

یک روش ابتکاری برای حل مسأله برش دوبعدی

مسعود ربانی

استادیار گروه مهندسی صنایع - دانشکده فنی - دانشگاه تهران

فریبرز جولای

استادیار گروه مهندسی صنایع - دانشکده فنی - دانشگاه تهران

مهدی خطیبی

کارشناس ارشد مدیریت صنعتی - تحقیق در عملیات

(تاریخ دریافت ۸۱/۸/۲۹، تاریخ تصویب ۸۲/۷/۱۲)

چکیده

یکی از مسائل شناخته شده در حوزه تحقیق در عملیات بویژه در زمینه برنامه ریزی تولید مسأله برش است. مسأله برش دوبعدی انطباق بیشتری با مسائل واقعی در حوزه های صنعتی دارد. از اهداف اصلی در بررسی مسأله برش دو بعدی کاهش ضایعات ناشی از برش است، این مسأله زمانی که ماده اولیه برش در حجم وسیعی برش یابد اهمیت بیشتری پیدا می کند. در این مقاله با هدف کاهش ضایعات، فضای جواب برای الگوی برش دو بعدی با یک گراف and-or توسعه داده می شود و سپس با کمک یک الگوریتم بهینه سازی احتمالی مبتنی بر روشهای بهینه سازی محلی یک روش شاخه-کران بهبود یافته برای کاهش فضای جستجوی مسأله ارائه می شود. نتایج ارائه شده مؤید قابلیت مناسب الگوریتم در مقایسه با الگوریتم های رایج می باشد.

واژه های کلیدی: برش دو بعدی، جستجوی گراف، گراف and-or، بهینه سازی احتمالی، بهینه سازی محلی.

مقدمه

با توجه به تعریف فوق می توان در حالت کلی مسأله برش دو بعدی را به صورت مدل زیر بیان کرد:

$$\text{Max } f = \sum_{i=1}^m v_i \cdot a_i$$

s.t. (P)

$$\sum_{i=1}^m (l_i \cdot w_i) \cdot a_i \leq L \cdot W$$

$$\forall i ; a_i \text{ is integer}$$

که در آن a_i برابر با تعداد قطعات تولید شده از نوع i است.

هر برش گیوتینی موازی با یکی از اضلاع صفحه مستطیل شکل انجام می شود و در اثر بکارگیری آن دو مستطیل جدید تولید می شود. در عین حال محدودیتی در تعداد برشهای گیوتینی بکار گرفته شده وجود ندارد. در این صورت مسأله non-staged نامیده می شود.

حل مسأله الگوی برش دوبعدی نامحدود و non-staged از روش برنامه ریزی پویا برای

مسأله برش دو بعدی نامحدود گیوتینی شامل برش یک صفحه اولیه مستطیل شکل به ابعاد (L, W) به m قطعه مستطیل شکل کوچکتر به ابعاد (l_i, w_i) ، $i=1, 2, \dots, m$ است. مفروضات عبارتند از:

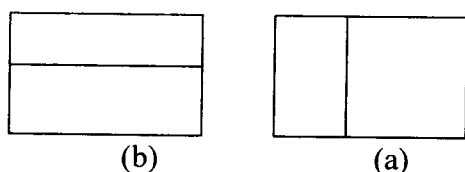
(الف) چرخش 90° برای هر قطعه مثل S به ابعاد (l_i, w_i) در الگوی برش امکانپذیر نیست به این ترتیب قطعه به ابعاد (w_i, l_i) را به عنوان یک قطعه جدید در الگوی برش مورد بررسی قرار می دهیم. پس هر قطعه مثل S در مجموعه قطعات مورد تقاضا متناظر با یک زوج مرتب مثل (l_i, w_i) است.

(ب) برای تعداد قطعات تولید شده از هر نوع به ابعاد (l_i, w_i) محدودیتی وجود ندارد.

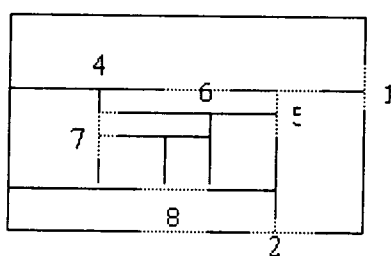
(ج) برشهای تولید شده در هر گام گیوتینی خواهند بود.

(د) برای هر قطعه به ابعاد (l_i, w_i) ارزش مطلوبیتی قائل می شویم که برابر است با $v_i = l_i \cdot w_i$ در اینصورت مسأله دقیقاً با می نیم کردن ضایعات حاصل از برش برابر است.

یک الگوی برش گیوتینی درست می کنند اگر این قطعات را بتوان از توالی یک سری برشهای گیوتینی ممکن بر روی صفحه (L, W) تولید کرد.



شکل ۱: (a) نمایش یک برش عمودی گیوتینی، (b) نمایش یک برش افقی گیوتینی.



شکل ۲: نمایش یک گیوتینی در ۸ گام.

گراف and-or

برای دستیابی به قطعات مستطیل شکل مورد تقاضای (l_i, w_i) ; $i=1,2,\dots,m$ ابتدا صفحه اولیه به مستطیل های جدیدتری بریده می شود. صفحات مستطیل شکل جدید نیز برای دستیابی به قطعات مورد تقاضا به صورت موفق برش داده می شوند (این برشها محدود به برشهای گیوتینی هستند). این کار مادامی که تمام قطعات تولید شده قطعه نهایی مورد تقاضا یا یک قطعه ضایعاتی باشند ادامه می یابد.

در مسأله برش دوبعدی صفحه مستطیل شکل اولیه به وسیله گره اولیه نمایش داده می شود و قطعات کوچک مورد تقاضا بوسیله گره های نهایی نمایش داده می شوند. قواعد جایجایی بر مبنای برشهای ممکن (برشهای قابل قبول) بیان می شوند. به این ترتیب یک زیر مجموعه از برشها را می توان برای برش صفحه اولیه (صفحه A در شکل ۳) بکار گرفت. چهار مستطیل منشعب شده (B, C, D و E در شکل ۳) نقش گره های میانی یا واسطه را در این گراف خواهند داشت. زیر مجموعه های دیگری از مجموعه برشهای ممکن را می توان برای این گره های میانی بکار گرفت و به همین ترتیب الی آخر. این

استفاده کرد و برای اولین بار با بهبود بخشیدن به روش Herz (1972) مسائل بزرگ را حل کرد. Herz برای اولین بار فضای پاسخ الگوی برش دوبعدی را بدون آنکه به بهیگی مسأله خللی وارد شود در غالب یک گراف بیان کرد. همچنین Morabito, Arenalez (1992) از یک روش تجربی شاخه-کران برای کاهش فضای پاسخ مسأله در جهت دستیابی به الگوی بهینه برش استفاده کردند. Alvarez, Parajon, M. Tamarit (2002) روشهای ابتکاری جدیدی برای حل مسأله برش دو بعدی ارائه کردند. این روشها به حل مسأله برش دوبعدی محدود، نامحدود، موزون و ناموزون می پردازند. به همین دلیل از نظر حل مسأله برش دوبعدی الگوریتمهایی عمومی محسوب می شوند و بر پایه الگوریتمهای مذکور، الگوریتمی مبتنی بر روش tabu search ارائه کرده اند که مسائل برش دو بعدی را با سرعت و دقت بالایی حل می کند.

