



بهینه‌سازی وزن قاب‌های سه‌بعدی در ساختمان‌های قاب خمشی فولادی متوسط با استفاده از تحلیل دینامیکی طیفی و روش‌های محاسبات نرم و مقایسه نتایج

حمیدرضا عباسیان جهرمی^{۱*}، سعید اصیل قره‌باغی^۲، محمدصادق اسحق‌ی خانقاه^۳

^۱ استادیار، دانشکده عمران، دانشگاه خواجه نصیرالدین طوسی، تهران، ایران

^۲ استادیار، دانشکده عمران، دانشگاه خواجه نصیرالدین طوسی، تهران، ایران

^۳ کارشناس ارشد، دانشکده عمران، دانشگاه خواجه نصیرالدین طوسی، تهران، ایران

تاریخچه داوری:

دریافت: ۱۳۹۷-۰۷-۱۰

بازنگری: ۱۳۹۷-۰۹-۱۰

پذیرش: ۱۳۹۷-۱۱-۱۷

ارائه آنلاین: ۱۳۹۷-۱۱-۱۷

کلمات کلیدی:

محاسبات نرم

الگوریتم ژنتیک بهبودیافته

الگوریتم مورچگان بهبودیافته

شبکه‌های عصبی

قاب خمشی فولادی

تحلیل دینامیکی طیفی

مدل‌سازی سه‌بعدی

خلاصه: با توجه به لزوم طراحی بهینه و اقتصادی سازه‌ها، کاهش وزن نهایی سازه طراحی شده که رابطه مستقیم با هزینه ساخت در پروژه‌ها دارد، اهمیت زیادی پیدا می‌کند. از طرفی با کاهش وزن مقاطع استفاده‌شده در سازه، پارامترهای مقاومتی المان مربوطه کاهش می‌یابد و سازه توان تحمل در مقابل بارهای وارده را از دست می‌دهد. پژوهش حاضر، سعی دارد با استفاده از تکنیک‌های محاسبات نرم، الگوریتمی ارائه کند که علاوه بر حداقل سازی وزن سازه، تمامی قیود آیین‌نامه نیز ارضا شود، علاوه بر آن از سرعت مناسبی نیز برخوردار باشد. در انتها نیز روش‌های مختلف محاسبات نرم، شامل الگوریتم ژنتیک بهبودیافته، الگوریتم مورچگان بهبودیافته و انواع مختلفی از شبکه‌های عصبی دولایه و سه‌لایه با روش‌های مختلف آموزش در روند بهینه‌سازی با یکدیگر و با روش رایج طراحی سازه مقایسه و نتایج آورده شده‌است که نتایج نشان از بهینه شدن قابل توجه وزن سازه در حدود ۱۴ تا ۳۳ درصد با استفاده از الگوریتم مورچگان و ژنتیک دارد و از بین این تکنیک‌ها سرعت و دقت بالای الگوریتم مورچگان بهبودیافته نسبت به باقی تکنیک‌های محاسبات نرم مشخص شده‌است. ضمناً برای تحلیل سازه‌ها، علاوه بر مدل‌سازی سه‌بعدی از روش تحلیل دینامیکی طیفی استفاده شده و روند بهینه‌سازی برای چهار نوع سازه، منظم، نامنظم در ارتفاع، نامنظم در پلان و نامنظم در پلان و ارتفاع صورت گرفته است تا حالت‌های مختلف مورد بررسی قرار گیرد.

۱- مقدمه

بهینه‌سازی در طول زمان به یک بخش ضروری از فعالیت‌های طراحی در همه رشته‌ها تبدیل شده‌است [۱]. بهینه‌سازی در سازه‌ها به‌منظور کمینه‌سازی متغیرهایی چون وزن و هزینه، با در نظر گرفتن مجموعه‌ای از محدودیت‌ها که به نام شرط شناخته می‌شوند، انجام می‌پذیرد [۲]. در مهندسی سازه یکی از زمینه‌های تحقیق، بهینه‌سازی طراحی ساختمان‌های تحت بارهای وارده می‌باشد. تکنیک‌های بهینه‌سازی در مهندسی سازه عموماً به سه روش تقسیم‌بندی می‌شود: ۱- روشی بر پایه ریاضیات یا گرادیان ۲- روش معیار بهینه بودن و ۳- الگوریتم‌های جستجوی تصادفی. [۳] تکنیک‌های ریاضی

بر پایه تغییرات تابع در فضای حل هستند. بنابراین الگوریتم به توابع پیوسته‌ای احتیاج دارد که هدف‌ها و محدودیت‌ها را نشان دهد. تحقیقات متعددی به‌وسیله روش‌های بهینه ریاضی انجام شده‌اند [۴-۸].

روش‌های موجود در محاسبات نرم برای بهینه‌سازی مسائلی مورد استفاده قرار می‌گیرد که حل آن با استفاده از روش‌های ریاضی یا امکان‌پذیر نیست و یا زمان و هزینه زیادی را صرف می‌کند. اصولاً تکنیک محاسبات نرم به مجموعه‌ای از روش‌ها و تکنیک‌ها گفته می‌شود که بر اساس هوش مصنوعی و یادگیری ماشین به‌منظور حل مسائل پیچیده بهینه‌سازی مورد استفاده قرار می‌گیرند. دلیل استفاده از محاسبات نرم برای بهینه‌سازی وزن سازه‌ها، توجه به این مسئله

* نویسنده عهده‌دار مکاتبات: habasian@kntu.ac.ir



۲- پیشینه تحقیق

سرعت رسیدن به پاسخ در الگوریتم‌های تکاملی همواره یکی از مسائل اصلی در بهینه‌سازی سازه‌ها بوده‌است. در سال ۲۰۰۹ نیز از و همکارانش [۹] به بهینه‌سازی هزینه ساخت قاب‌های فولادی به کمک الگوریتم ژنتیک پرداختند و توانستند تا حدودی به بهینه‌سازی آن کمک کنند. سپس قلی‌زاده و سماواتی در سال ۲۰۱۰ [۱۰] برای افزایش سرعت الگوریتم‌های تکاملی از شبکه‌های عصبی استفاده کردند و توانستند به موفقیت‌هایی دست یابند اما هم‌چنان بار محاسباتی بهینه‌سازی ساختمان به‌حدی بوده که محققان مجبور می‌شدند که ساده‌سازی‌های زیادی را وارد کار کنند. تحلیل دوبعدی ساختمان از جمله این موارد است. در این راستا در سال ۲۰۱۲ کاوه و طلعت‌اثری [۱۱] به کمک الگوریتم CSS به طراحی بهینه دوبعدی قاب‌های فولادی پرداختند. الگوریتم CSS با استفاده از قوانین حاکم بر فیزیک الکتروسیسته اقدام به جستجوی فضای مسئله می‌کند و توانسته به جواب‌های بهتری نسبت به الگوریتم قبلی برسد. یک سال بعد در سال ۲۰۱۳ کاوه و زاکیان [۱۲] الگوریتم CSS را با الگوریتم جستجو هارمونی ارتقاء یافته ترکیب کردند و توانستند به جواب‌های بهتری نسبت به حالت قبلی در طراحی بهینه سازه‌های فولادی برسند. در همین سال کوکیکی و عادل [۱۰] از الگوریتم ژنتیک دو فازی برای بهینه‌سازی سازه‌های سه‌بعدی سقفی فضاکار فولادی استفاده شده‌است. این سازه‌ها دو مورد از ۱۳ ایستگاه قطاری هستند که اعضای آن‌ها علاوه بر خمش و بارهای محوری، تحت اثر پیچش نیز قرار می‌گیرند. این بهینه‌سازی توانسته ۱۲ و ۷ درصد برای وزن این دو سازه کاهش داشته‌باشد.

در سال ۲۰۱۴ چین‌شنگ کائو و آی‌چنگ‌یه [۱۳] ابتدا تعداد زیادی ساختمان‌های مختلف طراحی کرده‌اند سپس با استفاده از اطلاعات به‌دست‌آمده یک شبکه عصبی مدل کرده‌اند که قادر به پیش‌بینی رفتار سازه تحت بارهای طراحی است. سپس با استفاده از الگوریتم‌های تکاملی، وزن این مدل تقریبی را بهینه کرده‌اند. این پژوهش علاوه بر این که از مدل تقریبی استفاده کرده، قاب فولادی را به صورت دوبعدی در نظر گرفته تا از بار محاسباتی کار بکاهد. در همین سال کاوه و زاکیان [۱۴] به کمک الگوریتم خفاش پیشرفته به طراحی بهینه سازه‌های خرپایی پرداختند. ایشان با در نظر گرفتن قیدهای محدود به حداقل کردن ابعاد مقاطع به‌کار رفته در سازه‌های

است که با روش‌های معمول نمی‌توان به حالت بهینه طراحی یک سازه رسید. به‌طور مثال، سازه‌ای را در نظر بگیرید که بیش از ۳۰۰ المان سازه‌ای (تیر و ستون) دارد؛ از طرفی مقاطع قابل قبول برای المان‌های این سازه تنها به ۴ عدد محدود شود، تعداد حالاتی که می‌توان برای طراحی این سازه در نظر گرفت، برابر ۴ به توان ۳۰۰ حالت است که چیزی بیش از 4×10^{180} حالت خواهد بود. واضح است که پیدا کردن حالت بهینه از بین این تعداد حالات، با روش‌های معمولی بسیار سخت و شاید غیرممکن باشد.

با توجه به لزوم طراحی بهینه و اقتصادی سازه‌ها، وزن نهایی سازه طراحی شده که رابطه مستقیم با هزینه ساخت دارد، مهم می‌شود. از طرفی با کاهش وزن سازه، پارامتر مقاومت کاهش می‌یابد و علاوه بر آن جابه‌جایی المان‌های سازه افزایش می‌یابد. با توجه به مسائلی که پژوهش‌های قبلی بررسی کرده‌اند و جایگاهی که سیستم قاب خمشی در ساختمان‌های فولادی دارد، به‌نظر می‌رسد بهینه‌سازی وزن سازه فولادی قاب خمشی توسط الگوریتم‌های تکاملی و بررسی ابزارهای دیگر محاسبات نرم برای رسیدن به جواب‌های دقیق‌تر با سرعت بالاتر، موضوعی قابل بررسی باشد. علاوه بر این، مدل‌سازی و طراحی سه‌بعدی ساختمان‌ها نیز با توجه به ابزارهای جدید محاسباتی و نرم‌افزارهایی چون MATLAB و SAP2000 قابل بررسی است و لزومی به مدل‌سازی دوبعدی و در نظر گرفتن فرض‌های ساده‌کننده، ایجادکننده خطا نمی‌باشد. یکی دیگر از اهداف اصلی این پژوهش، مقایسه عملکرد الگوریتم‌های مختلف تکاملی در مسائل سازه‌ای و دسته‌بندی الگوریتم‌ها از لحاظ بهینه بودن جواب، سرعت الگوریتم، تعداد تکرار کمتر و تعداد جمعیت اولیه است. به‌طور شفاف مقاله حاضر، به بررسی عملکرد ابزارهای محاسبات نرم شامل شبکه‌های عصبی، الگوریتم مورچگان و الگوریتم ژنتیک نامتوازن در طراحی سازه‌های فولادی و مقایسه این موارد با یکدیگر می‌پردازد. مراحل کار بدین ترتیب خواهد بود که ابتدا یک مدل ریاضی از مسئله تهیه شده، سپس با برقراری ارتباط بتن نرم‌افزارهای MATLAB و SAP2000 از طریق سیستم API از روش‌های محاسبات نرم استفاده می‌کند تا به ترکیب بهینه مقاطع برای ساختمان‌های قاب خمشی فولادی برسد. در نهایت پاسخ‌های بهینه ارائه‌شده به روش‌های مختلف، با یکدیگر و علاوه بر آن با پاسخی که نرم‌افزار SAP2000 به‌عنوان پاسخ بهینه ارائه می‌دهد، مقایسه خواهد شد.

تا نرم‌افزار مذکور به کمک الگوریتم‌های تکاملی چینی جدیدی از مقاطع را برای بررسی مجدد به نرم‌افزار 19 CSI SAP2000 ارائه می‌دهد. در واقع روند جستجو و انتخاب بر مبنای الگوریتم‌های تکاملی توسط MATLAB و تحلیل و بررسی قیود مقاومتی، تغییرشکل، شکل‌پذیری و نظایر آن، مندرج در آیین‌نامه توسط نرم‌افزار CSI SAP2000 19 صورت می‌گیرد.

۳- فرضیات سازه‌ای

این بخش به فرضیات موردنیاز برای استفاده از تکنیک‌های محاسبات نرم در مثال‌های سازه‌ای می‌پردازد. ساختمان‌هایی که در این مقاله برای نمونه استفاده شده‌اند، ساختمان‌های فولادی با قاب خمشی در هر دو جهت هستند. هدف بهینه کردن وزن این ساختمان‌ها با رعایت کردن قیود مطرح‌شده در آیین‌نامه‌های معتبر است. لازم‌به‌ذکر است که در این مقاله، یک بهینه‌سازی گسسته صورت گرفته‌است و برای طراحی صورت گرفته از آیین‌نامه‌های AISC360-2010 و آیین‌نامه بارگذاری ASCE7 استفاده شده‌است. هدف از بهینه‌سازی جاری، پیدا کردن بهترین مقطع از بین مقاطع احتمالی است. طبیعی است که دامنه مقاطع محتمل و قابل‌انتخاب برای بهینه‌سازی باید مشخص گردد. از آن‌جا که هدف بررسی قدرت روش‌های مختلف محاسبات نرم در بهینه‌سازی سازه‌ها است، انتخاب نوع مقاطع از اهمیت زیادی برخوردار نیست. در این راستا مقاطع محتمل ۳۲ عدد از مقاطع W به شرح جدول ۱ انتخاب شده‌است. پس به‌منظور بومی‌سازی تحقیق، استفاده از مقاطع متداول در ایران بر سایر مقاطع ترجیح دارد. با این‌همه، نتایج اولیه طراحی نشان داد که مقاطع رایج در برخی عناصر سازه‌ای جوابگوی نیازهای طراحی نیست. به‌عبارت دقیق‌تر، طراحی تیرورق در این موارد اجتناب‌ناپذیر می‌نمود. از آن‌جا که طراحی تیرورق پاسخ یکتا ندارد، خود می‌تواند به‌صورت یک مساله بهینه‌سازی مطرح شود که نتیجه نهایی آن، به وجود آمدن دو فرآیند تودرتوی بهینه‌سازی بود. این مساله باعث ایجاد پیچیدگی مضاعف در پیاده‌سازی برنامه واسط می‌شد. علاوه بر این، تمرکز محققین را از مساله بهینه‌سازی اصلی دور می‌نمود. در نهایت به نظر رسید چنانچه از مقاطع W استفاده شود نتایج نهایی را می‌توان به مقاطع رایج یا تیرورق‌هایی تقریباً معادل مقاطع مذکور تبدیل نمود. به‌طور خلاصه، استفاده از مقاطع W به دلایل زیر مورد توجه

خرپایی پرداختند که تا حدودی در این موضوع موفق شدند. فان و همکاران در سال ۲۰۱۵ [۱۵] با بهینه‌سازی یک قاب پرتال ساده سخت نوردشده توانست توانایی الگوریتم ژنتیک را نشان دهد. وی توانست به کمک الگوریتم ژنتیک تا ۵۳ درصد وزن مصالح مصرفی را کاهش دهد. اما الگوریتم ژنتیک در خصوص سازه‌های پیچیده دچار مشکل می‌شود. در همان سال نیز قلی زاده [۱۶] الگوریتم جدید کرم شب‌تاب اصلاح‌شده را برای بهینه‌سازی ساختمان به کار برد اما باز ناچار به استفاده از یک مدل تقریبی بود تا رفتار سازه را به کمک شبکه‌های عصبی پیش‌بینی کند. وی در پژوهشی دیگر [۱۷] نیز با ارائه الگوریتم گرگ خاکستری مجدداً به بهینه‌سازی مدل تقریبی پرداخت که به کمک شبکه عصبی RBF ساخته بود. از طرفی کارارو در سال ۲۰۱۶ با ارائه الگوریتم Search Group به مدل‌سازی قاب‌های دوبعدی فولادی پرداخت و توانست جواب‌های بهتری نسبت به پژوهش‌های پیشین کسب کند. با مرور پژوهش سابق در این حوزه، موارد زیر را می‌توان به‌عنوان نواقص و کاستی‌های آن‌ها نام برد:

