رابطه (۵-۲۰) می توان سطح مقطع متغیر طراحی پیوسته  $i$  ام یعنی  $A_i$  را در خلال عملیات GA ارزیابی نمود.

$$A_i = A_i^l + I_i \frac{A_i^u - A_i^l}{2^{l_{bi}} - 1} \quad ; \quad i = 1, \dots, ns \quad (5-20)$$

در رابطه فوق  $I_i$  مقدار صحیح زیر رشته  $i$  ام با طول  $l_{bi}$  پس از رمزگشایی است.

### ۵-۳-۷ تابع جریمه و تابع شایستگی مورد استفاده

همانگونه که قبلاً اشاره شد، یکی از مشکلات GA تعیین توابع مناسب و مقادیر پارامترهای دخیل در آن است. در این فصل با بکارگیری تعریف تابع جریمه رابطه (۲-۱) و استفاده از مفهوم تابع شایستگی (۲-۴) در فصل مبانی الگوریتم ژنتیک مورد بررسی قرار گرفت بهبود چشمگیری در عملیات GA بوجود می آید.

از فواید این کار به موارد ذیل می‌توان اشاره کرد:

الف- ممانعت از سمت‌گیری GA به سمت بهینه محلی.

ب- قابلیت کاوش هرچه بهتر فضای طراحی و شتاب بخشیدن به خاصیت تکاملی GA.

پ- کاهش وابستگی GA به تعاریف خاصی از تابع شایستگی، تابع جریمه و پارامترهای دخیل در آن.

ت- امکان استفاده از کامپیوترهای موازی در این روش و بکارگیری قابلیت‌های سوپر کامپیوترها در حل مسائلی که فضای طراحی در آن‌ها بزرگ می‌باشد.

س- امکان استفاده از این روش در بهینه‌سازی و طراحی سازه‌های واقعی که بدلیل بزرگ بودن فضای طراحی با روش‌های الگوریتم ژنتیک متداول قابل حل نمی‌باشد.

معمولاً هدف از بهینه‌یابی سازه‌ها کاهش وزن یا حجم مصالح مصرفی است که به نوعی با کاهش هزینه تمام شده در ارتباط است. در الفبای GA مطلب فوق بصورت حداکثر کردن تابع شایستگی بیان می‌شود. بدین منظور برای تابع شایستگی از روابط زیر که در فصل ۲ نیز اشاره شد، استفاده شده است.

$$F_i = [\phi_{\max}(\mathbf{X}) + \phi_{\min}(\mathbf{X})] - \phi_i(\mathbf{X})$$

$$\phi_i(\mathbf{X}) = f_i(\mathbf{X}) + f_{i,penalty} \quad ; i = 1, \dots, nc \quad (۵-۲۱)$$

در روابط فوق  $f_i(\mathbf{X})$ ،  $f_{i,penalty}$  و  $\phi_i(\mathbf{X})$  به ترتیب تابع هدف، تابع جریمه و تابع هدف اصلاح شده کروموزوم  $i$ ام و  $\phi_{\max}(\mathbf{X})$  و  $\phi_{\min}(\mathbf{X})$  به ترتیب حداکثر و حداقل تابع هدف اصلاح شده در جمعیت جاری است.  $\mathbf{X}$  بردار متغیرهای طراحی است و در صورتیکه تابع هدف وزن سازه در نظر گرفته شود، بردار  $\mathbf{X}$  حاوی متغیرهای طراحی سطح مقطع و متغیرهای بولی توپولوژیکی به تعداد  $ns+ne$  می‌باشد. همچنین چنانچه اعضای سازه مبنا گروه بندی شده باشند، سطح مقطع اعضای هر گروه یکسان در

نظر گرفته می‌شود. با توجه به آنکه برای کلیه اعضای سازه مینا، متغیر بولی توپولوژیکی در نظر گرفته شده است، لذا برخی از اعضای هر گروه می‌توانند حضور داشته باشند (با مقدار بیت یک) و برخی دیگر می‌توانند حضور نداشته باشند (با مقدار بیت صفر).  $F_i$  شایستگی کروموزوم  $i$  ام و  $nc$  تعداد کروموزوم‌های جمعیت (pop-size) می‌باشد.

برای تابع جریمه نیز علاوه بر تعریف مطرح شده بیان شده در فصل تکنیک‌های نوین الگوریتم ژنتیک تعریف ساده‌ای نیز وجود دارد که در اغلب مقالات، در حل مثال‌ها مورد استفاده قرار گرفته است.

$$f_{penalty} = f(\mathbf{X}) \cdot K \cdot C \quad ; \quad C = \sum_{q=1}^Q \max[0, g_q(\mathbf{X})] \quad (22-5)$$

$f_{i,penalty}$  مقدار تابع جریمه کروموزوم  $i$  ام می‌باشد. در این رساله مقادیر بیت‌های توپولوژیکی با توجه به سازه نزدیک به بهینه حاصل از مرحله قبل تعیین می‌شوند. روال کار به این ترتیب است که اعضای حاضر در سازه نزدیک به بهینه با احتمال ۹۰ تا ۹۵ در صد و اعضای حذف شده با احتمال ۵ تا ۱۰ در صد شانس حضور در جمعیت اولیه مرحله بعدی را پیدا می‌کنند.

### ۵-۳-۸ فرآیند انتخاب، مقادیر نرخ پیوند و جهش

در این فصل برای فرآیند انتخاب، روش چرخ رولت و در برخی از مثال‌ها از روش پیشنهادی مذکور در فصل ۲ استفاده شده است. برای پیوند نیز از روش قطع دو نقطه‌ای موثر بر کل طول کروموزوم با نرخ ۰/۹ استفاده شده است. برای متغیرهای طراحی سطح مقطع نرخ جهش ۰/۰۱ تا ۰/۱ و برای متغیرهای توپولوژیکی برای جلوگیری از تبدیل شدن عملیات GA به یک روش کاوش صرفاً تصادفی<sup>۱</sup>، نرخ جهش ۰/۰۰۱ تا ۰/۰۰۵ بکار گرفته شده است.

<sup>۱</sup> Random Search Method

### ۵-۳-۹ مثالهای عددی

در ادامه چهار مثال جهت بررسی عملکرد روشهای مطرح شده و مقایسه نتایج با سایر مقالات مطرح شده است. لیست مقاطع منفصلی که برخی از مثالها از آن استفاده می کنند، در جدول (۵-۱۸) درج شده است.

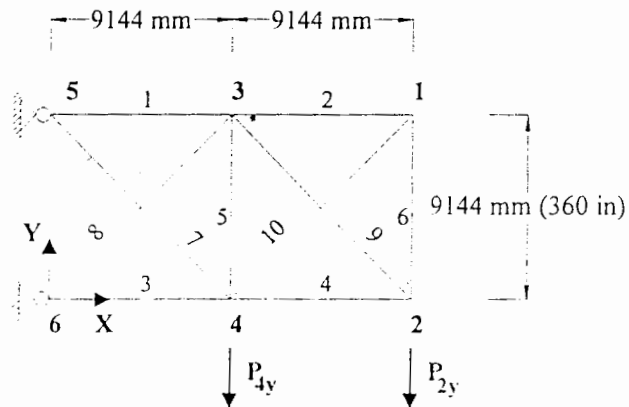
جدول (۵-۱۸) مجموعه مقاطع منفصل مورد استفاده در مثالها

$S_1 = \{ 1.62, 1.8, 2.38, 2.62, 2.88, 3.09, 3.13, 3.38, 3.63, 3.84, 3.87, 4.18, 4.49, 4.80, 4.97, 5.12, 5.74, 7.22, 7.97, 11.5, 13.5, 13.9, 14.2, 15.5, 16.0, 18.8, 19.9, 22.0, 22.9, 26.5, 30.0, 33.5 \} \text{ ( in}^2 \text{)}$
$S_1 = \{ 10.45, 11.61, 15.35, 16.90, 18.58, 19.94, 20.19, 21.81, 23.42, 24.77, 24.97, 26.97, 28.97, 30.97, 32.06, 33.03, 37.03, 46.58, 51.42, 74.19, 87.10, 89.68, 91.61, 100.0, 103.23, 121.29, 128.39, 141.94, 147.74, 170.97, 193.55, 216.13 \} \text{ ( cm}^2 \text{)}$
$S_2 = \{ 0.1, 0.5 \times I \text{ ( } I = 1, 2, \dots, 76 \text{)}, 39.81, 40.0 \} \text{ ( in}^2 \text{)}$
$S_2 = \{ 0.645, 3.226 \times I \text{ ( } I = 1, 2, \dots, 76 \text{)}, 256.84, 258.06 \} \text{ ( cm}^2 \text{)}$
$S_3 = \{ 6.45, 19.35, 32.26, 51.61, 64.51, 67.74, 77.42, 96.77, 109.68, 141.94, 154.84, 167.74, 180.64, 187.1, 200.0, 225.81 \} \text{ ( cm}^2 \text{)}$
$S_4 = \{ 0.01, 20.0, 40.0, 60.0, 80.0 \} \text{ ( cm}^2 \text{)}$
$S_5 = \{ 0.774, 1.255, 2.142, 3.348, 4.065, 4.632, 6.542, 7.742, 9.032, 10.839, 12.671, 14.581, 21.483, 34.839, 44.516, 52.903, 60.258, 65.226 \} \text{ ( cm}^2 \text{)}$

### مثال (۵-۵) خرپای مستوی شش گرهی

با توجه به مطالب بخش ۵-۴-۱، برای خرپای مستوی ۶ گرهی سازه مبنا به صورت گراف پایه ستاره ای با ۱۰ عضو، همانند شکل (۵-۱۴) در نظر گرفته شده است.

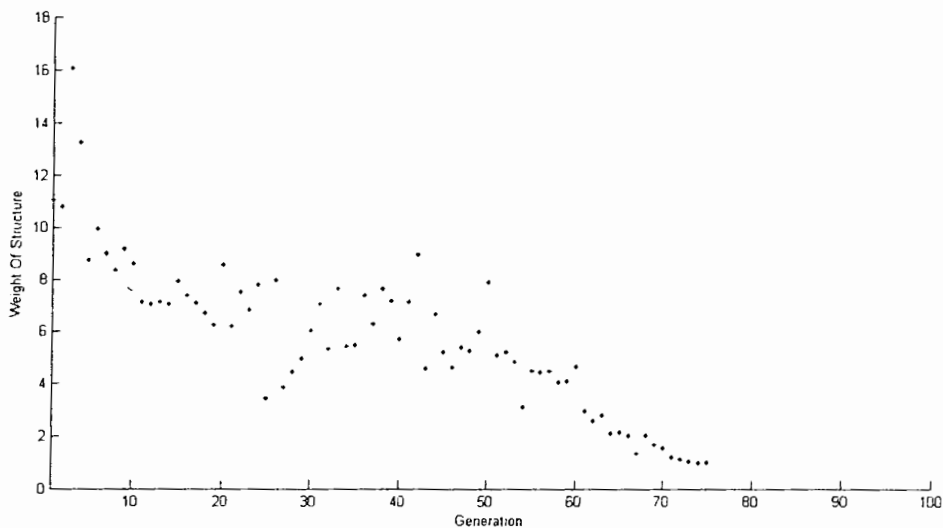
برای هر عضو یک متغیر طراحی سطح مقطع و یک متغیر طراحی توپولوژیکی بولی اختصاص یافته است. در صورتیکه طول زیر رشته متغیرهای طراحی سطح مقطع  $l_{ai} = 4$ ,  $i = 1, \dots, 10$  در نظر گرفته شود، طول هر کروموزوم برابر ۵۰ می شود.



شکل (۵-۱۵) گراف پایه ستاره ای برای خرپای مستوی ۶ گره ای

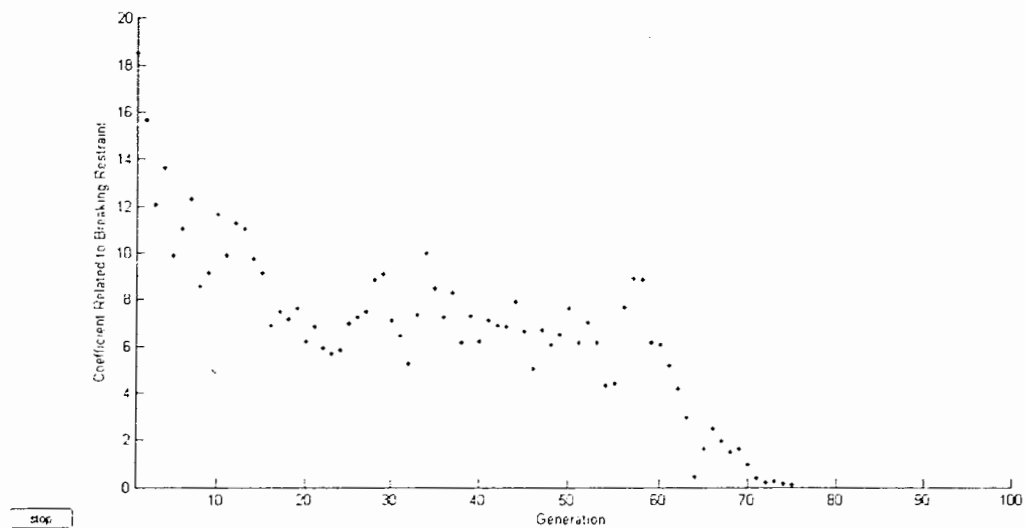
برای کلیه اعضا  $E = 68.97 \text{ GN/m}^2$  ( $10^4 \text{ Ksi}$ )،  $\rho = 27126.4 \text{ N/m}^3$  ( $0.1 \text{ lb/in}^3$ ) و تنش مجاز  $\pm 172.4 \text{ MN/m}^2$  ( $25 \text{ Ksi}$ ) می باشد. بارهای موثر  $P_{2y} = P_{4y} = -445.4 \text{ KN}$  ( $-100 \text{ Kips}$ ) و تغییر مکان مجاز در راستای  $y$  برای کلیه گره های دارای درجه آزادی فعال  $50.8 \text{ mm}$  ( $2 \text{ in}$ ) است. در صورتیکه از لیست مقاطع منفصل  $S_1$  استفاده شود.

