

بهینه‌سازی تحلیل خمیری قاب‌های خمشی با ارائه‌ی الگوریتم مکان‌یابی پژواکی دلفین باینری

Original Article

امیر ساعدی داریان* (استادیار)

سهیل پالیزی (کارشناس ارشد)

دانشکده‌ی عمران، آب و محیط زیست، دانشگاه شهید بهشتی

ندا فرهودی (استادیار)

مرکز تحقیقات راه، مسکن و شهرسازی، تهران، ایران

تحلیل سازه‌ها و بالانص تعیین مود خرابی آن‌ها از نیازهای اساسی در حوزه‌ی مهندسی عمران است. به دست آوردن این اطلاعات به خصوص در سازه‌های بلندمرتبه، فرایند دشواری است که حتی با استفاده از نرم‌افزارهای تخصصی نیز بسیار زمان‌بر است. در پژوهش حاضر، الگوریتم جدیدی به نام الگوریتم پژواک دلفین باینری ارائه شده است، که با اصلاح الگوریتم پژواک دلفین به دست آمده است. سپس تحلیل خمیری قاب‌های خمشی با استفاده از الگوریتم جدید مذکور، بهینه‌سازی شده است. مقایسه‌ی نتایج نشان می‌دهد که روش ارائه شده با دقت و سرعت بسیار بالا، نتایج بسیار مطلوبی ارائه کرده و مود خرابی سازه را نیز به خوبی نمایان ساخته است. در عین حال اصلاحات صورت گرفته در الگوریتم دلفین، کارکرد جدیدی بر الگوریتم اخیر افزوده است، که از این پس می‌تواند برای مسائل باینری استفاده شود.

amir_saeidi_d@yahoo.com
soheil.palizi@gmail.com
n.farhoudi@bhrc.ac.ir

واژگان کلیدی: تحلیل خمیری، تئوری کینماتیک، قاب خمشی، الگوریتم دلفین باینری.

۱. مقدمه

با هم ترکیب شدند، مکانیزم متناظر با ضریب بار کمینه به‌عنوان مکانیزم فروریزش سازه بیان می‌شود. یکی از مشکلات روش اخیر تحلیل سازه‌هایی است که دهانه‌ها و طبقات زیاد داشته باشند. در این شرایط، ترکیب مکانیزم‌های اولیه با هم برای یافتن مکانیزم بحرانی کاری بسیار خسته‌کننده و در مواردی غیرممکن است. لذا در چنین شرایطی نیاز به روشی است که بتوان با استفاده از آن، ترکیب مکانیزم‌ها را به صورت خودکار و با سرعت بالا به دست آورد. در راستای حل مشکل اخیر، بهترین گزینه استفاده از هوش مصنوعی است. الگوریتم‌های فراابتکاری بخشی از الگوریتم‌های حوزه‌ی هوش مصنوعی هستند که برای یافتن پاسخ بهینه به‌کار می‌روند. در ارتباط با مسئله‌ی تحلیل خمیری قاب‌های خمشی، کاوه و همکاران (۲۰۰۳ و ۲۰۰۴)^[۷-۹] و کوهاما و همکاران (۱۹۹۷)^[۱۰] مطالعاتی را درباره‌ی تعیین ضرایب بار فروریزش قاب‌های صفحه‌یی توسط الگوریتم ژنتیک ارائه کرده‌اند. کاوه و جهانشاهی تحلیل خمیری حالت حدی قاب‌ها را با استفاده از الگوریتم‌های ابتکاری و سیستم الگوریتم کلونی مورچگان مطالعه کرده‌اند.^[۱۱-۱۴] بعد از آن‌ها، جهانشاهی و همکاران (۲۰۱۶) بار فروریزش قاب‌های صفحه‌یی را با استفاده از روش شبکه‌ی عصبی و الگوریتم ژنتیک و همچنین الگوریتم مورچگان انجام داده و نتایج را با یکدیگر مقایسه کرده‌اند.^[۱۵] کاوه و همکاران نیز تحلیل‌های خمیری

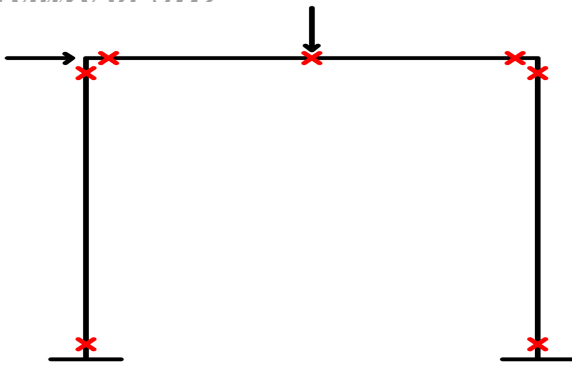
مسئله‌ی تحلیل و طراحی خمیری قاب‌ها از اواسط قرن بیستم توسط پژوهش‌گران مختلفی بررسی شده است. پژوهش‌گران مختلف در طول سال‌ها از روش‌های مختلف و متعددی جهت تحلیل سازه‌ها استفاده کرده‌اند. در این میان، روش مبتنی بر تئوری کینماتیک جایگاه ویژه‌ی دارد. تحلیل حدی، که بر اساس تئوری کینماتیک انجام می‌شود، در مقایسه با روش‌های تحلیل دینامیکی و استاتیکی غیرخطی روشی بسیار سریع و کم‌هزینه است و در عین حال نتایج قابل اعتمادی را نیز ارائه می‌دهد. در پژوهش حاضر، نیز از همین روش استفاده شده است. به این صورت که تمام مکانیزم‌های فروریزش ممکن در سازه بررسی و بار نهایی فروریزش آن‌ها محاسبه می‌شود. سپس کمینه‌ی مقدار بار فروریزش به‌عنوان بار فروریزش بحرانی و مکانیزم متناظر آن به‌عنوان مکانیزم فروریزش سازه معرفی می‌شود.

روشی که در آن مکانیزم‌های اولیه با هم ترکیب می‌شوند، نخستین بار توسط نیل و سایموند (۱۹۵۰ و ۱۹۵۲)^[۱۲] بیان شد و در سال‌های بعد نیز توسط سایر پژوهش‌گران توسعه داده شد.^[۳-۶] در روش اشاره شده، پس از این‌که مکانیزم‌های اولیه

* نویسنده مسئول

تاریخ: دریافت ۱۳۹۷/۹/۲۴، اصلاحیه ۱۳۹۷/۱۲/۱، پذیرش ۱۳۹۷/۱۲/۲۶.

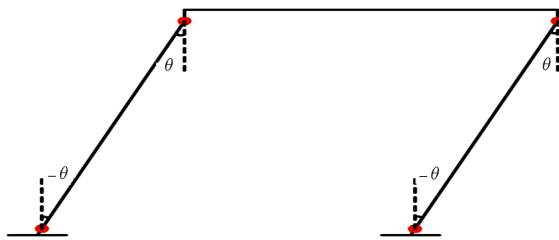
DOI:10.24200/J30.2019.51929.2443



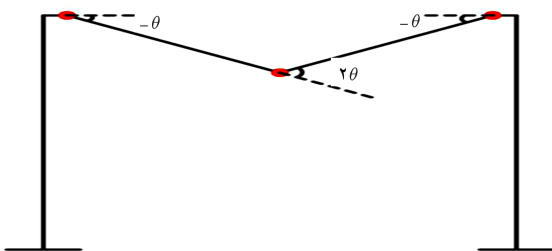
شکل ۱. نقاط بحرانی تشکیل مفصل.



الف) مکانیزم گره؛



ب) مکانیزم طبقه؛



ج) مکانیزم تیر.

شکل ۲. نحوه‌ی چیدمان مفاصل خمیری در قاب یک طبقه‌ی یک دهانه.

به‌منظور تعیین مکانیزم‌های مستقل از تئوری کار مجازی استفاده می‌شود. از آنجایی که مکانیزم‌های مذکور، حاصل تغییر شکل‌های بسیار بزرگ دورانی هستند، لذا در قیاس با آن‌ها می‌توان از تغییرشکل‌های محوری که بسیار کوچک هستند، صرف‌نظر کرد. معادله‌ی کار برای یک مکانیزم فرضی بیان می‌کند که مجموع انرژی ایجاد شده به دلیل نیروی اعمالی W_i و جابه‌جایی متناظر δ_i یا به عبارتی $\sum W_i \delta_i$

قاب‌های صفحه‌یی را با استفاده از بهینه‌سازی اجسام در حال برخورد و نسخه‌ی توسعه یافته‌ی آن انجام داده‌اند.^[۱۷،۱۶] همچنین از آخرین مطالعات صورت گرفته می‌توان به گریسو و همکاران (۲۰۱۷) اشاره کرد که به بررسی پیش‌بینی خرابی لرزه‌یی قاب‌های سازه‌یی با الگوریتم بهینه‌سازی ژنتیک پرداخته‌اند. در پژوهش حاضر، به منظور شبیه‌سازی حالت لرزه‌یی از بار یکنواخت ثابت به صورت ثقلی و بار جانبی افزایشی اعمال شده در طبقات استفاده شده است.^[۱۸]

الگوریتم‌های فراابتکاری الهام گرفته از طبیعت، با به تصویر کشیدن پدیده‌های بیولوژیکی یا فیزیکی، مسائل بهینه‌سازی را حل می‌کنند. گروهی از الگوریتم‌های الهام گرفته از طبیعت، شامل تکنیک‌های متداولی است که رفتار اجتماعی گروه‌های حیوانات را تقلید می‌کنند. الگوریتم‌های بهینه‌سازی فراابتکاری روزه روز در فرایندهای مهندسی محبوب‌تر می‌شوند، زیرا آن‌ها: ۱. به مفاهیم نسبتاً ساده تکیه می‌کنند و کار با آن‌ها آسان است، ۲. می‌توانند از بهینه‌ی محلی دور بمانند و ۳. می‌توانند در طیف گسترده‌یی از مسائل مربوط به رشته‌های مختلف استفاده شوند. استفاده‌ی فقط از الگوریتم‌های مذکور در مسائل باینری به گونه‌یی که بتوان پاسخ‌های بهتری را دریافت کرد، در گذشته در مورد برخی از مهم‌ترین الگوریتم‌های اخیر انجام شده است. نخستین بار الگوریتم باینری ازدحام ذرات به صورت نسخه‌یی جداگانه توسط کندی و ابرهارت (۱۹۹۷) ارائه شد.^[۱۹] چند سال بعد، همین الگوریتم با شیوه‌یی جدید توسط احمدیه و همکاران (۲۰۰۷) باینری شد.^[۲۰] همچنین میرجلالی و همکاران (۲۰۱۴) الگوریتم خفاش باینری و کومار و همکاران (۲۰۱۸)، الگوریتم وال باینری را ارائه کردند.^[۲۱،۲۲]

عمده‌ی فعالیت پژوهش‌گران، استفاده از الگوریتم ژنتیک و مورچگان به منظور بهینه‌یابی این نوع مسائل بوده است. در حالی‌که در سال‌های اخیر، با توجه به نیاز روزافزون تحلیل و طراحی سازه‌های حجیم‌تر و پیچیده‌تر، لزوم ارائه‌ی راهکارهای سریع‌تر و ارزان‌تر احساس می‌شود. با عنایت به این مهم در پژوهش حاضر از الگوریتم دلفین اصلاح شده، برای یافتن ضریب بار فروریزش و همچنین مکانیزم فروریزش سازه‌ی قاب خمشی با اتصال‌های صلب استفاده شده است، تا بتوان عملکرد الگوریتم‌های فراابتکاری جدیدتر را به لحاظ بهبود سرعت همگرایی مطالعه کرد. بدین منظور الگوریتم جدیدی مختص مسائل باینری ارائه شده است، که بر مبنای اعمال اصلاحات بر روی الگوریتم پژواک دلفین است، که توسط کاوه و فرهودی (۲۰۱۳) ارائه شده است.^[۲۳]

۲. تعیین مکانیزم‌های پایه

در تئوری تحلیل خمیری، غالباً قاب به صورت هم‌زمان تحت اثر بار ثقلی در مرکز دهانه و نیروی جانبی در هر طبقه قرار می‌گیرد. به‌عنوان نمونه در شکل ۱، نقاط بحرانی (مفاصل خمیری) برای یک قاب یک طبقه‌ی یک دهانه نشان داده شده است. این نقاط شامل نقاط گره‌یی و همچنین نقاط وسط تیرها در حالت بارگذاری متمرکز هستند. مطابق روش نیل و سایموند،^[۲۴] سه مکانیزم فروریزش ابتدایی در نظر گرفته می‌شود که شامل مکانیزم تیر، طبقه و گره هستند، که برای قاب یک طبقه‌ی یک دهانه به صورت الف، ب و ج در شکل ۲ مشاهده می‌شود.

باید توجه شود که برای یک قاب با m دهانه و n طبقه، تعداد mn مکانیزم تیر و n مکانیزم طبقه وجود دارد. بنابراین تعداد مکانیزم‌های مستقل برابر $n(m+1)$ خواهد بود. تعداد مکانیزم‌های گره‌یی هم برابر با تعداد گره‌های موجود در سازه منتهای گره‌های تکیه‌گاهی است.

را نمی دهد، ولی در مکانیزم های ترکیبی، مکانیزم گره عامل رسیدن به مقدار کمینه ی λ است.

