چشم‌انداز مدیریت صنعتی
شماره۳ ـ پاییز ۱۳۹۰
ص‌ص ۱۳۷- ۱۲۱

ترکیب الگوریتم پرواز پرندگان و الگوریتم ابتکاری CUL برای حل مسأله برش دو بعدی غیرگیوتینی با تقاضا

فائزه اسدیان اردکانی*، علی مروتی شریف‌آبادی**

چکیده

در این مقاله، مسأله برش دو بعدی با تقاضا مورد بررسی قرار می‌گیرد. در این مسأله با برش ورق‌های مستطیل شکل بزرگ، مستطیل‌های کوچک‌تر مورد نیاز باید به نحوی تولید شوند که ضمن تأمین تقاضا برای آنها، ضایعات یا تعداد ورق‌های مصرفی حداقل شود. مسأله برش، جزء مسائل NP-Hard است که روش‌های دقیق قادر، به حل عملی آنها نیستند. لذا در این مقاله با استفاده از الگوریتم پرواز پرندگان، الگوریتمی فراابتکاری برای حل مسأله برش دو بعدی با تقاضا ارائه شده است. برای بهبود کارایی این الگوریتم و جلوگیری از هم‌پوشانی در مسأله برش، الگوریتم ابتکاری CUL به‌کار گرفته شد. همچنین برای بررسی نتایج الگوریتم پیشنهادی (ترکیب الگوریتم‌های PSO و CUL) نرم‌افزاری تهیه شد که با در نظر گرفتن طول و عرض صفحه اصلی و با توجه به اندازه‌های قطعات و تعداد مورد تقاضا، بهترین الگوی برش ممکن را ارائه می‌دهد.

کلید واژه‌ها: الگوریتم پرواز پرندگان، الگوریتم پرواز پرندگان گسسته، الگوریتم CUL، مسأله برش دو بعدی.

تاریخ دریافت مقاله: ۹۰/۰۲/۲۰ ، تاریخ پذیرش مقاله: ۹۰/۰۷/۱۰.

* دانشجوی کارشناسی ارشد مدیریت صنعتی (نویسنده مسئول).
Email: faezehasadian@gmail.com

** استادیار مدیریت و حسابداری، دانشگاه یزد.

مقدمه

در فرایند تولید در بسیاری از صنایع، این نیاز وجود دارد که قطعات کوچک‌تری از راه برش اجسام بزرگ‌تر حاصل شوند و یا قطعات کوچک‌تر در یک جسم بزرگ‌تر جای داده شوند. در این فرایند معمولاً بخش‌هایی از جسم بزرگ‌تر به قطعاتی تبدیل می‌شوند که قابل استفاده در هیچ یک از محصولات تولیدی نیستند و ضایعات و دورریز محسوب می‌شوند. کاهش چنین ضایعاتی، نقش مهمی در کاهش هزینه‌ها دارد و به عنوان یکی از موضوعات علم تحقیق در عملیات ـ با نام مسأله‌ی برش ـ توجه بسیاری از محققان را در نیم قرن گذشته جلب کرده است[۱].

مسأله برش در بسیاری از صنایع تولید انبوه مانند کاغذسازی، شیشه‌سازی، برش الوار، پلاستیک‌سازی و نساجی کاربرد دارد [۱۹]. مسأله برش، جزء مسائل بهینه‌سازی ترکیبی است که کاربردهای فراوانی در علم کامپیوتر، مهندسی صنایع، فرایند ساخت و تولید و غیره دارد. تحقیقات زیادی درباره ابزار و روش‌های حل این نوع مسائل در صنعت صورت گرفته است [۸ ،۱۳، ۵۱،۱۴ ،۴۰، ۵۸].

در سال‌های اخیر، محققان بسیاری به شاخه‌های متفاوت مسأله برش پرداخته‌اند و روش‌های متعددی برای بهینه‌سازی این مسأله توسعه داده شده است [۲۲، ۲۵، ۲۶، ۳۴، ۵۳، ۵۷]. اولین شکل فرمولی برای این نوع مسائل را کانترویچ (۱۹۳۹) در صنایع کاغذسازی مطرح کرد [۳۸]. وی مدلی را برای مسأله برش یک بعدی با تقاضا بیان می‌کند که برای حل آن، ابتدا تمامی سبک‌های برش یک میله به قطعات مورد نیاز باید شناسایی شود. این امر، تنها برای مسائل خیلی کوچک، عملی خواهد بود [۳۲].

پیشینه تحقیق

محققین زیادی به مطالعه درباره مساله برش پرداخته‌اند. گیلمور و گموری (۱۹۶۱) با استفاده از برنامه‌ریزی خطی، روشی را برای ایجاد الگوی برش در مسائل برش یک بعدی ارائه دادند [۲۳]. آنها همچنین مسأله برش دو بعدی را به عنوان مسأله برش یک بعدی دومرحله‌ای در نظر گرفتند که در آن، اولین برش، موازی با یکی از اضلاع و سپس هر برش، عمود بر برش اول است. روش حل آنها بر مبنای برنامه‌ریزی عدد صحیح با تکنیک ایجاد ستون می‌باشد. هر ستون، یک الگوی برش شدنی را ارائه می‌کند؛ الگویی که از طریق حل یک مسأله کوله‌پشتی دو بعدی ایجاد می‌شود [۲۴]. علاوه بر این دو، محققان دیگری نیز از مدل‌های برنامه‌ریزی خطی عدد صحیح برای حل مسأله برش گیوتینی متعامد دومرحله‌ای نامحدود استفاده کرده‌اند [۲۰، ۲۱]. بیزلی [۵] برای یافتن یک سبک برش بهینه یا حل مسأله برش بدون تقاضا، از روش برنامه‌ریزی پویا استفاده کرد و رابطه‌های بازگشتی را بر این مبنا ارائه داد که حداکثر ارزش یک