در شکل ۵-۱۶ روند بهینه یابی سازه و همگرایی جوابها با استفاده از ترسیم تابع هدف که همان وزن سازه می باشد در مقابل تعداد نسلها مورد بررسی قرار گرفته است.



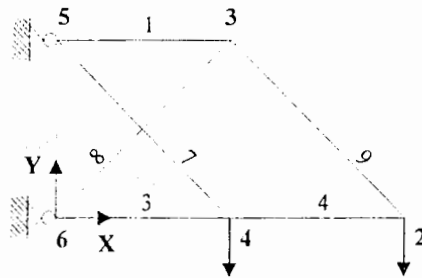
شکل (۵-۱۶) مقادیر تابع هدف (وزن سازه) در نسلهای مختلف

در شکل ۵-۱۷ مقادیر پارامتر  $C_g$  که نشان دهنده جمع مقادیر نقض قیود مربوط به نقض قیود تنش، تغییر مکان و یا کمانش می باشد برحسب پیشرفت نسلها نشان داده شده است. لازم بذکر است که به منظور اطمینان از عدم نقض قیود جواب های بهینه نهایی این سازه ها با استفاده از نرم افزار تحلیلی Etabs V8.31 مدل سازی گردید و با اختصاص مقاطع بدست آمده از بهینه سازی سازه با روش پیشنهادی سازه کنترل گردید تا نقض قیدی در آن وجود نداشته باشد.

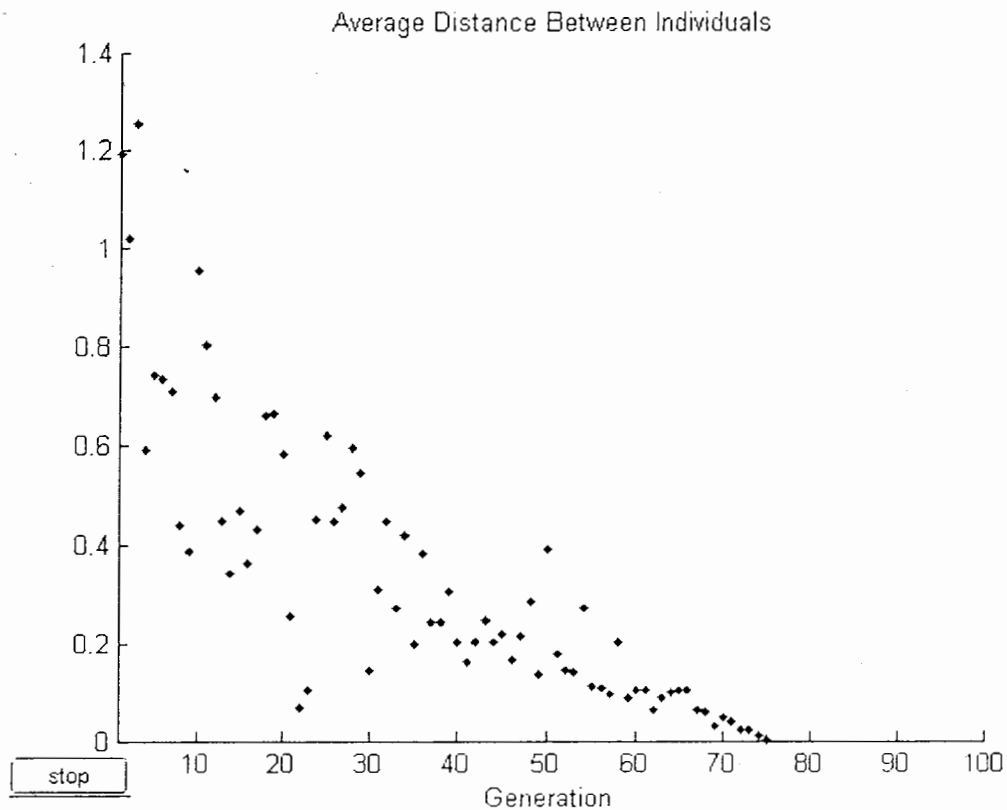


شکل (۵-۱۷) مقادیر نقض قیود در نسلهای مختلف

در شکل (۵-۱۸) توپولوژی بهینه و در جدول (۵-۱۸) نتایج آن نمایش داده شده است.



شکل (۵-۱۸) توپولوژی بهینه برای گراف پایه ستاره ای ۱۰ میله ای



شکل (۵-۱۹) میانگین فواصل مابین اعضای جمعیت و روند همگرایی برابها

در کسب نتایج توپولوژی بهینه در صورتیکه مقاطع از لیست  $S_1$  انتخاب شده باشند، نرخ پیوند  $0.09$ ، نرخ جهش برای متغیرهای طراحی سطح مقطع  $0.1$ ، نرخ جهش برای متغیرهای توپولوژیکی  $0.001$ ، تعداد نسل سازی در هر مرحله  $75$  و تعداد اعضای جمعیت،  $pop-size = 50$  در نظر گرفته شده است. توپولوژی نسل شصت و دوم حاصل شده است. در طول عملیات GA حدود  $35$  درصد کروموزوم ها، ناپایدار کننده سازه بوده اند که از میان آنها حدود  $90$  درصدشان ناپایدار کننده توپولوژیکی سازه بوده و به آسانی و مستقل از هر گونه عملیات تحلیل سازه توسط رابطه (۵-۱۹) شناسایی و به شدت جریمه شده اند. لازم بذکر است که استفاده از روش این رساله علاوه بر حصول نتایج بسیار خوب و افزایش سرعت همگرایی الگوریتم به سمت جواب های بهینه تا حد مطلوبی از خواص روش مرحله ای GA بشرحی که در بخشهای قبل اشاره شد نیز بهره می برد.

نتایج جدول (۲-۵) به ازای مجموعه مقاطع  $S_1$  با نتایج حاصل در مرجع [30] یکسان است. همچنین برای بررسی تاثیر گذاری مجموعه مقاطع منفصل انتخابی و یا مقاطع پیوسته بر روی طرح بهینه نهایی، در جدول (۲-۵) نتایج توپولوژی بهینه در حالتیکه مقاطع از لیست  $S_2$  انتخاب شده باشند و یا پیوسته با کرانه تحتانی، کرانه فوقانی و دقت بترتیب  $1/0$ ،  $34/0$  و  $0/01$  باشند، نمایش داده شده است. ملاحظه می شود در دو وضعیت اخیر توپولوژی بهینه نظیر شکل (۲۰-۵) ولی با وزن کمتر حاصل شده است. مطالعه بر روی سایر گرافهای پایه نیز توپولوژی بهینه ای نظیر شکل (۲۰-۵) نتیجه می دهد.

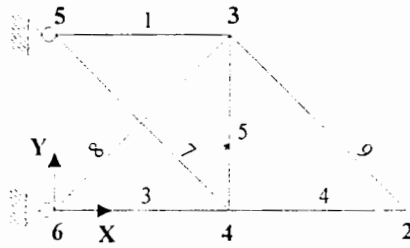
جدول (۱۹-۵) نتایج سطح مقطع و توپولوژی بهینه خرابای مستوی ۶ گرهی  
منتخب از لیست مقاطع  $S_1$  و  $S_2$

وزن (lb) N	سطح مقطع اعضا. $(in^2) cm^2$										مجموعه مقاطع منفصل انتخابی
	$A^{10}$	$A^9$	$A^8$	$A^7$	$A^6$	$A^5$	$A^4$	$A^3$	$A^2$	$A^1$	
22058 (4962.1)	-	141.94 (22.0)	141.94 (22.0)	46.58 (7.22)	-	-	100.0 (15.5)	128.39 (19.9)	-	193.55 (30.0)	$S_1$
21825 (4909.8)	-	135.48 (21.0)	135.48 (21.0)	38.71 (6.0)	-	-	100.0 (15.5)	154.84 (24.0)	-	187.1 (29.0)	$S_2$
21765 (4896.0)	-	139.35 (21.6)	137.42 (21.3)	39.42 (6.11)	-	-	96.13 (14.9)	140.64 (21.8)	-	193.55 (30.0)	مقاطع پیوسته

جدول (۲۰-۵) نتایج سطح مقطع و توپولوژی بهینه برای خرابای مستوی شش گرهی  
منتخب از لیست مقاطع  $S_3$  (اگر  $P_{4y}$  و  $P_{2y}$  بصورت دو وضعیت بار فرض شوند)

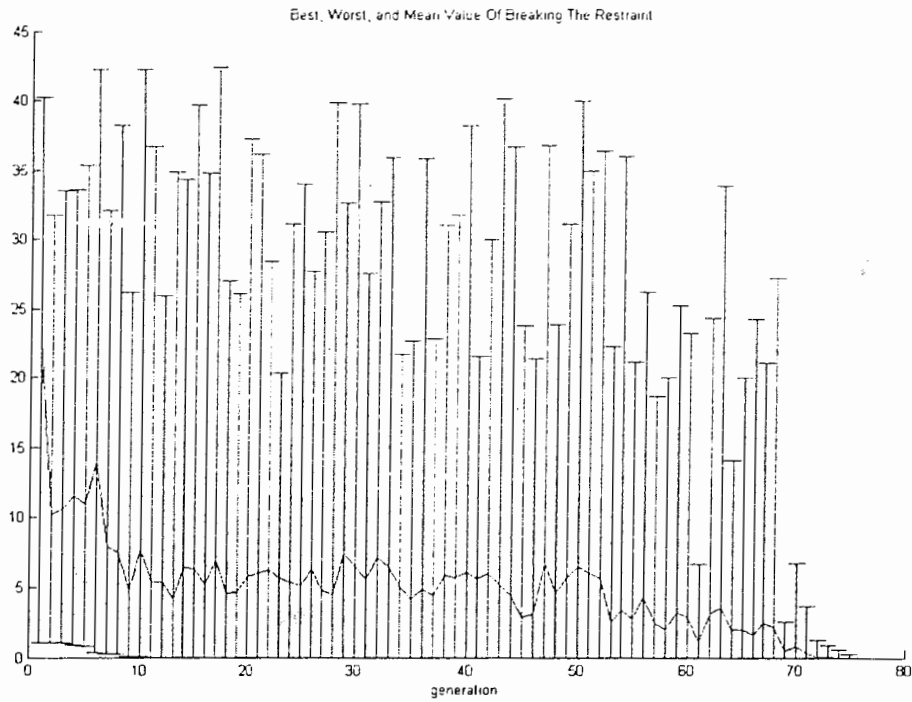
وزن N	سطح مقطع اعضا. $cm^2$										نتایج طرح بهینه
	$A^{10}$	$A^9$	$A^8$	$A^7$	$A^6$	$A^5$	$A^4$	$A^3$	$A^2$	$A^1$	
19712	32.26	109.68	109.68	32.26	19.35	-	77.42	109.68	19.35	167.74	chai, shi and sun [43]
19267	-	141.94	109.68	19.35	-	19.35	109.68	96.77	-	167.74	بر اساس روشهای این رساله





شکل (۵-۲۰) توپولوژی بهینه خرپای ۶ گرهی

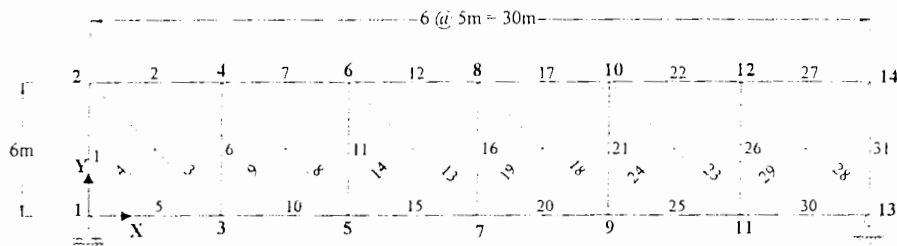
(اگر  $P_{2y}$  و  $P_{4y}$  بصورت دو وضعیت بار بر سازه اثر کند)



شکل (۵-۲۱) مقادیر نقض قیود در نسلهای مختلف

## مثال (۵-۵) خرپای مستوی ۱۴ گرهی

شکل (۵-۲۲) یک خرپای مستوی ۱۴ گرهی را نشان می‌دهد که تحت اثر بار متحرک متمرکز به مقدار  $3 \times 10^6 \text{ N}$  قرار دارد و به نوبت به گره های ۳، ۵، ۷، ۹ و ۱۱ اثر می‌کند. به عبارت دیگر در مجموع پنج وضعیت بار وجود دارد. برای این خرپا نیز سازه مبنا بصورت گراف پایه ستاره ای در نظر گرفته شده است.



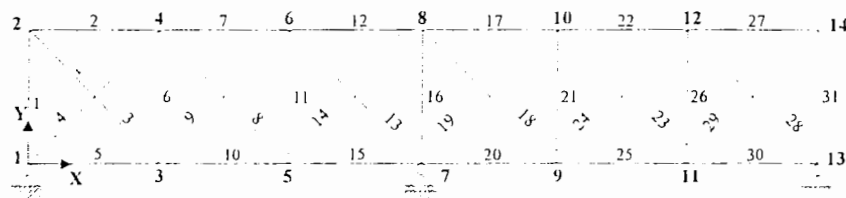
شکل (۵-۲۲) سازه مبنا برای خرپای ۱۴ گرهی (گراف پایه ستاره ای ۳۱ عضوی)

برای کلیه اعضا  $E = 210.0 \text{ GN/m}^2$ ،  $\sigma_t = 130.0 \text{ MN/m}^2$ ،  $|\sigma_c| = 104.0 \text{ MN/m}^2$  و حداکثر تغییر مکان مجاز در جهت  $y$ ،  $|\Delta_{ay}| = 0.01 \text{ m}$  فرض شده است. مقاطع پیوسته با کرانه تحتانی، فوقانی و دقت بترتیب  $0.01 \text{ m}^2$ ،  $0.1 \text{ m}^2$  و  $0.01 \text{ m}^2$  در نظر گرفته شده است. سطح مقطع اعضا همانند جدول (۵-۲۱) به چهار گروه دسته بندی شده اند و به کلیه اعضا به استثنای اعضای یال تحتانی خرپا، متغیر طراحی بولی توپولوژیکی اختصاص یافته است.

