

## Optimization the Problem of Packing Rectangular Shapes by using Imperialist Competitive Algorithm

**Motahreh Kargar**

PhD student, Department of Textile Engineering, Yazd University, Yazd, Iran,  
mo.kargar@stu.yazd.ac.ir

**Pedram Payvandy**\*

Assistant professor, Department of Textile Engineering, Yazd University, Yazd, Iran,  
peivandi@yazd.ac.ir

### Abstract:

Packing is one of the well-known problems in operation research, especially in production planning. The main objective of studying the packing problem is to reduce the wastes of cutting through optimization of packing of pieces. Packing is a kind of NP-hard problem that the precise methods are not able to solve it. In this paper, in order to achieve an optimal packing of Non-guillotine cutting problems, the meta-heuristic emerging Imperialist Competitive Algorithm was used and the results were compared with the output of the genetic algorithm, which is the typical algorithm in solving packing problems. To achieve better solutions, the parameters of all meta-heuristics were calibrated with Taguchi experiment method. The efficacy of the proposed approach was tested on a set of instances, taken from the literature, and the results of the proposed algorithm were tested statistically by ANOVA. The results of this study showed that the meta-heuristic emerging Imperialist Competitive algorithm is more efficient and faster in solving packing problems.

**Keywords:** Imperialist Competitive Algorithm, Genetic Algorithm, Packing Algorithms, Optimization, Packing Problems.

**Introduction:** Packing problems are problems which are difficult or sometimes impossible to solve exactly. Researchers have provided many different solutions based on heuristic and meta-heuristic to approximately solve these problems.

**Materials and Methods:** Imperialist Competitive Algorithm is a new evolutionary optimization method which is inspired by imperialistic competition Atashpaz-Gargari (2007). Like other evolutionary algorithms, it starts with an initial population which is called country and is divided into two types of colonies and imperialists which, together, form empires. Imperialistic competition among these empires forms the proposed evolutionary algorithm. During this competition, weak empires collapse and powerful ones take possession of their colonies. Imperialistic competition converges to a state in which there exists only one empire and colonies have the same cost function value as the imperialist. The pseudo code of Imperialist competitive algorithm is as follows:

- 1) Select some random points on the function and initialize the empires.
- 2) Move the colonies toward their relevant imperialist (Assimilation).
- 3) Randomly change the position of some colonies (Revolution).
- 4) If there is a colony in an empire which has lower cost than the imperialist, exchange the positions of that colony and the imperialist.
- 5) Unite the similar empires.
- 6) Compute the total cost of all empires.

---

\* Corresponding author

- 7) Pick the weakest colony (colonies) from the weakest empires and give it (them) to one of the empires (Imperialistic competition).
- 8) Eliminate the powerless empires.
- 9) If stop conditions satisfied, stop, if not go to 2.

After dividing all colonies among imperialists and creating the initial empires, these colonies start moving toward their relevant imperialist state which is based on assimilation policy

**Results and Discussion:** In this study, in order to achieve an optimal packing of non-guillotine cutting problems, at first the meta-heuristic algorithm of Imperialist Competitive Algorithm was used. Then the results were compared with the output of the genetic algorithm which is the typical algorithm in solving packing problems. The results showed that ICA had the better fitness function average than GA. Also, ICA needs less number of function evaluations. Therefore ICA is faster than GA in solving permutation packing problems.

### References

- Atashpaz-Gargari, E., & Lucas, C. (2007). "Imperialist Competitive Algorithm: An algorithm for optimization inspired by imperialistic competition". *IEEE Congress on Evolutionary Computation*, 4661-4667
- Ebrahimi, S., & Payvandy, P. (2013). "Optimization of the Link Drive Mechanism in a Sewing Machine Using Imperialist Competitive Algorithm". *International Journal of Clothing Science and Technology*, 26(3), 247 – 260.
- Bluma, C., & Schmid, V. (2013). "Solving the 2D bin packing problem by means of a hybrid evolutionary algorithm". *Procedia Computer Science*, 18, 899 – 908.

مدیریت تولید و عملیات، دوره ۹، پیاپی ۱۶، شماره ۱، بهار و تابستان ۱۳۹۷

دریافت: ۱۳۹۴/۱۱/۱۵ پذیرش: ۱۳۹۶/۱۱/۱۰

صص: ۱۸۰-۱۶۱

## بهینه‌سازی مسأله چیدمان قطعات منظم مستطیل شکل با استفاده از الگوریتم رقابت استعماری

مطهره کارگریده<sup>۱</sup>، پدram پیوندی<sup>۲\*</sup>

۱- دانشجوی دکترای، دانشکده نساجی، دانشگاه یزد، یزد، ایران، mo.kargar@stu.yazd.ac.ir

۲- استادیار، دانشکده نساجی، دانشگاه یزد، یزد، ایران، peivandi@yazd.ac.ir

**چکیده:** چیدمان یکی از مسائل شناخته‌شده در حوزه تحقیق در عملیات به‌ویژه در زمینه برنامه‌ریزی تولید است. هدف اصلی بررسی مسأله چیدمان، کاهش ضایعات ناشی از برش با استفاده از بهینه‌چینی قطعات است. مسائل چیدمان از نوع مسائل *ان‌پی-سخت* هستند که روش‌های دقیق قادر به حل آنها نیستند. برای بهینه‌سازی این نوع مسائل، در مقاله حاضر از الگوریتم نوظهور فرا ابتکاری رقابت استعماری استفاده و نتایج آن با نتایج الگوریتم ژنتیک مقایسه شده است. برای دستیابی به نتیجه بهتر، پارامترهای اولیه الگوریتم فرا ابتکاری با روش طراحی آزمایش‌های تاکوچی تنظیم شده است. کارآیی روش پیشنهادی با استفاده از مجموعه‌ای از مسائل معیار مطرح در این زمینه ارزیابی و کیفیت آن با استفاده از روش آماری ANOVA آزمون شده است. نتایج این پژوهش نشان می‌دهد الگوریتم رقابت استعماری، الگوریتمی کارآمدتر و سریع‌تر در حل این نوع مسائل است.

**واژه‌های کلیدی:** الگوریتم رقابت استعماری، الگوریتم ژنتیک، الگوریتم چیدمان، بهینه‌سازی، مسائل چیدمان

## مقدمه

در فرایند تولید بسیاری از صنایع نیاز است تا قطعات کوچک‌تر با برش از اجسام بزرگ‌تر حاصل یا به‌طور معادل، قطعه‌های کوچک‌تر در جسمی بزرگ‌تر جای داده شوند. در این عمل معمولاً بخش‌هایی از جسم بزرگ‌تر به قطعاتی بدون استفاده تبدیل و ضایعات و دورریز محسوب می‌شوند. کاهش چنین ضایعاتی نقش مهمی در کاهش هزینه‌ها دارد. مسأله چیدمان یکی از موضوعات علم تحقیق در عملیات است و توجه بسیاری از پژوهشگران را به خود جلب کرده است. در بسیاری از صنایع از جمله صنایع تولیدکننده بدنه خودرو، سازه‌های ساختمانی، کشتی و هواپیماسازی، کاغذ، پوشاک، چرم، سنگ‌بری و در فعالیت‌هایی مانند حمل‌ونقل، انبارداری و بسته‌بندی، مسأله چیدمان و یا مسائل مشابه آن مشاهده می‌شود.

تعداد الگوهایی که دسته‌ای از قطعات می‌توانند در کنار یکدیگر قرار دهند بسیار زیاد است و تعداد کمی از این الگوها ضایعات کم تولید می‌کنند؛ بنابراین استفاده از روال عمومی ریاضی برای حل این نوع مسائل امکان‌پذیر نیست. از دهه ۵۰ میلادی که رایانه‌ها به‌عنوان پردازنده‌های سریع و اقتصادی داده‌ها ظاهر شدند، به مسأله چیدمان توجه بیشتری شده است. در طی سال‌ها، دو روش اصلی اکتشافی و تقریب بر پایه برنامه‌ریزی خطی برای حل این نوع مسائل ارائه شد (هوپر و تورتون<sup>۱</sup>، ۲۰۰۱). استفاده از برنامه‌ریزی خطی بیشتر محدود به مسائل چیدمان یک‌بعدی یا دوبعدی با تعداد قطعات کم است. بیشتر توجه پژوهشگران به علت بزرگ‌بودن فضای جواب مسأله چیدمان، معطوف به روش‌های ابتکاری و فرا ابتکاری حل این‌گونه مسائل است؛ به‌نحوی که توانایی جستجوی هوشمندانه فضای جستجو را دارند (دیکهوف<sup>۲</sup>، ۱۹۹۰).

در روش‌های گوناگون حل مسائل چیدمان، از انواع الگوریتم‌های ابتکاری جایگذاری برای تبدیل مجموعه‌ای از قطعات به چیدمان معتبر استفاده می‌شود. چیدمان معتبر، چیدمانی است که در آن قطعات هیچ‌گونه هم‌پوشانی با یکدیگر ندارند و به‌طور کامل در داخل شیء مورد نظر قرار می‌گیرند. الگوریتم‌های ابتکاری جایگذاری با قراردادن گام به گام قطعات یا جابه‌جایی قطعات جایگذاری شده روی شیء، مطابق قانون‌های خاص، برای تولید چیدمان معتبر استفاده می‌شوند.

بورک<sup>۳</sup> و همکارانش (۲۰۰۶)، لیو و تنگ<sup>۴</sup> (۱۹۹۹)، چازل<sup>۵</sup> (۱۹۸۳) و اولیور و فریرا<sup>۶</sup> (۱۹۹۳) از جمله افرادی هستند که روش‌هایی ابتکاری برای چیدمان انواع قطعات ارائه کرده‌اند.