مستطیل برای تبدیل آن به قطعات، در یکی از سه حالت برش ندادن، برش دادن عمودی یا برش دادن افقی آن ایجاد می‌شود. این مسأله ـ با ابعاد مختلف ورق اصلی ـ مورد توجه هایفای [۳۰] نیز بوده است. بیزلی، روش شاخه و کرانی [۶] را ارائه می‌کند که در آن، مسأله برش دو بعدی با یک مدل صفر و یک صحیح فرمول‌بندی می‌شود و سپس از روش آزادسازی لاگرانژی و رویه زیرگرادیان برای حل مدل استفاده می‌شود. آگراوال [۲] از رویکرد انشعاب و تحدید برای حل مسأله برش اشکال مستطیلی ـ برای وقتی اندازه‌های قطعات، یکسان باشند استفاده کرد. کوی از الگوریتم انشعاب و تحدید و برنامه‌ریزی پویا برای حل مسأله برش دو بعدی با قطعات مستطیل هم‌اندازه استفاده کرد [۱۶،۱۵]. یانگ و همکاران [۶۰] از روش انشعاب و تحدید برای حل مسأله استفاده کردند آن هم در هنگامی برای هر قطعه منظور شده و مسأله به دنبال حداکثر کردن مجموع ارزش قطعات بریده شده است. چانی و همکاران [۹] روندی را برای برش قطعات از ورق با طول نامحدود ارائه کردند. این روند، تکنیک کوله‌پشتی یک بعدی را با بسته‌بندی قطعات به ترتیب، ترکیب می‌کند. پس از آن، این روش به مسائل برش دوبعدی تعمیم داده شد. زک [۶۱] روند خاصی را برای مسأله برش چندمرحله‌ای ارائه داد که سطرها و ستون‌ها به صورت پویا را ایجاد می‌کند. این روش که به ایجاد سطر و ستون اشاره دارد، از یک مسأله مشخص (مسأله کوله‌پشتی غیرخطی) بیان شده در چارچوب الگوریتم سیمپلکس تجدیدنظرشده استفاده می‌کند. چانگ و همکاران [۱۰] در ادبیات مسأله برش (به ویژه دو بعدی)، به مطالعه پرداختند. آن‌ها به روش‌شناسی‌های متفاوت استفاده شده برای حل این نوع مسائل و جنبه‌های عملی توجه ویژه داشتند. کریستوفیلد و وایت‌لاک [۱۱] روش درخت جستجویی را ارائه کردند که در آن با برشی افقی یا عمودی در هر گره در یکی از نقاط عرضی یا طولی ورق می‌توان تمامی سبک‌های برش را تولید کرد. این دو محقق برای پرهیز از سبک‌های تکراری، قوانینی را مطرح کردند و برای محدود کردن دامنه جستجو در هر گره، حد بالایی را از طریق حل یک مدل حمل و نقل و همچنین از رهگذر رابطه‌های بازگشتی گیلمور و گموری به دست آوردند. وانگ [۵۴] دو الگوریتم n مرحله‌ای برای مسأله برش دوبعدی محدود ارائه داد. الگوریتم‌ها به طور عمودی یا افقی مستطیل‌ها را به شکل یک الگوی گیوتینی بزرگ‌تر به وسیله‌ی قبول مستطیل‌هایی که درصد ضایعات از سطح انتظار از قبل تعیین شده تجاوز نمی‌کند، می‌سازند. سینترا و همکاران [۱۲] از برنامه‌ریزی پویا و تولید ستون برای حل مسأله برش دوبعدی گیوتینی چرخشی استفاده کردند. تحقیقاتی [۱۸، ۲۹، ۴۸] نیز به فعالیت‌های قبل از سال ۱۹۹۰ تخصیص یافت. برخی محققین به پیشینه مسأله برش دوبعدی غیرگیوتینی محدود پرداخته‌اند [۴، ۶، ۲۷، ۳۵، ۵۰].

از جمله روش‌های دیگر برای حل مسأله برش می‌توان به الگوریتم ترکیبی وانگ [۵۴] اشاره نمود. دیتریچ و یاکوتیزر[۱۷] نیز هیافتی ابتکاری را برای حل مسأله برش بیان نمودند که یک

الگوریتم تک‌گذری است که در آن، انتخاب قطعات برای جایگذاری در ورق‌های اصلی، بر اساس یک سری قوانین صورت می‌گیرد. این محققان با مطالعات تجربی سعی نموده‌اند بهترین قوانین را شناسایی نمایند.