جدول (۵-۲۱) گروه بندی اعضای خرپای ۱۴ گرهی

شماره اعضا	شماره گروه
2,7,12,17,22,27	1
5,10,15,20,25,30	2
3,4,8,9,13,14,18,19,23,24,28,29	3
1,6,11,16,21,26,31	4

همچنین حضور یا عدم حضور تکیه گاه در جهت  $x$  گره ۱ و جهت  $y$  گره ۷ بعنوان دو متغیر طراحی بولی در نظر گرفته شده است. به این ترتیب طول هر رشته کروموزوم در شروع عملیات GA در مرحله اول برابر ۴۳ حاصل می‌شود. تابع هدف حداقل کردن حجم سازه است. در شکل (۵-۷) توپولوژی بهینه و در جدول (۵-۵) نتایج آن به همراه مقایسه نتایج با مرجع [30] نمایش داده شده است.



شکل (۵-۲۲) توپولوژی بهینه خرپای مستوی ۱۴ گرهی

جدول (۵-۲۲) نتایج سطح مقطع و توپولوژی بهینه خرپای مستوی ۱۴ گرهی

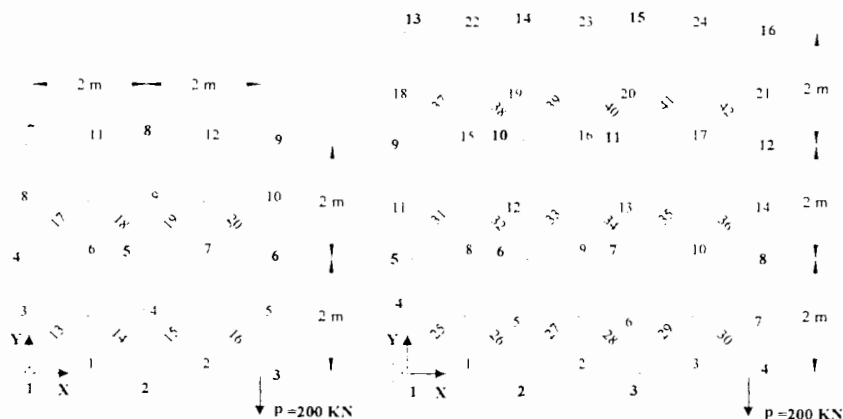
حجم $m^3$	وضعیت متغیرهای طراحی مربوط به درجات آزادی		سطح مقطع اعضای گروه ها، $m^2$				نتایج طرح بهینه
	گره ۷ در جهت $y$	گره ۱ در جهت $x$	$A_4$	$A_3$	$A_2$	$A_1$	
2.929	بسته	بسته	0.034	0.034	0.01	0.022	Rajan [30]
2.625	بسته	بسته	0.01	0.0154	0.01	0.0154	بر اساس روشهای این رساله

همانگونه که در شکل (۵-۲۲) مشاهده می‌شود در توپولوژی بهینه فقط درجات آزادی گره ۱ در جهت  $x$  و گره ۷ در جهت  $y$  بسته شده است و شکل سازه مینا بدون تغییر باقی مانده است. در حالیکه در مقاله راجان اعضای ۲، ۱۲، ۱۷، ۲۷، ۳، ۹، ۱۴، ۱۸، ۲۳، ۲۹، ۱، ۱۶، ۳۱ از سازه مینا حذف شده اند. در کسب نتایج بهینه نرخ پیوند ۰/۹، نرخ جهش برای متغیرهای طراحی سطح مقطع ۰/۱۰، نرخ جهش برای متغیرهای طراحی توپولوژیکی ۰/۰۰۱، تعداد نسل سازی در هر مرحله ۲۰ و

تعداد اعضای جمعیت  $pop-size=50$  در نظر گرفته شده است. طرح بهینه در مرحله پنجم به ازای  $2/79$  حاصل شده است. با اطلاعات فوق حدود  $4/5$  در صد از کروموزوم ها ناپایدار کننده سازه بوده اند.

### مثال (۵-۶) شبکه $2 \times 2$ و $3 \times 3$ خرپای مستوی

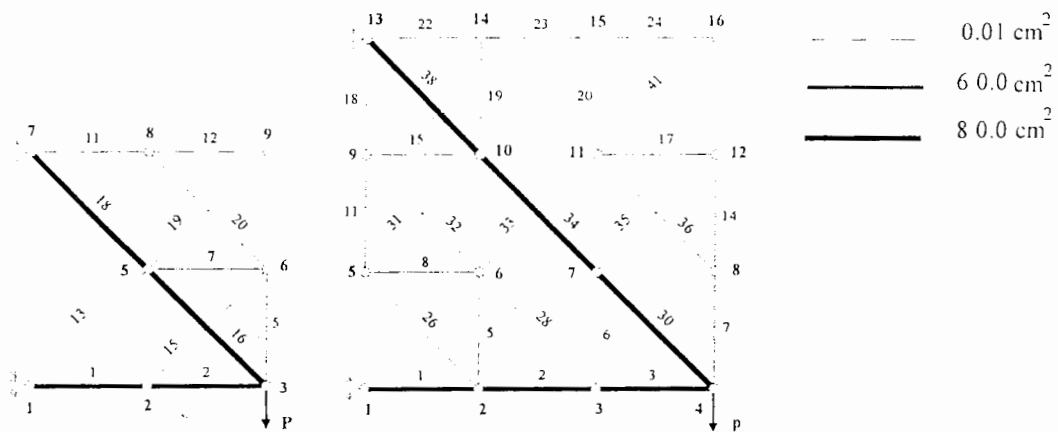
جهت بررسی ارتباط شکل توپولوژی بهینه با مسیرهای بحرانی انتقال نیرو و کنترل عملکرد الگوریتم در بخش (۵-۴-۲)، این مثال از مرجع [44] انتخاب شده است. در شکل (۵-۸) شبکه های  $2 \times 2$  و  $3 \times 3$  خرپای مستوی تحت اثر بار متمرکز  $P=200 \text{ KN}$  نمایش داده شده است. هدف حداقل کردن حجم سازه به ازای ارضای فقط قید تنش با مقدار تنش مجاز کششی و فشاری  $\pm 40/0 \text{ Mpa}$  است.



شکل (۵-۲۳) شبکه های  $2 \times 2$  و  $3 \times 3$  خرپای مستوی

مقاطع از لیست  $S_4$  انتخاب شده و مقدار اول آن سطح مقطع حداقل  $0/01 \text{ cm}^2$  در نظر گرفته شده است. آزادی اعضای با سطح مقطع  $0/01 \text{ cm}^2$  را حذف شده تلقی و قید تنش را برای آنها اعمال نمی کند. در عوض در روند بهینه یابی برای جلوگیری از تمایل به سمت اختیار سطح مقطع حداقل  $0/01 \text{ cm}^2$  برای کلیه اعضا جهت حداقل شدن حجم، یک کرانه فوقانی مجازی را برای تغییر مکانهای

تغییر مکانهای گرهی در نظر گرفته است. در این رساله به هر عضو یک متغیر طراحی سطح مقطع و یک متغیر بولی توپولوژیکی اختصاص یافته و برای جلوگیری از ناپایداری یا منفصل شدن توپولوژی بهینه بدون هر گونه فرض جنبی، قید تنش در مورد اعضای با سطح مقطع حداقل  $0.01 \text{ cm}^2$  نیز کنترل شده است. باین ترتیب اعضای منتخب برای حذف با مقدار بیت توپولوژیکی صفرکلاً از سازه کنار گذاشته شده و سازه باقیمانده مورد آنالیز و ارزیابی شایستگی قرار می گیرد. با توجه به اینکه لیست  $S_4$  حاوی ۵ سطح مقطع است، برای کلیه متغیرهای طراحی سطح مقطع مقدار  $a_{ii} = 2$  در کل مراحل GA در نظر گرفته شده است. شکل (۲۴-۵) توپولوژی بهینه و جدول (۲۱-۵) نتایج آنرا نمایش می دهد. همانگونه که در شکل (۲۴-۵) ملاحظه می شود ضخامت خطوط به ترتیب مقادیر سطح مقطع اعضا رسم شده است.

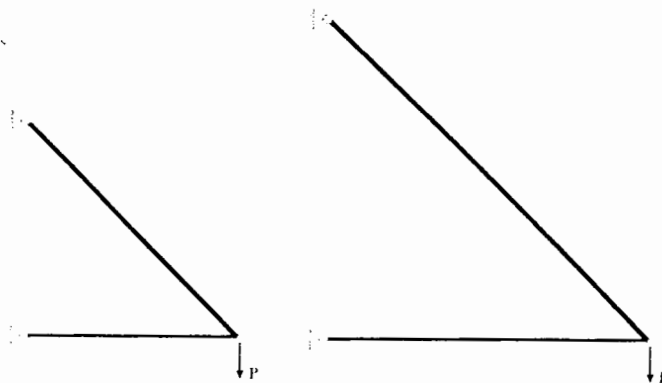


شکل (۲۴-۵) توپولوژی بهینه شبکه های  $2 \times 2$  و  $3 \times 3$  خرابای مستوی

جدول (۲۳-۵) نتایج سطح مقطع و توپولوژی بهینه شبکه های  $2 \times 2$  و  $3 \times 3$  خرابای مستوی

حجم، $m^3$	سطح مقطع و شماره اعضای توپولوژی بهینه			نتایج طرح بهینه
	$80.0 \text{ cm}^2$	$60.0 \text{ cm}^2$	$0.01 \text{ cm}^2$	
0.06927	16,18	1,2	5,7,11,12,13, 15,19,20	شبکه $2 \times 2$ خرابای مستوی
0.10393	30,34,38	1,2,3	5,6,7,8,11,14,15,17, 18,19,20,22,23,24,26 28,31,32,33,35,36,41	شبکه $3 \times 3$ خرابای مستوی

در صورتیکه اعضای با سطح مقطع کوچک  $0.1 \text{ cm}^2$  حذف و مفصل های بین اعضای اصلی باقیمانده برای حفظ پایداری سازه نیز حذف شوند، شکل (۵-۱۰) حاصل می شود که کاملاً متناسب با مسیرهای بحرانی انتقال بار است. حجم های بهینه حاصل از روش های پیشنهادی این فصل مندرج در جدول (۵-۲۳) همانند نتایجی است که در مرجع [44] بدست آمده است. در کسب توپولوژی بهینه شبکه های  $2 \times 2$  و  $3 \times 3$  نرخ پیوند  $0.19$ ، نرخ جهش متغیرهای طراحی سطح مقطع  $0.05$ ، نرخ جهش متغیرهای بولی توپولوژیکی  $0.001$  و تعداد نسل سازی در هر مرحله  $20$  منظور شده است.



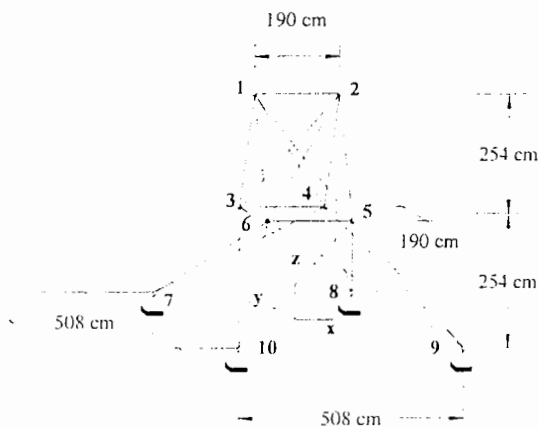
شکل (۵-۲۵) توپولوژی بهینه اصلاح شده شبکه های  $2 \times 2$  و  $3 \times 3$  خرپای مستوی

در شبکه  $2 \times 2$  به ازای تعداد اعضای جمعیت برابر  $100$ ،  $K = 2/75$ ، جواب بهینه در نسل هفتم حاصل شده است. همچنین در شبکه  $3 \times 3$  به ازای تعداد اعضای جمعیت برابر  $200$ ،  $K = 4/77$ ، جواب بهینه در مرحله دهم نسل شانزدهم حاصل شده است.

### مثال (۵-۶) خرپای فضایی ۱۰ گرهی

در شکل (۵-۲۵) خرپای فضایی ۲۵ میله ای با ۱۰ گره نمایش داده شده است. کلیه اعضا دارای  $E = 68/97 \text{ GN/m}^2$  و  $\rho = 27126/4 \text{ N/m}^3$  می باشد. اعضا همانند جدول (۵-۷) به هشت گروه

دسته بندی شده و تنش های مجاز کششی و فشاری هر گروه در آن درج شده است. تغییر مکان مجاز گره های ۱ و ۲ در جهات x و y برابر  $\pm 0.1889 \text{ cm}$ ، مقاطع از لیست  $S_5$  انتخاب و بارهای موثر بر سازه در جدول (۵-۲۴) نمایش داده شده است.



