

حل مسأله‌ی برش دوبعدی غیر گیوتینی با تقاضا با استفاده از الگوریتم بهینه‌سازی ازدحام ذرات

فائزه اسدیان اردکانی*
علی مروتی شریف‌آبادی**

چکیده

بهینه‌سازی چیدمان قطعات کاربردهای فراوانی در صنایع برش ورق فلزی، برش الوار، تولید شیشه، کاغذ و پوشاک دارد و به دلیل اهمیت کاهش ضایعات، روش‌های زیادی برای حل این مسأله ارائه شده است. یکی از بهترین روش‌ها استفاده از الگوریتم بهینه‌سازی ازدحام ذرات می‌باشد. در این پژوهش، مسأله‌ی برش دوبعدی با تقاضا مورد بررسی قرار می‌گیرد. در این مسأله باید با برش ورق‌های مستطیل شکل بزرگ، مستطیل‌های کوچک‌تر مورد نیاز به نحوی تولید شوند که ضمن تأمین تقاضای آن‌ها، ضایعات یا تعداد ورق‌های مصرفی حداقل شود. در این مقاله جهت حل این مسأله از الگوریتم بهینه‌سازی ازدحام ذرات استفاده شده است. به منظور بهبود کارایی این الگوریتم و جلوگیری از هم‌پوشانی در مسأله‌ی برش، الگوریتم ابتکاری CUL به کار گرفته شد. جهت حل مسأله‌ی فوق، نرم‌افزاری تهیه شد. این نرم‌افزار به دو حالت عمل می‌کند. در حالت اول با در نظر گرفتن طول و عرض صفحه‌ی اصلی، اندازه قطعات و تعداد مورد تقاضا، الگوی بهینه‌ی برش را ارائه می‌دهد. در حالت دوم، امکان دادن عرض‌های متفاوت به نرم‌افزار وجود دارد. در این حالت، نرم‌افزار پس از ارائه‌ی عرض بهینه، الگوی بهینه‌ی برش و طول بهینه‌ی صفحه‌ی اصلی را نیز برای کاربر مشخص می‌کند.

واژگان کلیدی: بهینه‌سازی، الگوریتم‌های فراابتکاری، الگوریتم بهینه‌سازی ازدحام ذرات گسسته، الگوریتم CUL، مسأله‌ی برش دوبعدی.

* کارشناسی ارشد مدیریت صنعتی گرایش تولید - دانشگاه یزد، ایران (نویسنده مسئول) Email: faezehasadian@gmail.com
** عضو هیئت علمی گروه مدیریت صنعتی دانشکده‌ی اقتصاد، مدیریت و حسابداری، دانشگاه یزد، ایران.

مقدمه

در فرآیند تولید بسیاری از صنایع نیاز است تا قطعات کوچک تری با برش اجسام بزرگ تر حاصل شوند و یا به طور معادل قطعات کوچک تری در یک جسم بزرگ تر جای داده شوند. در این عمل معمولاً بخش هایی از جسم بزرگ تر به قطعاتی تبدیل می شوند که قابل استفاده در هیچ یک از محصولات تولیدی نبوده و به عنوان ضایعات و دورریز محسوب می شوند. کاهش چنین ضایعاتی نقش مهمی در کاهش هزینه ها داشته و به عنوان یکی از موضوعات علم تحقیق در عملیات و تحت عنوان مسأله ی برش توجه بسیاری از محققان را در نیم قرن گذشته به خود جلب کرده است [۱].

مسأله ی برش در بسیاری از صنایع تولید انبوه مانند تولید کاغذ، تولید شیشه، برش الوار و نساجی که صفحات بزرگ باید به قطعات کوچک تر تقسیم شوند، به کار برده می شود [۲]. مسأله ی برش جزء مسائل بهینه سازی ترکیبی است و کاربردهای فراوانی در علم کامپیوتر، مهندسی صنایع، فرآیند ساخت و تولید و غیره دارد. تحقیقات زیادی در رابطه با ابزار و روش های حل این نوع مسائل در صنعت، صورت گرفته است [۳، ۴، ۵، ۶، ۷، ۸].

در سال های اخیر، محققان بسیاری به بررسی شاخه های متفاوت مسأله ی برش پرداخته اند. روش های متعددی برای بهینه سازی این مسأله توسعه داده شده است [۹، ۱۰، ۱۱، ۱۲، ۱۳، ۱۴]. اولین شکل فرمولی این نوع مسائل توسط کانتروویچ در سال ۱۹۳۹ و در صنایع کاغذسازی مطرح شد [۱۵]. وی برای مسأله ی برش یک بعدی با تقاضا، مدلی را مطرح می کند که برای حل آن ابتدا باید تمامی سبک های برش یک میله به قطعات مورد نیاز شناسایی شوند. این امر تنها برای مسائل خیلی کوچک عملی خواهد بود [۱۶].

گیلمور و گموری در سال ۱۹۶۱ با استفاده از برنامه ریزی خطی، روشی برای ایجاد الگوی برش در مسائل برش یک بعدی ارائه دادند [۱۷]. هم چنین این دو محقق مسأله ی برش دوبعدی را به عنوان مسأله ی برش یک بعدی دومرحله ای در نظر گرفتند. روش حل آن ها بر مبنای برنامه ریزی عدد صحیح با تکنیک ایجاد ستون می باشد [۱۸]. علاوه بر این دو محقق، محققان دیگری نیز از مدل های برنامه ریزی خطی عدد صحیح برای حل مسأله ی برش گیوتینی متعامد دومرحله ای نامحدود استفاده کرده اند [۱۹، ۲۰].

بیزلی [۲۱] برای یافتن یک سبک برش بهینه یا به عبارت دیگر حل مسأله‌ی برش بدون تقاضا از روش برنامه‌ریزی پویا استفاده کرده است. همین مسأله با ابعاد مختلف ورق اصلی مورد توجه هایفای [۲۲] بوده است. آگراوال [۲۳] از رویکرد انشعاب و تحدید جهت حل مسأله‌ی برش اشکال مستطیلی، زمانی که اندازه‌ی قطعات یکسان باشند، استفاده کرد. کوی نیز از الگوریتم انشعاب و تحدید و برنامه‌ریزی پویا برای حل مسأله-ی برش دوبعدی با قطعات مستطیل شکل هم‌اندازه استفاده کرده است [۲۴، ۲۵]. چانگ و همکاران [۲۶] به بررسی ادبیات مسأله‌ی برش (به ویژه دوبعدی)، پرداخته‌اند. آن‌ها به روش‌های متفاوت استفاده شده در حل این نوع مسائل و جنبه‌های عملی توجه ویژه داشتند. تحقیقاتی نیز به فعالیت‌هایی که قبل از سال ۱۹۹۰ انجام شده‌اند، تخصیص پیدا کرده‌اند [۲۷، ۲۸، ۲۹]. تعدادی از محققین به پیشینه‌ی مسأله‌ی برش دوبعدی غیرگیوتینی محدود پرداخته‌اند [۳۰، ۳۱، ۳۲، ۳۳، ۳۴].