از روش‌های فرا ابتکاری نیز برای حل مسأله برش استفاده شده است. لی و چان [۳۶] از روش شبیه‌سازی تبرید عناصر (SA) برای حل مسائل برش غیرگیوتینی دو یا سه‌بعدی استفاده کردند. لئونگ و همکاران [۳۷] عملکرد و کارایی الگوریتم ژنتیک (GA) و الگوریتم SA را در حل مسأله برش مقایسه کردند. بیزلی [۶،۷] از روشی ابتکاری برای حل مسأله برش دو بعدی غیرگیوتینی محدود استفاده کرده است. مصلحی و رضایی [۱] از روش ابتکاری SA برای حل مسأله برش دوبعدی با تقاضا استفاده کرده‌اند. یناسی و همکاران [۵۹] در روند حل ابتکاری، برای اضافه کردن قطعات بر اساس اصل تناسب تلاش کردند. تیواری و چاکربرتی [۴۹] از الگوریتم ژنتیک برای حل مسأله برش با فرض‌های گیوتینی و غیرگیوتینی استفاده کردند. سلیمان [۴۷] از روش ابتکاری ترتیبی سه مرحله‌ای برای حل مسأله برش دوبعدی با هدف کمینه سازی میزان برش استفاده کرده است. از پژوهش‌های دیگری که در آنها از رویکرد ابتکاری برای حل مسأله برش دوبعدی استفاده شده است، می‌توان به [۲۸، ۳۹] اشاره کرد. جیانگ و همکاران [۳۱] از روش PSO و الگوریتم ژنتیک برای حل مسأله برش دوبعدی غیرگیوتینی محدود استفاده کرده‌اند.

در این پژوهش، برش‌ها غیرگیوتینی در نظر گرفته شده‌اند. برش گیوتینی، برشی است که از یک ضلع آزاد مستطیل شروع می‌شود و موازی با یکی از اضلاع ورق اصلی، به ضلع آزاد مقابل ختم می‌شود. اما در برش غیرگیوتینی، چنین فرضی وجود ندارد.

روش‌شناسی

در این مقاله برای حل مسأله برش دوبعدی غیرگیوتینی با تقاضا ـ که در قسمت قبل تشریح شد ـ از روش الگوریتم پرواز پرندگان استفاده شده است. این روش، یکی از روش‌های فرا ابتکاری جدید می‌باشد که توانمندی آن در حل مسائل مختلف مدیریتی اثبات شده است. در ادامه، این روش تشریح می‌شود.

الگوریتم پرواز پرندگان[1]

این الگوریتم را جیمز کندی (روانشناس اجتماعی) و راسل‌ابرهارت (مهندس برق) [۳۳] در ۱۹۹۵ برای حل مسائل بهینه‌سازی ـ که ماهیت پیوسته بر جواب‌های آن‌ها حاکم است ـ مطرح کردند. بسیاری از نویسندگان، کار آنها را توسعه داده‌اند [۴۵، ۴۲]. خلاصه‌ای از توسعه، بهبود و کاربردهای این الگوریتم در [۵۲] آمده است.

این الگوریتم از رفتار اجتماعی دسته پرندگان و ماهی‌ها الهام گرفته شده است. دسته‌ای از پرندگان را که در محیطی به دنبال غذا می‌گردند در نظر بگیرید. هیچ یک از آنها اطلاعی از محل غذا ندارند، ولی در هر مرحله، فاصله خود تا محل غذا را می‌دانند. بر این اساس، بهترین رویکرد برای پیدا کردن غذا، پیروی از نزدیک‌ترین پرنده به غذا می‌باشد. الگوریتم پرواز پرندگان، این رفتار را در مسائل بهینه‌سازی شبیه‌سازی می‌کند [۴۱].

مزایای این الگوریتم عبارتست از:

۱. ریشه در زندگی مصنوعی و هوش محاسباتی دارد.
۲. مفاهیمی ساده دارد.
۳. پارامترهای اندکی دارد.
۴. در مقایسه با الگوریتم ژنتیک، عملگرهای تقاطع و جهش ندارد.
۵. برای حل مسائل گوناگون، مؤثر و قابل اجراست.
۶. اجرای آن ساده است.

معایب این الگوریتم، اندک است:

۱. کاربرد اصلی آن برای حل مسائل نامحدود است، اما با استفاده از روش جریمه می‌توان آن را برای مسائل محدود نیز به‌کار برد.
۲. توانایی کمی در جستجوی محلی دارد [۴۳].

در این الگوریتم، هر پرنده، یک جواب ممکن در فضای جستجوی مسأله می‌باشد. در ابتدا به وسیله گروهی از پرندگان که به طور تصادفی در فضای مسأله تولید شده‌اند، به این الگوریتم مقدار می‌دهیم و سپس جستجو برای رسیدن به بهترین جواب آغاز می‌شود [۴۱]. در هر مرحله از تکرار الگوریتم، پرنده به سمت موقعیت بهتر جابجا می‌شود. موقعیت بعدی برای هر پرنده با توجه به دو مقدار به دست می‌آید: اولین مقدار، بهترین موقعیتی است که پرنده تا کنون داشته است (pbest) و دومین مقدار، بهترین موقعیتی است که همه پرندگان تا کنون به دست آورده‌اند (gbest). به بیان دیگر، gbest را می‌توان بهترین pbest در کل گروه در نظر گرفت. این فرایند

1. Particle Swarm Optimization (PSO)

تا زمان رسیدن به شرط توقف ادامه پیدا می‌کند. شرط توقف در این الگوریتم، میل کردن سرعت پرندگان به سمت صفر یا رسیدن به تعداد تکرارهای در نظر گرفته شده است.
با توجه به مقادیرpbest و gbest، هر پرنده از روابط زیر برای تعیین موقعیت بعدی استفاده می‌کند:

$$v_{ij}(t+1) = w\, v_{ij}(t) + c_1r_1(p_{ij}(t) - x_{ij}(t)) + c_2r_2(g_{ij}(t) - x_{ij}(t)) \quad (1)$$

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

در روابط فوق، ثابت‌های c_1 و c_2، پارامترهای یادگیری (میزان تأثیر) را برای gbest و pbest تعیین می‌کنند (معمولاً ۲). r_1 و r_2 اعدادی تصادفی در محدوده [۰،۱] می‌باشند. xij(t) موقعیت کنونی هر پرنده، vij(t) سرعت حرکت پرندگان در آن مرحله، و w کنترل کننده ضریب اینرسی حرکت ذرات. در ابتدای اجرای الگوریتم، سرعت بیشتر و بعد از مدتی که به پاسخ نزدیک می‌شویم، به کندی کاهش می‌یابد. توابع مورد استفاده برای این منظور معمولاً باعث کاهشی خطی بعد از هر بار تکرار می‌شوند[۳]. شی (۱۹۹۸)، فرمول V را با اضافه کردن پارامتر W اصلاح کرد [۴۴].