در روش ارائه شده توسط واتود (۱۹۷۹)،^[۲۳] که روش معمول تحلیل حدی است، برای تولید مکانیزم های مستقل، مکانیزم های گره یی نیز محاسبه می شوند. این مسئله موجب افزایش حجم محاسبات می شود. لذا به همین منظور در پژوهش حاضر با توجه به شرایط مکانیزم های گره یی، آن ها بدون انجام محاسبات و به صورت خودکار اعمال شده اند. به این صورت که مفاصل موجود در هر گره در ماتریسی ذخیره می شوند و در هنگام ترکیب دو مکانیزم پایه با یکدیگر، در صورت وجود مفصل مشترک بین گره های آن ها، این مکانیزم گره وارد کار می شود و در غیر این صورت، دیگر در ترکیب اخیر بررسی نمی شود. این مسئله موجب کاهش قابل ملاحظه ی حجم عملیات در بخش حاضر شده است. در ادامه، باید ماتریس سازگاری C تشکیل شود. هر گره مقید نشده، درجه های آزادی انتقالی افقی و عمودی دارد. ماتریس سازگاری بر اساس تغییر طول اعضا در مختصات محلی نوشته می شود و سپس با روی هم گذاشتن آن ها با هم، ماتریس سازگاری در مختصات کلی شکل می گیرد (رابطه ی ۷):

$$C \vec{d} = e \quad (7)$$

همان طور که در شکل ۳ مشاهده می شود، برداری شامل جابه جایی مؤلفه های گره های مقید نشده است و e تغییر طول هر عضو در درجه های آزادی تعریف شده است. با استفاده از شکل ۳ می توان تغییر طول هر المان را به صورت رابطه ی ۸ به دست آورد:

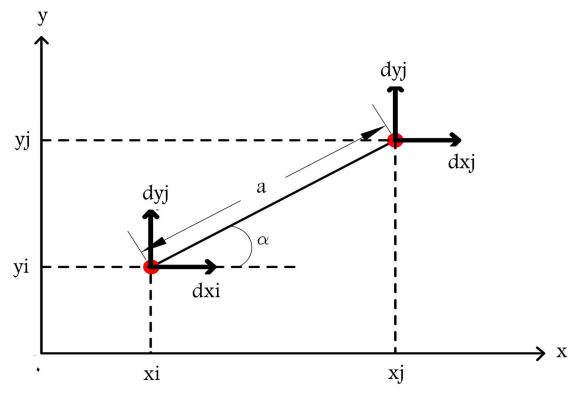
$$e = (d_{xj} - d_{xi}) \cos \alpha + (d_{yj} - d_{yi}) \sin \alpha \quad (8)$$

با نوشتن تغییر طول هر المان در مختصات کلی، ماتریس سازگاری C به دست می آید. به توجه به فرض تحلیل خمیری مبنی بر این که المان ها تغییر طول نمی دهند، معادله ی ۹ باید حل شود:

$$C \vec{d} = 0 \quad (9)$$

برای حل معادله ی ۹، گام اول اعمال حذف گوسی روی ماتریس C است.^[۲۴] پس از اعمال حذف گوسی، با کمی بیش می توان جابه جایی های مستقل و وابسته را از هم جدا کرد و ستون های متناظر با جابه جایی های مستقل را به صورت ماتریس همانی یا همان I نوشت (رابطه ی ۱۰):

$$[I|Ca] \left\{ \frac{d^i}{da} \right\} = \left\{ \begin{matrix} 0 \\ 0 \end{matrix} \right\} \quad (10)$$



شکل ۳. جزئیات یک المان در فضای کلی.

برابر با مجموع استهلاک انرژی داخلی به دلیل دوران هر مفصل خمیری با زاویه ی θ_i است، که لنگر خمیری متناظر M_{pi} است، یا به عبارتی $\sum M_{pi} \theta_i$. ضرب λ هم ضرب بار مورد نظر است (روابط ۱ و ۲):

$$\lambda \cdot W_{external} = W_{internal} \quad (1)$$

$$\sum \lambda W_i \theta_i = \sum M_{pi} \theta_i \quad (2)$$

برای یافتن کار داخلی مکانیزم های مستقل نیز می توان از روابط ۳ الی ۶ استفاده کرد:

$$W_{int} = M_{pb} \cdot |-\theta| + M_{pb} \cdot |\theta| + M_{pb} \cdot |-\theta| = 4M_{pb} \cdot \theta \quad (3)$$

$$W_{int} = \sum_{i=1}^{Nc} \gamma M_{pc} \cdot |\theta|$$

$$= M_{pc} \cdot |-\theta| + M_{pc} \cdot |\theta| + M_{pc} \cdot |-\theta| + M_{pc} \cdot |\theta|$$

$$= 4M_{pc} \cdot \theta \quad (4)$$

$$W_{int} = M_{pc} \cdot |-\theta| + M_{pb} \cdot |\theta|$$

$$W_{int} = \sum_{i=1}^{Nc} \gamma M_{pc} \cdot |\theta| =$$

$$M_{pc} \cdot |-\theta| + M_{pc} \cdot |\theta| + M_{pc} \cdot |-\theta| + M_{pc} \cdot |\theta|$$

$$= 4M_{pc} \cdot \theta \quad (5)$$

$$W_{int} = M_{pc} \cdot |-\theta| + M_{pb} \cdot |-\theta| \quad (6)$$

که در آنها، M_{pb} لنگر خمیری عضو تیری، M_{pc} لنگر خمیری ستون های قاب، Nc هم تعداد ستون های قاب است. لنگر خمیری همواره دوران مفصل را تحمل می کند، بنابراین مقادیر مطابق دوران ها استفاده شده است.

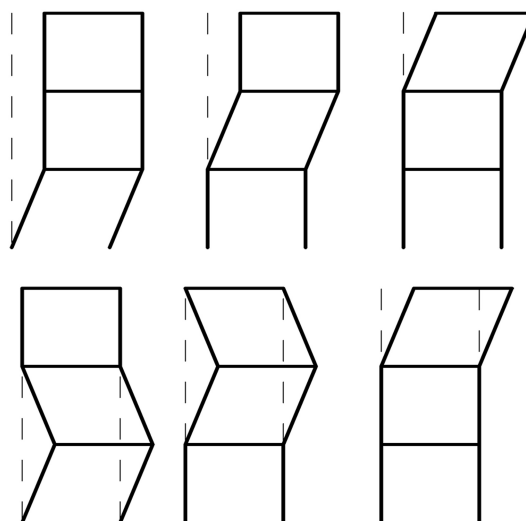
مکانیزم های مستقل به همراه مکانیزم های گره یی، معادلات اولیه برای یافتن جواب نهایی را فراهم می آورند. همان طور که مشاهده می شود، معادله ی کار را برای مکانیزم گره یی نمی توان نوشت. بدون دوران گره ها هیچ گونه حذف مفصلی که با آن بتوان به ضرب بار کمتر در معادله های ترکیبی رسید، وجود نخواهد داشت. ضرب بار فقط زمانی کاهش می یابد که ترکیب دو مکانیزم مستقل منجر به حذف یک مفصل خمیری شود. مطابق روش نیل و سایموند (۲۰۰۴)،^[۹] انتخاب مکانیزم های فروریزش ممکن از یک مکانیزم فروریزش اولیه آغاز می شود و با هدف کاهش دادن ضرب بار با مکانیزم های دیگر ترکیب می شود. زمانی که دیگر هیچ کاهشی در ضرب بار به دست آمده از رابطه ی کار مجازی مشاهده نشد، همین روند برای مکانیزم دیگر تکرار می شود. ضمن اینکه در طبقه ی بام، که فقط دو عضو به هم رسیده اند، مفصل خمیری در عضو با ظرفیت خمیری کمتر تشکیل خواهد شد، که در صورت رعایت ضابطه ی تیر ضعیف و ستون قوی، مفصل مذکور در تیر ایجاد می شود.

همان طور که در شکل ۱ مشاهده می شود، برای مکانیزم های گره و طبقه، مقدار دوران ها یکسان است، ولی جهت دوران آن ها ممکن است متفاوت باشد. در مکانیزم تیر، دوران دو انتها با هم برابر و با علامت یکسان است، ولی دوران مفصل میانی ۲ برابر و در جهت مخالف با دوران انتهای المان است. به این ترتیب کار داخلی مکانیزم گره برابر صفر است. با وجود این که مکانیزم گره به تنهایی تشکیل معادله ی کار

می‌آید (رابطه‌ی ۱۴):

$$M_p^T |r| = \text{کار مجازی داخلی} \quad (14)$$

به دلیل این‌که لنگر خمیری همواره دوران مفصل را تحمل می‌کند، مقدار دوران‌ها همواره مثبت در نظر گرفته می‌شوند. همان‌طور که ذکر شد، در طی شکل‌گیری معادلات اخیر از مکانیزم‌های گره‌یی صرف‌نظر شده است. عاملی که باعث می‌شود ضریب بار کاهش یابد، همین مکانیزم‌های گره‌یی هستند؛ وگرنه به دلیل این‌که در ترکیب دو مکانیزم مستقل با هم با استفاده از رابطه‌ی کار مجازی، با توجه به خطی بودن ترکیب، ضریب باری بین ضریب بار دو مکانیزم به دست آید که نمی‌تواند مکانیزم فروریزش باشد. مکانیزم‌های گره‌یی را می‌توان به صورت خودکار اعمال کرد. بدین منظور موقعیت مفاصل در هنگام محاسبه‌ی کار داخلی باید تعیین شود. به این معنی که هر گره شامل کدام مفاصل است. در هنگام ترکیب دو مکانیزم مستقل، هنگامی که مفاصل مشترک بین آن‌ها وجود داشته باشد، در صورتی که دوران گره‌یی بتواند کار داخلی را کاهش دهد، در ترکیب مکانیزم‌های موردنظر قرار می‌گیرد و در غیر این صورت، دیگر بررسی نخواهد شد. به نوبت هر کدام از مکانیزم‌های گره‌یی در نظر گرفته می‌شوند و حالتی که در آن کمیته‌ی کار داخلی به دست می‌آید، جواب موردنظر خواهد بود.



شکل ۴. وجود مکانیزم‌های فعال مشترک در ترکیب‌های اولیه.

بنابراین d^i یا همان جابه‌جایی‌های مستقل را می‌توان بر حسب d^d یا همان جابه‌جایی‌های وابسته به صورت رابطه‌ی ۱۱ نوشت:

$$d^i = -C^d d^d \quad (11)$$

مکانیزم‌های به دست آمده از روش اخیر در روش‌های برنامه‌ریزی خطی قابل استفاده است، مانند مطالعات انجام شده توسط گریسون^۱ و همکاران (۱۹۷۱).^[۲۵] اما جز در برخی موارد استثنا نمی‌توان در روش ترکیب مکانیزم‌ها از آن‌ها استفاده کرد. علت آن است که همان‌طور که در شکل ۴ مشاهده می‌شود، در مکانیزم‌های مذکور به دلیل وجود مفاصل فعال بسیار زیاد و حذف نشدن مفاصل مشترک سطح انرژی لازم برای هر مکانیزم ترکیبی بسیار بالاتر از مقدار واقعی است. نتیجتاً لازم است مجدداً کدی نوشته شود که مفاصل فعال مشترک در مکانیزم‌ها را حذف کند تا به مکانیزم‌های موردنظر مطابق با تئوری تحلیل خمیری رسید، که انواع این مکانیزم‌ها در ابتدای بخش کنونی بررسی شده است.

۴. ترکیب مکانیزم‌های پایه

با افزایش تعداد دهانه‌ها و تعداد طبقات سازه، تعداد حالت‌های ترکیب مکانیزم‌ها با هم افزایش می‌یابد. به همین جهت ترکیب کردن مکانیزم‌های پایه‌ی ذکر شده با هم به منظور یافتن مکانیزم فروریزش صحیح، امری بسیار خسته‌کننده و وقت‌گیر خواهد بود. قابی که n مکانیزم مستقل دارد، به تعداد $(2^n - n - 1)$ ترکیب ممکن از مکانیزم‌های مستقل آن وجود دارد که البته همه‌ی آن‌ها جزء مکانیزم‌های فروریختگی منطقی به حساب نمی‌آیند. همان‌طور که در پاراگراف قبل هم به آن اشاره شد، ترکیبی قابل قبول است که مکانیزم‌های گره‌یی باعث حذف مفصل و در نتیجه کاهش ضریب بار شود.

مطابق روش نیل و سایموند،^[۲۵] مکانیزم‌های فروریزش ممکن با شروع از یک مکانیزم پایه و سپس ترکیب آن با سایر مکانیزم‌ها انجام می‌شود، تا بتوان در اثر ترکیب آن‌ها، ضریب بار را کاهش داد. وقتی که مکانیزم دیگری وجود نداشت که بتواند ضریب بار را کاهش دهد، همین روند برای یک مکانیزم دیگر تا زمانی که همه‌ی حالت‌های ممکن بررسی شوند انجام می‌شود. کمیته‌ی ضریب بار به دست آمده همان ضریب بار بحرانی خواهد بود. این در حالی است که مسئله را می‌توان به صورت یک مسئله‌ی بهینه‌یابی مطرح کرد و برای حل آن از الگوریتم‌های بهینه‌یابی فرایبتکاری استفاده کرد.

۳. تعیین بارهای فروریزش

بار فروریزش در یک مکانیزم مشخص با استفاده از رابطه‌ی ۱۲ به دست می‌آید. جابه‌جایی‌ها و دوران‌های مکانیزم‌ها به صورت مجازی در نظر گرفته می‌شود و رابطه‌ی کار مجازی داخلی و خارجی به صورت رابطه‌ی ۱۲ نوشته می‌شود.

$$\lambda_c = \frac{\text{کار مجازی داخلی}}{\text{کار مجازی خارجی}} \quad (12)$$

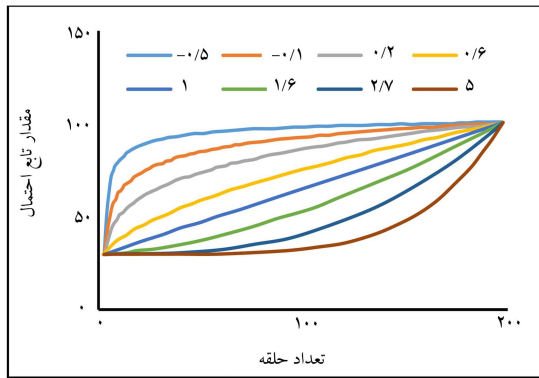
چنان‌که در بخش ۲ گفته شد، کار مجازی خارجی با ضرب نیروهای اعمال شده به گره‌ها (P) در جابه‌جایی متناظر آن گره برای هر مکانیزم مشخص به دست می‌آید (رابطه‌ی ۱۳):

$$\text{کار مجازی خارجی} = P^T d \quad (13)$$

کار مجازی داخلی هم با ضرب دوران‌ها در محل مفصل خمیری (r) در لنگر خمیری اعضایی که در آن‌ها مفصل تشکیل شده است، برای هر مکانیزم به دست

۵. الگوریتم مکان‌یابی پژواکی دلفین باینری

استفاده از الگوریتم‌های فرایبتکاری، نسل جدیدی از روش‌های بهینه‌یابی است که اخیراً برای حل مسائل پیچیده استفاده شده است. به دلیل عملکرد مطلوب و سهولت اجرا، روش‌های بهینه‌یابی به صورت گسترده در مسائل مختلفی استفاده شده‌اند. بهینه‌یابی در سازه‌ها یکی از شاخه‌های کاربرد الگوریتم‌های فرایبتکاری است. در الگوریتم بهینه‌یابی پژواکی دلفین^۲ از راهبرد دلفین‌ها برای شکار استفاده شده است. دلفین‌ها نوعی صدا به نام سونار^۳ را برای مشخص کردن موقعیت هدف خود تولید



شکل ۵. نمودار مرتبط با معادله ۱۵.