لئونگ<sup>۷</sup> و همکارانش (۲۰۱۲) روشی ابتکاری براساس جستجوی ممنوعه مبتنی بر برنامه‌ریزی غیرخطی برای حل مسأله چیدمان دوبعدی نامنظم ارائه کرده‌اند. در روش ابتکاری که اوزکان<sup>۸</sup> (۲۰۱۳) معرفی کرده است، برخلاف روش‌های دیگر در هر مرحله، محل چیدمان یک جفت قطعه به‌جای یک قطعه تعیین می‌شود.

الگوریتم‌های جستجوی فرا ابتکاری با استفاده از استراتژی‌های خاص مبتنی بر استفاده از نتایج موجود با الهام‌گرفتن از فرایندهای طبیعی یا فیزیکی توانایی پیدا کردن چیدمان مناسب، مؤثر و کارآمد را دارند. الگوریتم ژنتیک یکی از الگوریتم‌های فرا ابتکاری است که برای حل مسائل چیدمان بسیار استفاده شده است. پژوهشگرانی مانند فالكنیر و فالكنوار و دلچیمبر<sup>۹</sup> (۱۹۹۲)، هوانگ<sup>۱۰</sup> و همکارانش (۱۹۹۴)، رونارسون<sup>۱۱</sup> و همکارانش (۱۹۹۶)، داگلی و پوشیانندا<sup>۱۲</sup> (۱۹۹۷)، لیو و تنگ (۱۹۹۹)، هوپر و تورتون (۲۰۰۲)، ولنزویلا و وانگ<sup>۱۳</sup> (۲۰۰۱)، بورتفلد<sup>۱۴</sup>

(۲۰۰۶)، هیفی و هالاه<sup>۱۵</sup> (۲۰۰۳) و بورک و همکارانش (۲۰۰۶) مطالعاتی روی الگوریتم ژنتیک (ابزاری برای حل مسائل چیدمان) انجام داده‌اند.

الگوریتم انجماد تدریجی یکی دیگر از روش‌های فرا ابتکاری است که در حل مسائل چیدمان به کار گرفته شده است. داوسلند<sup>۱۶</sup> (۱۹۹۳)، لیا و چان<sup>۱۷</sup> (۱۹۹۷)، فاینبا<sup>۱۸</sup> (۱۹۹۹)، بیسیگل<sup>۱۹</sup> و همکارانش (۲۰۰۵)، از جمله پژوهشگرانی هستند که از الگوریتم انجماد تدریجی برای حل مسائل چیدمان استفاده کرده‌اند.

سومین الگوریتم فرا ابتکاری که در حل مسائل چیدمان از آن استفاده می‌شود، الگوریتم جستجوی ممنوعه است. از جمله پژوهشگرانی که عملکرد الگوریتم جستجوی ممنوعه را برای حل مسائل چیدمان ارزیابی کرده‌اند عبارت‌اند از لودی<sup>۲۰</sup> و همکارانش (۲۰۰۴)، وی<sup>۲۱</sup> و همکارانش (۲۰۱۱)، آوارزوالد<sup>۲۲</sup> و همکارانش (۲۰۰۷)، بورک و همکارانش (۲۰۰۶) و پورزا و مورابیتو<sup>۲۳</sup> (۲۰۰۶).

شاین و کیتا<sup>۲۴</sup> (۲۰۱۲) از الگوریتم فرا ابتکاری ازدحام ذرات و وانگ<sup>۲۵</sup> (۲۰۱۰) از الگوریتم ژنتیک انطباقی برای حل مسائل چیدمان دوبعدی استفاده کرده‌اند.

مسأله چیدمان بین، نوعی مسأله چیدمان است که در آن قطعات باید روی تعدادی شیء با طول و عرض ثابت چیده شوند.

آمارو<sup>۲۶</sup> و همکارانش (۲۰۱۳) و جونور<sup>۲۷</sup> و همکارانش (۲۰۱۳) سعی در حل مسأله چیدمان دوبعدی قطعات نامنظم با استفاده از ترکیب الگوریتم ژنتیک با الگوریتم‌های چیدمان، ابتکاری مختلف داشتند.

دوسلند<sup>۱۶</sup> (۱۹۹۳) از جمله اولین پژوهشگرانی بود که از روش‌های فرا ابتکاری برای حل مسائل چیدمان قطعات با اشکال منظم استفاده کرد. الگوریتم انجماد تدریجی دوسلند هر دو دسته از راه‌حل‌های معتبر و نامعتبر را بررسی و راه‌حلی با کمترین هم‌پوشانی را جستجو می‌کند. جیکوب<sup>۲۸</sup> (۱۹۹۶) روش فرا ابتکاری برای حل مسائل چیدمان دوبعدی قطعات منظم مستطیل شکل و نامنظم ارائه کرد. وی مسأله چیدمان را مسأله جایگشتی در نظر گرفته و از الگوریتم ژنتیک برای پیدا کردن دنباله‌ای مناسب از قطعات برای جایگذاری و از الگوریتم جایگذاری پایین-چپ برای تبدیل دنباله قطعات به یک چیدمان معتبر استفاده کرد. داگلی و پوشیانندا<sup>۲۹</sup> (۱۹۹۷) از روش دومرحله‌ای براساس الگوریتم شبکه‌های عصبی و الگوریتم ژنتیک برای حل مسأله چیدمان دوبعدی قطعات با اشکال مستطیل شکل استفاده کردند. روش داگلی و پوشیانندا قادر است چیدمان‌هایی با بازدهی ۹۵٪ تا ۹۷٪ تولید کند.

سوک و بینگل<sup>۳۰</sup> (۲۰۰۶) دو روش جدید برای حل مسائل چیدمان دوبعدی معرفی کردند. آنان همانند روش جیکوب مسأله چیدمان را مانند مسأله جایگشتی مدل‌سازی کردند و از الگوریتم‌های ژنتیک و انجماد تدریجی برای پیدا کردن بهترین دنباله از قطعات و از الگوریتم جایگذاری پایین-چپ-جهشی برای تبدیل دنباله قطعات به چیدمانی معتبر استفاده کردند. لی<sup>۳۱</sup> و همکارانش (۲۰۰۹) روشی مشابه روش سوک و بینگل با استفاده از الگوریتم جایگذاری تنظیم‌کننده ارتفاع و الگوریتم ژنتیک ارائه کردند. وانگ (۲۰۱۰) روشی براساس ترکیب دو الگوریتم فرا ابتکاری ژنتیک و انجماد تدریجی برای حل مسائل چیدمان دوبعدی قطعات منظم ارائه کرد. در این بخش تنها روش‌های ابتکاری و فرا ابتکاری حل مسائل چیدمان منظم دوبعدی تشریح شد. برای مطالعات بیشتر درباره انواع روش‌های ابتکاری و فرا ابتکاری حل مسائل چیدمان مختلف، به مطالعه کارگر و پیوندی (۱۳۹۳) مراجعه شود.

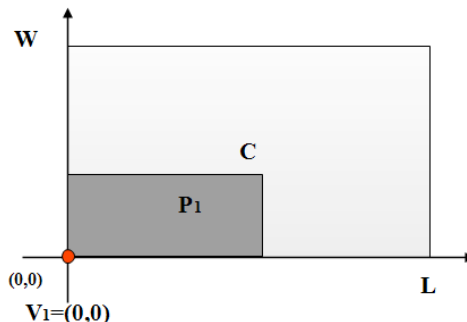
در این پژوهش مسأله برش قطعات منظم مستطیل بررسی و حل شده است. ورودی این نوع مسائل، دنباله‌ای محدود از مستطیل‌هایی با ابعاد مشخص (قطعات) است که باید بدون هم‌پوشانی داخل فضایی مستطیل شکل (شیء) به‌عرض مشخص و طول نامحدود جایگذاری شوند. هدف از حل مسأله، یافتن دنباله‌ای از قطعات است که کمترین طول فضای مستطیل شکل را اشغال کنند.

با بررسی روش‌های حل مسائل چیدمان مشاهده می‌شود نوع الگوریتم فرا ابتکاری استفاده‌شده نقش به‌سزایی در دستیابی به جواب بهینه دارد. در این راستا به بررسی کارایی الگوریتم‌های فرا ابتکاری نوظهور در حل این نوع مسائل توجه بسیاری شده است؛ از جمله این الگوریتم‌ها، الگوریتم رقابت استعماری<sup>۳۳</sup> است. آتش‌پز و لوکاس، الگوریتم رقابت استعماری را ابداع کرده و در کاربردهای گوناگون توانمندی خود را نسبت به الگوریتم‌های فرا ابتکاری دیگر به اثبات رسانده است (ابراهیمی و پیوندی، ۲۰۱۳) و (اورتمن<sup>۳۳</sup>، ۲۰۱۰). در این پژوهش نیز به حل مسأله چیدمان از این روش توجه بسیاری شده است و نتایج حاصله با الگوریتم معمول و پذیرفته‌شده در این نوع مسائل یعنی الگوریتم ژنتیک مقایسه شد. در این پژوهش برای کارایی بهتر الگوریتم‌های فرا ابتکاری، پارامترهای الگوریتم‌های فرا ابتکاری به‌روش تاگوچی تنظیم شده است.

### شرح مسأله

مسائل چیدمان شامل پیدا کردن آرایشی مناسب از چیدمان مجموعه‌ای محدود از قطعات در یک یا چند شیء مشخص است و هدف رایج این فرایند، به حداکثر رساندن بهره‌برداری و به حداقل رساندن "دورریز" مواد اولیه، در کمترین زمان ممکن است. در فرایند چیدمان باید اطمینان حاصل شود که هیچ هم‌پوشانی بین قطعات وجود ندارد (لینز<sup>۳۴</sup> و همکاران، ۲۰۰۳).