الگوریتم پرواز پرندگان گسسته[۱]

کندی و ابرهارت [۳۳] اولین نسخه از الگوریتم پرواز پرندگان گسسته دوارزشی را توسعه دادند. جزئیات بیشتر در مورد ادبیات آن در [۵۵، ۵۶] آمده است.

از آنجا مسأله برش، ماهیتی گسسته دارد و الگوریتم پرواز پرندگان استاندارد قادر به حل چنین مسائلی نمی‌باشد، در این مقاله از الگوریتم پرواز پرندگان گسسته ـ که برای بهینه‌سازی در چنین فضاهایی مناسب می‌باشد ـ استفاده شده است. این الگوریتم به شرح زیر می‌باشد [۴۶]:

هر پرنده، یک جواب ممکن و نشانگر نقطه‌ای در فضای جستجوی چندبعدی است. در ابتدا الگوریتم با مجموعه جواب‌های تصادفی شروع می‌شود و به هر پرنده، به صورت تصادفی، سرعتی نسبت داده می‌شود. سپس با توجه به تابع هدف، مقدار تابع هدف برای هر پرنده محاسبه می‌شود. تابع هدف برای هر مسأله، جداگانه تعریف می‌شود و با توجه به مسأله، ساده یا پیچیده می‌باشد. در هر تکرار، مقدار تابع هدف برای هر پرنده با بهترین مقدار تابع هدف آن پرنده تا به حال (pbesti) مقایسه می‌شود. اگر مقدار فعلی از pbest بهتر باشد، جایگزین pbest می‌شود و در غیر این صورت، pbest باقی می‌ماند. بهترین مقدار تابع هدف در بین تمام pbestها تا به حال، gbest نامیده می‌شود. به همین منوال، اگر pbest پرنده i ام بهتر از gbest باشد، جایگزین gbest می‌شود و شماره پرنده این مقدار (i) ذخیره می‌شود. در غیر این صورت، gbest باقی می‌ماند. در گام بعدی، سرعت هر پرنده با استفاده از فرمول V تعدیل می‌شود. سپس معیار

1. Discrete Particle Swarm Optimization (DPSO)

بهینه‌سازی برای پایان الگوریتم بررسی می‌شود. اگر شرط خاتمه حاصل شده باشد، جواب به دست آمده به عنوان بهترین جواب در نظر گرفته می‌شود و در غیر این صورت، فرایند ادامه می‌یابد.

الگوریتم ابتکاری CUL

در این مقاله، برای هر قطعه i، تعداد مشخصی تقاضا وجود دارد. برای مثال، ۴ واحد تقاضا برای قطعاتی با ابعاد ۵×۲ وجود دارد که باید در ورق اصلی قرار گیرند و فرایند به همین ترتیب برای سایر قطعات ادامه می‌یابد.

گام‌های پیشنهادی برای الگوریتم CUL به صورت زیر است:

گام۱: قطعات را به صورت صعودی بر اساس x_{ip} مرتب کن. (x_{ip} , y_{ip}) مختصات گوشه بالای چپ p امین نسخه از قطعه i در ورق است.

گام۲: اولین قطعه را با توجه به مختصات گوشه بالای چپ در ورق اصلی قرار بده.

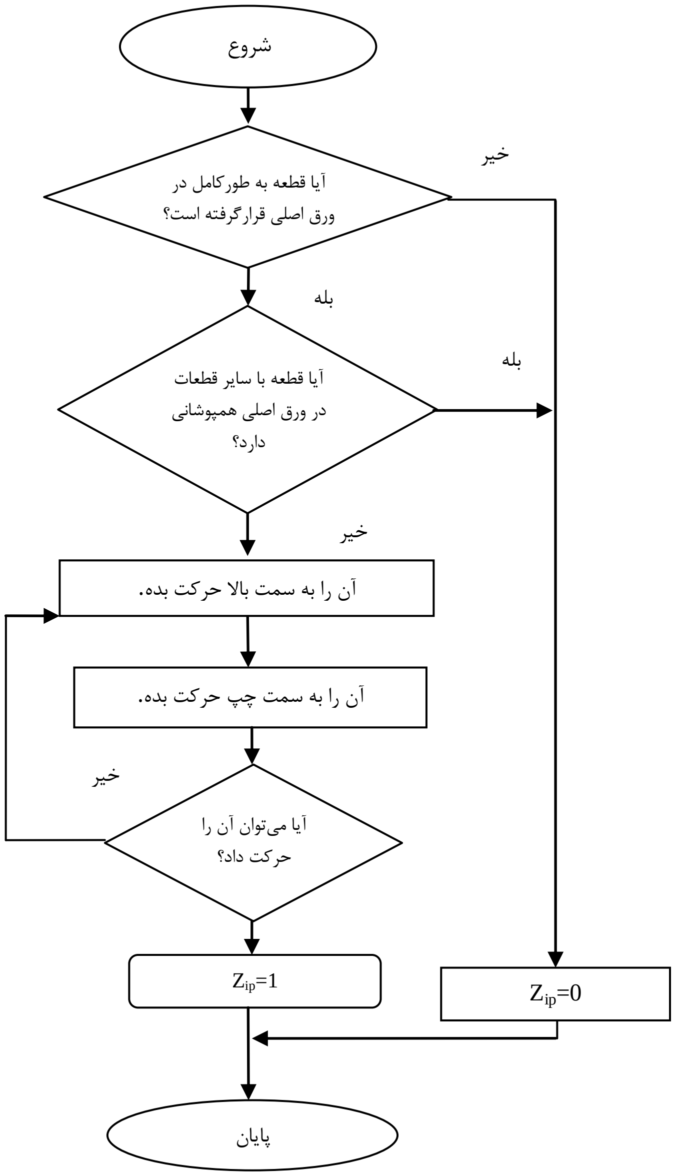
گام۳: حرکت قطعه را بر اساس روش "حرکت یک قطعه" (شکل ۱) برنامه‌ریزی کن.