کند، باید تعیین شود. در این جا، تغییر CF مطابق منحنی ارائه شده در شکل ۵ و با استفاده از رابطه ۵ در نظر گرفته شده است:

$$PP(\text{Loop}) = PP_1 + (1 - PP_1) \frac{\text{Loop}_i^{\text{power} - 1}}{(\text{LoopNumber})^{\text{power} - 1}} \quad (15)$$

PP احتمال از پیش تعریف شده، PP_1 ضریب همگرایی اولین حلقه که در آن پاسخ‌ها به صورت تصادفی انتخاب شده‌اند، Loop_i تعداد حلقه‌ی فعلی و Power درجه‌ی منحنی است. همان‌طور که مشاهده می‌شود، منحنی معادله‌ی فوق از درجه‌ی Power است.

Loop number تعداد حلقه‌هایی است که در آن الگوریتم باید به نقطه‌ی همگرایی برسد. این عدد توسط کاربر تعیین می‌شود. شکل ۵، تغییرات PP با تغییر Power را با استفاده از رابطه‌ی پیشنهادی ۱۵ نشان می‌دهد.

فلوچارت الگوریتم پژواک دلفین مرتبط با رابطه‌ی ۱۵ در شکل ۶ نشان داده شده است. گام‌های اصلی پژواک دلفین برای بهینه‌یابی گسسته به این صورت است:

- شروع با NL موقعیت برای یک دلفین به صورت تصادفی. این گام شامل ساخت ماتریس $L_{NL \times NV}$ است که در آن NL تعداد موقعیت‌ها و NV تعداد متغیرهاست (و یا بعد هر موقعیت).
- محاسبه‌ی PP مربوط به حلقه با معادله ۱۵.
- محاسبه‌ی برازندگی هر موقعیت. برازندگی در حالتی باید تعریف شود که جواب‌های بهتر مقادیر بیشتری را بگیرند. به بیان دیگر، هدف بهینه‌یابی، باید بیشینه‌سازی برازندگی باشد.

- محاسبه‌ی برازندگی تجمعی مطابق قوانین دلفین.^[۲۳]

- در این مرحله برای متغیر j ($j = \frac{1}{NV}$) احتمال انتخاب i ($i = \frac{1}{AL_j}$) بر اساس رابطه‌ی ۱۶ محاسبه می‌شود.^[۲۳]

$$P_{ij} = \frac{AF_{ij}}{\sum_{i=1}^{LA_j} AF_{ij}} \quad (16)$$

- در این مرحله $P(\text{Loop}_i)$ یعنی احتمال به دست آمده در هر حلقه به بهترین جواب اختصاص داده می‌شود و بقیه‌ی احتمال در بین سایرین توزیع می‌شود (در واقع احتمال به فضای جستجوی الگوریتم دلفین اختصاص می‌یابد).

- با توجه به احتمالات محاسبه شده برای انتخاب‌ها (کل فضای نمونه) مکان‌های مرحله‌ی بعد محاسبه می‌شوند و مجدداً گام‌های قبل تکرار می‌شوند.

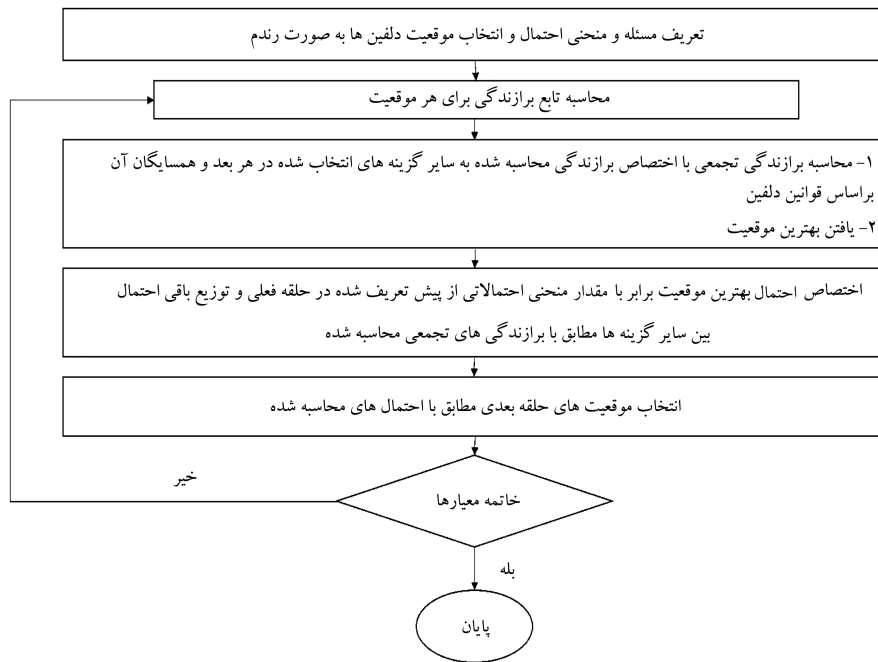
می‌کنند. یک دلفین قادر است صداهایی به شکل کلیک‌ها را تولید کند. بسامد کلیک‌های مذکور بالاتر از صدای استفاده شده برای ارتباط برقرار کردن است و بین گونه‌های مختلف متفاوت است. زمانی که صدا به جسمی برخورد می‌کند، بخشی از انرژی موج صدا به سمت دلفین باز می‌گردد. به محض رسیدن پژواک، دلفین یک کلیک دیگر تولید می‌کند. مدت زمان بین کلیک و پژواک، دلفین را قادر می‌سازد تا فاصله از جسم را محاسبه کند. قدرت در حال تغییر سیگنال، به دلیل این‌که از دو سمت سر دلفین دریافت می‌شود، دلفین را قادر می‌سازد تا جهت را هم به دست آورد. با ادامه‌ی ساعت کردن کلیک و دریافت پژواک به صورت پیوسته به روش پژواک دلفین، دلفین‌ها می‌توانند اجسام را ردیابی کنند.

می‌توان دریافت که پژواک دلفین در برخی جوانب، بسیار شبیه به یک مسئله‌ی بهینه‌یابی است. روند جستجو برای شکار با استفاده از پژواک در دلفین‌ها مشابه با یافتن پاسخ بهینه‌ی یک مسئله است. دلفین‌ها ابتدا تمام فضای جستجو را برای یافتن طعمه بازرسی می‌کنند. به محض این‌که یک دلفین به هدف نزدیک می‌شود، جستجوی خود را محدود می‌کند و به صورت تدریجی کلیک‌های خود را افزایش می‌دهد تا بر موقعیت تمرکز کند. با در نظر گرفتن یک مسئله‌ی بهینه‌یابی، دو فاز قابل بررسی است: ۱. الگوریتم تمام فضای جستجو را بررسی می‌کند تا یک جستجوی کلی انجام شود. این مرحله با بررسی برخی نقاط تصادفی در فضای جستجو انجام می‌شود. ۲. به بررسی جواب‌های بهتر به دست آمده از مرحله‌ی قبل تمرکز می‌شود. این همان مشخصه‌ی اصلی الگوریتم‌های فراابتکاری است.

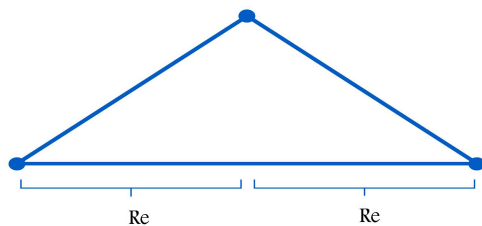
با استفاده از الگوریتم DE، کاربر باید قادر باشد که نسبت جواب‌های تولید شده در فاز ۱ به جواب‌های تولید شده در فاز ۲ را مطابق یک منحنی از پیش تعریف شده تغییر دهد. به بیان دیگر، جستجوی کلی به صورت تدریجی با حالت تعریف شده‌ی کاربر به جستجوی محلی تغییر می‌کند. کاربر یک منحنی تعریف می‌کند که در آن همگرایی بهینه‌یابی باید اعمال شود. سپس الگوریتم پارامترهای خود را طوری تنظیم می‌کند که قادر باشد از منحنی پیروی کند. روش ذکر شده با اختصاص احتمال زیادت، وقوع بهترین جواب در مقایسه با دیگران عمل می‌کند. به بیان دیگر، برای هر متغیر گزینه‌های مختلفی در جامعه‌ی مناسب وجود دارد. در هر حلقه، الگوریتم، احتمال انتخاب بهترین گزینه‌یابی که در آخرین مرحله‌ی به دست آمده را مطابق با منحنی همگرایی کاربر تعریف می‌کند. با استفاده از منحنی همگرایی، ضابطه‌ی همگرایی به الگوریتم دیکته می‌شود و سپس همگرایی الگوریتم وابستگی کمتری به پارامتر دارد. منحنی همگرایی می‌تواند هر منحنی که با شیب ملایم صعودی می‌شود، باشد.

قبل از شروع بهینه‌یابی، فضای جستجو باید با استفاده از این قانون تقسیم بندی شود:

- ترتیب فضای جستجو: برای هر متغیری که قرار است در طی این فرایند بهینه شود، مقادیر نهایی فضای جستجو به صورت صعودی و یا نزولی مرتب می‌شوند و اگر آنها شامل بیش از یک مشخصه باشند، ترتیب‌بندی بر اساس مهمترین آنها انجام می‌شود. با استفاده از این روش، برای متغیر j ، بردار A_j به طول LA_j ساخته می‌شود که شامل همه‌ی مقادیر نهایی ممکن برای ژأمین متغیر است. با قرار دادن بردارهای مذکور کنار هم به عنوان ستون‌های ماتریس، ماتریس $Matrix Alternatives_{MA \times NV}$ ساخته می‌شود که در آن MA برابر با $\max(LA_j)_{j=1:NV}$ و NV برابر تعداد متغیرهاست. به علاوه، یک منحنی مطابق با این موضوع که ضریب همگرایی CF در طی فرایند بهینه‌یابی باید تغییر



شکل ۶. فلوچارت الگوریتم دلفین.



شکل ۷. مثلث احتمال در الگوریتم دلفین.

طرفی، اگر از حد معقولی هم بیشتر انتخاب شود، به دلیل همگرایی سریع‌تر آن، امکان گرفتاری در بهینه‌ی محلی و عدم در نظر گرفتن برخی قسمت‌های فضای جستجو وجود دارد. معمولاً در بیشتر مسائل این احتمال ۰/۱ یا ۱۰ درصد فرض می‌شود. علی‌رغم قابلیت‌های ذکر شده در الگوریتم پژوهاک دلفین و قدرت بالای آن در حل مسائلی که تاکنون در پژوهش‌های مختلفی بالاخص در یافتن مقاطع بهینه و جانمایی بهینه‌ی سیستم‌های باربر جانبی داشته است، حل مسائلی که به صورت باینری هستند، به کمک الگوریتم پژوهاک دلفین قدرتمند، کمتر مورد توجه پژوهش‌گران بوده است. لذا در پژوهش حاضر با ایجاد اصلاحاتی در الگوریتم پژوهاک دلفین، الگوریتم اخیر برای مسائل باینری گسترش یافته و سپس برای مسئله‌ی تحلیل خمیری قاب‌های خمشی، که کد آن در نرم‌افزار متلب به صورت باینری نوشته شده و در بخش‌های قبل توضیحاتی در باره‌ی آن ارائه شده، استفاده شده است. لکن الگوریتم اصلاح شده‌ی پژوهاک دلفین برای هر نوع مسئله باینری، که تحلیل خمیری هم جزئی از آن است، قابل استفاده خواهد بود.

در ابتدا مثلث احتمال که به عنوان پیشنهاد در الگوریتم دلفین مطرح شده است، حذف می‌شود. چرا که زمانی که فقط دو گزینه‌ی ۰ و ۱ در فضای جستجو برای هر متغیر وجود دارد، استفاده از مثلث همسایگی که برازندگی را در همسایگی هر گزینه توزیع می‌کند، در عمل بی‌معناست (شکل ۷).

در مرحله‌ی بعد تغییراتی در تعریف (PP_1) صورت گرفته است. همان‌طور که

پارامترهای این الگوریتم شامل این موارد هستند:

الف) تعداد حلقه: برای هر مسئله‌ی بهینه‌یابی، هزینه‌ی محاسباتی با افزایش تعداد حلقه‌ها بیشتر می‌شود. تعداد حلقه‌ها زمانی که به دقت بالا نیاز است، با آنالیز حساسیت می‌تواند تعیین شود.

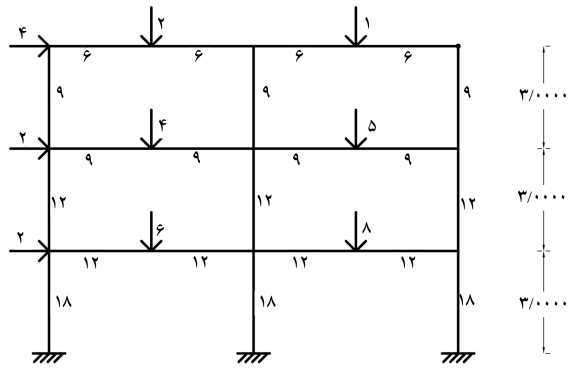
ب) رابطه‌ی منحنی همگرایی: توصیه شده است که با یک منحنی خطی کار را آغاز کرد و سپس از منحنی‌هایی که زمان بیشتری (تعداد حلقه‌های بیشتر) را صرف می‌کنند، استفاده کرد. یعنی ابتدا با $Power = 1$ کار را آغاز کرد که معمولاً هم نتایج خوبی به دست می‌دهد و سپس حالت‌های $Power < 1$ را امتحان کرد تا بهبود نتایج را با آن کنترل کرد.

ج) شعاع مؤثر (Re): پارامتر شعاع مؤثر بهتر است که بر اساس اندازه‌ی فضای جستجو انتخاب شود. معمولاً توصیه شده است که کمتر از $1/4$ فضای جستجو انتخاب شود.

د) تعداد موقعیت‌ها (NL): پارامتر تعداد موقعیت‌ها همان اندازه‌ی جمعیت در الگوریتم ژنتیک است، که باید به صورت منطقی انتخاب شود.

ه) ϵ : پارامتر ϵ در اعمال متغیرهای تصادفی به کار می‌آید و بهتر است کمتر از کوچک‌ترین برازندگی ممکن انتخاب شود.

