ارائه تکنیکی مبتنی بر جستجوی المان‌ها برای تعیین مسیر تقسیم پاره‌سازه‌ها در روش اجزای محدود بوسیله‌ی الگوریتم ژنتیک

وحید شیرعلی‌نژاد[1]، حمید مسلمی[2]

۱- کارشناسی ارشد مهندسی عمران-سازه. دانشکده فنی و مهندسی دانشگاه شاهد

۲- استادیار گروه مهندسی عمران-سازه. دانشکده فنی و مهندسی دانشگاه شاهد

vahidshiralinezhad@gmail.com

خلاصه

یکی از تکنیک‌های موثر برای افزایش سرعت در حل اجزای‌محدود[1]، استفاده از تکنیک پاره‌سازه‌ها[2] می‌باشد که با تقسیم سازه به قسمت‌های کوچکتر، سرعت حل معادلات همزمان، بسیار افزایش می‌یابد. یکی از چالش‌های موجود در این تکنیک نحوه‌ی تقسیم پاره‌سازه‌ها در مسائل مختلف می‌باشد به گونه‌ای که یک تقسیم‌بندی نامناسب ناشی می‌تواند از راندمان این تکنیک بکاهد. در این مقاله تکنیکی بر مبنای جستجوی المان و با استفاده از الگوریتم ژنتیک ارائه شده است که با جابجا کردن المان‌ها بین پاره‌سازه‌ها، تابع هزینه مجموع عملیات را در گام‌های متوالی به حداقل می‌رساند.

کلمات کلیدی: روش اجزای‌محدود، روش پاره‌سازه‌ها، الگوریتم ژنتیک، مسیر تقسیم پاره‌سازه‌ها

۱. مقدمه

امروزه برای تحلیل سازه‌ها تمایل زیادی به روش‌های‌عددی ایجاد شده‌است که دلیل این امر دشوار بودن و صرف زمان زیاد برای تحلیل می‌باشد و حتی گاهی در مسیر تحلیل به معادلاتی برخورد می‌شود که پاسخ تحلیلی ندارند و همین باعث عدم رسیدن به پاسخ مناسب برای تحلیل می‌باشد. روش‌های عددی فارغ از تحلیل‌های مرسوم تحلیلی، از روش‌های مخصوص به خود این تحلیل‌ها را انجام می‌دهند ولی قدری تقریب در پاسخ‌ها وارد می‌شود که صرفه جویی زمانی ناشی از استفاده از این روش‌ها از نظر اغلب مهندسان ارزش وارد شدن قدری تقریب در تحلیل‌ها را دارد.

از روش‌های عددی کارآمدی که امروزه بسیار مورد استفاده قرار می‌گیرد، روش اجزای‌محدود می‌باشد. گاها مشاهده می‌شود که حجم محاسبات به قدری زیاد است که حتی روش‌های عددی نیز زمان زیادی برای تحلیل نیاز دارند که همین امر باعث شد که روش مناسب برای تحلیل با سرعت بیشتر برای سازه‌های حجیم ارائه گردد که یکی از این روش‌ها، روش پاره‌سازه‌ها می‌باشد که زمان تحلیل را بشدت کاهش می‌دهد. در این روش، سازه به چندین زیرسازه‌ی کوچکتر تحت عنوان پاره‌سازه تقسیم می‌شود و سپس با توجه به الگوریتم این روش، پاره‌سازه‌ها تحلیل شده و سپس پاسخ نهایی که ترکیبی از پاسخ‌های پاره‌سازه‌ها و تاثیر آنها بر روی یکدیگر می‌باشد، حاصل می‌شود.

از عوامل مهم در افزایش سرعت تحلیل در این روش یافتن تعداد بهینه‌ی پاره‌سازه‌ها و همچنین مسیر تقسیم این پاره‌سازه‌ها و تقسیم المان‌ها در بین پاره‌سازه‌ها می‌باشد و در صورتی که این دو به درستی انجام نشوند به مطلوبیت کافی در این روش دست پیدا نخواهد شد. به همین منظور روش‌های بسیاری برای این امور ارائه شده است که هیچکدام روش مطلق و نهایی نمی‌باشند و به همین دلیل تلاش برای رسیدن به پاسخ نهایی تقسیم پاره‌سازه‌ها همچنان ادامه دارد. در این مقاله تمرکز بر روی یافتن مسیر مناسب برای تقسیم پاره سازه‌ها می‌باشد.

