مهندسی صنایع و مدیریت شریف، (زمستان ۱۳۹۶)
دوره‌ی ۱-۳۳، شماره‌ی ۲/۲، ص. ۸۵-۹۴

Original Article

ارائه‌ی یک مدل برنامه‌ریزی عدد صحیح جدید و یک حد پایین مناسب برای مسائل ماشین‌های موازی یکسان با معیار کمینه‌سازی دیرکرد کل کارها

سید محمدتقی فاطمی‌قمی* (استاد)
واحد اداک (دانشجوی کارشناسی ارشد)
دانشکده‌ی مهندسی صنایع و سیستم‌های مدیریت، دانشگاه صنعتی امیرکبیر
فریبرز جولای (استاد)
دانشکده‌ی مهندسی صنایع، دانشگاه تهران

در این مقاله مسئله‌ی توالی عملیات برروی ماشین‌های موازی یکسان با معیار کمینه‌سازی مجموع دیرکرد کل کارها بررسی می‌شود. مدل برنامه‌ریزی عدد صحیح مختلط کارایی برای مسئله‌ی مورد نظر ارائه می‌شود؛ سپس مدلی پیشنهادی برای به‌دست‌آوردن حد پایین بهتر و کاراتر از یکی از حدود پایین موجود در پیشینه‌ی پژوهش‌های مسئله ارائه می‌شود. مسئله‌ی ماشین‌های موازی یکسان با تابع هدف کمینه‌سازی مجموع دیرکرد کل کارها تعمیم‌یافته‌ی مسئله‌ی تک ماشینی است و این مسئله جزء مسائل NP-hard دسته‌بندی می‌شود. از این‌رو مدل ارائه‌شده توانایی حل بهینه‌ی مسائل با اندازه‌ی بزرگ در زمان منطقی را ندارد. به‌همین‌دلیل برای حل مسئله در اندازه‌های متوسط و بزرگ و نیز ارزیابی کارایی حدپایین به‌دست‌آمده از مدل پیشنهادی و حد پایین موجود در پیشینه، الگوریتم فراابتکاری شبیه‌سازی تبرید اصلاح شده‌یی که برای اولین بار از عملگر تقاطع و جهش برای ایجاد جواب همسایگی بهره می‌برد، ارائه می‌شود.

واژگان کلیدی: ماشین‌های موازی، مجموع دیرکرد کل کارها، الگوریتم شبیه‌سازی تبرید اصلاح‌شده، برنامه‌ریزی عدد صحیح مختلط.

fatemi@aut.ac.ir
v.adak65@gmail.com
fjolai@ut.ac.ir

۱. مقدمه

مجموعه‌یی از N کار $i = \{۱, ۲, ..., N\}$ وجود دارد که برای زمان‌بندی برروی m ماشین یکسان موازی در نظرگرفته شده‌اند. وقفه در انجام کارها مجاز نیست و هر ماشین در هر زمان تنها قادر به انجام یک کار است و هر کار تنها باید روی یک ماشین پردازش شود. زمان انجام فرایند هر کار و موعد تحویل آن مشخص است و با p_i و d_i نشان داده می‌شوند.[۱] برای نشان‌دادن این مسئله طبق نمادگذاری گراهام و همکاران از نماد $P_m || \sum T_i$ استفاده می‌شود[۲] که در آن T_i زمان دیرکرد کار i و برابر با $T_i = \max(۰, C_i - d_i)$ است. C_i زمان اتمام پردازش کار i است.

با توجه به اینکه مسئله‌ی ماشین‌های موازی تعمیم‌یافته‌ی مسئله‌ی تک‌ماشینی است و مسئله‌ی تک‌ماشینی با تابع هدف کمینه‌سازی مجموع دیرکرد کل کارها یک مسئله‌ی NP-hard است،[۱] به‌دست‌آوردن یک حد پایین سریع و با جواب نزدیک به جواب بهینه برای این گونه مسائل حائز اهمیت است. زیرا برای حل مسائل با اندازه‌ی بزرگ با روش‌های ابتکاری و شاخه‌وکران نیاز به یک حد پایین کارا ضروری به نظر می‌رسد.