بزرگ‌ترین مزیت الگوریتم‌های فراابتکاری این است که وابستگی کمتری به پارامترها و متغیرها دارند و در نتیجه کاربرد آن‌ها ساده‌تر است و در عین حال مشکل به دام افتادن در بهینه‌ی محلی را کاهش داده‌اند. الگوریتم پژوهاک دلفین، علاوه بر داشتن خصوصیت ذکر شده، ویژگی دیگری دارد که آن را از سایر الگوریتم‌های هم رده‌ی خود نیز متمایز می‌کند و آن هم اختصاص احتمال از پیش تعریف شده به بهترین جواب با تعریف $P(Loop)$ در هر مرحله است، که مقدار آن در هر حلقه متفاوت است. وجود خاصیت اخیر در الگوریتم دلفین باعث می‌شود که همگرایی به صورت کنترل شده انجام شود و در عین حال جواب با احتمال خیلی کمی، گرفتار پاسخ‌های بهینه‌ی محلی شود. احتمال حلقه‌ی نخست هر چه بزرگ‌تر باشد، فضای جستجو کوچک‌تر خواهد شد و الگوریتم به جواب نهایی نزدیک‌تر خواهد شد. از



شکل ۹. قاب نمونه‌ی یک.

```

for i=1 to NV
    k=BL(i)
    l=(1-PP)*m
    for j=1:NL
        if k=0
            P(2,i)=(1-PP)(1-l)+0.5l
            P(1,i)=1-P(2,i)
        else k=1
            P(1,i)=(1-PP)(1-l)+0.5
            P(2,1)=1-P(1,i)
        end
    end
end
    
```

شکل ۸. شبه کد تغییرات الگوریتم دلفین.

جهانشاهی (۲۰۰۴)^[۹] بررسی شده‌اند، در بخش حاضر ارائه شده است. نتایج حاصل از تحلیل اشاره شده در پژوهش حاضر با نتایج موجود مقایسه شده است. توجه شود که قاب‌های ذکر شده، قاب‌های با مقاطع و بارگذاری فرضی هستند که جواب دقیق تحلیل خمیری آن‌ها موجود هستند و صحت‌سنجی کد تحلیل خمیری و همچنین عملکرد الگوریتم جدید ارائه شده با مثال‌های ارائه شده انجام شده است. به منظور بررسی نحوه‌ی عملکرد الگوریتم دلفین باینری، تمامی مثال‌ها با استفاده از الگوریتم گرگ خاکستری (GWO)^[۲۶] و الگوریتم وال (WOA)^[۲۷] نیز تحلیل شده است (توضیحات مختصری از الگوریتم‌های اشاره شده نیز در پیوست ارائه شده است).

در هر مثال، علاوه بر هندسه و بارگذاری، ظرفیت خمیری اعضا بر روی هر شکل نشان داده شده است. به دلیل این‌که از هر واحدی می‌توان برای حل استفاده کرد، هیچ واحد خاصی برای مثال‌های موردنظر در نظر گرفته نشده است. به منظور مقایسه‌ی عملکرد الگوریتم‌های اشاره شده در هر مثال، از مقادیر RMSE^۶ (رابطه‌ی ۱۷) و MAE^۷ (رابطه‌ی ۱۸) استفاده شده است. این مقادیر به صورت آماری در برخی نوشتارها به منظور مقایسه‌ی عملکرد الگوریتم‌ها استفاده شده است.^[۲۸،۲۹]

$$RMSE = \sqrt{\frac{\sum_{i=1}^n (h_i^t - t_i^t)^2}{n}} \quad (20)$$

$$MAE = \frac{1}{n} \sum_{i=1}^n |h_i - t_i| \quad (21)$$

که در آنها، h_i و t_i به ترتیب مقدار واقعی و مقدار محاسبه شده هستند و n تعداد دفعات آزمایش است.

۱.۶. نمونه‌ی یک

نمونه‌ی یک، یک قاب ۲ دهانه در ۳ طبقه است (شکل ۹). بارگذاری و ظرفیت لنگر خمیری اعضا هم در شکل ۹ نشان داده شده است. در جدول ۱، نحوه‌ی نام‌گذاری مکانیزم‌های مستقل در کد نوشته شده به همراه ضریب بار فروریزش آن‌ها ارائه شده است.

در تنظیمات الگوریتم دلفین، تعداد موقعیت‌ها (NL) برابر ۱۰ با تعداد حلقه‌ی ۵۰ انتخاب شده است. همان‌طور که در بخش قبل توضیح داده شد، کد الگوریتم دلفین به صورت باینری بهینه‌یابی شده است. به همین دلیل تغییراتی روی کد الگوریتم دلفین انجام شد، تا بتوان برای نخستین بار از الگوریتم دلفین برای بهینه‌یابی به صورت باینری استفاده کرد.

گفته شد، احتمال پاسخ حلقه‌ی اول در غالب مسائل ۱۰٪ فرض می‌شود. این در حالی است که در حالت باینری، این موضوع مشکلاتی را در همگرایی ایجاد می‌کند. دلیل آن هم این است که در الگوریتم مکان‌یابی پژواکی دلفین، برای هر متغیر، PP_1 احتمال اختصاص داده شده به بهترین گزینه در گام اول است و $(1 - PP_1)$ احتمال اختصاص داده شده به سایر گزینه‌هاست. از آنجایی که در فضای باینری، ۲ گزینه بیشتر وجود ندارد، (PP_1) احتمال مربوط به گزینه‌ی برتر است و $(1 - PP_1)$ احتمال مربوط به گزینه‌ی دیگر و از آنجایی که احتمال گزینه‌ی برتر باید بزرگ‌تر باشد، (PP_1) باید بزرگ‌تر از ۰.۵٪ باشد. پس الگوریتم با احتمالی بالای ۰.۵٪ پاسخ مناسب را نگه می‌دارد.

در ادامه، به منظور جلوگیری از همگرایی زودرس، پارامتر (ϵ) که وظیفه‌ی اضافه کردن پاسخ‌های تصادفی به الگوریتم را دارد، به صورت رابطه‌ی ۱۷ تعریف می‌شود:

$$\epsilon = (1 - PP_1) * \text{Random} \quad (17)$$

که در آن $(Random)$ نسبتی است که تعداد پاسخ‌های تصادفی متناسب با آن تغییر می‌کند و (ϵ) درصد پاسخ‌های تصادفی در هر گام است، که به تدریج به سمت صفر میل می‌کند.

با توجه به تعاریف انجام شده، برای هر متغیر، در هر گام، احتمال مربوط به گزینه‌ی برتر از رابطه‌ی ۱۸ محاسبه می‌شود:

$$(PP(\text{Loop}i)) * (1 - \epsilon) + 0.5 \epsilon \quad (18)$$

در نتیجه، احتمال مربوط به گزینه‌ی دیگر از رابطه‌ی ۱۹ به دست می‌آید:

$$(1 - PP(\text{Loop}i)) * (1 - \epsilon) + 0.5 \epsilon \quad (19)$$

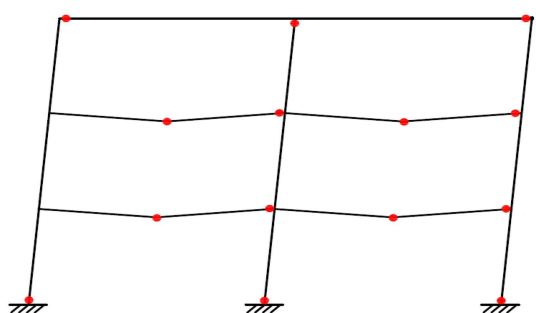
به منظور جمع‌بندی، شبه کد ارائه شده در شکل ۸ می‌تواند مؤثر باشد. که در آن، NV تعداد متغیرهای مسئله، NL تعداد موقعیت‌ها، BL بهترین موقعیت با توجه به برانزنگی و m هم درصدی است که بیان‌گر تعداد پاسخ‌های تصادفی انتخابی در هر گام است.

۶. صحت‌سنجی روش پیشنهادی

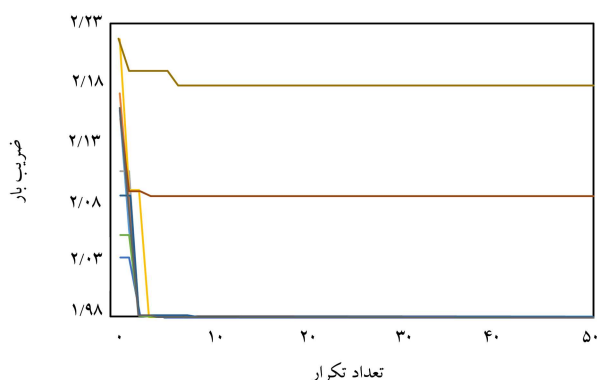
به منظور اطمینان از صحت عملکرد روش تحلیل حدی مبتنی بر تئوری کینماتیکی و الگوریتم بهینه‌سازی دلفین اصلاح شده، که به منظور ترکیب مکانیزم‌های پایه در روش پیشنهادی استفاده شده است، چند نمونه از قاب‌هایی که توسط کاوه و

جدول ۲. جزئیات شکل ۹.

شماره‌ی تحلیل	تعداد تکرار	مود فروریزش	ضریب بار
۱	۱۴	{۱۱۱۱۱۱۰۰}	۱٫۹۷۵۶
۲	۶	{۱۱۱۱۱۱۰۰}	۱٫۹۷۵۶
۳	۱۰	{۱۱۱۱۱۱۰۰}	۱٫۹۷۵۶
۴	۵	{۱۱۱۱۱۱۰۰}	۱٫۹۷۵۶
۵	۱۵	{۱۱۱۱۱۱۰۰}	۱٫۹۷۵۶
۶	۱۱	{۱۱۱۱۱۱۰۰}	۱٫۹۷۵۶
۷	۸	{۱۱۱۱۱۱۰۰}	۱٫۹۷۵۶
۸	۱۲	{۱۱۱۱۱۱۰۰}	۱٫۹۷۵۶
۹	۳	{۱۱۱۱۱۱۰۰}	۱٫۹۷۵۶
۱۰	۱۳	{۱۱۱۱۱۱۰۰}	۱٫۹۷۵۶



شکل ۱۱. مود خرابی نمونه‌ی یک.



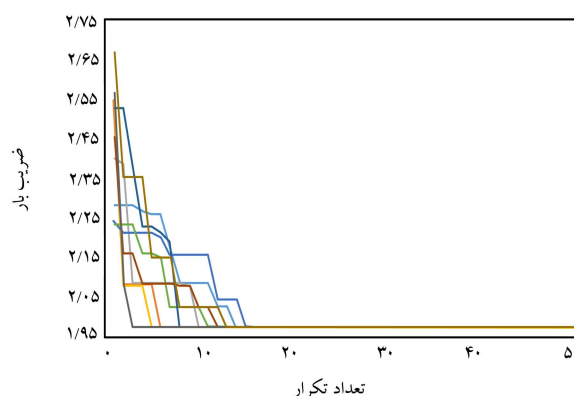
شکل ۱۲. نتایج تحلیل نمونه‌ی یک با الگوریتم گرگ خاکستری.

با الگوریتم‌های گرگ خاکستری و وال هم انجام شد، که نمودار همگرایی ۱۰ مرتبه تحلیل برای هر کدام از آنها، به ترتیب در شکل‌های ۱۲ و ۱۳ و جزئیات هر کدام در جدول‌های ۳ و ۴ ارائه شده است.

نتایج حاصل از تحلیل در سه الگوریتم دلفین باینری، گرگ خاکستری و وال نشان می‌دهد که الگوریتم دلفین باینری که در نوشتار حاضر ارائه شده است، در هر ۱۰ مرتبه تحلیل پاسخ دقیق را نتیجه داد. در حالی که تحلیل‌های ۸ و ۱۰ مرتبه الگوریتم گرگ خاکستری، با وجود این‌که نتایجی نزدیک به ضریب بار فروریزش بحرانی داشت، مود فروریزش متفاوتی را ارائه کرد. در الگوریتم وال هم چهار تحلیل شامل تحلیل‌های ۵، ۶، ۸ و ۱۰ جواب دقیق نبودند و فقط تحلیل ۸ نتیجه‌ی بسیار نزدیک به پاسخ واقعی سازه داشت. به طور خلاصه و البته تقریبی می‌توان گفت در مثال خاص نمونه‌ی یک، الگوریتم دلفین باینری با خطای صفر، الگوریتم گرگ

جدول ۱. نام‌گذاری مکانیزم‌های مستقل نمونه‌ی یک.

شماره‌ی مکانیزم	موقعیت مکانیزم	ضریب بار
۱	طبقه‌ی اول	۴٫۵
۲	تیر طبقه‌ی اول دهانه‌ی چپ	۲٫۶۷
۳	تیر طبقه‌ی اول دهانه‌ی راست	۲
۴	طبقه‌ی دوم	۴
۵	تیر طبقه‌ی دوم دهانه‌ی چپ	۳
۶	تیر طبقه‌ی دوم دهانه‌ی راست	۲٫۴
۷	طبقه‌ی سوم	۴٫۵
۸	تیر طبقه‌ی سوم دهانه‌ی چپ	۴
۹	تیر طبقه‌ی سوم دهانه‌ی راست	۸

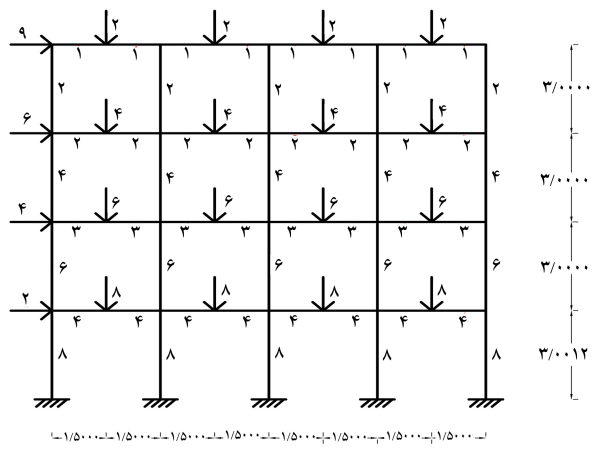


شکل ۱۰. نتایج تحلیل نمونه‌ی یک با الگوریتم دلفین باینری.