مجموعه‌ای از چندضلعی‌ها  $P=(P_1, P_2, \dots, P_n)$  و شیء مستطیل شکل  $C=C(W, L)$  با عرض ثابت  $W$  و طول متغیر  $L$  را در نظر بگیرید. پایین‌ترین و چپ‌ترین رأس شیء  $C$  در نقطه مبدأ  $(0, 0)$  قرار دارد و موقعیت چندضلعی  $P_i$  از طریق مختصات نقطه مرجع آن  $V_i=(x_i, y_i)$  نسبت به مبدأ شیء  $C$  بیان می‌شود. نقطه مرجع یکی از رئوس یا مرکز ثقل چندضلعی است که در این مسأله، پایین‌ترین و چپ‌ترین رأس چندضلعی در نظر گرفته شده است. شکل ۱ نمایی از مسأله چیدمان نشان می‌دهد. در این مسأله فرض شده است که قطعات اجازه دوران ندارند.



شکل ۱- مدلی از مسأله چیدمان

مسأله چیدمان چندضلعی‌های  $P_i$  روی شیء  $C$  با توجه به شکل ۱ به صورت مدل ذیل شرح داده می‌شود:

Minimize L

s.t.

$$\emptyset = (P_i \oplus V_i) \cap (P_j \oplus V_j) \quad (i \leq n \geq 1)$$

$$(P_i \oplus V_i) \subseteq C(W, L) \quad (i \leq n \geq 1)$$

$$V_i \in R^2 \quad (i \leq n \geq 1)$$

$$L \geq 0$$

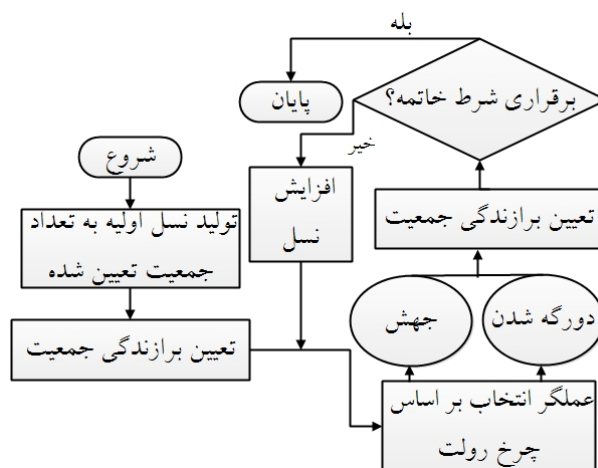
منظور از  $i$  ترتیب قطعات برای جایگذاری است.

در پژوهش حاضر هدف حداقل کردن طول  $L$  از شیء به کمک یافتن بهترین دنباله از ترتیب قطعات برای جایگذاری روی شیء است؛ به این منظور از الگوریتم فرا ابتکاری رقابت استعماری استفاده می‌شود. این الگوریتم، الگوریتم جستجوی نوین و قدرتمندی است. در این روش فرض می‌شود فضای جستجوی شامل ترتیب‌های مختلف از قطعات برای جایگذاری روی شیء است. الگوریتم فرا ابتکاری استفاده شده در طول چند دهه اجرا روی فضای جستجو به سمت دنباله‌ای همگرا می‌شود که شیء با طول کمتر تولید کند.

### الگوریتم‌های فرا ابتکاری

#### الگوریتم ژنتیک

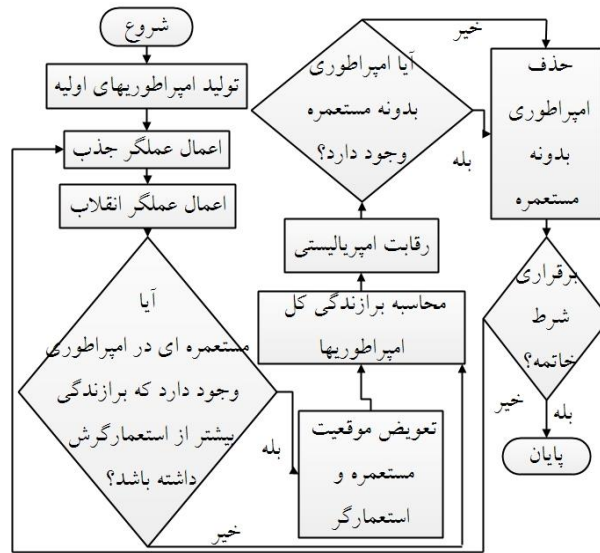
الگوریتم ژنتیک<sup>۳۵</sup> روشی برای جستجوی تصادفی عددی است که از فرآیند ساده‌شده تکامل طبیعی تقلید می‌کند. الگوریتم روی جمعیتی از پاسخ‌ها عمل کرده و با به کار بردن اصل بقای بهترین و تکامل، جواب‌های بهتر و مناسب‌تر ایجاد می‌کند. جزئیات روش الگوریتم ژنتیک استفاده شده در این مقاله، به صورت فلوجارت در شکل ۲ نشان داده شده است (میشل<sup>۳۶</sup>، ۲۰۰۵).



شکل ۲- فلوجارت الگوریتم ژنتیک

### الگوریتم رقابت استعماری

این الگوریتم به فرایند استعمار، به عنوان مرحله‌ای از تکامل اجتماعی-سیاسی بشر نگرینده و با مدل‌سازی ریاضی این پدیده تاریخی، از آن به عنوان منشأ الهام الگوریتم قدرتمند در زمینه بهبودسازی بهره می‌گیرد. فلوجارت الگوریتم رقابت استعماری در شکل ۳ نشان داده شده است (ابراهیمی و پیوندی، ۲۰۱۳).



شکل ۳- فلوچارت الگوریتم رقابت استعماری

در ابتدای اجرای الگوریتم، باید کشورها به صورت تصادفی تولید شوند. در الگوریتم بهینه‌سازی، هدف یافتن جواب بهینه برحسب متغیرهای مسأله است. در یک مسأله بهینه‌سازی  $N_{var}$  بعدی، یک کشور آرایه‌ای است که به صورت زیر تعریف می‌شود:

$$country = [p_1, p_2, p_3, \dots, p_{N_{var}}] \quad (1)$$

$p_i (i=1, 2, 3, \dots, N_{var})$  نشان‌دهنده متغیرهای بهینه‌سازی است. بهینه‌سازی بعد از یافتن حداقل مقدار هزینه به پایان خواهد رسید. در این مرحله تعدادی از کشورهای قوی اولیه، استعمارگر در نظر گرفته می‌شوند. باقیمانده کشورها مستعمراتی را تشکیل می‌دهند که هرکدام متعلق به یک امپراطوری هستند. چگونگی شکل‌گیری امپراطوری‌های اولیه در شکل ۴ نشان داده شده است. مطابق شکل ۴ استعمارگران قویتر، مستعمرات بیشتری به خود اختصاص داده‌اند.



شکل ۴- استعمارگران اولیه تولیدشده و مستعمرات آنها

امپریالیست‌ها سعی دارند با اعمال سیاست جذب، کشورهای مستعمره را در راستای ابعاد مختلف اجتماعی - سیاسی به خود نزدیک کنند. این بخش از فرایند استعمار در الگوریتم بهینه‌سازی به صورت حرکت مستعمرات

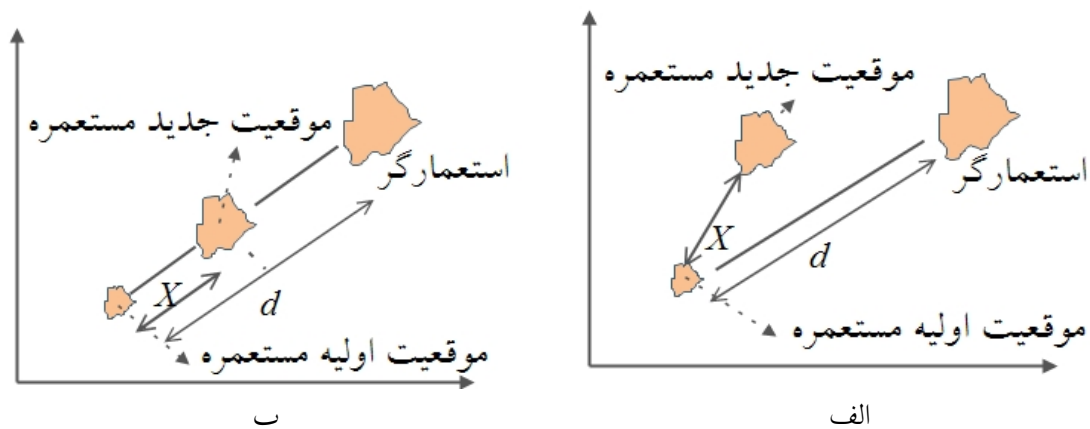


به سمت کشور امپریالیست، مدل شده است. شکل (۵-الف) شمای کلی این حرکت را نشان می دهد. همان گونه که در این شکل نشان داده شده است، کشور مستعمره<sup>۳۷</sup> به اندازه  $x$  واحد در جهت خط واصل مستعمره به استعمارگر<sup>۳۸</sup> حرکت کرده و به موقعیت جدید کشانده می شود.

در این شکل، فاصله میان استعمارگر و مستعمره با  $d$  نشان داده شده است.  $x$  نیز عددی تصادفی با توزیع یکنواخت (یا هر توزیع مناسب دیگر) است که توزیع  $x$  به صورت رابطه ۲ نمایش داده می شود.

$$x \sim U(0, \beta \times d) \quad (2)$$

در این رابطه،  $\beta$  ضریبی بزرگتر از یک و  $d$  فاصله بین استعمارگر و مستعمره است. وجود ضریب  $\beta > 1$  باعث می شود کشور مستعمره در حین حرکت به سمت کشور استعمارگر، از جهت های مختلف به آن نزدیک شود.



شکل ۵- الف: حرکت مستقیم مستعمره به سمت استعمارگر مربوطه. ب: حرکت مستعمره به سمت استعمارگر با انحراف زاویه ای.