بیلگ و همکاران[۳] یک الگوریتم جست‌وجوی ممنوعه را برای مسئله‌ی کمینه‌سازی دیرکرد کل چندماشینی به‌کار گرفته‌اند. آنقیلنفی و پائولوچی[۴] مسئله‌ی مطالعه‌شده توسط بیلگ و همکاران را بررسی کردند و یک الگوریتم فراابتکاری ترکیبی جدید را برای آن پیشنهاد کردند که در آن، ویژگی‌های الگوریتم‌های جست‌وجوی ممنوعه، شبیه‌سازی تبرید، و جست‌وجوی همسایگی ترکیب شده است.

عزیز اوغلو و کرجا[۵] مسئله‌ی زمان‌بندی کارها برروی ماشین‌های موازی یکسان با کمینه‌سازی تأخیر کلی را بررسی کردند. براساس ادعای نویسندگان، کار آن‌ها اولین کاری است که این مسئله را بهینه می‌کند. آنها یک الگوریتم شاخه‌وکران را برای حل مسئله طراحی کردند. این الگوریتم با یک حد بالای اولیه شروع می‌شود که با یکی از رویکردهای زیر به‌دست می‌آید:

۱. فهرستی براساس کوتاه‌ترین زمان پردازش؛

* نویسنده مسئول

تاریخ: دریافت ۱۳۹۴/۶/۱۸، اصلاحیه ۱۳۹۴/۱۱/۱۷، پذیرش ۱۳۹۵/۲/۶.

۲. فهرستی براساس زودترین تاریخ تحویل؛

۳. فهرست α که کارها را براساس مقادیر α_i به‌صورت غیرنزولی مرتب می‌کند $(\alpha_i = \max\{p_i, d_i\})$.

برای به‌دست‌آوردن حد پایین، مسئله را به این‌صورت آزادسازی می‌کنند که بریدگی کارها مجاز باشد. درواقع، بخشی از یک کار می‌تواند در یک زمان انجام شود و در زمان دیگر باقی مانده‌ی کار انجام شود. با این فرض که کارها براساس کوتاه‌ترین زمان پردازش مرتب شده‌اند، حد پایین برابر است با: $LB = \sum_i \max\{0, \frac{\sum_{j=1}^{i} P_j}{m} - d_{[i]}\}$ که برای تمام iها $d_{[i]} \le d_{[i+1]}$ است.

الگوریتم پیشنهادی فقط قادر به حل مسائل با حداکثر ۱۵ کار و ۳ ماشین است و نمی‌تواند مسائلی با اندازه‌ی بزرگ‌تر (مانند ۱۵ کار و ۴ ماشین ـ ۲۰ کار و ۲ ماشین) را در مدت زمان کم‌تر از ۱۵ دقیقه حل کند.

تاناکا و آراکی[۶] یک الگوریتم شاخه‌وکران جدید را برای مسائل زمان‌بندی ماشین‌های موازی یکسان طراحی کردند. آنها در الگوریتم خود از روش آزادسازی لاگرانژ برای به‌دست‌آوردن یک حد پایین استفاده کردند. آنها با درنظرگرفتن یک ضریب، مربوط به محدودیت در دسترس‌بودن ماشین و اضافه‌کردن به تابع هدف به حل مسئله پرداختند. آنها برای به‌دست آوردن مقدار بهینه‌ی ضرایب لاگرانژ از دوگان مسئله‌ی آزادشده استفاده کرده‌اند. نتایج محاسباتی آنها نشان می‌دهد که بسیاری از مثال‌ها بدون احتیاج به شاخه‌زنی در همان گره مادر حل می‌شوند. درحالی‌که این الگوریتم قادر است فقط مسائل با تعداد حداکثر ۲۵ کار با هر تعداد ماشین را حل کند. همچنین شیم و کیم[۷] یک الگوریتم شاخه‌وکران را برای مسئله‌ی ماشین‌های موازی ارائه کردند. در این الگوریتم ابتدا برای به‌دست آوردن یک حد بالای اولیه از الگوریتم شبیه‌سازی تبرید (SA) استفاده می‌شود. به‌منظور انتخاب گره برای شاخه‌زنی، روش depth-first مورد استفاده قرار می‌گیرد. الگوریتم شاخه‌وکران پیشنهادی آنها قادر است مسائل تا سقف ۳۰ کار و تا سقف ۵ ماشین را در مدت زمان قابل قبول حل کند. در ادامه منسندیک و همکاران[۸] مسئله‌ی کمینه‌سازی دیرکرد کل کارها را بررسی کردند که در آن کارها می‌توانند فقط در زمان‌های ثابت ازپیش‌تعیین‌شده تحویل شوند. آن‌ها با استفاده از الگوریتم شاخه‌وکران مسئله را در اندازه‌ی کوچک حل کردند و یک حد پایین با استفاده از به تأخیرانداختن زمان تحویل برخی از کارها به‌دست آوردند. آنها، برای حل مسئله در اندازه‌ی بزرگ، یک الگوریتم جست‌وجوی ممنوعه و یک الگوریتم ترکیبی ژنتیک را ارائه کردند.

