



بهینه سازی وزن سازه ها با استفاده از الگوریتم جستجوی گرانشی و بوسیله مقاطع فولادی گسسته

محمد حسین وفایی^۱، حسین ابراهیمی فرسنگی^۲

چکیده

جهت بهینه سازی گسسته ی وزن (Gravitational Search Algorithm) در این پژوهش الگوریتم جستجوی گرانشی خرپاهای دو بعدی و سازه های فضاکار با استفاده از مقاطع فولادی مورد استفاده قرار گرفته است. برنامه ی کامپیوتری الگوریتم کد نویسی شده است و نتایج بدست آمده از بهینه سازی وزن MATLAB جستجوی گرانشی و برنامه ی آنالیزگر در نرم افزار چندین مثال متداول بهینه سازی سازه ای توسط الگوریتم های بهینه سازی رایج همچون الگوریتم وراثتی، الگوریتم جمعیت مورچگان، الگوریتم جمعیت ذرات، الگوریتم جستجوی هارمونی و چندین الگوریتم دیگر با نتایج بدست آمده از الگوریتم جستجوی گرانشی مقایسه گردیده است که این مقایسه ها کارایی این الگوریتم ابتکاری نوین را در حل مسائل بهینه سازی سازه ای نمایش می دهد.

کلمات کلیدی

الگوریتم های ابتکاری، خرپای دوبعدی، سازه فضاکار، بهینه سازی گسسته، الگوریتم جستجوی گرانشی

Weight optimization of structures using Gravitational Search Algorithm with discrete steel profiles

Mohammad Hossein Vafae, Hossein Ebrahimi Farsangi

ABSTRACT

In this paper, the Gravitational Search Algorithm (GSA) is employed for discrete optimization of truss structures using some steel profiles. GSA is compared to other heuristic methods including Genetic Algorithm, Ant colony optimization, Particle swarms optimizer, Harmony Search and some other methods. The performance of the GSA is evaluated through some numerical examples. The results demonstrate the efficiency of the presented method in structural optimization.

۱. دانشجوی کارشناسی ارشد عمران سازه، دانشگاه شهید باهنر کرمان، Mohammadhosseinvafae@yahoo.com

۲. استادیار بخش مهندسی عمران، دانشگاه شهید باهنر کرمان، Ebrahimi@mail.uk.ac.ir



امروزه با بزرگ شدن مسائل و اهمیت یافتن سرعت رسیدن به پاسخ، روش‌های کلاسیک جوابگویی حل بسیاری از مسائل نیست، چرا که فضای جستجو با افزایش بعد مسئله به صورت نمایی گسترش می‌یابد و روش‌های کلاسیک به خاطر محدودیت‌های مختلف مقرون به صرفه نیستند [۱]. با توجه به ملاحظات فوق امروزه بیشتر از الگوریتم‌های جستجوی تصادفی به جای جستجوی همه جانبه استفاده می‌شود و استفاده از الگوریتم‌های تکاملی و الگوریتم‌های جستجوی ابتکاری در سالهای اخیر رشد چشمگیری داشته است. الگوریتم‌های جستجوی ابتکاری الگوریتم‌هایی هستند که با الهام از فرایندهای فیزیکی بیولوژیکی و طبیعی به وجود آمده‌اند. مانند الگوریتم وراثتی با الهام از علم وراثت و تکامل (۱۹۷۵)، پخت شبیه سازی شده با الهام از مشاهدات ترمودینامیک (۱۹۸۳)، جستجوی تابو بر پایه پاسخ حافظه (۱۹۸۶)، الگوریتم ایمنی با شبیه‌سازی سیستم دفاعی انسان (۱۹۸۶)، الگوریتم جستجوی جمعیت مورچگان با شبیه‌سازی رفتار مورچه‌ها در هنگام جستجوی غذا (۱۹۹۱) و بهینه‌سازی جمعیت ذره‌ها با تقلید از رفتار اجتماعی پرندگان (۱۹۹۵) که هر کدام از الگوریتم‌های فوق خصوصیات منحصر به خود را دارند [۲-۱۲].

الگوریتم وراثتی یک الگوریتم جستجوی موازی است که با یک جمعیت اولیه از جواب‌های مسئله شروع شده و به مرور نسل‌های بعدی به روز می‌شوند. فرایندی که جمعیت را به سمت بهترین‌ها سوق می‌دهد و زادوولد اعضا به گونه‌ای است که به اعضای بهتر شانس بیشتری برای تولید فرزند می‌دهد [۱۳-۱۵]. در الگوریتم پخت شبیه سازی شده عامل جستجو کننده با سرعتی متناسب با دما حرکت تصادفی می‌کند و موقعیت بعدی را در صورتی که

موقعیت بهتری باشد انتخاب و در غیر این صورت با یک احتمال جای خود را تغییر نمی‌دهد [۱۶]. در الگوریتم ایمنی جواب‌های مسئله به صورت آنتی‌بادی و تابع هدف به عنوان آنتی‌ژن فرض می‌شوند. آنتی‌بادی‌ای که بیشتر با آنتی‌ژن منطبق شود جواب مسئله است [۱]. الگوریتم جمعیت مورچه‌ها از اجتماعی از مورچه‌های مصنوعی تشکیل شده است و اغلب برای حل مسائلی که بصورت گراف قابل بیان هستند به کار می‌رود. در این الگوریتم مورچه‌ها مسیر بین گره‌ها را برای بررسی وضعیت جواب مسئله جستجو می‌کنند. هر مورچه کشفیات خود را از طریق ردپا در مسیرها به جا می‌گذارد و از طریق ردپای سایر مورچه‌ها از تجربیات آنها بهره می‌برد [۲]. الگوریتم بهینه‌سازی جمعیت ذره‌ها از اجتماعی از ذرات تشکیل شده که در فضای مسئله حرکت می‌کنند. هر ذره بهترین موقعیت کشف شده‌ی جمعیت تاکنون را می‌بیند و به سمت بهترین موقعیتی که خودش تاکنون کشف کرده و بهترین موقعیتی که تاکنون توسط ذرات جمعیت پیدا شده است شتاب می‌گیرد [۱۲].