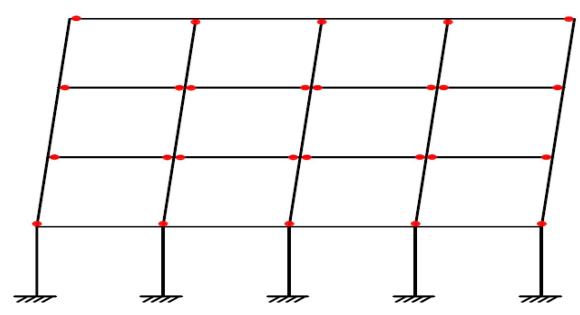
ابتدا با استفاده از الگوریتم دلفین اصلاح شده، بهینه‌یابی انجام شد. برای این‌که بتوان عملکرد الگوریتم دلفین اصلاح شده را در مواقعی که الگوریتم داخل طعمه می‌افتد، ارزیابی کرد، برنامه ۱۰ بار تحلیل شد و هر مرتبه خروجی‌های الگوریتم استخراج شدند تا تعداد خطاها ارزیابی شوند. نمودار همگرایی ۱۰ مرتبه تحلیل ذکر شده در شکل ۱۰ مشاهده می‌شود و جزئیات آن نیز در جدول ۲ ارائه شده است، که برای هر تحلیل انجام شده، تعداد تکراری که الگوریتم پس از آن به عددی همگرا شده است، در ستون دوم جدول ۲ مشاهده می‌شود. همچنین مود فروریزشی، که الگوریتم متناظر با ضریب بار محاسبه شده به دست می‌دهد، نیز در ستون سوم ارائه شده است. توجه شود که این اعداد به صورت باینری یا ۰ و ۱ هستند و هر عدد بیان‌گر یک مکانیزم پایه است. ۰ یا ۱ بودن اعداد ذکر شده نشان می‌دهد که آیا مکانیزم مذکور با شماره‌ی بیان شده در جدول ۱، در مود فروریزش حضور دارد یا خیر. در ستون چهارم هم ضریب بار فروریزش مکانیزم‌های پایه‌ی اشاره شده محاسبه شده است، که برای محاسبه‌ی آن از تئوری کار مجازی، که روابط آن در فصل دوم اشاره شده، استفاده شده است.

ضریب بار فروریزش بحرانی واقعی قاب نمونه‌ی یک دقیقاً برابر ۱٫۹۷۵۶ است.^[۹] نکته‌ی جالب این است که الگوریتم اصلاح شده برای مثال حاضر که تعداد مکانیزم‌های پایه‌ی آن نسبت به سازه‌های بلندمرتبه کمتر است، در هر ۱۰ مرتبه تحلیل، دقیقاً همین عدد را نتیجه داد. مکانیزم فروریزش نهایی قاب در شکل ۱۱ مشاهده می‌شود.

حال به جهت بررسی بهبود پاسخ‌ها با الگوریتم دلفین اصلاح شده، همین مثال



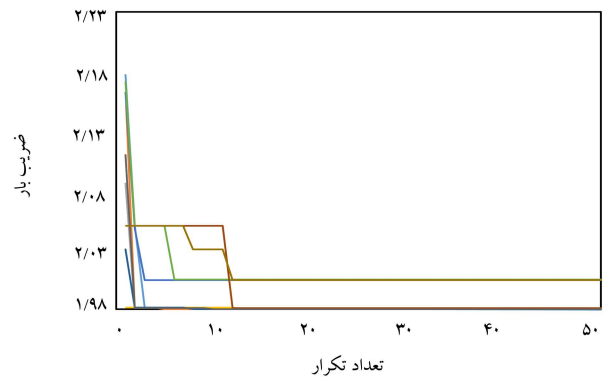
شکل ۱۴. نمونه‌ی دوم.



شکل ۱۵. مود خرابی نمونه‌ی دوم.

جدول ۵. نحوه‌ی نام‌گذاری مکانیزم‌های مستقل نمونه‌ی دو.

ضریب بار	موقعیت مکانیزم	شماره‌ی مکانیزم
۱/۳۳	طبقه‌ی اول	۱
۱	تیر طبقه‌ی اول دهانه‌ی اول	۲
۱	تیر طبقه‌ی اول دهانه‌ی دوم	۳
۱	تیر طبقه‌ی اول دهانه‌ی سوم	۴
۱	تیر طبقه‌ی اول دهانه‌ی چهارم	۵
۱/۱۱	طبقه‌ی دوم	۶
۱	تیر طبقه‌ی دوم دهانه‌ی اول	۷
۱	تیر طبقه‌ی دوم دهانه‌ی دوم	۸
۱	تیر طبقه‌ی دوم دهانه‌ی سوم	۹
۱	تیر طبقه‌ی دوم دهانه‌ی چهارم	۱۰
۰/۹۵	طبقه‌ی سوم	۱۱
۱	تیر طبقه‌ی سوم دهانه‌ی اول	۱۲
۱	تیر طبقه‌ی سوم دهانه‌ی دوم	۱۳
۱	تیر طبقه‌ی سوم دهانه‌ی سوم	۱۴
۱	تیر طبقه‌ی سوم دهانه‌ی چهارم	۱۵
۰/۸۳	طبقه‌ی چهارم	۱۶
۱	تیر طبقه‌ی چهارم دهانه‌ی اول	۱۷
۱	تیر طبقه‌ی چهارم دهانه‌ی دوم	۱۸
۱	تیر طبقه‌ی چهارم دهانه‌ی سوم	۱۹
۱	تیر طبقه‌ی چهارم دهانه‌ی اول	۲۰



شکل ۱۳. نتایج تحلیل نمونه‌ی یک با الگوریتم وال.

جدول ۳. جزئیات شکل ۱۰.

شماره‌ی تحلیل	تعداد تکرار	مود فروریزش	ضریب بار
۱	۳	{۱۱۱۱۱۱۰۰}	۱/۹۷۵۶
۲	۶	{۱۱۱۱۱۱۰۰}	۱/۹۷۵۶
۳	۸	{۱۱۱۱۱۱۰۰}	۱/۹۷۵۶
۴	۴	{۱۱۱۱۱۱۰۰}	۱/۹۷۵۶
۵	۶	{۱۱۱۱۱۱۰۰}	۱/۹۷۵۶
۶	۵	{۱۱۱۱۱۱۰۰}	۱/۹۷۵۶
۷	۹	{۱۱۱۱۱۱۰۰}	۱/۹۷۵۶
۸	۴	{۱۱۱۱۱۰۱۱۰}	۲/۰۷۸۹
۹	۵	{۱۱۱۱۱۱۰۰}	۱/۹۷۵۶
۱۰	۷	{۰۰۱۱۱۱۱۰}	۲/۱۷۲۴

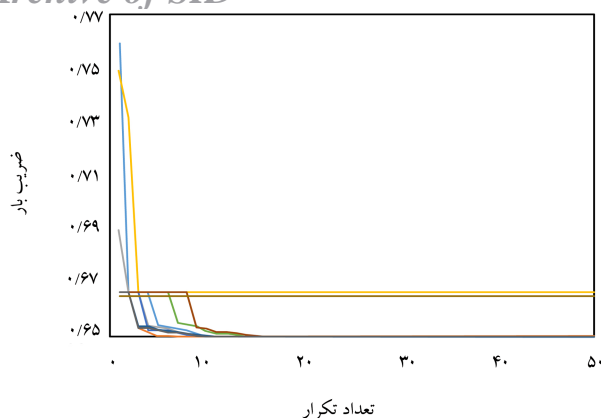
جدول ۴. جزئیات شکل ۱۱.

شماره‌ی تحلیل	تعداد تکرار	مود فروریزش	ضریب بار
۱	۱۳	{۱۱۱۱۱۱۰۰}	۱/۹۷۵۶
۲	۵	{۱۱۱۱۱۱۰۰}	۱/۹۷۵۶
۳	۱۲	{۱۱۱۱۱۱۰۰}	۱/۹۷۵۶
۴	۱۳	{۱۱۱۱۱۱۰۰}	۱/۹۷۵۶
۵	۳	{۰۰۱۰۰۰۰۰۰}	۲/۰۰۰۰
۶	۶	{۰۰۱۰۰۰۰۰۰}	۲/۰۰۰۰
۷	۸	{۱۱۱۱۱۱۰۰}	۱/۹۷۵۶
۸	۱۲	{۱۱۱۱۱۱۱۰}	۱/۹۷۶۷
۹	۱۰	{۱۱۱۱۱۱۰۰}	۱/۹۷۵۶
۱۰	۱۲	{۰۰۱۰۰۰۰۰}	۲/۰۰۰۰

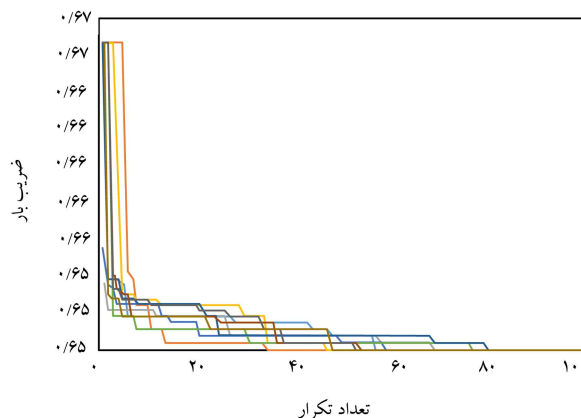
خاکستری با خطای ۲۰٪ و الگوریتم وال با خطای ۴۰٪ قادر به ارائه‌ی پاسخ دقیق فروریزی قاب هستند. در مثال دوم، قاب بزرگ‌تری در نظر گرفته شده است تا افزایش تعداد مکانیزم‌های مستقل در مقدار خطای الگوریتم‌ها مجدداً بررسی شود.

۲.۶. نمونه‌ی دوم

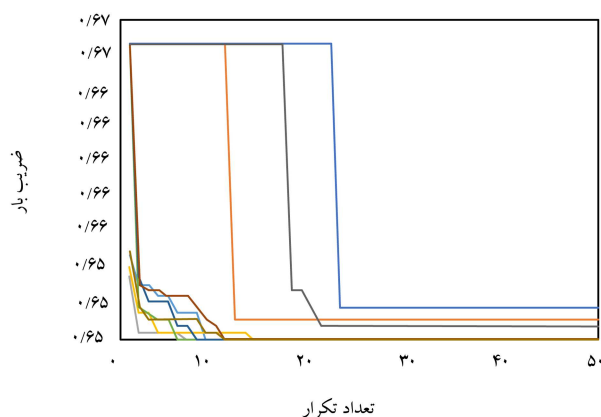
نمونه‌ی دوم، یک قاب با ۴ طبقه و ۴ دهانه است (شکل ۱۴)، که ضریب بار واقعی آن ۰/۶۵ است، و مود فروریزش متناظر آن در شکل ۱۵ مشاهده می‌شود.^[۹] در جدول ۵، نحوه‌ی شماره‌گذاری مودهای پایه ارائه شده است. قاب نمونه‌ی



شکل ۱۷. نتایج تحلیل نمونه‌ی دوم با استفاده از الگوریتم گرگ خاکستری.



شکل ۱۶. نتایج تحلیل نمونه‌ی دوم با استفاده از الگوریتم دلفین باینری.



شکل ۱۸. نتایج تحلیل نمونه‌ی دوم با استفاده از الگوریتم وال.

دوم، ۲۰ مکانیزم پایه دارد، که ۱۶ مکانیزم پایه از نوع تیر و ۴ مکانیزم پایه هم از نوع طبقه هستند. همان‌طور که گفته شد، ضریب بار فروریزش واقعی قاب برابر ۰/۶۵ است و متناظر با مود فروریزش شکل ۱۵ است، که در آن مکانیزم‌های ۶، ۱۱ و ۱۶ به ترتیب مکانیزم‌های طبقات ۲، ۳ و ۴ هستند که در ترکیب شرکت دارند و سایر مکانیزم‌های پایه، که کد باینری آنها صفر است و شامل مکانیزم همگی تیرها و مکانیزم طبقه‌ی اول هستند، در مکانیزم بحرانی حضور ندارند.

برای پارامترهای الگوریتم دلفین (اصلاح شده و اصلاح نشده) در مثال دو، تعداد موقعیت‌ها (NL) برابر با ۵۰ و تعداد حلقه‌ها هم برابر با ۱۵۰ انتخاب شدند. مطابق روند مثال یک، ابتدا با استفاده از الگوریتم دلفین باینری، ۱۰ مرتبه تحلیل انجام شد و در هر مرتبه، نمودار همگرایی استخراج شد، که نتایج آن در شکل ۱۶ و جزئیات مربوط به آن هم در جدول ۶ ارائه شده است. مطابق مثال یک، به منظور بررسی دقت الگوریتم ذکر شده، نتایج حاصل از الگوریتم‌های گرگ خاکستری و وال به ترتیب در شکل‌های ۱۷ و ۱۸ به همراه جزئیات آنها در جدول‌های ۷ و ۸ ارائه شده است.

همان‌طور که از نمودارهای همگرایی مثال کنونی در الگوریتم دلفین باینری مشخص است، الگوریتم دلفین باینری جواب دقیق را بدون خطا و بدون گرفتاری در طعمه در هر ۱۰ مرتبه تحلیل نتیجه داده است. تعداد تکرار این الگوریتم‌ها در ستون دوم جدول ۵ ارائه شده است. همان‌طور که مشاهده می‌شود، تعداد تکرارها برای همگرایی در مثال دوم نسبت به مثال یک، که قاب کوچک‌تری بود، بسیار بیشتر است. در نتیجه، زمان تحلیل در نرم‌افزار متلب نسبت به مثال قبلی بیشتر شده

است، ولی دقت الگوریتم همچنان حفظ شد و در هیچ طعمه‌یی گرفتار نشد. همان‌طور که از جداول اخیر مشخص است، در مثال دوم هم در الگوریتم دلفین باینری هر ۱۰ مرتبه تحلیل پاسخ دقیقی به دست آمد و می‌توان گفت تقریباً با خطایی نزدیک به صفر در یک مثال با تعداد مکانیزم‌های بالا به جواب رسیده است. این در حالی است که در الگوریتم گرگ خاکستری مجدداً خطاهایی گاه به گاه مشاهده می‌شود، و فقط دو تحلیل ۴ و ۱۰ پاسخ‌هایی ارائه دادند که اگرچه ضریب بار فروریزش نزدیک به ضریب بار فروریزش واقعی را نتیجه دادند، اما مکانیزم‌های فروریزش تفاوت‌هایی دارد که باید بیشتر به آنها توجه شود. در الگوریتم وال هم ۳

جدول ۶. جزئیات شکل ۱۵.