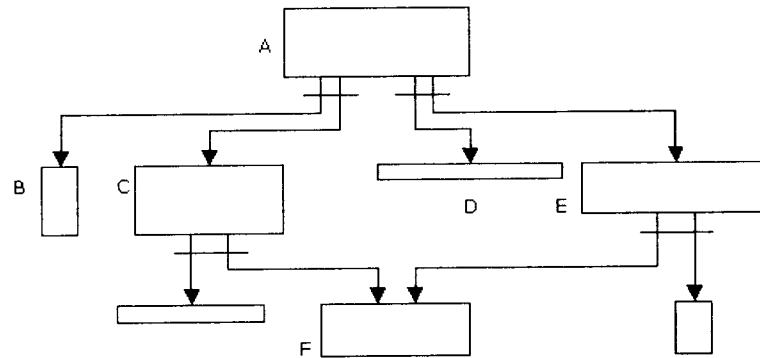
Hifi (1997) یک الگوریتم ترکیبی برای حل مسأله برش دو بعدی گیوتینی موزون و ناموزون ارائه کرد. این الگوریتم ترکیبی از روش جستجوی درختی و برنامه ریزی پویاست. در این الگوریتم با استفاده از برنامه ریزی پویا در هر گره از درخت جستجو، حدود پایین و بالای جدیدی محاسبه می شود. الگوریتم فوق کارآیی زیادی در حل مسائل بزرگ در زمان کوتاه دارد.

توسعه فضای پاسخ مسأله

برای بیان فضای پاسخ مسأله از گراف and-or استفاده خواهیم کرد که تمامی حالات ممکن برای برش صفحه های اولیه و میانی را در بر می گیرد. قبل از آنکه به توسعه گراف بپردازیم چند تعریف که برای بیان مسأله مورد نیاز خواهند بود ارائه می شوند.

تعریف - برش گیوتینی: یک برش را گیوتینی گویند در صورتی که نتیجه بکارگیری آن بر روی یک صفحه مستطیل شکل دو مستطیل جدید باشد. (نمایش یک برش گیوتینی در شکل (۱) ارائه شده است).

تعریف - الگوی برش گیوتینی: مستطیل اولیه A_0 با ابعاد (L, W) را در نظر بگیرید در اینصورت مجموعه ای از قطعات مستطیل شکل کوچکتر بر روی این مستطیل



شکل ۳: نمایش یک گراف and-or در سه سطح.

بگیرید. از گره اولیه یک و فقط یک یال را در نظر بگیرید (که حتما باید یک and-arc یا یک or-arc باشد) و از هر گره منتج از گره اولیه که یال انتخاب شده به آنها اشاره دارد دوباره یک و فقط یک یال را انتخاب کنید، این عمل را تا زمانی انجام دهید که از تمام گره های منتج شده از گرهی که در هر بار در نظر می گیرید گره های دیگری منشعب نشود. این توالی و ترتیب پیمودن گراف and-or یک مسیر کامل نامیده می شود.

مسیر کاملی که در بالا توضیح داده شد توالی یک سری برشهای گیوتینی را که از صفحه اولیه شروع می شود و به یک سری مستطیل ختم می شود نشان می دهد و تولید یک الگوی برش می کند که البته یک پاسخ شدنی برای مسأله P است. به این ترتیب مسیر کاملی که در این میان باید انتخاب شود آن مسیری است که تابع هدف f را ماکزیمایز می کند. و به این ترتیب بهترین الگوی برش را نمایش خواهد داد.

چیدمانهای مختلف از مجموعه قطعات یکسان مورد تقاضا در یک مستطیل الگوهای برش یکسان تولید می کنند. Herz (1972) نشان داد بدون اینکه به کلیت بحث خللی وارد شود تمام برشهای گیوتینی را می توان در دو مجموعه زیر خلاصه کرد:

$$X = \{ x | x = \sum_{i=1}^m \alpha_i \cdot l_i ; l_0 \leq x \leq L - l_0, \alpha_i \geq 0 \text{ and integer} \}$$

برشهای افقی:

$$Y = \{ y | y = \sum_{i=1}^m \beta_i \cdot w_i ; w_0 \leq y \leq W - w_0; \beta_i \geq 0 \text{ and integer} \}$$

عمل تا زمانی بکار گرفته می شود که قطعات مستطیل شکل (گره های میانی تولید شده) تبدیل به گره های نهایی شوند. نمایش فرآیند بیان شده در بالا در شکل (۳) ارائه شده است.

Rich (1983) گراف and-or را به عنوان ساختاری مناسب برای نمایش روش حل مسائلی که می توان آنها را با تجزیه به مسائل کوچکتری تبدیل کرد بیان می کند، که البته همگی آنها باید حل شود. نمایش این تجزیه یا کاهش مسأله به مسائل کوچکتر یالهایی تولید می کند که ما آنها را and-arc می نامیم. یک and-arc می تواند به چندین گره منشعب از گره اولیه ختم شود، که همگی آنها باید حل شوند (در شکل (۳) یک and-arc منشعب شده از گره اولیه A به وسیله یک خط کوچک که دو خط ختم شونده به گرههای B و C را به هم وصل می کند نمایش داده شده است). به همین ترتیب مانند گرافهای معمولی که از یک گره اولیه می توان چندین گره انشعابی داشته باشیم در این گراف نیز می توان از یک گره اولیه چندین and-arc مختلف انشعاب داد (چنانچه در شکل (۳) نمایش داده شده است). به همین دلیل ما گراف حاصل را فقط یک and-graph نمی نامیم و آنرا and-or graph می نامیم.