کاوالیو و ورنر[۹] راه‌حلی برای کمینه‌سازی تأخیر کلی در مسائل ماشین‌های موازی پیشنهاد کردند که در آن تمام کارها تاریخ تحویل یکسانی دارند درحالی‌که المغربی و پارک[۱۰] مسائلی را مطرح کردند که در آن تاریخ تحویل هر کار با زمان پردازش آن کار برابر است. آرکین و راندی[۱۱] الگوریتمی را برای کمینه‌سازی تأخیر وزن‌دار مربوط به مسائلی که در آن‌ها وزن هر کار با کسری از زمان پردازش آن کار برابر است، ارائه کردند. انلو و ماسون[۱۲] برای مسئله‌ی ماشین‌های موازی بدون قطع کار برای توابع هدف کمینه‌سازی مجموع وزنی زمان اتمام کارها و میکسپن، به‌طور جداگانه یک مدل برنامه‌ریزی عدد صحیح مختلط ارائه کردند.

با توجه به پیشینه‌ی پژوهش‌های مرتبط با مسئله‌ی زمان‌بندی ماشین‌های موازی ارائه‌ی یک مدل جدید برنامه‌ریزی خطی عدد صحیح مختلط کارا برای مسئله‌ی مذکور با تابع هدف کمینه‌سازی دیرکرد کل کارها مناسب است و نیز توسعه یا ارائه‌ی حد پایین‌هایی که بهتر از حد پایین‌های موجود در پیشینه‌ی پژوهش باشد با توجه به پیچیدگی مسئله در اندازه‌های متوسط و بزرگ حائز اهمیت است. از این‌رو، در این مقاله با ارائه‌ی یک مدل برنامه‌ریزی عدد صحیح مختلط برای مسئله‌ی با تابع هدف کمینه‌سازی مجموع دیرکرد کل کارها به حل مسئله در ابعاد کوچک به‌صورت بهینه پرداخته می‌شود و یک مدل که حد پایینی بهتر از حد پایین اوغلو و کرجا به دست می‌دهد، نیز ارائه می‌شود. همچنین برای مقایسه‌ی حد پایین‌ها و حل مسئله در اندازه‌ی متوسط و بزرگ الگوریتم شبیه‌سازی اصلاح شده‌یی ارائه می‌شود.

مقاله ساختار زیر را دارد. در بخش ۲ مدل برنامه‌ریزی عدد صحیح مختلط و مدل ارائه‌شده برای حد پایین بیان می‌شود. در بخش ۳ الگوریتم شبیه‌سازی تبرید اصلاح‌شده آورده می‌شود. بخش ۴ به نتایج محاسباتی برای تمام حالات مختلف ارائه‌شده می‌پردازد. بخش آخر به نتیجه‌گیری و پیشنهادهایی برای پژوهش‌های آتی اختصاص داده شده است.

۲. مدل ریاضی

۱.۲. مدل پیشنهادی

در این بخش ابتدا به بیان تعاریف و نمادها پرداخته می‌شود و سپس مدل برنامه‌ریزی عدد صحیح مختلط ارائه و سپس مدل پیشنهادی برای حد پایین آورده می‌شود.

۲.۲. تعاریف و نمادها

۱.۲.۲. فرضیات مربوط به کارها و ماشین

- ماشین‌ها در هر زمان به‌طور پیوسته در دسترس هستند و در ابتدا می‌توانند هر یک از کارها را انجام دهند.
- ماشین‌ها زمان بی‌کاری و توقفات مانند خرابی و تعمیر ندارند.
- با شروع یک کار عملیات تا انتهای کار ادامه پیدا می‌کند؛ به‌عبارت دیگر انقطاع در کارها جایز نیست.
- زمان انجام هر کار و زمان موعد تحویل آنها از قبل مشخص و غیر قابل تغییر است.