در این بین الگوریتم جستجوی گرانشی که در سال ۲۰۰۹ توسط راشدی و همکاران ارائه گشت، یکی از جدیدترین انواع الگوریتم‌های جستجوی تصادفی می‌باشد و با الهام از قانون گرانش که با داستان معروف افتادن سیب از درخت توسط نیوتن کشف گردید ایجاد گردیده است [۱]. این الگوریتم جستجوی تصادفی نوین در این پژوهش ابزار بهینه‌سازی سازه‌ای واقع گردیده است و نتایج حاصل از آن با نتایج برخی از الگوریتم‌های رایج همچون الگوریتم وراثتی [۳]، الگوریتم جمعیت مورچگان [۲]، الگوریتم جمعیت ذرات [۱۲]، الگوریتم جستجوی هارمونی [۱۰] و چندین الگوریتم دیگر مقایسه گردیده، کارایی این الگوریتم در حل مسائل بهینه‌سازی سازه‌ای سنجیده می‌گردد. بخش‌های بعدی این مقاله به ترتیب بخش دوم مربوط به معرفی و فرمولبندی مسئله‌ی بهینه‌سازی گسسته مورد نظر، بخش سوم مربوط به معرفی الگوریتم جستجوی گرانشی،



بخش چهارم حل چند مثال عددی و بخش پنجم نتایج حاصل و بحث بر روی این نتایج می‌باشند.

۲- فرمولبندی مسئله بهینه‌سازی

همچون هر مسئله بهینه‌سازی دیگری، مسائل بهینه‌سازی سازه‌ای و بویژه مسئله بهینه‌سازی وزن خرپاها نیز نیاز به تعریف تابع هدف به صورت تابعی از متغیرهای طراحی دارند. این رابطه میبایست در برگیرنده‌ی شرایط و قیود مسئله نیز باشد و این قیود را نیز به نحوی لحاظ نماید. در این پژوهش رویکرد توابع پنالتی و بطور خاص تابع پنالتی درجه‌ی دوم برای لحاظ نمودن قیود مسئله بهینه‌سازی برگزیده شده است و چنانکه در نتایج بخش چهارم خواهیم دید تناسب خوبی نیز با الگوریتم جستجوی گرانشی از خود نشان می‌دهند. در صورتی که p_1 تا p_n به ترتیب پنالتی‌های مربوط به قیود اول تا n ام باشند، تابع هدف از رابطه‌ی شماره‌ی ۱ محاسبه می‌گردد.

$$F(x) = \left(\sum_{i=1}^n \gamma_i \cdot L_i \cdot A_i \right) + \sum_{i=1}^{noc} P_i \cdot r_i \quad (1)$$

در رابطه‌ی فوق Y وزن واحد جرم، L_i طول عضو i ام و A_i سطح مقطع عضو i ام می‌باشد. $F(x)$ تابع هدف و noc تعداد قیود مسئله بهینه‌سازی می‌باشند.

۳- الگوریتم جستجوی گرانشی

در این پژوهش الگوریتم جستجوی گرانشی که در سال ۲۰۰۹ توسط راشدی و همکاران ارائه گشت بعنوان الگوریتم بهینه‌ساز مورد استفاده قرار گرفته است و بدین جهت در ابتدا نیازمند تعریف کامل مفاهیم و ایده‌های بکار گرفته‌شده در این الگوریتم می‌باشیم. در این قسمت مفاهیم اصلی و کلیدی بکار رفته در الگوریتم جستجوی گرانشی شرح داده می‌شوند و در بخش بعدی مثال‌های عددی در رابطه با کاربرد آن بصورت مفصل ارائه خواهند گشت. در طبیعت یکسری قوانین کلی حاکمند که قانون گرانشی یکی از معروفترین آنهاست. وجود جاذبه در داستان افتادن سیب

از درخت توسط نیوتن مورد توجه قرار گرفت و وی اینگونه نتیجه‌گیری نمود که هر جسمی در این جهان اجسام دیگر را با نیرویی بسمت خود جذب می‌کند که این نیرو با حاصل ضرب جرم دو جسم (جسم کشنده و کشیده شده) نسبت مستقیم و با مجذور فاصله‌ی بین آنها نسبت عکس دارد. رابطه‌ی شماره‌ی ۲ همین نیرو را معرفی می‌نماید. در الگوریتم جستجوی گرانشی نیز، هر عامل بر عامل دیگر در راستای هر بعد یا هر متغیر طراحی نیرو وارد می‌نماید.

$$F_{ij}^d(t) = G(t) \frac{M_i(t) \cdot M_j(t)}{R_{ij}(t) + \epsilon} (x_j^d(t) - x_i^d(t)) \quad (2)$$

$G(t)$ ثابت گرانش است و از آنجا که در طبیعت ثابت شده است این ثابت با آهنگ بسیار کندی رو به کاهش دارد، در الگوریتم گرانشی نیز تابعی نزولی به آن نسبت داده می‌شود. در ادامه در مورد این تابع نزولی بیشتر بحث می‌گردد. M جرم نسبت داده شده به هر عامل و R فاصله‌ی بین اجرام است. ϵ عددی مثبت و بسیار کوچک در نظر گرفته می‌شود. x نیز موقعیت هر عامل را در راستای بعد d ام آن نمایش می‌دهد. الگوریتم بهینه‌سازی گرانشی از نیروی اکتشافی نیوتن الهام گرفته شده است و همانند هر الگوریتم ابتکاری الهام گرفته‌شده از طبیعت دارای پارامترهای قابل تنظیم بصورت شهودی می‌باشد. عوامل جستجوگر فضای جستجوی مسئله در این الگوریتم مجموعه‌ای از اجرام می‌باشند که می‌توانند به صورت سیاره‌های یک منظومه در نظر گرفته شوند. منظومه‌ی بهینه مثل یک سیاهچاله سیاره‌ها را بسوی خود می‌کشند.