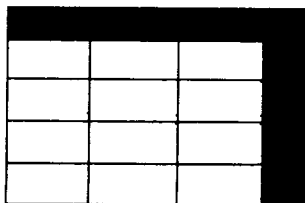
ارائه الگوی برش در قالب یک مسیر کامل در گراف and-or

ترتیب و توالی زیر در یک گراف and-or را در نظر

حد پایین هر گره در روش شاخه-کران

برای محاسبه حد پایین هر گره تنها شرطی که باید رعایت شود شدنی بودن پاسخ ارائه شده به عنوان حد پایین برای آن گره است به همین منظور تعریف زیر را ارائه می دهیم که مبنای محاسبه حد پایین در گراف and-or خواهد بود.

تعریف-الگوهای برش متشابه: یک الگوی برش گیوتینی که ماکزیم تعداد ممکن، از یک نوع قطعه مورد تقاضا از یک صفحه را بدون آنکه این قطعات بخواهند در الگوی برش دوران داشته باشند تولید کند، یک الگوی برش متشابه نامیده می شود.



شکل ۴ نمایش یک الگوی برش متشابه. (قسمت تیره در شکل نمایانگر ضایعات ناشی از برش است).

اگر الگوی برش متشابه بر حسب قطعه i با ابعاد (l_i, w_i) را بر روی صفحه اولیه به ابعاد (L, W) اعمال کنیم $[L/l_i], [W/w_i]$ تعداد قطعات تولید شده از این نوع بر روی صفحه مورد نظر اولیه خواهد بود، که در آن $[X]$ همان جزء صحیح X است.

قطعه مورد تقاضا i به ابعاد (l_i, w_i) را در نظر بگیرد، در اینصورت برش متشابه از نوع قطعه i بر روی این صفحه اولیه ارزش مطلوبیت زیر را تولید می کند.

$$\varphi_i(N) = V_i \cdot [L/l_i] \cdot [W/w_i] \quad (2)$$

با توجه به تعریف بالا و با تأکید مجدد بر این نکته که برشهای متشابه همواره در فضای شدنی مسأله قرار می گیرند حد پایین برای گره N را به صورت زیر تعریف می کنیم:

$$\varphi^0(N) = \max_{i \in M(N)} \{ \varphi_i(N) \}$$

(۳)

که در آن:

که در آن l_0 و w_0 کوتاهترین طول و کوتاهترین عرض در بین ابعاد قطعات مورد تقاضا هستند.

تعریف- ارزش یک گره نهایی: در صورتی که گره نهایی یک قطعه مورد تقاضا با ابعاد (l_i, w_i) را نمایش بدهد ارزش آن برابر است با $V_i = l_i \cdot w_i$ و در غیر اینصورت ارزش آن $V_i = 0$ است. به این ترتیب اگر (L, W) یک گره نهایی در گراف and-or باشد ارزش آن با $F_0(L, W)$ نمایش داده می شود و به صورت زیر تعریف می شود:

$$F_0(L, W) = \begin{cases} l_i \cdot w_i & \text{If } (L, W) = (l_i, w_i) \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

تعریف- ارزش یک مسیر کامل: ارزش یک مسیر کامل برابر است با مجموع ارزش گره های نهایی آن. نتیجه: مسأله یافتن الگوی بهینه برش دقیقاً با مسأله یافتن پرارزشترین مسیر کامل در گراف and-or معادل است. این مسیر زمانی مشخص می شود که گره اولیه حل شده باشد.

روش شاخه-کران

چنانچه گفته شد توسعه گراف and-or برای نمایش فضای جواب مسأله الگوی برش دو بعدی روش کارآمدی می باشد. نکته مهمی که در مورد گراف فوق باید رعایت شود بکارگیری مکانیزمی در استراتژی جستجوی گراف است که از شاخه سازی بیهوده جلوگیری کند. این مسأله زمانی اهمیت خود را نمایان می سازد که مسأله از نظر گستردگی فضای جواب دارای ابعاد بزرگی باشد. به این ترتیب اگر بتوان مکانیزمی ایجاد کرد که از تولید گره های جدیدی که ادامه جستجو در آنها ما را به جواب بهتری نمی رساند جلوگیری کنیم برای رسیدن به جواب فضای کمتری را جستجو خواهیم کرد. روش پیشنهادی مقاله حاضر برای رسیدن به این هدف روش شاخه-کران است. به این ترتیب یک حد پایین و یک حد بالا در هر گره تولید می شود تا تصمیم گیری در جهت تولید گره های جدید از هر گره انجام شود.

الگوریتم (SA) Simulated Annealing

SA یک الگوریتم بهینه سازی احتمالی است (Kirkpatrick S, Gellat C, Vecchi M (1983)). در الگوریتمهای بهینه سازی محلی یک پاسخ اولیه شدنی به عنوان جواب فعلی مسأله انتخاب می شود. پس از آن در همسایگی این نقطه به جستجوی پاسخ های بعدی می پردازند به این ترتیب اگر نقطه ای در همسایگی پاسخ فعلی پیدا شود که جواب بهتری برای مسأله ارائه دهد به عنوان جواب مسأله جایگزین پاسخ اولیه می شود، این فرآیند تا زمانی انجام می شود که هیچ جواب بهتری برای مسأله بدست نیاید.

SA نیز به همین روش عمل می کند اما یک تفاوت اساسی با روش فوق دارد. حرکت از جواب فعلی به جواب دیگری در همسایگی این نقطه فقط در صورتی که جوابهای بهتری بدست آوریم انجام نمی شود، در SA این امکان وجود دارد که با یک احتمال مشخص (که به صورت تصادفی تولید می شود) پاسخ های بدتر نیز پذیرفته شوند. پذیرفته شدن پاسخ های بدتر به این دلیل انجام می شود که امکان فرار از تله می نیمم محلی فراهم شود. الگوریتمهای کاهنده نسبت به قرار گرفتن در تله می نیمم محلی حساسیت لازم را ندارند در نتیجه امکان اینکه اینگونه الگوریتمها در تله می نیمم محلی قرار گیرند وجود دارد. چنانچه بیان شد تفاوت SA در مقایسه با سایر الگوریتمهای کاهنده در پذیرش پاسخهای بدتر است. برای درک ساز و کار موجود در الگوریتم SA برای پذیرش پاسخهای بدتر فرض کنید در نقطه X به عنوان جواب فعلی مسأله قرار داریم به این ترتیب اگر نقطه ای مثل X_{n+1} در همسایگی X یافت شود که به ازای آن جواب تابع هدف بهبود نیابد، مقدار ΔE به صورت زیر محاسبه می شود:

$$\Delta E = F(x_{n+1}) - F(x) \quad (7)$$

که میزان بدتر شدن پاسخ مسأله را نشان می دهد. تابع توزیع احتمال بولتزمن به صورت زیر محاسبه می شود. Spinelis, Papadopoulos, McGregor S(2000):

$$P(x_{n+1}) = e^{(-\Delta E/T)} \quad (8)$$

$$M(N) = \{i \mid l_i \leq L, w_i \leq W, i=1,2, \dots, m\} \quad (4)$$

حد پایینی که ارائه شد توسط Morabito, Arenalez (1992) استفاده شده است و در عین حال چون الگوی برش متشابه یک جواب شدنی برای مسأله است قابل قبول است. در مقاله حاضر نیز با توجه به قابلیت مذکور از همین روش استفاده گردید.