۲.۲.۲. نمادهای به‌کاررفته

n: تعداد کار بدون انقطاع برای پردازش روی یکی از l ماشین در نظر گرفته شده است. برای آزادسازی و به‌دست آوردن حد پایین با استفاده از مدل پیشنهادی زمان پردازش هرکار بر تعداد ماشین‌ها تقسیم شده و رو به بالا گرد شده است که به نحوی انقطاع کار در مدل پیشنهادی و نه در مدل برنامه‌ریزی عدد صحیح مختلط در نظر گرفته شده است؛

i, k: شمارنده‌ی کارها $\{1, 2, ..., n\}$؛

r: شمارنده‌ی ماشین‌ها $\{1, 2, ..., l\}$؛

M: یک عدد بزرگ مثبت؛

p_i: زمان انجام عملیات کار iام؛

d_i: موعد تحویل کار i؛

pt_i: زمان پردازش اصلاح‌شده، $pt_i = ceil(\frac{P_i}{l})$؛

m_k: kامین زمان انجام به ترتیب نزولی زمان پردازش اصلاح‌شده‌ی کارها؛

T_{ij}: حداقل دیرکرد کار iام اگر jامین کار انجام شود (i نشان‌دهنده‌ی کار و j نشان‌دهنده‌ی نوبت انجام کار i است).

متغیرهای تصمیم‌گیری

$Y_{i,k}$: متغیر دودویی است که وقتی کار i بعد از کار k انجام شود عدد ۱، و در غیر این‌صورت عدد صفر را می‌پذیرد، $i \in \{1, 2, ..., n-1\}, k > i$؛

$Z_{i,r}$: متغیر دودویی است که وقتی کار j توسط ماشین rام پردازش شود عدد ۱، و در غیر این‌صورت عدد صفر را می‌پذیرد، $r \in \{1, 2, ..., l\}$؛

C_i: متغیر پیوسته برای تعیین زمان تکمیل نهایی کار i؛

T_i: متغیر پیوسته برای تعیین دیرکرد کار i؛

$X_{i,j}$: اگر کار iام jامین کار انجام شود، مقدار ۱، و در غیر این‌صورت مقدار صفر را می‌گیرد.

۳.۲. مدل برنامه‌ریزی ریاضی عدد صحیح مختلط

$$\text{minimizing} = \sum_{i=1}^{n} T_i \tag{1}$$

s.t :

$$\sum_{r=1}^{m} Z_{i,r} = 1 \qquad \forall i \tag{2}$$

$$C_i \geq P_i \qquad \forall i \tag{3}$$

$$C_i \geq C_k + P_i - M*(1 - Y_{i,k}) - M*(2 - (Z_{i,r} + Z_{k,r})) \quad \forall r, i \in \{1, 2, ..., n-1\}, \ \ k > i \tag{4}$$

$$C_k \geq C_i + P_k - M*Y_{i,k} - M*(2 - (Z_{i,r} + Z_{k,r})) \quad \forall r, i \in \{1, 2, ..., n-1\}, \ \ k > i \tag{5}$$

$$T_i \geq C_i - d_i \qquad \forall i \tag{6}$$

$$T_i, C_i \geq 0 \qquad \forall i \tag{7}$$

$$Y_{i,k}, Z_{i,r} \in \{0, 1\} \tag{8}$$

رابطه‌ی ۱ تابع هدف است. که مقدار آن مجموع دیرکرد کل کارهاست. مجموعه‌ی محدودیت ۲ مشخص می‌کند هرکار باید روی یک ماشین از میان ماشین‌های موازی پردازش شود. مجموعه‌ی محدودیت ۳ مطمئن می‌سازد که زمان تکمیل هرکار بزرگ‌تر از زمان پردازش آن کار است. مجموعه‌ی محدودیت‌های ۴ و ۵ با هم بیان می‌کنند که اگر دو کار i و k روی یک ماشین rام پردازش شوند، از پردازش هم‌زمان آن دو کار جلوگیری می‌شود و کارها به نوبت روی ماشین rام پردازش می‌شوند. محدودیت ۶ زمان دیرکرد کارها را محاسبه می‌کند. رابطه‌های ۷ و ۸ متغیرهای تصمیم مدل را تعریف می‌کنند.