سلیمان [۳۵] از روش ابتکاری ترتیبی سه مرحله‌ای برای حل مسأله‌ی برش دوبعدی با هدف حداقل کردن میزان برش، استفاده کرده است. از پژوهش‌های دیگری که از رویکرد ابتکاری برای حل مسأله‌ی برش دوبعدی استفاده کرده‌اند، می‌توان به [۳۶، ۳۷] اشاره کرد.

از روش‌های فراابتکاری نیز جهت حل مسأله‌ی برش استفاده شده است. لی و چان [۳۸] از روش شبیه‌سازی تبرید عناصر (SA) برای حل مسائل برش غیرگیوتینی دو یا سه‌بعدی استفاده کردند. لئونگ و همکاران [۳۹] عملکرد و کارایی الگوریتم ژنتیک (GA) و SA را در مورد مسأله‌ی برش با هم مقایسه کردند. بیزلی [۳۱، ۴۰] از روشی ابتکاری برای حل مسأله برش دوبعدی غیرگیوتینی محدود استفاده کرده است. مصلحی و رضایی [۱] از روش ابتکاری SA برای حل مسأله‌ی برش دوبعدی با تقاضا استفاده کرده‌اند. تیواری و چاکربرتی [۴۱] از الگوریتم ژنتیک برای حل مسأله‌ی برش با فرض‌های گیوتینی و غیرگیوتینی استفاده کردند. جیانگ و همکاران [۴۲] از روش الگوریتم بهینه‌سازی ازدحام ذرات (PSO) و GA برای حل مسأله‌ی برش دوبعدی غیرگیوتینی محدود استفاده کرده‌اند.

در این مقاله برای حل مسأله‌ی برش دوبعدی غیرگیوتینی با تقاضا، از الگوریتم

1- Simulated Annealing (SA)

2- Genetic Algorithm (GA)

3- Particle Swarm Optimization (PSO)

بهینه‌سازی ازدحام ذرات استفاده شده است. این روش، از جمله روش‌های جدید فراابتکاری می‌باشد که توانمندی خود را در حل مسائل مختلف مدیریتی اثبات کرده است [۴۳، ۴۴، ۴۵، ۴۶، ۴۷]. از این الگوریتم به شکل محدود در حل مسائل ساده‌ی برش نیز استفاده شده است. در این مقاله ضمن استفاده از این الگوریتم برای حل مسأله‌ی برش دوبعدی غیرگیوتینی با تقاضا برای کاهش زمان حل و جلوگیری از هم-پوشانی، از الگوریتم ابتکاری CUL استفاده شده است. هم‌چنین نرم‌افزار طراحی شده، به دو حالت برای کاربر قابل استفاده است. در ادامه به توضیح بیشتری پیرامون این موضوع پرداخته خواهد شد.

این پژوهش در پنج بخش تنظیم شده است. در بخش بعدی به تشریح مسأله‌ی مورد بررسی پرداخته و در بخش سوم در مورد الگوریتم پیشنهادی به تفصیل توضیح داده شده است. بخش چهارم این مقاله، نتایج حاصل از تجزیه و تحلیل داده‌ها را ارائه می‌دهد. در نهایت نتیجه‌گیری و پیشنهادهایی برای تحقیقات آتی در بخش پنجم ارائه شده است.

تشریح مسأله

در این پژوهش، برش‌ها غیرگیوتینی در نظر گرفته شده است. برش گیوتینی برشی است که از یک ضلع آزاد مستطیل شروع شده و موازی با یکی از اضلاع ورق اصلی، به ضلع آزاد مقابل ختم می‌شود. اما در برش غیرگیوتینی چنین فرضی وجود ندارد. در این مقاله، یک ورقه‌ی اصلی به اندازه‌ی مشخصی در نظر گرفته شده و با برش آن، تقاضای قطعات با اندازه‌های مختلف تأمین می‌شود. برای هر قطعه‌ی i ، به تعداد مشخصی تقاضا وجود دارد. هدف از حل این مسأله حداقل کردن مقدار ضایعات با توجه به تأمین تقاضاست.

برای مسأله‌ی تشریح شده، فرض‌های زیر به منظور ساده‌سازی در نظر گرفته شده است [۳۹]:

- ۱- طول و عرض هر قطعه از بعد متناظر ورقه‌ی اصلی، تجاوز نمی‌کند.
- ۲- تمام برش‌ها در ورقه‌ی اصلی بی‌نهایت باریک می‌باشند؛ یعنی لبه‌های قطعات، مساحتی را اشغال نمی‌کنند.

۳- قطعات ممکن است در هر جای ورقه‌ی اصلی و در مجاورت هر قطعه‌ی دیگری قرار گیرند؛ یعنی هیچ محدودیتی وجود ندارد که دو قطعه نتوانند با هم قرار گیرند.

الگوریتم پیشنهادی

الگوریتم ابتکاری CUL

در این مقاله، برای هر قطعه‌ی i ، به تعداد مشخصی تقاضا وجود دارد. برای مثال برای قطعه‌ی با ابعاد 17×10 ، ۴ واحد تقاضا، وجود دارد که باید در ورقه‌ی اصلی قرار گیرند و به همین ترتیب برای سایر قطعات.

گام‌های پیشنهادی برای الگوریتم CUL به صورت زیر است:

گام ۱: قطعات را به صورت صعودی براساس x_{ip} مرتب کن. (x_{ip}, y_{ip}) مختصات گوشه‌ی بالای چپ p امین نسخه از قطعه‌ی i در ورقه است.

گام ۲: اولین قطعه را در ورقه‌ی اصلی مطابق با مختصات گوشه بالای چپ قرار بده.

گام ۳: حرکت قطعه را براساس روش "حرکت یک قطعه" (شکل ۱) برنامه‌ریزی کن.