شکل (۵-۲۶) خرپای فضایی ۱۰ گرهی

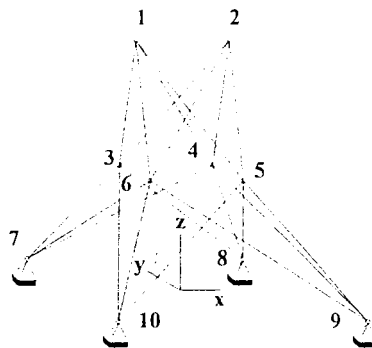
جدول (۵-۲۴) گروه بندی اعضا و تنش های مجاز کششی و فشاری اعضای هر گروه

شماره گروه	شماره گره دو انتهای اعضای هر گروه	تنش مجاز فشاری، $ \sigma_c $ $\text{N/cm}^2$	تنش مجاز کششی، $\sigma_t$ $\text{N/cm}^2$
1	1-2	24204	27590
2	1-4, 2-3, 1-5, 2-6	7994	27590
3	2-5, 2-4, 1-3, 1-6	11936	27590
4	3-6, 4-5	24204	27590
5	3-4, 5-6	24204	27590
6	3-10, 6-7, 4-9, 5-8	4662	27590
7	3-8, 4-7, 6-9, 5-10	4662	27590
8	3-7, 4-8, 5-9, 6-10	7664	27590

جدول (۲۵-۵) بارگذاری موثر بر خرپای فضایی ۱۰ گرهی

وضعیت بار	شماره گره	$(KN), P_x$	$(KN), P_y$	$(KN), P_z$
1	1	4.45	44.5	-22.25
	2	0.0	44.5	-22.25
	3	2.225	0.0	0.0
	6	2.225	0.0	0.0
2	1	0	89	-22.25
	2	0	-89	-22.25

در مرحله اول عملیات GA، مقدار  $l_{ai} = 4$  ;  $i = 1, \dots, 8$  در نظر گرفته شده و به هر عضو یک متغیر طراحی بولی توپولوژیکی اختصاص یافته است. به این ترتیب طول هر کروموزوم در عملیات GA برابر ۵۷ می‌باشد. توپولوژی بهینه در شکل (۲۶-۵) و نتایج آن در جدول (۲۵-۵) نمایش داده شده است. در کسب نتایج بهینه نرخ پیوند ۰/۹، نرخ جهش برای متغیرهای طراحی سطح مقطع و متغیرهای بولی توپولوژیکی ۰/۰۵، تعداد نسل سازی در هر مرحله ۱۰۰، تعداد اعضای جمعیت برابر ۵۰ در نظر گرفته شده است حدود ۷ در صد کروموزوم ها ناپایدار و از میان آنها حدود ۵۲ در صدشان دارای ناپایداری توپولوژیکی است که به آسانی توسط رابطه (۱۹-۵) شناسایی شده است.



شکل (۲۷-۵) توپولوژی بهینه خرپای فضایی ۱۰ گرهی



جدول (۵-۲۶) نتایج مقاطع توپولوژی بهینه خرپای فضایی ۱۰ گرهی

وزن، N	سطح مقطع گروه های اعضا، $cm^2$								نتایج طرح بهینه
	A <sub>8</sub>	A <sub>7</sub>	A <sub>6</sub>	A <sub>5</sub>	A <sub>4</sub>	A <sub>3</sub>	A <sub>2</sub>	A <sub>1</sub>	
2733.5	21.483	12.671	3.348	0	0	21.483	12.671	0	Chai, Shi and Sun [43]
2517.24	14.581	12.671	6.542	-	-	21.483	10.839	-	بر اساس روشهای این رساله

### ۵-۳-۱۰ نتیجه گیری

در این فصل بهینه یابی مقاطع و توپولوژی خرپاها با استفاده از مفاهیم روش سختی و الگوریتم ژنتیک

مورد بررسی قرار گرفت. فهرست کارهای انجام شده و نتایج حاصل بشرح ذیل می باشد :

۱- نتایج توپولوژی بهینه به سازه مبنا با تعداد گره های مفروض بستگی دارد. گاهی استفاده از

سازه مبنای پر عضو به نحوی که بین هر دو گره، عضوی در نظر گرفته شده باشد ( گراف

کامل ) سبب می شود تا طول کروموزوم ها زیاد و در نتیجه سرعت همگرایی کند و بهینه

محلی حاصل شود. با توجه به آنکه مدل ریاضی هر سازه اسکلتی توسط نظریه گرافها قابل

نمایش است، در این فصل برای سازه های میله ای رایج، گراف پایه ستاره ای بعنوان سازه

مبنای مناسب پیشنهاد گردید.

۲- به کلیه اعضای سازه مبنا متغیر طراحی بولی توپولوژیکی اختصاص یافت. در روند بهینه یابی

در صورتیکه مقدار بیت توپولوژیکی صفر شود، عضو مربوطه از سازه مبنا حذف و تحلیل سازه

و ارزیابی شایستگی برای سازه باقیمانده ( با بیتهای توپولوژیکی یک برای اعضای آن ) صورت

می پذیرد. به این ترتیب پایداری توپولوژی بهینه تضمین خواهد شد.

۳- با بررسی مثالها مشخص گردید که در بهینه یابی مقاطع و توپولوژی خرپاها با استفاده از GA،

بین ۱۰ الی ۳۰ درصد کروموزوم ها ناپایدار و از بین آنها ۵۰ الی ۹۰ درصدشان ناپایدار

توپولوژیکی هستند. لذا در این رساله با استفاده از رابطه مربوط به درجات نامعینی استاتیکی سازه و یا با محاسبه دترمینان ماتریس سختی، ناپایداریهای توپولوژیکی به آسانی شناسایی و کروموزوم‌های مربوطه بشدت جریمه شده‌اند.

## ۴-۵ بهینه‌سازی مقطع هندسه و توپولوژی سازه‌های خرپایی

### ۴-۵-۱ مقدمه

همانطور که پیشتر نیز ذکر شد، طرح بهینه سیستم‌های سازه‌ای از جمله خرپاها به طرح بهینه مقاطع<sup>۱</sup>، طرح بهینه شکل<sup>۲</sup>، طرح بهینه توپولوژی<sup>۳</sup> و یا ترکیبی از آنها دسته‌بندی می‌شود. در این قسمت ما به بهینه‌سازی همزمان مقاطع، هندسه و توپولوژی سازه‌های خرپایی می‌پردازیم و ضمن بیان تکنیک‌های نوین در الگوریتم ژنتیک و روش‌های پیشنهادی مراجع به حل مثالهایی در حالت‌های مختلف پرداخته و به مقایسه نتایج و نمودارهای مربوطه با مراجع مختلف می‌پردازیم.

همانطور که می‌دانیم، در مسائل واقعی طراحی سازه معمولاً ما با فضای طراحی بزرگ، تعداد متغیرهای زیاد و قیود متفاوتی دست به‌گریبان هستیم. استفاده از الگوریتم‌های جستجوی تصادفی از جمله الگوریتم ژنتیک در حل این مسائل بدلیل عدم کاوش مؤثر فضای طراحی، تولید کروموزوم‌های مرده به علت وجود قیود متفاوت و افتادن در دام بهینه‌های محلی و معمولاً کارایی چندانی نداشته و تاکنون کمتر طراحی به حل این گونه مسائل با روش الگوریتم ژنتیک پرداخته است.

برای رفع این مشکل، ما با بیان تکنیک‌های نوین الگوریتم ژنتیک مطرح شده در فصل سوم، از جمله استفاده از مفهوم الگوریتم ژنتیک توزیعی (DGA) و تقسیم فضای طراحی به چندین زیر‌فضا و اعمال الگوریتم ژنتیک در هرکدام از این زیر‌فضاها به‌طور جداگانه و همچنین تعاریف توابع جریمه و توابع شایستگی پیشنهادی مراجع و بکارگیری روش‌های مختلف برای عملگرهای انتخاب، پیوند و

<sup>1</sup> Sizing Optimal Design

<sup>2</sup> Shape Optimal Design

<sup>3</sup> Topology Optimal Design

حتی جهش گام بزرگی را در حل این دست مسائل برداشته ایم، که با مقایسه نتایج بدست آمده توسط این روش های مطرح شده با مراجع مختلف، علاوه بر بهبود نتایج شاهد افزایش چشمگیر سرعت همگرایی الگوریتم به سمت جوابهای بهینه هستیم که این امر خود بدلیل کاهش بهتر فضای طراحی می باشد.

### ۵-۴-۲ فرمولبندی مساله بهینه یابی مقاطع، هندسه و توپولوژی خریابا

فرمولبندی مساله بهینه یابی مقاطع منفصل سازه های میله ای (خریابا) با استفاده از GA

بصورت زیر تعریف می شود :

پیدا کنید :

$$\mathbf{X} = [x_1, x_2, \dots, x_{ns}]^T ; x_s \in S ; s = 1, \dots, ns \quad (23-5)$$

تا کمینه شود :

$$\phi(\mathbf{X}) = f(\mathbf{X}) + f_{penalty} \quad (24-5)$$

در معرض قیود زیر :

$$g_i(\mathbf{X}) = \left| \frac{\sigma_i}{\sigma_a} \right| - 1 \leq 0 ; i = 1, \dots, ne \quad (25-5)$$

$$g_j(\mathbf{X}) = \left| \frac{\Delta_j}{\Delta_a} \right| - 1 \leq 0 ; j = 1, \dots, ndof \quad (26-5)$$

$$\begin{cases} g_k(\mathbf{X}) = \left| \frac{\sigma_k}{\sigma_{cr}} \right| - 1 \leq 0 ; k = 1, \dots, mc \\ \sigma_{cr} = \frac{\pi^2 E \alpha_k X_k}{l_k^2} \end{cases} \quad (27-5)$$

در روابط فوق  $\mathbf{X}$  بردار متغیرهای طراحی،  $X_s$  سطح مقطع عرضی گروه  $S$  ام متعلق به لیست پروفیل های موجود<sup>۱</sup>  $S$ ،  $ns$  تعداد متغیرهای طراحی سطح مقطع به تعداد گروه بندی اعضا<sup>۲</sup>،  $f(\mathbf{X})$  تابع هدف که معمولاً وزن یا حجم سازه است،  $f_{penalty}$  تابع جریمه طرح های غیر قابل قبول<sup>۳</sup> مربوط به نقض قیود پاسخ سازه،  $\phi(\mathbf{X})$  تابع هدف اصلاح شده،  $\sigma_i$  تنش محوری عضو  $i$  ام،  $\sigma_a$  تنش مجاز

<sup>1</sup>Unacceptable Design

<sup>2</sup> Encoding

<sup>3</sup> Decoding

محوری، تعداد اعضای خرپا،  $\Delta_z$  تغییر مکان درجه آزادی زام،  $\Delta_a$  تغییر مکان مجاز، ndof تعداد درجات آزادی فعال گره های خرپا،  $\sigma_k$  تنش فشاری عضو k ام،  $X_k$  سطح مقطع عرضی عضو k ام متعلق به گروه S ام،  $l_k$  طول عضو k ام، mc تعداد اعضای فشاری،  $\alpha_k$  ضریب کمانش وابسته به شکل سطح مقطع عرضی عضو k ام ( $I_k = \alpha_k \cdot X_k^2$ )، و  $g_i(X)$ ،  $g_j(X)$  و  $g_k(X)$  به ترتیب قيود تنش محوری، تغییر مکان و کمانش اوپلری می باشند.

اگر تابع هدف وزن سازه باشد، بصورت زیر تعریف می شود:

$$f(\mathbf{X}) = \sum_{i=1}^{ne} X_i l_i \rho_i \quad (28-3)$$

که در آن  $\rho_i$  وزن مخصوص مصالح عضو i ام متعلق به گروه S ام است.

در این مرحله نخست طول زیر رشته متغیرهای طراحی سطح مقطع مانند قسمت قبل محاسبه می شود و طول زیر رشته هر متغیر طراحی هندسی نیز (مختصات قابل تغییر گرهی) توسط رابطه زیر تعیین می گردد:

$$l_c = \text{INT} \left[ 1 + \frac{\log\left(\frac{c^u - c^l}{\varepsilon}\right)}{\log 2} \right] \quad (29-5)$$

در رابطه بالا  $l_c$  طول زیر رشته متغیر طراحی هندسی،  $c^u$  و  $c^l$  به ترتیب کرانه های فوقانی و تحتانی مختصات قابل تغییر گرهی و  $\varepsilon$  دقت مورد نیاز برای مختصات گرهی است. در ادامه عملیات، اعداد تصادفی دودویی به زیر رشته ها و به تعداد کروموزوم های واقع در جمعیت (pop-size) اختصاص می یابد.

در مرحله رمزگشایی مقدار نظیر سطح مقطع برای هر متغیر طراحی از رابطه زیر حاصل می شود:

$$A_i = S_{(l_i+1)} \quad ; \quad i = 1, \dots, n \quad (30-5)$$

در رابطه بالا  $I_i$  مقدار صحیح<sup>۱</sup> زیر رشته  $i$  ام پس از نگاشت به سیستم دهگانی،  $A_i$  سطح مقطع متغیر طراحی  $i$  ام در موقعیت  $I_i + 1$  لیست پروفیل‌های موجود  $S$  و  $n$  تعداد متغیرهای طراحی سطح مقطع می‌باشد. مقدار فیزیکی هر متغیر طراحی هندسی پس از رمزگشایی بصورت زیر بدست می‌آید :

$$C_i = C_i^l + I_i \frac{C_i^u - C_i^l}{2^{l_{c,i}} - 1} \quad ; \quad i = 1, \dots, n \quad (۳۱-۵)$$

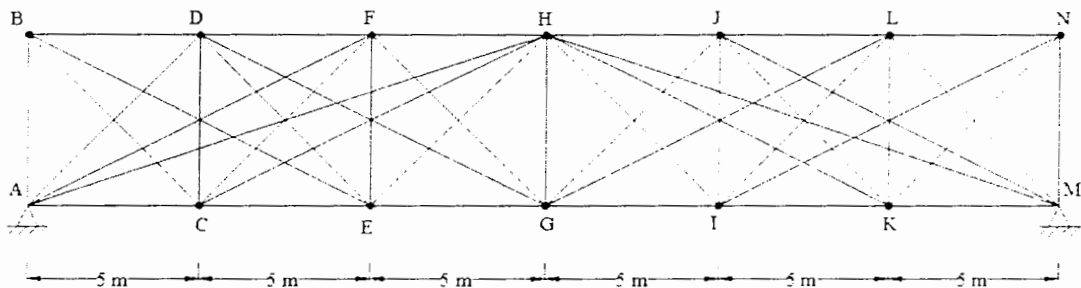
در رابطه بالا  $C_i$ ،  $C_i^u$  و  $C_i^l$  به ترتیب مقادیر فیزیکی، کرانه فوقانی و کرانه تحتانی متغیر طراحی هندسی  $i$  ام،  $I_i$  و  $l_{c,i}$  به ترتیب مقدار صحیح و طول زیر رشته متغیر طراحی هندسی  $i$  ام و  $n$  تعداد کل متغیرهای طراحی هندسی (مختصات قابل تغییر گرهی) است. پس از رمزگشایی، هر کروموزوم معرف یک طرح هندسی با مختصات گرهی و سطح مقطع‌های جدید می‌باشد. لذا با تحلیل سازه به تعداد pop-size و تعیین شایستگی هر کروموزوم، فرآیندهای انتخاب، پیوند و جهش همانند مطالب مطرح شده در فصل دوم بکار گرفته می‌شود

<sup>۱</sup> Decimal Integer<sup>۲</sup> Hasançebi & Erbatur<sup>۳</sup> Wu & Chow

### ۵-۴-۳ مثالهای عددی

#### مثال (۵-۷) خرپای مستوی ۱۴ گرهی

در این مثال با استفاده از روشهای مطرح شده در این رساله و استفاده از تابع جریمه ۱-۲ و تابع شایستگی ۲-۴، بهینه سازی همزمان سطح مقطع، هندسه و توپولوژی سازه خرپایی ۱۴ گرهی در دو حالت، با امکان همپوشانی اعضا و بدون امکان همپوشانی اعضا انجام شده و نتایج با مرجع [۱] مقایسه گردیده است. برای این خرپا سازه مبنا بصورت گراف پایه ای که در شکل ۵-۲۷ نشان داده شده است در نظر گرفته شده است و تحت اثر بار متمرکز به مقدار  $3 \times 10^6 \text{ N}$  قرار دارد که به نوبت به گره های ۳، ۵، ۷، ۹ و ۱۱ اثر می کند و به عبارت دیگر در مجموع پنج وضعیت بارگذاری وجود دارد که در شکل ۵-۲۸ نشان داده شده است.