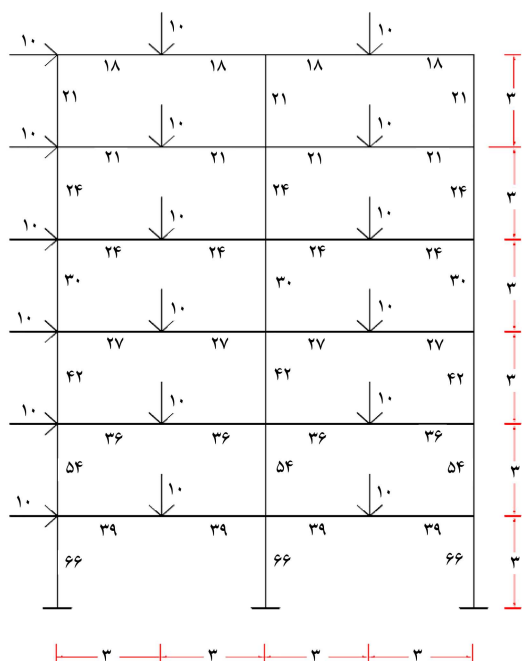
ضریب بار	مود فروریزش	تعداد تکرار	شماره‌ی تحلیل
۰/۶۵	{۰۰۰۰۰۰ ۱۰۰۰۰۰ ۱۰۰۰۰۰ ۱۰۰۰۰۰}	۵۷	۱
۰/۶۵	{۰۰۰۰۰۰ ۱۰۰۰۰۰ ۱۰۰۰۰۰ ۱۰۰۰۰۰}	۳۵	۲
۰/۶۵	{۰۰۰۰۰۰ ۱۰۰۰۰۰ ۱۰۰۰۰۰ ۱۰۰۰۰۰}	۵۸	۳
۰/۶۵	{۰۰۰۰۰۰ ۱۰۰۰۰۰ ۱۰۰۰۰۰ ۱۰۰۰۰۰}	۳۵	۴
۰/۶۵	{۰۰۰۰۰۰ ۱۰۰۰۰۰ ۱۰۰۰۰۰ ۱۰۰۰۰۰}	۵۹	۵
۰/۶۵	{۰۰۰۰۰۰ ۱۰۰۰۰۰ ۱۰۰۰۰۰ ۱۰۰۰۰۰}	۳۱	۶
۰/۶۵	{۰۰۰۰۰۰ ۱۰۰۰۰۰ ۱۰۰۰۰۰ ۱۰۰۰۰۰}	۶۹	۷
۰/۶۵	{۰۰۰۰۰۰ ۱۰۰۰۰۰ ۱۰۰۰۰۰ ۱۰۰۰۰۰}	۳۷	۸
۰/۶۵	{۰۰۰۰۰۰ ۱۰۰۰۰۰ ۱۰۰۰۰۰ ۱۰۰۰۰۰}	۳۸	۹
۰/۶۵	{۰۰۰۰۰۰ ۱۰۰۰۰۰ ۱۰۰۰۰۰ ۱۰۰۰۰۰}	۴۸	۱۰

جدول ۷. جزئیات شکل ۱۶.

ضریب بار	مود فروریزش	تعداد تکرار	شماره‌ی تحلیل
۰٫۶۵۰۰	{۰۰۰۰۰۰ ۱۰۰۰۰۰ ۱۰۰۰۰۰ ۱۰۰۰۰۰}	۱۱	۱
۰٫۶۵۰۰	{۰۰۰۰۰۰ ۱۰۰۰۰۰ ۱۰۰۰۰۰ ۱۰۰۰۰۰}	۷	۲
۰٫۶۵۰۰	{۰۰۰۰۰۰ ۱۰۰۰۰۰ ۱۰۰۰۰۰ ۱۰۰۰۰۰}	۱۰	۳
۰٫۶۶۶۷	{۱۰۰۰۰۰ ۱۰۰۰ ۱۱۱۰ ۱۰۰ ۱۰۰۰۰۰}	۳	۴
۰٫۶۵۰۰	{۰۰۰۰۰۰ ۱۰۰۰۰۰ ۱۰۰۰۰۰ ۱۰۰۰۰۰}	۱۰	۵
۰٫۶۵۰۰	{۰۰۰۰۰۰ ۱۰۰۰۰۰ ۱۰۰۰۰۰ ۱۰۰۰۰۰}	۱۴	۶
۰٫۶۵۰۰	{۰۰۰۰۰۰ ۱۰۰۰۰۰ ۱۰۰۰۰۰ ۱۰۰۰۰۰}	۱۱	۷
۰٫۶۵۰۰	{۰۰۰۰۰۰ ۱۰۰۰۰۰ ۱۰۰۰۰۰ ۱۰۰۰۰۰}	۱۶	۸
۰٫۶۵۰۰	{۰۰۰۰۰۰ ۱۰۰۰۰۰ ۱۰۰۰۰۰ ۱۰۰۰۰۰}	۱۰	۹
۰٫۶۶۶۷	{۱۱۰۰۰۰ ۱۱۱۰۰۰ ۱۱۰ ۱۱۱۰ ۱۰ ۱}	۱	۱۰

جدول ۸. جزئیات شکل ۱۷.

ضریب بار	مود فروریزش	تعداد تکرار	شماره‌ی تحلیل
۰٫۶۵۰۰	{۰۰۰۰۰۰ ۱۰۰۰۰۰ ۱۰۰۰۰۰ ۱۰۰۰۰۰}	۹	۱
۰٫۶۵۱۲	{۰۰۰۰۰۰ ۱۰۰۰ ۱۱۰۰۰۰ ۱۰۰۰۰۰}	۱۲	۲
۰٫۶۵۰۰	{۰۰۰۰۰۰ ۱۰۰۰۰۰ ۱۰۰۰۰۰ ۱۰۰۰۰۰}	۷	۳
۰٫۶۵۰۰	{۰۰۰۰۰۰ ۱۰۰۰۰۰ ۱۰۰۰۰۰ ۱۰۰۰۰۰}	۱۴	۴
۰٫۶۵۱۹	{۰۰۰۰۰۰ ۱۱۰۰۰ ۱۰۰۰۰۰ ۱۰ ۱۰ ۱}	۲۳	۵
۰٫۶۵۰۰	{۰۰۰۰۰۰ ۱۰۰۰۰۰ ۱۰۰۰۰۰ ۱۰۰۰۰۰}	۶	۶
۰٫۶۵۰۰	{۰۰۰۰۰۰ ۱۰۰۰۰۰ ۱۰۰۰۰۰ ۱۰۰۰۰۰}	۸	۷
۰٫۶۵۰۰	{۰۰۰۰۰۰ ۱۰۰۰۰۰ ۱۰۰۰۰۰ ۱۰۰۰۰۰}	۱۱	۸
۰٫۶۵۰۸	{۰۰۰۰۰۰ ۱۰۰۰۰۰ ۱۰ ۱۰۰ ۱۰۰۰۰۰}	۲۱	۹
۰٫۶۵۰۰	{۰۰۰۰۰۰ ۱۰۰۰۰۰ ۱۰۰۰۰۰ ۱۰۰۰۰۰}	۱۱	۱۰



شکل ۱۹. قاب نمونه‌ی سوم.

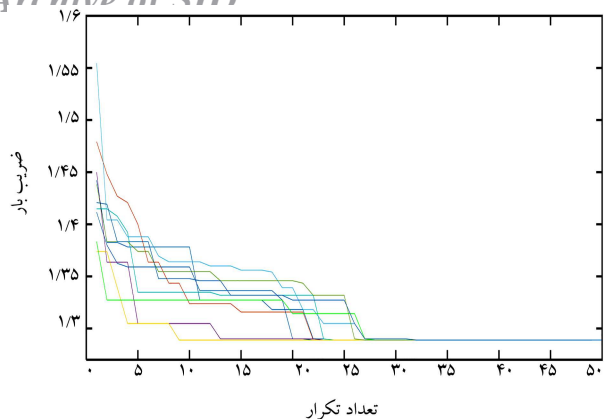
بار خطا مشاهده شد، که مشابه با مثال یک، بیشترین خطا را در بین سه الگوریتم ذکر شده داشت.

۳.۶. نمونه‌ی سوم

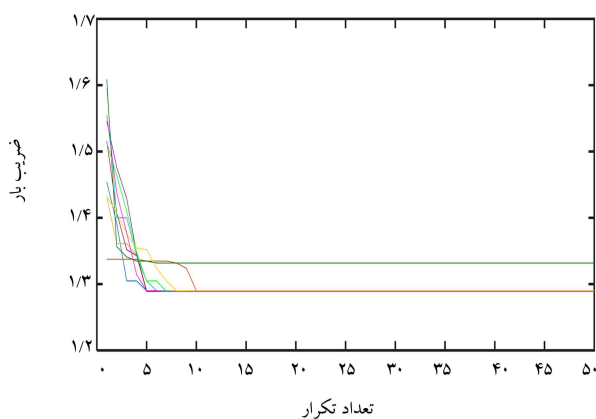
قاب نمونه‌ی سوم، ۶ طبقه و ۲ دهانه با ۱۸ مکانیزم فروریزش پایه بوده است. قاب و مشخصات آن در شکل ۱۹ و شماره‌گذاری مکانیزم‌های پایه هم در جدول ۹ ارائه شده است. پاسخ دقیق حاصل از تحلیل خمیری قاب مذکور هم موجود است. [۱۷] ضریب بار فروریزش واقعی در قاب نمونه‌ی سوم، ۱٫۲۸۸۹ است. مود فروریزش بحرانی متناظر در شکل ۲۰ مشاهده می‌شود.

مطابق مثال‌های قبل، هر الگوریتم ۱۰ مرتبه تحلیل شده است. نتایج حاصل شده در شکل‌های ۲۱ الی ۲۳ مشاهده می‌شود، که جزئیات آن‌ها در جدول‌های ۱۰ الی ۱۲ (بعد از هر نمودار همگرایی) ارائه شده است.

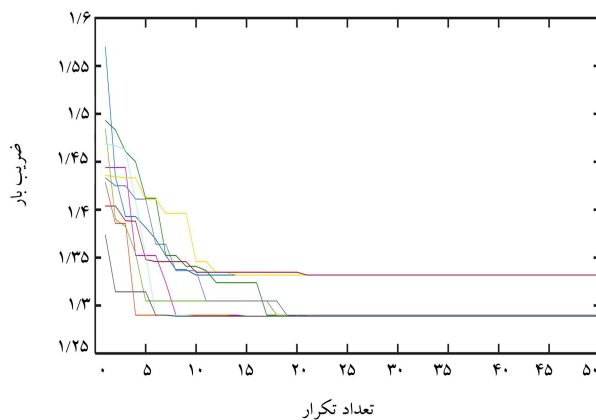
در مثال حاضر نیز الگوریتم دلفین باینری بدون خطا توانست در هر ۱۰ مرتبه، ضریب بار دقیق و مود فروریزش دقیق را ارائه دهد. الگوریتم گرگ خاکستری فقط دو مرتبه دچار خطا شد. این در حالی است که اگر ضریب بار فروریزش را تا دو رقم اعشار محاسبه کند، پاسخ فقط در یک مرتبه دچار خطا شده است؛ که اگر چه خطای مذکور فقط مربوط به ضریب بار فروریزش است و مود فروریزش متناظر آن در هر صورت کمی متفاوت است. تعداد خطاهای الگوریتم وال در مثال سوم،



شکل ۲۱. نتایج تحلیل نمونه‌ی سوم با استفاده از الگوریتم دلفین باینری.



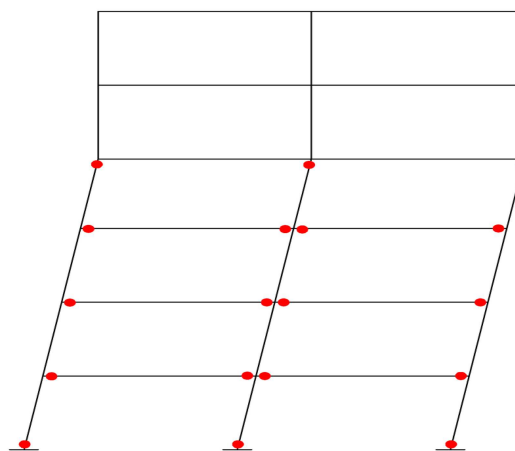
شکل ۲۲. نتایج تحلیل نمونه‌ی سوم با استفاده از الگوریتم گرگ خاکستری.



شکل ۲۳. نتایج تحلیل نمونه‌ی سوم با استفاده از الگوریتم وال.

جدول ۹. نحوه‌ی شماره‌گذاری مکانیزم‌ها.

شماره‌ی مکانیزم	موقعیت مکانیزم	ضریب بار
۱	طبقه‌ی اول	۲/۲
۲	تیر طبقه‌ی اول دهانه‌ی اول	۵/۲
۳	تیر طبقه‌ی اول دهانه‌ی دوم	۵/۲
۴	طبقه‌ی دوم	۲/۱۶
۵	تیر طبقه‌ی دوم دهانه‌ی اول	۴/۸
۶	تیر طبقه‌ی دوم دهانه‌ی دوم	۴/۸
۷	طبقه‌ی سوم	۲/۱
۸	تیر طبقه‌ی سوم دهانه‌ی اول	۳/۶
۹	تیر طبقه‌ی سوم دهانه‌ی دوم	۳/۶
۱۰	طبقه‌ی چهارم	۲
۱۱	تیر طبقه‌ی چهارم دهانه‌ی اول	۳/۲
۱۲	تیر طبقه‌ی چهارم دهانه‌ی اول	۳/۲
۱۳	طبقه‌ی پنجم	۱/۶
۱۴	تیر طبقه‌ی پنجم دهانه‌ی اول	۲/۸
۱۵	تیر طبقه‌ی پنجم دهانه‌ی دوم	۲/۸
۱۶	طبقه‌ی ششم	۴/۲
۱۷	تیر طبقه‌ی ششم دهانه‌ی اول	۲/۴
۱۸	تیر طبقه‌ی ششم دهانه‌ی دوم	۲/۴



شکل ۲۴. مود فروریزش واقعی نمونه‌ی سوم.

مثال سوم کمتر از الگوریتم گرگ خاکستری بود. نتیجه‌ی اخیر نشان می‌دهد که با وجود اینکه تعداد خطاهای وال بیشتر بود، اما این خطاها نسبت به الگوریتم گرگ خاکستری به جواب واقعی و دقیق نزدیک‌تر بوده است.