- ساده‌سازی‌های گسترده به دلیل پرهیز از حجم محاسباتی بالا و افزایش سرعت رسیدن به جواب
- اکثر مدل‌سازی‌های صورت‌گرفته به علت کاهش حجم محاسبات به صورت دوبعدی صورت گرفته
- نادیده گرفتن بخشی از قیود حاکم بر طراحی سازه‌ها
- نادیده گرفتن بخشی از بارهای وارد بر سازه
- تیپ‌بندی سازه قبل از فرآیند طراحی به‌منظور کاهش متغیرهای طراحی و به تبع آن کاهش حجم محاسبات در موارد متعدد
- روند کار در این مقاله به این صورت خواهد بود که یک برنامه کامپیوتری در فضای نرم‌افزار MATLAB نوشته شده که در آن طراحی سازه به کمک الگوریتم‌های تکاملی توسط ترکیبی از MATLAB و CSI SAP2000 19 انجام شده‌است. در واقع، نرم‌افزار MATLAB داده‌های موردنیاز برای مدل‌سازی سازه از جمله ابعاد هندسی، مقطع‌المان‌های سازه‌ای، بارگذاری و نظایر آن را ایجاد می‌کند. سپس این موارد برای تحلیل و بررسی ارضاء قیود آیین‌نامه‌ای، به نرم‌افزار CSI SAP2000 19 منتقل خواهد شد. در ادامه نرم‌افزار CSI SAP2000 19 نتایج تحلیلی که صورت گرفته و وضعیت هر کدام از المان‌های سازه‌ای را به MATLAB ارائه می‌دهد

جدول ۱. لیست مقاطع استفاده شده در فرآیند بهینه‌سازی

Table 1. Sections Used in Optimizing Process

1) W8x10	2)W8x15	3)W8x21	4)W8x48	5)W10x12	6)W10x39	7)W10x100	8)W12x40
9) W12x87	10)W12x210	11)W14x30	12)W14x74	13)W14x159	14)W14x342	15)W14x665	16)W16x57
17)W18x50	18)W18x106	19)W18x234	20)W21x55	21)W21x122	22)W24x68	23)W24x146	24)W24x306
25)W27x146	26)W27x307	27)W30x124	28)W30x292	29)W33x169	30)W33x387	31)W36x332	32)W36x361

ضریب نامعینی برای هر دو جهت برابر ۱ در نظر گرفته خواهد شد.

۴- پیاده سازی الگوریتم‌های محاسبات نرم

۴-۱- حل مسئله به کمک الگوریتم ژنتیک بهبود یافته

در مسائل بهینه‌سازی که از الگوریتم‌های تکاملی استفاده می‌شود، غالباً دلیل اصلی برای استفاده از روش‌های فراابتکاری، پیچیده بودن تابع هدف و بزرگ بودن فضای پاسخ مسئله است به گونه‌ای که بررسی تمام دامنه پاسخ امکان‌پذیر نمی‌باشد. الگوریتم ژنتیک کلاسیک در وهله هم به بررسی قیود مسئله می‌پردازد و هم به سراغ محاسبه تابع هدف می‌رود. در واقع برای تمامی پاسخ‌ها در جمعیت هر نسل هم قیودات را محاسبه می‌کند و هم تابع هدف را و از این طریق سعی می‌کند به جواب بهینه برسد.

این مسئله در صورتی است که در طراحی بهینه سازه‌ها، بار محاسباتی بررسی قیود بسیار بیشتر از بررسی تابع هدف است. قیود حاکم بر مسئله شامل تمام قیود مندرج در آیین‌نامه اعم از قیود مربوط به مقاومت سازه و تغییرشکل‌ها، در حالت‌های حدی نهایی و بهره‌برداری می‌باشد که مشخصاً از حجم محاسباتی بسیار بالایی برخوردار است. از طرف دیگر تابع هدف برابر مجموع وزن المان‌های سازه‌ای است که به سرعت محاسبه می‌گردد.

در واقع در مقاله حاضر، با اعمال تغییراتی در روند الگوریتم ژنتیک، الگوریتمی مطرح شده‌است که صرفاً قیود را برای پاسخ‌هایی بررسی

قرار گرفت. نخست آن که استفاده از مقاطع W باعث کاهش تعداد جواب‌های ممکن می‌شد. علاوه بر این، از آنجا که تحلیل هر سازه و ارزیابی تابع هدف نیازمند زمانی قابل‌ملاحظه بود، استفاده از مقاطع W و کاهش تعداد جواب‌های ممکن، به کاستن حجم محاسبات و زمان مورد نیاز کمک شایان می‌نمود.

مشخصات مصالحی که در این مسئله فرض شده‌است در جدول ۲ آمده‌است. مقادیر موجود در جدول ۲ برای فولاد مصرفی از مبحث دهم مقررات ملی [۱۸] و برای بتن از مبحث نهم مقررات ملی [۱۹] استفاده شده‌است. مقادیری که برای بارهای وارد بر سازه در نظر گرفته شده، در جدول ۳ آمده‌است.

قیود حاکم بر بهینه‌سازی، تمامی قیود مندرج در آیین‌نامه AISC360-2010 می‌باشد که برای سازه‌های فولادی قاب خمشی مطرح گشته‌است. بهینه‌سازی، برای چهار نوع ساختمان، اعم از منظم، نامنظم در پلان، نامنظم در ارتفاع، نامنظم در ارتفاع و پلان، انجام شده‌است. این بخش به ویژگی‌های مشترک به در بین هر چهار ساختمان می‌پردازد. خاک منطقه محل احداث پروژه از نوع تیپ سه و ساختمان‌ها با اهمیت زیاد در نظر گرفته می‌شود. سقف سازه‌ها از نوع تیرچه بلوک در نظر گرفته شده‌است. برای اعمال نیروی زلزله به سازه، از تحلیل دینامیکی طیفی استفاده می‌شود. با توجه به این که از سیستم قاب خمشی استفاده شده و با فرض این که حداقل دو دهانه مقاوم در برابر بارهای جانبی در هر طرف مرکز جرم وجود دارد،

جدول ۲. مشخصات مصالح فرض شده

Table 2. The specifications of Material

تنش گسیختگی نهایی kg/m^2	تنش تسلیم kg/m^2	نسبت پواسون	مدول الاستیسیته kg/m^2	وزن واحد حجم kg/m^3	جرم واحد حجم kg/m^3
37×10^6	24×10^6	۰/۳	2×10^{11}	۷۸۵۰	۷۸۵۰
-	25×10^5	۰/۱۵	$2/495 \times 10^9$	۲۵۰۰	۲۵۰۰

شکل ۷: دستگاه آزمایش سطح شیبدار

Table 3. Loads on the structure

مقدار kgf/m^2	توضیحات	نوع بار وارده
۴۷۰	بار مرده سقف بام	الف) بارهای ثقلی
۵۱۰	بار مرده سقف طبقات	
۵۳۰ (kgf/m)	بار مرده دیوارهای پیرامونی	
۲۵۰	بارهای مرده دیوار جان‌پناه	ب) بارهای زنده
۱۱۵	بار معادل تیغه‌ها	
۱۵۰	بار زنده بام	
۲۰۰	بار زنده طبقات	ج) بارهای جانبی
-	بار جانبی حاکم بر ساختمان‌ها بار زلزله در نظر گرفته شده‌است. شتاب مبنای طرح برابر ۰/۳۵ نوع کاربری سازه‌ها مسکونی فرض شده در نتیجه ساختمان‌ها با اهمیت متوسط در نظر گرفته می‌شوند و مقدار ضریب رفتار برای هر دو جهت برابر ۵ منظور خواهد شد.	

این بند تا حدی که دیگر تأثیری در مرتب کردن جمعیت نداشته باشد: تفاوت الگوریتم ژنتیک بهبود یافته با الگوریتم ژنتیک کلاسیک در این بخش است. تابع هدف ثانویه در رابطه ۱ مشخص شده‌است. مانند:

$$C(X^k) = \begin{cases} W(X^k) + P(X^k) & \text{if } W(X^k) = \min\{W(X^i) | i \in \{1, 2, \dots, n_{pop}\}\} \\ W(X^k) & \text{if else} \end{cases} \quad (1)$$

که در آن X^k مسیری است که کروموزوم k ام پیشنهاد داده و n_{pop} تعداد جمعیت موجود است و تابع وزن المان‌های سازه‌ای که همان تابع هدف اولیه است و در رابطه ۲ تعریف می‌گردد:

$$W(X^k) = \sum_{i=1}^{n_e} A(X_i^k) L_i \gamma_i \quad (2)$$

که در رابطه فوق x_i^k مقطعی است که زن i ام از کروموزوم k ام برای المان سازه‌ای i ام، پیشنهاد کرده و $A(x_i^k)$ مساحت آن است. L_i طول المان i ام و γ_i مقدار وزن مخصوص آن هستند و نهایتاً n_e تعداد المان‌های سازه‌ای مسئله می‌باشد.

تابع هدف ثانویه این گونه تعریف می‌شود که برای بهترین پاسخ، یک تابع جریمه $P(X^k)$ به شرح زیر اضافه می‌گردد:

$$P(X^k) = W(X^k) \times V(X^k) \quad (3)$$

می‌کند که در جمعیت نسل جاری، به بهترین جواب رسیده‌باشد. این مهم با مطرح کردن نوعی تابع دو ضابطه‌ای برای تابع هدف و گنجاندن قیود مسئله در آن صورت می‌گیرد. در واقع قیود مسئله حذف گردیده و به‌طور غیرمستقیم در تابع هدف مطرح می‌گردد و در واقع مسئله جدیدی برای بهینه‌سازی مطرح می‌شود. مسئله جدید به‌گونه‌ای مطرح می‌گردد که قیدی برای بررسی نداشته و صرفاً از یک تابع هدف تشکیل می‌شود به‌گونه‌ای که اگر تابع هدف مسئله جدید بهینه گردد، هم تابع هدف برای مسئله اولیه بهینه شده و هم قیود حاکم بر مسئله اولیه ارضاء خواهد شد. مراحل الگوریتم ژنتیک بهبود یافته به قرار زیر است:

۱- انتخاب تصادفی جمعیت اولیه: جمعیت اولیه باید به صورت

تصادفی انتخاب شود اما حتماً باید تمامی اعضای جمعیت باید، تمامی قیود حاکم بر مسئله را ارضاء کند. منظور از جمعیت اولیه در این الگوریتم تعیین تصادفی المان‌های سازه است.

۲- مرتب کردن جمعیت بر اساس تابع هدف اولیه: در این

مرحله جمعیت موجود را بر اساس تابع هدف مورد نظر مسئله مرتب کرده و بهترین جواب این جمعیت مشخص می‌گردد. منظور از تابع هدف اولیه، همان تابع مورد نظر مسئله یعنی وزن است که باید بهینه گردد.

۳- مرتب کردن جمعیت بر اساس تابع هدف ثانویه و تکرار

جدیدی انتخاب می‌شوند.

۵- ادغام و تشکیل جمعیت نسل جدید: جمعیت اصلی،

فرزندان و جهش‌یافتگان با یکدیگر ادغام شده و بهترین‌ها انتخاب می‌شوند و باقی حذف خواهند شد، به گونه‌ای که تعداد جمعیت اولیه ثابت باقی بماند.

۶- اجرای مجدد موارد بالا در صورت عدم ارضای شرایط

خاتمه: مراحل فوق تا حدی تکرار می‌گردند که جواب مورد نظر به دست بیاید و شرایط خاتمه ارضاء گردد. شرایط خاتمه نیز می‌تواند انواع مختلفی داشته باشد.

پارامترهای الگوریتم ژنتیک کلاسیک و ارتقاء یافته نیز به این ترتیب اتخاذ گردید که نرخ لقاح برابر ۰/۶، نرخ جهش برابر ۰/۰۵، نرخ تاثیر جهش برابر ۰/۲ و فشار انتخاب برابر ۸ در نظر گرفته شده است.

۴-۲- حل مسئله به کمک الگوریتم کلونی مورچگان

ارتقاء یافته

در الگوریتم کلونی مورچگان، عامل‌های مسئله که مورچه‌ها هستند، بهترین مسیر را از بین تمامی مسیرهای موجود انتخاب می‌کنند. جهت شبیه‌سازی مسئله بهینه‌سازی ساختمان فولادی به کمک الگوریتم مورچگان مراحل زیر باید صورت می‌گیرد:

۱- آماده‌سازی: تمامی تنظیمات اولیه که برای شروع کار

الگوریتم لازم است، از جمله وارد کردن اطلاعات و فرضیات سازه‌ای و مقادیر اولیه برای شروع الگوریتم در این مرحله انجام می‌گیرد.

۲- حرکت مورچه‌ها: برای هر کدام از مورچه‌ها، ابتدا یک مسیر

با احتمال معکوس هزینه‌ها، ساخته می‌شود و پس از ساخته شدن، آن مسیر ارزیابی می‌گردد. در فرایند بهینه‌سازی این مقاله منظور از مورچه مقاطع سازه‌ای انتخاب شده و وزن آن‌ها می‌باشد.

۳- بررسی قیود: پس از آن که تمام مورچه‌ها مسیرهای خود را

طی کردند (به عبارت دیگر تمامی مقاطع سازه مشخص شد)، تا زمانی که قیدهای آیین‌نامه‌ای، توسط نرم‌افزار SAP2000 برای بهترین ترکیب المان‌ها چک نشده است، اقداماتی به ترتیب انجام می‌شود: الف) مسیری که بهترین مورچه طی کرده (المان‌هایی که انتخاب شده است)، توسط نرم‌افزار SAP2000 به سازه تبدیل می‌شود. ب) سازه‌ی ساخته شده، آنالیز و تحلیل می‌شود. ت) اگر قیودی از آیین‌نامه برآورده نشده بود، مقدار مشخصی جریمه، به هزینه بهترین

که در آن $V(X^k)$ مقدار تخطی از قیود آیین‌نامه است و به صورت

زیر تعریف می‌گردد:

$$V(X^k) = \begin{cases} \sum_m \sum_n \max(R_{mn} - 1, 0) > V_{\min} & \text{if } \exists m, n \in R_{mn} > 1 \\ 0 & \text{if else} \end{cases} \quad (4)$$

که در آن R_{mn} نسبت مقدار مورد نیاز به مقدار مجاز تلاش l_m

از المان سازه‌ای l_m و V_{\min} برابر است با:

$$V_{\min} = \frac{(AL\gamma)_{\min}}{(AL\gamma)_{\max}} + 1 \quad (5)$$

بعد از اعمال جریمه فوق، پاسخ‌های موجود مجدداً مرتب گردیده و در صورت تغییر بهترین پاسخ، مجدداً برای بهترین پاسخ روند بالا انجام می‌شود. این روند تا زمانی تکرار خواهد شد که یا بهترین پاسخ جمعیت قیود را ارضاء کند یا تمامی اعضای جمعیت، بررسی شده باشند.

در واقع بعد از اتمام این مرحله، دو حالت ممکن برای جمعیت وجود دارد؛ یا این که تمامی پاسخ‌های جمعیت قیود را ارضاء نمی‌کنند یا این حالت به وجود می‌آید که یک عضو جمعیت علاوه بر ارائه بهینه‌ترین پاسخ، قیود مسئله را ارضاء می‌کند و باقی جمعیت یا قیود را ارضاء نکرده‌اند یا از این منظر بررسی نشده‌اند.