[1] Finite Element
[2] Substructure

استفاده از ایده‌ی پاره‌سازه‌ها از سال ۱۹۶۳ توسط پرزیمینسکی مطرح گردید[۱]. برای یافتن مسیر مناسب برای تقسیم پاره‌سازه‌ها باید از روش‌هایی استفاده کرد که در این خصوص مطالعاتی انجام گرفته است که شماری از آنها در زیر آورده شده است:

در سال ۱۹۹۰ پوتن و همکارانش روشی را تحت عنوان ORB[1] ارائه کردند. این روش به این گونه بود که تقسیم‌بندی کاملا ابتدایی و تنها برای مقاصد آزمایشی بکار برده‌شد. این تقسیم‌بندی به این ترتیب بود که هر بازه‌ی مختصاتی را به دو نیم بازه‌ی مساوی از لحاظ مختصاتی تقسیم می‌کرد[۲].

در سال ۱۹۹۱ سیمون روش جدیدی تحت عنوان RGB[2] ارائه نمود که روشی به مراتب قوی‌تر از روش ORB بود ولی تنها تفاوت آنها این بود که در ابتدا، در این روش تقسیم‌بندی بر خلاف ORB در دو جهت انجام می‌گرفت [۳] و معیار تقسیم نیز مرکز سطح هر المان می‌بود که با بوجود آمدن حجم محاسباتی بالا در این روش از مجموعه‌های برنامه‌نویسی شده‌ی جورج و همکارانش استفاده شد[۴].

در سال ۲۰۰۳ این روش با مبانی ماتریس‌ها و بردارهای ویژه ترکیب نموده شد و بدین گونه در پاسخ‌ها بهبود چشم‌گیری مشاهده گردید. بعد از این کار تقسیم دیگر براساس مختصات انجام نمی‌شد و براساس بردارهای ویژه این کار انجام می‌گرفت[۵].

پیشنهاد بعدی روش RCB[3] بود که تقسیم‌بندی متفاوتی را نسبت به روش‌های قبلی ارائه داد. این روش تقسیم‌بندی را بر روی محیط یک دایره انجام می‌دهد که توسط گیلبرت پایه‌گذاری شد[۶] و در سال ۲۰۱۷ این روش گسترش داده شد و تقسیم‌بندی نهایی ارائه گردید. در این کار با مدنظر قرار دادن دو شرط المان برابر و گره مشترک کمتر، تقسیم‌بندی را انجام می‌دهند. از نقاط ضعف این روش می‌توان به مقدار زیاد سعی و خطا برای رسیدن به تقسیم‌بندی مناسب اشاره نمود[۷].

نوع دیگری از روش‌های تقسیم‌بندی، روش‌هایی با عمر نسبتا کم و جدید، تحت عنوان روش‌های "تقیسم‌بندی با کمک تقسیم‌بندی اولیه" می‌باشد. از جمله این روش‌ها می‌توان به ... ITS[4], PR[5], DLS[6], LNA[7] اشاره نمود که نحوه‌ی انجام تقسیم‌بندی با این الگوریتم‌ها تا حدود زیادی مشابه می‌باشد و تنها در اپراتورهای جستجوی آن تغییراتی ایجاد شده است. روش ITS روشی مبتنی بر تئوری گراف‌ها می‌باشد ولی از روش‌های بسیار جدید می‌باشد. این روش با یک تقسیم‌بندی ابتدایی شروع به تقسیم‌بندی می‌کند و ابتدا تقسیم‌بندی ابتدایی را بهینه می‌کند و سپس وارد مرحله‌ی دوم می‌شود. برای تقسیم‌بندی نیز از دو اپراتور حرکت و اپراتور جابجایی استفاده می‌کند. اپراتور حرکت یک حل بهینه‌ی محلی را انجام می‌دهد و سپس اپراتور جابجایی با جابجایی المان‌های ناهمگون سعی در بهبود ثانویه‌ی تقسیم‌بندی می‌نماید. این روش دارای سه فاز می‌باشد:

الف- بهبود نزولی ب- بهبود تنوعی پ- اغتشاش [۸]