شکل (۵-۲۸) سازه مبنا برای خرپای ۱۴ گرهی (گراف پایه ستاره ای ۳۱ عضوی)

برای کلیه اعضا  $E = 210.0 \text{ GN/m}^2$ ،  $\sigma_t = 130.0 \text{ MN/m}^2$ ،  $|\sigma_c| = 104.0 \text{ MN/m}^2$  و حداکثر تغییر مکان مجاز برای گره های C، E، G، I و K در جهت y،  $|\Delta_{ay}| = 0.01 \text{ m}$  فرض شده است. مقاطع پیوسته با کرانه تحتانی، فوقانی و دقت بترتیب  $0.01 \text{ m}^2$ ،  $0.1 \text{ m}^2$  و  $0.01 \text{ m}^2$  در نظر گرفته شده است. سطح مقطع اعضا همانند جدول ۵-۲۷ به چهار گروه دسته بندی شده اند و به کلیه اعضا به استثنای اعضای یال تحتانی خرپا، متغیر طراحی بولی توپولوژیکی اختصاص یافته است.

جدول (۵-۲۷) اطلاعات طراحی خرابای ۱۴ گرهی

متغیرهای طراحی					
متغیرهای طراحی سطح مقطع :					
$A_{BD}=A_{DF}=A_{FH}=A_{HJ}=A_{JL}=A_{LN}$ $A_{AC}=A_{CE}=A_{EG}=A_{GI}=A_{IK}=A_{KM}$ $A_{BC}=A_{AD}=A_{DE}=A_{CF}=A_{FG}=A_{EH}=A_{HI}=A_{GJ}=A_{JK}=A_{IL}=A_{LM}=A_{KN}$ $A_{AB}=A_{CD}=A_{EF}=A_{GH}=A_{IJ}=A_{KL}=A_{MN}=A_{AF}=A_{CH}=A_{BE}=A_{DG}=A_{AH}=A_{GL}=A_{IN}=A_{JM}=A_{HK}=A_{HM}$					
متغیرهای طراحی هندسی :					
$y_B=y_N ; y_D=y_L ; y_F=y_J ; y_H$					
متغیرهای طراحی توپولوژیکی :					
1- AC, CE, EG, GI, IK, KM	2- AB, MN	3- BD, LN	4- BC, KN		
5- AD, LM	6- CD, KL	7- DF, JL	8- DE, IL		
9- CF, JK	10- EF, IJ	11- FH, HJ	12- FG, GJ		
13- EH, IJ	14- GH	15- DG, GL	16- BE, IN		
17- AF, JM	18- CH, HK	19- AH, HM			
اطلاعات مربوط به قیود					
قیود تنش محوری :					
$(\sigma_t)_i \leq 130 \text{ Mpa} ; i=1, \dots, 41$					
قیود تغییر مکان :					
$ (\sigma_c)_i  \leq 104 \text{ Mpa} ; i=1, \dots, 41$					
قیود متغیرهای طراحی هندسی :					
$ u_i  \leq 0.01, m ; i=1, \dots, 41$					
$2m \leq y_B, y_D, y_F, y_H, y_J, y_L, y_N \leq 8m$					
لیست پرونیلهای موجود					
$A_i \in S = \{ 0.01, 0.016, 0.022, 0.028, 0.034, 0.04, 0.046, 0.052, 0.058, 0.064, 0.07, 0.076, 0.082, 0.088, 0.094, 0.1 \} (\text{cm}^2)$					
اطلاعات بارگذاری					
	$F_z$	$F_y$	$F_x$	گره ها	وضعیت بار
	$-3 \times 10^6$	.	.	۳	۱
	$-3 \times 10^6$	.	.	۵	۲
	$-3 \times 10^6$	.	.	۷	۳
	$-3 \times 10^6$	.	.	۹	۴
	$-3 \times 10^6$	.	.	۱۱	۵
خصوصیات مکانیکی مصالح					
$2.1 \times 10^{11} \text{ pa}$				مدول الاستیسیته، E	

عملیات بهینه‌سازی برای این سازه به تعداد ۲۰ بار در هر دو حالت اجرا گردیده است. که در این اجراها تعداد نسلها برابر ۳۰۰ و طول کرموزوم‌ها نیز ۲۰۰ در نظر گرفته شده است و احتمال پیوند برابر ۰/۱۰ و نرخ جهش نیز ۰/۰۱ در نظر گرفته شده است. لازم بذکر است که در این مثال اعضایی که تنش در آنها کمتر از مقدار  $1E-5$  می‌باشد تنش آنها برابر صفر در نظر گرفته می‌شود.

#### ۵-۴-۳-۱ متغیرهای هندسی

تعداد متغیرهای هندسی در این مثال ۴ عدد می‌باشد و مختصات عمودی گره‌های N, L, J, H, می‌توانند مقادیری بین ۲ تا ۸ متر را اختیار کنند. در ضمن سازه ما نسبت به عضو میانی GH متقارن می‌باشد و گره‌های B و N، D، L و F و J دارای مختصات عمومی یکسانی هستند و میزان تغییر مکان عمودی برای گره‌های C، E، G، I و K به مقدار 0.01 m محدود شده است.

#### ۵-۴-۳-۲ متغیرهای سطح مقطع

راجان برای نشان دادن متغیرهای پیوسته سطح مقطع از ۴ بیت دو دویی استفاده کرد و لیست سطح مقطعیهای مورد استفاده وی که ما نیز از آنها استفاده کرده ایم، بصورت زیر بوده است:

$$\{0.01, 0.016, 0.022, 0.028, 0.034, 0.04, 0.046, 0.052, 0.058, 0.064, 0.07, 0.076, 0.082, 0.088, 0.094, 0.1\} m^2$$

در این مثال ما بجای استفاده از قید کمانش از قید تنش فشاری استفاده کرده ایم که این قید به همراه قیود دیگر، همچنین مشخصات مصالح مانند مدول الاستیسیته و گروه بندی اعضا در جدول xxxx نشان داده شده است. لازم بذکر است که، در برنامه نرم افزاری نوشته شده جهت بهینه‌سازی با استفاده از الگوریتم ژنتیک توسط ما، در قسمت انتخاب قیود این امکان قرار داده شده است که در مثال‌های مختلف تنها با انتخاب نوع قیود مانند قید کمانش، تنش و یا تغییر مکان و وارد کردن



مقادیر حدی آن ها و پارامترهای مربوطه هر نوع قیدی را که می‌خواهیم در نظر بگیریم. به طور مثال در این مثال خاص قیود تغییرمکان و تنش انتخاب شده است و قید کمانش حذف شده است.

### ۵-۴-۳ متغیرهای توپولوژی

سازه مورد بررسی نسبت به متغیرهای توپولوژی متقارن می‌باشد و در جدول ۵-۲۷ گروه بندی این متغیرهای توپولوژیکی نشان داده شده است. نکته ای که در مورد متغیرهای توپولوژیکی باید در نظر داشت این است که در حالتی که امکان همپوشانی برای اعضا وجود دارد ماتریس همسایگی تعریف شده با ماتریس همسایگی تعریف شده برای حالتی که امکان همپوشانی وجود ندارد متفاوت است که این ماتریس همسایگی در برنامه نوشته شده در محیط نرم افزاری Matlab V7.0 با ماتریس IXFull نشان داده شده است.

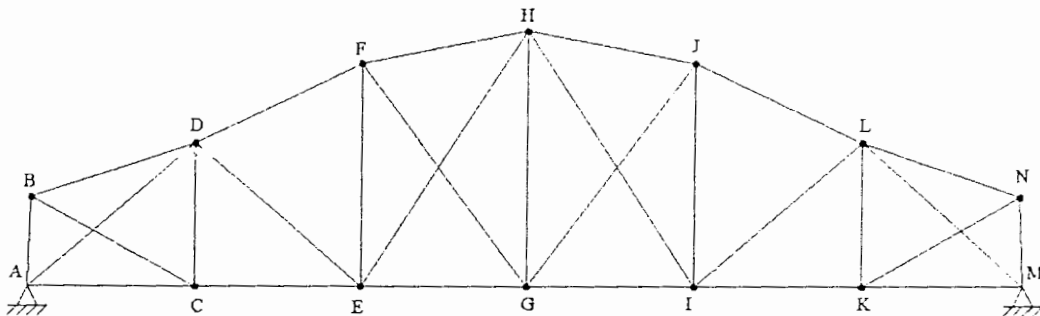
### ۵-۴-۴ نتایج

با استفاده از روش انجام شده، در حالتی که برای اعضا امکان همپوشانی وجود داشته باشد، تعداد نسلهای مورد نیاز برای کسب حجم بهینه ۲۴۱ نسل می‌باشد و در حالتی که اعضا امکان همپوشانی ندارند تعداد نسلهای مورد نیاز برای کسب بهینه ترین حالت حجم سازه ۲۴۷ نسل می‌باشد. تعداد نسلهایی که راجان برای کسب جوابهای بهینه بکار گرفت ۹۶ می‌باشد و شکل ۵-۲۸ بهترین طراحی بدست آمده توسط راجان را نشان می‌دهد. با مقایسه نتایج بدست آمده از روش انجام شده با نتایج موجود در مرجع [1] هنگامی که الگوریتم مراحل سعی و خطا بیشتری را انجام می‌دهد صرفه جویی قابل ملاحظه ای در حجم سازه حاصل می‌گردد. این صرفه جویی به میزان ۱۲٪ در حالتی که برای اعضا امکان همپوشانی وجود دارد و ۷/۱۶٪ برای حالتی که امکان همپوشانی اعضا وجود ندارد، می‌باشد.

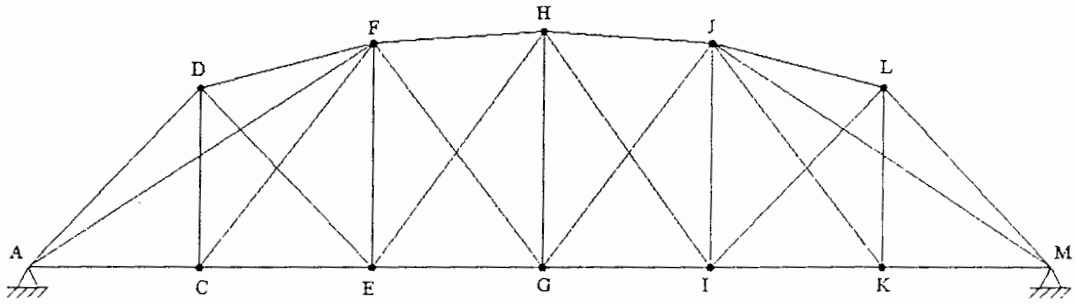
جدول ۵-۲۸ خلاصه نتایج بدست آمده برای این مثال را نشان می‌دهد. در طراحی هایی که در آنها از همپوشانی اعضا جلوگیری به عمل آمده است به نسبت طراحی هایی که در آن ها امکان همپوشانی وجود دارد، حجم سازه کمی بیشتر می‌باشد. با این وجود در ۱۲ بار از ۲۰ بار اجرای عملیات بهینه سازی برای این مثال، نتایج بدست آمده توسط روش رساله، از نتایج بدست آمده توسط راجان بهتر بود و سازه بهینه بدست آمده توسط روش انجام شده نسبت به سازه بهینه بدست آمده توسط راجان دارای حجم کمتری بود. لازم بذکر است که جلوگیری از همپوشانی اعضا این فایده را برای ما به همراه دارد که تعداد طراحی های قابل ساخت بیشتری را برای ما بدست می‌دهد. شکل های ۵-۳۰ و ۵-۳۱ بهترین طراحی های بدست آمده توسط روش انجام شده در این رساله را نشان می‌دهد که با نتایج بدست آمده توسط راجان در شکل ۵-۲۹ مقایسه گردیده است. بمنظور اطمینان از عدم نقض قیود، توسط سازه بهینه بدست آمده توسط روش انجام شده علاوه بر کنترل ماتریس خروجی نقض قیود Cg از برنامه نوشته شده در نرم افزار Matlab V7.0 به کنترل سازه بدست آمده با نرم افزار تحلیلی Etabs V8.31 پرداختیم که با اختصاص متغیرهای بدست آمده سطح مقطع، هندسه و توپولوژی حاصل از عملیات بهینه سازی و بررسی تغییر مکانهای گرهی مربوط به گره های I, G, E, C و K و همچنین کنترل تنش اعضا و پایداری سازه، هیچ گونه نقض قیودی در سازه بهینه بدست آمده با این روش بچشم نخورد.