طبق قانون اول نیوتن هر جسم حالت سکون یا حرکت یکنواخت خود را بر روی خط راست حفظ می‌کند مگر اینکه تحت تاثیر نیرو یا نیروهایی مجبور به تغییر آن حالت شود. طبق قانون دوم نیوتن وقتی به جسمی نیرویی وارد

می‌شود شتابی می‌گیرد که به نیرو و جرم جسم بستگی دارد. هرچه نیرو بزرگتر باشد شتاب نیز بزرگتر است و هرچه جرم جسم بزرگتر باشد شتاب آن کوچکتر است. این قانون نیز مطابق با رابطه‌ی شماره‌ی ۳ بیان می‌گردد.

$$F_i^d = M_i \cdot a_i^d \quad (3)$$

سیستم را بصورت مجموعه‌ای از m جرم تصور کنید. موقعیت هر جرم نقطه‌ای از فضا است که جوابی از مسئله است. موقعیت بعد d از جرم i با X_i^d نمایش داده می‌شود. پس با توجه به آنچه گفته شد برای یک جرم به تعداد ابعاد مسئله مورد نظر موقعیت در هر تکرار وجود دارد.

بنا بر همین وجود d موقعیت ذکر شده در راستای هر بعد و برای هر متغیر طراحی، دارای به ترتیب d نیرو، شتاب، سرعت و نهایتاً موقعیت به روز شده در راستای d بعد مختلف (d به تعداد متغیرهای طراحی) خواهیم بود. رابطه‌ی شماره‌ی ۲ نیروهای وارد از سوی هر عامل بر عامل دیگر در راستای بعد d ام را نشان می‌دهد.

برایند نیروهای وارد بر هر عامل از سوی سایر عوامل و در راستای بعد d ام از رابطه‌ی شماره‌ی ۴ محاسبه می‌گردد.

$$F_i^d = \sum_{j=1}^{n-1} rand \cdot F_{ij}^d \quad (4)$$

این رابطه برایند نیروهای وارد بر هر عامل از سوی عوامل دیگر در راستای هر بعد را مجموع ضربی از تک تک نیروهای وارد بر آن عامل از سوی سایر عوامل و در راستای بعد مورد نظر تعریف می‌نماید. $rand$ در رابطه‌ی ۴ عددی تصادفی با انتخابی یکنواخت از بازه صفر تا یک می‌باشد. با تقسیم نیروهای برایند بدست آمده بر هر عامل در راستای هر بعد بر جرم نسبت داده شده به آن عامل توسط الگوریتم جستجوی گرانشی شتاب وارد بر آن عامل در راستای همان بعد بدست خواهد آمد. روابط ۵ و ۶ به

ترتیب شتاب در راستای هر بعد (هر متغیر طراحی) و جرم نسبت داده شده به آن عامل را بدست می‌دهند.

$$a_i^d = \frac{F_i^d}{M_i} \quad (5)$$

$$M_i = \frac{fit_i(t) - worst(t)}{best(t) - worst(t)} \quad (6)$$

fit برازندگی هر عامل، $worst$ بدترین برازندگی موجود در آن تکرار که در مسائل بیشینه‌یابی کمترین برازندگی و در مسائل کمینه‌یابی بیشترین آنها خواهد بود. همچنین بهترین برازندگی در هر تکرار می‌باشد که در مسائل کمینه‌یابی کمترین و در مسائل بیشینه‌یابی بیشترین برازندگی موجود در آن تکرار خواهد بود. مطابق رابطه‌ی شماره‌ی ۷ سرعت در تکرار بعدی برای هر عامل جستجوگر و در راستای هر بعد برابر با حاصل جمع ضربی از سرعت فعلی و شتاب بدست آمده از رابطه‌ی ۵ در راستای همان بعد می‌باشد.

$$V_i^d(t+1) = rand_i \times V_i^d(t) + a_i^d(t) \quad (7)$$

و در نهایت رابطه‌ی بدست آمده برای موقعیت بروز شده عوامل بصورت رابطه‌ی شماره‌ی ۸ می‌باشد که پایان بخش یک تکرار الگوریتم جستجوی گرانشی است.

$$x_i^d(t+1) = x_i^d(t) + V_i^d(t+1) \quad (8)$$

رابطه‌ی شماره‌ی ۸ موقعیت مرحله‌ی بعدی را مجموع موقعیت قبلی و سرعت مرحله‌ی بعدی که از رابطه‌ی شماره‌ی ۷ بدست آمد می‌داند. در تکرار اول تمامی سرعتها برای همه‌ی عوامل و در راستای تمامی ابعاد برابر با صفر در نظر گرفته می‌شوند. این الگوریتم نیز همانند تمامی الگوریتم‌های جستجوی تصادفی در ابتدا با ایجاد جمعیت اولیه‌ی تصادفی و پخش نمودن آنها در فضای جستجو به کاوش فضای جستجوی مسائل بهینه‌سازی می‌پردازد و با



استفاده از تابع هدف معرفی شده در بخش پیشین و طی نمودن مراحل بحث شده در بخش حاضر، دست به بروز نمودن موقعیت عوامل خود (ایجاد برانندگی‌های بهتر یا تغییر موقعیت عوامل) خواهد زد. در توالی تکرارها، این روند منجر به یافتن جواب بهینه می‌گردد. مراحل حل مسئله‌ی بهینه‌سازی به کمک الگوریتم جستجوی گرانشی به قرار زیر می‌باشد:

گام اول - ابتدا جمعیت اولیه که مجموعه‌ای از جواب‌های مسئله‌ی بهینه‌سازی است ایجاد می‌گردد.