محاسبه حد بالا

Beasley(1985) حد بالا برای هر گره مثل N به ابعاد (L,W) را به این ترتیب بیان کرد:

$$u(N) = \max \sum_{i \in M(N)} v_i \cdot a_i$$

$$s.t. \sum_{i \in M(N)} (l_i \cdot w_i) \cdot a_i \leq (L \cdot W)$$

$$a_i \geq 0 ; i \in M(N) \quad (5)$$

روش فوق برای محاسبه حد بالا کارآیی لازم را دارد. لیکن برای کاهش محاسبات در گراف and-or (کاهش در تعداد گره هایی که شمارش می شوند) حد بالای ارائه شده در بالا به صورت زیر تصحیح می شود:

$$u(N) = \max \sum_{i \in M(N)} v_i \cdot a_i$$

$$s.t. \sum_{i \in M(N)} (l_i \cdot w_i) \cdot a_i \leq (L \cdot W)$$

and

$$a_i \geq 0 ; i \in M(N) \text{ integer} \quad (6)$$

به این ترتیب فرض عدد صحیح بودن برای a_i وارد شروط مسأله می شود.

در نظر گرفتن شرط عدد صحیح بودن برای a_i این امکان را فراهم خواهد کرد تا حد بالای هر گره به صورت واقعی محاسبه شود. در اینجا از روش بهینه سازی احتمالی Simulated Annealing استفاده می کنیم تا میزان پیچیدگی محاسباتی را کاهش دهیم و اثرات بکارگیری الگوریتم پیشنهادی را بر روی گراف and-or مورد بررسی قرار می دهیم.

هر قطعه برابر است با حاصلضرب طول و عرض آن $(v_i = l_i \cdot w_i)$ و پس از تعاریف زیر شرح الگوریتم ارائه می شود:

n : شمارنده دفعات تکرار الگوریتم.

X_n : بردار X در گام n ام الگوریتم.

F_n : مقدار تابع هدف مسأله (۶) در گام n ام الگوریتم.

F^* : بهترین مقدار بدست آمده برای تابع هدف مسأله (۶) تا مرحله فعلی الگوریتم.

X^* : بهترین مقدار برای بردار X تا مرحله فعلی الگوریتم که به ازای آن $F(x) = F^*$

T_0 : مقدار اولیه برای پارامتر دما.

T_n : مقدار پارامتر دما در تکرار n ام الگوریتم.

$P(X_n)$: احتمال بولتزمن به ازای بردار X_n در صورتی که تابع هدف به ازای این بردار بهبود نیابد.

α : ضریب کاهش پارامتر T که عددی است متعلق به بازه $(0, 1)$.

k : عددی ثابت است که پس از هر مرحله کاهش دما و پس از تکرار الگوریتم به تعداد آن فرآیند کاهش دما انجام می شود.

p : احتمال عدم پذیرش پاسخهای بدتر.

۱- یافتن جواب اولیه:

تنها شرط موجود در الگوریتم SA برای جواب اولیه شدنی بودن آنست (Abduelmola, Taboun 2000). به همین منظور برای هر گره مثل N در هر جای گراف and-or جواب اولیه برای حد بالا برابر با حد پایین قرار داده می شود که با توجه به روابط (۲) و (۳) به صورت زیر محاسبه می شود:

$$\varphi_i = V_i \cdot [L/l_i] \cdot [W/w_i]$$

$$\varphi^0 = \max_i \varphi_i ; i=1, 2, \dots, m$$

φ^0 برابر است با ماکزیمم ارزشهای مطلوبیت بدست آمده در نتیجه بکارگیری الگوهای برش یکسان برای تمامی مستطیلهای مورد تقاضا.

به این ترتیب اگر حد پایین گره در اثر بکارگیری الگوی برش متشابه برای قطعه نوع i محاسبه شود، بردار اولیه شدنی X_0 به صورت زیر تولید می شود:

$$x_0 = (0, \dots, 0, a_i, 0, \dots, 0)$$

T در اصطلاح دمای فعلی الگوریتم نامیده می شود (Pirlot(1996). احتمال بدست آمده در بالا با عدد p که احتمال پذیرش پاسخهای بدتر می باشد و در هر بار تکرار الگوریتم به صورت تصادفی تولید می شود مقایسه می شود. اگر $P(X_{n+1}) \geq p$ باشد جواب بدتر پذیرفته می شود و در غیر اینصورت پذیرفته نخواهد شد. نکته دیگر در الگوریتم SA مقدار دمای فعلی T می باشد. پارامتر T در هر از چند گاهی کاهش می یابد تا احتمال پذیرش پاسخهای بدتر را کاهش دهد. کاهش دما هر چه کندتر انجام شود بهینه سازی بهتر صورت می گیرد ضمن اینکه در تکرارهای اولیه برای الگوریتم، مقدار T باید به اندازه کافی بزرگ باشد تا امکان پذیرش جوابهای بدتر فراهم شود، به این ترتیب امکان قرار گرفتن در تله می نیمم محلی کاهش می یابد (Pirlot(1996).

بکارگیری الگوریتم SA برای محاسبه حد بالا در مسأله الگوی برش

الگوریتم پیشنهادی شامل ۴ بخش خواهد بود:

۱- یافتن جواب اولیه.

۲- یافتن جوابهای بعدی در همسایگی جوابهای قبلی.

۳- فرآیند کاهش دما.

۴- بررسی شروط نهایی برای پایان دادن به الگوریتم.