با این روش در واقع صرفاً قیود مسئله برای جواب‌هایی مطرح می‌شود که در روند بالا حداقل یک بار، بهترین جواب ممکن باشد و برای همه پاسخ‌ها به سراغ محاسبات وقت‌گیر بررسی قیود رجوع نمی‌شود. البته با نگاهی به عرصه طبیعت که خاستگاه اصلی الگوریتم ژنتیک است مشاهده می‌شود نظام مقایسه در توالی و رشد نسل‌ها در طبیعت نیز این‌گونه نیست که همه گروه به‌طور یکسان رقابت کنند و چه بسا نژادهای قوی‌تر مورد امتحان سخت‌تری نیز واقع می‌گردند. در واقع در الگوریتم ژنتیک نامتوازن نیز این معنا به گونه‌ای اعمال شده است و صرفاً پاسخ‌هایی مورد بررسی بیشتر و سخت‌گیرانه‌تری می‌شوند که در آزمون اولیه بهتر ظاهر شده‌اند و جواب‌های بهینه‌تری ارائه داده‌اند.

۴- انجام عمل لقاح و جهش: در این مرحله، از بین

کروموزوم‌های موجود، با احتمال عکس هزینه آن‌ها والدین انتخاب شده و برای ایجاد جمعیت فرزندان ترکیب می‌شوند. از طرفی اعضای از جمعیت برای اعمال جهش انتخاب شده و با احتمالی اعضای

باقی متغیرها نیز همان طور هستند که قبلا تعریف شده‌اند. در واقع نحوه انجام کار به این صورت خواهد بود که ابتدا وزن تمامی سازه‌های انتخابی مورچه‌ها مشخص می‌شود، سپس قیود آیین‌نامه‌ای برای کم‌هزینه‌ترین سازه، توسط نرم‌افزار SAP2000 چک می‌شود، آنگاه اگر قیود آیین‌نامه‌ای برقرار شد، الگوریتم به مرحله بعد می‌رود، در غیر این صورت، هزینه مربوط به مورچه مذکور دچار جریمه شده و این فرآیند برای مورچه دوم (که در حال حاضر کمترین هزینه دارد) تکرار می‌شود تا به حالتی برسد که یا مورچه‌ای پیدا شود که کمترین هزینه را داشته باشد و قیود آیین‌نامه‌ای را ارضاء کند یا تمامی مورچه‌های موجود از لحاظ قیود آیین‌نامه‌ای بررسی گردد، آنگاه الگوریتم به مرحله بعدی خواهد رفت.

در مرحله بعد، مورچه‌هایی که از لحاظ آیین‌نامه‌ای چک نشده‌اند، یک‌میزان جریمه متحمل می‌شوند تا نسبت به مورچه‌ای که کمترین هزینه را داشته و قیود آیین‌نامه را ارضاء کرده، عقب‌تر باشند. باید توجه کرد که مورچه‌هایی که قیود آیین‌نامه‌ای را ارضاء نمی‌کنند، بعد از اعمال جریمه، میزان هزینه مربوط به آن‌ها، از حداکثر وزن ممکن برای سازه نیز بیشتر شود. برای اطمینان از این موضوع، پارامتری به نام V_{min} عریف می‌شود که حداقل ممکن وزن سازه را به حداکثر آن تبدیل کند. با این شرط مطمئن می‌توان مطمئن شد که مورچه‌هایی که قیود مسئله را ارضاء نمی‌کنند، بدون شک از تمامی مورچه‌هایی که قیود مسئله را ارضاء می‌کنند، هزینه بیشتری خواهند داشت.

۴-۳- حل مسئله به کمک شبکه‌های عصبی

در نظر بگیرید که اطلاعات مشخصی از سازه‌های مختلف، شامل مشخصات مصالح مصرفی، ضخامت سقف بتنی، اطلاعات بارگذاری اعم از بارهای ثقلی طبقات و بام و بارهای جانبی، مقاطع مورد استفاده برای المان‌ها و در نهایت مقدار Ratio برای هر المان موجود باشد. در صورت وجود این اطلاعات به‌میزان کافی، می‌توان شبکه عصبی را آموزش داد که برای مقادیر مختلف Ratio در یک سازه خاص با بارگذاری مشخص، مقاطعی که باید برای المان‌های سازه‌ای انتخاب شود را مشخص کند. واضح است که شبکه عصبی توجهی به مفاهیم فیزیکی و معادلات حاکم بر یک سازه ندارد و صرفاً با تجربه‌ای که از آموزش داده‌های موجود به‌دست آورده، مقاطع را حدس می‌زند. اگر بتوان، شبکه عصبی مصنوعی را آموزش داد که بتواند مسائل

مورچه اضافه می‌شود. ث) مجدداً مسیر با کمترین هزینه انتخاب شده و قیود آیین‌نامه برای آن مورد بررسی قرار می‌گیرد.

۴- به‌روزرسانی فرم‌ها: پس از آن که شرط قسمت سوم برآورده شد: الف) مورچه‌هایی که قیود آیین‌نامه‌ای برای آن‌ها چک نشده‌است، مقدار مشخصی جریمه (کمتر از مقدار جریمه قبلی) به مقدار هزینه آن‌ها اضافه می‌شود. ب) هر مورچه بر روی یال‌هایی که از آن‌ها عبور کرده‌است، به نسبت عکس هزینه خود، فرمومون می‌ریزد. ت) فرمومون تمام یال‌ها، در اثر تبخیر کمتر می‌شود.

۵- چک کردن شرایط خاتمه: در صورت نیاز (برآورده نشدن شرایط خاتمه) از بند دوم، این فرآیند تکرار می‌شود.

۶- تابع هدف در الگوریتم مورچگان: در مسئله بهینه‌سازی سازه‌ها که مورد بحث است، تابع هدف عبارت است از وزن المان‌های سازه‌ای، که به راحتی به دست می‌آید، اما بررسی قیود حاکم، که شامل الزامات آیین‌نامه‌ای است و توسط برنامه SAP2000 صورت می‌گیرد، امری بسیار زمان‌بر است. در واقع در این مورد برای صرفه‌جویی در وقت باید نسبت به کاهش تعداد دفعات بررسی قیود اقدام لازم صورت گیرد. یعنی صرفاً قیود حاکم برای مواردی بررسی شود که مقدار تابع هدف را بهینه می‌کند. برای این منظور، تابع هدف به صورت زیر تعریف می‌شود:

$$C(X^k) = \begin{cases} W(X^k) + P_1(X^k) & \text{if } W(X^k) = \min\{W(X^i) | i \in \{1, 2, \dots, n_{pop}\}\} \\ W(X^k) + P_2(X^k) & \text{if else} \end{cases} \quad (6)$$

که در آن $P_1(X^k)$ تابع جریمه‌ای است که به مورچه‌هایی (مقاطع سازه) تعلق می‌گیرد که کمترین میزان هزینه را نداشته باشند (این مسئله در هر بار تکرار بررسی می‌گردد) و مقدار آن برابر است با:

$$P_1(X^k) = W(X^k) \times V(X^k) \quad (7)$$

و $P_2(X^k)$ تابع جریمه‌ای است که به مورچه‌هایی تعلق می‌گیرد که در تکرار مدنظر، حتی یک بار کمترین میزان هزینه را نداشته و مقدار آن برابر است با:

$$P_2(X^k) = 0.5 \times P_1(X^k) \quad (8)$$

ستون طبقه ششم، یکسان در نظر گرفته می‌شود. با این حساب ۱۴ تیپ ستون خواهد داشت که مجموع تیپ‌های نهایی برابر ۳۲ عدد خواهد بود. در واقع شبکه عصبی ۳۲ خروجی خواهد داشت که عبارت است از شماره مقطعی که برای هر کدام از این ۳۲ تیپ استفاده شده است.

کاری که پس از آموزش شبکه عصبی صورت می‌گیرد، درخواست شماره مقاطع برای سازه مورد نظر در صورتی که مقادیر میانگین حسابی، میانگین هندسی، میانگین هارمونیک و میانه Ratioها برابر ۰/۹۵، بازه تغییرات آن (۰/۰۵+ و ۰/۰۵-)، بیشترین مقدار آن برابر یک و کمترین مقدار آن برابر ۰/۹ باشد. با توجه به اینکه آموزش شبکه عصبی با سطح اطمینان ۹۵٪ انجام شده است لذا با ۹۵٪ اطمینان می‌توان مطمئن بود که خروجی شبکه عصبی سازه ای با حالت بهینه باشد.

شبکه عصبی مورد نظر از نوع پرسپترون دو لایه (یک لایه پنهان) و سه لایه (دو لایه پنهان) انتخاب شده و با ۲۲ ورودی، ۵۰ نورون در لایه‌های پنهانی، ۳۲ نورون در لایه خروجی و ۳۲ خروجی در نظر گرفته شده است. ۷۰ درصد داده‌ها، به‌عنوان داده آموزش، ۱۵ درصد به‌عنوان داده آزمایش و ۱۵ درصد نیز به‌عنوان داده‌های اعتبارسنجی منظور شده‌اند.

۲- جمع‌آوری اطلاعات اولیه مورد نیاز شبکه عصبی: برای

ساختن یک شبکه عصبی، وجود اطلاعات اولیه به‌میزان کافی الزامی است. اطلاعاتی که به کمک آن‌ها، شبکه عصبی آموزش ببیند. برای این منظور از نرم‌افزارهای MATLAB و SAP2000 استفاده شده است، بدین صورت که نرم‌افزار MATLAB، به‌صورت تصادفی، ضخامت سقف طبقات، شتاب مبنای طرح، ضریب اهمیت ساختمان، دیگر پارامترهای مربوطه به بارگذاری زلزله، بار دیوارها، بار مرده طبقات و بام، بارهای زنده و در نهایت مقطع هر المان را مشخص می‌کند، در واقع اطلاعات مربوط به سازه را می‌سازد و به نرم‌افزار SAP2000 ارائه می‌دهد. نرم‌افزار SAP2000 بعد از انجام آنالیز و چک کردن قیود طراحی، اطلاعات هر کدام از المان‌ها که در واقع همان Ratio آن‌ها می‌باشد را به نرم‌افزار MATLAB ارائه می‌دهد تا این اطلاعات ذخیره شود. این فرآیند به تعداد هزار بار برای هر کدام از سازه‌ها تکرار می‌گردد، بدین ترتیب اطلاعات اولیه برای آموزش شبکه عصبی به‌دست می‌آید.

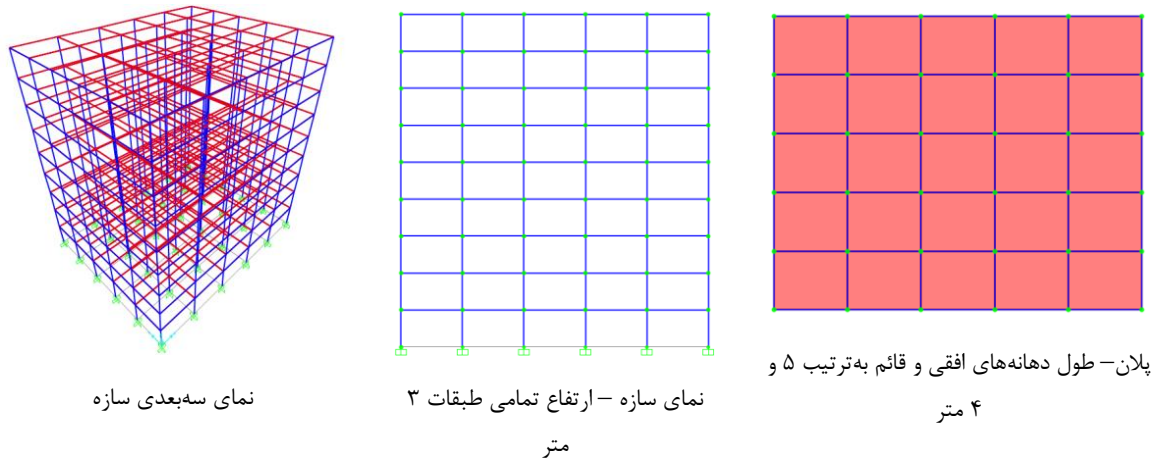
فوق‌الذکر را با دقت مناسبی انجام دهد، آنگاه کافی است میزان Ratio برای تمامی المان‌ها تقریباً برابر یک قرار داده شود، تا احتمالاً مقاطع پیشنهادی شبکه عصبی، بهینه باشد.

۱- داده‌های ورودی و خروجی شبکه عصبی: در واقع هدف اصلی به‌دست آوردن شبکه عصبی است که بتواند با کمترین میزان خطا، با در دست داشتن، مشخصات سازه، مشخصات بارگذاری و دیگر موارد برای یک میزان ضریب اطمینان مشخص، مقاطع المان‌های سازه را تخمین بزند. برای این مقصود داده‌های ورودی شبکه عصبی به‌ترتیب زیر خواهد بود:

۱) ضخامت سقف طبقات، به‌صورت یک عدد تصادفی بین ۱۰ تا ۲۵ سانتی‌متر. ۲) شتاب مبنای طرح، ضریب اهمیت ساختمان، پارامترهای T0 و S از مقادیر مندرج در آیین‌نامه به‌صورت تصادفی. ۳) اطلاعات بارگذاری اعم از بارهای دیوارهای بام، بار مرده دیوارهای طبقات، بار مرده کف طبقات، بار مرده کف بام، بار زنده طبقات، بار زنده بام به‌صورت تصادفی از بازه رایج برای هر کدام از انواع بارها و در نهایت ۴) اطلاعات آماری مربوط به Ratioها که شامل میانگین حسابی، میانگین هندسی، میانگین هارمونیک، میانه، میزان تغییرات، انحراف معیار، بیشینه و کمینه آن می‌باشد. با این حساب تعداد ورودی‌های شبکه عصبی برابر ۲۲ عدد خواهد بود.

پارامترهای مربوطه به ضرایب اطمینان، با اقدامات آماری بر روی مقادیری به‌دست می‌آید که نرم‌افزار SAP2000 ارائه می‌دهد. لازم به‌ذکر است که نرم‌افزار مذکور، معکوس ضرایب اطمینان را به‌عنوان Ratio می‌دهد که مفهوم نسبت میزان موجود تلاش، به میزان مجاز آن را دارد. دلیل استفاده از مقادیر آماری فوق، کاهش تعداد ورودی‌های شبکه عصبی است. در غیر این صورت تعداد ورودی‌های شبکه عصبی بسیار زیاد خواهد شد و سیستم‌های موجود قادر نیستند محاسبات لازم را انجام دهند.