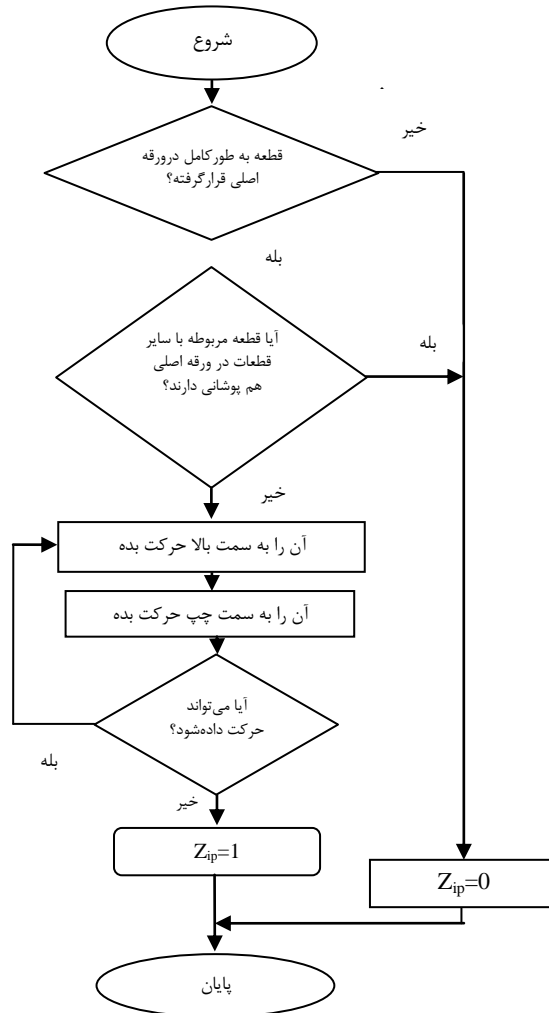
گام ۴: اگر همه‌ی قطعات، چیده شدند، به گام ۵ برو؛ در غیر این صورت، قطعه‌ی بعدی را براساس مختصات گوشه بالای چپ قرار بده، به گام ۳ برو.

گام ۵: قطعه‌ی $z_{ip}=0$ را در گوشه‌ی پایین راست ورقه قرار بده. $z_{ip}=0$ یعنی نسخه‌ی p ام قطعه‌ی i در موقعیت (l_0, w_0) قرار نگرفته است و در غیر این صورت $z_{ip}=1$ است.

گام ۶: حرکت قطعه را براساس روش "حرکت یک قطعه" برنامه‌ریزی کن.

گام ۷: اگر تمام قطعات $z_{ip}=0$ ، چیده شدند، فرایند حل مسأله پایان یافته؛ در غیر این صورت، قطعه‌ی بعدی $z_{ip}=0$ را گوشه‌ی پایین راست ورقه قرار بده، به گام ۶ برو.

روش "حرکت یک قطعه" در شکل (۱) نشان داده شده است.



شکل ۱. روش "حرکت یک قطعه"

الگوریتم بهینه‌سازی ازدحام ذرات

الگوریتم بهینه‌سازی ازدحام ذرات (PSO) توسط جیمز کندی (روانشناس اجتماعی) و راسل سی ابرهارت (مهندس برق) [۴۸] برای مسائل بهینه‌سازی که ماهیت پیوسته بر جواب‌های آن‌ها حاکم است، در سال ۱۹۹۵ مطرح شد و توسط پژوهش‌های بسیاری از نویسندگان توسعه پیدا کرده است [۴۹، ۵۰، ۵۱، ۵۲، ۵۳، ۵۴، ۵۵، ۵۶]. این الگوریتم از رفتار اجتماعی دسته‌ای ذرات و ماهی‌ها الهام گرفته است. [۵۷].

در الگوریتم PSO، هر ذره یک جواب ممکن در فضای جستجوی مسأله می‌باشد. در ابتدا الگوریتم PSO به وسیله‌ی گروهی از ذرات که به طور تصادفی در فضای مسأله تولید شده‌اند، مقداردهی شده و سپس جستجو برای رسیدن به بهترین جواب آغاز می‌شود [۵۷]. در هر مرحله از تکرار الگوریتم، ذره به سمت موقعیت بهتر جابه‌جا می‌شود. موقعیت بعدی برای هر ذره با توجه به دو مقدار به دست می‌آید: اولین مقدار، بهترین موقعیتی است که آن ذره تاکنون داشته است ($pbest$) و دومین مقدار، بهترین موقعیتی است که تا به حال توسط کل ذره‌های آن اجتماع به دست آمده است ($gbest$). به بیان دیگر $gbest$ را می‌توان بهترین $pbest$ در کل گروه در نظر گرفت. این فرآیند تا زمانی که شرایط توقف حاصل شود، ادامه پیدا می‌کند. شرط توقف در این الگوریتم، میل کردن سرعت ذرات به سمت صفر یا رسیدن به تعداد تکرارهای در نظر گرفته شده است.

با توجه به مقادیر $pbest$ و $gbest$ ، هر ذره از روابط زیر برای تعیین موقعیت بعدی استفاده می‌کند:

$$1) v_{ij}(t+1) = w v_{ij}(t) + c_1 r_1 (p_{ij}(t) - x_{ij}(t)) + c_2 r_2 (g_{ij}(t) - x_{ij}(t))$$

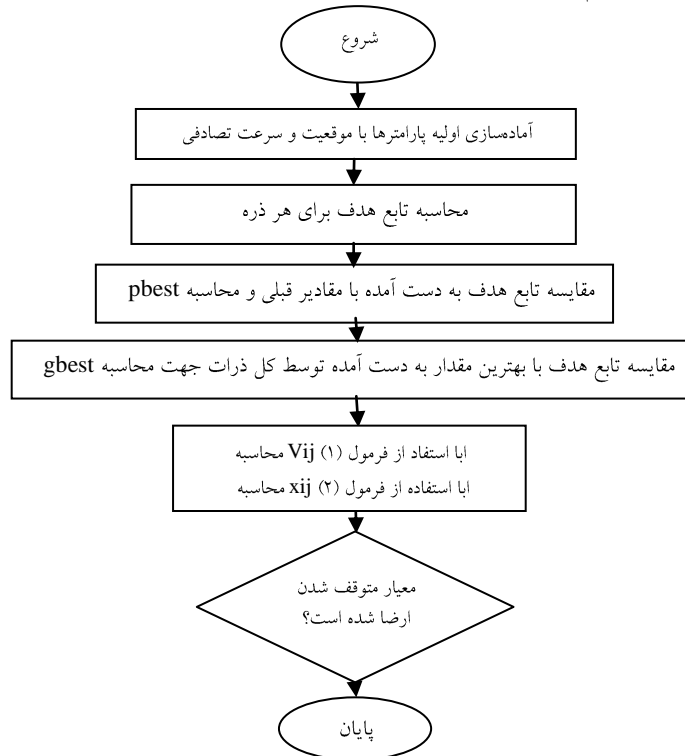
$$2) x_{ij}(t+1) = x_{ij}(t) + v_{ij}(t+1)$$

در روابط فوق، ثابت‌های c_1 و c_2 ، پارامترهای یادگیری (میزان تأثیر) را برای $gbest$ و $pbest$ تعیین می‌کنند و معمولاً برابر با ۲ انتخاب می‌شوند. r_1 و r_2 اعدادی تصادفی در محدوده‌ی [۰، ۱] می‌باشند [۵۰، ۵۱، ۵۲، ۵۳، ۵۴، ۵۶، ۵۸]. $x_{ij}(t)$ ، موقعیت کنونی هر ذره و $v_{ij}(t)$ ، سرعت حرکت ذرات در آن مرحله می‌باشد. w پارامتری است که ضریب اینرسی حرکت ذرات را کنترل می‌کند. در ابتدای اجرای الگوریتم، سرعت بیشتر می‌شود و بعد از مدتی که به پاسخ نزدیک می‌شویم، به کندی کاهش می‌یابد. توابع مورد استفاده برای این منظور معمولاً کاهشی خطی را بعد از هر بار تکرار سبب می‌شوند [۵۹]. در سال ۱۹۹۸، شی فرمول V را با اضافه کردن پارامتر W اصلاح کرد [۶۰].