قبل از اینکه به تفصیل شرح الگوریتم بیان شود ذکر یک نکته که در طول الگوریتم از آن استفاده خواهد شد ضروری است.

هر الگوی برشی را می توان به صورت یک بردار نمایش داد. فرض کنید A_0 مستطیل اولیه باشد و m قطعه کوچکتر مورد تقاضا باشند. در این صورت اگر نتیجه حاصل از برش به ترتیب a_1, a_2, \dots, a_m قطعه برای مستطیلهای نوع $1, 2, \dots, m$ باشد، نتیجه الگوی برش را می توان به صورت بردار زیر نمایش داد:

$$A = (a_1, a_2, \dots, a_m)$$

با توجه به توضیح بالا و فرض اینکه ارزش مطلوبیت

یعنی الگوی برش شامل a_i قطعه نوع i است و تعداد قطعات تولید شده برای سایر انواع قطعات صفر است. x_0 بردار مبدأ ما خواهد بود. در الگوریتم SA با مسأله می نیمم سازی روبرو هستیم. با قرار دادن بردار x_0 در تابع هدف مسأله (۶) و ضرب کردن آن در عدد ۱- مقدار F^* و x^* در این مرحله از الگوریتم محاسبه می شوند:

$$F^* = F(x_0)$$

$$x^* = x_0$$

n ، پارامتر شمارنده الگوریتم برابر با عدد ۱ در نظر گرفته می شود.

۲- مقداری دلخواه که کارآیی لازم را طبق تعریف داشته باشد برای پارامتر دمای اولیه معین می شود.

$$T_0 = 10$$

۳- انتقال از بردار فعلی به برداری در همسایگی بردار اولیه.

مقادیر زیر بهنگام می شوند:

$$x_n = x^*$$

$$F_n = F^*$$

در این مرحله از الگوریتم برداری در همسایگی بردار اولیه (بردار x_0 در گام اول الگوریتم) تولید می شود. برای یافتن بردار بعدی در همسایگی بردار فعلی در هر گام از الگوریتم به صورت زیر عمل می شود.

۳-۱: ابتدا یک اندیس از درآیه های بردار فعلی به صورت تصادفی انتخاب می شود. برای این منظور عددی تصادفی بین ۱ تا $m+1$ تولید می شود. با در نظر گرفتن جزء صحیح عدد بدست آمده در واقع اندیس درآیه ای از بردار به صورت تصادفی انتخاب می شود که لزوماً باید اندیسی غیر از اندیس انتخاب شده در گام قبل باشد و آنرا j می نامیم.

۳-۲-۱: انجام عمل انتقال:

برای انجام عمل انتقال درآیه های بردار اولیه به صورت

زیر ویرایش می شوند:

$$a_j := a_j - 1$$

$$a_i := a_j + 1$$

که به این ترتیب در گام اول بردار مقصد (x_1) به صورت

زیر خواهد بود:

$x_1 := (0, 0, \dots, (a_i - 1), 0, \dots, (a_j + 1), \dots, 0)$
البته در گامهای بعدی نیز به همین صورت عمل انتقال انجام می شود ولی لزوماً تمام درآیه های غیر از i و j در بردار مقصد صفر نخواهند بود.
۳-۲-۲: اگر بردار جدید شدنی بود (در فضای پاسخ مسأله قرار داشت) مقدار تابع هدف به ازای آن محاسبه می شود. با توجه به شرط فضای پاسخ مسأله (۶) داریم:

$$F(x_n) \leq L.W \Rightarrow F_n = F(x_n)$$

در غیر این صورت به گام ۳-۱ برگشته بردار مقصد دوباره بصورت تصادفی انتخاب می شود.

۳-۲-۳: محاسبه تابع هدف:

اگر مقدار تابع هدف به ازای بردار جدید کاهش پیدا کرد مقدار F^* و x^* جایگزین مقادیر جدید می شوند.

$$F_n < F^* \Rightarrow \begin{cases} F^* = F_n \\ x^* = x_n \end{cases}$$

در غیر اینصورت اختلاف مقدار فعلی تابع هدف با بهترین مقدار تابع هدف تا این مرحله با توجه به رابطه (۷) محاسبه می شود:

$$\Delta E := (-F(x_n)) - F^*$$

عدد تصادفی p بین صفر و یک تولید می شود:

$$P = \text{Rand}[0, 1]$$

و تابع احتمال بولتزمن به ازای مقدار بدست آمده از تابع هدف و با توجه به رابطه (۸) محاسبه می شود:

$$P(x_n) = e^{(-\Delta E/T)}$$

اگر p کمتر از احتمال بدست آمده در این مرحله باشد مقادیر F^* و x^* با این مقادیر تعویض می شوند (پاسخهای بدتر پذیرفته می شوند).

$$p < P(x_n) \Rightarrow \begin{cases} F^* = F_n \\ - \\ x^* = x_n \end{cases}$$

۴- شرط خاتمه الگوریتم:

در صورتی که به میزان حداقل $\epsilon\%$ از دفعات تکرار الگوریتم جواب بهبود نیافته باشد الگوریتم خاتمه می یابد (Pirlot(1996)

بر بهینه سازی محلی است. در این روش برای هر گره تمامی برشهای ممکن بکار گرفته می شوند و در نتیجه تمامی برشهای منشعب از گره اولیه تولید می شوند. پس از آن با استفاده از روش شاخه-کران بهترین مسیر انتخاب می شود و بقیه مسیرها کنار گذاشته می شوند (Morabito, Arenalez (1992).

الگوریتم جستجوی گراف and-or

گام ۱: فرض کنید Root مجموعه تمامی گره هایی باشد که در انتظارگسترده شدن باشند (در گام اول فقط شامل صفحه اولیه است).

گام ۲: مادامی که Root یک مجموعه تهی نباشد:

۱-۲: فرض کنید N اولین گرهی باشد که منتظرگسترده شدن است. تمامی گره های قابل انشعاب N باید تولید شوند.

۲-۲: بهترین مسیر ممکن منشعب از گره اولیه انتخاب و از بقیه صرف نظر شود. اگر گره هایی در مجموعه Root وجود دارند که همسطح گره N هستند و منتظر گسترده شدن هستند در مجموعه Root قرار می گیرند.