به‌دلیلی که در بالا به آن اشاره شد، تعداد خروجی‌ها نیز باید کاهش یابد. برای این منظور از مفهوم تیپ‌بندی المان‌های سازه‌ای استفاده می‌شود. تیپ‌بندی به‌این ترتیب صورت می‌گیرد که تیرها به دو دسته تیر کناری و تیر داخلی تقسیم می‌شود که با توجه به تعداد طبقات، ۱۸ تیپ تیر وجود خواهد داشت. ستون‌ها نیز به دو دسته ستون کناری و ستون داخلی تقسیم‌بندی می‌شود و علاوه بر آن ستون‌های طبقه نهم با ستون‌های طبقه هشتم و ستون طبقه هفتم با



شکل ۱. سازه (الف) - منظم در پلان و ارتفاع
Fig. 1. Structure (A)- Regular in plan and height

۵- تحلیل و بررسی نتایج

۵-۱- سازه منظم در پلان و ارتفاع

در مورد شبکه‌های عصبی، هشت نوع مختلف مشتمل بر پرسپترون دولایه و سه لایه با روش‌های آموزش Levenberg-Marquadet، نیوتون، گرادیان شیب و اندازه حرکت مورد بررسی قرار گرفت و از میان آن‌ها شبکه عصبی پرسپترون سه لایه با روش آموزش Levenberg-Marquadet بهترین پاسخ را ارائه داد که نتایج آن در جدول ۴ آمده است و از آوردن نتایج باقی حالت‌ها به جهت طولانی نشدن مقاله، خودداری شد. در شبکه‌های عصبی برای هر چهار نوع سازه، قیود مسئله مورد بررسی قرار نمی‌گیرد و صرفاً به این مقدار بسنده می‌شود که میانگین، ماکزیمم و مینیمم نسبت‌ها در بازه مطلوب باشد. به همین دلیل، امکان دارد پیشنهادی که شبکه عصبی ارائه می‌دهد بعضی از قیود مسئله را ارضاء نکند، به همین خاطر بعضی از المان‌ها در مدل پیشنهادی شبکه عصبی مقادیر مجاز آیین‌نامه را رعایت نمی‌کنند که در شکل ۲ قابل مشاهده است. با توجه به جداول و شکل‌های مربوط به این بخش، نتایج زیر

سازه‌ای که در این بخش بررسی می‌گردد، یک ساختمان ۹ طبقه است، که ارتفاع هر طبقه ۳ متر می‌باشد. کاربری سازه مورد نظر مسکونی در نظر گرفته شده است. مساحت هر طبقه برابر ۵۰۰ متر مربع است. کلیه طبقات تیپ هستند و سازه در ارتفاع و پلان منظم است. سیستم ساختمانی در هر دو جهت قاب خمشی متوسط بوده و تمامی اتصالات آن‌ها گیردار هستند. در ادامه در جداول و شکل‌های مختلف، نتایج حاصل از تکنیک‌های محاسبات نرم به همراه نتایج حاصل از طراحی خودکار نرم‌افزار SAP2000 مقایسه شده است. مقایسه روش الگوریتم ژنتیک بهبودیافته و الگوریتم مورچگان بهبودیافته از منظر نرخ همگرایی، تعداد تکرار و دیگر موارد در جدول ۵ و شکل ۳ آمده است.

جدول ۴. نتایج حاصل از چهار روش برای سازه منظم در پلان و ارتفاع

Table 4. Results of four methods for regular structures in plan and height

بیشترین مقطع استفاده شده	کمترین مقدار Ratio	بیشترین مقدار Ratio	انحراف معیار Ratio	میانگین Ratio	تعداد مقاطع انتخابی	درصد تغییرات وزن نسبت به SAP	وزن بر واحد سطح (کیلوگرم بر متر مربع)	وزن کل سازه (تن)	
W18x50	۰/۳۹	۰/۹۵	۰/۱۵	۰/۷۲	۲۰	۰	۷۸/۲	۳۵۲	SAP2000
W18x50	۰/۲	۰/۹۸	۰/۱۴	۰/۶۵	۲۱	- ۲/۹	۷۵/۹۲	۳۴۲	ژنتیک ارتقاء یافته
W8X48	۰/۶۲	۰/۹۵	۰/۰۷	۰/۸۳	۲۱	- ۱۴/۷	۶۶/۶۹	۳۰۰	مورچگان ارتقاء یافته
W18x50	۰/۲	۱/۱۹	۰/۲	۰/۶۶	۱۱	- ۰/۸	۷۷/۶۷	۳۴۹	شبکه عصبی

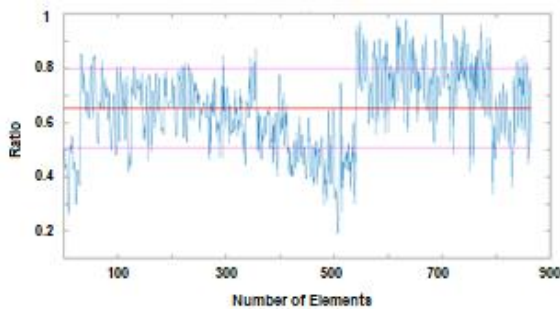
جدول ۵. نتایج مختص الگوریتم‌های تکاملی برای سازه منظم در پلان و ارتفاع
Table 5. Results of Applying Evolutionary Algorithms for regular structures in plan and height

تعداد جمعیت	تعداد فراخوانی تابع	درصد فراخوانی تابع	تعداد فراخوانی تابع	تعداد فراخوانی تابع	تعداد تکرار	
اولیه	به فراخوانی تابع وزن	نرم افزار SAP	وزن	وزن		
۱۰	۳۲/۸۶	۸۹۲۲	۲۷۱۶۴	۲۷۱۶۴	۴۳۳۶	ژنتیک ارتقاء یافته
۱۰	۶۹/۲۷	۶۱۱۷	۸۸۳۰	۸۸۳۰	۸۸۳	مورچگان ارتقاء یافته

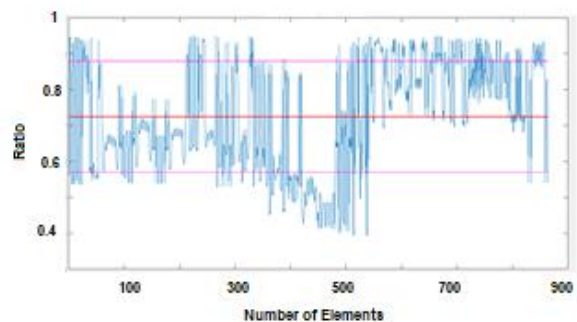
شده است که به طور تقریبی می توان گفت الگوریتم ژنتیک ارتقاء یافته توانسته بیش از ۶۰ درصد در زمان بهینه سازی صرفه جویی نماید. همین مسئله با نسبت های کمتر برای الگوریتم مورچگان برقرار است. نتایج حاصل از الگوریتم ژنتیک ارتقاء یافته و الگوریتم مورچگان ارتقاء یافته، نشان از کاهش ۲/۹ و ۱۴۷/۷ درصدی در وزن سازه و به تبع آن هزینه ها داشته است که میزان قابل توجهی است. - بیشترین میزان میانگین نسبت ها و کمترین میزان انحراف معیار، برای بهینه ترین جواب، یعنی جواب الگوریتم مورچگان است چرا که هر چه میانگین به عدد یک نزدیک تر شود به نحو بهتری از

حاصل گردید:

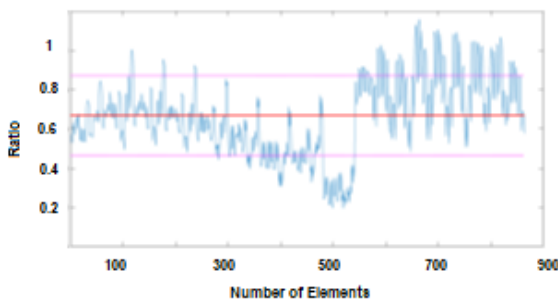
- کمترین میزان وزن سازه مربوط به الگوریتم مورچگان ارتقاء یافته است که به بهینه ترین جواب و کمترین وزن بین روش های موجود رسیده است. - درصد فراخوانی تابع آنالیز به فراخوانی تابع وزن در الگوریتم ژنتیک برابر ۳۲/۸۶ درصد به دست آمد که نشان از تاثیر قابل توجه ارتقایی دارد که در الگوریتم ژنتیک داده شده است. در واقع از تمامی مواردی که توسط الگوریتم ژنتیک ارتقاء یافته مورد بررسی قرار گرفته است، تنها برای ۳۲/۸۶ درصد آن ها، تابع بررسی قیود فراخوانی



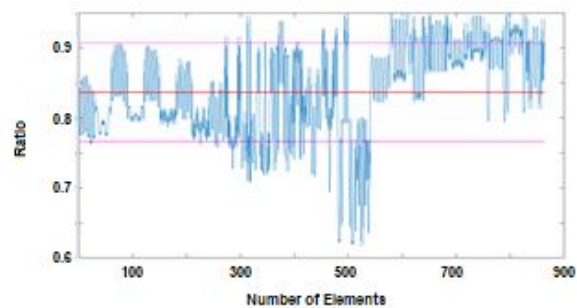
نتایج الگوریتم ژنتیک ارتقاء یافته



نتایج نرم افزار SAP2000



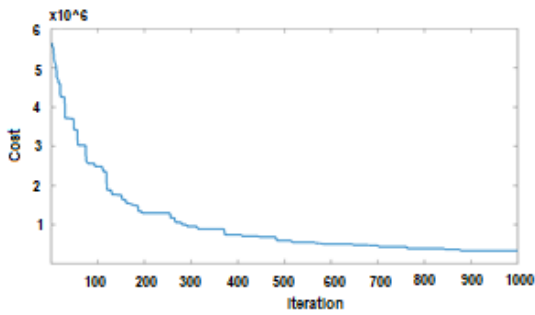
نتایج شبکه های عصبی



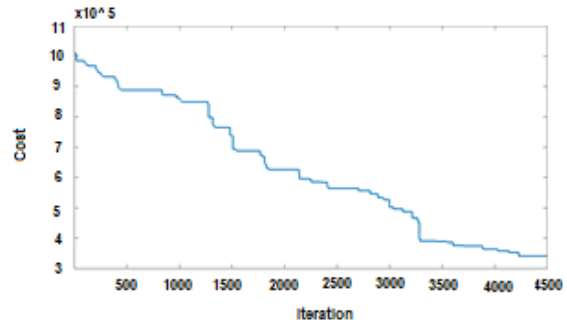
نتایج الگوریتم مورچگان ارتقاء یافته

شکل ۲. مقادیر Ratio برای هر المان سازه ای در نتایج نهایی روش های مختلف برای سازه منظم در پلان و ارتفاع (در تمامی اشکال محور افقی تعداد تکرار و محور قائم Ratio را نشان می دهد. خط قرمز متوسط مقدار Ratio را نشان می دهد).

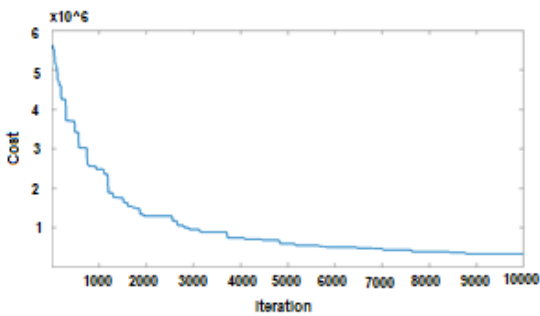
Fig. 2. Ratio values for each element in different methods for regular structures in plan and height



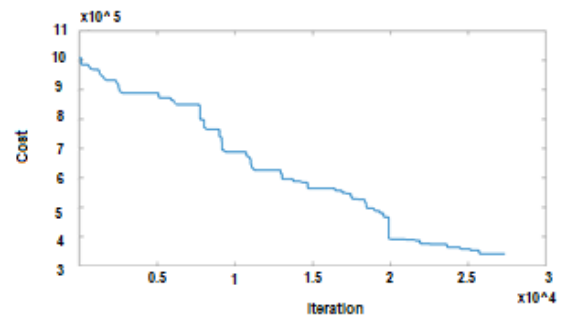
روند کاهش هزینه بر حسب تعداد تکرار در الگوریتم مورچگان ارتقاء یافته



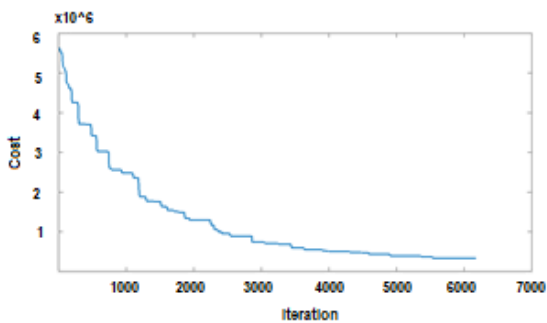
روند کاهش هزینه بر حسب تعداد تکرار در الگوریتم ژنتیک ارتقاء یافته



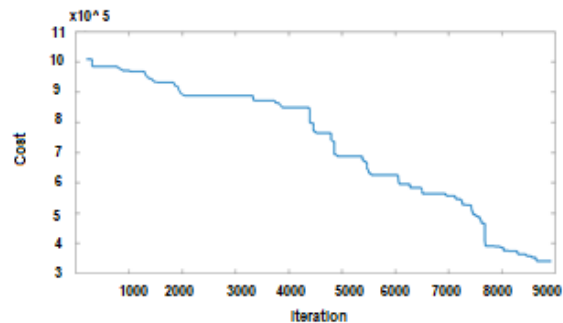
روند کاهش هزینه بر حسب تعداد فراخوانی تابع وزن



روند کاهش هزینه بر حسب تعداد فراخوانی تابع وزن



روند کاهش هزینه بر حسب تعداد فراخوانی تابع SAP2000



روند کاهش هزینه بر حسب تعداد فراخوانی تابع SAP2000

شکل ۳. روند کاهش تابع هزینه در الگوریتم‌های تکاملی برای سازه منظم در پلان و ارتفاع (در تمامی اشکال فوق محور افقی تعداد تکرار و محور قائم هزینه می باشد).

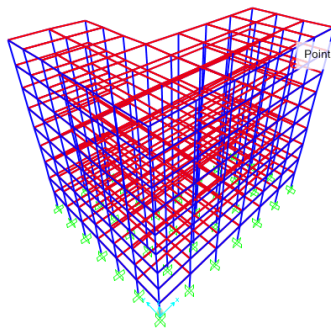
Fig. 3. The process of reducing the cost function in evolutionary algorithms for regular structures in plan and height

همان‌طور که مشاهده می‌کنید جواب حاصل از شبکه عصبی پرسپترون نسبت‌هایی بزرگ‌تر از یک به دست آورده به طوری که بیشترین مقدار آن‌ها برابر ۱/۱۹ می‌باشد. این مسئله از ابتدا قابل پیش‌بینی بود چرا که هیچ معیار دقیق و متقنی برای شبکه عصبی فرض نشده بود تا از افزایش نسبت‌ها به بیشتر از یک خودداری کند.

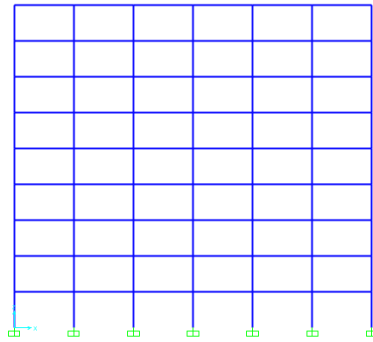
الگوریتم مورچگان توانسته کمترین مقدار Ratioها را تا مقدار ۰/۶۲ افزایش دهد که نشان می‌دهد توانسته از بخش اعظمی از

ظرفیت مقاطع استفاده شده و هر چه انحراف معیار کاهش یابد نشان از پراکنده نبودن نسبت‌ها و متمرکز بودن آن‌ها حول میانگین دارد.

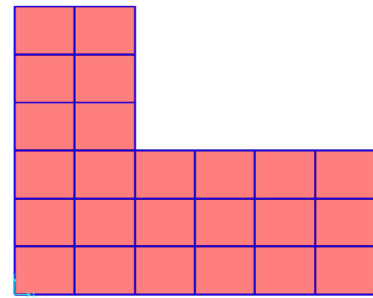
تعداد مقاطع انتخابی بین نرم‌افزار SAP2000، الگوریتم ژنتیک و الگوریتم کلونی مورچگان تقریباً برابر بوده و همه این موارد تعداد مشخصی مقطع را برای طرح خود در نظر گرفته‌اند، اما شبکه عصبی پرسپترون، به دلیل این که در المان‌ها تیپ‌بندی صورت داده است، صرفاً ۱۱ المان را به کار گرفته است.



نمای سه بعدی سازه



نمای سازه- ارتفاع تمامی طبقات ۳ متر



پلان سازه- طول دهانه های افقی و قائم به ترتیب ۴ و ۵ متر

شکل ۴. سازه (ب) - نامنظم در پلان و منظم در ارتفاع

Fig. 4. Structure (B)-Irregular in plan and regular in height

تکنیک های محاسبات نرم به همراه نتایج حاصل از طراحی خودکار نرم افزار SAP2000 برای سازه نامنظم در پلان و منظم در ارتفاع مقایسه شده است. مقایسه روش الگوریتم ژنتیک بهبودیافته و الگوریتم مورچگان بهبودیافته از منظر نرخ همگرایی، تعداد تکرار و دیگر موارد در جدول ۷ و شکل ۶ آمده است.