در سیاست جذب، حرکت مستعمره به سمت استعمارگر مربوطه، لزوماً مستقیم نیست. مستعمره به صورت غیرمستقیم، همراه با انحراف به سمت استعمارگر مربوطه نزدیک می شود. این انحراف احتمالی با افزودن زاویه ای تصادفی به مسیر جذب مستعمرات انجام می گیرد. شکل (۵-ب) این حالت را نشان می دهد. بدین منظور به جای حرکت به اندازه  $x$  به سمت کشور استعمارگر و در جهت بردار واصل مستعمره به استعمارگر، به همان میزان ولی با انحراف  $\theta$  در مسیر طی می کند. حرکت  $\theta$  را به صورت تصادفی و با توزیع یکنواخت یا هر توزیع دلخواه و مناسب دیگر نیز می توان استفاده کرد؛ بنابراین:

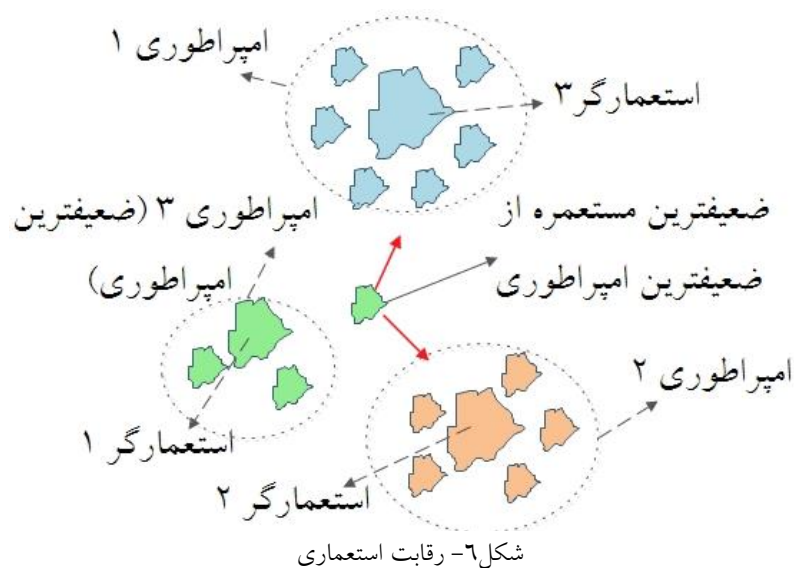
$$\theta \sim (-\lambda, \lambda) \quad (3)$$

در این رابطه  $\lambda$  پارامتری دلخواه است که افزایش آن باعث افزایش جستجوی اطراف امپریالیست می شود و کاهش آن نیز باعث می شود تا مستعمرات تا حد ممکن به بردار واصل مستعمره به استعمارگر نزدیک حرکت کنند.

انقلاب، عملیاتی اساسی در الگوریتم و به معنای جابجایی تصادفی یک مستعمره به یک موقعیت جدید تصادفی است. انقلاب در الگوریتم از به دام افتادن پاسخ‌ها در بهینه‌های محلی جلوگیری و فضای جستجوی بیشتری را فراهم می‌کند.

در حین حرکت مستعمره به سمت استعمارگر، ممکن است بعضی از مستعمرات به موقعیت بهتری از استعمارگر برسند (هزینه کمتر از هزینه استعمارگر). در این حالت جای مستعمره و استعمارگر عوض می‌شود و الگوریتم با موقعیت استعمارگر جدید ادامه می‌یابد.

قدرت یک امپراطوری برابر است با قدرت کشور استعمارگر به اضافه درصدی از میانگین قدرت مستعمرات آن. هر امپراطوری که از قدرتش کاسته شود، در جریان رقابت‌های امپریالیستی حذف خواهد شد. بدین صورت که یک یا چند مستعمره ضعیف از ضعیف‌ترین امپراطوری‌ها حذف و در تصاحب سایر امپراطوری‌ها در خواهد آمد. لزوماً قوی‌ترین امپراطوری مستعمرات مذکور را تصاحب نخواهد کرد؛ بلکه امپراطوری‌های قویتر، احتمال تصاحب بیشتری دارند. برای مدل‌کردن این واقعیت، فرض می‌شود امپراطوری در حال حذف، ضعیف‌ترین امپراطوری موجود است. بدین ترتیب، برای تصاحب مستعمرات ضعیف، رقابتی میان کلیه امپراطوری‌ها ایجاد می‌شود. شکل ۶ استراتژی رقابت ذکر شده را نشان می‌دهد.



### الگوریتم رقابت استعماری گسسته

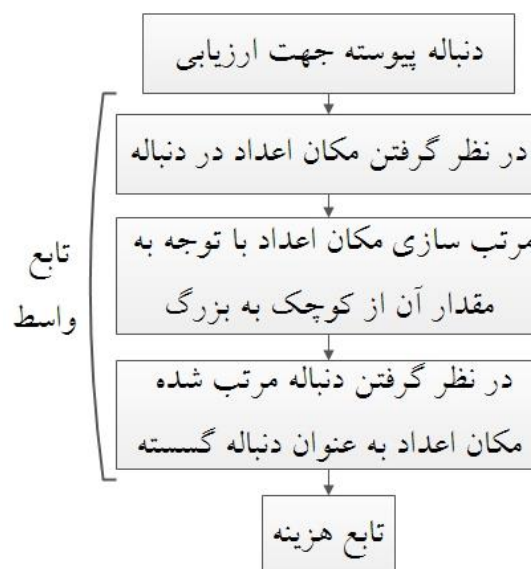
الگوریتم اولیه که آتش‌پز و لوکاس<sup>۳۹</sup> (۲۰۰۷) ارائه کرده‌اند قابلیت اجرا روی مسائل پیوسته را دارد. در مسائل بهینه‌چینی قطعات هدف یافتن دنباله‌ای از ترتیب قطعات برای تولید پاسخ بهینه است؛ بنابراین مسأله از انواع مسائل جایگشتی است. از آنجایی که مسائل جایگشتی از نمونه بارز مسائل گسسته هستند، استفاده از الگوریتم رقابت استعماری معمول برای حل این گونه مسائل امکان‌پذیر نیست. راهکار مناسب برای حل این موضوع تبدیل متغیرهای گسسته به متغیرهای پیوسته در طول اجرای الگوریتم است. در این راهکار فرض می‌شود هدف بهینه‌سازی مسأله پیوسته است و تمام عملیات برای مسأله پیوسته انجام می‌شوند؛ بنابراین الگوریتم رقابت استعماری

روی مجموعه‌ای از متغیرهای پیوسته عمل و متغیرهای بهینه‌سازی را تنها در مرحله ارسال به تابع هزینه، گسسته می‌کند. برای این منظور استفاده از یک تابع واسط در مرحله ارسال متغیرها به تابع هزینه لازم است. تابع واسط متغیر پیوسته را هنگام ارزیابی میزان هزینه، گسسته و به تابع هزینه مسأله بهینه‌سازی ارسال می‌کند. سپس این مقدار هزینه به الگوریتم رقابت استعماری برای ادامه عملیات برگردانده می‌شود. در اتمام اجرای الگوریتم رقابت استعماری مقادیر پاسخ‌های بهینه تولیدشده در الگوریتم، پیوسته هستند. این مقادیر پیوسته با تابع واسط، گسسته و به‌عنوان جواب اصلی مسأله گسسته نمایش داده می‌شوند.

در این مقاله برای حل مسأله بهینه‌چینی قطعات با استفاده از الگوریتم رقابت استعماری، ابتدا برای تولید کشورها به تعداد  $N\_var$  اعداد حقیقی تصادفی بین بازه صفر تا یک تولید و به تعداد کشور مورد نیاز تکرار می‌شوند. سپس از کشورهای تولیدشده برای استفاده در الگوریتم رقابت استعماری استفاده می‌شود.

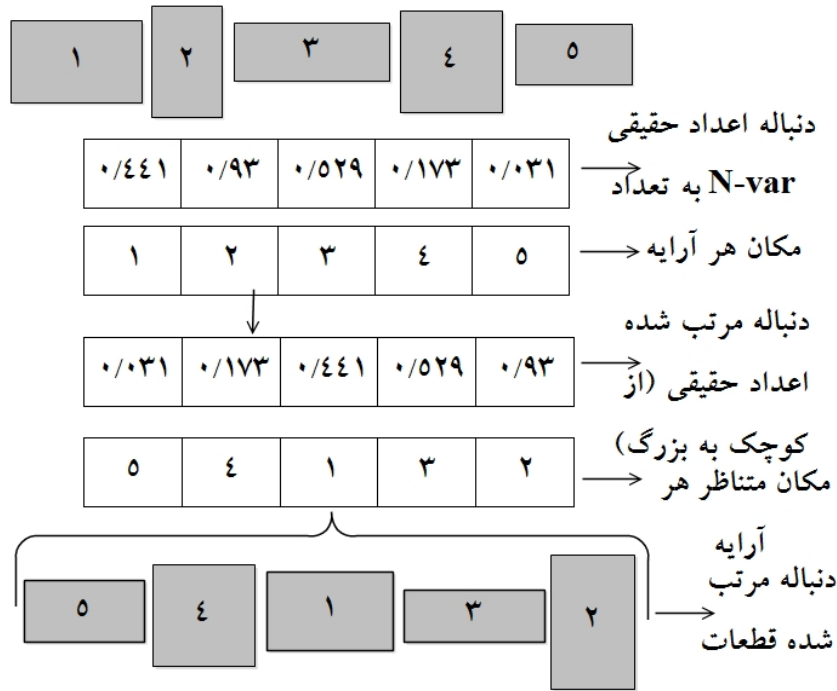
در مرحله ارسال متغیرها به تابع هزینه، از مکان مرتب‌شده (از کوچک به بزرگ و یا برعکس) هر کشور به‌عنوان مقادیر آن برای ارزیابی قدرت هر کشور استفاده می‌شود (تابع واسط پیشنهادی برای تبدیل دنباله اعداد پیوسته به دنباله اعداد گسسته).