ابتدا گراف به دو دسته تقسیم می‌شود S1, S2 و حال I={S1,S2} و سپس برای اینکه متوجه شویم که جابجایی موردنظر مناسب بوده است یا خیر، پارامتری با عنوان f(I) مطرح گردید که تا زمانی که مقدار این پارامتر بیشتر شود یعنی جابجایی مناسب بوده است. بعدها با توجه به اینکه این روش زمان نسبتا زیادی را صرف این محاسبات می‌نماید، روش باکت ارائه شد که مقداری تقریب را وارد این روش می‌نماید ولی در رسیدن به پاسخ مناسب، زمان کمتری صرف خواهد شد.[۹]

اعداد موجود در باکت‌ها با توجه به تعداد اتصالشان به گره‌های داخل پاره‌سازه‌ی مربوط به خود و تعداد اتصالات به پاره‌سازه‌ی دیگر حاصل می‌شود. ابتدا این روش را برای دو پاره سازه استفاده کردند ولی در ادامه با تغییر در اپراتورها توانستند این روش را برای تعداد بالاتر پاره‌سازه نیز استفاده نمایند [۸].

1 Orthogonal recursive bisection
2 Recursive graph bisection
3 Recursive circle bisection
4 Iterated tabu search
5 Path relinking
6 Descent local search
7 Lagrangian net algorithm

۲. روش پاره‌سازه‌ها

استفاده از ایده‌ی پاره‌سازه‌ها ابتدا در سال ۱۹۶۳ توسط پرزیمنسکی ارائه شد. ابتدا باید سازه‌ی اصلی را به پاره‌سازه‌ها تقسیم نموده و سپس وارد الگوریتم حل این روش شد.
در این روش ابتدا معادله ی نیرو–جابجایی نوشته می‌شود که بقرار زیر می‌باشد:

$$\begin{pmatrix} F_i \\ F_e \end{pmatrix} = \begin{pmatrix} K_{ii} & K_{ie} \\ K_{ei} & K_{ee} \end{pmatrix} \begin{pmatrix} d_i \\ d_e \end{pmatrix} \qquad (۱)$$

رابطه‌ی (۱) برای هر پاره‌سازه نوشته می‌شود. در رابطه‌ی (۱) اندیس i مربوط به گره‌های موجود در سطح مشترک و اندیس e مربوط به گره‌های داخلی در هر پاره‌سازه می‌باشد.

ابتدا ماتریس‌های سختی هر پاره‌سازه را به روش اجزای‌محدود که بدون پاره‌سازه می‌باشد نوشته و سپس المان‌هایی که در یک پاره‌سازه می‌باشند را اسمبل کرده و سپس شرایط مرزی نیز اعمال می‌شود و سپس با توجه به موقعیت هر گره در پاره‌سازه (داخلی یا مرزی بودن)، ماترس سختی به فرم $\begin{pmatrix} K_{ii} & K_{ie} \\ K_{ei} & K_{ee} \end{pmatrix}$ درآورده می‌شود. باید دقت نمود که K_{ee},K_{ii} ,K_{ie} , K_{ei} لزوما یک درایه از ماتریس نبوده و هر یک به طور مجزا یک ماتریس می‌باشند.
با توجه به رابطه‌ی (۱) روابط مهمی که برای حل پاره‌سازه‌ها کمک می‌کنند، بدست آورده می‌شود:

$$d_e=[K_{ee}]^{-1}[F_e - K_{ie}\, d_i] \qquad (۲)$$
$$F_i - K_{ie}[K_{ee}]^{-1}F_e = (K_{ii} - K_{ie}[K_{ee}]^{-1}K_{ei})\; d_i \qquad (۳)$$