با توجه به نتایج سازه منظم در پلان و ارتفاع، برای این بخش نیز شبکه عصبی پرسپترون سه لایه با روش آموزش Levenberg-Marquadt انتخاب شد و نتایج ارائه شده بر این مبنا است. با توجه به جداول و شکل های مربوط به این بخش، نتایج زیر حاصل گردید:

- مشابه قسمت قبل، کمترین میزان وزن سازه مربوط به الگوریتم مورچگان ارتقاء یافته است که به بهینه ترین جواب و کمترین وزن بین روش های موجود رسیده است.

- درصد فراخوانی تابع آنالیز به فراخوانی تابع وزن در الگوریتم

ظرفیت مقاطع همه المان ها استفاده کند.

- تعداد تکرار الگوریتم ژنتیک تفاوت فاحش و معناداری با تعداد تکرار در الگوریتم مورچگان دارد. این مسئله از نمودارهای مربوط به این دو الگوریتم نیز مشخص بود. در کال الگوریتم ژنتیک نسبت به الگوریتم مورچگان به مراتب از سرعت کمتری برخوردار است.

۵-۲- سازه نامنظم در پلان و منظم در ارتفاع

سازه ای که در این بخش بررسی می گردد، یک ساختمان ۹ طبقه است، که ارتفاع هر طبقه ۳ متر می باشد. کاربری سازه مورد نظر مسکونی در نظر گرفته شده است. مساحت هر طبقه برابر ۴۸۰ مترمربع است. کلیه طبقات تیپ هستند و سازه در ارتفاع منظم و در پلان نامنظم است. سیستم ساختمانی در هر دو جهت قاب خمشی متوسط بوده و تمامی اتصالات آن ها گیردار هستند. در ادامه در جداول و شکل های مختلف، نتایج حاصل از

جدول ۶. نتایج حاصل از چهار روش برای سازه نامنظم در پلان و منظم در ارتفاع

Table 6. Results of four methods for irregular structures in plan and regular in height

بیشترین مقدار استفاده شده	کمترین مقدار Ratio	بیشترین مقدار Ratio	انحراف معیار Ratio	میانگین Ratio	تعداد مقاطع انتخابی	درصد تغییرات وزن نسبت به SAP	وزن بر واحد سطح (کیلوگرم بر مترمربع)	وزن کل سازه (تن)	
W18x50	۰/۲	۰/۹۶	۰/۱۷	۰/۶۲	۱۶	۰	۱۴۳/۸۹	۶۲۱	SAP2000
W10x100	۰/۴۲	۰/۹۱	۰/۰۸	۰/۶۷	۲۱	- ۳/۷	۱۳۷/۶۱	۵۹۸	ژنتیک ارتقاء یافته
W18x50	۰/۳۳	۰/۹۶	۰/۱۲	۰/۷۴	۲۲	- ۳۳/۳	۹۶/۰۳	۴۱۴	مورچگان ارتقاء یافته
W18x50	۰/۲	۱/۲	۰/۲	۰/۶۵	۱۱	- ۴۴/۶	۷۹/۲	۳۴۴	شبکه عصبی

جدول ۷. نتایج مختص الگوریتم‌های تکاملی برای سازه نامنظم در پلان و منظم در ارتفاع

Table 7. Results of Applying Evolutionary Algorithms for irregular structures in plan and regular in height

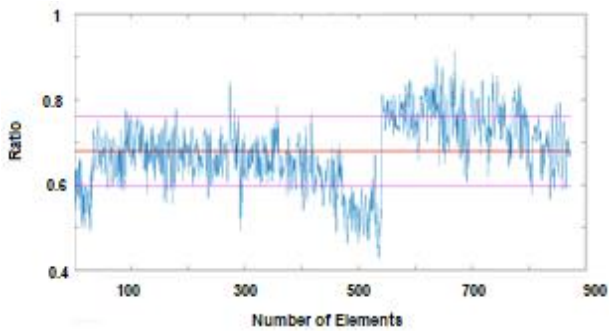
تعداد جمعیت اولیه	درصد فراخوانی تابع آنالیز به فراخوانی تابع وزن	تعداد فراخوانی نرم افزار SAP	تعداد فراخوانی تابع وزن	تعداد تکرار	ژنتیک ارتقاء یافته
۱۰	۳۲٫۸۶	۸۹۲۲	۲۷۱۶۴	۴۳۳۶	ژنتیک ارتقاء یافته
۱۰	۶۹/۲۷	۶۱۱۷	۸۸۳۰	۸۸۳	مورچگان ارتقاء یافته

۵-۳- سازه منظم در پلان و نامنظم در ارتفاع

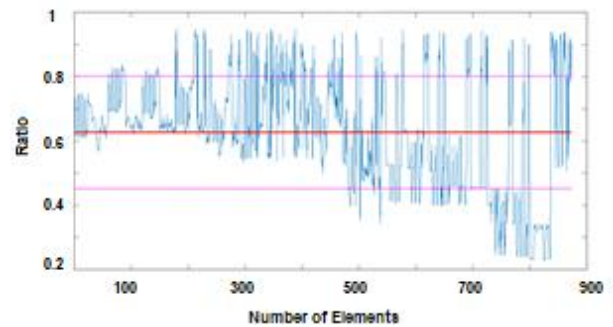
سازه‌ای که در این بخش بررسی می‌گردد، یک ساختمان ۹ طبقه است، که ارتفاع طبقه اول و دوم ۴ متر و باقی طبقات ۳ متر می‌باشد. کاربری سازه مورد نظر مسکونی در نظر گرفته شده است. مساحت هر طبقه برابر ۵۰۰ متر مربع است. کلیه طبقات تیپ هستند و سازه در ارتفاع نامنظم و در پلان منظم است. سیستم ساختمانی در هر

ژنتیک ارتقاء یافته مشابه حالت قبل به دست آمده و برابر ۳۲/۸۶ درصد است، در نتیجه مؤید تاثیر تغییراتی است که در الگوریتم ژنتیک صورت گرفته است. همچنین این مسئله برای الگوریتم مورچگان ارتقاء یافته نیز به نسبت کمتر برقرار است.

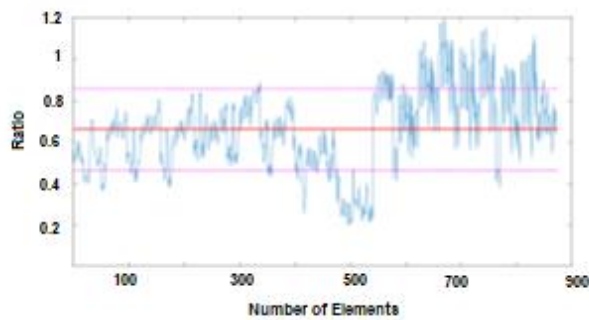
- باقی نتایج حاصل در حالت سازه منظم در پلان و ارتفاع، برای این بخش نیز صادق است.



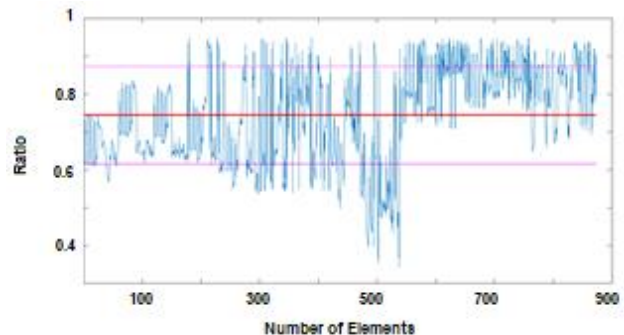
نتایج الگوریتم ژنتیک ارتقاء یافته



نتایج نرم افزار SAP2000



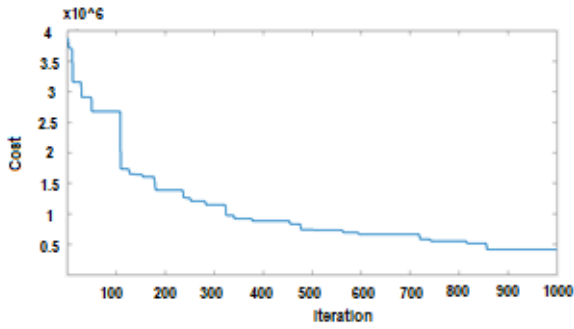
نتایج شبکه‌های عصبی



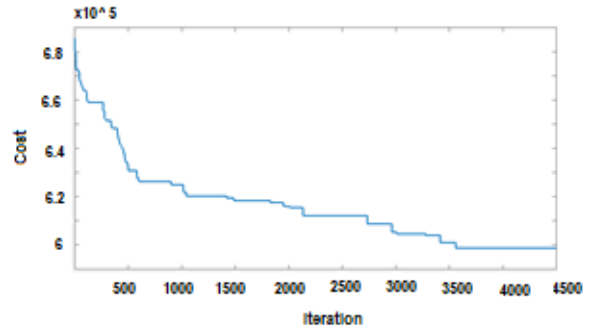
نتایج الگوریتم مورچگان ارتقاء یافته

شکل ۵. مقادیر Ratio برای هر المان سازه‌ای در نتایج نهایی روش‌های مختلف برای سازه نامنظم در پلان و منظم در ارتفاع (در تمامی اشکال محور افقی تعداد تکرار و محور قائم Ratio را نشان می‌دهد. خط قرمز متوسط مقدار Ratio را نشان می‌دهد).

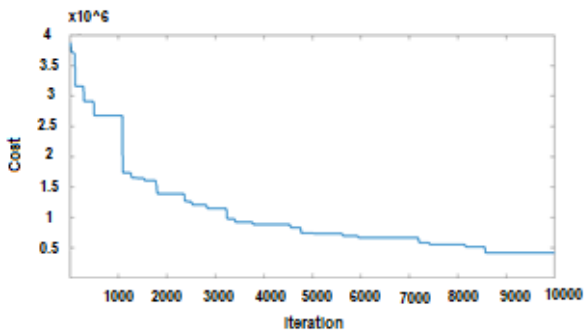
Fig. 5. Ratio values for each element in different methods for irregular structures in plan and regular in height



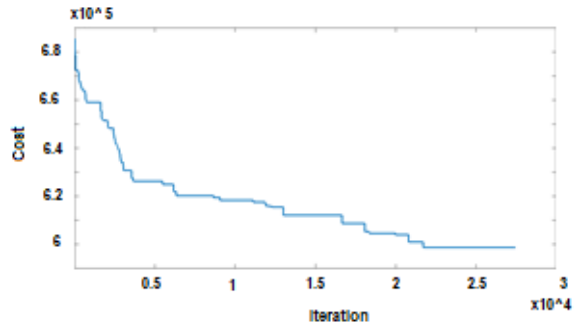
روند کاهش هزینه بر حسب تعداد تکرار در الگوریتم مورچگان ارتقاء یافته



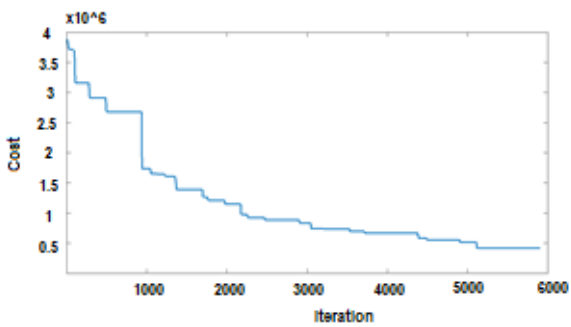
روند کاهش هزینه بر حسب تعداد تکرار در الگوریتم ژنتیک ارتقاء یافته



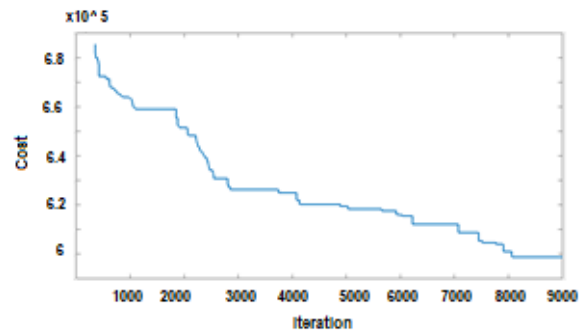
روند کاهش هزینه بر حسب تعداد فراخوانی تابع وزن



روند کاهش هزینه بر حسب تعداد فراخوانی تابع وزن



روند کاهش هزینه بر حسب تعداد فراخوانی تابع SAP2000



روند کاهش هزینه بر حسب تعداد فراخوانی تابع SAP2000

شکل ۶. روند کاهش تابع هزینه در الگوریتم‌های تکاملی برای سازه نامنظم در پلان و منظم در ارتفاع (در تمامی اشکال فوق محور افقی تعداد تکرار و محور قائم هزینه می‌باشد).

Fig. 6. The process of reducing the cost function in evolutionary algorithms for irregular structures in plan and regular in height

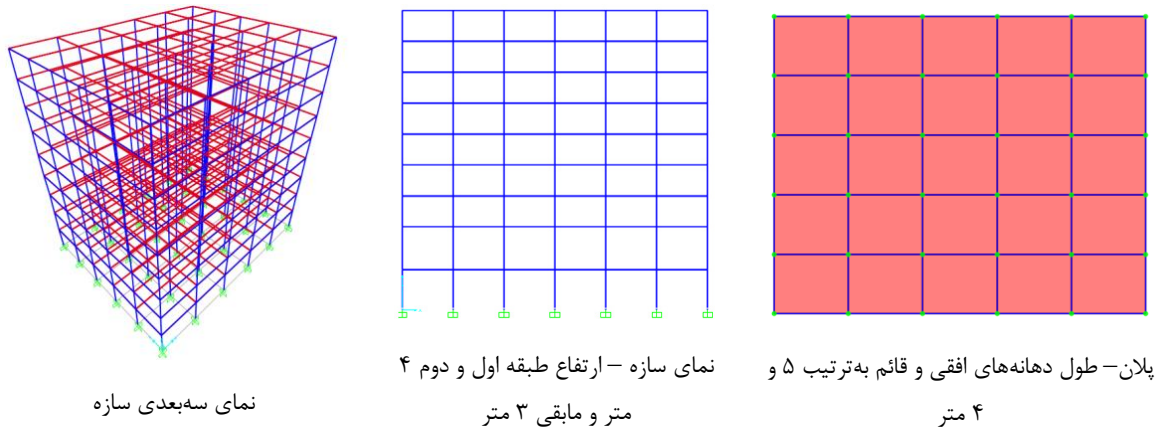
مقایسه شده است.

مقایسه روش الگوریتم ژنتیک بهبود یافته و الگوریتم مورچگان بهبود یافته از منظر نرخ همگرایی، تعداد تکرار و دیگر موارد در جدول ۹ و شکل ۸ آمده است.

برای این بخش نیز شبکه عصبی پرسپترون سه لایه با روش آموزش

دو جهت قاب خمشی متوسط بوده و تمامی اتصالات آن‌ها گیردار هستند. پلان و نمای ساختمان همانند شکل ۷ است.

در ادامه در جداول و شکل‌های مختلف، نتایج حاصل از تکنیک‌های محاسبات نرم به همراه نتایج حاصل از طراحی خودکار نرم افزار SAP2000 برای سازه منظم در پلان و نامنظم در ارتفاع



شکل ۷. سازه (ج) - منظم در پلان و نامنظم در ارتفاع

Fig. 7. Ratio values for each element in different methods for regular structures in plan and irregular in height

ساختمان همانند شکل ۱۰ است. در ادامه در جداول و شکل های مختلف، نتایج حاصل از تکنیک های محاسبات نرم به همراه نتایج حاصل از طراحی خودکار نرم افزار SAP2000 برای سازه منظم در پلان و نامنظم در ارتفاع مقایسه شده است.