گام۴: پس از چینش همه قطعات، به گام ۵ برو. در غیر این صورت، قطعه بعدی را بر اساس مختصات گوشه بالای چپ قرار بده. به گام ۳ برو.

گام۵: قطعه $z_{ip}=0$ را در گوشه پایین راست ورق قرار بده. $z_{ip}=0$ یعنی نسخه p ام قطعه i ام در موقعیت (l_0,w_0) قرار نگرفته است. در غیر این صورت $z_{ip}=1$ است.

گام۶: حرکت قطعه را بر اساس روش "حرکت یک قطعه" برنامه‌ریزی کن.

گام۷: اگر تمام قطعات $z_{ip}=0$ چیده شدند، فرایند حل مسأله پایان یافته است. در غیر این صورت، قطعه بعدی $z_{ip}=0$ را در گوشه پایین راست ورق قرار بده. به گام ۶ برو.



شکل ۱. روش "حرکت یک قطعه"

الگوریتم پرواز پرندگان، حرکت پرندگان، حول بهترین پرنده پیاده‌سازی می‌کند و الگوریتم CUL سعی در هدایت بهترین پرنده دارد.

تجزیه و تحلیل داده‌ها

در این مقاله برای حل مسأله برش دوبعدی غیرگیوتینی، پارامترهای پیشنهادی الگوریتم پرواز پرندگان به صورت زیر تعریف می‌شوند:

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

۲- **موقعیت**[2]: موقعیت X شامل طول و عرض قطعات مطابق با توالی قرارگیری آنهاست. این موقعیت، شامل چند ماتریس فرعی (ابعاد ماتریس‌های فرعی، دو برابر تعداد قطعات است) ـ که دربرگیرنده موقعیت هر پرنده هستند ـ است. لذا تعداد سطرهای ماتریس X، دو برابر تعداد سطرهای S می‌باشد.

۳- **سرعت**[3]: برای هر پرنده، به صورت تصادفی، سرعتی ایجاد می‌شود که در حین اجرای الگوریتم، طبق رابطه (۳) به روز می‌شود.

٤- **تابع هدف**: در الگوریتم پیشنهاد شده، هدف عبارتست از حداقل کردن ضایعات ناشی از برش ورق اصلی که پس از کسر بزرگترین مستطیل از قسمت باقی‌مانده محاسبه می‌شود.

۵- **روابط سرعت و موقعیت**: سرعت هر پرنده با توجه به pbest و gbest و با استفاده از رابطه‌ی (۳) تغییر می‌کند. در الگوریتم پرواز پرندگان گسسته، به دلیل وجود متغیرهای گسسته، فرمول V با فرمول V در الگوریتم پرواز پرندگان استاندارد فرق دارد:

$$V_i^{t+1} = W_{iter} \otimes V_i^t \oplus (c_1 * r_1) \otimes (p_{besti} \Theta X_i^t) \oplus (c_2 * r_2) \otimes (p_{gbest} \Theta X_i^t) \quad (۳)$$

c_1 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.

---

1. Swarm
2. position
3. Velocity

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

$$X_i^{t+1} = x_i^t + v_i^{t+1} \quad (۴)$$

در رابطه (۳)، عملگر $\oplus$ بیانگر جمع خاص، عملگر $\otimes$ بیانگر ضرب خاص و عملگر Θ بیانگر تفریق خاص است. توضیح بیشتر در باره این عملگرها در منبع [۴۶] آمده است.

محدودیت‌ها

فرض‌های زیر برای ساده‌سازی در نظر گرفته می‌شوند [۳۷]:

۱- طول و عرض هر قطعه، از بعد متناظر ورق اصلی، نیست.

۲- تمام برش‌ها در ورق اصلی، بسیار بسیار باریک می‌باشند. به عبارت دیگر، لبه‌های قطعات، مساحتی را اشغال نمی‌کنند.