در واقع با استفاده از این روش الگوریتم رقابت استعماری روی اعداد حقیقی اعمال می‌شود؛ ولی هنگام ارزیابی از مکان مرتب‌شده اعداد حقیقی داخل هر آرایه استفاده می‌شود که خصوصیات لازم در مسائل گسسته مانند عدد طبیعی بودن و تکرارناپذیری را دارد. شکل ۷ فلوچارت گسسته‌سازی متغیرها و انتقال آنها به تابع هزینه را نشان می‌دهد.



شکل ۷- فلوچارت گسسته‌سازی متغیرها قبل از انتقال به تابع هزینه

شکل ۸ مراحل تبدیل دنباله اعداد حقیقی به دنباله قطعات (توسط تابع واسط) را نشان می‌دهد.

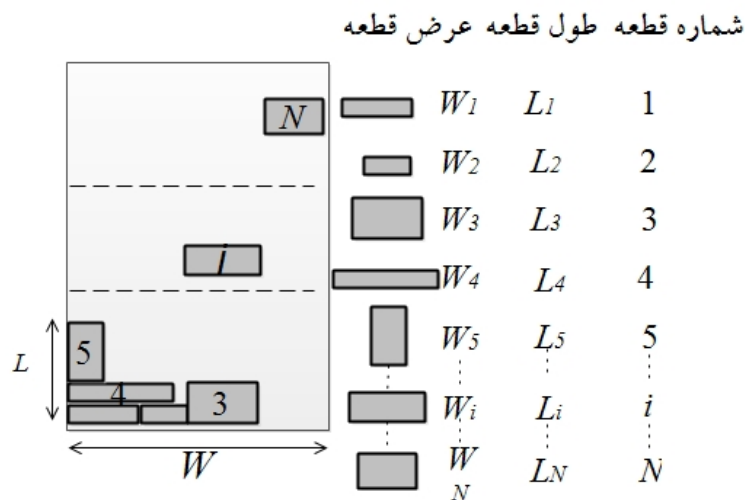


شکل ۸ - مراحل تبدیل دنباله اعداد حقیقی به دنباله قطعات

### الگوریتم‌های ابتکاری جایگذاری

برای محاسبه تابع هدف در طول اجرای الگوریتم جستجوی فرا ابتکاری، لازم است مجموعه قطعات مطابق ترتیب یافت شده با الگوریتم جایگذاری روی شی جایگذاری شوند و بازدهی شی به عنوان تابع هدف محاسبه شود. همان‌طور که در شکل ۹ مشاهده می‌شود، مسأله چیدمان شامل جایگذاری  $N$  قطعه  $(i = 1, 2, 3, 4, 5, \dots, N)$  با عرض  $w_i$  و طول  $l_i$  در فضای مستطیل شکل به عرض  $W$  است. در صورتی که فضای طولی اشغال شده در چیدمان برابر  $L$  باشد، بازدهی چیدمان از رابطه ۴ محاسبه می‌شود.

$$Efficiency = \frac{\sum_{i=1}^N l_i w_i}{L \times W} \quad (4)$$



شکل ۹ - نمونه‌ای از چیدمان  $i$  قطعه  $(i = 1, 2, 3, 4, 5, \dots, N)$  در عرض  $W$

روش‌های زیادی برای جایگذاری قطعات پیشنهاد شده است. این روش‌ها شامل، روش‌های الگوریتم جایگذاری پایین-چپ، الگوریتم پایین-چپ-جاذبه‌دار، الگوریتم پایین-چپ-جهشی، الگوریتم پایین-چپ-توسعه‌یافته و الگوریتم بهترین تناسب هستند.

در این پژوهش، ابتدا بین عملکرد و سرعت اجرای پنج الگوریتم ابتکاری جایگذاری پایین-چپ (BL)، الگوریتم پایین-چپ-جاذبه‌دار (BLLT)، الگوریتم پایین-چپ-جهشی (BLF)، الگوریتم پایین-چپ-توسعه یافته (BLD) و الگوریتم بهترین تناسب (BF)، مقایسه‌ای انجام شده است. سپس الگوریتم جایگذاری بهینه برای ترکیب با الگوریتم‌های فرا ابتکاری انتخاب شده‌اند.

### تنظیم پارامترها

انتخاب مناسب پارامترها نقش بسیار مهمی در کارایی روش‌های فرا ابتکاری دارد. در این پژوهش، برای تعیین سطح مناسب هر پارامتر، از روش تاگوچی استفاده شده است (تاگوچی<sup>۴۰</sup> و همکاران، ۲۰۰۵). این روش بر پایه حداکثرکردن معیار عملکرد (که نرخ سیگنال به نویز نامیده می‌شود)، برای تعیین سطوح بهینه فاکتورهای مؤثر در محاسبات است.

به‌طور کلی روش تاگوچی به‌صورت زیر است:

- برای هر آزمایش نرخ سیگنال به نویز و جدول آنالیز واریانس مشخصه‌های کیفیت آن محاسبه می‌شود.
  - برای فاکتورهایی که اثر مهمی بر نرخ سیگنال به نویز دارند، سطحی که نرخ سیگنال به نویز بالاتری دارد به‌منزله سطح بهینه فاکتور انتخاب می‌شود.
  - فاکتوری که اثر مهمی بر نرخ سیگنال به نویز ندارد، اما بر میانگین پاسخ اثر مهمی دارد، سطح با مقدار هدف بهتر انتخاب می‌شود.
  - برای فاکتورهایی که اثر مهمی بر دو نرخ سیگنال به نویز و میانگین پاسخ ندارند، سطح با مقیاس اقتصادی به‌منزله مقدار بهتر برای الگوریتم در نظر گرفته می‌شود.
- براساس این روش زمانی که مسأله از نوع بهینه‌سازی و پاسخ تا حد ممکن کوچک مدنظر است از رابطه ۵ استفاده خواهد شد. در این رابطه منظور از S سیگنال و N مقادیر نویز در نظر گرفته شده است.

$$SN = -10 \log \left( 1 / n \sum_{i=1}^n y_i^2 \right) \quad (5)$$

آزمایشات برای تنظیم پارامترهای الگوریتم رقابت استعماری و ژنتیک در سه سطح انجام شده است. جدول‌های ۱ و ۲ مقدار پارامترهای الگوریتم رقابت استعماری و ژنتیک را نشان می‌دهند.

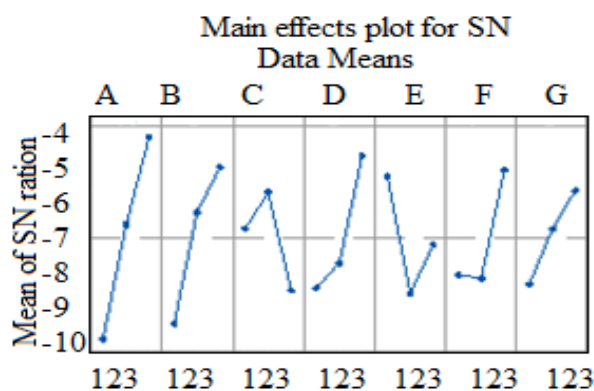
جدول ۱- پارامترهای الگوریتم رقابت استعماری و سطوح آن پارامتر

| سطح  |      |     | پارامتر             |
|------|------|-----|---------------------|
| ۳    | ۲    | ۱   |                     |
| ۱۰۰۰ | ۷۵۰  | ۵۰۰ | تعداد کشورها (A)    |
| ۱۰۰  | ۷۵   | ۵۰  | تعداد امپراطوری (B) |
| ۱۰۰  | ۷۵   | ۵۰  | تعداد دهه‌ها (C)    |
| ۲    | ۱    | ۰/۵ | ضریب جذب (D)        |
| ۰/۷  | ۰/۵  | ۰/۳ | ضریب انحراف (E)     |
| ۰/۴  | ۰/۳  | ۰/۲ | نرخ انقلاب (F)      |
| ۰/۲  | ۰/۱۵ | ۰/۱ | نرخ مشارکت (G)      |

جدول ۲- پارامترهای الگوریتم ژنتیک و سطوح آن پارامتر

| سطح  |     |     | پارامتر      |
|------|-----|-----|--------------|
| ۳    | ۲   | ۱   |              |
| ۱۰۰۰ | ۷۵۰ | ۵۰۰ | جمعیت اولیه  |
| ۱۰۰  | ۷۵  | ۵۰  | تعداد نسل‌ها |
| ۰/۸  | ۰/۷ | ۰/۶ | نرخ ترکیب    |
| ۰/۴  | ۰/۳ | ۰/۲ | نرخ جهش      |

مطابق تعداد پارامترها و سطوح مختلف آنها، طرح تاگوچی L27 برای تنظیم پارامترهای الگوریتم رقابت استعماری و طرح تاگوچی L9 برای تنظیم پارامترهای الگوریتم ژنتیک استفاده شده است. مقادیر نرخ S/N برای الگوریتم رقابت استعماری در شکل ۱۰ نشان داده شده است.



شکل ۱۰- میانگین نرخ S/N در هر سطح برای تنظیم پارامترهای الگوریتم رقابت استعماری

نتایج به دست آمده به روش تاگوچی نشان می‌دهد، سطوح ۳، ۲، ۳، ۱، ۳ و ۳ به ترتیب بهترین سطوح برای پارامترهای تعداد کشورها، تعداد امپراطوری، تعداد دهه‌ها، ضریب جذب، ضریب انحراف، نرخ انقلاب و نرخ مشارکت الگوریتم رقابت استعماری و سطوح ۳، ۲ و ۲ به ترتیب بهترین سطوح برای پارامترهای جمعیت اولیه، تعداد نسل‌ها، نرخ ترکیب و نرخ جهش الگوریتم ژنتیک هستند.