مقایسه روش الگوریتم ژنتیک بهبودیافته و الگوریتم مورچگان بهبودیافته از منظر نرخ همگرایی، تعداد تکرار و دیگر موارد در جدول ۱۱ و شکل ۱۰ آمده است.

برای این بخش نیز شبکه عصبی پرسپترون سه لایه با روش آموزش Levenberg-Marquadet انتخاب شد و نتایج ارائه شده بر این مینا است.

همان طور که مشاهده می شود نتایج این بخش نیز مانند حالت های سابق و تأییدکننده آن ها می باشد.

به منظور اعتبارسنجی روش پیشنهادی مقاله حاضر لازم است که نتایج حاصله با نتایج تحقیق های مشابه مورد مقایسه قرار گیرد. ارائه

Levenberg-Marquadet انتخاب شد و نتایج ارائه شده است.

همان طور که مشاهده می شود نتایج این بخش نیز مانند حالت های سابق است. با توجه به نتایج به دست آمده، هر چه مسئله پیچیده تر می شود، قدرت الگوریتم های تکاملی نسبت به انتخاب خودکار نرم افزار SAP2000 بیشتر نمایان می شود و اختلاف در نتایج بیشتر خود را نشان می دهد.

۴-۵ - سازه نامنظم در پلان و ارتفاع

سازه ای که در این بخش بررسی می گردد، یک ساختمان ۹ طبقه است، که ارتفاع طبقه اول و دوم ۴ متر و باقی طبقات ۳ متر می باشد و این اختلاف در ارتفاع باعث نامنظمی در ارتفاع خواهد شد. کاربری سازه مورد نظر مسکونی در نظر گرفته شده است. مساحت هر طبقه برابر ۵۰۰ متر مربع است. کلیه طبقات تیب هستند و سازه در پلان و ارتفاع نامنظم است. سیستم ساختمانی در هر دو جهت قاب خمشی متوسط بوده و تمامی اتصالات آن ها گیردار هستند. پلان و نمای

جدول ۸. نتایج حاصل از چهار روش برای سازه منظم در پلان و نامنظم در ارتفاع

Table 8. Results of four methods for regular structures in plan and irregular in height

بیشترین مقطع استفاده شده	کمترین مقدار Ratio	بیشترین مقدار Ratio	انحراف معیار Ratio	میانگین Ratio	تعداد مقاطع انتخابی	درصد تغییرات وزن نسبت به SAP	وزن بر واحد سطح (کیلوگرم بر متر مربع)	وزن کل سازه (تن)	روش
W12X40	۰/۲	۰/۹۶	۰/۱۷	۰/۵۹	۳۲	۰	۱۱۰/۸	۴۹۹	SAP2000
W21X55	۰/۱۲	۰/۹۷	۰/۱۷	۰/۵۹	۲۳	- ۹/۸	۱۳۴/۳۴	۴۵۰	ژنتیک ارتقاء یافته
W18X50	۰/۴	۰/۹۵	۰/۱۲	۰/۷۷	۱۵	- ۲۰/۳۴	۸۸/۲۶	۳۸۱	مورچگان ارتقاء یافته
W8X48	۰/۱۸	۱/۲۵	۰/۲۳	۰/۶	۱۹	- ۱۹/۶۰	۸۹/۰۸	۳۸۲	شبکه عصبی

جدول ۹. نتایج مختص الگوریتم‌های تکاملی برای سازه منظم در پلان و نامنظم در ارتفاع

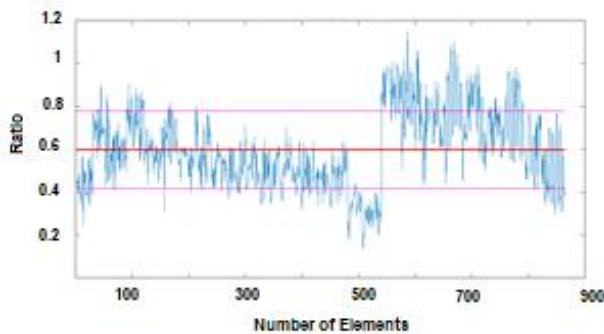
Table 9. Results of Applying Evolutionary Algorithms for regular structures in plan and irregular in height

تعداد جمعیت اولیه	تعداد فراخوانی تابع آنالیز	درصد فراخوانی تابع آنالیز	تعداد فراخوانی نرم افزار SAP	تعداد فراخوانی تابع وزن	تعداد تکرار	نتیجه
۱۰	۳۲/۴۵	به فراخوانی تابع وزن	۸۸۷۳	۲۷۳۴۱	۳۵۱۰	ژنتیک ارتقاء یافته
۱۰	۷۳/۵۷		۶۳۴۲	۸۶۲۰	۸۶۲	مورچگان ارتقاء یافته

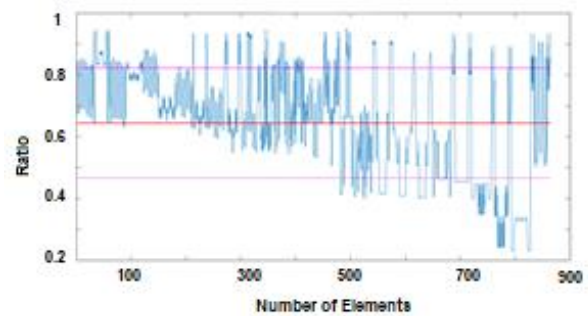
در المان‌های خرپا با عنایت به طول آن‌ها در برخی المان‌ها شاهد افزایش وزن و در برخی دیگر از المان‌ها شاهد کاهش وزن خواهیم بود. به عنوان مثال در مقاله اسکندر و همکاران [۲۰] مساله طراحی بهینه یک خرپای ۱۰ عضوی مطرح شده و نتیجه به صورت زیر مورد اشاره قرار گرفته است:

ملاحظه می‌شود که میزان مساحت مورد نیاز برای اعضای کششی و فشاری در الگوریتم‌های مختلف به صورت متفاوت حاصل شده است. لذا تشخیص اینکه یک الگوریتم برای مسایل طراحی حتی در حد

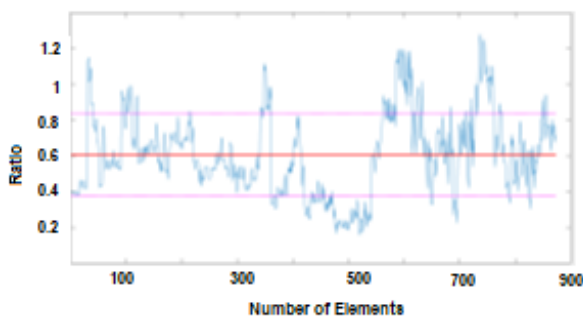
هرگونه مساله طراحی مهندسی به دلیل طبیعت طراحی، ساده نبوده و حتی برای ساده ترین حالات موجود در ادبیات فنی تشخیص اینکه واقعا الگوریتم بهینه سازی موجود به پاسخ یکتا رسیده باشد امکان پذیر نیست. به عنوان مثال در ادبیات فنی، مساله طراحی بهینه خرپاها مورد بررسی قرار گرفته است. در ساده ترین حالت، طراحی یک خرپای ۱۰ عضوی توسط الگوریتم‌های مختلف منجر به انتخاب‌های متفاوت برای هر یک از اعضای فشاری و کششی می‌شود. اما روی هم رفته، وزن‌های حاصل از الگوریتم‌ها به هم نزدیک خواهد بود چرا که با تغییر



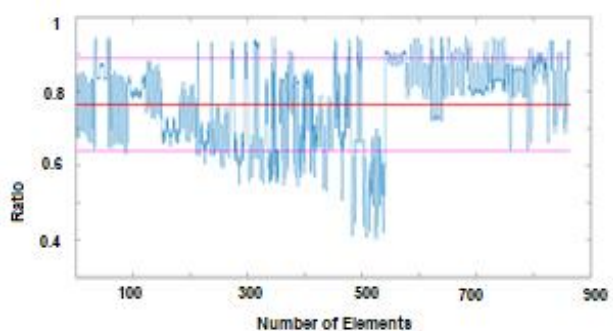
نتایج الگوریتم ژنتیک ارتقاء یافته



نتایج نرم افزار SAP2000



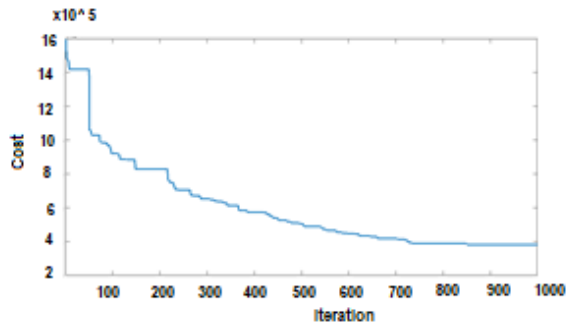
نتایج شبکه‌های عصبی



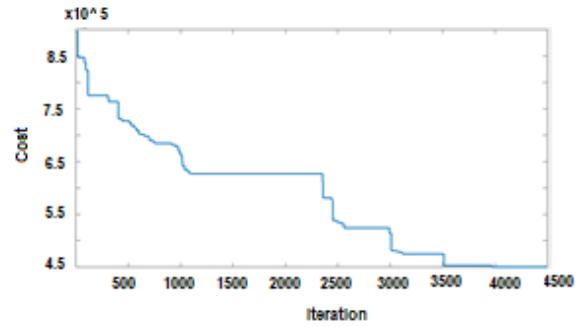
نتایج الگوریتم مورچگان ارتقاء یافته

شکل ۸. مقادیر Ratio برای هر المان سازه‌ای در نتایج نهایی روش‌های مختلف برای سازه منظم در پلان و نامنظم در ارتفاع (در تمامی اشکال محور افقی تعداد تکرار و محور قائم Ratio را نشان می‌دهد. خط قرمز متوسط مقدار Ratio را نشان می‌دهد).

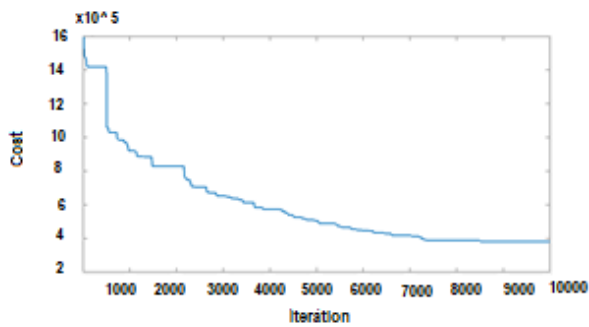
Fig. 8. The process of reducing the cost function in evolutionary algorithms for regular structures in plan and irregular in height



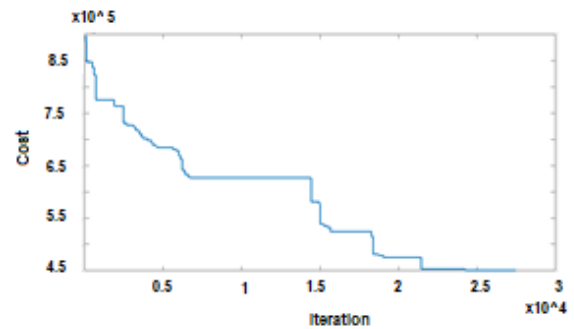
روند کاهش هزینه بر حسب تعداد تکرار در الگوریتم مورچگان ارتقاء یافته



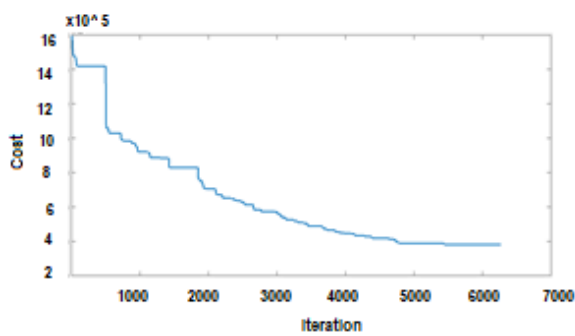
روند کاهش هزینه بر حسب تعداد تکرار در الگوریتم ژنتیک ارتقاء یافته



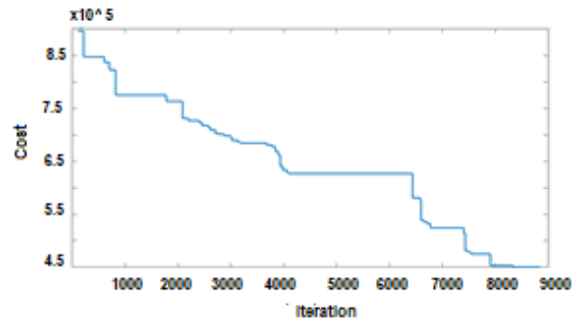
روند کاهش هزینه بر حسب تعداد فراخوانی تابع وزن



روند کاهش هزینه بر حسب تعداد فراخوانی تابع وزن



روند کاهش هزینه بر حسب تعداد فراخوانی تابع SAP2000



روند کاهش هزینه بر حسب تعداد فراخوانی تابع SAP2000

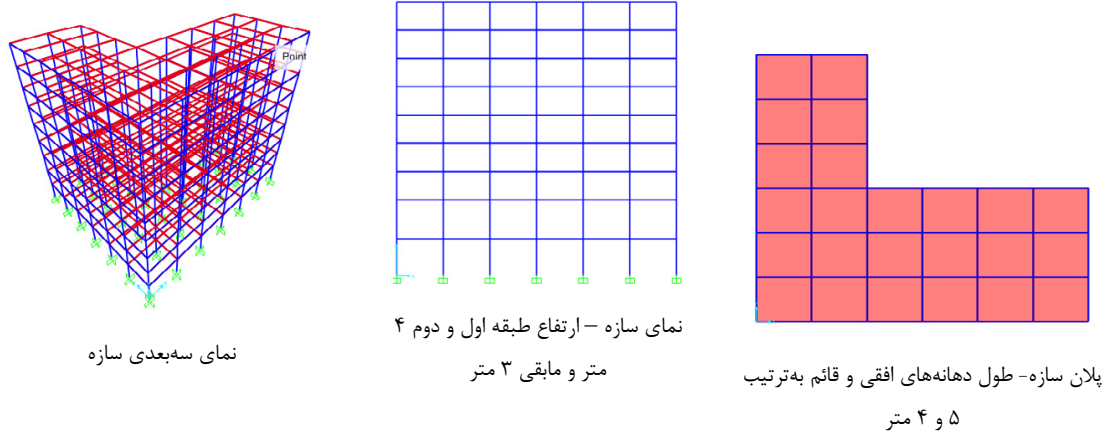
شکل ۹. روند کاهش تابع هزینه در الگوریتم‌های تکاملی برای سازه منظم در پلان و نامنظم در ارتفاع (در تمامی اشکال فوق محور افقی تعداد تکرار و محور قائم هزینه می باشد).

Fig. 9. Ratio values for each element in different methods for irregular structures in plan and height

۶- نتیجه‌گیری و ارائه پیشنهادها

به دلیل افزایش هزینه‌های ساخت سازه‌ها، توجه بیشتر طراحان به طراحی سازه‌هایی با حداقل وزن و حداکثر ظرفیت باربری می‌باشد. براین اساس نیز پژوهش حاضر به بررسی کاربرد ابزارهای موجود در

۱۰ المان مناسب‌ترین است آسان نیست. از این‌رو در مقاله حاضر نیز اعتبارسنجی روش‌های بهینه‌سازی بر پایه تایید اعتبار مفاهیم الگوریتم‌های مورد استفاده و پیاده‌سازی در مطالعه موردی و مقایسه با نتایج نرم افزار مورد توجه قرار گرفت.