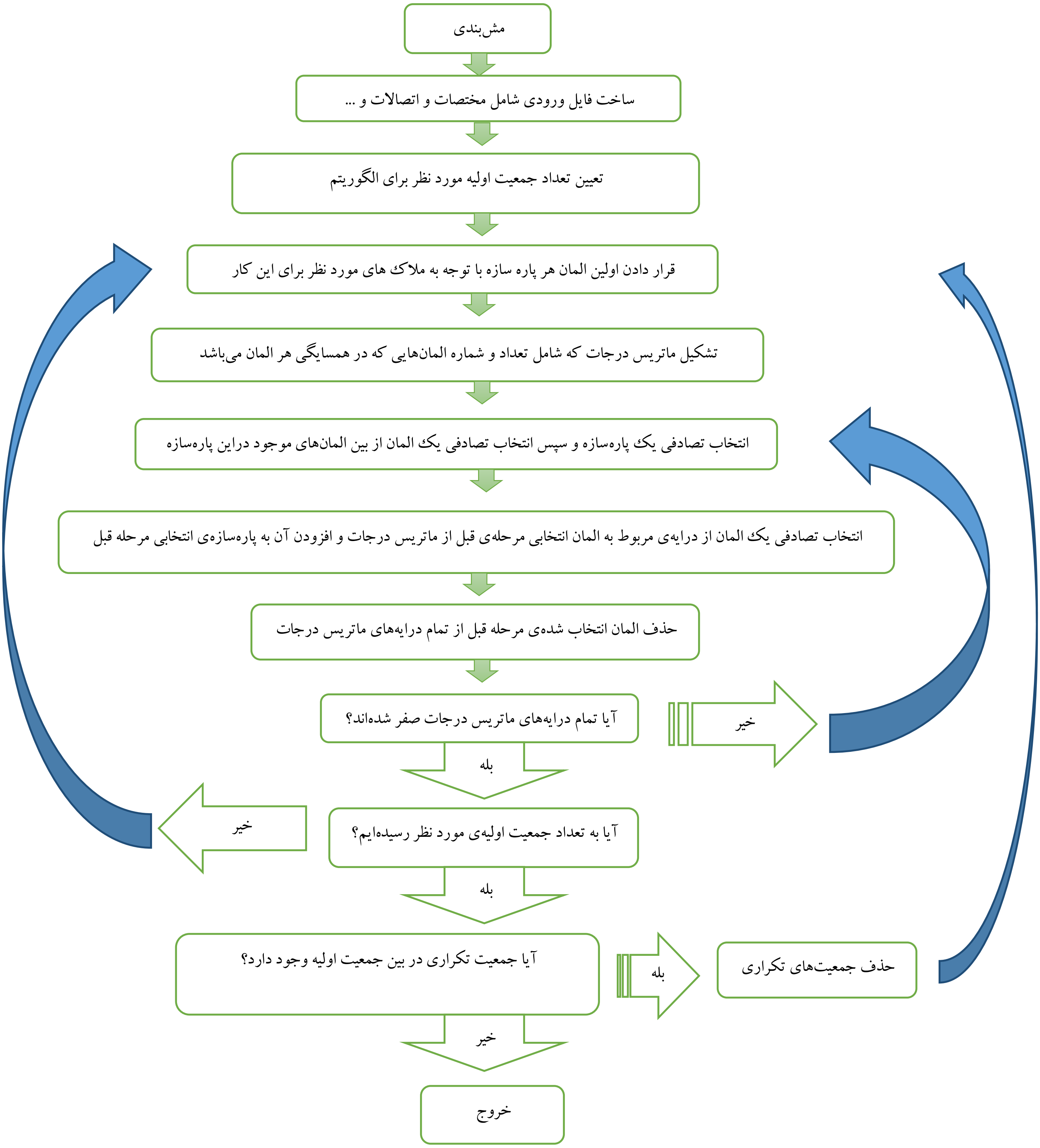
به کمک رابطه‌ی (۳) مقدار d_i محاسبه می‌شود و سپس با رابطه‌ی (۲) مقدار جابجایی نقاط داخلی موجود در داخل هر پاره‌سازه نیز بدست آورده می‌شود. بعد از بدست آمدن d_i و d_e باید اثر پاره‌سازه‌های دیگر بر روی پاره‌سازه‌ی مورد نظر را نیز اعمال کرد. در گره‌های داخلی، پاره‌سازه‌های کناری اثری بر مقدار جابجایی ندارند، چون اندرکنشی با این گره‌ها ندارند. ولی در مورد گره‌های مرزی که با پاره‌سازه‌های دیگر در ارتباط است باید مقدار جابجایی هر گره با گره متناظر خود در سایر پاره‌سازه‌ها جمع شده و بدین ترتیب جابجایی گره‌های مرزی حاصل می‌شود [۱۰].