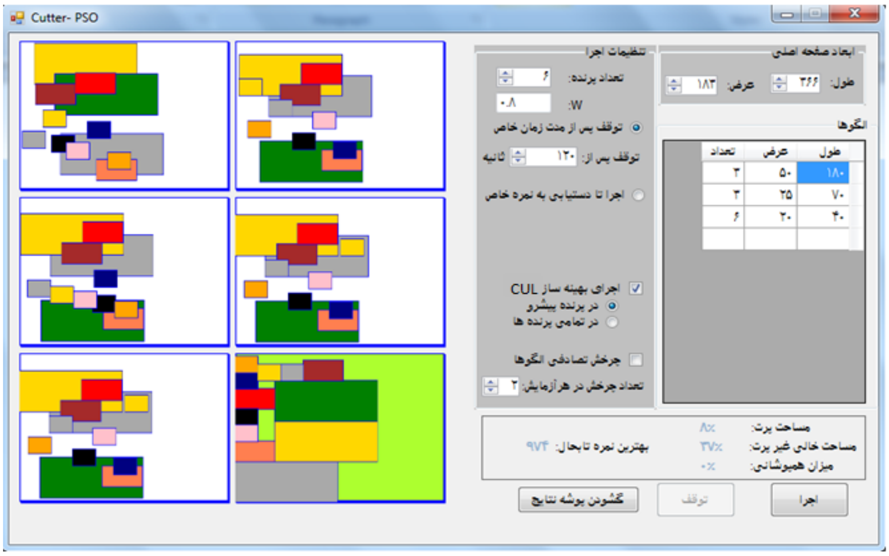
۳- قطعات ممکن است در هر جای ورق اصلی و در مجاورت هر قطعه دیگری قرار گیرند. به عبارت دیگر، هیچ محدودیتی وجود ندارد که دو قطعه نتوانند در کنار قرار گیرند.

برای بررسی الگوریتم پیشنهادی، به کمک زبان برنامه‌نویس C# و بر اساس الگوریتم‌های پرواز پرندگان و CUL، نرم‌افزاری تهیه شد که با در نظر گرفتن طول و عرض صفحه اصلی و با توجه به اندازه‌های مورد تقاضا، بهترین برش ممکن را ارائه می‌دهد. با فرض اینکه اندازه صفحه اصلی ۱۸۳×۳۶۶ باشد و تقاضایی به شکل زیر موجود باشد، الگوی بهینه برش به دست می‌آید:

جدول ۱. داده‌های مسأله برش

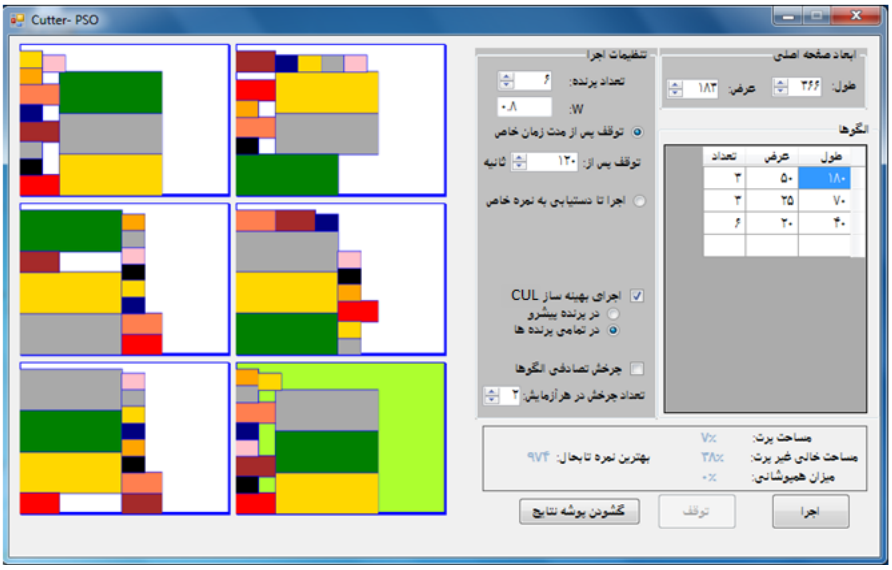
طول	عرض	تعداد مورد نیاز
۱۸۰	۵۰	۳
۷۰	۲۵	۳
۴۰	۲۰	۶

پارامترها در نرم افزار به شکل زیر تنظیم شده‌اند:



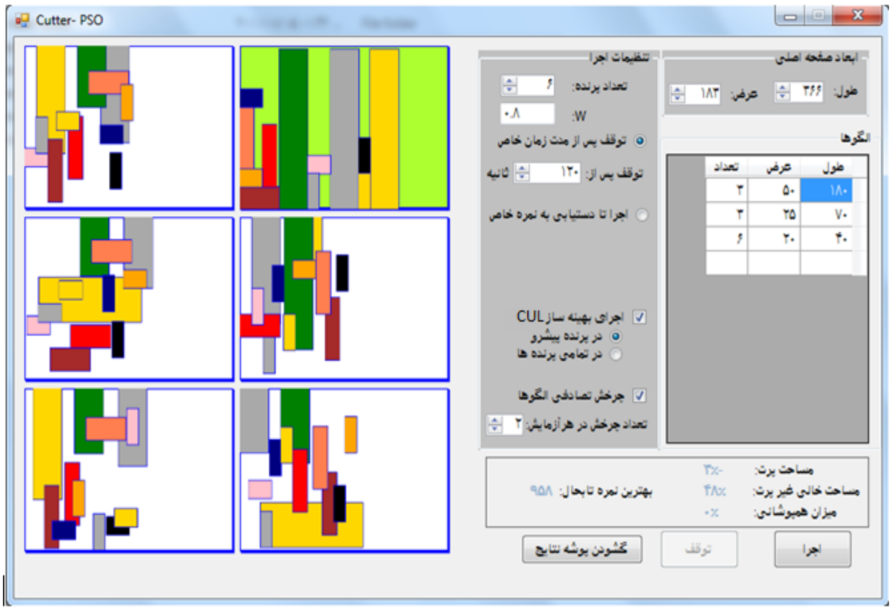
شکل ۲. پارامترهای روش الگوریتم پرواز پرندگان برای در حل مسأله‌ی برش