جدول (۵-۲۸) نتایج سطح مقطع، هندسه و توپولوژی بهینه خرپای مستوی ۱۴ گرهی

حجم ( $m^3$ )	مقادیر فیزیکی متغیرهای هندسی (m)				سطح مقطع اعضای گروه ها ( $m^2$ )				نتایج طرح بهینه
	$y_4$	$y_3$	$y_2$	$y_1$	$A_4$	$A_3$	$A_2$	$A_1$	
5.10	7.66	6.68	4.31	2.33	0.034	0.022	0.016	0.053	Rajan [30] (همپوشانی ممنوع)
4.61	6.85	6.50	5.21	---	0.016	0.028	0.016	0.052	بر اساس روشهای رساله (همپوشانی مجاز)
4.70	7.09	6.66	5.08	---	0.034	0.040	0.022	0.058	بر اساس روشهای رساله (همپوشانی ممنوع)

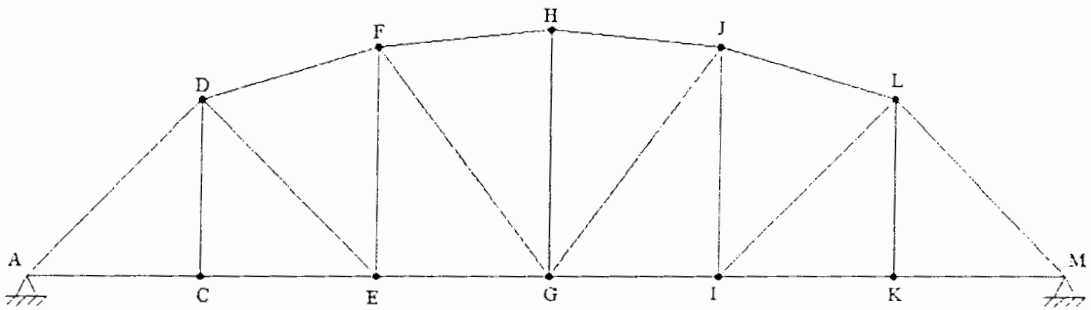


شکل (۵-۲۹) طراحی بهینه خرپای مستوی ۱۴ گرهی توسط راجان (۱۹۹۵)



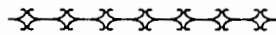
شکل (۵-۳۰) طراحی بهینه خرپای مستوی ۱۴ گرهی توسط روش رساله

(با امکان همپوشانی اعضا)



شکل (۵-۳۱) طراحی بهینه خرپای مستوی ۱۴ گرهی توسط روش رساله

(بدون امکان همپوشانی اعضا)



# فصل ۱

## نتایج و پیشنهادات

## ۶-۱ نتایج

در این رساله بهینه یابی مقاطع، هندسه و توپولوژی خریاها با استفاده از مفاهیم روش سختی و الگوریتم ژنتیک با ایجاد یک بسته نرم افزاری در محیط نرم افزار Matlab V7.0 مورد بررسی قرار گرفت. فهرست نتایج حاصل و پیشنهادات بشرح ذیل می باشد :

۱- عدم شناخت درست از پارامترها و روابط حاکم در GA از قبیل تعداد اعضای جمعیت، تعداد نسلها، نرخ پیوند، نرخ جهش، نوع تابع جریمه و مقادیر ثابتهای دخیل در آن، نوع تابع شایستگی و چگونگی فرآیند انتخاب از جمله عواملی است که سبب می شود GA بهینه محلی را بدست آورد.

۲- در این رساله، برای رفع این نقایص، یک بسته نرم افزاری جهت بهینه سازی سازه های خریایی تهیه شده است که در آن امکان تغییر گزینه های مربوط به الگوریتم ژنتیک و یا اعمال قیود مختلف تنها با انتخاب گزینه مربوطه در ابزار الگوریتم ژنتیک امکان پذیر می باشد. علاوه بر این، در این بسته نرم افزاری امکان ترسیم گراف های مختلف توسط برنامه در حین اجراهای مختلف وجود دارد که این امکان را به کاربر می دهد که با انتخاب گزینه های مختلف مربوط به عملگرهای GA بهترین آنها را برای هر مثال انتخاب کرده و حتی با استفاده از این گراف ها همگرایی و یا عدم همگرایی الگوریتم به سمت جواب های بهینه و یا حتی افتادن در دام بهینه های محلی را به خوبی تشخیص دهد و رفع کند.

۳- یکی دیگر از مشکلات الگوریتم ژنتیک، حجم بالای عملیات محاسباتی توسط آن می باشد که این مشکل با بهبود سرعت و توانایی کامپیوترها در سالهای اخیر به همراه استفاده از برخی از روشهای الگوریتم ژنتیک توسعه یافته ( $DGA$ ) که در این رساله به آنها پرداخته شده است و در آنها امکان استفاده از سیستم های کامپیوتری موازی وجود دارد، تا حدود زیادی مرتفع شده است.

۴- یکی از موارد مهم در بهینه سازی سازه ها، امکان وجود ناپایداریهای موضعی یا کلی در طرحهای بهینه می باشد که ممکن است علیرغم ارضاء قید تغییر مکان در سازه مخفی باشد که در این رساله با استفاده از با استفاده از رابطه مربوط به درجات نامعینی استاتیکی سازه و یا با محاسبه دترمینان ماتریس سختی، ناپایداریهای توپولوژیکی به آسانی شناسایی و کروموزوم های مربوطه بشدت جریمه شده اند. اینکار تا حد زیادی حجم عملیات عددی را کاهش و پایداری توپولوژی بهینه را تضمین می کند. با بررسی مثالها مشخص گردید که در بهینه یابی مقاطع و توپولوژی خرپاها با استفاده از GA، بین ۱۰ الی ۳۰ درصد کروموزوم ها ناپایدار و از بین آنها ۵۰ الی ۹۰ درصدشان ناپایدار توپولوژیکی هستند.

۵- نتایج توپولوژی بهینه به سازه مبنا با تعداد گره های مفروض بستگی دارد. گاهی استفاده از سازه مبنای پر عضو به نحوی که بین هر دو گره، عضوی در نظر گرفته شده باشد ( گراف کامل ) سبب می شود تا طول کروموزوم ها زیاد و در نتیجه سرعت همگرایی کند و بهینه محلی حاصل شود. با توجه به آنکه مدل ریاضی هر سازه اسکلتی توسط نظریه گرافها قابل نمایش است، برای سازه های میله ای رایج، گراف پایه ستاره ای بعنوان سازه مبنای مناسب پیشنهاد گردید.

۶- به کلیه اعضای سازه مبنا متغیر طراحی بولی توپولوژیکی اختصاص یافت. در روند بهینه یابی در صورتیکه مقدار بیت توپولوژیکی صفر شود، عضو مربوطه از سازه مبنا حذف و تحلیل سازه و ارزیابی شایستگی برای سازه باقیمانده ( با بیتهای توپولوژیکی یک برای اعضای آن ) صورت می پذیرد. به این ترتیب پایداری توپولوژی بهینه تضمین خواهد شد.

۷- استفاده از محیط نرم افزار Matlab V7.0 برای برنامه نویسی و ایجاد بسته نرم افزاری جهت بهینه سازی سازه ها بدلیل سازگاری این نرم افزار با عملیات ماتریسی بسیار مناسب بوده و بدلیل استفاده از m فایل های جداگانه برای فایل ورودی سازه، اپراتورهای مختلف الگوریتم

ژنتیک مانند انواع مختلف عملگرهای پیوند، انتخاب و یا حتی جهش و فایل های خروجی جداگانه مانند انواع مختلف نمودارهای خروجی کارکردن با این مثالها را هرچه بیشتر آسانتر می کند. همچنین ایجاد یک جعبه ابزار مانند آنچه در شکل ۵-۳ نشان داده شده است، انتخاب پارامترهای مختلف الگوریتم ژنتیک و نمودارهای خروجی را برای ما بیش از پیش آسان می کند.

- ۸- در این فصل بهینه یابی مقاطع منفصل و هندسه پیوسته خرپاها مورد بررسی قرار گرفت. ملاحظه گردید گاهی بهینه یابی سطح مقطع و هندسه وزن بیشتری نسبت به حالتی که فقط بهینه یابی مقاطع برای هندسه اولیه انجام شود، نتیجه می دهد. این مساله بدلیل بزرگ شدن دامنه فضای طراحی ناشی از طول شدن طول کروموزوم ها و ترکیبات بسیار زیادی است که مقاطع منفصل و مختصات قابل تغییر گرهی می توانند از خود نشان دهند.
- ۹- بمنظور اطمینان از عدم نقض قیود، توسط سازه بهینه بدست آمده، علاوه بر کنترل ماتریس خروجی نقض قیود Cg از برنامه نوشته شده در نرم افزار Matlab V7.0 به کنترل سازه بهینه با نرم افزار تحلیلی Etabs V8.31 پرداختیم و با اختصاص متغیرهای بدست آمده سطح مقطع، هندسه و توپولوژی حاصل از عملیات بهینه سازی و بررسی تغییر مکانهای گرهی، تنش اعضا و پایداری سازه، از عدم نقض قیود در سازه بهینه اطمینان حاصل گردید

## ۶-۲ پیشنهادات

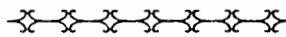
- ۱- مسائل واقعی دنیای سازه ها اغلب دارای تعداد زیادی از متغیرهای طراحی می باشد و اغلب بدلیل وجود قیود متفاوت حاکم بر مسئله، فضای قابل قبول طراحی بسیار محدود می باشد. بنابراین بیشتر کروموزوم های تولیدی در الگوریتم ژنتیک کروموزوم های مرده می باشند و قیود را اقناع نمی کنند و در این موارد، الگوریتم ژنتیک نمی تواند فضای طراحی را



به طور مؤثری مورد کاوش قرار دهد. بنابراین وجود مکانیزمی که در آن کروموزوم های مرده تولید نشود بسیار مهم می باشد. پیشنهاد می گردد در تحقیقات آتی محققان بر روی شیوه های جدید حل این مشکل بپردازند.

۲- پیشنهاد می گردد محققان هرچه بیشتر برروی شیوه هایی از الگوریتم ژنتیک که در آن ها امکان استفاده از قابلیت های کامپیوترهای موازی وجود دارد بپردازند. تا امکان استفاده از عملیات بهینه سازی با استفاده از الگوریتم ژنتیک در سازه های واقعی هرچه بیشتر بوجود آید.

۳- پیشنهاد می گردد محققان به ایجاد یک لینک مناسب بین عملیات بهینه سازی سازه ها و ابزار بهینه سازی الگوریتم ژنتیک موجود در نرم افزار Matlab V7.0 بپردازند و هرچه بیشتر از قابلیت های این نرم افزار جدید در امر بهینه سازی سازه ها استفاده شود.

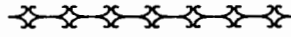


# فهرست مراجع

- [1] Kevin M. Gillman. Optimization of Shape, Size, and Topology Design Variables in Trusses With a Genetic Algorithms, Ph.D. thesis Brigham Young University, April (2005)
- [2] V.Kalatjari, "truss optimization using genetic algorithm and force method." Phd Thesis, in Persian, science and technology University, , Tehran. (2001)
- [3] Cha Özgür Yeniay. "penalty function methods for constrained optimization with genetic algorithms." *Mathematical and Computational Applications*, Vol. 10, No.1, pp. 45-56, 2005.
- [4] Kang Seok Lee, Member, ASCE, and Zong Woo Geem (2004). "An New Structural Optimization Method for Structures with Discrete Sizing Variables." ASCE, 118(5), 1233-1250.
- [5] Jeffrey Smith, Jessica Hodgins, " *Multi* Creating Models of Truss Structures with Optimization". Carnegie Mellon University, the Association for Computing Machinery, Inc.( 2002 ACM 1-58113-521-1/02/0007)
- [6] Riccardo Poli and W.B. Langdon, "Genetic Programming with One-Point Crossover and Point Mutation." The University of Birmingham (UK), D. (1997).
- [7] Rajan, S. D. (1995). "Sizing, shape, and topology design optimization of trusses using genetic algorithm." *J. Struct. Engrg.*, ASCE, 121(10), 1480-1487.
- [8] A. Kaveh, *Optimal Strcutural Analysis*, RSP (John Wiley), UK, (1997).

- [9] A. Kaveh, *Structural Mechanics: Graph and Matrix Methods*, RSP (John Wiley), U.K., 2nd Edition, (1995).
- [10] A.Kaveh, M.Sabaghian and V.Kalatjari, Optimal topology of trusses using the graph theory, The First Conference of Iranian Society of Civil Engineers. ( Edited by M.Tehranizadeh ), 185-192, in Persian, Tehran, (1999).
- [11] A. Kaveh and V. Kalatjari, Genetic algorithm for discrete-sizing optimal design of trusses using the force method, *Int. J. Numer. Meths. Engrg.*. To appear. (2001).
- [12] Charlotte Larsen, Joen Sindholt, Optimization of compliant mechanisms using genetic algorithms, Technical University of Denmark, June. 2003.
- [13] Andrew Burton. (1992). "Truss Optimization Using Genetic Algorithms." Mathematics and Computer Science, Gonzaga University Spokane, WA 99258. July,2003.
- [14] [GA Toolbox] A. Chipperfield, P. Flemming, H. Pohlheim and C. Fonseca: *Genetic Algorithm Toolbox, for use with Matlab, ver. 1.2*. University of Sheffield. <http://www.shef.ac.uk/gaipp/ga-toolbox/>
- [15] "Genetic Algorithm and Direct Search Toolbox Release Notes." *Mathwork*. May, 2004.
- [16] Bradley J. Lucier. (1998). "Program Optimization for Faster Genetic Programming." , Purdue University Dept of Computer Science W Lafayette. IN 47907.
- [17] Yasuhiro KIDA, Hiroshi KAWAMURA. "Multi-Objective Optimization of Spatial Truss Structures by Genetic Algorithm." *Department of Architecture and Civil Engineering, Kobe University, Rokkodai, Nada 1-1, Kobe 657-8501, Japan* January 27, 2000
- [18] Toropov, V. V., Mahfouz, S. Y. and Westbrook, R. K. (1999), "Discrete design optimization of three-dimensional steel structures using a genetic algorithm." *Short Paper Proceedings of the 3rd World Congress of Structural and Multidisciplinary Optimization*, Buffalo, NY, May, 269-271.
- [19] A.Kaveh and V.Kalatjari, Discrete sizing and continuous shape optimal design of trusses using the force method and genetic algorithms. submitted for publication. (2001).