کندی و ابره‌ارت [۴۸] اولین نسخه‌ی الگوریتم بهینه‌سازی ازدحام ذرات گسسته‌ی دوازده‌گانه را توسعه دادند. جزئیات بیشتر در مورد ادبیات الگوریتم بهینه‌سازی ازدحام ذرات گسسته^۱ در [۶۱، ۶۲] آمده است.

از آن جایی که مسأله‌ی برش، ماهیتی گسسته دارد و PSO استاندارد قادر به حل چنین مسائلی نمی‌باشد، در این مقاله از DPSO که برای بهینه‌سازی در چنین فضاهاهی مناسب می‌باشد، استفاده شده است [۶۳]. از این گذشته برای هدایت بهترین ذره از الگوریتم CUL استفاده شده است. از آنجا که الگوریتم PSO حرکت ذرات حول بهترین ذره را پیاده‌سازی می‌کند، CUL سعی در هدایت بهترین ذره دارد. استفاده از الگوریتم پیشنهادی در کنار الگوریتم PSO، ضمن کاهش قابل توجه زمان حل مسأله‌ی برش مورد نظر، از هم‌پوشانی قطعات، نیز جلوگیری خواهد کرد. در ادامه نشان داده شده است که استفاده از CUL چگونه کیفیت جواب را بهبود می‌دهد.

فلوچارت الگوریتم PSO در شکل (۲) نشان داده شده است.



شکل ۲. فلوچارت الگوریتم PSO. (Malviya and Kumar Pratihar, 2011)

تجزیه و تحلیل داده‌ها

برای بررسی الگوریتم پیشنهادی به کمک زبان برنامه‌نویسی C# و براساس

الگوریتم‌های PSO و CUL نرم‌افزاری تهیه شد. نرم‌افزار پیشنهادی قابلیت تنظیم داشته و کاربر می‌تواند به دو صورت از آن استفاده کند. در حالت اول، این نرم‌افزار با در نظر گرفتن طول و عرض صفحه اصلی و با توجه به اندازه و تعداد قطعات مورد تقاضا، بهترین برش ممکن را ارائه می‌دهد. در حالت دوم از بین عرض‌های معین، بهترین عرض انتخاب می‌شود. در این حالت علاوه بر تعیین بهترین الگوی برش، طول بهینه برای عرض انتخاب شده نیز تعیین می‌شود. نرم‌افزار روی کامپیوتر Pentium 4 با سرعت نسبتاً بالا، با فرض این که در حالت اول اندازه‌ی صفحه اصلی ۳۹۰×۲۶۵ باشد و تقاضایی به شکل زیر موجود باشد، اجرا شد و الگوی بهینه‌ی برش به دست آمد.

جدول ۱. داده‌های مسأله‌ی برش

تعداد مورد نیاز	عرض	طول
۱	۱۵۰	۱۱۰
۳	۱۰۰	۸۰
۲	۳۵	۶۰
۲	۵۵	۶۰

در این مقاله برای حل مسأله برش دوبعدی غیر گیوتینی، پارامترهای PSO پیشنهادی به صورت زیر تعریف می‌شوند:

۱- جمعیت اولیه (Swarm): S با جمعیتی تصادفی شروع می‌شود. اندازه S بسته به مسأله، متفاوت بوده و هیچ معیار مشخصی برای تعیین اندازه S وجود ندارد. S در نرم‌افزار مورد استفاده در این مقاله، به گونه‌ای در نظر گرفته شده است، که قابل تنظیم است. طول هر ذره در S برابر با تعداد قطعاتی است که باید بسته‌بندی شوند. هر ذره یک ترکیب است که شامل اعداد صحیح و توالی قرارگیری قطعات در ورقه‌ی اصلی می‌باشد. $S = [S_{ij}]$ نشانگر Swarm می‌باشد که تعداد ذرات $i = 1, 2, \dots$ و تعداد قطعات $j = 1, 2, \dots$

۲- موقعیت (position): موقعیت X شامل طول و عرض قطعات مطابق با توالی قرارگیری آن‌ها است. این موقعیت شامل چند ماتریس فرعی (ابعاد ماتریس‌های فرعی، دو برابر تعداد قطعات است) که دربرگیرنده موقعیت هر ذره می‌باشد، است. بنابراین

تعداد سطرهای ماتریس X ، دو برابر تعداد سطرهای S می‌باشد.

۳- سرعت (Velocity): برای هر ذره به صورت تصادفی، سرعتی ایجاد می‌شود و در حین اجرای الگوریتم طبق رابطه (۳) به روز می‌شود.

۴- تابع هدف: در الگوریتم پیشنهاد شده هدف، حداقل کردن ضایعات مربوط به برش ورق اصلی است که پس از کم کردن بزرگ‌ترین مستطیل از قسمت باقی‌مانده محاسبه می‌شود.

۵- روابط سرعت و موقعیت: سرعت هر ذره با توجه به $pbest$ و $gbest$ و با استفاده از رابطه‌ی (۳) تغییر می‌کند. در DPSO به دلیل وجود متغیرهای گسسته، فرمول V متفاوت از فرمول V در PSO استاندارد می‌باشد:

$$c_1 \quad V_i^{t+1} = W_{iter} \otimes V_i^t \oplus (c_1 * r_1) \otimes (p_{besti} \ominus X_i^t) \oplus (c_2 * r_2) \otimes (p_{gbest} \ominus X_i^t) \quad (3)$$

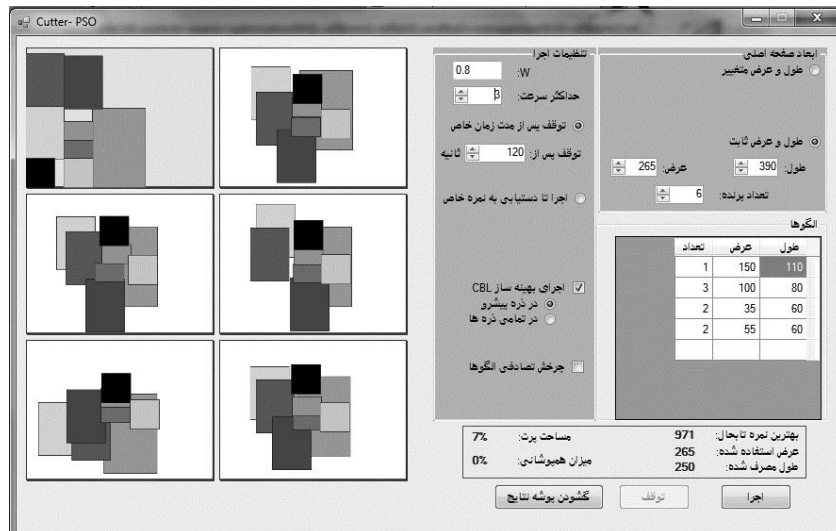
و c_2

ضرایب یادگیری، r_1 و r_2 اعداد تصادفی در بازه $[0, 1]$ و W_{iter} ضریب اینرسی می‌باشد. در این مقاله، W مقدار عددی است که قبل از اجرای نرم‌افزار قابل تنظیم است.