۳. ارائه تکنیکی مبتنی بر جستجوی المان‌ها برای تعیین مسیر تقسیم پاره‌سازه ها

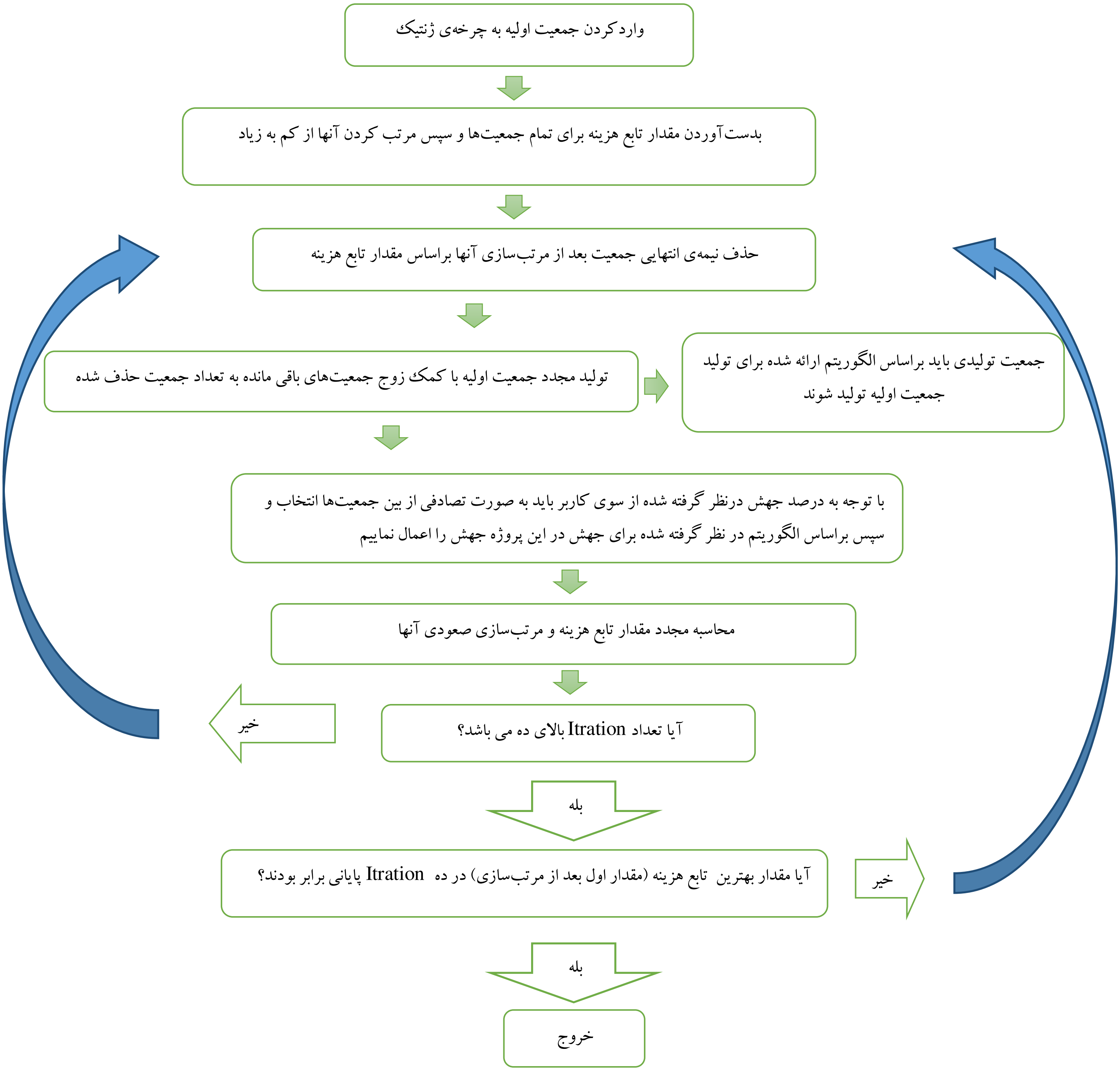
در این روش با توجه به اینکه در تحلیل پاره‌سازه، بخش زیادی از زمان تحلیل را معکوس‌کردن ماتریس‌های سختی داخلی و نیز ماتریس سختی مربوط به گره‌های مرزی به خود اختصاص می‌دهد سعی در این است که حجم ماتریس‌های سختی را کاهش دهیم که البته در تحلیل دینامیکی این صرف زمان برای معکوس کردن ماتریس‌های سختی بسیار بیشتر نمود پیدا می‌کند. با این کار علاوه بر این که زمان معکوس‌کردن کاهش می‌یابد بلکه چون حجم اغلب ماتریس‌ها کاهش یافته است در اکثر عملیات‌های جبری که بر روی ماتریس‌ها انجام می‌شود زمان کمتری صرف خواهد شد. البته با توجه به اینکه زمان معکوس‌کردن با مقیاس ۳ (Order) وارد تابع هزینه می‌شود تمرکز بیشتر بر روی ماتریس‌های سختی‌ای خواهد بود که قرار است معکوس شوند.