گام دوم - هر یک از سازه‌های جمعیت جاری تحت بارهای وارده تحلیل استاتیکی می‌شوند.

گام سوم - با استفاده از نتایج تحلیل سازه، مقدار تابع جریمه برای هر عضو محاسبه می‌شود.

گام چهارم - برانندگی متناظر با هر یک از اعضای جمعیت اولیه محاسبه گردد.

گام پنجم - بهترین و بدترین برانندگی‌ها محاسبه می‌گردند.

گام ششم - پارامتر جرم متناظر برای هر یک از عوامل جستجو محاسبه می‌گردد.

گام هفتم - محاسبه‌ی پارامتر ثابت گرانش برای تکرار جاری انجام می‌شود.

گام هشتم - محاسبه‌ی نیروهای وارد بر هر جرم از سوی سایر اجرام انجام می‌گردد.

گام نهم - محاسبه‌ی برایندهای نیروهای وارد بر هر جرم در راستای هر بعد.

گام دهم - محاسبه‌ی شتاب وارد بر هر عامل در راستای هر بعد.

گام یازدهم - محاسبه‌ی سرعت هر عامل در راستای هر بعد.

گام دوازدهم - محاسبه‌ی مکان جدید اشغال شده توسط هر عامل.

گام سیزدهم - کنترل شرایط توقف و در صورت عدم برآورده شدن شرایط بازگشت به گام دوم.

در این پژوهش از رابطه‌ی نمایی و البته نزولی شماره‌ی ۹ برای محاسبه‌ی ثابت گرانش در هر تکرار استفاده شده است.

$$G(t) = G_0 e^{-\alpha \frac{t}{T}} \quad (9)$$

در این رابطه e عدد نپر، α ثابتی مثبت (در اینجا ۲۰)، t شماره‌ی تکرار حاضر و T تعداد کل تکرارها خواهد بود. G_0 نیز میزان ابتدایی ثابت گرانش در تکرار اول بوده و معادل ۱۰۰ در نظر گرفته شده است. در بخش بعدی چند مثال عددی با استفاده از الگوریتم جستجوی گرانشی و به جهت نمایش کارایی این الگوریتم در حل مسائل بهینه‌سازی سازه‌ای حل خواهند شد.

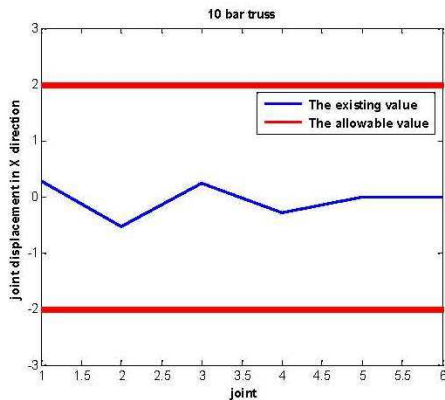
۴- مثال‌های عددی

در این قسمت دو مثال بهینه‌سازی سازه‌ای با سطح مقطع‌های انتخاب شده از میان پروفیل‌های مشخص و به صورت متغیرهای طراحی گسسته به وسیله‌ی الگوریتم جستجوی گرانشی بهینه خواهند گردید و نتایج حاصل از آن، با نتایج بدست آمده از حل همان مثال‌ها و به وسیله‌ی سایر الگوریتم‌های جستجوی تصادفی رایج مقایسه خواهد شد. در مورد هر مثال نمودار همگرایی الگوریتم جستجوی گرانشی و همچنین نمودارهای قیود مسئله ارائه گردیده اند.

۴-۱- مثال اول

بعنوان مثال اول خریای ۱۰ عضوی شکل ۱ را توسط الگوریتم جستجوی گرانشی بهینه خواهیم نمود. این خریا دارای اعضای با مدول الاستیسیته‌ی ۱۰۰۰۰۰ ksi ($\frac{kg}{m^3} 2767/9$) و چگالی ۰/۱ ($Mpa 68950$) می‌باشد. نیروهای مشخص شده با حرف P در شکل ۱ معادل ۱۰۰ kips ($kg 45400$) می‌باشند.

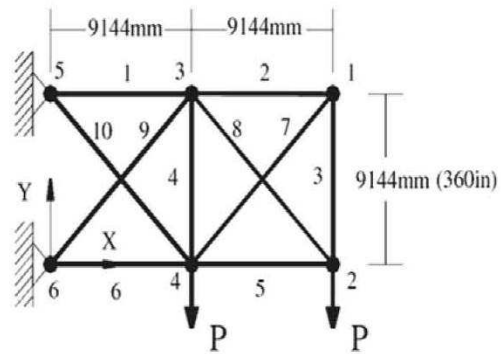




شکل (۲): تغییر مکان گرهی در راستای X

جدول (۲): مقایسه‌ی عملکرد الگوریتم‌ها

GSA Present work	Kaveh & Shojaee [۲]	Camp et.al [۴]	Meth.
۲۱۶/۰۷	۲۱۶/۰۷	۲۱۶/۰۷	$x_1 (cm^2)$
۱۰/۴۵	۱۰/۴۵	۱۰/۴۵	$x_2 (cm^2)$
۱۰/۴۵	۱۰/۴۵	۱۰/۴۵	$x_3 (cm^2)$
۱۰/۴۵	۱۰/۴۵	۱۰/۴۵	$x_4 (cm^2)$
۹۱/۵۹	۹۱/۵۹	۸۷/۰۷	$x_5 (cm^2)$
۱۴۷/۷	۱۴۷/۷	۱۷۰/۹۲	$x_6 (cm^2)$
۱۰/۴۵	۱۰/۴۵	۱۰/۴۵	$x_7 (cm^2)$
۱۴۱/۹	۱۴۱/۹	۱۴۱/۹	$x_8 (cm^2)$
۱۴۷/۷	۱۴۷/۷	۱۴۷/۷	$x_9 (cm^2)$
۵۱/۰۸	۵۱/۰۸	۴۶/۵۶	$x_{10} (cm^2)$
۲۴۹۲/۸	۲۴۹۲/۸	۲۵۲۲/۸	Weight (kg)