## نتایج پژوهش

در این پژوهش ابتدا عملکرد سه الگوریتم جایگذاری (پایین - چپ، الگوریتم پایین - چپ - جاذبه‌دار، الگوریتم پایین - چپ - جهشی) با یکدیگر مقایسه، سپس الگوریتم جایگذاری مناسب انتخاب و با الگوریتم رقابت استعماری و ژنتیک ترکیب شده‌اند. برای پیاده‌سازی الگوریتم رقابت استعماری و الگوریتم ژنتیک از برنامه‌نویسی تحت زبان برنامه‌نویسی متلب نسخه ۲۰۱۴a استفاده شد. برای ارزیابی عملکرد روش پیشنهادی از ۲۲ مجموعه داده معیار استفاده شده است که شامل ۱۷ تا ۹۷ قطعه و قابل چیدمان به صورت ۱۰۰ درصد هستند (هوپر، ۲۰۰۰). جدول ۳ مشخصات ۲۲ مجموعه داده‌های معیار را نشان می‌دهد.

جدول ۳- مشخصات مجموعه داده‌ها

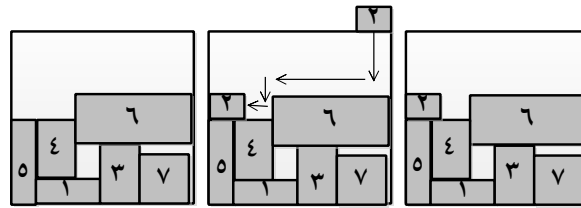
| نام مجموعه داده معیار   | تعداد | سایز بهینه شیء | مرجع                  |
|-------------------------|-------|----------------|-----------------------|
| J1                      | ۲۵    | ۴۰*۲۰          | (جیکوب، ۱۹۹۶)         |
| J2                      | ۵۰    | ۴۰*۲۰          | (جیکوب، ۱۹۹۶)         |
| T1a, T1b, T1c, T1d, T1e | ۱۷    | ۲۰۰*۲۰۰        | (هوپر و تورتون، ۲۰۰۱) |
| T2a, T2b, T2c, T2d, T2e | ۲۵    | ۲۰۰*۲۰۰        | (هوپر و تورتون، ۲۰۰۱) |
| T3a, T3b, T3c,          | ۳۹    | ۲۰۰*۲۰۰        | (هوپر و تورتون، ۲۰۰۱) |
| T4a, T4b, T4c,          | ۴۹    | ۲۰۰*۲۰۰        | (هوپر و تورتون، ۲۰۰۱) |
| T5a, T5b,               | ۷۲    | ۲۰۰*۲۰۰        | (هوپر و تورتون، ۲۰۰۱) |
| T6a, T6b,               | ۹۷    | ۲۰۰*۲۰۰        | (هوپر و تورتون، ۲۰۰۱) |

برای مقایسه عملکرد الگوریتم‌های جایگذاری نام برده شده در بخش (۴) از لحاظ سرعت و قابلیت در تولید چیدمان بهینه، از مجموعه داده T1a استفاده شد. به این منظور تمام جایگشت‌های ممکن مجموعه داده T1a که شامل ۱۷ حالت است با استفاده از پنج الگوریتم جایگذاری نام برده شده به چیدمان معتبر تبدیل شدند و مدت زمان اجرا و بازدهی چیدمان تولیدی هریک از الگوریتم‌های ابتکاری با یکدیگر مقایسه شدند. سرعت و بازدهی چیدمان‌های حاصل در جدول ۴ گزارش شده است.

جدول ۴- نتایج اجرای الگوریتم‌های ابتکاری جایگذاری در چیدمان مجموعه داده T6a،

| الگوریتم ابتکاری جایگذاری | میانگین زمان هر جایگذاری (ثانیه) | میانگین بازدهی چیدمان | بهترین بازدهی چیدمان | انحراف معیار |
|---------------------------|----------------------------------|-----------------------|----------------------|--------------|
| BL                        | ۰/۹۱۸                            | ۴۸/۷۲                 | ۹۳/۷۵                | ۹/۵۲         |
| BLLT                      | ۱/۰۱۷                            | ۶۰/۷۷                 | ۱۰۰                  | ۶/۴۷         |
| BLF                       | ۲/۱۱۹                            | ۶۵/۰۸                 | ۱۰۰                  | ۱۰/۱۲        |
| BLD                       | ۱/۰۴۵                            | ۵۳/۸۵                 | ۹۳/۷۵                | ۶/۳۹         |
| BF                        | ۲/۸۳۰                            | ۴۷/۸۷                 | ۸۴/۴۵                | ۱۱/۵۰        |

نتایج نشان می‌دهد الگوریتم پایین-چپ-جهشی قادر است چیدمان‌هایی با بازدهی بالایی تولید کند؛ اما سرعت اجرای این الگوریتم پایین است. همچنین نتایج نشان می‌دهد علی‌رغم سرعت بالای الگوریتم پایین-چپ و پایین-چپ توسعه یافته، موفق به تولید چیدمان با بازدهی کامل برای این مجموعه داده نشده‌اند. همچنین میانگین بازدهی چیدمان‌های حاصل از آنها نسبت به الگوریتم پایین-چپ-جهشی و الگوریتم پایین-چپ-جاذبه‌دار، پایین است. نتایج حاصل از اجرای الگوریتم بهترین تناسب از هر دو لحاظ سرعت اجرا و بازدهی چیدمان حاصل قابل قبول نیست. این موضوع نشان می‌دهد این الگوریتم برای استفاده در این نوع چیدمان مناسب نیست. از مقایسه نتایج نمایش داده شده در جدول ۴ نتیجه می‌شود الگوریتم جایگذاری پایین-چپ-جاذبه‌دار قادر است با سرعت مناسب جواب‌های قابل قبولی تولید کند. در پژوهش دیگری که نویسندگان این پژوهش انجام داده‌اند نیز مناسب بودن الگوریتم جایگذاری پایین-چپ-جاذبه‌دار به اثبات رسیده است (کارگر و پیوندی، ۱۳۹۳)؛ بنابراین در این پژوهش برای تبدیل دنباله قطعات به چیدمان معتبر از الگوریتم پایین-چپ-جاذبه‌دار استفاده می‌شود. شکل ۱۱ چیدمان دنباله‌ای از قطعات مستطیلی با جایگشت (۲، ۶، ۴، ۷، ۳، ۰، ۱، ۵)، توسط الگوریتم جایگذاری پایین-چپ-جاذبه‌دار را نشان می‌دهد.



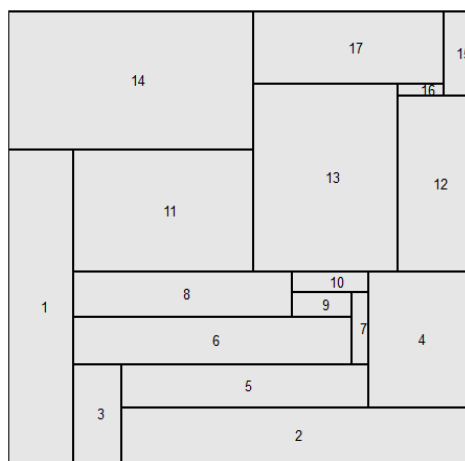
شکل ۱۱- جایگذاری دنباله‌ای از قطعات مستطیلی با جایگشت (۲، ۶، ۴، ۷، ۳، ۰، ۱، ۵) توسط الگوریتم جایگذاری پایین-چپ-جاذبه‌دار،

همان‌طور که بیان شد در این پژوهش برای ارزیابی عملکرد الگوریتم رقابت استعماری در حل مسائل چیدمان، از مقایسه آن با الگوریتم ژنتیک استفاده شده است. الگوریتم ژنتیک، الگوریتم معمول در حل این‌گونه مسائل است. چیدمان هریک از ۲۲ مجموعه داده معیار ۲۰ مرتبه با هریک از دو الگوریتم فرا ابتکاری آزمون شده‌اند. بهترین نتایج به دست آمده با هریک از الگوریتم‌های فرا ابتکاری برای ۲۲ مجموعه داده‌ها در جدول ۵ نمایش داده شده است. حداکثر زمان اجرا برای ۲۲ مجموعه داده معیار با هریک از الگوریتم‌های فرا ابتکاری، ۱۵ دقیقه بوده است. از مقایسه نتایج جدول ۴ مشاهده می‌شود الگوریتم رقابت استعماری به چیدمان با بازدهی ۱۰۰٪، ۴ مجموعه داده دست یافته است و در نهایت ۱۲ چیدمان با بازدهی بالاتر نسبت به الگوریتم ژنتیک تولید کرده است. همچنین مشاهده می‌شود میانگین بازدهی چیدمان‌های تولید شده با الگوریتم رقابت استعماری ۲/۱۲ درصد بالاتر از میانگین بازدهی چیدمان‌های تولید شده با الگوریتم ژنتیک است. شکل ۱۲، دو نمونه چیدمان‌های بهینه یافت شده با الگوریتم رقابت استعماری را نشان می‌دهد.