نتایج محاسباتی

الگوریتم hillclimbing با بکارگیری روش پیشنهاد شده در این مقاله برای محاسبه حد بالا و روش Morabito, Arenalez(1992) برای محاسبه پنج مسأله از قبل حل شده به طور مجزا بکار گرفته شده است. این پنج مسأله برگرفته از مرجع فوق می باشند که پاسخ بهینه آنها در همان مقاله در دسترس است. شرح مسائل در جدول (۱) ارائه شده است. در جدول (۱) سطر دوم شامل ابعاد صفحه اولیه و ۱۰ سطر بعدی شامل قطعات مورد تقاضا در هر مسأله است. در تمامی این مسائل ارزش هر قطعه برابر با حاصلضرب طول در عرض آن قطعه است.

اعداد بدست آمده در جدول (۲) بیانگر نتیجه حل مسائل جدول (۱) با استفاده از روش پیشنهاد شده در این مقاله و روش Morabito, Arenalez(1992) است.

۵- شرط کاهش دما:

اگر k بار الگوریتم تکرار شده باشد برای کاهش دما ضریب α تعیین می شود.

$$\alpha \in (0,1)$$

هر چه α به عدد یک نزدیکتر باشد فرآیند کاهش دما کندتر انجام خواهد شد (Pirlot(1996). دما کاهش داده می شود و پس از بهنگام شدن مقادیر زیر، محاسبات گام ۳ تا برقراری شرط خاتمه الگوریتم انجام می شود.

$$T_{n+1} = \alpha \cdot T_n$$

$$k=0$$

$$n=n+1$$

بکارگیری روش شاخه کران در گراف

and-or

پس از آنکه فضای جواب الگوی برش دو بعدی برای مسأله برش با کمک گراف and-or مشخص شد گام نهایی برای حل مسأله، یافتن الگوی بهینه برش خواهد بود. نکته اساسی در جستجوی گراف and-or بکارگیری روش شاخه-کران در جهت جلوگیری از شاخه سازیهای بیهوده است.

فرض کنید $B(N)$ ارزش بهترین مسیر شناخته شده برای گره N باشد، (در مورد گره اولیه $B(N)$ همان حد پایین گره است که بر اساس بهترین الگوی برش متشابه بدست می آید) در اینصورت هر دو گره N_1 و N_2 که با یک and-arc از گره N منشعب شده اند نشان دهنده یک برش موفق گیوتینی خارج شده از N هستند به این ترتیب اگر $B(N) < \varphi^0(N_1) + \varphi^0(N_2)$ باشد، مقدار $\varphi^0(N_1) + \varphi^0(N_2)$ جایگزین $B(N)$ می شود. طبیعتاً اگر $B(N) > u(N_1) + u(N_2)$ باشد نیازی به انجام این برش نخواهد بود چرا که در نتیجه بکارگیری این برش جواب بهتری ایجاد نمی شود.

استراتژی جستجوی گراف and-or

پس از آنکه ارزش بکارگیری روش شاخه-کران در گراف and-or مشخص شد به یک استراتژی جستجو برای در نوردیدن گراف نیاز است. به همین منظور از استراتژی جستجوی hillclimbing استفاده می شود. روش جستجوی hillclimbing یک روش تجربی مبتنی

در اینصورت اگر ارزش مطلوبیت هر قطعه مثل i را برابر با ابعاد آن در نظر بگیریم:

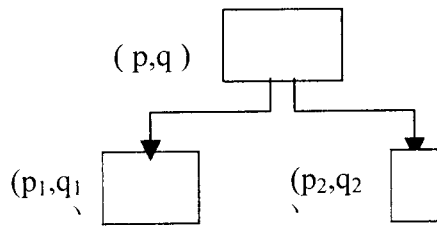
$$u(p,q) = (p,q) \cdot \max \{ (l_i \cdot w_i) / (l_i \cdot w_i) ; i \in M(N) \}$$

که به این ترتیب:

$$u(p,q) = p \cdot q$$

(۹)

فرض کنید برش موفقی انجام داده ایم که نتیجه آن تولید دو گره بوده است. در اینصورت:



با توجه به رابطه (۹):

$$u(p,q) = p \cdot q$$

$$u(p_1, q_1) = p_1 \cdot q_1$$

$$u(p_2, q_2) = p_2 \cdot q_2$$

شرط شاخه سازی آنست که حد پایین گره ریشه ای از مجموع حدود بالای گرههای منشعب کمتر باشد. با توجه به رابطه (۴) داریم:

$$\varphi^0(p,q) \leq u(p_1, q_1) + u(p_2, q_2)$$

با توجه به رابطه بالا و با توجه به رابطه (۸) می توان نتیجه گیری زیر را انجام دهیم:

$$\varphi^0(p,q) \leq u(p,q) = p \cdot q$$

یعنی فرض Morabito, Arenalez(1992) برای شاخه سازی آنست که حد پایین گره ریشه ای از مساحت آن کمتر باشد که البته همواره این شرط برقرار است. به عبارت دیگر با توجه به تعریف حد پایین هر گره، همواره حد پایین گره ریشه از مجموع مساحتهای دو گره منشعب کوچکتر است و در نتیجه روش شاخه کران در عمل از شاخه سازی که نتیجه بهتری در بر نخواهند داشت جلوگیری نخواهد کرد. البته برای جلوگیری از شاخه سازی بی مورد نویسندگان مقاله فوق از یک روش تجربی استفاده کرده اند که روش موفقی نیز بوده است.

اعدادی که در این جدول به عنوان پاسخ مسأله به روشهای بالا آورده شده اند، بیانگر مجموع ارزشهای گرههای نهایی (رجوع شود به رابطه (۱)) بهترین مسیر کاملی است که الگوریتم hillclimbing پس از گسترانیده شدن گراف and-or به روشهای مذکور به آنها دست پیدا می کند. سطر مربوط به تعداد گرههای پیموده شده نیز نشان می دهد که روش پیشنهادی این مقاله فضای جواب کمتری را برای بدست آوردن بهترین مسیر کامل جستجو می کند.

چنانچه دیده می شود در مسائل ۱ و ۵ با روش پیشنهاد شده در این مقاله علاوه بر دستیابی به جواب، بهینه، تعداد گره هایی که برای بدست آوردن جواب، جستجو شده اند بسیار کمتر هستند. در مورد مسائل ۲ و ۳ و ۴ نیز هر دو روش به یک جواب دست پیدا کرده اند ولی روش پیشنهادی مقاله با جستجوی گره های کمتری به این جوابها دست پیدا کرده است.