در پایان تکرار t ام، موقعیت هر ذره x_i^t با استفاده از رابطه‌ی v_i^{t+1} به روز می‌شود و موقعیت جدید هر ذره به دست می‌آید:

$$c_2 \quad X_i^{t+1} = x_i^t + v_i^{t+1} \quad (4)$$

در رابطه‌ی ۳، عملگر \oplus بیانگر جمع خاص، عملگر \otimes بیانگر ضرب خاص و عملگر \ominus بیانگر تفریق خاص است. توضیح بیشتر این عملگرها در منبع [۶۳] آمده است. پارامترها در نرم‌افزار، به شکل زیر تنظیم شده‌اند:



شکل ۳. پارامترهای روش PSO برای حل مسأله‌ی برش در حالت اول

همان‌طور که در شکل (۳) مشخص است تعداد ۶ ذره در نظر گرفته شده و الگوی بهینه‌ی برش الگویی است که زمینه‌ی آن سبز رنگ است. لازم به ذکر است که در سایر ذرات غیر از بهترین ذره، هم‌پوشانی وجود دارد که دلیل آن، اعمال بهینه‌ساز CUL فقط در ذره‌ی پیشرو است. در صورتی که همین مسأله برای حالتی که CUL برای تمام ذره‌ها اعمال شود، نرم‌افزار به شکل زیر خواهد بود:



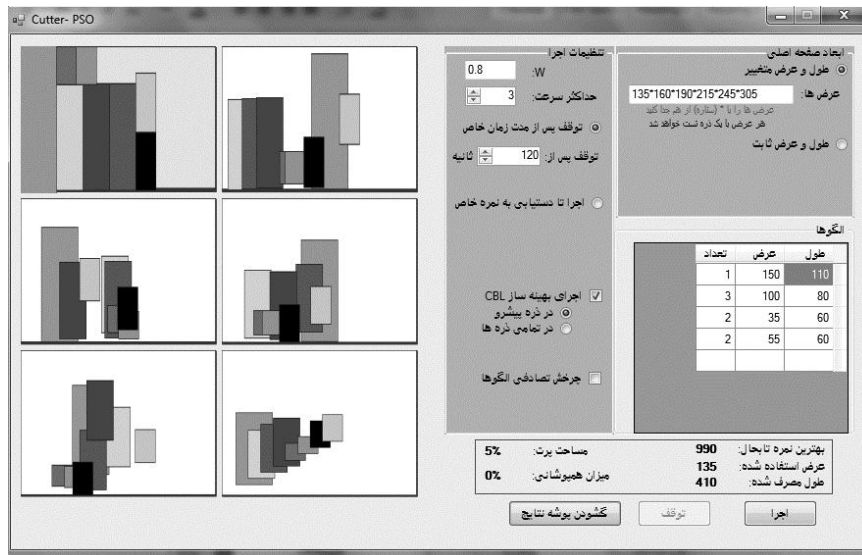
شکل ۴. اعمال الگوریتم CUL روی تمام ذره‌ها

همان طور که در شکل (۴) مشاهده می‌شود، اعمال الگوریتم CUL روی تمام ذرات ضمن این که تمام جواب‌ها را به جواب‌های شدنی تبدیل می‌کند، هم‌پوشانی‌ها را نیز از بین می‌برد. در نتیجه استفاده از الگوریتم پیشنهادی CUL در کنار الگوریتم PSO، ضمن کاهش قابل توجه زمان حل مسأله‌ی برش مورد نظر، از هم‌پوشانی قطعات نیز جلوگیری خواهد کرد و کیفیت جواب را بهبود می‌بخشد.



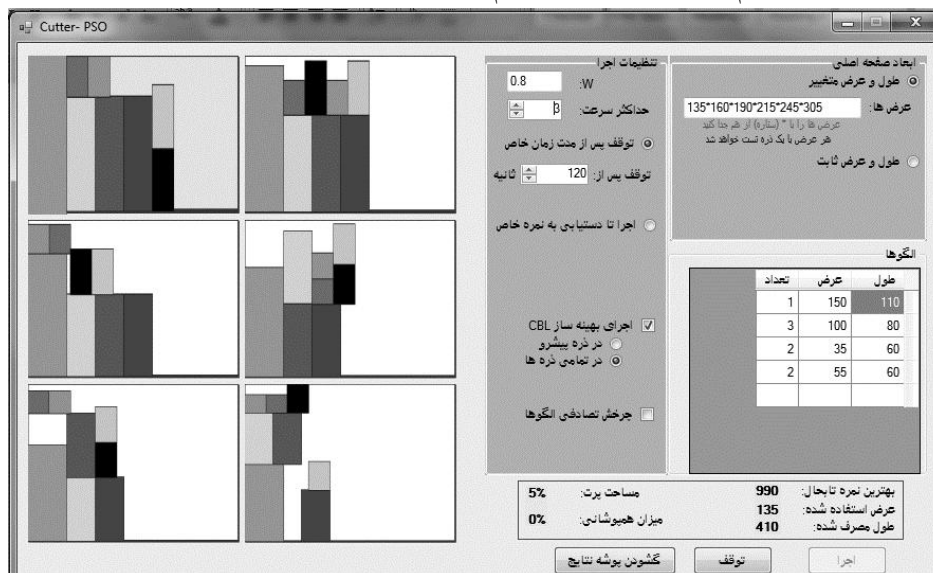
شکل ۵. اجرای الگوریتم پیشنهادی در صورت عدم اعمال الگوریتم CUL

همان طور که در شکل (۵) دیده می‌شود، در صورت عدم اعمال الگوریتم CUL در این نرم‌افزار، هم‌پوشانی قابل توجهی ایجاد شده و کارایی کاهش می‌یابد. همان طور که در رابطه با حالت دوم گفته شد؛ کاربر می‌تواند، عرض‌های موردنظر را به نرم‌افزار بدهد. در این حالت، نرم‌افزار پس از تعیین عرض بهینه، الگوی بهینه‌ی برش و طول بهینه‌ی صفحه‌ی اصلی را نیز برای کاربر مشخص می‌کند. این حالت از نرم‌افزار برای صنایعی که به منظور برش ورقه‌ی اصلی، عرض‌های از پیش تعیین شده دارند، بسیار مناسب می‌باشد. برای نشان دادن حالت دوم ۶ عرض مختلف ۱۳۵، ۱۶۰، ۱۹۰، ۲۱۵، ۲۴۵ و ۳۰۵ با همان تقاضای قبلی در نظر گرفته شده است. پارامترها، به شکل زیر تنظیم شده‌اند:



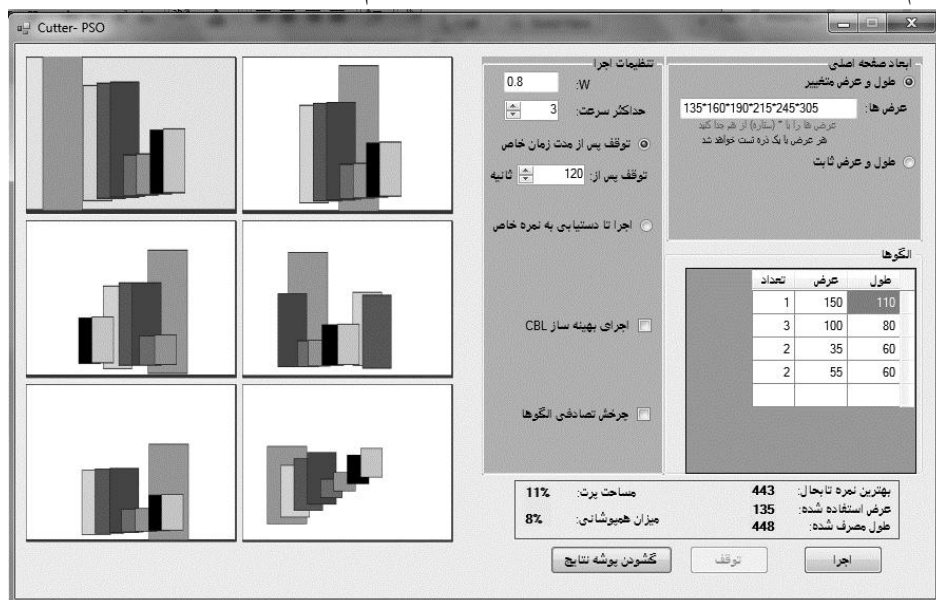
شکل ۶. پارامترهای روش PSO برای حل مسأله‌ی برش در حالت دوم

در شکل (۶) نیز تعداد ۶ ذره در نظر گرفته شده و الگوی بهینه‌ی برش الگویی است که زمینه‌ی آن سبز رنگ شده است. در این حالت نیز به دلیل عدم اعمال بهینه‌ساز CUL در تمامی ذرات، هم‌پوشانی وجود دارد. در صورتی که همین مسأله برای حالتی که CUL برای تمام ذره‌ها اعمال شود، نرم‌افزار به شکل زیر خواهد بود:



شکل ۷. اعمال الگوریتم CUL روی تمام ذره‌ها

همان‌طور که در شکل (۷) مشاهده می‌شود اعمال الگوریتم CUL روی تمام ذرات، تمام جواب‌ها را به جواب‌های شدنی تبدیل می‌کند و هم‌پوشانی‌ها را از بین می‌برد.



شکل ۸. اجرای الگوریتم پیشنهادی در صورت عدم اعمال الگوریتم CUL

همان‌طور که در شکل (۸) دیده می‌شود، در صورت عدم اعمال الگوریتم CUL در این نرم‌افزار، هم‌پوشانی قابل توجهی ایجاد می‌شود.

مطالعاتی که در آن‌ها الگوریتمی برای مسئله برش دوبعدی غیرگوتینی ارائه شده است، برای یک ورق اصلی ضایعات زیر ۶ درصد، برای مسائلی با حداکثر ۳۵ قطعه، قابل قبول بوده و در حدود استاندارد محسوب می‌شود [۳۸، ۴۰، ۳۹، ۶۵]. در این پژوهش نیز همان‌طور که در شکل‌های بالا مشخص است، ضایعات زیر ۶ درصد است که همین موضوع، کارایی الگوریتم و نرم‌افزار پیشنهادی را اثبات می‌کند.

هم‌چنین در تعدادی از این الگوریتم‌ها لازم نیست تمامی قطعات موجود در یک ورق جایگذاری شوند، در حالی که در الگوریتم پیشنهادی در مسئله دشوارتر برش با تقاضا کلیه قطعات باید در ورقه‌ی اصلی جایگذاری شوند. علاوه بر این در اکثر مسائل برش، طول و عرض ورقه‌ی اصلی ثابت است، در حالی که در این پژوهش علاوه بر وجود این ویژگی، امکان تعیین عرض‌های متفاوت توسط کاربر وجود دارد.

نتیجه‌گیری

در این مقاله، مسأله‌ی برش دوبعدی غیرگیوتینی با تقاضا مورد بررسی قرار گرفت. اکثر محققان به مسأله‌ی برش با هدف کاهش ضایعات در ورق به طور مجزا، بدون در نظر گرفتن تقاضای قطعات پرداخته‌اند، در حالی که اکثر مسائل واقعی برش به صورت مسأله‌ی برش با تقاضا هستند. البته این مسأله جزء مسائل NP-Hard بوده و نیاز به روشی فراابتکاری برای حل عملی آن خواهد بود. لذا در این مقاله با استفاده از الگوریتم بهینه‌سازی ازدحام ذرات به حل مسأله‌ی برش پرداخته شده است. نتایج محاسباتی نشان می‌دهد که استفاده از الگوریتم CUL، باعث بهبود عملکرد الگوریتم بهینه‌سازی ازدحام ذرات می‌شود. در این مقاله با ارائه‌ی نرم‌افزاری، کارایی الگوریتم پیشنهادی مورد بررسی قرار گرفته است.

محاسبه پارامترهای مربوط به الگوریتم PSO با استفاده از روش‌های هوش مصنوعی مانند شبکه‌های عصبی مصنوعی و الگوریتم ژنتیک می‌تواند نتایج جالب توجهی را ارائه دهد. تحلیل حساسیت نیز می‌تواند در بهینه‌یابی پارامترهای مربوط به الگوریتم PSO مورد استفاده قرار گرفته و نتایج را بهبود دهد. همچنین پیشنهاد می‌شود در قسمت تنظیم پارامترها از ابزارهای آماری مانند RSM استفاده شود و یا برای هر پارامتر ۳ بازه در نظر گرفته شود و ترکیبات به وجود آمده (سعی و خطا) بررسی شوند تا حالت بهینه به دست آید.