همان‌طور که در شکل ۲ نشان داده شده است، ۶ پرنده در نظر گرفته شده‌اند و الگوی بهینه برش، همان است که زمینه آن سبز است. لازم به ذکر است که در سایر پرنده‌ها، غیر از بهترین پرنده، همپوشانی وجود دارد، زیرا بهینه‌ساز CUL فقط در مورد پرنده پیشرو اعمال می‌شود. این در حالی است که همین مسأله برای حالتی که CUL برای تمام پرنده‌ها اعمال می‌شود، به شکل زیر خواهد بود:



شکل ۳. اعمال الگوریتم CUL در مورد تمام پرنده‌ها

همان‌طور که در شکل ۳ مشاهده می‌شود، اعمال الگوریتم CUL در مورد تمام پرنده‌ها ضمن اینکه تمام جواب‌ها را به جواب‌های شدنی تبدیل می‌کند و همپوشانی‌ها را از بین می‌برد، زمان پردازش و حل مسأله را افزایش می‌دهد که اجرای CUL برای تمام پرنده‌ها را غیر ضروری می‌سازد. در اعمال الگوریتم می‌توان اجازه چرخش قطعات را ایجاد کرد. در صورتی که اجازه چرخش تصادفی قطعات در همین مسأله داده شود، پرنده‌ها رفتاری به شکل زیر خواهند داشت:



شکل ۴. اعمال الگوریتم پرواز پرندگان با فرض چرخش قطعات

همان‌طور که در شکل ۴ دیده می‌شود، طول و عرض قطعات الزاماً در راستای طول و عرض صفحه‌ی اصلی نیست.

بحث و نتیجه‌گیری

در این مقاله، مسأله برش دوبعدی با تقاضا مورد بررسی قرار گرفت. اکثر محققان به مسأله برش پرداخته‌اند که هدف از تحقیقات آنها کاهش ضایعات ورق، به طور مجزا، بدون در نظر گرفتن تقاضای قطعات است. این در حالی است که اکثر مسائل واقعی برش، به صورت مسئله برش با تقاضا هستند. البته این مسأله در زمره مسائل NP-Hard است و برای حل عملی آن، نیاز به روشی فرا ابتکاری خواهد بود. لذا الگوریتمی فرا ابتکاری در این مقاله ارائه شده است که با استفاده از الگوریتم پرواز پرندگان، به حل مسأله برش پرداخته است.

نتایج محاسبات نشان می‌دهد که استفاده از الگوریتم CUL باعث بهبود عملکرد الگوریتم پرواز پرندگان می‌شود. در این مقاله، با ارائه نرم‌افزاری، کارایی الگوریتم پیشنهادی بررسی شده است. در خاتمه پیشنهاد می‌شود تحقیقات بیشتری برای پیاده‌سازی این الگوریتم (نرم‌افزار) در صنایع مختلف انجام شو. زیرا به نظر می‌رسد در صورت شناسایی شرایط و محدودیت‌های عملی هر صنعت و اعمال آنها در الگوریتم (مانند گیوتینی بودن برش‌ها، دسته‌بندی قطعات، امکان استفاده از ضایعات برای تولید قطعات کوچک‌تر، دخالت دادن هزینه‌های برش در هزینه‌های استفاده از یک ورق و...) هزینه‌های ناشی از برش را می‌توان به نحو مؤثرتری کاهش داد. علاوه بر این، در صورت تحقیق بیشتر در مورد خصوصیات تئوری مطرح در روش‌های فرا ابتکاری، امکان دستیابی به الگوریتمی کاراتر منتفی نخواهد بود.

منابع

۱. مصلحی، قاسم. رضایی، علیرضا. (۱۳۸۳)، "ارائه الگوریتمی برای مسئله برش دوبعدی با تقاضا". استقلال، سال ۲۳، شماره ۲.

2. 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.
3. Alatas, B.; Akin, E. A.; Ozers, B.l. (۲۰۰۷). Chaos-embedded particle swarm optimization algorithms, *Chaos, Solitons and Fractals*.
4. 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.
5. Beasley, J.E. (1985). Algorithms for Unconstraind Two-Dimensional Guillotine Cutting. *Operation's Research Society*.,Vol.36, No.4, PP.297-306.
6. Beasley, J.E. (1985). An Exact Two-Dimensional Non-Guillotine Cutting Tree Search Procedure. *Operation's Research*., Vol.33, No.1, PP.49-64.
7. Beasley, J.W. (۲۰۰۰). A Population Heuristic for Constrained Two-Dimensional Non-Guillotine Cutting. *http://mscmga.ms.ic.ac.uk/jebihtml*.
8. 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.
9. Chauny, F.; Loulou, R.; Sadones, S.; Soumis, F. (1987). A two phase heuristic for the two-phase heuristic for strip packing: algorithm and probabilistic analysis. *Operational Research Letters*, 6, pp. 25–33.
10. 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.
11. Christofides, N.; Whitlock, C. (1977). An algorithm for the two dimensional cutting problem. *Operations Research*., 25, pp. 30–44.
12. Cintra, G.F., et al. (2008). Algorithms for two-dimensional cutting stock and strip packing problems using dynamic programming and column generation. *European Journal of Operational Research*., 191, pp. 61–85.
13. 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.
14. 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.
15. Cui, Y. (2005). Dynamic programming algorithms for the optimal cutting of equal rectangles. *Applied Mathematical Modeling*., 29, pp. 1040-1053.
16. Cui, Y. (2006). Simplest optimal cutting patterns for equal rectangles. *Operations Research Letters,* 34, pp. 630-638.
17. Dietrich, R.D.; Yakowitz, S.J. (1991). A Rule-Based Aproach to The Trim-Loss Problem. *INT.J.PROD.RES*., Vol.29, No.20, PP.401-415.
18. Dowsland, K.A.; Dowsland, W.B. (1992). Packing problems. European Journal of Operational Research., 56, pp. 2-14.