الگوریتمی که تحت عنوان hillclimbing برای جستجوی گراف and-or مطرح کردیم ماهیت ابتکاری دارد. به این ترتیب هر گره وقتی به صورت کامل منشعب شد به صورت محلی بهینه سازی می شود (بهترین انشعاب انتخاب می شود و ما بقی کنار گذاشته می شوند). در این صورت ممکن است گره هایی وجود داشته باشند که در انشعابهای بعدی جواب بهتری بدهند ولی در اثر بکارگیری این روش کنار گذاشته شوند.

Morabito, Arenalez(1992) به این مسأله اشاره کرده اند که برای جلوگیری از پیچیدگی در حل مسأله از در نظر گرفتن شرط عدد صحیح بودن برای محاسبه حد بالا جلوگیری کرده اند. با توجه به این فرض که $\forall i ; a_i$ عدد صحیح نیست و با توجه به پاسخی که نویسندگان فوق برای مسأله بیان کرده اند مقایسه گرهها به روش شاخه کران در عمل بیهوده خواهد بود.

Morabito, Arenalez (1992) پاسخ مسأله (۵) را با توجه به تعریف $M(N)$ در رابطه (۴) به این صورت بیان می کنند:

$$u(p,q) = (p,q) \cdot \max \{ v_i / (l_i \cdot w_i) ; i \in M(N) \}$$

جدول ۱: شرح مسائل حل شده. برگرفته از منبع (Arenalez, Morabito 1992).

	مسئله ۱	مسئله ۲	مسئله ۳	مسئله ۴	مسئله ۵
(L,W)	(۱۰۰.۱۵۶)	(۲۵۳.۲۹۴)	(۳۱۸.۴۷۳)	(۵۰۱.۵۵۶)	(۷۵۰.۸۰۶)
(l _i ,w _i) i=1,2,...,10	(۲۷.۳۲)	(۱۳۰.۱۳۰)	(۱۴۱.۱۲۸)	(۱۳۶.۱۱۳)	(۲۰۵.۱۶۳)
	(۱۶.۵۰)	(۴۹.۱۳۴)	(۱۰۹.۱۵۱)	(۸۸۱.۱۷۷)	(۱۲۱.۲۵۷)
	(۴۷.۶۶)	(۴۵.۵۲)	(۷۶.۱۴۲)	(۲۳۷.۲۳۷)	(۳۵۶.۳۴۳)
	(۳۷.۴۶)	(۶۴.۱۳۴)	(۲۰۴.۱۲۴)	(۱۸۴.۱۶۳)	(۲۷۶.۲۳۶)
	(۳۱.۵۱)	(۳۸.۱۰۰)	(۱۱۰.۱۶۵)	(۱۵۳.۱۸۲)	(۲۳۰.۲۶۴)
	(۴۷.۵۱)	(۱۲۱.۱۴۶)	(۱۴۲.۱۶۸)	(۲۳۳.۱۸۳)	(۳۵۰.۲۶۶)
	(۴۸.۴۴)	(۴۸.۱۲۳)	(۸۲.۲۳۸)	(۲۴۱.۱۵۵)	(۳۶۱.۲۲۵)
	(۴۹.۴۵)	(۸۳.۷۹)	(۱۶۹.۷۵)	(۲۴۶.۱۶۰)	(۳۶۹.۲۳۲)
	(۵۶.۷۸)	(۴۲.۱۸۷)	(۱۶۵.۱۸۵)	(۲۹۶.۲۷۸)	(۴۴۵.۴۰۴)
	(۲۱.۸۷)	(۱۱۲.۹۲)	(۱۶.۱۸۴)	(۱۰۳.۳۱۰)	(۱۵۵.۴۴۹)

جدول ۲: مقایسه عددی بکارگیری الگوریتم SA برای محاسبه حد بالا در الگوریتم جستجو با روش Morabito, Arenalez (1992).

مسئله ۵	مسئله ۴	مسئله ۳	مسئله ۲	مسئله ۱	
۵۷۷۸۸۲	۲۶۵۷۶۸	۱۴۲۸۱۷	۷۲۱۷۲	۱۵۰۲۴	جواب بهینه
۵۷۷۸۸۲	۲۶۱۷۵۸	۱۴۰۰۰۰	۷۲۱۴۲	۱۵۰۲۴	جواب به روش مقاله حاضر
۵۷۷۸۸۲	۲۶۱۷۵۸	۱۴۰۰۰۰	۷۲۱۴۲	۱۵۰۲۴	جواب به روش Morabito, Arenalez(1992)
۱۰۴	۴۱	۱۰۴	۳۹۰	۸۷	تعداد گره های پیموده شده به روش مقاله حاضر
۱۷۶	۱۷۹	۲۱۰	۱۰۷۳	۲۹۹	تعداد گره های پیموده شده به روش Morabito, Arenalez(1992)

جدول ۳: مقایسه عددی بکارگیری الگوریتم SA برای محاسبه حد بالا در الگوریتم جستجو با روش (Morabito, Arenalez (1992) در حل مسائل برگرفته از مقاله (Beasley(1985).

شماره مسأله	جواب بهینه Beasley (1985)	جواب به روش Arenalez, Morabito (1992)	جواب به روش مقاله حاضر	تعداد گرههای پیموده شده به روش	تعداد گرههای پیموده شده به روش مقاله حاضر
۱	۵۶۴۶۰	۵۶۴۶۰	۵۶۴۶۰	۹۱	۳
۲	۶۰۰۵۳۶	۶۰۰۷۶	۵۹۴۷۶	۱۹۹	۱۰۵
۳	۶۱۰۳۶	۵۸۹۷۷	۵۷۸۷۶	۴۰۶	۱۷۱
۴	۶۱۶۹۸	۶۱۶۹۸	۶۱۶۹۸	۱۰۹۸	۱۹۸
۵	۲۴۶۰۰۰	۲۴۶۰۰۰	۲۴۶۰۰	۴۴	۱
۶	۲۳۸۹۹۸	۲۲۵۷۷۱	۲۲۵۷۷۱	۱۷۱	۳۹
۷	۲۴۲۵۶۷	۲۴۲۵۶۷	۲۳۰۶۲۰	۳۰۶	۶۵
۸	۲۴۶۶۳۳	۲۴۳۳۷۹	۲۳۷۹۶۸	۱۲۲۹	۳۲۸
۹	۹۷۱۱۰۰	۹۷۱۱۰۰	۹۷۱۱۰۰	۳۹	۹
۱۰	۹۸۲۰۲۵	۹۶۳۸۴۸	۹۳۴۸۹۵	۲۱۱	۲۵
۱۱	۹۸۰۰۹۶	۹۶۰۱۴۸	۹۶۰۱۴۸	۳۴۸	۱۸۹
۱۲	۹۷۹۹۸۶	۹۶۷۰۴۴	۹۴۵۲۲۶	۷۰۲	۱۰۱

جدول ۴: نتایج حل مسأله ۱۳ از مقاله (Beasley(1985).