در نهایت به عنوان پیشنهادهای کاربردی پیشنهاد می‌شود، جهت انجام این الگوریتم (نرم‌افزار) در صنایع مختلف تحقیقات بیشتری انجام شود. زیرا به نظر می‌رسد در صورت شناسایی شرایط و محدودیت‌های عملی هر صنعت و اعمال آن‌ها در الگوریتم (مانند گیوتینی بودن برش‌ها، دسته‌بندی قطعات و امکان استفاده از ضایعات برای تولید قطعات کوچک‌تر، دخالت دادن هزینه‌های مربوط به برش در هزینه‌ی استفاده از یک ورق و...) می‌توان به نحو مؤثرتری هزینه‌های ناشی از برش را کاهش داد. علاوه بر این، امکان رسیدن به الگوریتمی کاراتر، با تحقیق بیشتر روی خصوصیات تنوری مطرح در روش‌های فراابتکاری نیز منتفی نخواهد بود.

منابع و مأخذ

۱. مصلحی، قاسم. رضایی، علیرضا. (۱۳۸۳)، "ارائه الگوریتمی برای مسئله برش دوبعدی با تقاضا". *استقلال*، سال ۲۳، شماره ۲.
2. Dyckhoff, H. (1990). **A typology of cutting and packing problems**. *European Journal of Operational Research.*, 44, pp. 145–159.
3. Chambers, M.L.; Dyson, R.G. (1976). **The cutting stock problem in the flat glass industry**. *Operations Research Quarterly.*, Vol 27, Nr. 4, pp. 949-957.
4. Coffman, E.G.; Garey, M.R.; Johnson, D.S.; Tarjan, R.E. (1980). **Performance Bounds for Level-Oriented Two-Dimensional Packing Algorithms**. *SIAM Journal on Computing.*, 9, 4, pp. 808-826.
5. Coffman, E.G.; Shor, P.W. (1990). **Average-Case Analysis of Cutting and Packing in Two Dimensions**. *European Journal of Operational Research.*, 44, 2, pp. 134-145.
6. Paul, R.J. (1979). **A production scheduling problem in the glass container industry**. *Operations Research.*, Vol 27, Nr. 2, pp. 290-302.
7. Van-dat, C.; Hifi, mhand, and Le cun , bertrand, . (1997). **Constrained two dimensional cutting stock problems , a best first branch and bound algorithm**. *laboratoire PR&SM-CNRS URA 1525, University of Versailles, paris*.
8. Whitlock, C.; Christofides, N. (1977). **an Algorithm for Two-Dimensional Cutting Problems**. *Operations Research.*, Vol. 25, Nr. 1, January-February, pp. 30- 44.
9. Gass, S. (1985). *Linear Programming, Methods and Applications*. McGraw-Hill
10. Gradišar, M.; Jesenko, J.; Resinovič, G. (1997). **Optimization of roll cutting in clothing industry**. *Computer & Operations Research.*, 24, pp. 945-953.
11. Gradišar, M.; Resinovič, G.; Jesenko, J.; Kljajić, M. (1999). **A sequential heuristic procedure for one-dimensional cutting**. *European Journal of Operational Research.*, 114, pp. 557-568.
12. Kopač, J. (2002). **Cutting forces and Their Influence on the Economics of Machining**. *Strojniški vestnik - Journal of Mechanical Engineering.*, 48 3, pp. 121-132.
13. Vasko, F.; Newhart, D.; Stott, K. (1999). **A hierarchical approach for one-dimensional cutting stock problems in the steel industry that maximizes yield and minimizes overgrading**. *European Journal of Operational Research.*, 114, pp. 72-82.
14. Westerlund, T.; Harjunkoski, I.; Isaksson, J. (1998). **Solving a Production Optimization Problem in a Paper-Converting Mill with MILP**. *Computers & Chemical Engineering.*, 22, pp. 563-570.
15. Nazemi, J.(2005). **Kiln Planning a Cutting Stock Approach**, Social Research Research Network.
16. Kantorovich, L.V. (1939). **Mathematical methods of organizing and planning production**. *Mgmt. Sci.*, Vol. , pp.363-422.

17. Gilmore, P.C.; Gomory, R.E. (1961). **A Linear Programming Approach to the Cutting-Stock Problem**. Operations Research., Vol. 9, pp. 849-859.
18. Gilmore, P.C.; Gomory, R.E. (1965). **Multi-stage cutting stock problems of two and more dimensions**. Operations Research., 13, pp. 94-120.
19. Farley, A.A. (1988). **Mathematical programming models for cutting stock problems in the clothing industry**. Journal of the Operational Research Society., 39, pp. 41-53.
20. Farley, A.A. (1990). **A note on bounding a class of linear programming problems, including cutting stock problems**. Operations Research., 38, pp. 922-923.
21. Beasley, J.E. (1985). **Algorithms for Unconstrained Two-Dimensional Guillotine Cutting**. Operation's Research Society., Vol.36, No.4, PP.297-306.
22. Hifi, M.; Ouafi, R. (1977). **Best-first search And Dynamic Programming Methods for Cutting Problems: The cases of One or More Stock Plates**. Computers ind.Eng., Vol.32, No.1, PP.187-205.,.
23. Agrawal, P.K. (1993). **Minimizing trim loss in cutting rectangular blanks of a single size from a rectangular sheet using orthogonal guillotine cuts**. European Journal of Operational Research., 64, pp. 410-422.
24. Cui, Y. (2005). **Dynamic programming algorithms for the optimal cutting of equal rectangles**. Applied Mathematical Modeling., 29, pp. 1040-1053.
25. Cui, Y. (2006). **Simplest optimal cutting patterns for equal rectangles**. Operations Research Letters, 34, pp. 630-638.
26. Cheng, C.H.; Feiring, B.R.; Cheng, T.C. (1994). **The cutting stock problem-a survey**. International Journal of Production Economics., 36, pp. 291-305.
27. Dowsland, K.A.; Dowsland, W.B. (1992). **Packing problems**. European Journal of Operational Research., 56, pp. 2-14.
28. [^] Haessler, R.W.; Sweeney, P.E. (1991). **Cutting stock problems and solution procedures**. European Journal of Operational Research., 54, pp. 141-150.
29. Sweeney, P.E.; Paternoster, E.R. (1992). **Cutting and packing problems: a categorized, application-orientated research bibliography**. Journal of the Operational Research Society., 43, pp. 691-706.
30. Arenales, M.; Morabito, R. (1995). **An AND/OR-graph approach to the solution of two-dimensional non-guillotine cutting problems**. European Journal of Operational Research., 84, pp. 599-617.
31. Beasley, J.E. (1985). **An Exact Two-Dimensional Non-Guillotine Cutting Tree Search Procedure**. Operation's Research., Vol.33, No.1, PP.49-64.
32. Hadjiconstantinou, E.; Christofides, N. (1995). **An exact algorithm for general, orthogonal, two-dimensional knapsack problems**. European Journal of Operational Research., 83, pp. 39-56.