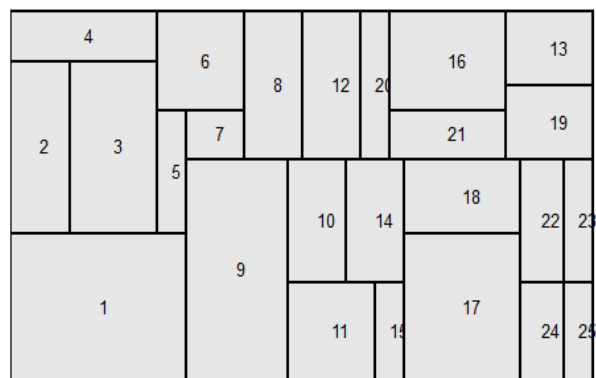


جدول ۵- بهترین نتایج به دست آمده با هریک از الگوریتم‌های فرا ابتکاری برای ۲۲ مجموعه داده‌ها معیار

| الگوریتم فرا ابتکاری |                         | نام مجموعه داده معیار |
|----------------------|-------------------------|-----------------------|
| الگوریتم ژنتیک       | الگوریتم رقابت استعماری |                       |
| ۹۳/۷۵                | ۱۰۰                     | J1                    |
| ۹۳/۷۵                | ۹۳/۷۵                   | J2                    |
| ۹۴/۷۸                | ۹۵/۰۹                   | T1a                   |
| ۹۴/۴۵                | ۹۷/۸۳                   | T1b                   |
| ۹۵/۳۲                | ۱۰۰                     | T1c                   |
| ۹۵                   | ۹۵/۶۷                   | T1d                   |
| ۹۵/۶۰                | ۱۰۰                     | T1e                   |
| ۹۵/۶۹                | ۹۵/۱۳                   | T2a                   |
| ۹۳/۴۵                | ۹۵/۴۳                   | T2b                   |
| ۹۳/۶۳                | ۹۵/۹۰                   | T2c                   |
| ۹۳/۴۹                | ۱۰۰                     | T2d                   |
| ۹۳/۳۲                | ۹۴/۳۲                   | T2e                   |
| ۹۳/۱۶                | ۹۱/۰۵                   | T3a                   |
| ۹۳/۸۹                | ۹۵/۸۹                   | T3b                   |
| ۹۲/۶۳                | ۹۵/۷۸                   | T3c                   |
| ۹۱/۳۲                | ۹۱/۵۹                   | T4a                   |
| ۹۱/۵۹                | ۹۲/۹۶                   | T4b                   |
| ۹۰/۴۵                | ۹۱/۸۸                   | T4c                   |
| ۸۹/۲۸                | ۹۲/۷۸                   | T5a                   |
| ۹۲/۰۳                | ۹۱/۹۸                   | T5b                   |
| ۹۰/۱۵                | ۹۲/۲۱                   | T6a                   |
| ۹۱/۸۸                | ۹۱/۰۲                   | T6b                   |
| ۹۳/۱۱                | ۹۵/۲۳                   | میانگین               |



T1e (length = 200, util. = 100%)



J1 (length = 15, util. = 100%)

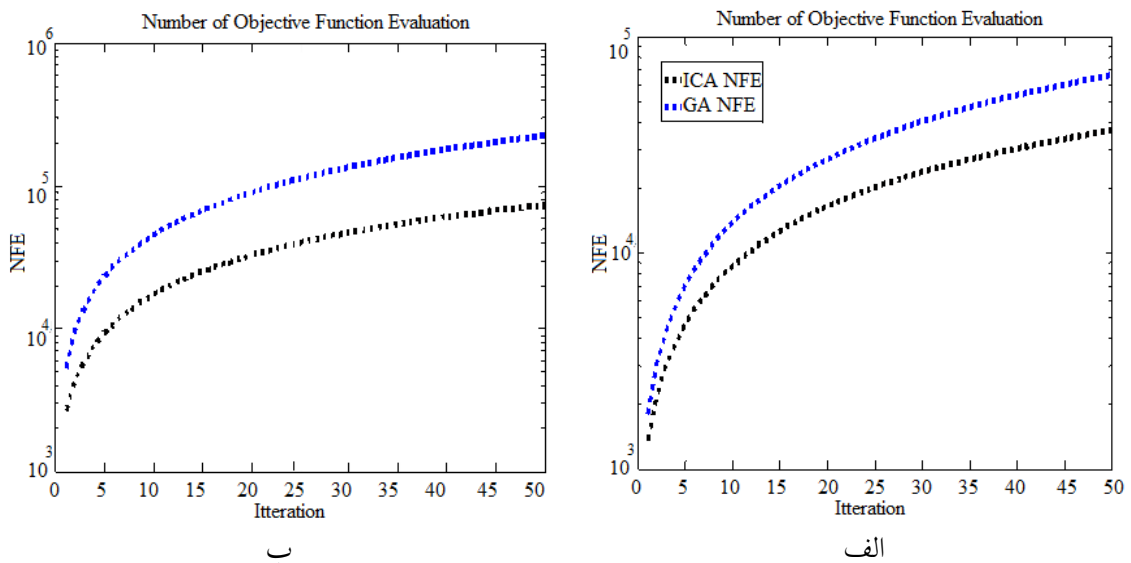
شکل ۱۲- چیدمان بهینه یافت شده با الگوریتم رقابت استعماری

برای مقایسه دقیق‌تر عملکرد الگوریتم رقابت استعماری و الگوریتم ژنتیک در حل این نوع مسائل چیدمان، از آزمون آماری ANOVA برای بررسی معنادار بودن تفاوت نتایج حاصل از این دو الگوریتم استفاده شده است. جدول ۶ خلاصه این نتایج را نشان می‌دهد.

جدول ۶- خلاصه نتایج آزمون ANOVA

| منبع تغییرات | مجموع مربعات | درجه آزادی | میانگین مربعات | نسبت F | سطح معنی‌دار |
|--------------|--------------|------------|----------------|--------|--------------|
| بین‌گروهی    | ۵۵/۱۰۱       | ۲          | ۲۷/۵۵۱         | ۵/۲۴۳  | ۰/۰۰۸        |
| درون‌گروهی   | ۳۳۱/۰۴۶      | ۶۳         | ۵/۲۵۵          | -      | -            |
| کل           | ۳۸۶/۱۴۷      | ۶۵         | -              | -      | -            |

همان‌طور که جدول ۵ نشان می‌دهد تفاوت نتایج حاصل از دو الگوریتم فرا ابتکاری معنادار است. در روش حل مسائل چیدمان علاوه بر قابلیت روش در دستیابی به جواب بهینه، مدت زمان حل مسئله نیز پارامتر با اهمیت در ارزیابی روش است. به دلیل آنکه نوع نرم‌افزار و نحوه کدنویسی برنامه تأثیر زیادی در میانگین زمان اجرای الگوریتم دارد، پارامتر میانگین زمان اجرای الگوریتم، پارامتر مناسبی برای مقایسه نیست؛ از این‌رو برای بررسی دقیق‌تر هر الگوریتم پارامتر تعداد فراخوانی تابع هدف نیز در نظر گرفته شده است. در واقع زمان‌برترین قسمت هر الگوریتم بخش ارزیابی مقدار تابع هدف است و از مقدار آن زمان لازم الگوریتم، مستقل از مشخصات رایانه تقریب زده می‌شود. در عین حال با هر بار فراخوانی تابع هدف، جواب یافته‌شده با الگوریتم ارزیابی می‌شود؛ بنابراین الگوریتمی قوی‌تر است که با تعداد فراخوانی کمتر تابع هدف، نتایج بهتری ارائه کند؛ برای مثال شکل ۱۳ تغییرات تعداد فراخوانی تابع هدف (NFE) در طول نسل برای بهترین جواب الگوریتم‌های ژنتیک و الگوریتم رقابت استعماری را برای مجموعه قطعات J1 و J2 نشان می‌دهد.



شکل ۱۳- تغییرات تعداد فراخوانی تابع هدف در طول نسل برای بهترین جواب با الگوریتم‌های ژنتیک و رقابت استعماری برای الف) مجموعه قطعات J1، ب) مجموعه قطعات J2

همان‌طور که از نمودارهای شکل ۱۳ مشخص است، تعداد فراخوانی تابع هدف با الگوریتم ژنتیک در چیدمان هر دو مجموعه قطعات بیشتر از الگوریتم رقابت استعماری است. این موضوع به‌منزله زمان بیشتر اجرای الگوریتم ژنتیک است. همچنین الگوریتم ژنتیک با وجود زمان اجرای بیشتر چیدمان‌هایی با میانگین بازدهی کمتر تولید کرده است؛ بنابراین الگوریتم رقابت استعماری جایگزین خوبی برای الگوریتم ژنتیک در حل مسائل چیدمان قطعات مستطیل است.

### نتیجه‌گیری

در این مقاله برای حل مسائل برش قطعات منظم مستطیل شکل، روش فرا ابتکاری براساس الگوریتم نوظهور رقابت استعماری ارائه شد. کارایی روش پیشنهادی با استفاده از ۲۲ مجموعه داده معیار ارزیابی و نتایج حاصل با نتایج الگوریتم ژنتیک مقایسه شد.

عملکرد دو روش فرا ابتکاری براساس دو فاکتور قابلیت روش در دستیابی به چیدمانی با بازدهی بالاتر و سرعت روش ارزیابی شده است.

براساس آنالیز آماری نتایج حاصل از اجرای روش‌های فرا ابتکاری مشاهده شد نتایج حاصل از ICA تفاوت معناداری با نتایج حاصل از GA دارد و الگوریتم ICA قادر به دستیابی به چیدمان‌هایی با بازدهی بالاتر است. همچنین الگوریتم ICA به چیدمان صد در صد ۴ نمونه از ۲۲ مجموعه داده دست یافته است. الگوریتم رقابت استعماری قادر است با تعداد NFE بسیار کمتر به جواب مشابه با الگوریتم ژنتیک دست یابد. از آنجا که تعداد NFE کمتر در طول اجرا الگوریتم فرا ابتکاری نشان‌دهنده سرعت بیشتر اجرای الگوریتم است، الگوریتم رقابت استعماری روشی سریع‌تر در حل این‌گونه مسائل جایگشتی است.