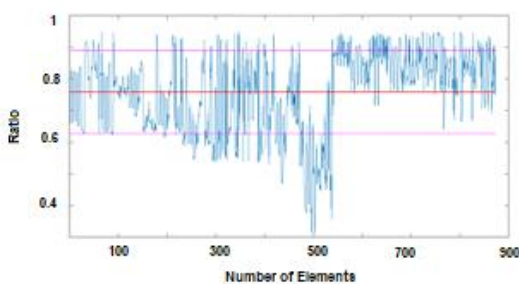
شکل ۱۰. سازه (د) - نامنظم در پلان و ارتفاع

Fig. 10. The process of reducing the cost function in evolutionary algorithms for irregular structures in plan and height

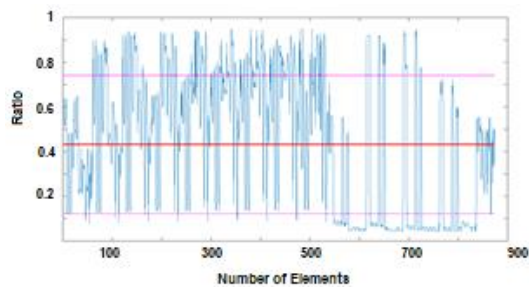
جدول ۱۰. نتایج حاصل از چهار روش برای سازه نامنظم در پلان و ارتفاع

Table 10. Results of four methods for irregular structures in plan and height

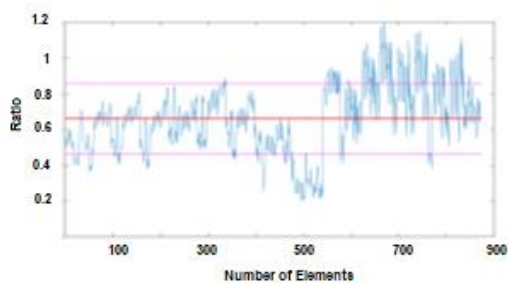
بیشترین مقطع استفاده شده	کمترین مقدار Ratio	بیشترین مقدار Ratio	میانگین انحراف معیار Ratio	تعداد مقاطع انتخابی	درصد تغییرات وزن نسبت به SAP	وزن بر واحد سطح (کیلوگرم بر متر مربع)	وزن کل سازه (تن)	
W8X10	۰/۰۹	۰/۹۶	۰/۳	۱۱	۰	۱۰۵/۵۵	۴۷۳	SAP2000
W16X57	۰/۳	۰/۹۴	۰/۱۳	۲۱	- ۱۴/۱۱	۹۴/۰۹	۴۰۶	ژنتیک ارتقاء یافته
W18X50	۰/۳۵	۰/۹۴	۰/۱۲	۱۵	- ۲۰/۳۸	۸۷/۲۲	۳۷۷	مورچگان ارتقاء یافته
W18X50	۰/۲	۰/۹۸	۰/۲	۱۱	- ۲۲/۳۳	۸۵/۰۸	۳۸۳	شبکه عصبی



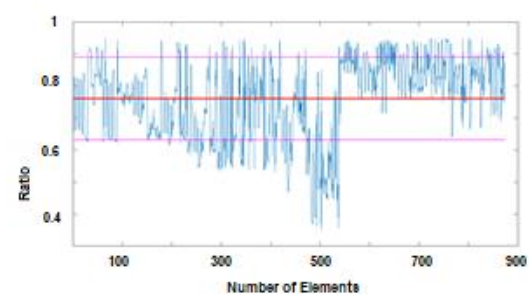
نتایج الگوریتم ژنتیک ارتقاء یافته



نتایج نرم افزار SAP2000



نتایج شبکه های عصبی



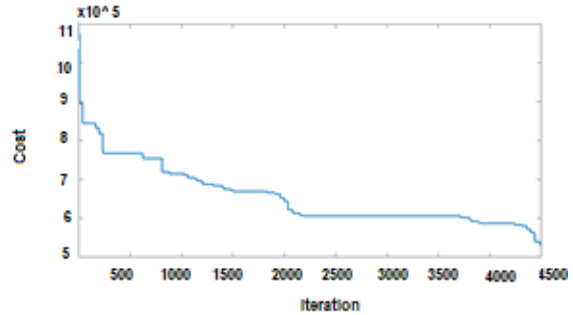
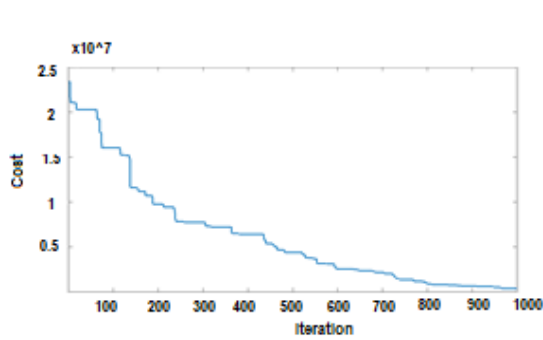
نتایج الگوریتم مورچگان ارتقاء یافته

شکل ۱۱. مقادیر Ratio برای هر المان سازه ای در نتایج نهایی روش های مختلف برای سازه نامنظم در پلان و ارتفاع (در تمامی اشکال محور افقی تعداد تکرار و محور قائم Ratio را نشان می دهد. خط قرمز متوسط مقدار Ratio را نشان می دهد).

جدول ۱۱. نتایج مختص الگوریتم‌های تکاملی برای سازه نامنظم در پلان و ارتفاع

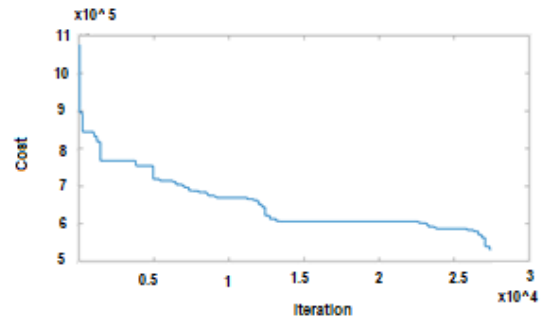
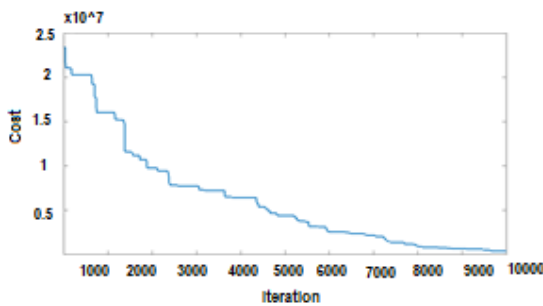
Table 11. Results of Applying Evolutionary Algorithms for irregular structures in plan and height

تعداد جمعیت اولیه	درصد فراخوانی تابع آنالیز به فراخوانی تابع وزن	تعداد فراخوانی نرم افزار SAP	تعداد فراخوانی تابع وزن	تعداد تکرار	
۱۰	۳۶،۱۴	۹۹۳۶	۲۷۴۹۱	۴۵۰۰	ژنتیک ارتقاء یافته
۱۰	۶۱،۸۳	۶۱۸۳	۱۰۰۰۰	۱۰۰۰	مورچگان ارتقاء یافته



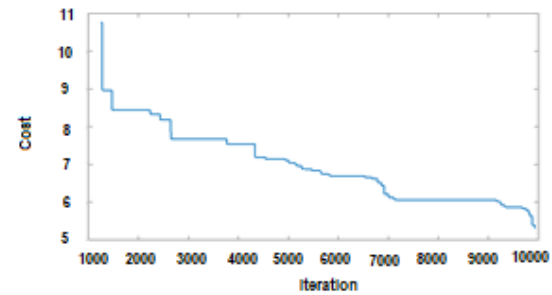
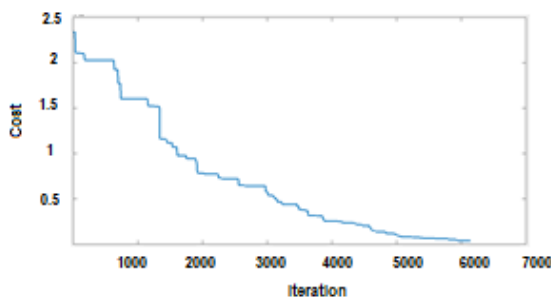
روند کاهش هزینه بر حسب تعداد تکرار در الگوریتم مورچگان ارتقاء یافته

روند کاهش هزینه بر حسب تعداد تکرار در الگوریتم ژنتیک ارتقاء یافته



روند کاهش هزینه بر حسب تعداد فراخوانی تابع وزن

روند کاهش هزینه بر حسب تعداد فراخوانی تابع وزن



روند کاهش هزینه بر حسب تعداد فراخوانی تابع SAP2000

روند کاهش هزینه بر حسب تعداد فراخوانی تابع SAP2000

شکل ۱۲. روند کاهش تابع هزینه در الگوریتم‌های تکاملی برای سازه نامنظم در پلان و ارتفاع (در تمامی اشکال فوق محور افقی تعداد تکرار و محور قائم هزینه می باشد).

جدول ۱۲. مقایسه طراحی بهینه برای خرابی ۱۰ عضوی

متغیرها	WCA	PSO[۲۱]	PSOC[۲۱]	HPSO[۲۱]	Shih & Yang [۲۲]	Kaveh & Rahami [۲۳]	Kaveh & Hassani [۲۴]	MBA [۲۵]	GA[۲۶]
A1	۳۳.۵	۳۰	۳۰	۳۰	۳۳.۵	۳۳.۵	۳۳.۵	۳۰	۳۳.۵
A2	۱.۶۲	۱.۶۲	۱.۸	۱.۶۲	۱.۶۲	۱.۶۲	۱.۶۲	۱.۶۲	۱.۶۲
A3	۲۲.۹	۳۰	۲۶.۵	۲۲.۹	۲۲.۹	۲۲.۹	۲۲.۹	۲۲.۹	۲۲
A4	۱۴.۲	۱۳.۵	۱۵.۵	۱۳.۵	۱۵.۵	۱۴.۲	۱۴.۲	۱۶.۹	۱۵.۵
A5	۱.۶۲	۱.۶۲	۱.۶۲	۱.۶۲	۱.۶۲	۱.۶۲	۱.۶۲	۱.۶۲	۱.۶۲
A6	۱.۶۲	۱.۸	۱.۶۲	۱.۶۲	۱.۶۲	۱.۶۲	۱.۶۲	۱.۶۲	۱.۶۲
A7	۷.۹۷	۱۱.۵	۱۱.۵	۷.۹۷	۷.۹۷	۷.۹۷	۱۱.۵	۷.۹۷	۱۴.۲
A8	۲۲.۹	۱۸.۸	۱۸.۸	۲۶.۵	۲۲	۲۲.۹	۲۲	۲۲.۹	۱۹.۹
A9	۲۲	۲۲	۲۲	۲۲	۲۲	۲۲	۱۹.۹	۲۲.۹	۱۹.۹
A10	۱.۶۲	۱.۸	۳.۰۹	۱.۸	۱.۶۲	۱.۶۲	۱.۶۲	۱.۶۲	۲.۶۲
Weight (lb)	۵۴۹۰.۷۳	۵۵۸۱.۷۶	۵۵۹۳.۴	۵۵۳۱.۹	۵۴۹۱.۷	۵۴۹۰.۷۴	۵۵۱۷.۷۲	۵۵۰۷.۷	۵۶۱۳.۸

سایر روش‌های بهینه‌سازی مورد استفاده قرار گیرد. پیشنهاد می‌گردد که محققین آتی موضوع پژوهش حاضر را در قالب‌های خمشی ویژه، با توجه به قانون تیر قوی، ستون ضعیف مد نظر قرار دهند. علاوه بر این بررسی سازه‌های بتنی و نحوه بهینه‌سازی آن‌ها به کمک محاسبات نرم می‌تواند به‌عنوان موضوعی برای پژوهش‌های آتی لحاظ گردد.

مراجع

- [1] M.-B. Prendes-Gero, M.-I. Álvarez-Fernández, F. López-Gayarre, J.-M. Drouet, J.R.-V. Junco, Cost optimization of structures using a genetic algorithm with Eugenic Evolutionary theory, *Structural and Multidisciplinary Optimization*, 54(2) (2016) 199-213.
- [2] S. Gholizadeh, O. Samavati, Structural optimization by wavelet transforms and neural networks, *Applied Mathematical Modelling*, 35(2) (2011) 915-929.
- [3] J. Akbari, M.S. Ayubirad, Seismic optimum design of steel structures using gradient-based and genetic algorithm methods, *International Journal of Civil Engineering*, 15(2) (2017) 135-148.
- [4] G. Gülay, H. Boduroğlu, An algorithm for the optimum design of braced and unbraced

محاسبات نرم در طراحی بهینه سازه‌ها با هدف کاهش وزن سازه‌ها پرداخته‌است. از آنجایی‌که مقالات متعددی در ادبیات پژوهش به بررسی این موضوع پرداخته‌اند، نگارندگان با استفاده از دو الگوریتم ژنتیک و مورچگان بهبودیافته سعی در افزایش سرعت و دقت جواب‌های حاصل از بهینه‌سازی طراحی سازه‌ها اقدام نمودند. علاوه بر این با هدف مقایسه نتایج الگوریتم‌های مورد اشاره با شبکه‌های عصبی، از شبکه عصبی پرسپترون دو و سه‌لایه برای بهینه‌سازی استفاده شده و نتایج با یکدیگر مقایسه گردید. برقراری ارتباط بین نرم‌افزارهای MATLAB و SAP2000 که از سخت‌ترین بخش‌های پژوهش بود و منبع مناسبی برای این کار وجود نداشت به‌عنوان یکی از مهمترین نوع‌آوری‌های این تحقیق انجام شد و با استفاده از آن طراحی سازه مورد بحث، به‌صورت سه‌بعدی مدل شد و انواع نامنظمی‌ها در ساختمان مورد بررسی قرار گرفت.

نتایج نشان داد که الگوریتم ژنتیک ارتقاءیافته و الگوریتم مورچگان ارتقاءیافته توانستند وزن سازه را براساس نوع سازه مورد نظر از ۱۴ تا ۳۳ درصد نسبت به طراحی خودکار نرم‌افزار کاهش دهند. نتایج شبکه عصبی پرسپترون پیشنهادی در این پژوهش نیز منتج به کاهش وزن سازه شد اما توانمندی لازم در ارضای تمامی قیود آیین‌نامه‌ای را ندارد. در نتیجه نمی‌تواند به‌عنوان یک روش مجزا به جهت بهینه‌سازی استفاده گردد اما با توجه به سرعت بالای آن می‌تواند در ترکیب با

Structures, 93 (2015) 36-49.

[16] S. Gholizadeh, Performance-based optimum seismic design of steel structures by a modified firefly algorithm and a new neural network, *Advances in Engineering Software*, 81 (2015) 50-65.

[17] S. Gholizadeh, F. Fattahi, Optimum design of steel structures for earthquake loading by grey wolf algorithm, (2015).

[18] I.N.C. Bureau, The National Iranian Building Code Vol. 10 (design and implementation of steel buildings), in: Fourth Edition, Iran, 2013.

[19] I.N.C. Bureau, The National Iranian Building Code Vol. 9 (design and implementation of concrete buildings), in: Fourth Edition, Iran, 2015.

[20] H. Eskandar, A. Sadollah, Bahreininejad, Weight optimization of truss structures using water cycle algorithm, *INTERNATIONAL JOURNAL OF OPTIMIZATION IN CIVIL ENGINEERING*, 3(1) (2013) 115-129.

[21] L. Li, Z. Huang, F.J.C. Liu, Structures, A heuristic particle swarm optimization method for truss structures with discrete variables, 87(7-8) (2009) 435-443.

[22] L. Li, Z. Huang, F.J.C. Liu, Structures, A heuristic particle swarm optimization method for truss structures with discrete variables, 87(7-8) (2009) 435-443.

[23] A. Kaveh, H. Rahami, Analysis, design and optimization of structures using force method and genetic algorithm, *International Journal for Numerical Methods in Engineering*, 65(10) (2006) 1570-1584.