۴.۲. مدل پیشنهادی برای محاسبه‌ی حد پایین مسئله

$$\text{minimizing} = \sum_{i=1}^{n} \sum_{j=1}^{n} T_{i,j} * X_{i,j} \tag{9}$$

s.t :

$$\sum_{i=1}^{n} X_{i,j} = 1 \qquad \forall j = 1...n \tag{10}$$

$$\sum_{j=1}^{n} X_{i,j} = 1 \qquad \forall i = 1...n \tag{11}$$

$$T_{ij} = \begin{cases} \begin{cases} \sum_{i=1}^{j} Pt_i - d_i, \Leftarrow \sum_{i=1}^{j} Pt_i - d_i > 0 & \forall i = 1...j \\ \Leftarrow \text{if } Pt_i \subseteq m_i, & \\ 0 & \text{otherwise} \end{cases} \\ \begin{cases} Pt_i + \sum_{i=1}^{j-1} Pt_i - d_i, \Leftarrow Pt_i & \\ + \sum_{i=1}^{j-1} Pt_i - d_i > 0 & \forall i = 1...j \\ \Leftarrow \text{if } Pt_i \not\subset m_i, & \\ 0 & \text{otherwise} \end{cases} \end{cases} \tag{12}$$

$$X_{i,j} \in \{0, 1\} \tag{13}$$

رابطه‌ی ۹ مجموع دیرکرد کل کارها را کمینه می‌کند. محدودیت ۱۰ بیان می‌کند که هر موقعیت مختص به انجام یک کار است و دو کار در یک موقعیت انجام نمی‌شود (موقعیت به معنی نوبت پردازش کار است. یعنی اگر کار iام در jامین نوبت انجام شود مقدار ۱ را خواهد گرفت). محدویت ۱۱ به این معنی است که هر کار تنها در یک موقعیت می‌تواند انجام شود. محدودیت ۱۲ شالوده‌ی حل این مدل است و در اصل تنها تغییر محسوس جالب این مدل با مدل الگوریتم تخصیص است و در واقع پارامتری کارا به‌دست می‌دهد. این پارامتر مهم حداقل دیرکرد کار i را اگر در jامین نوبت پردازش شود، می‌دهد.

در پیوست مقاله نشان داده خواهد شد که مدل حد پایین را می‌دهد.

۳. الگوریتم شبیه‌سازی تبرید

ایده‌ی اصلی الگوریتم شبیه‌سازی تبریدشده از متروپلیس (۱۹۵۳) گرفته شده است.[۱۳] او در این الگوریتم ماده را به‌عنوان سیستمی متشکل از اجزای مختلف شبیه‌سازی کرده است. این الگوریتم روال سردشدن مواد را با کاهش تدریجی دما تا رسیدن به یک نقطه‌ی تعادل دمایی شبیه‌سازی می‌کند. ۳۰ سال بعد کیریک پاتریک و همکارانش روشی پیشنهاد کردند که براساس آن می‌توان این الگوریتم را برای مسائل بهینه‌سازی به‌کار برد.[۱۴]

در این روند جست‌وجو الگوریتم SA با یک T_0 (دمای اولیه) و S_0 (جواب اولیه) شروع شده که ابتدا اولین جواب (S_c) و بهترین جواب (S_B) است. یک جواب همسایگی (S_N) با تعویض بعضی عناصر (S_c) به‌دست می‌آید. تابع هدف به‌ازای S_N و S_c با E_N و E_c نشان داده می‌شود. اگر E_N از E_c بهتر باشد، S_N به‌عنوان S_c جدید پذیرفته می‌شود و در تکرار بعد S_N جدید تولید می‌شود. ولی چنانچه E_N از E_c بدتر باشد، از معیار متروپلیس استفاده می‌شود و S_N با احتمال $\exp^{(\frac{-\Delta E}{T})}$ به‌عنوان S_c جدید پذیرفته می‌شود. که در آن $\Delta E = E_N - E_c$ است. چنانچه E_c از E_B بهتر باشد، S_c به‌عنوان S_B جدید پذیرفته می‌شود؛ در غیر این‌صورت S_B بدون تغییر باقی می‌ماند. این فرایند در هر سطح دما به تعداد مشخصی تکرار می‌شود و پارامتر T به آرامی و با استفاده از تابع کاهش دما تا هنگامی که شرط توقف برقرار شود، کاهش می‌یابد. از علامت منفی برای مسائل کمینه‌سازی و از علامت مثبت در معیار متروپلیس برای مسائل بیشینه‌سازی استفاده می‌شود.[۱۵]