در این روش با توجه به نکات ذکر شده، باید حجم ماتریس‌های سختی داخلی هر پاره‌سازه و حجم ماتریس‌های سختی مرزی را کاهش دهیم که ماتریس سختی مرزی با توجه به رابطه‌ی (۳) ترکیبی از K_{ee} و K_{ii}و K_{ie} و K_{ei} که در آن تنها ماتریس K_{ee} هر پاره‌سازه معکوس می‌شود و پس از آنکه ضریب d_i در رابطه‌ی (۳) محاسبه شد به ماتریس سختی گره‌های مرزی کلی رسیده‌ایم که حال باید این ماتریس معکوس شود پس با توجه به تاثیر زیاد معکوس‌کردن ماتریس‌ها در تحلیل، قسمت‌هایی از تحلیل که نیاز به معکوس دارد را مشخص می‌کنیم که در بدست‌آوردن پاسخ گره‌های مرزی ابتدا به معکوس تمام ماتریس‌های سختی داخلی پاره‌سازه‌ها و سپس به معکوس ماتریس سختی بدست‌آمده برای تمام پاره‌سازه‌ها نیاز خواهد شد. بعد از این کار برای بدست‌آوردن پاسخ گره‌های داخلی هر پاره‌سازه به معکوس ماتریس‌های داخلی هر پاره‌سازه نیاز خواهد شد که همان K_{ee} می‌باشد.

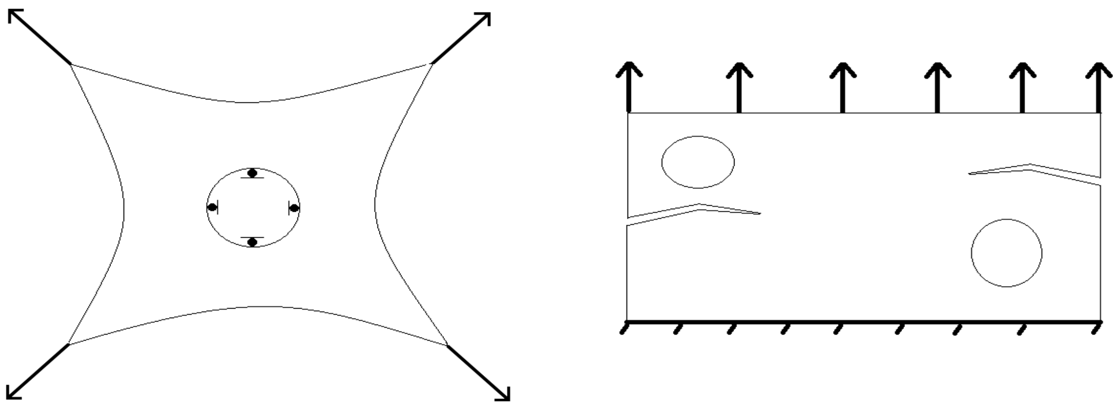
فلوچارت تشکیل جمعیت اولیه:



فلوچارت ژنتیک اعمالی بر روی جمعیت اولیه:



به کمک این روش که در فلوچارت‌های صفحات قبل توضیح داده شده است دو مثال زیر را تحلیل و بررسی نموده و خروجی‌های آن را به نمایش می‌گذاریم. مثال اول یک غشای سوراخ‌دار می باشد که از چهار نقطه خارج آن، آن را تحت کشش قرار داده‌ایم. این مثال ۶۸۷ گره و ۱۰۶۹ المان دارد و ۴ گره نیز جزو گره‌های تکیه‌گاهی می‌باشند که این مثال را با دو پاره‌سازه تحلیل خواهیم نمود. مثال دوم یک ورق سوراخ‌دار می‌باشد که دچار ترک نیز شده است و دارای ۱۰۶۱ گره و ۱۷۷۲ المان می‌باشد و ۹ گره جزو گره‌های تکیه‌گاهی می‌باشندکه این مثال را با ۵ پاره‌سازه تحلیل خواهیم نمود.