- [20] Shen-Yen Chen, using genetic algorithms for the optimal design of structural systems, phd theses (December 1997).
- [21] Yusuke Tanimura. (2001). "Discussion on Distributed Genetic Algorithms for Designing Truss Structures." genetic and evolutionary computation conference, 2001, pp.1085-1090





**روش تحلیل سازه**

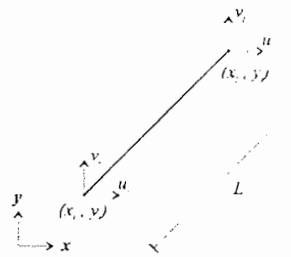
## روش تحلیل سازه

برای انجام محاسبات تغییر مکان سازه و بدنبال آن محاسبه تنش و کرنش در اعضای سازه خرابایی از یک روش خطی اجزای محدود (*FEM*) استفاده شده است. در این روش تغییر مکان در هر گره در جهت‌های اصلی مورد محاسبه قرار می‌گیرد و در هر قسمت با توجه به نیاز می‌توان آنها را مورد استفاده قرار داد. این روش برای تبدیل به برنامه کامپیوتری بسیار مناسب می‌باشد.

## الف- ماتریس سختی

ماتریس سختی  $K$  از ماتریس‌های سختی المان‌های سازه بدست می‌آید بدین صورت که در ابتدا برای هریک از اعضا یک ماتریس سختی محاسبه می‌شود و سپس این ماتریس‌ها در درون ماتریس سختی  $K$  وارد می‌شود.

برای ایجاد ماتریس  $K$  برای یک سازه خرابایی، هر کدام از اعضای خرپا را در نظر می‌گیریم.



شکل (الف-۱) درجات آزادی اعضا (*dof*)

هر کدام از اعضای خرپا ۴ درجه آزادی دارند. یکی از این درجات آزادی با حرکت گره به اندازه  $u_1$  در جهت درجه آزادی اول فعال می‌شود. و با استفاده از قانون هوک می‌توان نیروی محوری ایجاد شده در عضو مورد نظر را بدست آورد.

$$\sigma = E\varepsilon$$

که در آن  $\sigma$  تنش در عضو،  $E$  مدول الاستیسیته یانگ و  $\varepsilon$  مقدار کرنش می‌باشد. برای بدست آوردن کرنش در جهت محوری، لازم است که تغییر مکان را بر روی جهت محوری تصویر کنیم. که داریم:

$$L = \sqrt{(x_j - x_i)^2 + (y_j - y_i)^2}$$

$$c = \cos \beta = \frac{x_j - x_i}{L}$$

$$s = \sin \beta = \frac{y_j - y_i}{L}$$

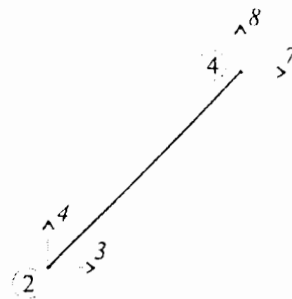
سپس، نیرو برای تغییر مکان های کوچک را می‌توان به صورت زیر نوشت:

$$F = \sigma A = \frac{EA}{L} cu_i$$

که در آن  $A$  سطح مقطع عرضی،  $L$  طول اولیه یا طول عضو تغییر شکل نیافته و  $cu_i$  تغییر مکان تصویر شده بر روی عضو می‌باشد.

نیروی ایجاد شده در یکی از دو انتهای عضو را در جهت دو درجه آزادی تصویر می‌کنیم و بهمین

ترتیب نیروی ایجاد شده در انتهای دیگر عضو. (شکل الف-۲ را ببینید)



شکل (الف-۲) عضو پس از تغییر مکان گرهی ( $u_i > 0, u_j = v_i = v_j = 0$ )

با نوشتن معادله تعادل داریم:

$$p_i = -p_j = Fc$$

$$q_i = -q_j = Fs$$

این روش با که با حرکت دادن یک گره در امتداد یکی از درجات آزادی آن گره انجام می‌شود را می‌توان به طریق زیر نشان داد:

$$\frac{EA}{L} \begin{Bmatrix} c^2 \\ cs \\ -c^2 \\ -cs \end{Bmatrix} u_i = \begin{Bmatrix} p_i \\ q_i \\ p_j \\ q_j \end{Bmatrix} \quad (\text{الف-۱})$$

این شیوه برای هریک از سه درجه آزادی این المان تکرار می‌شود و با تکمیل آن معادله تعادل به شکل زیر نوشته می‌شود:

$$\frac{EA}{L} \begin{bmatrix} c^2 & cs & -c^2 & -cs \\ cs & s^2 & -cs & -s^2 \\ -c^2 & -cs & c^2 & cs \\ -cs & -s^2 & cs & s^2 \end{bmatrix} \begin{Bmatrix} u_i \\ v_i \\ u_j \\ v_j \end{Bmatrix} = \begin{Bmatrix} p_i \\ q_i \\ p_j \\ q_j \end{Bmatrix} \quad (\text{الف-۲})$$

که در آن ماتریس سختی  $k$  عضو عبارتست از:

$$k = \frac{EA}{L} \begin{bmatrix} c^2 & cs & -c^2 & -cs \\ cs & s^2 & -cs & -s^2 \\ -c^2 & -cs & c^2 & cs \\ -cs & -s^2 & cs & s^2 \end{bmatrix} \quad (\text{الف-۳})$$

ماتریس سختی  $k$  عضو را برای تمام المان‌ها محاسبه می‌کنیم و ماتریس سختی  $K$  کلی را برای کل سازه به روشی که در ادامه آورده می‌شود مورد محاسبه قرار می‌دهیم.

هر گره سازه دارای یک شماره می‌باشد و همین‌طور هر درجه آزادی نیز دارای یک شماره است. به‌طور مثال، گره شماره یک دارای درجات آزادی ۱ و ۲ می‌باشد و گره شماره دو دارای درجات آزادی ۳ و ۴ می‌باشد و سایر گره‌ها نیز به همین ترتیب.

سپس ماتریس سختی عضو  $k$  را در ماتریس سختی کل سازه  $K$  که بوسیله تعداد درجات آزادی فعال که مشخص‌کننده تعداد سطرها و تعداد درجات آزادی که نیروها بر روی آنها تصویر می‌شود که مشخص‌کننده ستونهای ماتریس سختی کل سازه  $K$  می‌باشد، وارد می‌کنیم.



پس معادله مورد استفاده برای تحلیل به صورت زیر می‌شود.

$$Ku = f_{ends} \quad (\text{الف-۴})$$

یا

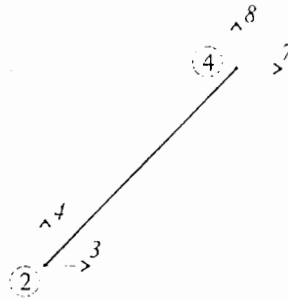
$$u = K^{-1} f_{ends} \quad (\text{الف-۵})$$

که در آن  $f_{ends}$  نیروهای انتهایی عضو می‌باشد.

### مثال: رویهم گذاری ماتریس سختی

در یک طراحی خرپا، یک عضو را که بین دو گره ۲ و ۴ قرار گرفته و دارای درجات آزادی ۳، ۴، ۷

و ۸ می‌باشد را در نظر بگیرید.



شکل (الف-۳) عضو قرار گرفته بین گره های ۲ و ۴ به همراه درجات آزادی آن

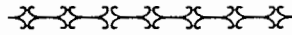
ماتریس سختی این عضو به صورت زیر بدست می‌آید:

$$k = \frac{EA}{L} \begin{bmatrix} k_{33} & k_{34} & k_{37} & k_{38} \\ k_{43} & k_{44} & k_{47} & k_{48} \\ k_{73} & k_{74} & k_{77} & k_{78} \\ k_{83} & k_{84} & k_{87} & k_{88} \end{bmatrix} \quad (\text{الف-۶})$$

و می‌توان این ماتریس را با توجه به محل آن در ماتریس سختی کل سازه  $K$  وارد کرد، که در آن

$$k = \frac{EA}{L} \begin{bmatrix} \dots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \dots & k_{33} & k_{34} & \dots & k_{37} & k_{38} & \dots \\ \dots & k_{43} & k_{44} & \dots & k_{47} & k_{48} & \dots \\ & \vdots & \vdots & \ddots & \vdots & \vdots & \\ \dots & k_{73} & k_{74} & \dots & k_{77} & k_{78} & \dots \\ \dots & k_{83} & k_{84} & \dots & k_{87} & k_{88} & \dots \\ & \vdots & \vdots & & \vdots & \vdots & \ddots \end{bmatrix} \quad (\text{الف-۷})$$

با انجام دادن این کار برای سایر اعضا، ماتریس سختی کل سازه  $K$  بدست می‌آید.



**`%Linear analysis for truss structure`**

`%Date 15. 2.2005`

`%File : truss3.m`

`%Editor : Ehsan_Eeighami`

`%E_mail : ehsan_1638@yahoo.com *`

```
ndof = size(X,1)*size(X,2);           % Number of degrees of freedom
ne = size(IX,1);                       % Number of elements
nb = size(bound,1);                     % Number of boundaries
sprintf('Number of DOF %d,Number of elements %d',ndof,ne);
```

```

stress=zeros(Nind,ne)
eepsilon=zeros(ne,Nind);
stress=zeros(Nind,ne);
for f = 1:Nind

    xxxxxxxxxxxxxxxxxxxInitialize Global Variablesxxxxxxxxxxxxxxxxxxxx

    K=zeros(ndof,ndof);
    F(ndof,1)=0;                               % Force vector
    U(ndof,1)=0;                               % Displacement vector
    R(ndof,1)=0;                               % Residual vector

    xxxxxxxxxxxxxxxxxxxAdded resetting of F,U,Rxxxxxxxxxxxxxxxxxxxx.

    F = zeros(ndof,1);
    U = zeros(ndof,1);
    R = zeros(ndof,1);

    xxxxxxxxxxxxxxxxxxxInitialize Stiffness Matrixxxxxxxxxxxxxxxxxxxxx.

    K=zeros(ndof,ndof);

    xxxxxxxxxxxDeformation Of Node Cordinate Matrixxxxxxxxxxxxx

    X = [ 31.75  6.350
          6.350 25.40
          C(f,1) C(f,2)
          6.350 19.05
          C(f,3) C(f,4);
          6.350 12.70
          C(f,5) C(f,6)
          6.350 6.350
          C(f,7) C(f,8)
          6.350 0.00
          0.00 0.00  ]

    xxxxxxxxxxxxxxxxxxxAssemble global stiffness matrixxxxxxxxxxxxxxxxxxxxx.

    for e = 1:ne
        ddx = X(IX(e,2),1) - X(IX(e,1),1);
        ddy = X(IX(e,2),2) - X(IX(e,1),2);
        E = 6.895e4;
        edof = [2*IX(e,1)-1 2*IX(e,1) 2*IX(e,2)-1 2*IX(e,2)];
        L = sqrt((ddx^2 + ddy^2));
        ke = A(i,e)*E/L^3 * [ddx^2 ddx*ddy -ddx^2 -ddx*ddy
                           ddx*ddy ddy^2 -ddx*ddy -ddy^2
                           - ddx^2 -ddx*ddy ddx^2 ddx*ddy
                           - ddx*ddy -ddy^2 ddx*ddy ddy^2 ];

        K(edof,edof) = K(edof,edof)+ ke;
    end
end

```

```

end
for j=1:nb
K(2*bound(j,1)+bound(j,2)-2,:)=0;
K(:,2*bound(j,1)+bound(j,2)-2)=0;
K(2*bound(j,1)+bound(j,2)-2,2*bound(j,1)+bound(j,2)-2)=1;
end
F(2*P(1:size(P,1),1)+P(1:size(P,1),2)-2) = P(1:size(P,1),3);

xxxxxxxxxxxxxxxxxxxxFE_ANALYSISxxxxxxxxxxxxxxxxxxxx.
U = K\F;

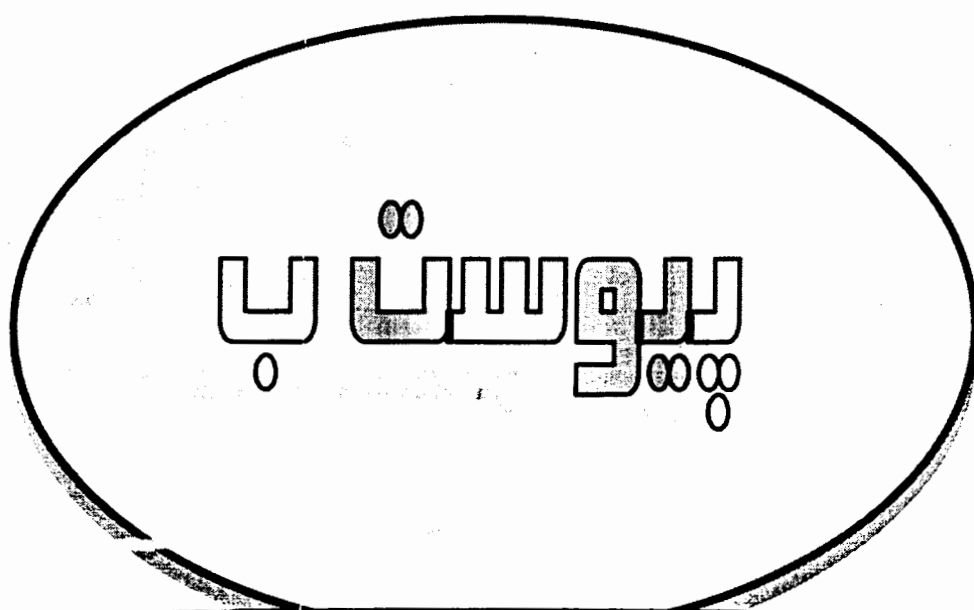
xxxxxxxxxxxxCalculates the strain in each barxxxxxxxxxxxx.
epsilon;[] =

xxxxxThe length of the trusses in the undeformed statexxxxxxx
IXLength;[]=
for e=1:size(IX,1)
if IX(e,3)==1
ddx=X(IX(e,2),1)-X(IX(e,1),1);
ddy=X(IX(e,2),2)-X(IX(e,1),2);
IXLength=[IXLength;sqrt((ddx^2+ddy^2);[
end;
end
%The length of the trusses in the deformed state
IXDefLength=[]

for e=1:size(IX,1)
if IX(e,3) == 1
ddx=X(IX(e,2),1)+U(2*IX(e,2)-1,1)-(X(IX(e,1),1)+ U(2*IX(e,1)-1,1));
ddy=X(IX(e,2),2)+U(2*IX(e,2),1)-(X(IX(e,1),2)+U(2*IX(e,1),1));
IXDefLength = [IXDefLength;sqrt((ddx^2+ ddy^2))];
end
end
epsilon=(IXDefLength-IXLength)./IXLength;
uu(:,f) = U;
eepsilon(:,f) = epsilon;
end
U = uu;
epsilon = eepsilon;

xxxxxxxxxxxxCalculates the stress in each barxxxxxxxxxxxx.
stress = 6.895E4*epsilon