33. Lai, K.K.; Chan, W.M. (1997). **An evolutionary algorithm for the rectangular cutting stock problem.** International Journal of Industrial Engineering., 4, pp. 130-139.
34. Tsai, R.D.; Malstrom, E.M.; Meeks, H.D. (1988). **A two-dimensional palletizing procedure for warehouse loading operations.** IIE Transactions., 20, pp. 418-425.
35. Suliman, S.M.A. (2006). **A sequential heuristic procedure for the two-dimensional cutting-stock problem.** Int. J. Production Economics., 99, pp. 177-185.
36. Hadjiconstantinou, E.; Iori, M. (2007). **A hybrid genetic algorithm for the two-dimensional single large object placement problem.** European Journal of Operational Research., 183, pp. 1150-1166.
37. Ono, T.; Ikeda, T. (1998). **Optimization of two-dimensional guillotine cutting by genetic algorithms.** Proceedings of Sixth European Congress on Intelligent Techniques and Soft Computing.
38. Lai, K. K.; Chan, W.M. (1997). **Developing a Simulated Annealing Algorithm for the Cutting Stock Problem.** Computer and Industrial Engineering., Vol. 33, pp. 115-127.
39. Leung, T. W.; Yung, C. H.; Truett Marvin, D. (2001). **Applications of Genetic Search and Simulated Annealing to the Two-dimensional Non-guillotine Cutting Stock Problem.** Computer and Industrial Engineering., Vol. 40, pp. 201-214.
40. Beasley, J.W.(2000). **A Population Heuristic for Constrained Two-Dimensional Non-Guillotine Cutting.**
<http://mscmga.ms.ic.ac.uk/jebihtml>.
41. Tiwari, S.; Chakraborti, N. (2006). **Multi-objective optimization of a two-dimensional cutting problem using genetic algorithms.** Journal of Materials Processing Technology., 173, pp. 384-393.
42. Jiang, J.Q.; Xing, X.L.; Yang, X.W.; Liang, Y.C. (2004). **A Hybrid Algorithm Based on PSO and Genetic Operation and Its Applications For Cutting Stock Problem.** Proceedings of The Third Int. Conference on Machine Learning and Cybernetics., Shangai, pp. 2198-2201.
43. Liu, B.; Wang, L.; & Jin, Y. H. (2008). **An effective hybrid PSO-based algorithm for flow shop scheduling with limited buffers.** Computers and Operations Research, 35, pp. 2791-2806.
44. Chen, M. (2011). **A hybrid ANFIS model for business failure prediction utilizing particle swarm optimization and subtractive clustering,** Information Sciences.
45. Wang, H. S.; Che, Z. H.; & Wu, C. (2010). **Using analytic hierarchy process and particle swarm optimization algorithm for evaluating product plans.** Expert Systems with Applications, 37, pp. 1023-1034.
46. Marinakis, Y.; Marinaki, M.; Doumpos, M.; & Zopounidis, C. (2009). **Ant colony and particle swarm optimization for financial classification problems.** Expert Systems with Applications, 36, pp. 10604-10611.
47. Onut, S.; Tuzkaya, U. R.; & Dog˘ac, B. (2008). **A particle swarm optimization algorithm for the multiple-level warehouse layout design problem.** Computers and Industrial Engineering, 54, pp. 783-799.

48. Kennedy, J.; Eberhart, R. (1995). **Particle Swarm Optimization**. in: Proceedings of IEEE International Conference on Neural Networks, Perth, Australia, pp. 1942-1948.
49. Shi, Y.; Eberhart, R.C. (1999). **Empirical study of particle swarm optimization**. in: Proc. IEEE Int. Congr. Evolutionary Computation., vol 3, pp. 101-106.
50. Jia, D.L.; Zheng, G. X.; Qu, B.Y.; & Khurram Khan, M. (2011). **A hybrid particle swarm optimization algorithm for high-dimensional problems**, Computers & Industrial Engineering.
51. Kiranyaz, S.; Pulkkinen, J.; & Gabbouj, M. (2011). **Multi-dimensional particle swarm optimization in dynamic environments**, Expert Systems with Applications, 38, pp. 2212–2223.
52. Shi, Y.; Liu, H.; Gao, L.; & Zhang, G. (2011). **Cellular particle swarm optimization**, Information Sciences, 181, pp. 4460–4493.
53. Chen, M.R.; Li, X.; Zhang, X.; & Lu, Y. Z. (2010). **A novel particle swarm optimizer hybridized with extremal optimization**, Applied Soft Computing, 10, pp. 367–373.
54. Liu, Y.; Wang, G.; Chen, H.; Dong, H.; Zhu, X.; & Wang, S. (2011). **An Improved Particle Swarm Optimization for Feature Selection**, Journal of Bionic Engineering, 8, pp. 191–200
55. Liua, B.; Wanga, L.; Liud, Y.; Qiane, B.; & Jin, Y. H. (2010). **An effective hybrid particle swarm optimization for batch scheduling of polypropylene processes**, Computers and Chemical Engineering, 34, pp. 518–528
56. Robati, A.; Barani, G. A.; Nezam Abadi Pour, H.; Fadaee, M. J.; Rahimi Pour Anaraki, J. (2011). **Balanced fuzzy particle swarm optimization**, Applied Mathematical Modelling.
57. Poli, R.; Kennedy, J. (2007). **Particle swarm optimization An overview**. Springer Science., pp. 33–57.
58. Navalertporn, T.; Afzulpurkar, N.V. (2011). **Optimization of tile manufacturing process using particle swarm optimization**, Swarm and Evolutionary Computation, 1, pp. 97–109.
59. Alatas, B.; Akin, E. A.; Ozers, B.I. (2007). **Chaos-embedded particle swarm optimization algorithms**, Chaos, Solitons and Fractals.
60. Shi, Y.; Eberhart, R.C. (1998). **A modified particle swarm optimizer[C]**. In: IEEE World Congress on Computational Intelligence., pp. 69-73.
61. Wang, K.P.; Huang, L.; Zhou, C.G.; Pang, W. (2003). **Particle Swarm Optimization For Traveling Salesman Problem**. Proceedings of The Second International Conference on Machine Learning and Cybernetics, Xi'an, pp. 1583-1585.
62. Wang, C.; Zhang, J.; Yang, J.; Hu, C.; Liu, J. (2005). **A Modified Particle Swarm Optimization Algorithm and its Application for Solving Traveling Salesman Problem**. Neural Networks and Brain., ICNN&B '05. Int. Conference on, 2, pp. 689-694.
63. Soke, A.; Bingul, Z. Applications of Discrete PSO Algorithm to Two-Dimensional Non-Guillotine Rectangular Packing Problems. **Conf. on Genetic and Evolutionary Methods**.

64. Malviya, R.; Kumar Pratihari, D. (2011). **Tuning of neural networks using particle swarm optimization to model MIG welding process**, Swarm and Evolutionary Computation.
65. Jakobs, S. (1996). **On genetic algorithms for the packing of polygons**. European Journal of Operational Research, 88, pp. 165- 181.