شکل (۱): صورت مثال‌های مورد تحلیل با الگوریتم ارائه شده

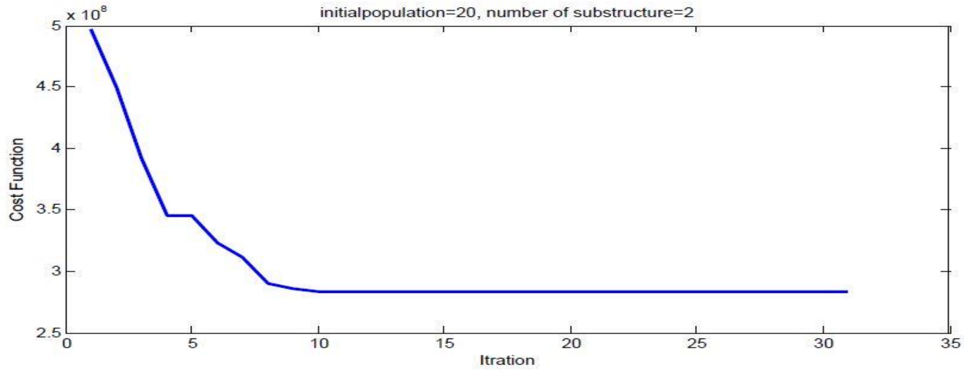
خروجی مثال غشای سوراخ‌دار بالا در جدول (۱) و خروجی مثال ورق سوراخ‌دار در جدول (۲) نشان داده شده است که بوسیله‌ی فلوچارت‌های ارائه شده در صفحات بعد تحلیل شده‌اند و سپس نمودار تغییرات تابع هزینه و در آخر نیز تصویر نحوه‌ی تقسیم المان‌ها بین پاره‌سازه‌ها ارائه شده‌است.

جدول(۱): خروجی مثال غشای سوراخ‌دار

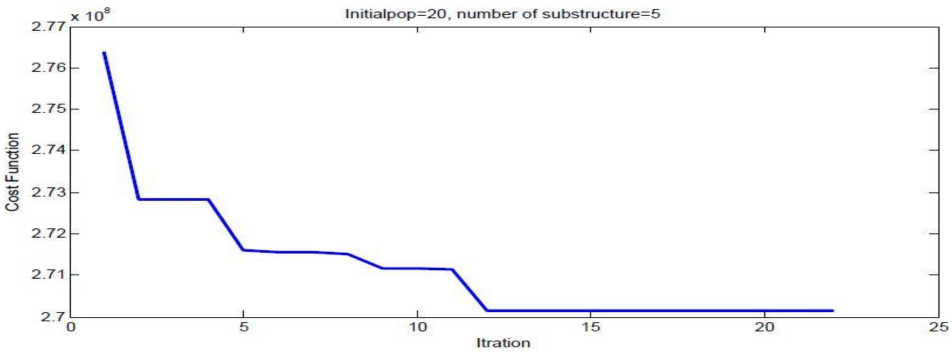
غشای سوراخ‌دار						
تعداد گره‌ها=۶۸۷ تعداد المان‌ها=۱۰۶۹ تعداد گره‌های تکیه‌گاهی=۴		پاره‌سازه اول	پاره‌سازه دوم	گره مرزی	Itration	Cost Function
ورودی	گره مرزی	۳۳۲	۳۳۳	۱۸	۳۱	۴۹۷۲۱۹۸۹۶
	تعداد المان	۵۴۷	۵۴۹	–		
خروجی	گره مرزی	۲۲۸	۲۲۷	۲۲۸		۲۸۳۲۱۴۲۹۶
	تعداد المان	۵۴۳	۵۵۳	–		

جدول(۲): خروجی مثال ورق سوراخ‌دار

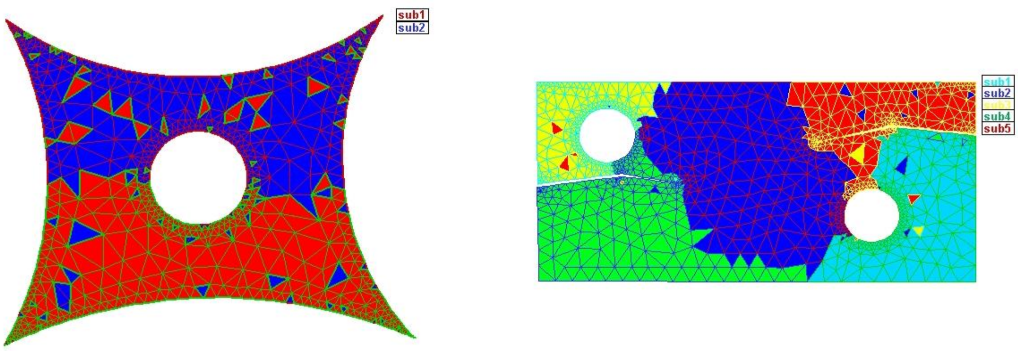
ورق سوراخ دار									
تعداد گره ها=۱۰۶۱ تعداد المان ها=۱۷۷۲ تعداد گره های تکیه گاهی=۹		پاره‌سازه اول	پاره‌سازه دوم	پاره‌سازه سوم	پاره‌سازه چهارم	پاره‌سازه پنجم	گره مرزی	Itration	Cost Function
ورودی	گره مرزی	۱۶۷	۱۹۹	۱۷۹	۲۲۲	۱۸۶	۹۹	۲۲	۲۷۶۳۸۸۲۹۶
	تعداد المان	۳۱۳	۳۳۵	۳۵۹	۳۹۶	۳۶۹	–		
خروجی	گره مرزی	۱۴۹	۱۷۹	۱۷۸	۱۸۴	۱۸۲	۱۸۰		۲۷۰۱۵۹۱۳۶
	تعداد المان	۳۵۲	۳۴۶	۳۴۳	۳۵۷	۳۷۴	–		



شکل (۲): نمودار مقدار تغییرات تابع هزینه برای مثال غشای سوراخ‌دار



شکل (۳): نمودار مقدار تغییرات تابع هزینه برای مثال ورق سوراخ‌دار



شکل (٤): نمایش تقسیم المان‌ها در بین پاره‌سازه‌ها با الگوریتم ارائه شده در این مقاله

۴. نتیجه‌گیری

با توجه به اینکه در چرخه‌ی ژنتیک اعمالی بر روی جمعیت اولیه و همچنین با توجه به اینکه المان‌ها و پاره‌سازه‌ها برای انتقال و جابجایی به صورت تصادفی انتخاب می‌شوند تا هیچ اجباری برای تشکیل پاره‌سازه‌ها نباشد و الگوریتم خود، به تقسیم بندی پاره سازه ها برسد، همانطورکه پیش بینی می‌شد تعداد گره‌های مرزی در اغلب گام‌ها زیاد می‌شود، پس با توجه به این موضوع، نحوه تولید جمعیت به سمتی سوق داده می‌شود که تعداد گره‌های مرزی در ابتدای مسیر در حالت کمترین مقدار خود قرار گیرد و سپس با وارد شدن به الگوریتم و اعمال چرخه‌ی ژنتیک تعداد گره‌های داخلی و مرزی بین پاره‌سازه‌ها به تعادل نسبی برسد تا کمترین حجم ماتریس‌های سختی را داشته باشد.

۶. مراجع

1. Przemieniecki J (1963). Matrix structural analysis of substructure. *AIAA Journal*. 1. pp: 138-147.
2. Pothen A. Simon H. Liou P (1990). Partitioning sparse matrices with eigenvectors of graphs. *SIAM J. Matrix Analytical Application*. 11. pp: 577-684.
3. Simon H (1991). Partitioning of unstructured problems for parallel processing. *computing systems in engineering*.2. pp:135-148.
4. George A. Liu J (1981). Computer Solution of Large Sparse Positive Definite Systems. Prentice-Hall, Englewood Cliffs, New Jersey.
5. Saad Y (2003). Iterative Methods for Sparse Linear Systems. second edition. the Society for Industrial and Applied Mathematics (SIAM), PWS publishing, boston.
6. Gilbert J. Miller G. Teng S (1994).. Geometric Mesh Partitioning: Implementation and Experiment. *Technical Report, Xerox Alto Research Center*.
7. Shahab U. Masroor H. Suleman M. Tareq M. Khalid J. Muhammad A. Habibullah J . (2017). Mesh partitioning and efficient equation solving techniques by distributed finite element methods: A Survey. *Archives of Computational Methods in Engineering*. (in press).
8. Fuda M. Jin-Kao H. Yang W (2017). An effective iterated tabu search for the maximum bisection problem. *Computers and Operations Research*. 81.pp: 78-89.
9. Fuda M (2016). Multiple Operator Metaheuristics for Graph Partitioning Problems. Computational Complexity, Universite d'Anger, English.
10. Logan, D. (2012) "Finite Element Method", Fifth Edition, Global Engineering: Chiristopher M. short.