```



**بسته نرم افزاری بهینه یابی فریها**

**با استفاده از الگوریتم ژنتیک**

**و نرم افزار Matlab**

## بسته نرم افزاری بهینه یابی خرپاها با استفاده از الگوریتم ژنتیک

این بسته نرم افزاری که با استفاده از برنامه نویسی در محیط نرم افزار Matlab V7.0 بوجود آمده است. دارای پیکره بندویی به صورت یک فایل ورودی جداگانه برای مختصات گرهی سازه، مشخصات مصالح، شرایط مرزی، نیروهای اعمال شده بر سازه، همچنین لیست مقاطع مورد استفاده و نیز سایر پارامترهای مورد استفاده در الگوریتم ژنتیک می باشد. که این فایل به نام Calladine.m می باشد. که نمونه ای از این فایل بصورت زیر می باشد:

```
% Truss structure
% Date : 15.12.2004
% File : calladine.m
% Editor : ehsan_zeighami
% E-mail : ehsan_1638@yahoo.com
% Element type : Truss
% No.of Nodes : 6
% No.of Elements : 10
clear all
% Nodal Coordinates X(x,y)
X = [
    18.288 9.144
    18.288 0.000
    9.144 9.144
    9.144 0.000
    0.000 9.144
    0.000 0.000
];
% Topology matrix IX(node1,node2,propno)
IXFull = [
    1 2 1;
    1 3 1;
    1 4 1;
    2 3 1;
    2 4 1;
    3 4 1;
    3 5 1;
    3 6 1;
    4 5 1;
    4 6 1;
];
IXF = IXFull;
% Boundary conditions bound(node,dof,disp),
bound = [
    5 1 0.000000000
    5 2 0.000000000
    6 1 0.000000000
    6 2 0.000000000
];
% Actuation forces P(node,dof,force),
P = [
```

```
2 2 -445400.00000e-6
4 2 -445400.00000e-6
```

];

% Boundry Condition Related to Shape Variable,

```
%Cu = [31.115 6.223 24.765 6.223 18.415 6.223 12.065 6.223];
```

```
%Cl = [19.685 -5.715 13.335 -5.715 6.985 -5.715 0.635 -5.715];
```

```
%Step = [0.0254 0.0254 0.0254 0.0254 0.0254 0.0254 0.0254 0.0254];
```

قسمت بعدی این بسته نرم افزاری شامل فایل اصلی خود الگوریتم ژنتیک به نام GA1.m می باشد که کلیه عملیات مربوط به الگوریتم را انجام می دهد و یک فایل تحلیل سازه به نام truss3.m که در پیوست الف مراحل آن به طور کامل تشریح گردید و در نهایت یک فایل که کار ترسیم شکل سازه اولیه و سازه های بهینه شده و تغییر شکل یافته را نشان می دهد. لازم به توضیح است که برنامه اصلی در حین اجرا زیر برنامه هایی را فراخوانی می کند مربوط به اپرتورهای الگوریتم ژنتیک می باشد مانند تولید جمعیت اولیه که به دلیل زیاد بودن حجم این فایلها یک کپی از این فایلها در CD ضمیمه پایان نامه آورده شده است تا برای خواننده قابل استفاده باشد.

```
clear all;
```

```
% *****
```

```
% INPUT
```

```
% *****
```

```
calladine;
```

```
Nind = 2; %number of individuals in population
```

```
Nsg = 10; %number of Section Groups.
```

```
%Nsv = 8; %number of Shape Variables.
```

```
mutRate = 0.001; %Mutation Rate
```

```
crossRate = 0.95; %crossover Rate
```

```
gen = 1 ; %Generations
```

```
E = 6.895E4; %modulus of elastisity (Mpa)
```

```
s = 1e-4*[10.45 11.61 15.35 16.90 18.588 19.94 20.19 21.81 23.42 24.77 24.97 26.97 28.97 30.97 32.06
```

```
33.03 37.03 46.58 51.42 74.19 87.10 89.68 91.61 100.0 103.23 121.29 128.39 141.94 147.74 170.97
```

```
193.55 216.13];
```

```
%availabe sction
```

```
density = 0.0272e6; %density of element(N/m3)
```

```
%alfa=10; %Buckling Coefficinet
```

```
ne = 10; %no. of element
```

```
%k = 10;
```

```
% *****
```

```
% Calculation Of Lenght Of Cromosom
```

```
% Related to Sizing Variable
```

```
% *****
```

```
for i = 1:Nsg
```

```
Lsv(i) = fix((log(max(size(s)))/log(2))+0.95);
```

```
end
```

```
Lsvt = sum(sum(Lsv)); %Lenght of Sizing Variables.
```

```
% *****
```

```

%          Calculation Of Lenght Of Cromosom
%          Related to Shape Variable
% *****
%for i = 1:Nsv
%  Lc(i) = fix(1+log2((Cu(i)-Cl(i))/Step(i))/log(2));
%end
%  Lct = sum(sum(Lc)); %Lenght Of Cromosom Related
%          % to Shape Variable

% *****
%          generate initial population
% *****
Lind = Lsvt+ne;
newPop = makePopBars(Nind,Lind);
% *****

    for f = 1:gen

% *****
%          Decoding of Cromosom Related To Topology Variable
% *****
IXFull=IXF;
for i=1:Nind
    for h=1:ne
        rt = newPop(i,Lsvt+h);
        if rt==0
            IXFull(h,:)=0;
        end
    end
    rtt=(sum(IXFull)');
    rtt = find(rtt==0);
    df = size(rtt);
    IXFull=sort(IXFull);
    for v=1:df(1)
        IXFull(1,:)=[];
    end
end
g = size(IXFull);
ne = g(1);

% *****
%          Decoding of Cromosom Related To Sizing Variable
% *****
for k = 1:Nind
    for i = 1:ne
        I3 = newPop(k,((Lsv(i)*(i-1)+1):Lsv(i)*i));
        for j = 1:Lsv(i)
            I3(j) = I3(j)*(2^(Lsv(i)-j));
        end
        jj = (sum(I3)');
        I4(k,i) = jj;
    end
    for i = 1:ne
        I44(k,i) = s(I4(k,i)+1);
    end
end
I44:          % Sizing Decoded Matrix

```



```

% *****
% ***** Deformation of Area Matrix *****
% *****
A = I44;
Area = A;

% *****
% Calculation Of Element Lenght
% *****
IX = IXFull;
for i = 1:Nind
    for e = 1:ne
        dx = X(IX(e,2),1) - X(IX(e,1),1);
        dy = X(IX(e,2),2) - X(IX(e,1),2);
        L(i,e) = sqrt((dx^2 + dy^2)); %Length Of Element
    end
end

% *****
% Weight Function
% *****
Weight1 = density*L.*Area;
Weight = (sum(Weight1)');
% *****
% *****
Pop = newPop;
IX = IXFull;
% *****
% Calculate displacement & check stress
% *****
Truss3;
% *****
% break the restraint
% *****
stressnorm = (abs(stress/172.4))-1;
for i = 1:Nind
    for j = 1:ne
        if stressnorm(i,j)<0
            stressnorm(i,j) = 0;
        end
    end
end

Cg = stressnorm;
Cg = (sum(Cg'))';
% *****
% Penalty Function
% *****
k=10;
penalty = zeros(size(Weight));
for h = 1:Nind
    penalty = k*Cg.*Weight;
end
penalty;
% *****
% Modified Goal Function
% *****
Mgoal = Weight+penalty;

```

```

Mgoal;
% *****
%           Fitness Function
% *****
Fitness = zeros(size(Mgoal));
Fitness = (max(Mgoal)+min(Mgoal))-Mgoal;
Fitness;
% *****
%           Selection Procedure(rws)
% *****
%resetting variables
f1 = zeros(size(Fitness));    % fi/(sigma(fi))
f2 = zeros(size(Fitness));    % fi/(sigma(fi))tagamoe
f3 = zeros(size(Fitness));    % random number
f1 = Fitness./(sum(Fitness));
f2(1) = f1(1);
for i = 2:Nind
    f2(i) = f2(i-1)+f1(i);
end
f2;
f3 = rand(Nind,1);
% *****
%           (rws)
% *****
t=zeros(Nind,1);
for i = 1:Nind
    t(i) = min(find (f2>f3(i)));
end
t;           %number of selected population
% *****
%           selected population
% *****
for i = 1:Nind
    newpop1(i,:) = newPop(t(i),:);
end
newPop = newpop1;
% *****
% *****
%           Genetic Operator
% *****
% *****

% *****
% *****
%           Crossover
% *****
%resetting variables
% *****
newPop2 = zeros(size(newPop));    % selected population
newPop3 = zeros(size(newPop));    % unselected population
ff2 = [];
ff3 = [];
% *****
rand3 = rand(Nind,1);
crossed = rand3<crossRate;
% *****
for i = 1:Nind
    if crossed(i) == 1

```

```

        newPop2(i,:) = newPop(i,:);
    else
        newPop2(i,:) = 0;
    end
end
% *****
for i = 1:Nind
    if crossed(i) == 0
        newPop3(i,:) = newPop(i,:);
    else
        newPop3(i,:) = 0;
    end
end
% *****
    newPop2;
    newPop3;
% *****
    sum2 = sum(newPop2');
    ff2 = find(sum2==0);
    sort(ff2);
    for i = 1:size(ff2)
        newPop2((ff2(i)-i+1):end) = [];
    end

%_ *****
    sum3 = sum(newPop3');
    ff3 = find(sum3==0);
    sort(ff3);
    for i = 1:size(ff3)
        newPop3((ff3(i)-i+1):end) = [];
    end

%_ *****
    newPop2;
    newPop3;
% *****
b = size(newPop2);
newPop22 = zeros(size(newPop2));
b1 = round(b(1)/2-0.1);
cS = round(2+(Lind-3)*rand(b1,2));
cS = (sort(cS'))';
op = size(cS);
oS = zeros(2*op(1),2);
for u = 1:op(1)
    oS(2*u-1,:) = cS(u,:);
    oS(2*u,:) = cS(u,:);
end
cS = oS;
for i = 1:2:2*b1
    newPop22(i,1:cS(i,1)) = newPop2(i,1:cS(i,1));
    newPop22(i,cS(i,1)+1:cS(i,2)-1) = newPop2(i+1,cS(i,1)+1:cS(i,2)-1);
    newPop22(i,cS(i,2):end) = newPop2(i,cS(i,2):end);

    newPop22(i+1,1:cS(i,1)) = newPop2(i+1,1:cS(i,1));
    newPop22(i+1,cS(i,1)+1:cS(i,2)-1) = newPop2(i,cS(i,1)+1:cS(i,2)-1);
    newPop22(i+1,cS(i,2):end) = newPop2(i+1,cS(i,2):end);
end
end

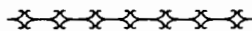
```

```

% *****
if b1~= b(1)/2
    newPop22(end,:) = newPop2(end,:);
end
newPop22;
% *****
r1 = size(newPop22);
newPop = zeros(Nind,Lind);
for i = 1:r1(1)
    newPop(i,:) = newPop22(i,:);
end
for i = (r1(1)+1):Nind
    newPop(i,:) = newPop3((i-r1(1)),:);
end
newPop:
% *****
%          Mutation
% *****
mut = rand(Nind,Lind);
t2 = mut<mutRate;
for i = 1:Nind
    for j = 1:Lind
        if t2(i,j) == 1
            if newPop(i,j) == 1
                newPop(i,j) = 0;
            else
                newPop(i,j) = 1;
            end
        end
    end
end
newPop:
end
newPop:
plott:

function [P] = makePopBars(Individuals,Bits)
P=makePop(Individuals,Bits)
for ind = 1:Individuals
for i = 1:Bits
random = rand;
number = random ;
bit = round(number);
P(ind,i)=bit;
end
end

```



## **Abstract**

This thesis is about the optimization of Trusses structures. Trusses structures are those constructed of long, thin members. There are three types of design variables that can be considered in a Trusses structure. These are size variables, topology variables, and shape variables.

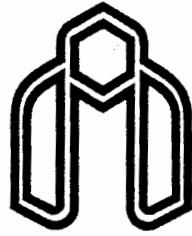
A size variable dictates the size of a member, which could be the crosssectional area, the moment of inertia, etc.

Topology variables dictate the presence or absence of members in a structure.

Shape variables dictate the joint coordinates of a structure. Size variables are ideally represented with discrete section property values. Topology variables are represented as discrete integer values. However, shape variables are typically represented as continuous, or real number values.

The purpose of these contributions was to make optimization using a genetic algorithm more accessible to design engineers.

The Genetic algorithm was demonstrated in Many original examples as well as an example from a published work. The algorithm was effective in finding lighter designs.



Shahrood University Of Technology  
Faculty Of Civil Engineering  
Structural Department

# **Sizing, Shape and Topology Optimization Of Trusses Using Genetic Algorithm And Matlab**

By:  
**Ehsan Zeighami**

Supervisor:  
**Dr. Vahidreza kalatjari**

September 2005