در نهایت نتیجه می‌شود الگوریتم ICA روشی کارآمدتر نسبت به الگوریتم ژنتیک در حل مسائل چیدمان است. در حل مسائل چیدمان با استفاده از روش‌های فرا ابتکاری، تعداد بهینه‌های محلی زیاد باعث سردرگمی الگوریتم در پیدا کردن مسیر درست به سمت بهینه مطلق می‌شود. راهکاری که برای حل این مشکل مناسب دیده می‌شود، تأثیر دادن پارامتر دیگری به‌غیر از طول چیدمان در تابع هدف است. به این‌گونه که تابع هدف دو چیدمان با طولی یکسان از مجموعه قطعات یکسان، از یکدیگر متمایز هستند. مناسب دیده می‌شود در پژوهش‌های آینده این موضوع بررسی شود.

### منابع

- Kargar, M., Payvandy, P. (2015) "Application of heuristic methods in marker making", 9<sup>th</sup> NTC, Iran, Tehran [In Persian]
- Kargar, M., Payvandy, P. (2015) "An Overview for Marker Making Methods Using Heuristic and Metaheuristic Algorithms", *Textile Science and Technology*, 11(2), 17-28, [In Persian]
- Alvarez, V. R., Parreno, F., & Tamarit, J. M. (2007). "A Tabo search algorithm for a two-dimensional non-guillotine cutting problem". *European Journal of Operational Research*, 183, 1167-1182.

- Amaro, B., Pinheiro, P. R., & Saraiva, R. D. (2013). "A hybrid methodology for tackling the irregular strip packing problem". in Proceedings of the 11th IFAC Workshop on Intelligent Manufacturing Systems (IMS '13), 11, 396–401.
- Atashpaz-Gargari, E., & Lucas, C. (2007). "Imperialist Competitive Algorithm: An algorithm for optimization inspired by imperialistic competition". IEEE Congress on Evolutionary Computation, 4661-4667.
- Beisiegel, B., Kallrath, J., Kochetov, Y., & Rudnev, A. (2005). "Simulated annealing based algorithm for the 2D bin packing problem with impurities". International CoNFERENCE of the German Operations Research Society (GOR), Bremen.
- Bortfeldt, A. (2006). "A genetic algorithm for the two-dimensional strip packing problem with rectangular pieces". *European Journal of Operational Research*, 172, pp. 814-837.
- Burke, E. K., Hellier, R., Kendall, G., & Whitwell, G. (2006). "A new Bottom- Left-Fill heuristic algorithm for the 2d irregular packing problem". *European Journal of Operational Research*, 54, 587-601.
- Burke, E. k., Kendall, G., & Whitwell, G. (2004). "A new placement heuristic for the orthogonal stock-cutting problem". *European Journal of Operational Research*, 52(4), 655–671.
- Chazelle, B. (1983). "The Bottom-Left Bin-Packing Heuristic: An Efficient Implementation". *J. IEEE. T. Comput*, 32, 697-707.
- Dagli, C.H., & Poshyanonda, P. (1997). "New approaches to nesting rectangular patterns". *Journal of Intelligent Manufacturing*. 8(3), 177-190.
- Dowsland, K. (1993). "Some experiments with simulated annealing techniques for packing problems". *European Journal of Operational Research*, 68, 389-391.
- Dyckhoff, H. (1990). "Typology of cutting and packing problems". *European Journal of Operational Research*, 44, 145-159.
- Ebrahimi, S., & Payvandy, P. (2013). "Optimization of the Link Drive Mechanism in a Sewing Machine Using Imperialist Competitive Algorithm". *International Journal of Clothing Science and Technology*, 26(3), 247 – 260.
- Faina, L. (1999). "An application of simulated annealing to the cutting stock problem". *European Journal of Operational Research*, 114, 542-556.
- Falkenauer, E. & Delchambre, A. (1992). "A genetic algorithm for bin-packing and line balancing". Proceedings of the 1992 IEEE International CoNFERENCE on Robotics and Automation, 1186-1192.
- Hifi, M., & Hallah, R. M. (2003). "Hybrid algorithm for the two dimensional layout problem: the cases of regular and irregular shapes". *Journal of International Transactions in Operational Research*, 10, 195-216.
- Hopper E., & Turton B. C.H. (2001). "A Review of the Application of Meta-Heuristic Algorithms to 2D Strip Packing Problems". *Artificial Intelligence Review*, 16, 257 – 300.
- Hopper, E., & Turton, B.C.H. (2001). "An empirical investigation of meta-heuristic and heuristic algorithms for a 2D packing problem". *European Journal of Operational Research*, 128(1), 34-57.
- Hopper. E. (2000). Two-dimensional packing utilizing evolutionary algorithms and other meta-heuristic methods. Ph.D Thesis, University of Wales, Cardi.
- Hwang, S. M., Cheng, Y. K., & Horng, J. T. (1994). "On solving rectangle bin packing problems using genetic algorithms". Proceedings of the 1994 IEEE International CoNFERENCE on Systems, Man and Cybernetics, 1583-1590.
- Jakobs, S., (1996). "On genetic algorithms for the packing of polygons". *European Journal of Operational Research*. 88, 149- 165.
- Junior, B. A. , Pinheiro, P. R. , & Saraiva, R. D. (2013). "Tackling the irregular strip packing problem by hybridizing genetic algorithm and bottom-left heuristic". in Proceedings of the IEEE Congress on Evolutionary Computation (CEC '13), 3012–3018, Cancun, Mexico.

- Lai, K. K., & Chan, J. W. M. (1997). "Developing a simulated annealing algorithm for the cutting stock problem". *Computers & Industrial Engineering*, 32, 115-127.
- Leung, S. C. H., Lin, Y., & Zhang, D. (2012). "Extended local search algorithm based on nonlinear programming for twodimensional irregular strip packing problem". *Computers and Operations Research*, (39), 678–686.
- Li, M., Huang, P.J., & Zhou, Z. (2009). "Optimal Layout of Rectangular Parts Based on Niche Genetic Algorithm". *Journal of Hunan University*, 1, 322-330.
- Lins, L., Lins, S., & Morabito, R. (2003). "An L-Approach For Packing (l, w) Rectangles Into Rectangular And L-Shaped Pieces". *Journal of Operational Research Society*, 54(7), 777-789.
- Liu, D., & Teng, H. (1999). "An improved BL-algorithm for genetic algorithms of the orthogonal packing of rectangles". *European Journal of Operational Research*, 112(2), 413-419.
- Lodi, A., Martello, S. & Vigo, D. (2004). "TSPack: A Unified Tabu Search Code for Multi-Dimensional Bin Packing Problems". *Annals of Operations Research*, 131(1), 203-203.
- Mccell, J. (2005). "Genetic Algorithm for Modeling and Optimization". *Journal of Computational and Optimization*, 184, 205-222.
- Oliveira, J. F., & Ferreira, J. S. (1993). "Algorithms for nesting problems". *Journal of Intelligent Manufacturing*, 39, 255-276.
- Ortmann, F. (2010). Heuristics for online rectangular packing problems, PhD dissertation, University of Stellenbosch.
- Ozcan, E., Kai, Z., & Drake, J. H. (2013). "Bidirectional best-fit heuristic considering compound placement for two dimensional orthogonal rectangular strip packing". *Expert Systems with Applications*, 40, 4035–4043.
- Pureza, V., Morabito, R. (2006). "Some experiments with a simple Tabu search algorithm for the manufacturer's pallet loading problem". *Journal of Computers & Operations Research*, 33, 804-819.
- Runarsson, T. P., Jonsson, M. T., & Jenson P. (1996) "Dynamic dual bin packing using fuzzy objectives". 8th IEEE International CoNference on Collaborative. Nagoya, 219-222.
- Shin, Y. B., & Kita, E. (2012). "Solving two-dimensional packing problem using particle swarm optimization". *Computer Assisted Methods in Engineering and Science*, 19, 241–255.
- Soke, A., & Bingul, Z. (2006). "Hybrid genetic algorithm and simulated annealing for two-dimensional non-guillotine rectangular packing problems". *Engineering Applications of Artificial Intelligence*, 19, 557–567.
- Taguchi, G., Chowdhury, S., & Wu, Y. (2005). *Taguchi's Quality Engineering Handbook*. Wiley, New Jersey.
- Valenzuela, C. L., & Wang, P. Y. (2001). "Heuristics for large strip packing problems with guillotine patterns: An empirical study". *Metaheuristic International CoNference 2001*, Porto.
- Wang, B. (2010). "An Adaptive Genetic Algorithm for 2D Packing Problem". *Modern Applied Science*, 4(4), 1- 4.
- Wei, L., Oon, W-C., Zhu, W., & Lim, A. (2011). "A skyline heuristic for the 2d rectangular packing and strip packing problems". *European Journal of Operational Research*, 215, 337–346.

<sup>1</sup> Hopper and Turton

<sup>2</sup> Dyckhoff

<sup>3</sup> Burke

<sup>4</sup> Liu and Teng

<sup>5</sup> Chazelle

<sup>6</sup> Oliveira and Ferreira

<sup>7</sup> Leung

- 8 Ozcan
- 9 Falkenauer and Delchambre
- 10 Hwang
- 11 Runarsson
- 12 Dagli and Poshyanonda
- 13 Valenzuela and Wang
- 14 Bortfeldt
- 15 Hifi and Hallah
- 16 Dowsland
- 17 Lai and Chan
- 18 Faina
- 19 Beisiegel
- 20 Lodi
- 21 Wei
- 22 Alvarez and Parreno
- 23 Pureza and Morabito
- 24 Shin and Kita
- 25 Wang
- 26 Amaro and Pinheiro
- 27 Junior
- 28 Jakobs
- 29 Poshyananda
- 30 Soke and Bingul
- 31 Li
- 32 Imperialist Competitive Algorithm – ICA
- 33 Ortmann
- 34 Lins
- 35 Genetic Algorithm - GA
- 36 Micell
- 37 Colony
- 38 Imperialist
- 39 Lucas
- 40 Taguchi