شماره مسأله	جواب به روش Beasley (1985)	جواب به روش Arenalez, Morabito(1992)	جواب به روش مقاله حاضر	تعداد گرههای پیموده شده به روش مقاله حاضر	تعداد گرههای پیموده شده به روش مقاله حاضر
۱۳	۸۸۶۸۹۵۰	۸۹۹۷۷۸۰	۸۹۹۷۷۸۰	۵۲۳۷	۳۴۱۶

جستجوی hillclimbing در گراف and-or این مسأله تا حدی اجتناب ناپذیر است.

در مسأله شماره ۱۳ که نتایج حل آن در جدول (۴) ارائه شده است، یک مسأله واقعی برگرفته شده از مقاله (Beasley(1985) به هر دو روش حل شده است که نتایج آن در این جدول گزارش شده است. روش حل این مسأله در مقاله مذکور روشی ابتکاری است که به سبب برخی محدودیتها در حل این مسأله با استفاده از روش حل اصلی مقاله ارائه شده است. نکته قابل توجه این که روش پیشنهادی این مقاله توانسته است جواب بهتری برای مسأله مذکور بدست آورد.

نتیجه گیری

در این مقاله گراف and-or برای نمایش فضای

برای مقایسه بیشتر بین دو روش و اطمینان از کارایی الگوریتم SA برای محاسبه حد بالای هر گره در گراف and-or، الگوریتم پیشنهاد شده در این مقاله و الگوریتم پیشنهاد شده توسط (Morabito, Arenalez(1992) برای حل سیزده مسأله از پیش حل شده توسط (Beasley(1985) بکار گرفته شده است که نتایج حل آنها در جدول (۳) و (۴) ارائه شده است. چنانچه در جدول شماره (۳) مشاهده می شود در مسائل شماره ۱، ۴، ۵ و ۹ الگوریتم پیشنهاد شده در این مقاله به پاسخ بهینه دست یافته است. در سایر موارد پاسخهای بدست آمده با جواب بهینه مسأله اندکی اختلاف دارد که با توجه به کاهش فضای مسأله در روش پیشنهادی این مقاله و با توجه به ماهیت ابتکاری روش

می کند. الگوریتم hillclimbing برای جستجو در گراف معرفی شد و چندین مسأله از قبل حل شده به روش پیشنهادی در این مقاله و روشهای قبلی حل شدند. با بکارگیری روش پیشنهاد شده در مقاله موفق شدیم به جوابهای بهینه یا نزدیک به بهینه با درنوردیدن فضای کمتری از فضای پاسخ مسأله دست پیدا کنیم. نتایج جداول ۲،۳ و ۴ نشان می دهد که سطح قابل قبولی از صفحات اولیه به قطعات مورد تقاضا تخصیص داده می شود و با بکارگیری روش پیشنهاد شده در مقاله ضمن کاهش فضای پاسخ مسأله، ضایعات حاصل از برش کاهش می یابد.

جواب مسأله برش دو بعدی گیوتینی نامحدود و با هدف کاهش ضایعات حاصل از برش ارائه شد و روش شاخه کران برای جستجو در گراف فوق معرفی شد. با بکارگیری الگوریتم Simulated Annealing موفق شدیم فرض عدد صحیح بودن تعداد قطعات مورد تقاضا را در مدل ریاضی بیان شده برای حد بالا بگنجانیم و به این ترتیب حد بالایی گره به صورت واقعی با این الگوریتم که ماهیتی ابتکاری دارد محاسبه شد. نتیجه محاسبه حد بالا به این روش آنست که مقایسه بین گره اولیه و گره های منشعب از آن در گراف and-or به صورت واقعی تر انجام می شود و در نتیجه از شاخه سازیهای بی مورد جلوگیری

مراجع

- 1 - Morabito, R. N. and Arenalez, M. N. (1992). "An and-or-graph approach for two-dimensional cutting problems." *European Journal of Operational Research*, Vol. 58, PP. 263-271.
- 2 - Morabito, R. N. and Arenalez, M. N. (1995). "An and/or-graph approach of two-dimensional non-guillotine cutting problems." *European Journal of Operational Research*, Vol. 84, PP. 599-617.
- 3 - Beasley, J. E. (1985). "Algorithms for unconstrained two-dimensional guillotine cutting." *Operational Research Society*, Vol. 36, No. 4, PP. 297-306.
- 4 - Herz, J. C. (1972). "Recursive computational procedure for two-dimensional stock cutting." *IBM Journal of Research and Development*, Vol. 16, PP. 462-469.
- 5 - Alvarez Valdez, R., Parajon, A. and Tamarit, J. M. (2000). "A tabu search algorithm for large-scale guillotine two-dimensional cutting problem." *Annals of Operations research*, Vol. 96, PP. 245-254.
- 6 - Mhand Hifi. (1997). "The DH/KD algorithm: A hybrid approach for unconstrained two-dimensional cutting problems." *European Journal of Operational Research*, Vol. 97, No. 1, PP. 41-52.
- 7 - Rich, E. (1983). *Artificial intelligence*, McGraw-Hill, New York.
- 8 - Marc Pirlot. (1996). "General local search methods." *European Journal of Operational Research*, Vol. 92, PP. 493-511.
- 9 - Diomidis Spinellis, Chrissoleon Papadopoulos and MacGregor Smith, J. (2000). "Large production Line Optimization using simulated annealing." *International Journal of Production Research*, Vol. 38, No. 3, PP. 509-541.
- 10 - Irkpatrickm, S., Gellat, C. and Vecchi M. (1983). "Optimization by simulated annealing." *Science*, Vol. 220, PP. 671-679.
- 11 - bduelmola, A. I. and Taboun, S. M. (2000). "A simulated annealing algorithm for designing cellular manufacturing systems with productivity consideration." *Productin Planning & Control*, Vol. 11, No. 6, PP. 589-597.

واژه های انگلیسی به ترتیب استفاده در متن

- | | | | |
|------------------|-----------------|-------------------------|------------------|
| 1 - Non-staged | 2 - Tabu search | 3 - Simulated Annealing | 4 - Hillclimbing |
| 5 - And-or graph | 6 - And graph | 7 - And arc | |