[24] A. Kaveh, M. Hassani, Simultaneous analysis, design and optimization of structures using force method and ant colony algorithms, (2009).

[25] A. Sadollah, A. Bahreininejad, H. Eskandar, M. Hamdi, Mine blast algorithm for optimization of truss structures with discrete variables, *Computers & Structures*, 102 (2012) 49-63.

[26] S. Rajeev, C. Krishnamoorthy, Discrete optimization of structures using genetic algorithms, *Journal of structural engineering*, 118(5) (1992) 1233-1250.

steel frames under earthquake loading, *Earthquake engineering & structural dynamics*, 18(1) (1989) 121-128.

[5] F. Erbatur, M. Al-Hussainy, Optimum design of frames, *Computers & structures*, 45(5-6) (1992) 887-891.

[6] V. Thevendran, N.D. Gupta, G. Tan, Minimum weight design of multi-bay multi-storey steel frames, *Computers & structures*, 43(3) (1992) 495-503.

[7] E.J. O'Brien, A. Dixon, Optimal plastic design of pitched roof frames for multiple loading, *Computers & structures*, 64(1-4) (1997) 737-740.

[8] S. Hernandez, Optimum design of steel structures, *Journal of Constructional Steel Research*, 46 (1) (1998), 374-378.

[9] N.B.H. Ali, M. Sellami, A.-F. Cutting-Decelle, J.-C. Mangin, Multi-stage production cost optimization of semi-rigid steel frames using genetic algorithms, *Engineering Structures*, 31(11) (2009) 2766-2778.

[10] M. Kociecki, H. Adeli, Two-phase genetic algorithm for topology optimization of free-form steel space-frame roof structures with complex curvatures, *Engineering Applications of Artificial Intelligence*, 32 (2014) 218-227.

[11] A. Kaveh, S. Talatahari, Charged system search for optimal design of frame structures, *Applied Soft Computing*, 12(1) (2012) 382-393.

[12] A. Kaveh, P. Zakian, Optimal design of steel frames under seismic loading using two meta-heuristic algorithms, *Journal of Constructional Steel Research*, 82 (2013) 111-130.

[13] C.-S. Kao, I. Yeh, Optimal design of reinforced concrete plane frames using artificial neural networks, *Computers and Concrete*, 14(4) (2014) 445-462.

[14] A. Kaveh, P. Zakian, Enhanced bat algorithm for optimal design of skeletal structures, (2014).

[15] D.T. Phan, J.B. Lim, T.T. Tanyimboh, A. Wrzesien, W. Sha, R. Lawson, Optimal design of cold-formed steel portal frames for stressed-skin action using genetic algorithm, *Engineering*

پیوست:

بخشی از کد منبع الگوریتم ژنتیک برای سازه منظم در پلان و ارتفاع

```

%% GA
disp('Start GA alghoritm')

%% Problem Definition
disp('Problem Definition')

global NFE;
NFE = 0;
NFESAP = 0;

CostFunction = @(Gene) GeneWeight(Gene,
Element, WSection); % Cost Function

NumOfWSection = size(WSection,2);

NumOfAllElement = numel(Element);
nVar = NumOfAllElement; % Number of
Decision Variables
VarSize=[1 nVar]; % Decision Variables
Matrix Size

%% GA Parameters
disp('GA Parameters')

MaxIt = 2000; % Maximum Number of Iterations
nPop = 10; % Population Size

pc = 0.6; % Crossover Percentage
nc = 2*round(pc*nPop/2); % Number of Offsprings
(Parnets)

pm = 0.2; % Mutation Percentage
nm = round(pm*nPop); % Number of Mutants

mu = 0.1; % Mutation Rate

ANSWER = questdlg('Choose selection
method:','Genetic Algorith',...
'Roulette Wheel','Tournament','Random','Roulette
Wheel');

UseRouletteWheelSelection =
strcmp(ANSWER,'Roulette Wheel');
UseTournamentSelection =
strcmp(ANSWER,'Tournament');
UseRandomSelection =
strcmp(ANSWER,'Random');
    
```

```

if UseRouletteWheelSelection
beta = 8;    % Selection Pressure
end

if UseTournamentSelection
TournamentSize = 3; % Tournamnet Size
end

pause(0.1);
%% Initialization
disp('Initialization')

empty_individual.Position = [];
empty_individual.Cost = [];
empty_individual.CheckSF = [];

DataBase(2).ElementSectionNumber = 1;

pop = repmat(empty_individual,nPop,1);
TotalCAPVMin      =      WSection(1).Area/
WSection(end).Area-1;

Z = NumberOfStory*NumOfElement;
BeamVarSize = [1 Z];
ColumnVarSize = [1 NumOfAllElement-Z];

CAPV(NumOfAllElement) = 0;

for i=1:nPop
TotalCAPV = 1;
CounterW = 0;
while TotalCAPV ~= 0
CounterW = CounterW + 1;
pop(i).Position(1:Z) = randi([1 NumOfWSection-
4],BeamVarSize);
for m=Z+1:NumOfAllElement
pop(i).Position(m) = randi([1 NumOfWSection]);
Tabu = pop(i).Position(m);
while (WSection(Tabu).Area < 0.03)
pop(i).Position(m) = randi([1
NumOfWSection]);
Tabu = pop(i).Position(m);
end
end

disp(['Create Primary Population, Number Of Pop=
' num2str(i) ', Number Of Try = ' num2str(CounterW)])
Ret = SapModel.SetModelIsLocked(false());
if Ret ~= 0
disp(['! Error at SapModel.SetModelIsLocked
Number = ' num2str(CounterW)]);
end

```

```

for k = 1:NumOfAllElement
Ret = FrameObj.SetSection(...
Element(k).Name, WSection(pop(1).
Position(k)).Name);
if Ret ~= 0
disp(['! Error at FrameObj.SetSection
NumOfAllElement = ' num2str(k)])
end
end
Ret = Analyze.RunAnalysis();
if Ret ~= 0
disp(['! Error at Analyze.RunAnalysis '])
end
Ret = DesignSteel.StartDesign();
if Ret ~= 0
disp(['! Error at DesignSteel.StartDesign '])
end
Ret = SelectObj.All;
if Ret ~= 0
disp(['! Error at SelectObj.All '])
end
[Ret,DesignedNumberItems,DesignedFrameName,
DesignedRatio, ...
DesignedRatioType, DesignedLocation,
DesignedComboName, ...
DesignedErrorSummary,
DesignedWarningSummary ] = ...
DesignSteel.GetSummaryResults(",
DesignedNumberItems, DesignedFrameName,...
DesignedRatio, DesignedRatioType,
DesignedLocation, DesignedComboName,...
DesignedErrorSummary,
DesignedWarningSummary, SAP2000v19.eItemtype.
SelectedObjects);
if Ret ~= 0
disp(['! Error at DesignSteel.GetSummaryResults
'])
end
NFESAP = NFESAP+1;

DataBase(NFESAP).ElementSectionNumber =
pop(1).Position;
DataBase(NFESAP).BeamSectionNumber =
pop(1).Position(1:Z);
DataBase(NFESAP).ColumnSectionNumber =
pop(1).Position(Z+1:end);

for l = 1:DesignedNumberItems
DataBase(NFESAP).Design.Ratio(l) =
DesignedRatio(l);
DataBase(NFESAP).Design.RatioType(l) =
DesignedRatioType(l);

```



```

DataBase(NFESAP).Design.
Location(DesignedNumberItems)           =
DesignedLocation(l);
DataBase(NFESAP).Design.
ComboName{DesignedNumberItems}         =
DesignedLocation(l);
DataBase(NFESAP).Design.ErrorSummary{Des
ignedNumberItems} = DesignedLocation(l);
DataBase(NFESAP).Design.WarningSummary{
DesignedNumberItems} = DesignedLocation(l);
end

```

```

Ret = SelectObj.ClearSelection;
if Ret ~= 0
disp('! Error at SelectObj.ClearSelection ')
end

```

```

for k = 1:NumOfAllElement
CAPV(k) = max(DesignedRatio(k)-1, 0);
end
TotalCAPV = sum(CAPV(:));
if TotalCAPV > 0
TotalCAPV = max(TotalCAPV,TotalCAPVMin);
end
for k = 1:NumOfAllElement
if (DesignedRatio(k)==0)
TotalCAPV = max(TotalCAPV,TotalCAPVMin);
end
end
end
end

```

```
for i=1:nPop
```

```
% Evaluation
```

```
pop(i).Cost = CostFunction(pop(i).Position);
```

```
% Safty Factor
```

```
pop(i).CheckSF = 1;
```

```
end
```

```
% Sort Population
```

```
Costs = [pop.Cost];
```

```
[Costs, SortOrder] = sort(Costs);
```

```
pop = pop(SortOrder);
```

```
% Store Best Solution
```

```
BestSol = repmat(empty_individual,MaxIt,1);
```

```
% Array to Hold Best Cost Values
```

```
BestCost = zeros(MaxIt,1);
```

```

% Store Cost
WorstCost = pop(end).Cost;

% Array to Hold Number of Function Evaluations
nfe = zeros(MaxIt,1);
nfesap = zeros(MaxIt,1);

%% GA Main Loop
disp('GA Main Loop')

for it = 1:MaxIt

% Calculate Selection Probabilities
if UseRouletteWheelSelection
P = exp(-beta*Costs/WorstCost);
P = P/sum(P);
end

% Crossover
popc = repmat(empty_individual,nc/2,2);
for k = 1:nc/2
% Select Parents Indices
if UseRouletteWheelSelection
i1 = RouletteWheelSelection(P);
i2 = RouletteWheelSelection(P);
end
if UseTournamentSelection
i1 = TournamentSelection(pop,TournamentSize);
i2 = TournamentSelection(pop,TournamentSize);
end
if UseRandomSelection
i1 = randi([1 nPop]);
i2 = randi([1 nPop]);
end

% Select Parents
p1 = pop(i1);
p2 = pop(i2);

% Apply Crossover
[popc(k,1).Position, popc(k,2).Position] = ...
Crossover(p1.Position, p2.Position);

% Evaluate Offsprings
popc(k,1).Cost = CostFunction(popc(k,1).
Position);
popc(k,2).Cost = CostFunction(popc(k,2).
Position);

```

```

end
popc = popc(:);

% Mutation
if (rand < 0.1)
popm = repmat(empty_individual,nm,1);

for k = 1:nm
% Select Parent
i = randi([1 nPop]);
p = pop(i);

% Apply Mutation
popm(k).Position = Mutate(p.Position, mu,
NumOfWSection);

% Evaluate Mutant
popm(k).Cost = CostFunction(popm(k).
Position);
end
% Create Merged Population

pop = [pop; popc; popm];
else
pop = [pop; popc];
end

% Sort Population
Costs = [pop.Cost];
[Costs, SortOrder] = sort(Costs);
pop = pop(SortOrder);

% Safty Factor
for k = 1:size(pop,1)
if isempty(pop(k).CheckSF)
pop(k).CheckSF = 0;
end
end

% Update Worst Cost
WorstCost = max(WorstCost, pop(end).Cost);
CounterW = 0;
while (pop(1).CheckSF == 0)
CounterW = CounterW + 1;
disp(['Number Of Check Safty Factor = '
num2str(CounterW)])
Ret = SapModel.SetModelIsLocked(false());
if Ret ~= 0
disp(['! Error at SapModel.SetModelIsLocked
Number = ' num2str(CounterW)]);
end
end

```

```

for k = 1:NumOfAllElement
Ret = FrameObj.SetSection(...
Element(k).Name, WSection(pop(1).
Position(k)).Name);
if Ret ~= 0
disp(['! Error at FrameObj.SetSection
NumOfAllElement = ' num2str(k)])
end
end
Ret = Analyze.RunAnalysis();
if Ret ~= 0
disp(['! Error at Analyze.RunAnalysis ')
end
Ret = DesignSteel.StartDesign();
if Ret ~= 0
disp(['! Error at DesignSteel.StartDesign ')
end
Ret = SelectObj.All;
if Ret ~= 0
disp(['! Error at SelectObj.All ')
end
[Ret,DesignedNumberItems,DesignedFrameName,
DesignedRatio, ...
DesignedRatioType, DesignedLocation,
DesignedComboName, ...
DesignedErrorSummary,
DesignedWarningSummary ] = ...
DesignSteel.GetSummaryResults(",
DesignedNumberItems, DesignedFrameName,...
DesignedRatio, DesignedRatioType,
DesignedLocation, DesignedComboName,...
DesignedErrorSummary,
DesignedWarningSummary, SAP2000v19.eItemtype.
SelectedObjects);

```

```
NFESAP = NFESAP+1;
```

```

DataBase(NFESAP).ElementSectionNumber =
pop(1).Position;
DataBase(NFESAP).BeamSectionNumber =
pop(1).Position(1:Z);
DataBase(NFESAP).ColumnSectionNumber =
pop(1).Position(Z+1:end);

```

```

for i = 1:DesignedNumberItems
DataBase(NFESAP).Design.Ratio(i) =
DesignedRatio(i);
DataBase(NFESAP).Design.RatioType(i) =
DesignedRatioType(i);
DataBase(NFESAP).Design.
Location(DesignedNumberItems) =
DesignedLocation(i);

```

```

DataBase(NFESAP).Design.
ComboName{DesignedNumberItems}          =
DesignedLocation(i);
DataBase(NFESAP).Design.ErrorSummary{Desig
nedNumberItems} = DesignedLocation(i);
DataBase(NFESAP).Design.WarningSummary{D
esignedNumberItems} = DesignedLocation(i);
end

```

```

if Ret ~= 0
disp('! Error at DesignSteel.GetSummaryResults
')
end
Ret = SelectObj.ClearSelection;
if Ret ~= 0
disp('! Error at SelectObj.ClearSelection ')
end
pop(1).CheckSF = 1;
for k = 1:NumOfAllElement
CAPV(k) = max(DesignedRatio(k)-1, 0);
end
TotalCAPV = sum(CAPV(:));
if TotalCAPV > 0
TotalCAPV = max(TotalCAPV,TotalCAPVMin);
end
for k = 1:NumOfAllElement
if (DesignedRatio(k)==0)
TotalCAPV = max(TotalCAPV,TotalCAPVMin);
end
end
end

```

```

pop(1).Cost = pop(1).Cost*(1+TotalCAPV);

```

```

Costs = [pop.Cost];
[Costs, SortOrder] = sort(Costs);
pop = pop(SortOrder);
end

```

```

% Truncation
pop = pop(1:nPop);
Costs = Costs(1:nPop);

```

```

% Store Best Solution Ever Found
BestSol = pop(1);

```

```

% Store Best Cost Ever Found
BestCost(it) = BestSol.Cost;

```

```

% Store NFE
nfe(it) = NFE;
nfesap(it) = NFESAP;

```

```
% Show Iteration Information
disp(['Iteration ' num2str(it) ': NFE = '
num2str(nfe(it)) ', NFESAP = ' num2str(nfesap(it)) ',
Best Cost = ' num2str(BestCost(it))]);
%for i = 1:NumOfAllElement
% if DesignedRatio(i) > 1
% disp(num2str(i))
% end
%end

end

%% Results

figure;
plot(nfe,BestCost,'LineWidth',2);
xlabel('NFE');
ylabel('Cost');

figure;
plot(nfesap,BestCost,'LineWidth',2);
xlabel('NEFSAP');
ylabel('Cost');

figure;
plot(BestCost,'LineWidth',2);
xlabel('IT');
ylabel('Cost');
```

چگونه به این مقاله ارجاع دهیم
H.R. Abbasianjahromi, S. Asil Gharebaghi, M. Sadegh Eshaghi Khanghah, Optimizing the Weight of 3D Steel structures by spectrum dynamic analysis and soft computing algorithm, Amirkabir J. Civil Eng., 52(4) (2020) 889-918.
DOI: [10.22060/ceej.2019.15072.5823](https://doi.org/10.22060/ceej.2019.15072.5823)