۷. نتیجه‌گیری

در پژوهش حاضر، تحلیل قاب‌های خمشی با استفاده از تحلیل حدی مبتنی بر تئوری کینماتیکی توسعه یافته و مکانیزم‌های خرابی پایه برای قاب تعیین شده‌اند. در ادامه،

۴ مرتبه بود، که مجدداً خطای زیادی نسبت به دو الگوریتم دیگر دارد. در انتها، جدول ۱۳، به منظور مقایسه‌ی عملکرد هر کدام از الگوریتم‌های ذکر شده در مسئله‌ی تحلیل خمیری سازه‌ها به کمک مقادیر آماری در ابتدای بخش حاضر، برای بیان خطاها ارائه شده است.

نکته‌ی جالب توجه این است که با وجود این که تعداد دفعاتی که الگوریتم وال پاسخی به جز پاسخ دقیق را نسبت به الگوریتم گرگ خاکستری ارائه داد، در هر ۳ نمونه بیشتر بود؛ خطای محاسبه شده به هر دو روش در همه‌ی مثال‌ها به جز

جدول ۱۰. جزئیات شکل ۲۱.

ضریب بار	مود فروریزش	تعداد تکرار	شماره‌ی تحلیل
۱٫۲۸۸۹	{۱۰۰۰۱۰۰۰۱۰۰۰۱۰۰۰۰۰۰۰۰}	۲۲	۱
۱٫۲۸۸۹	{۱۰۰۰۱۰۰۰۱۰۰۰۱۰۰۰۰۰۰۰۰}	۲۴	۲
۱٫۲۸۸۹	{۱۰۰۰۱۰۰۰۱۰۰۰۱۰۰۰۰۰۰۰۰}	۲۲	۳
۱٫۲۸۸۹	{۱۰۰۰۱۰۰۰۱۰۰۰۱۰۰۰۰۰۰۰۰}	۲۴	۴
۱٫۲۸۸۹	{۱۰۰۰۱۰۰۰۱۰۰۰۱۰۰۰۰۰۰۰۰}	۲۳	۵
۱٫۲۸۸۹	{۱۰۰۰۱۰۰۰۱۰۰۰۱۰۰۰۰۰۰۰۰}	۲۸	۶
۱٫۲۸۸۹	{۱۰۰۰۱۰۰۰۱۰۰۰۱۰۰۰۰۰۰۰۰}	۹	۷
۱٫۲۸۸۹	{۱۰۰۰۱۰۰۰۱۰۰۰۱۰۰۰۰۰۰۰۰}	۲۸	۸
۱٫۲۸۸۹	{۱۰۰۰۱۰۰۰۱۰۰۰۱۰۰۰۰۰۰۰۰}	۲۷	۹
۱٫۲۸۸۹	{۱۰۰۰۱۰۰۰۱۰۰۰۱۰۰۰۰۰۰۰۰}	۳۲	۱۰

جدول ۱۱. جزئیات شکل ۲۲.

ضریب بار	مود فروریزش	تعداد تکرار	شماره‌ی تحلیل
۱٫۲۸۸۹	{۱۰۰۰۱۰۰۰۱۰۰۰۱۰۰۰۰۰۰۰۰}	۱۱	۱
۱٫۲۸۸۹	{۱۰۰۰۱۰۰۰۱۰۰۰۱۰۰۰۰۰۰۰۰}	۱۳	۲
۱٫۲۹۰۰	{۱۰۰۰۱۰۰۰۱۰۰۰۱۰۰۰۰۰۰۰۰}	۶	۳
۱٫۲۸۸۹	{۱۰۰۰۱۰۰۰۱۰۰۰۱۰۰۰۰۰۰۰۰}	۷	۴
۱٫۲۸۸۹	{۱۰۰۰۱۰۰۰۱۰۰۰۱۰۰۰۰۰۰۰۰}	۵	۵
۱٫۲۸۸۹	{۱۰۰۰۱۰۰۰۱۰۰۰۱۰۰۰۰۰۰۰۰}	۸	۶
۱٫۳۳۱۸	{۱۰۰۰۱۰۰۰۱۰۰۰۱۰۰۰۱۰۰}	۶	۷
۱٫۲۸۸۹	{۱۰۰۰۱۰۰۰۱۰۰۰۱۰۰۰۰۰۰۰۰}	۱۰	۸
۱٫۲۸۸۹	{۱۰۰۰۱۰۰۰۱۰۰۰۱۰۰۰۰۰۰۰۰}	۷	۹
۱٫۲۸۸۹	{۱۰۰۰۱۰۰۰۱۰۰۰۱۰۰۰۰۰۰۰۰}	۷	۱۰

جدول ۱۲. جزئیات شکل ۲۳.

ضریب بار	مود فروریزش	تعداد تکرار	شماره‌ی تحلیل
۱٫۳۳۱۸	{۱۰۰۰۱۰۰۰۱۰۰۰۱۰۰۰۱۰۰}	۱۰	۱
۱٫۲۸۸۹	{۱۰۰۰۱۰۰۰۱۰۰۰۱۰۰۰۰۰۰۰۰}	۱۵	۲
۱٫۲۸۸۹	{۱۰۰۰۱۰۰۰۱۰۰۰۱۰۰۰۰۰۰۰۰}	۲۰	۳
۱٫۳۳۱۸	{۱۰۰۰۱۰۰۰۱۰۰۰۱۰۰۰۱۰۰}	۱۴	۴
۱٫۳۳۱۸	{۱۰۰۰۱۰۰۰۱۰۰۰۱۰۰۰۱۰۰}	۲۱	۵
۱٫۲۸۸۹	{۱۰۰۰۱۰۰۰۱۰۰۰۱۰۰۰۰۰۰۰۰}	۲۱	۶
۱٫۲۸۸۹	{۱۰۰۰۱۰۰۰۱۰۰۰۱۰۰۰۰۰۰۰۰}	۱۴	۷
۱٫۲۸۸۹	{۱۰۰۰۱۰۰۰۱۰۰۰۱۰۰۰۰۰۰۰۰}	۱۰	۸
۱٫۲۸۸۹	{۱۰۰۰۱۰۰۰۱۰۰۰۱۰۰۰۰۰۰۰۰}	۱۹	۹
۱٫۲۹۰۰	{۱۰۰۰۱۰۰۰۱۰۰۰۱۰۰۰۰۰۰۰۰}	۸	۱۰

جدول ۱۳. خطای RMSE و MAE حاصل از سه الگوریتم بهینه‌یابی.

الگوریتم	نمونه‌ی اول		نمونه‌ی دوم		نمونه‌ی سوم	
	تعداد خطا	MAE	RMSE	MAE	RMSE	تعداد خطا
دلفین باینری	۰	۰	۰	۰	۰	۰
گرگ خاکستری	۲	۰٫۰۳۰۰۰	۰٫۰۷۰۲۹	۰٫۰۰۳۳۴	۰٫۰۷۴۶۸	۲
وال	۴	۰٫۰۷۴۳	۰٫۱۳۳۷	۰٫۰۰۰۳۹	۰٫۰۰۰۷۵۳	۴

Archive of SID

به منظور استفاده در مسائل باینری بر اساس اصول موجود در الگوریتم پژواک دلفین ارائه شد، که ضمن افزایش دقت همگرایی، حجم محاسبات به شدت کاهش یافته است. بررسی نتایج ارائه شده در پژوهش حاضر در خصوص تحلیل خمیری قاب‌های خمشی نشان می‌دهد که روش اشاره شده در عین سرعت بالا، دقت بالایی نیز دارد و به خوبی قادر به تعیین مود خرابی قاب مورد نظر است. در نهایت نیز نتایج الگوریتم دلفین باینری با دو الگوریتم گرگ خاکستری و وال مقایسه شده است.

با ارائه الگوریتم فراابتکاری دلفین اصلاح شده، مکانیزم‌های پایه با یکدیگر ترکیب و مکانیزم فروریزش نهایی تعیین شده است. همچنین در پژوهش حاضر، به منظور کاهش حجم محاسبات و نیز افزایش سرعت مکانیزم‌های گره‌ی بدون دخالت در طول محاسبات به صورت خودکار محاسبه شده‌اند و این مسئله موجب کاهش قابل توجه حجم محاسبات و افزایش سرعت شده است. در ادامه، الگوریتم مکان‌یابی پژواکی دلفین برای مسائل باینری توسعه داده شد و الگوریتم بهینه‌یابی جدیدی

پانوشته‌ها

1. Grierson
2. dolphin echolocation (DE)
3. sonar
4. grey wolf optimization algorithm
5. whale optimization algorithm
6. root mean square error
7. mean absolute error

منابع (References)

1. Neal, B.G. and Symonds, P.S. "The rapid calculation of plastic collapse loads for a framed structure", *Proceedings of the Institution of Civil Engineers, London*, **1**(3), pp. 58-100 (1952).
2. Neal, B.G. and Symonds, P.S. "The calculation of collapse loads for framed structures", *Journal of the Institution of Civil Engineers*, **35**(1), pp. 21-40 (1950).
3. Watwood, Vernon B. "Mechanism generation for limit analysis of frames", *Journal of the Structural Division*, **105**(1), pp. 1-15 (1979).
4. Deeks, A.J. "Automatic computation of plastic collapse loads for frames", *Computers and structures*, **60**(3), pp. 391-402 (1996).
5. Chen, W.F. and Sohal, I., *Plastic Design and Second-order Analysis of Steel Frames*, Springer, New York Inc. (1995).
6. Wong, M.B., *Plastic Analysis and Design of Steel Structures*, Butterworth-Heinemann, 1st Edn (2009).
7. Kaveh, A. and Khanlari, K. "Collapse load factor for rigid-plastic analysis of frames using a genetic algorithm", *Civil-Comp. Proceedings*, **78**, ISBN 0-948749-92-X (2003).
8. Kaveh, A. and Khanlari, K. "Collapse load factor of planar frames using a modified genetic algorithm", *Commun Numer. Methods Eng*, **20**, pp. 911-925 (2004).
9. Kaveh, A. and Jahanshahi, M. "Plastic analysis of planar frames using kinematic method and genetic algorithm", *Asian J. Civil Eng. (Building and Housing)*, Tehran, Iran, **5**(3-4), pp. 145-160 (in persian) (2004).
10. Kohama, Y., Takada, T., Kozawa, N. and et al. "Collapse analysis of rigid frames by genetic algorithm", *In Proceedings of the Computer Aided Optimum Design of Structures*, pp. 193-202 (1997).
11. Kaveh, A. and Jahanshahi, M. "Plastic design of frames using heuristic algorithms", *In Topping BHV, Montero G, Montenegro R, Edotors, Proceedings of the 8th International Conference on Computational Structures Technology*, Stirlingshire, Scotland, Civil-Comp Press. Paper No 108 (2006).
12. Kaveh, A. and Jahanshahi, M. "Plastic limit analysis of frames using ant colony systems", *Computers & Structures*, **86**(11-12), pp. 1152-1163 (2008).
13. Jahanshahi, M., Pouraghajan, M. and Pouraghajan, M. "Enhanced ACS algorithms for plastic analysis of planar frames", *Comput. Methods Civil Eng*, **4**(1), pp. 65-82 (2013).
14. Kaveh, A., Jahanshahi, M. and Khanzadi, M. "Plastic analysis of frames using genetic algorithm and ant colony algorithm", *Asian J. Civil Eng*, **9**(3), pp. 227-246 (2008).
15. Jahanshahi, M., Maleki, E. and Ghiami, A. "On the efficiency of artificial neural networks for plastic analysis of planar frames in comparison with genetic algorithms and ant colony systems", *Neural Comput. Appl*, (published online) (2016). <http://dx.DOI.org/10.1007/s00521-016-2228-5>
16. Kaveh, A. and Ghafari, M.H. "Plastic analysis of planar frames using CBO and ECBO algorithms", *Int. J Optim. Civil Eng*, **5**(4), pp. 479-492 (2015).
17. Kaveh, A. and Ilchi Ghazaan, M. "Computer codes for colliding bodies optimization and it's enhanced version", *Int. J. Optim. Civil Eng*, **3**(4), pp. 321-332 (2014).
18. Greco, A., Cannizzaro, F. and Pluchino, A. "Seismic collapse prediction of frame structures by means of genetic algorithm", *J. Engineering Structures*, **143**, pp. 152-168 (2017).
19. Kennedy, J. and Eberhart, R.C. "A discrete binary version of the particle swarm algorithm", *Systems, Man, and Cybernetics. Computational Cybernetics and Simulation, 1997 IEEE International Conference on 5. IEEE* (1997).
20. Ahmadi Khanesar, M., Teshnehlab, M. and Aliyari Shoorehdeli, M. "A novel binary particle swarm optimization", *Control & Automation, MED'07. Mediterranean Conference on. IEEE* (2007).
21. Mirjalili, S., Mirjalili, S.M. and Yang, X.-Sh. "Binary bat algorithm", *Neural Computing and Applications*, **25**(3-4), pp. 663-681(2014).
22. Srikanth, R.K., Panwar, L., Panigrahi, B.K. and et al. "Binary whale optimization algorithm: a new meta-heuristic approach for profit-based unit commitment

problems in competitive electricity markets”, *Engineering Optimization*, **51**(3), pp. 369-389 (2019).

23. Kaveh, A. and Farhoudi, N. “A new optimization method: dolphin echolocation”, *J. Advances in Engineering Software*, **59**, pp. 53-70 (2013).

24. Strang, G. “Linear algebra and its applications (academic, New York), *Google Scholar*, pp. 138-143 (1976)

25. Grierson, D.E. and Gladwell, G.L.M. “Collapse load using linear programming”, *J. Struct Div*, **97**, pp. 1561-1573 (1971).

26. Mirjalili, S., Mirjalili, S.M. and Lewis, A. “Grey wolf optimizer”, *Advances in Engineering Software*, **69**, pp. 46-61 (2014).

27. Mirjalili, S. and Lewis, A. “The whale optimization algorithm”, *Advances in Engineering Software*, **95**, pp. 51-67 (2016).

28. Gandomi, A.H., Mohammadzadeh, D., Pérez-Ordóñez, J.L. and et al. “Linear genetic programming for shear strength prediction of reinforced concrete beams without stirrups”, *Applied Soft Computing*, **19**, pp. 112-120 (2014).

29. Beheshti Aval, S.B., Ketabdari, H. and Asil Gharebaghi, S. “Estimating shear strength of short rectangular reinforced concrete columns using nonlinear regression and gene expression programming”, *Structures*, **12**, Elsevier (2017).