19. Dyckhoff, H. (1990). A typology of cutting and packing problems. *European Journal of Operational Research.*, 44, pp. 145–159.
20. 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.
21. Farley, A.A. (1990). A note on bounding a class of linear programming problems, including cutting stock problems. *Operations Research*., 38, pp. 922–923.
22. Gass, S. (1985). Linear Programming, Methods and Applications. *McGraw-Hill*.
23. Gilmore, P.C.; Gomory, R.E. (1961). A Linear Programming Approach to the Cutting–Stock Problem. *Operations Research*., Vol. 9, pp. 849–859.
24. Gilmore, P.C.; Gomory, R.E. (1965). Multi-stage cutting stock problems of two and more dimensions. *Operations Research*., 13, pp. 94–120.
25. Gradišar, M.; Jesenko, J.; Resinovič, G. (1997). Optimization of roll cutting in clothing industry. *Computer & Operations Research*., 24, pp. 945-953.
26. 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.
27. Hadjiconstantinou, E.; Christofides, N. (1995). An exact algorithm for general, orthogonal, two-dimensional knapsack problems. *European Journal of Operational Research*., 83, pp. 39-56.
28. 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.
29. Haessler, R.W.; Sweeney, P.E. (1991). Cutting stock problems and solution procedures. *European Journal of Operational Research*., 54, pp. 141-150.
30. 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,.
31. 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.
32. Kantorovich, L.V. (1939). Mathematical methods of organizing and planning production. *Mgmt. Sci*.,Vol. , pp.363-422.
33. Kennedy, J.; Eberhart, R. (1995). Particle Swarm Optimization. *in: Proceedings of IEEE International Conference on Neural Networks*, Perth, Australia, pp. 1942-1948.
34. Kopač, J. (2002). Cutting forces and Their Influence on the Economics of Machining. *Strojniški vestnik - Journal of Mechanical Engineering*., 48 3, pp. 121-132.
35. 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.
36. 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.

37. Leung, T. W.; Yung, C. H.; Troutt 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.
38. Nazemi, J.(2005). Kiln Planning a Cutting Stock Approach, *Social Research Research Network.*
39. 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*.
40. Paul, R.J. (1979). A production scheduling problem in the glass container industry. *Operations Research*., Vol 27, Nr. 2, pp. 290–302.
41. Poli, R.; Kennedy, J.(2007). Particle swarm optimization An overview. *Springer Science.*, pp. 33–57.
42. Ratnaweera, A.; Halgamuge, S.K. (2004). Self-Organizing Particle Swarm Optimizer with Time-Varying Acceleration Coefficients. *IEEE Transactions on Evolutionary Computation*., vol. 8, no 3, pp 240-255.
43. Satheesh Kumar, R.M.; Asokan, P.; Kumanan, S. (2007). Design of loop layout in flexible manufacturing system using non-traditional optimization technique. *Springer-Verlag London Limited*.
44. Shi, Y.; Eberhart, R.C. (1998). A modified particle swarm optimizer[C]. *In: IEEE World Congress on Computational Intelligence*., pp. 69-73.
45. Shi, Y.; Eberhart, R.C. (1999). Empirical study of particle swarm optimization. *in: Proc. IEEE Int. Congr. Evolutionary Computation*., vol 3, pp. 101-106.
46. Soke, A.; Bingul, Z. Applications of Discrete PSO Algorithm to Two-Dimensional Non-Guillotine Rectangular Packing Problems. *Conf. on Genetic and Evolutionary Methods.*
47. Suliman, S.M.A.(2006). A sequential heuristic procedure for the two-dimensional cutting-stock problem. *Int. J. Production Economics*., 99, pp. 177–185.
48. 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.
49. 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.
50. 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.
51. Van–dat, C.; Hifi,mhand, and Le cun , bertrand,. (1997). Constrained two dimensional cutting stock problems , a best first branch and bound algorithm. *laborattoire PR&SM–CNRS URA 1525, University of Versailles, paris.*
52. van den, B.F. (2001). An Analysis of Particle Swarm Optimizers. *PhD Thesis, University of Pretoria*.
53. 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.

54. Wang, P.Y. (1983). Two Algorithms for Constraind Two-Dimensional Cutting Stock Problems. *Opns.Res*.,Vol.31,No.3,PP.573-586.
55. 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.
56. 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.
57. 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.
58. Whitlock, C.; Christofides, N. (1977). an Algorithm for Two–Dimensional Cutting Problems. *Operations Research*., Vol. 25, Nr. 1, January–February, pp. 30– 44.
59. Yanasee, H.H.; Zinober, A.S.I.; Harris, R.G. (1991). Two dimensional cutting stock withmultiple stock sizes. *Journal of Operational Research Society*., 42, pp. 673–683.
60. Young-Gun, G., et al. (2003). A best-first branch and bound algorithm for unconstrained two-dimensional cutting problems. *Operation Research Letters*, 31, pp. 301-307.
61. Zak, E.J. (2002). Row and column generation technique for a multistage cutting stock problem. *Computers and Operations Research*., 29, pp. 1143–1156.