شکل (۱): خرابی ۱۰ عضوی مثال اول

قیود این مسئله‌ی بهینه‌سازی از دو نوع قید حداکثر تنش مجاز و حداکثر تغییر مکان تکیه گاهی می‌باشند. حداکثر تنش کششی و فشاری اعضا معادل ۲۵ ksi (۱۷۲/۳۷۵ Mpa) می‌باشد و حداکثر تغییر مکان گرهی آن ۲ in (۵/۰۸ cm) در نظر گرفته شده‌اند. جدول شماره‌ی ۱ مقادیر سطح مقطع‌های پروفیل‌های مورد نظر برای این مثال را ارائه می‌دهد.

جدول (۱): سطح مقطع پروفیل‌های موجود (cm^2).

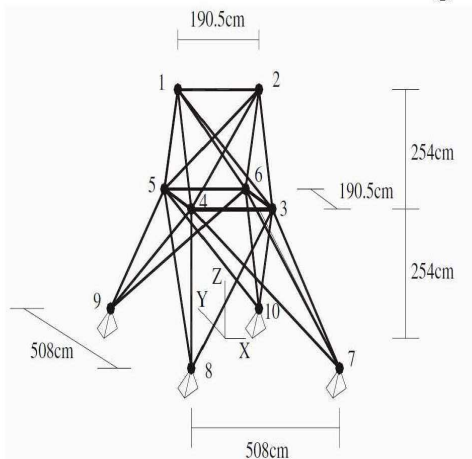
۱۰/۴۵	۱۱/۶۱	۱۲/۸۳	۱۳/۷۳	۱۵/۳۵	۱۶/۸۹
۸/۵۷	۱۸/۸۹	۱۹/۹۳	۲۰/۱۸	۲۱/۸	۲۲/۳۸
۲۲/۸۹	۲۳/۴۱	۲۴/۷۶	۲۴/۹۶	۲۵/۰۲	۲۶/۹۶
۲۷/۲۲	۲۸/۹۶	۲۹/۶	۳۰/۹۶	۳۲/۰۵	۳۳/۰۲
۳۸/۳۱	۴۶/۵۶	۵۱/۰۸	۷۴/۱۷	۸۷/۰۷	۸۹/۶۵
۹۱/۵۹	۹۹/۹۷	۱۰۳/۲	۱۰۹	۱۲۱/۲۶	۱۲۸/۳۵
۱۴۱/۹	۱۴۷/۷	۱۸۳/۸۲	۱۹۳/۵	۲۱۶/۰۷	

جدول شماره‌ی ۲ نتایج بدست آمده از حل این مثال توسط الگوریتم جستجوی گرانشی و نتایج تحقیقات محققین دیگر را ارائه می‌دهد. شکل‌های ۳ و ۴ نیز تغییر مکان‌های گرهی را با تغییر مکان‌های مجاز مقایسه می‌نماید.

الگوریتم جستجوی گرانشی با وجود جمعیت اولیه‌ای معادل ۲۰ عامل، در تکرار ۵۷ ام به جواب ۲۴۹۲/۸ کیلوگرم همگرا می‌گردد که تعداد ارزیابی تنها ۱۱۴۰ آنالیز را نمایش می‌دهد. این میزان آنالیز در الگوریتم‌های جستجوی تصادفی تعدادی بسیار ناچیز و درخور توجه می‌باشد. در ضمن قید حاکم بر مسئله در حل این مثال تغییر مکان گرهی گره دوم در راستای قائم می‌باشد که به خوبی و با استفاده‌ی مناسب از ظرفیت مجاز ارضاء گردیده است. در مورد نتیجه‌ی بدست آمده از الگوریتم جستجوی گرانشی و دلایل آن در بخش نتایج بیشتر بحث خواهد شد.

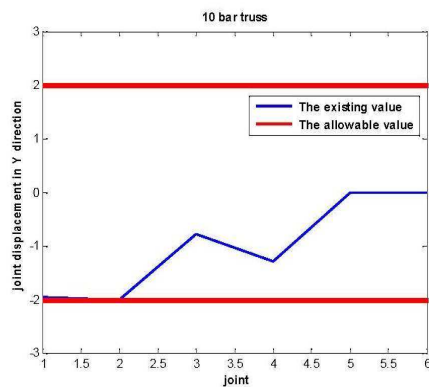
۴-۲- مثال دوم

در این قسمت خرپای ۲۵ عضوی شکل ۶ را توسط الگوریتم جستجوی گرانشی بهینه خواهیم نمود. این خرپا نیز دارای اعضایی با مدول الاستیسیته‌ی 10000 ksi (68950 Mpa) و چگالی $0.1 \frac{\text{lb}}{\text{in}^3}$ ($2767/9 \frac{\text{kg}}{\text{m}^3}$) می‌باشد.



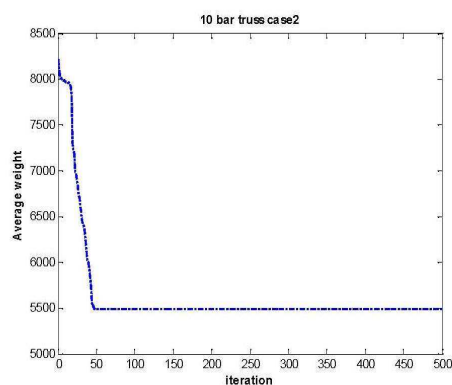
شکل (۶): خرپای ۲۵ عضوی مثال دوم

قیود این مثال نیز حداکثر تنش کششی و فشاری معادل 40 ksi ($275/6 \text{ Mpa}$) و تغییر مکان گرهی در هر سه

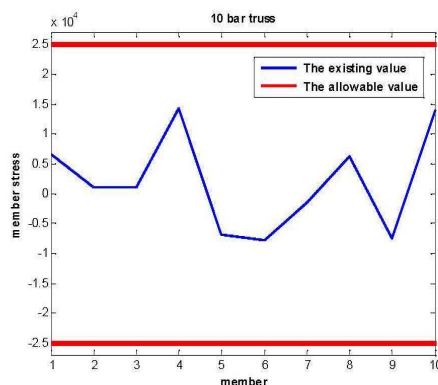


شکل (۳): تغییر مکان گرهی در راستای Y

شکل شماره ۴ نمودار همگرایی الگوریتم جستجوی گرانشی را در حل این مثال نمایش می‌دهد. شکل شماره ۵ نیز نمودار تنش‌های موجود اعضا در برابر تنش‌های حداکثری را نمایش می‌دهد.



شکل (۴): همگرایی الگوریتم جستجوی گرانشی



شکل (۵): تنش‌های اعضا در برابر تنش‌های حداکثری

الگوریتم ابتکاری دیگر نیز در جدول شماره ۵ با یکدیگر مقایسه شده‌اند.

جدول (۴): سطح مقطع پروفیل های فولادی مثال دوم

۰/۶۴۵	۱/۲۹	۱/۹۳	۲/۵۸	۳/۲۲
۳/۸۷	۴/۵۱	۵/۱۶	۵/۸	۶/۴۵
۷/۰۹	۷/۷۴	۸/۳۸	۹/۰۳	۹/۶۷
۱۰/۳۲	۱۰/۹۶	۱۱/۶۱	۱۲/۲۵	۱۲/۹
۱۳/۵۴	۱۴/۱۹	۱۴/۸۳	۱۵/۴۸	۱۶/۱۲
۱۶/۷۷	۱۸/۰۶	۱۹/۳۵	۲۰/۶۴	۲۱/۹۳

جدول (۵): مقایسه‌ی عملکرد الگوریتم‌ها

GSA	Kaveh & Shojaee [۲]	Erbatur et al. [۶]	Meth.
۰/۶۴۵	۰/۶۴۵	۰/۶۴۵	$x_1 (cm^2)$
۲/۵۸	۱/۹۳	۱۲/۲۵	$x_2 (cm^2)$
۲۱/۹۳	۲۱/۹۳	۲۰/۶۴	$x_3 (cm^2)$
۰/۶۴۵	۰/۶۴۵	۰/۶۴۵	$x_4 (cm^2)$
۱۰/۹۶	۱۳/۵۴	۷/۰۹	(cm^2) x_5
۶/۴۵	۶/۴۵	۵/۸	$x_6 (cm^2)$
۳/۲۲	۳/۲۲	۲/۵۸	$x_7 (cm^2)$
۲۱/۹۳	۲۱/۹۳	۲۱/۹۳	$x_8 (cm^2)$
۲۱۹/۵	۲۲۰/۱	۲۲۴/۲	Weight (kg)

راستا برابر با 0.3 in (0.889 cm) می‌باشد. نیروهای وارد بر سازه نیز در گره اول نیروی ۱ kips (4.454 KN) در راستای X و دو نیروی ۱۰ kips (44.54 KN) در راستای Y و Z می‌باشند. همچنین دو نیروی ۱۰ kips (44.54 KN) در راستای منفی Y و Z گره دوم و نیروی ۰/۵ kips (2.227 KN) و ۰/۶ kips (2.672 KN) در راستای X و به ترتیب در گره‌های سوم و چهارم وارد می‌شوند.

۲۵ عضو این سازه مطابق جدول شماره ۳ در هشت گروه دسته بندی می‌گردند.

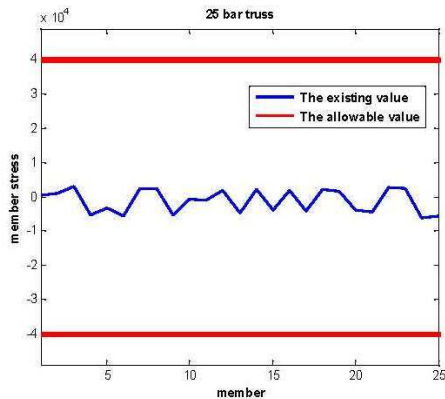
جدول (۳): گروه بندی و گره های ابتدا و انتهای اعضا

گروه	اعضا	عضو ابتدا و انتها
۱	۱	۱:۲
۲	۲، ۳، ۴ و ۵	۱:۴ - ۲:۳ - ۱:۵ - ۲:۶
۳	۶، ۷، ۸ و ۹	۱:۳ - ۲:۴ - ۱:۶
۴	۱۰ و ۱۱	۳:۶ - ۴:۵
۵	۱۲ و ۱۳	۳:۴ - ۴:۶
۶	۱۴، ۱۵، ۱۶ و ۱۷	۳:۱۰ - ۴:۷ - ۴:۹ - ۵:۸
۷	۱۸، ۱۹، ۲۰ و ۲۱	۳:۸ - ۴:۷ - ۴:۹ - ۵:۱۰
۸	۲۲، ۲۳، ۲۴ و ۲۵	۳:۷ - ۴:۸ - ۴:۹ - ۵:۱۰

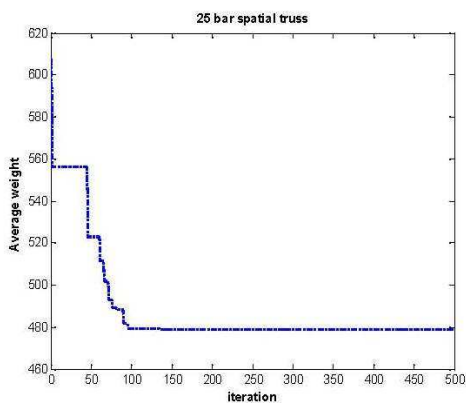
سطح مقطع پروفیل های فولادی مورد استفاده در بهینه سازی این سازه در جدول شماره ۴ آورده شده است. نتایج حاصل از الگوریتم جستجوی گرانشی و چند



شکل شماره ۱۰ تنش‌های موجود را در کنار تنش‌های حداکثری نمایش می‌دهد و شکل ۱۱ نمودار همگرایی الگوریتم جستجوی گرانشی را در حل این مثال ارائه می‌دهد.