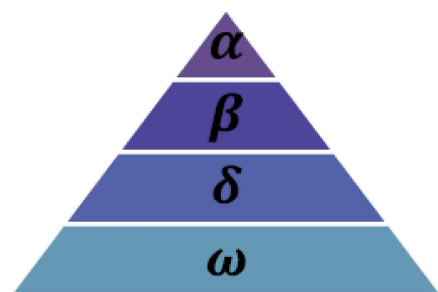
پیوست

در بخش پیوست مروری بر توضیحات مربوط به الگوریتم گرگ خاکستری و وال که در نمونه‌ها استفاده شده‌اند، انجام شده است.

— الگوریتم گرگ خاکستری

گرگ‌های خاکستری به‌عنوان شکارچینی هستند که بیشتر تمایل دارند به صورت گروهی زندگی کنند. جمعیت گروه به طور میانگین ۵ تا ۱۲ عضو است. از بزرگترین مشخصه‌های آن‌ها این است که آن‌ها رتبه‌بندی اجتماعی غالبی مطابق شکل (پ - ۱) دارند.

رهبران گروه متشکل از یک زن و یک مرد هستند که آلفا نامیده می‌شوند، که غالباً مسئول تصمیم‌گیری برای شکار، مکان خواب، زمان بیدار شدن و ... هستند. تصمیمات آلفا به گروه دیکته می‌شود. دومین رده در رتبه‌بندی گرگ‌های خاکستری، بتاست. گرگ‌های بتا گرگ‌های تحت امر آلفا هستند، که به آلفا در تصمیم‌گیری و سایر فعالیت‌های گروه کمک می‌کنند. گرگ‌های بتا می‌توانند مرد یا زن باشند و بهترین گزینه برای جایگزینی آلفا هستند. زمانی که گرگ‌های آلفا بمیرند یا خیلی پیر شوند. پایین‌ترین رتبه‌ی گرگ‌ها امگا است. آن‌ها مجبورند از سایر گرگ‌های بالاتر تبعیت کنند و آخرین اعضایی هستند که اجازه‌ی غذا خوردن دارند. اگر یک گرگ آلفا، بتا، یا امگا نبود، این گرگ، تحت امر یا دلتا نامیده می‌شود. گرگ‌های دلتا باید از آلفا و بتا اطاعت کنند، ولی بر امگا برتری دارند. پیش‌آهنگ‌ها،^۸ نگهبان‌ها،^۹ بزرگ‌ترها،^{۱۰} شکارچیان^{۱۱} و سرپرست‌ها^{۱۲} به این گروه تعلق دارند. علاوه بر رتبه‌بندی اجتماعی گرگ‌ها، شکار گروهی یک رفتار جالب دیگر از گرگ‌های خاکستری است. فازهای



شکل پ - ۱. رتبه‌بندی اجتماعی گرگ‌ها.

اصلی شکار آن‌ها به این صورت است:

- ردیابی، تعقیب و نزدیک شدن به طعمه.
- دنبال کردن، محاصره کردن و اذیت کردن طعمه تا زمانی که از حرکت بایستد.
- حمله به سمت طعمه.

در الگوریتم گرگ خاکستری، این روش شکار به‌صورت ریاضی به منظور بهینه‌یابی مدل شده است. در این الگوریتم، بهترین جواب به‌عنوان آلفا، دومین و سومین بهترین جواب به ترتیب به‌صورت بتا و دلتا تعریف می‌شوند و بقیه‌ی پاسخ‌ها امگا فرض می‌شوند. شکار معمولاً توسط آلفا هدایت می‌شود، هر چند بتا و دلتا هم گاه‌ها در شکار شرکت می‌کنند. برای شبیه‌سازی ریاضی فرض می‌شود که آلفا (بهترین کاندید پاسخ)، بتا و دلتا اطلاعات بهتری از موقعیت طعمه دارند. بنابراین ۳ پاسخ اول ذخیره می‌شوند و الزام می‌کنند که سایر اعضا (شامل امگا) موقعیت خود را طبق موقعیت بهترین پاسخ به روز کنند.

به‌طور خلاصه فرایند جستجو با ساخت یک جمعیت رندم از گرگ‌ها در الگوریتم آغاز می‌شود. در طی فرایند تکرار آلفا، بتا و دلتا موقعیت محتمل طعمه را حدس می‌زنند و سپس هر پاسخ کاندید فاصله‌ی خود را از طعمه به روز می‌کند. در ادامه، روابط مربوط به این الگوریتم به همراه توضیحات آن ارائه شده است (روابط الف الی ج):

$$\vec{D} = \left| \vec{C} \cdot \vec{X}_p(t) - \vec{X}(t) \right| \quad (\text{الف})$$

$$\vec{X}(t+1) = \vec{X}_p(t) - \vec{A} \cdot \vec{D} \quad (\text{ب})$$

$$\vec{A} = 2\vec{a} \cdot \vec{r}_1 - \vec{a} \quad (\text{پ})$$

$$\vec{C} = 2 \cdot \vec{r}_2 \quad (\text{ت})$$

$$\vec{D}_\alpha = \left| \vec{C}_1 \cdot \vec{X}_\alpha - \vec{X} \right|, \vec{D}_\beta = \left| \vec{C}_r \cdot \vec{X}_\beta - \vec{X} \right|, \vec{D}_\delta = \left| \vec{C}_r \cdot \vec{X}_\delta - \vec{X} \right| \quad (\text{ث})$$

$$\vec{X}_1 = \vec{X}_\alpha - \vec{A}_1 \cdot (\vec{D}_\alpha),$$

$$\vec{X}_r = \vec{X}_\beta - \vec{A}_r \cdot (\vec{D}_\beta), \vec{X}_r = \vec{X}_\delta - \vec{A}_r \cdot (\vec{D}_\delta) \quad (\text{ج})$$

$$\vec{X}(t+1) = \frac{\vec{X}_1 + \vec{X}_r + \vec{X}_r}{3} \quad (\text{ج})$$

Archive of SID

الف) مکانیزم احاطه کردن انقباضی: این رفتار با کاهش مقدار \vec{a} در رابطه‌ی D به دست می‌آید. توجه شود که بازه‌ی نوسان \vec{A} هم توسط \vec{a} کاهش می‌یابد. به بیان دیگر، \vec{A} یک مقدار رندم در بازه‌ی $[-a, a]$ است که در آن a از 2 تا صفر در طی فرایند تکرار کاهش می‌یابد. با اختصاص مقادیر رندم برای \vec{A} در $[-1/1]$ موقعیت جدید جستجوگر در هر نقطه بین موقعیت اصلی جستجوگر و موقعیت بهترین جستجوگر فعلی قابل تعیین است.

ب) به روزکردن موقعیت به صورت اسپیرال: این رویکرد ابتدا فاصله‌ی بین وال که در نقطه‌ی (X, Y) و طعمه که در نقطه‌ی (X^*, Y^*) محاسبه می‌شود. یک رابطه‌ی اسپیرال بین موقعیت وال و طعمه برای شبیه‌سازی حالت مارپیچی وال‌های گوژپشت به شکل رابطه‌ی (ط) تعیین می‌شود:

$$\vec{X}(t+1) = \vec{D}^l \cdot e^{bl} \cdot \cos(2\pi l) + \vec{X}^*(t) \quad (\text{ط})$$

که در آن، $|\vec{X}^*(t) - \vec{X}(t)| = \vec{D}^l$ و بیان‌گر فاصله‌ی وال نام تا طعمه (بهترین پاسخی که تا کنون به دست آمده است)، b ثابتی است برای تعیین شکل اسپیرال لگاریتمی و l عددی رندم در بازه‌ی $[-1/1]$ است.

بر اساس تغییرات بردار \vec{A} رویکرد مشابهی می‌تواند برای جستجوی طعمه استفاده شود. در واقع، وال‌ها به صورت رندم بر اساس موقعیت یکدیگر جستجو را انجام می‌دهند. بنابراین برای \vec{A} مقادیر رندم بزرگتر از 1 و کوچک‌تر از -1 در نظر گرفته می‌شود، تا جستجوگرها را وادار به این کند که از وال مرجع دور شوند. برخلاف حالت قبل، موقعیت جستجوگر بر اساس انتخاب رندم آن به جای انتخاب بهترین جستجوگری که قبلاً به دست آمده است، انجام می‌شود. این مکانیزم $|\vec{A}| > 1$ به الگوریتم وال این اجازه را می‌دهد که جستجوی کلی انجام دهد. مدل ریاضی آن به صورت رابطه‌های \vec{D} و \vec{C} بیان می‌شود:

$$\vec{D} = \left| \vec{C} \cdot \vec{X}_{rand} - \vec{X} \right| \quad (\text{ظ})$$

$$\vec{X}(t+1) = X_{rand} - \vec{A} \cdot \vec{D} \quad (\text{ع})$$

که در آنها، \vec{X}_{rand} بردار موقعیت رندم است (یک وال رندم) که از جمعیت موجود انتخاب شده است.

الگوریتم WOA با مجموعه‌ی پاسخ‌های رندم شروع می‌شود. در هر تکرار جستجوگرها موقعیت خود را یا با توجه به جستجوگرهایی که به صورت رندم انتخاب شده‌اند و یا با توجه به بهترین پاسخ به دست آمده به روز می‌کنند. پارامتر a از 2 تا صفر کاهش می‌یابد. یک جستجوگر رندم، زمانی که $|\vec{A}| > 1$ انتخاب می‌شود، در حالی که بهترین پاسخ زمانی انتخاب می‌شود که $|\vec{A}| < 1$ است. بسته به مقدار p ، الگوریتم قادر است که بین هر اسپیرال یا حرکت دایره‌ی جابه‌جا شود. نهایتاً، الگوریتم WOA با ارضاء ضابطه‌ی اتمام، پایان می‌یابد.

از دید تئوری، الگوریتم WOA به عنوان یک بهینه‌یاب گلوبال قابل در نظر گرفتن است. به دلیل این‌که مکانیزمی که قبلاً توضیح داده شد، فضای جستجویی را در همسایگی بهترین پاسخ تعریف می‌کند و به سایر جستجوگرها این اجازه را می‌دهد که بهترین پاسخ موجود را از داخل دامنه‌ی مذکور استخراج کنند. علاوه بر این، با کاهش A برخی تکرارها به فاز بازرسی اختصاص می‌یابند ($|\vec{A}| \geq 1$) و بقیه به فاز استخراج اختصاص داده می‌شوند. در نهایت، برای الگوریتم وال هم فقط دو پارامتر اصلی داخلی باید تنظیم شوند، که A و C هستند.

که در آنها، t بیان‌گر شماره‌ی تکرار کنونی، \vec{A} و \vec{C} بردار ضرایب، \vec{X}_p موقعیت بردار طعمه و \vec{X} هم موقعیت بردار گرگ خاکستری است. بردارهای \vec{A} و \vec{C} به صورت روابط (ب) و (ت) به دست می‌آیند و در آن مؤلفه‌ی a به صورت خطی از 2 به 0 در طی مراحل تکرار کاهش می‌یابند و همچنین \vec{r}_1 و \vec{r}_2 بردارهای رندم در بازه‌ی $[0/1]$ هستند. در رابطه‌ی ج، موقعیت گرگ‌ها به روز می‌شود. در واقع آلفا، بتا و دلتا موقعیت طعمه را حدس می‌زنند و سایر گرگ‌ها موقعیت خود را به صورت رندم در اطراف طعمه به روز می‌کنند.

با توجه به موارد گفته شده در الگوریتم گرگ خاکستری فقط دو پارامتر a و C باید تنظیم شود، که این سهولت کار با الگوریتم یکی از قابلیت‌های بیشتر الگوریتم‌های بهینه‌یابی فراابتکاری است.

-- الگوریتم وال

وال‌ها بزرگترین پستانداران جهان هستند و عمدتاً به عنوان شکارچی در نظر گرفته می‌شوند. یکی از بزرگترین گونه‌های وال‌ها، وال‌های گوژپشت هستند که ابعادی به اندازه‌ی یک اتوبوس مدرسه دارند و طعمه‌ی مورد علاقه‌ی آن‌ها، حلزون‌های کوچک و گروه ماهی‌های کوچک است. جالب‌ترین موضوع درباره‌ی وال‌های گوژپشت، روش مخصوص شکار آن‌هاست، که موجودات باد شده را به صورت گروهی در نزدیکی سطح دریا شکار می‌کنند. مشاهده شده است که این جستجو با ساختن حباب‌های متفاوت به صورت دایره‌هایی شبیه 9 انجام می‌شود. مانور غذا خوردن با تله‌ی ساختن حباب‌های اسپیرال برای انجام بهینه‌یابی به صورت ریاضی مدل شده است.

وال‌های گوژپشت می‌توانند موقعیت طعمه را شناسایی کنند و آن‌ها را احاطه کنند. از آنجایی که موقعیت طراحی بهینه از قبل مشخص نیست، الگوریتم WOA فرض می‌کند که بهترین پاسخ کاندید فعلی، طعمه‌ی هدف است یا این پاسخ نزدیک به بهینه است. بعد از این‌که بهترین جستجوکننده تعریف شد، سایر جستجوکننده‌ها تلاش خواهد کرد که موقعیت خود را به سمت بهترین جستجوکننده به روز کنند. این رفتار با روابط (ح) و (خ) معرفی می‌شوند:

$$\vec{D} = \left| \vec{C} \cdot \vec{X}^*(t) - \vec{X}(t) \right| \quad (\text{ح})$$

$$\vec{X}(t+1) = \vec{X}^*(t) - \vec{A} \cdot \vec{D} \quad (\text{خ})$$

که در آنها، t شماره‌ی تکرار فعلی، \vec{A} و \vec{C} بردارهای ضرایب، X^* بردار موقعیت بهترین پاسخی که قبلاً به دست آمده است و X بردار موقعیت هستند. ضمن این‌که در صورت وجود پاسخ بهتر X^* در هر تکرار باید به روز شود. بردارهای \vec{A} و \vec{C} هم به صورت رابطه‌های (د) و (ذ) محاسبه می‌شوند:

$$\vec{A} = 2\vec{a} \cdot \vec{r} - \vec{a} \quad (\text{د})$$

$$\vec{C} = 2 \cdot \vec{r} \quad (\text{ذ})$$

که در آنها، t به صورت خطی از 2 به صفر در طی تکرارها کاهش می‌یابد و \vec{r} بردار رندم در بازه‌ی $[0/1]$ است.

به منظور این‌که رفتار ایجاد حباب به صورت ریاضی مدل شود، دو رویکرد به این صورت طراحی شده است: