الگوریتم چینش بهینه پره‌های توربین با در نظر گرفتن نامیزانی اولیه دیسک

سید هادی قادری[۱*]، احسان حاجی اسماعیلی[۲]

اطلاعات مقاله

دریافت مقاله: ۱۳۹۲/۰۱/۱۴
پذیرش مقاله: ۱۳۹۳/۰۸/۲۴

واژگان کلیدی:

چینش بهینه پره‌های توربین،
نامیزانی اولیه دیسک،
روش ابتکاری،
گروه‌بندی اعداد،
بهینه‌سازی.

چکیده

در توربین‌ها، وجود نامیزانی در روتور می‌تواند سبب کاهش بازدهی، فرسایش زودهنگام قطعات و ایجاد خسارات شود. نابالانس بودن روتور ممکن است ناشی از اختلاف جرم و مکان مرکز جرم پره‌ها نسبت به حالت ایده‌آل، نامیزانی خود دیسک و همچنین ناشی از تفاوت جرم و توزیع غیریکنواخت سایر قطعات سوار بر روتور نظیر شیم‌ها باشد. پیش از بالانس دینامیکی روتور، ابتدا پره‌ها و سایر قطعات به گونه‌ای چیده می‌شوند که مجموعه دیسک و قطعات سوار بر آن کمترین نامیزانی استاتیکی ممکن را داشته باشند. بدین منظور، ابتدا ممان پره‌ها و سایر قطعات اندازه‌گیری شده و چینشی تعیین می‌گردد که کمترین نامیزانی هر ردیف از روتور را در پی داشته باشد. الگوریتم‌هایی برای یافتن چینشی از پره‌ها که نامیزانی کمی داشته باشند پیشنهاد شده است. با این حال چون در این الگوریتم‌ها وجود نامیزانی اولیه خود دیسک و تفاوت جرم و توزیع غیریکنواخت سایر قطعات نظیر شیم‌ها در نظر گرفته نمی‌شود ممکن است مجموعه روتور همچنان نامیزانی قابل توجهی داشته باشد. در این مقاله روشی ارائه شده است که در آن چینشی برای پره‌ها و شیم‌ها (در صورت وجود) به دست می‌آید که نامیزانی دیسک را نیز خنثی کرده و سبب می‌شود که هر ردیف از روتور نامیزانی بسیار کمی داشته باشد. نتایج این الگوریتم با دیگر روش‌های موجود و همچنین با دو مورد پره‌چینی دو شرکت معتبر مقایسه شده است. نامیزانی چینش به دست آمده از الگوریتم ارائه شده ۱۰ برابر کمتر از کمترین نامیزانی قابل دستیابی با روش‌های دیگر بوده، برای دو مورد مذکور تا حد زیادی بهبود یافته است.

۱- مقدمه

میزان بالانس بودن روتور یکی از موثرترین عوامل در تعیین طول عمر و اعتمادپذیری توربین است. وجود نامیزانی در روتور می تواند سبب کاهش بازدهی توربین، افزایش تنش‌ها و خستگی در روتور و یاتاقان‌ها و در نهایت خراب شدن توربین شود. نابالانس بودن روتور ممکن است ناشی از تفاوت جرم و فاصله مرکز جرم پره‌ها تا محور روتور، نامیزانی خود دیسک و همچنین ناشی از تفاوت جرم و توزیع غیریکنواخت سایر قطعات سوار بر روتور نظیر شیم‌ها باشد. در صنعت ساخت توربین، هر پره با انجام مراحل مختلف مانند ریخته‌گری دقیق و ماشینکاری تولید شده و تعداد زیادی از این پره‌ها بر روی روتور سوار می شوند. در فرآیند ساخت پره‌ها تفاوت در جرم و توزیع جرم آنها اجتناب‌ناپذیر می‌باشد و از آنجا که هر روتور دارای چندین ردیف پره بوده و در هر ردیف تعداد زیادی پره وجود دارد این تفاوت جرم و توزیع جرم آنها نامیزانی قابل توجهی را بوجود خواهد

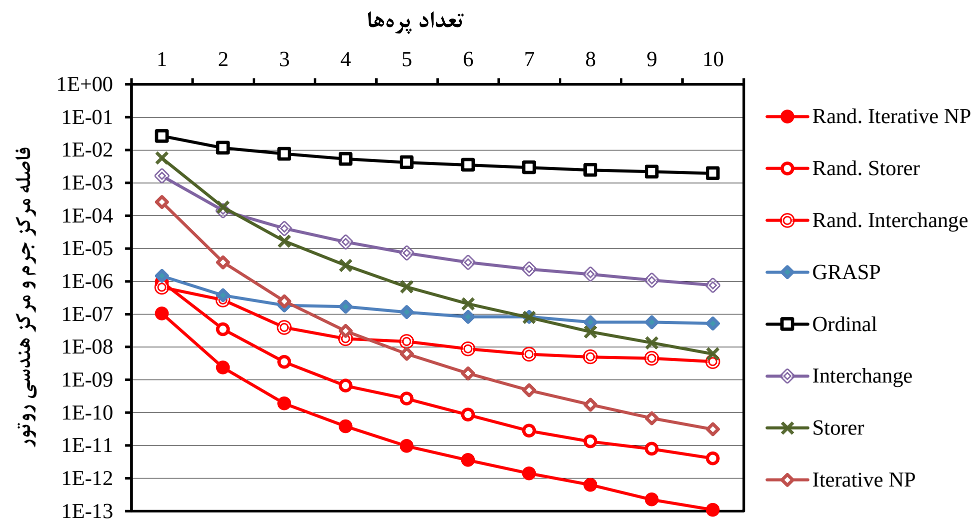
* پست الکترونیک نویسنده مسئول: s.h.ghaderi@shahroodut.ac.ir
۱. استادیار، دانشکده مهندسی مکانیک، دانشگاه صنعتی شاهرود
۲. دانشجوی کارشناسی ارشد، دانشکده مهندسی مکانیک، دانشگاه فردوسی مشهد

آورد. همچنین در نگهداری توربین لازم است توربین و پره‌های سوار بر آن به صورت دوره‌ای بازرسی شده و در صورت نیاز تعمیر و یا تعویض گردند. با این کار ممان جرمی (جرم × فاصله مرکز جرم پره تا محور روتور) پره‌ها عوض شده و بالانس مجدد روتور ضروری خواهد بود. علاوه بر پره‌ها قطعات دیگری نیز بر روی روتور سوار می‌شوند که توزیع غیریکنواخت آنها نیز می‌تواند سبب بوجود آمدن نامیزانی در روتور شود.

در فرآیند بالانس کردن روتور ابتدا پره‌ها و سایر قطعات به گونه‌ای چیده می‌شوند که مجموعه دیسک و قطعات سوار بر آن در هر ردیف کمترین نامیزانی ممکن را داشته باشد (بالانس استاتیکی.) سپس وزنه‌هایی درون روتور جای داده می شوند تا این نامیزانی باقیمانده را نیز از بین ببرند (بالانس دینامیکی.) برای یافتن چینش مناسب پره‌ها و سایر قطعات سوار بر روتور می توان از روش‌های عملی و کارگاهی و یا از الگوریتم‌ها و برنامه‌های رایانه‌ای استفاده نمود. با توجه به این که روش‌های عملی و کارگاهی زمان‌بر بوده و نتایج به دست آمده در این روش‌ها نیز با بهترین جواب ممکن فاصله زیادی دارد اگر ابتدا ممان پره‌ها و سایر قطعات اندازه‌گیری شده و با استفاده از الگوریتم‌ها و برنامه‌های رایانه‌ای چینشی تعیین گردد که کمترین نامیزانی مجموعه روتور را در پی داشته باشد زمان و هزینه بالانس کردن پره‌ها به شدت کاهش خواهد یافت [۱].

یک روش رایانه‌ای برای یافتن چینش مناسب پره‌ها و قطعات می تواند امتحان کردن تمام چینش‌های ممکن و انتخاب بهترین جواب به دست آمده باشد. با وجود اینکه در این روش کمترین نامیزانی حاصل می‌شود اما زمان اجرا الگوریتم آن قدر طولانی است که حتی برای تعداد کم قطعات نیز در عمل غیر ممکن می باشد. از این رو روش‌های دیگری پیشنهاد شده است که با نام روش‌های ابتکاری (Heuristic) مبتنی بر تجربه شناخته می شوند.

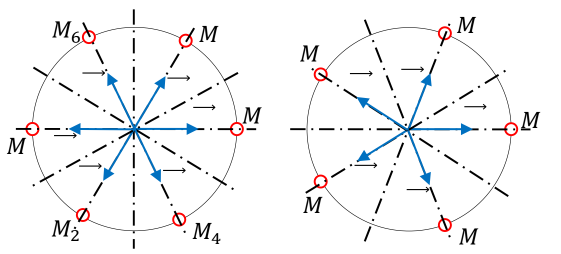
لاپورته و مرکیور [۲]، فتحی [۳] و پیتسولیز [۴] مساله یافتن چینش مناسب پره‌ها را به صورت مساله گمارش مرتبه دو (Quadratic assignment problem) مدل کرده ، هر کدام از روش ابتکاری خاصی برای حل این مدل استفاده نمودند. آمیونی [۵] روشی را ارائه کرد که در آن ابتدا سنگین‌ترین پره با دومین پره سنگین، یک جفت پره را تشکیل داده که بر روی روتور روبروی هم قرار می‌گیرند. به همین ترتیب هر پره با پره سبک‌تر از خود به صورت جفت پره در نظر گرفته می شوند. سپس چینشی برای این جفت پره‌ها بیان می شود.



شکل ۱- مقایسه روش‌های موجود برای مساله پره‌چینی بهینه. در این نمودار جرم پره‌ها اعداد تصادفی با میانگین ۱۰۰ و انحراف معیار ۱,۶۶ بوده که از توزیع نرمال پیروی می‌کنند و شعاع روتور برابر ۱۰۰ در نظر گرفته شده است.

استورر [۶] نیز روشی را ارائه کرد که در آن چینش اولیه تصادفی از پره‌ها در نظر گرفته شده و پره‌ها با استفاده از یک روش ابتکاری حول یکی از محورهای تقارن مقطع روتور جابجا می شوند تا نامیزانی کمی حاصل شود. در نهایت با جابجا کردن پره‌ها حول محور تقارنی عمود بر محور قبلی روند کاهش نامیزانی ادامه می یابد. استورر [۷] همچنین روش دیگری را ارائه کرد که در آن جابجایی حول محوری عمود بر بردار نامیزانی انجام شده، در نتیجه سرعت بیشتری نسبت به روش قبلی دارد. همچنین این الگوریتم جواب‌های نهایی بهتری را نیز نتیجه می دهد. این الگوریتم گروه‌بندی مکرر اعداد (Iterative number-partitioning heuristic) نام دارد. از آنجایی که در این دو روش جواب نهایی وابسته به چینش اولیه پره‌هایی است که الگوریتم بر روی آن اعمال شده است با اجرای این روش با چینش‌های اولیه مختلف برای مدت زمان مشخص و انتخاب کمترین نامیزانی به دست آمده جواب نهایی بهبود خواهد یافت. این الگوریتم حالت تصادفی گروه‌بندی مکرر اعداد (randomized iterative number-partitioning heuristic) نام دارد. شکل مقایسه‌ای بین نامیزانی چینش به دست آمده از روش‌های نام برده را نشان می دهد. در این نمودار جرم پره‌ها اعداد تصادفی با میانگین ۱۰۰ و انحراف معیار ۱٫۶۶ بوده که از توزیع نرمال پیروی می‌کنند. شعاع روتور برابر ۱۰۰ انتخاب شده و زمان اجرا حالت تصادفی الگوریتم‌ها (چهار منحنی پایین‌تر) برابر ۱۰ دقیقه در نظر گرفته شده است. در این نمودار برای به دست آوردن هر کدام از نقاط حالت تصادفی الگوریتم‌ها ۳۰ مجموعه و برای سایر الگوریتم‌ها ۱۰۰۰ مجموعه پره مختلف مورد آزمایش قرار گرفته و میانگین آنها نشان داده شده است. در این نمودار محور افقی تعداد پره‌ها و محور عمودی فاصله مرکز جرم تا مرکز هندسی روتور را نشان می‌دهد.

همانطور که در نمودار شکل مشاهده می‌شود، شکل تصادفی گروه‌بندی مکرر اعداد چینشی از پره‌ها را نتیجه می‌دهد که نسبت به روش‌های دیگر نامیزانی کمتری دارد. با این حال در این الگوریتم به دلیل در نظر نگرفتن نامیزانی خود دیسک و تفاوت جرم و توزیع غیریکنواخت سایر قطعات نظیر شیم‌ها، نامیزانی روتور ممکن است با کمترین نامیزانی قابل دستیابی تفاوت زیادی داشته باشد. در این مقاله روشی ارائه شده است که در آن چینشی برای پره‌ها و شیم‌ها به دست می‌آید که نامیزانی دیسک را نیز خنثی کرده، سبب می شود که هر ردیف از روتور نامیزانی بسیار کمی داشته باشد. این الگوریتم با اعمال تغییراتی بر روی حالت تصادفی تقسیم مکرر اعداد نتیجه می شود. از این رو در ادامه ابتدا الگوریتم تقسیم مکرر اعداد و حالت تصادفی آن توضیح داده شده و سپس الگوریتمی ارائه می شود که در آن جرم شیم‌ها و نامیزانی اولیه دیسک نیز در نظر گرفته شده است.

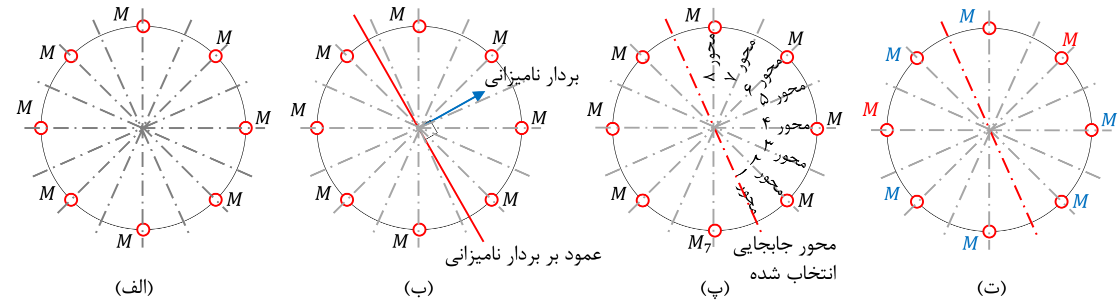


(الف) (ب)

شکل ۱- محورهای جابجایی برای روتور با (الف) تعداد پره‌های زوج و (ب) تعداد پره‌های فرد.

۲- الگوریتم کمینه کردن نامیزانی چینش پره‌ها

در هر توربین، چندین ردیف و در هر ردیف تعداد زیادی پره وجود دارد. این پره‌ها با توجه به فرآیند ساخت و همچنین به دلیل تعمیرات، دارای جرم و مرکز جرم متفاوتی می باشند. در صورتی که با ترتیب مناسبی بر روی دیسک سوار نشوند، نامیزانی قابل توجهی را به وجود خواهند آورد. از این رو لازم است در هر ردیف، پره‌ها با چینشی که تفاوت جرم یکدیگر را خنثی کرده و کمترین نامیزانی را نتیجه دهند قرار گیرند. برای n پره با ممان‌های M_i می توان این مساله را به صورت زیر بیان نمود.



شکل ۲- اجرای الگوریتم کمینه کردن نامیزانی برای توربینی با ۸ پره. (الف) چینش تصادفی اولیه. (ب) محاسبه بردار نامیزانی و رسم خط عمود بر آن. (ت) انتخاب محور جابجایی مناسب. (ث) جابجا کردن پره‌های روبرو به هم بر اساس جواب به دست آمده از مساله تقسیم اعداد.

$$\min \left| \sum_{i=۱}^{n} \sum_{j=۱}^{n} M_i \vec{e_j} x_{i,j} \right| \tag{۱}$$

$$x_{i,j} = \begin{cases} ۱ & \text{if blade } i \text{ is assigned to location } j \\ ۰ & \text{otherwise} \end{cases}$$

که در آن $\vec{e_j}$ همانگونه که در شکل ۱ نشان داده شده بردار یکه ای است که از مرکز هندسی روتور به سمت j امین جایگاه قرار گرفتن پره‌ها نشانه می رود. ماتریس $x_{i,j}$ جواب مساله و بیانگر آن است که کدام پره باید در جایگاه j ام قرار گیرد. در ادامه الگوریتمی برای یافتن چنین چینشی از پره‌ها ارائه شده است.

شکل ۱ مقطع شماتیک یک روتور به همراه پره‌های سوار بر آن را نشان می‌دهد. خطوط نشان داده شده در شکل که از روی پره‌ها و یا از بین آنها عبور می‌کنند را محورهای جابجایی نام‌گذاری می‌کنیم. در الگوریتم ارائه شده ابتدا چینش تصادفی در نظر گرفته شده و محور جابجایی مناسب انتخاب می‌گردد. سپس پره‌هایی که نسبت به این محور روبروی هم قرار می‌گیرند با استفاده از یک روش ابتکاری با یکدیگر جابجا می‌شوند تا مقدار نامیزانی کمی حاصل شود. به بیان دقیق‌تر در این الگوریتم:

۱- چینش تصادفی اولیه‌ای همانند شکل ۲-(الف) در نظر گرفته می‌شود.

۲- بردار نامیزانی این چینش محاسبه شده و از مرکز روتور خطی عمود بر این بردار رسم می گردد، شکل ۲-(ب). بردار نامیزانی برابر است با $\sum_{i=۱}^{n} M_i \vec{e_j}$.

۳- نزدیک‌ترین محور جابجایی به خط ترسیم شده انتخاب می‌شود. به این ترتیب می‌توان با جابجا کردن پره‌ها حول این محور، نامیزانی را بیشتر کاهش داده و در نتیجه سریعتر به جواب رسید، شکل ۲-(پ).

۴- نا میزانی جفت پره‌هایی که نسبت به این محور روبروی هم قرار می گیرند محاسبه می‌شود. این نامیزانی برای پره‌های M_i و M_j که نسبت به محور جابجایی روبروی هم قرار گرفته اند و با محور جابجایی زاویه θ دارند برابر است با $d_k = |M_i - M_j| \sin\theta$.

برای مثال نشان داده شده در شکل ۲ نامیزانی جفت پره‌هایی روبرو به هم عبارت است از:

$$d_۱ = |M_۲ - M_۵| \sin\frac{\pi}{۸}$$
$$d_۲ = |M_۳ - M_۸| \sin\frac{۳\pi}{۸}$$
$$d_۳ = |M_۱ - M_۴| \sin\frac{۳\pi}{۸}$$
$$d_۴ = |M_۶ - M_۷| \sin\frac{\pi}{۸}$$

۵- با استفاده از یک روش ابتکاری نامیزانی جفت پره‌های روبرو به هم، d_k ها، به گونه‌ای به دو دسته تقسیم می شوند که مجموع ممان‌های یک گروه کمترین اختلاف را با مجموع ممان‌های گروه دیگر داشته باشد. این مساله که به مساله تقسیم اعداد (Number partitioning problem) مشهور است در بخش بعد توضیح داده شده است. پره‌های مربوط به یک گروه ثابت نگه داشته شده و در گروه دیگر پره‌های روبرو به هم با یکدیگر جابجا می شوند. به این ترتیب نامیزانی چینش به دست آمده حول

محور انتخاب شده برابر با تفاضل مجموع دو گروه عددی می باشد که در مرحله ۵ به‌دست آمد. به عنوان مثال اگر در مرحله ۵ نامیزانی جفت پره‌های روبروی هم به دو گروه $\{d_۲\}$ و $\{d_۱, d_۳, d_۴\}$ تقسیم شده باشند جابجایی پره‌ها مطابق شکل ۲-(ت) صورت خواهد گرفت.

۶- برای بهبود بالانس، با این چینش جدید مراحل ۲ تا ۶ مجددا تکرار می گردند و این چرخه تا زمانی که چینش جدید پره‌ها در چرخه‌های قبل حاصل نشده باشد ادامه پیدا می‌کند. اگر چینش به دست آمده، پیش از این در چرخه‌های قبل به دست آمده باشد، ادامه اجرای الگوریتم تنها چینش‌های به دست آمده در چرخه‌های قبل را تکرار خواهد کرد و چرخه بی پایانی از جواب‌های تکراری حاصل خواهد شد.

۷- در صورت رسیدن به یک چینش تکراری، در ادامه محورهای جابجایی دیگر به ترتیب نزدیکی به محور جابجایی عمود بر بردار نامیزانی، انتخاب شده و با اجرای مراحل ۴ تا ۶ بالانس حول این محورها انجام می‌شود. به عنوان مثال اگر آخرین چینش، یعنی چینش تکراری، مطابق شکل ۲-(پ) باشد محورهای جابجایی ۲، ۸، ۳، ۷، ۴، ۶ و ۵ به ترتیب انتخاب شده و برای هر محور مراحل ۴ تا ۶ اعمال می شود.

۸- اگر طی این مراحل چینش جدیدی حاصل شد با شروع از مرحله ۲ اجرا الگوریتم ادامه خواهد یافت. اما در صورتی که چینش جدیدی حاصل نشود اجرا برنامه خاتمه یافته و کمترین نامیزانی به دست آمده در طول اجرای برنامه به خروجی الگوریتم داده می‌شود.

در این روش جواب نهایی به دست آمده وابسته به چینش اولیه‌ای است که الگوریتم بر روی آن اعمال شده است. می‌توان این الگوریتم را برای مدت زمان مشخصی با چینش‌های اولیه تصادفی مختلف اجرا کرده و کوچکترین جواب به دست آمده را به عنوان جواب نهایی در نظر گرفت. این الگوریتم حالت تصادفی گروه‌بندی مکرر اعداد نام دارد.

۳- مساله تقسیم اعداد

همانطور که گفته شد در مرحله ۵ از الگوریتم چینش پره‌ها اختلاف ممان جفت پره‌ها به گونه‌ای به دو گروه تقسیم می‌شوند که مجموع اعداد یک گروه کمترین تفاضل را با مجموع اعداد گروه دیگر داشته باشد. اگر اختلاف ممان‌ها را با d_i نشان دهیم این مساله را می‌توان با مدل ریاضی زیر بیان نمود.

$$\min\left|\sum_{i\in s} d_i - \sum_{i\in s'} d_i\right| \qquad (۲)$$

که در آن گروه‌های s و s' جواب‌های مساله هستند. برای یافتن بهترین جواب این مساله که به مساله تقسیم اعداد مشهور است تاکنون روشی جز امتحان تمام حالات پیشنهاد نشده است. از سوی دیگر امتحان تمام این حالات برای مسائلی با ابعاد متوسط نیز بسیار زمانبر بوده و در عمل امکان‌پذیر نمی‌باشد. به عنوان مثال برای ۶۰ پره مجموعه اختلاف ممان پره‌ها دارای ۳۰ عضو بوده که برای آن $۲^{۲۹}$ حالت وجود دارد. از این رو روش‌های دیگری که با نام روش‌های ابتکاری شناخته می‌شوند پیشنهاد شده است. در این روش‌ها معمولا زمان لازم برای حل مساله بسیار کم و جواب به دست آمده نزدیک به بهترین جواب ممکن می‌باشد. رایج‌ترین روش ابتکاری برای تقسیم اعداد به دو گروه روش تفاضل اعداد می‌باشد که اولین بار توسط کارمارکار و کارپ ارائه شده [۸] و با نام KK نیز شناخته می شود. در ادامه روش تفاضل اعداد و دو روش تعمیم‌یافته آن توضیح داده شده است.

در تقسیم مجموعه $D_n\{d_۱, d_۲, \ldots, d_n\}$ اگر دو عدد مانند d_i و d_j در دو گروه متفاوت قرار داده شوند باید سایر اعداد را به گونه‌ای چید که تفاضل بوجود آمده $|d_i - d_j|$ را نیز خنثی کنند. به عبارت دیگر با قرار دادن d_i و d_j در دو گروه متفاوت مجموعه‌ی D_n با مجموعه‌ی $D_{n-۱}\{d_۱, \ldots,$ $|d_i - d_j|, \ldots, d_{i-۱}, d_{i+۱}, \ldots, d_{j-۱}, d_{j+۱}, \ldots, d_n\}$ جایگزین شده و مساله تقسیم اعداد برای این مجموعه جدید مطرح می‌گردد. تعداد اعضای این مجموعه جدید یک عضو کمتر از مجموعه اولیه است. اگر مجددا دو عضو از

مجموعه جدید انتخاب شده و در دو گروه متفاوت قرار داده شوند تعداد اعضای مجموعه باز هم یک عضو کاهش می‌یابد. می‌توان این کار را تا زمانی که تنها یک عضو در مجموعه باقی بماند ادامه داد. به این ترتیب عدد باقیمانده تفاضل دو گروه خواهد بود. در صورتی که قرار دادن d_i و d_j ها در گروه‌های متفاوت درست باشد تفاضل به دست آمده نیز کمترین تفاضل ممکن خواهد بود.

در روش KK در هر مرحله بزرگترین عدد، u، و دومین عدد بزرگ مجموعه، v، که کمترین اختلاف را با u دارد انتخاب و در دو گروه متفاوت قرار داده می‌شوند. به این ترتیب بزرگترین اعداد از مجموعه حذف و کوچکترین عدد ممکن متناظر با آن یعنی $|u-v|$ به مجموعه اضافه می‌شود. به عبارت دیگر در الگوریتم KK

۱- اعداد به ترتیب از بزرگ به کوچک چیده شده و در لیستی قرار داده می‌شوند.
۲- دو عددی که در بالای لیست قرار دارند، u و v، از مجموعه حذف شده و اختلاف آنها محاسبه می‌گردد، $|u-v|$.
۳- با شروع از پایین لیست $|u-v|$ با اعداد موجود در لیست مقایسه می‌شود تا اینکه این تفاضل کوچکتر مساوی یکی از اعداد موجود در لیست باشد. $|u-v|$ را در زیر این عدد قرار می‌دهیم.

مراحل ۲ و ۳ تا زمانی که تعداد اعضای مجموعه به یک عضو کاهش یابد تکرار خواهند شد. این عضو باقیمانده اختلاف مجموع اعداد دو گروه خواهد بود.

به عنوان مثال در تقسیم مجموعه $\{۴,۶,۵,۷,۸\}$ به دو گروه ابتدا اعداد به ترتیب از بزرگ به کوچک قرار داده می‌شوند، $(۸,۷,۶,۵,۴)$. سپس دو عدد ابتدای لیست انتخاب و با تفاضل آنها، $۸-۷=۱$، جایگزین می‌گردند . این کار آنقدر تکرار می‌شود تا تنها یک عضو در مجموعه باقی بماند. یعنی:

$$(۸,۷,۶,۵,۴) \xrightarrow{۸-۷=۱} (۶,۵,۴,۸-۷)$$
$$\xrightarrow{۶-۵=۱} (۴,۸-۷,۶-۵)$$
$$\xrightarrow{۴-۱=۳} (۴-۸+۷,۶-۵)$$
$$\xrightarrow{۳-۱=۲} (۴-۸+۷-۶+۵)$$

عدد باقیمانده، یعنی ۲، اختلاف مجموع اعداد گروه $\{۴,۷,۵\}$ و $\{۸,۶\}$ است.

شکل تصادفی این الگوریتم که توسط استورر [۶] پیشنهاد شد، می‌تواند باعث بهبود جواب نهایی الگوریتم شود. در این نسخه تعمیم یافته ترتیب بزرگ به کوچک بودن اعداد در لیست اولیه تا حدودی به هم خورده، مراحل ۲ و ۳ از الگوریتم KK بر روی لیست جدید اعمال می‌شود. این بر هم زدن تصادفی ترتیب اعداد و اعمال مراحل ۲ و ۳ الگوریتم KK بارها تکرار می‌شود و کمترین جواب به دست آمده به خروجی الگوریتم داده می‌شود. به بیان دقیق‌تر در این الگوریتم برای تقسیم اعداد a_i به دو گروه

۱- اعداد b_i به صورت $b_i = a_i + RND(\theta)$ تعریف می‌شود که در آن $RND(\theta)$ عددی تصادفی بین ۰ و θ می باشد. θ را پارامتر بهبود نام‌گذاری می‌کنیم.
۲- اعداد را مطابق ترتیب بزرگ به کوچکی آنها در لیستی قرار می‌دهیم. بنابراین لیست شامل اعدادی بوده که ترتیب آنها به صورت تصادفی و متناسب با پارامتر بهبود به هم خورده است.
۳- مراحل ۲ و ۳ از الگوریتم KK بر روی لیست مرحله ۲ اعمال می‌شود تا اینکه تنها یک عضو در لیست باقی بماند.

مراحل ۱، ۲ و ۳ را N بار تکرار کرده و کمترین جواب به‌دست آمده در این تکرارها را به عنوان جواب نهایی الگوریتم در نظر می‌گیریم.

در این الگوریتم دفعات تکرار مناسب وابسته به اندازه پارامتر بهبود می‌باشد. برای یک θ مشخص اگر N کوچک انتخاب شود بسیاری از ترتیب‌هایی که به وسیله این پارامتر بهبود می‌توانند بوجود آیند، امتحان نشده و ممکن است کاهشی در جواب نهایی حاصل نشود. در صورتی که پارامتر بهبود بزرگ و تعداد دفعات تکرار کوچک انتخاب شود، حتی ممکن است تفاضل مجموع دو گروه افزایش یابد. همچنین اگر N بزرگ باشد برنامه بارها لیست اولیه یکسانی را مورد آزمایش قرار داده، زمان اجرا برنامه بدون اینکه جواب بهتری حاصل شود افزایش می‌یابد.

استورور [۶] حالت ساده شده‌ای از این الگوریتم را پیشنهاد می‌کند که در آن تنها ترتیب ۲۰ عدد بالای لیست به هم خورده و متناسب با این تعداد، N برابر ۱۰۰۰ در نظر گرفته شده است. به بیان دقیق‌تر برای تقسیم اعداد a_i به دو گروه

۱- اعداد b_i به صورت $b_i = a_i + RND(\theta)$ برای ۲۰ عدد بزرگتر a_i و به صورت $b_i = a_i$ برای سایر a_iها تعریف می‌شود. a_iها مطابق ترتیب بزرگ به کوچکی b_iها در لیستی قرار داده می‌شوند.

۲- مراحل ۲ و ۳ از الگوریتم KK بر روی لیست مرحله ۱ اعمال شده و آنقدر تکرار می‌گردد که تنها یک عضو در لیست باقی بماند.

مراحل ۱ و ۲ را ۱۰۰۰ بار تکرار نموده و کمترین جواب را به عنوان جواب نهایی الگوریتم در نظر می‌گیریم.

برای به دست آوردن پارامتر θ مناسب مجموعه‌هایی با تعداد عضو ۱۰، ۲۰، ۳۰، ۴۰، ۶۰، ۸۰ و ۱۰۰ عدد تصادفی با توزیع نرمال بین ۰ و ۹۹۹،۹۹۹،۹۹۹ مورد آزمایش قرار گرفتند که نتایج آن در شکل ۳ نشان داده شده است. برای به‌دست آوردن هر کدام از نقاط این نمودار ۱۰۰ مجموعه مورد آزمایش قرار گرفته و میانگین جواب‌های به دست آمده بر روی نمودار نشان داده شده است.

همانطور که در این نمودار مشاهده می شود مقدار جواب با افزایش θ تا $\theta = ۰.۴(a_{max} - \mathrm{a_{min}})$ کاهش یافته و پس از این مقدار تغییرات آن ناچیز می‌گردد. از این رو مقدار مناسب پارامتر θ برابر $۰.۴(\mathrm{a_{max}} - \mathrm{a_{min}})$ در نظر گرفته می‌شود.

۴- الگوریتم حذف نامیزانی اولیه دیسک

علاوه بر نامیزانی ناشی از تفاوت جرم و مرکز جرم پره‌ها خود دیسک نیز می‌تواند دارای نامیزانی اولیه‌ای باشد. در این بخش الگوریتمی برای حذف این نامیزانی اولیه، $\vec{R}$، ارائه شده است. در این الگوریتم، چینشی از پره‌ها حاصل می شود که دارای نامیزانی‌ای برابر نامیزانی اولیه دیسک و در خلاف جهت آن می‌باشد و به این ترتیب نامیزانی دیسک را خنثی می‌کند.

برای انجام این کار ابتدا به ممان یکی از پره‌ها نامیزانی اولیه R اضافه شده و الگوریتم کمینه کردن نامیزانی بر روی این مجموعه اعمال می‌شود. سپس با کم کردن مقدار اضافه شده، از ممان این پره چیدمان به دست آمده دارای نامیزانی R در محل این پره خواهد بود و در نتیجه نامیزانی مجموعه دیسک و پره‌ها کمینه خواهد شد. به بیان ریاضی همان‌گونه که در شکل ۴-(الف) نشان داده شده است، مساله

$$\min \left| \vec{R} + \sum_{i=۱}^{n} \sum_{j=۱}^{n} M_i \, \vec{e_j} x_{i,j} \right| \tag{۳}$$

با مساله

$$\begin{gathered} \min \left| \sum_{i=۱}^{n} \sum_{j=۱}^{n} M_i' \, \vec{e_j} x_{i,j} \right| \\ M_۱' = M_۱, \quad \ldots, \quad M_{i-۱}' = M_{i-۱}, \quad M_i' = M_i + R, \\ M_{i+۱}' = M_{i+۱}, \ldots, M_n' = M_n \end{gathered} \tag{۴}$$

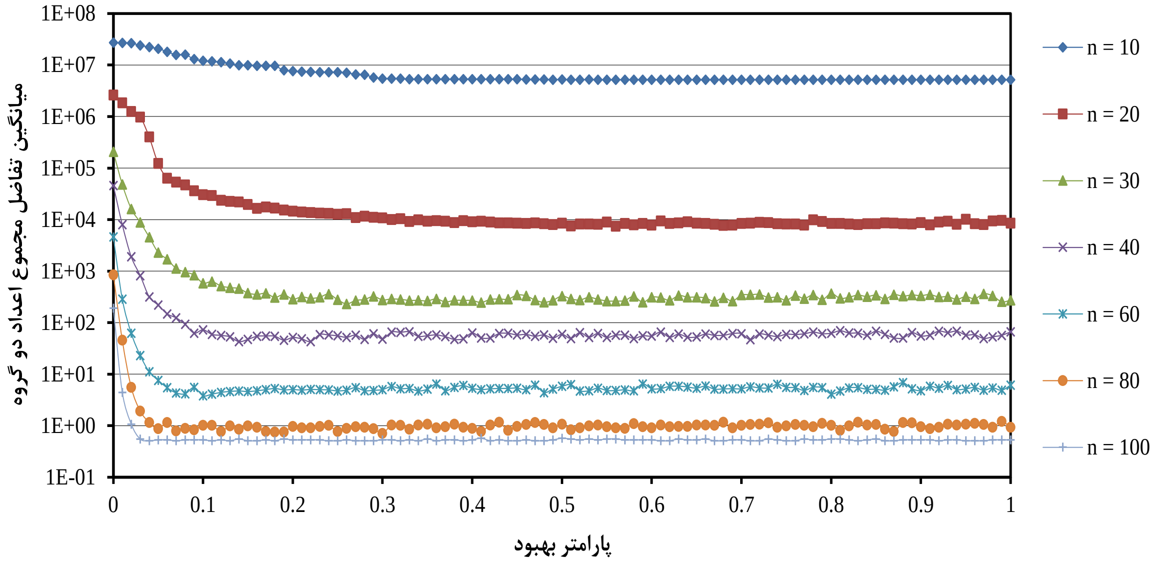
جایگزین شده و الگوریتم ارائه شده در بخش ۲ بر روی مجموعه $\{M_i'\}$ اعمال می گردد. به این ترتیب برای ممان‌های $\{M_۱, \ldots, M_{i-۱}, M_i + R, M_{i+۱}, \ldots, M_n\}$ چینشی به دست خواهد آمد که نامیزانی نزدیک به صفرداشته باشد و متناظر با آن نامیزانی حاصل از چینش پره‌ها با ممان $\{M_۱, \ldots, M_{i-۱}, M_i, M_{i+۱}, \ldots, M_n\}$ نزدیک به R خواهد بود، شکل ۵-(ب). بردار $\vec{U}$ نامیزانی ناشی از چینش پره‌ها را نشان می دهد حال می توان با حفظ چینش پره‌ها، آنها را به گونه ای بر روی دیسک جابجا کرد که نامیزانی پره‌ها در خلاف جهت نامیزانی اولیه دیسک قرار گرفته و آن را خنثی کند، شکل ۴-(پ).

به بیان دیگر برای حذف نامیزانی اولیه دیسک

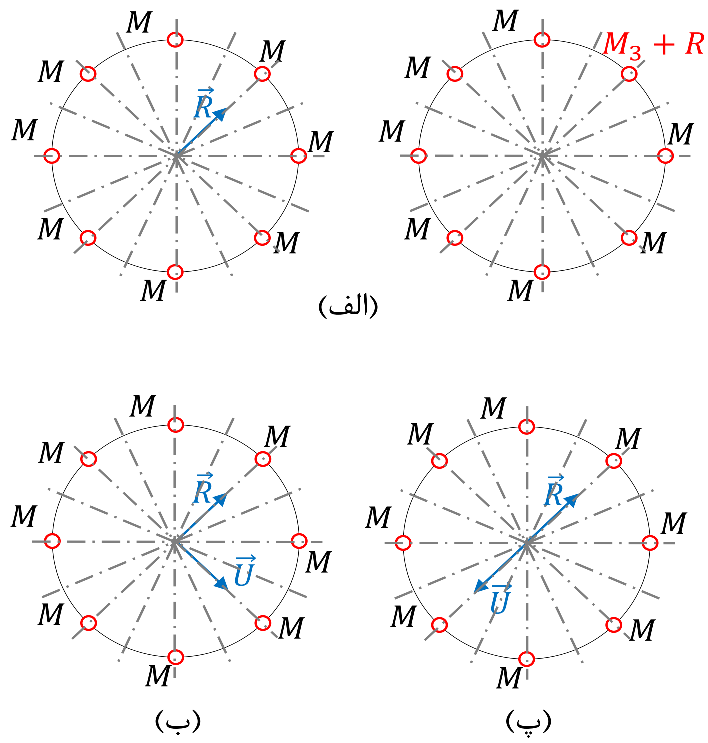
۱- مقدار نامیزانی دیسک، R، به ممان یکی از پره‌ها، M_i، اضافه می‌شود.

۲- الگوریتم کمینه کردن نامیزانی بر روی ممان‌های $\{M_۱, \ldots, M_{i-۱}, M_i + R, M_{i+۱}, \ldots, M_n\}$ اعمال می‌شود.

۳- پره‌ها با همان چینش به دست آمده بر روی دیسک می‌چرخند تا پره $\mathrm{M_i}$ در خلاف جهت بردار نامیزانی اولیه دیسک، $\vec{\mathrm{R}}$، قرار گرفته و نامیزانی دیسک با نامیزانی مجموعه پره‌ها خنثی شود.



شکل ۳- تاثیر پارامتر بهبود بر تفاضل مجموع دو گروه در مساله تقسیم اعداد.



شکل ۴- حذف نامیزانی اولیه دیسک. (الف) معادل قرار دادن مساله نامیزانی اولیه دیسک و نامیزانی پره‌ها با مساله ساده تر. (ب) یافتن چینش مناسب برای مساله ساده شده. (پ) جابجا کردن پره‌ها با حفظ چینش به دست آمده، به گونه ای که نامیزانی اولیه دیسک با نامیزانی پره‌ها خنثی شود.

به عنوان مثال برای دیسکی با ممان اولیه $R = ۰.۳۳$ در زاویه ۰٫۴۵ درجه و پره‌هایی با ممان‌های

$$M_۱ = ۱۰.۰۵, M_۲ = ۱۰.۲۵, M_۳ = ۹.۹۱, M_۴ = ۹.۹۰,$$
$$M_۵ = ۱۰.۱۱, M_۶ = ۹.۸۳, M_۷ = ۹.۹۶, M_۸ = ۱۰.۱۵$$

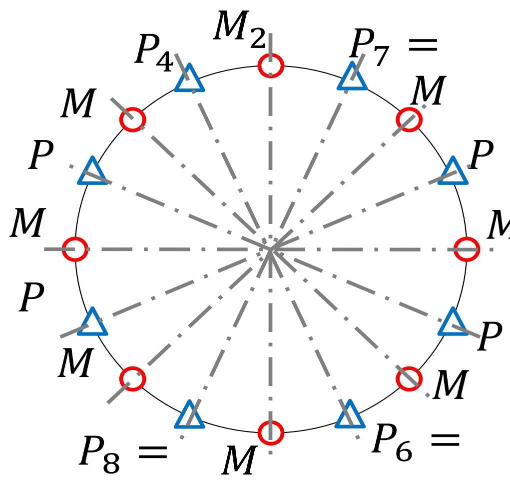
که در شکل ۴-(الف) نشان داده شده است، ابتدا مجموعه $\{M_i'\}$ را به صورت $\{۱۰٫۰۵, ۱۰٫۲۵, ۹٫۹۱+۰٫۳۳, ۹٫۹۰,$ $۱۰٫۱۱, ۹٫۸۳, ۹٫۹۶, ۱۰٫۱۵\}$ تشکیل داده و الگوریتم ارائه شده در بخش ۲ را بر روی آن اعمال می کنیم. چینش به دست آمده برای $\{M_i'\}$ به صورت زیر خواهد بود.

$$(M_۶', M_۸', M_۱', M_۳', M_۴', M_۷', M_۲', M_۵')$$

حال پره‌ها را به همان ترتیب به دست آمده برای $\{M_i'\}$ بر روی روتور قرار می دهیم، شکل ۴-(ب). به این ترتیب ممان ناشی از چینش پره‌ها برابر ۰٫۲۹ و در زاویه ‎-۴۴٫۸۲ درجه نسبت به افق خواهد بود. حال می توان مطابق شکل ۴-(پ) پره‌ها را ۹۰ درجه در جهت ساعت گرد چرخاند که در نتیجه آن بردار نامیزانی ناشی از چینش پره‌ها در خلاف جهت بردار نامیزانی اولیه دیسک قرار گرفته و یکدیگر را خنثی کنند. با این کار نامیزنی مجموعه روتور و پره‌ها برابر ۰٫۰۴ خواهد شد. در صورتی که الگوریتم بخش ۲ مستقیما بر روی ممان پره‌ها اعمال می شد نامیزانی مجموعه دیسک و پره‌ها بیش از ۷ برابر مقدار به دست آمده می بود.

۵- الگوریتم حذف نامیزانی ناشی از شیم‌ها

در بعضی از انواع پره‌ها، برای ثابت نمودن آنها در جای خود بر روی دیسک از قطعاتی به نام شیم استفاده می‌شود. تعداد این شیم‌ها معمولا کمتر از تعداد پره‌ها بوده و دارای جرم‌های متفاوت می‌باشند. این قطعات نیز اگر با ترتیب مناسبی چیده نشوند سبب بوجود آمدن نامیزانی در روتور خواهند شد. در این بخش الگوریتمی ارائه شده است که در آن چینش بهینه برای پره‌ها و شیم‌ها بطور همزمان حاصل می‌شود.



شکل ۵- حذف نامیزانی مجموعه پره‌ها و شیم‌ها.

برای این کار چینش اولیه‌ای از پره‌ها و شیم‌ها در نظر گرفته می‌شود که در آن ممان پره‌ها و شیم‌ها به صورت یک در میان قرار گرفته و الگوریتم کمینه کردن نامیزانی بر روی این چینش اعمال می‌شود. برای این حالت، در الگوریتم کمینه کردن نامیزانی محورهای جابجایی همان محورهای جابجایی در حالت کمینه کردن ممان پره‌ها می‌باشد و محور جابجایی مناسب از بین این محورها انتخاب می‌گردد. به بیان دیگر نباید محور جابجایی انتخاب کرد که از بین یک پره و یک شیم بگذرد. جابجایی حول چنین محوری سبب به هم خوردن ترتیب یک در میان بودن پره‌ها و شیم‌ها خواهد شد. در حالیکه جابجایی حول محورهایی که از روی پره‌ها یا شیم‌ها می‌گذرند ترتیب یک در میان بودن پره‌ها و شیم‌ها را به هم نمی‌زند. به بیان دقیق‌تر برای کمینه کردن نامیزانی در چینش n پره با ممان‌های $\{M_i\}$ و m شیم با ممان‌های $\{P_i\}$ به صورت زیر عمل می کنیم.

۱- چینش اولیه‌ای که در آن پره‌ها و شیم‌ها به صورت یک در میان قرار می‌گیرند در نظر گرفته می‌شود. برای انجام این کار $n - m$ شیم با ممان صفر به مجموعه شیم‌ها اضافه می‌شود تا ترتیب یک در میان بودن حفظ شود. به عنوان مثال برای ۸ پره و ۵ شیم مطابق شکل ۵ سه شیم با ممان صفر در نظر گرفته می شود.

برای مقایسه نتایج الگوریتم و برنامه ارائه شده با جواب‌های شرکت‌های معتبر نیروگاهی، این الگوریتم را بر روی پره‌های ۵ ردیف از توربین BBC نیروگاه مشهد اعمال شده و با نتایج ارائه شده توسط شرکت‌های سولزر و توربین شهریار مقایسه شده است. در شکل ۷ نتایج به دست آمده از الگوریتم ارائه شده و جواب‌های شرکت سولزر و توربین شهریار مقایسه شده است. همانطور که در این شکل مشاهده می‌شود الگوریتم ارائه شده می‌تواند چینشی با نامیزانی ای تا ۱۰^۸ برابر کوچکتر از نامیزانی چینش‌های ارائه شده توسط این شرکت‌ها را در پی داشته باشد.

۷- نتیجه‌گیری

در این مقاله الگوریتمی ارائه شد که می تواند برای تعداد بیش از ۴۰ پره نامیزانی را تا ۱۰ برابر نسبت به کمترین نامیزانی قابل دستیابی با روش‌های دیگر کاهش دهد.

همچنین در مقایسه با نامیزانی چینش ارائه شده شرکت سولزر و توربین شهریار، در چینش حاصل از الگوریتم و برنامه ارائه شده تا ۱۰^۸ برابر کاهش نامیزانی را شاهد هستیم که در عمل به معنای از بین بردن نامیزانی است. این قدرت الگوریتم در ارائه چینش بهینه پره‌ها به همراه توانایی آن برای در نظر گرفتن نامیزانی اولیه دیسک و نامیزانی سایر قطعات سوار بر روتور مانند شیم‌ها، الگوریتم ارائه شده را به الگوریتمی ایده‌آل برای برای استفاده در فرآیند بالانس کردن روتور تبدیل می‌کند.

۸- تقدیر و تشکر

با سپاس از جناب آقای دکترسید مجید هاشمیان که در تهیه اطلاعات مورد نیاز ما را یاری رساندند.

۹- مراجع

[۱] Mason, A., RFonnqvist, M. (1997). “Solution methods for the balancing of jet turbines”. Computers and Operations Research, Vol. 24, pp.153–167.

[۲] Laporte, G., Mercure, H. (1988). “Balancing hydraulic turbine runners: a quadratic assignment problem”. European Journal of Operational Research, Vol. 35, pp. 378–381.

[۳] Fathi, Y., Ginjupalli, K.K. (1993). “A mathematical model and a heuristic for the turbine balancing problem”. European Journal of Operational Research, Vol. 63, pp. 336–342.

[٤] Pitsoulis, L.S., Pardalos, P.M., Hearn, D.W. (2001). “Approximate solutions to turbine balancing problem”. European Journal of Operational Research, Vol. 130, pp. 147–155.

[٥] Amiouny, S.V., Bartholdi, J.J., Vande, J.H.V. (2000). “Heuristics for balancing turbine fans”. Operations Research, Vol. 48, pp. 591–602.

[٦] Storer, R.H., (1999). “Extensions of and uses for the di8erencing algorithm for number partitioning”. Report No. ۹۹T-۰۹, Department of Industrial and Manufacturing Systems Engineering, Lehigh University, Bethlehem, Pennsylvania.

[۷] Storer, R.H., Choi, W. (2004). “Heuristic algorithms for a turbine-blade-balancing problem”. Computers and Operations Research, Vol. 31, pp. 1245–1258.

[۸] Karmakar, N.R.M., Karp, R.M. (1982). “The differencing method for set partitioning”. Report No. UCB/CSD 82/113, Computer Science Division, University of California, Berkley.