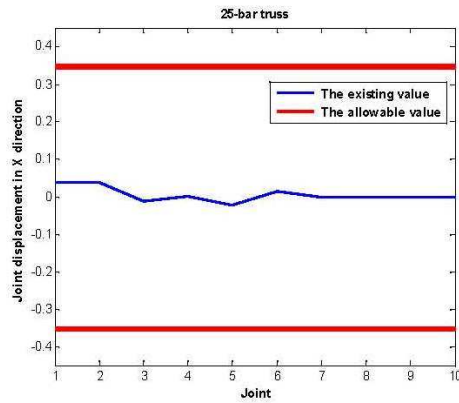
شکل (۱۰) :: تنش موجود اعضا و تنش حداکثری



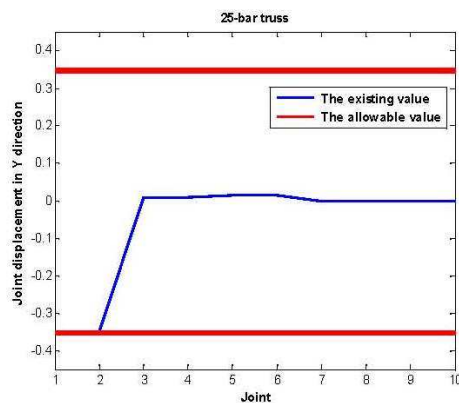
شکل (۱۱) :: نمودار همگرایی (وزن بر حسب پاند)

الگوریتم جستجوی گرانشی با تعداد ۲۰ جمعیت اولیه و در ۱۴۷ تکرار به جواب ۲۱۹/۵ کیلوگرم دست یافته است. تعداد ۲۹۴۰ ارزیابی انجام شده توسط الگوریتم جستجوی گرانشی در حل این مسئله در نوع خود نشانگر قدرت همگرایی مناسب این الگوریتم در کنار کاوش و بهره‌وری قابل قبول آن می‌باشد.

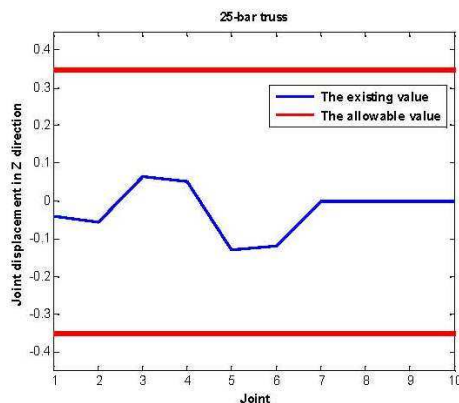
نمودار تغییر مکان گرهی در راستای هر سه بعد در اشکال ۸، ۷ و ۹ آورده شده‌اند.



شکل (۷) :: تغییر مکان گرهی در راستای X



شکل (۸) :: تغییر مکان گرهی در راستای Y



شکل (۹) :: تغییر مکان گرهی در راستای Z

۵- نتیجه گیری

الگوریتم‌های جستجوی تصادفی را به وسیله نتایج نهایی بدست آمده از حل مسائل گوناگون توسط آنها و همچنین زمان حل مسئله با یکدیگر مقایسه می‌نمایند و اینگونه تلقی می‌گردد که هر چه الگوریتم در زمان کوتاه‌تر جواب مناسب‌تری برای مسئله بیابد، کارایی بیشتری در حل آن مسئله از خود نشان داده است. زمان دستیابی به جواب بهینه را با مقایسه‌ی تعداد ارزیابی‌های انجام شده می‌سنجند. نکته‌ی قابل بحث دیگر این است که هرگز نمی‌توان چنین ادعا نمود که یک الگوریتم خاص در حل تمامی انواع مسائل بهینه‌سازی عملکرد مناسب‌تری را از خود نشان می‌دهد و اصولاً همین عامل، دلیلی بر ارزشمند بودن ایجاد الگوریتم‌های جستجوی تصادفی نوین و یا ایجاد بهبود در الگوریتم‌های موجود می‌باشد. از مقایسه‌ی نتایج بدست آمده توسط الگوریتم جستجوی گرانشی در مثال‌های عددی فوق با نتایج سایر الگوریتم‌های رایج، نزدیکتر شدن الگوریتم جستجوی گرانشی به بهینه‌ی مطلق مشهود است. تعداد ارزیابی‌های مورد نیاز الگوریتم جستجوی گرانشی در دستیابی به جواب‌های نهایی ارائه شده نیز در مقایسه با سایر الگوریتم‌ها میزان بسیار کمتری را نمایش می‌دهد که این قدرت همگرایی مناسب در کنار نتایج مناسب بدست آمده جز با وجود کاوش و بهره‌وری مناسب الگوریتم جستجوی گرانشی میسر نمی‌باشد. در مورد مثال اول گرچه نتایج پایانی مرجع ۲ و الگوریتم جستجوی گرانشی یکسان هستند اما نبایستی از یاد برد که الگوریتم جستجوی گرانشی برای یافتن این جواب‌نهایی تنها ۱۱۴۰ ارزیابی را مورد استفاده قرار داده است در حالی که مرجع شماره‌ی ۲ تعداد ۱۰۹۴۰ ارزیابی را مورد استفاده قرار داده است [۲] که زمان دستیابی به جواب بهینه را

بسیار بالا می‌برد. پیشتر ذکر گردید که کارایی الگوریتم‌های جستجوی ابتکاری علاوه بر نتایج نهایی بدست آمده با زمان دست‌یابی به نتایج نهایی مقایسه می‌گردد. در مورد مثال اول فضای جستجوی کوچک مسئله عاملی است برای اینکه الگوریتم‌های جستجوی تصادفی مختلف در کاوش این فضای جستجوی کوچک موفق عمل نمایند و لذا ارزیابی عملکرد الگوریتم‌ها از دایره‌ی نتایج وارد حیطه‌ی زمان دستیابی به جواب نهایی مناسب گردد. اهمیت دستیابی به جواب بهینه‌ی مناسب در زمان کوتاه تا آنجاست که می‌توان آنرا یکی از علل جایگزینی روش‌های کلاسیک با الگوریتم‌های جستجوی تصادفی دانست. در حل مثال دوم نیز الگوریتم جستجوی گرانشی نه تنها جواب نهایی بدست آمده را بهبود بخشیده است، بلکه تعداد ارزیابی‌ها را نیز از میزان ۱۱۲۵۰ ارزیابی به ۲۹۴۰ ارزیابی کاهش داده است [۲]. کاوش مناسب فضای جستجو در کنار بهره‌وری مناسب در تکرارها و گام‌های پایانی از علل این دستیابی به جواب نهایی مناسب در زمان قابل قبول توسط الگوریتم جستجوی گرانشی می‌باشد. یکی از مسائلی که از دیر باز باعث نگرانی محققین در زمینه‌ی الگوریتم‌های ابتکاری بوده است همگرایی زودرس و افتادن در دام بهینه‌های محلی است که با توجه به دست‌یابی به نتایج مناسب از سوی الگوریتم گرانشی این نگرانی در مورد این الگوریتم ابتکاری نوین به چشم نمی‌خورد و دستیابی به نتایج مناسب در تعداد ارزیابی پایین با توجه به نتایج مناسب بدست آمده به جای همگرایی زودرس، قدرت همگرایی بالا تلقی می‌گردد. به عنوان تحقیقات آینده و با توجه به نمودار همگرایی الگوریتم جستجوی گرانشی به نظر می‌آید که این الگوریتم در گام‌های اولیه پس از بهبود نسبی جواب در هر تکرار به



10- KS. Lee, ZW. Geem 2004, Computers and Structures, "A new structural optimization method based on the harmony search algorithm".

11- A. Kaveh, S. Talatahari 2009, Computers and Structures, "Size optimization of space trusses using Big Bang-Big Crunch algorithm".

12- S. He, E. Prempain, QH. Wu 2004, Eng Optimiz, "An improved particle swarm optimizer for mechanical design optimization problems".

13- H. Adeli, NT. Cheng 1993, Structures, Journal of Aerospace Engineering, "Integrated genetic algorithm for optimisation of space".

14- A. Kaveh, V. Kalatjari 2003, International Journal for Numerical Methods in Engineering, "Topology optimisation of trusses using genetic algorithm, force method and graph theory".

15- CY. Lin, P. Hajela 1992, Engineering Optimisation, "Genetic algorithms in optimisation problems with discrete and integer design variables".

16- L. Lamberti 2008, Computers and Structures, "An efficient simulated annealing algorithm for design optimization of truss structures".

یکباره قدرت کاوش مناسب خود را در چند گام از دست می‌دهد و چندین تکرار با برازندگی حداکثری ثابت سپری می‌گردند. گرچه پس از سپری شدن چند تکرار الگوریتم روند مناسب پیشین را باز می‌یابد اما به نظر می‌رسد که ایجاد شرایطی برای بهبود کاوش الگوریتم مذکور در شرایط عنوان شده راه حل مناسبی برای این نقیصه باشد. از آنجا که ایجاد بهبود در هر الگوریتم جستجوی تصادفی نیازمند آشنایی با نقاط ضعف و قدرت آن الگوریتم می‌باشد لذا بنظر می‌رسد این نقیصه خود راهی برای ایجاد بهبود در الگوریتم جستجوی گرانشی است.

۶- مراجع

1- E. Rashedi, H. Nezamabadi-pour, S. Saryazdi 2009, Information sciences, "GSA: a gravitational search algorithm".

2- A. Kaveh, S. Shojaee 2007, Int. J. Numer. Meth. Engng, "Optimal design of skeletal structures using ant colony optimization".

3- S. Rajeev, CS. Krishnamoorthy 1992, Journal of Structural Engineering, "Discrete optimisation of structures using genetic algorithm".

4- C. Camp, S. Pezeshk, G. Cao 1998, Journal of Structural Engineering, "Optimized design of two-dimensional structures using a genetic algorithm".

5- DM. Zhu 1986, Engineering Optimization, "An improved Templeman's algorithm for optimum design of trusses with discrete member sizes".

6- F. Erbatur, O. Hasancebi, I. Tutuncu, H. Kilic 2000, Computers and Structures, "Optimal design of planar and space structures with genetic algorithms".

7- A. Kaveh, S. Shojaee 2004, Proceedings of the 4th on Engineering Computational Technology, Lisbon, September, "Optimal design of scissor-link foldable structures using genetic algorithm".

8- LF. Escudero 1988, European Journal of Operations Research, "An inexact algorithm for the sequential ordering problem".

9- A. Kaveh, V. Kalatjari 2002, Int J Numer Methods, "Genetic algorithm for discrete sizing optimal design of trusses using the force method".