۱.۳. اجزا و پارامترهای الگوریتم SA

۱.۱.۳. نمایش ساختار جواب

ساختار جواب (توالی کارها) برای مثال برای پردازش ۶ کار روی ۳ ماشین به‌صورت زیر است:

۱	۳	۸	۲	۵	۷	۴	۶

در این ساختار اعداد بزرگ‌تر از تعداد کارها برای جداکردن کارها و اختصاص آنها به ماشین مورد نظر به‌کار می‌روند و موقعیت قرارگیری اعداد کوچک‌تر و مساوی با تعداد کارها در این جایگشت نشان‌دهنده‌ی نوبت پردازش آنها روی ماشین مربوطه است. مثلاً برای ساختار جواب نشان‌داده‌شده کار ۶ و ۴ روی ماشین اول، کار ۵ و ۲ روی ماشین دوم، و کار ۳ و ۱ روی ماشین سوم پردازش می‌شوند.

۲.۱.۳. انتخاب جواب اولیه

با توجه به مطالب بیان‌شده در نمایش ساختار جواب، جواب اولیه به‌صورت جایگشت تصادفی از $n + m - ۱$ عدد به‌دست می‌آید و برای این جایگشت به‌دست‌آمده مقدار تابع هدف محاسبه می‌شود و به‌عنوان جواب اولیه در نظر گرفته می‌شود.

۳.۱.۳. سازوکار جواب همسایه

می‌توان از دو روش برای ایجاد همسایگی استفاده کرد. انتخاب جواب همسایگی از بین جواب‌های امکان‌پذیر به‌صورت تصادفی یا برطبق ضابطه‌یی خاص براساس روش ابتکاری مخصوص به هر مسئله صورت می‌پذیرد.[۱۶]

در این مقاله برای ایجاد جواب همسایگی از عملگرهای تقاطع[۱] تک نقطه‌یی و جهش[۲] الگوریتم ژنتیک (GA) استفاده می‌شود که در هر بار این کار با استفاده از عملگر جهش یا عملگر تقاطع به‌صورت تصادفی انجام می‌شود.

- **عملگر جهش:** عملگر جهش خود شامل سه عملیات تعویض، معکوس‌سازی، و میان‌گذاری است که در این مقاله از روش میان گذاری آن استفاده می‌شود.
- **روش میان‌گذاری:** در این روش دوعدد تصادفی بین یک و طول کروموزوم منهای یک والد تولید می‌شود، که عدد اولی نشان‌دهنده‌ی ژن انتخابی و عدد دوم نشان‌دهنده‌ی موقعیتی است که آن ژن در آن موقعیت قرار داده می‌شود.
- **عملگر تقاطع:** در عملگر تقاطع به دو کروموزوم والد نیاز است و چون الگوریتم SA همیشه با یک کروموزوم کار می‌کند، کروموزوم والد دوم از راه جایگشت تصادفی $n + m - ۱$ همانند تولید جواب اولیه استفاده می‌شود و بعد عملگر تقاطع تک نقطه‌یی روی کروموزوم‌های والد انجام می‌گیرد و دو کروموزوم جدید به‌دست می‌آید. مقدار هر کروموزوم جدید محاسبه می‌شود و کروموزومی با جواب کمتر به‌عنوان جواب بهتر به مقایسات بعدی فرستاده می‌شود.

۴.۱.۳. انتخاب دمای اولیه و کاهش دما

چنانچه دمای اولیه پایین باشد، احتمال پذیرش جواب‌های بدترکاهش می‌یابد و ممکن است سیستم در جواب بهینه‌ی محلی باقی بماند. اگر بخواهیم جواب نهایی مستقل از جواب شروع باشد، دمای اولیه به اندازه‌ی کافی زیاد در نظر گرفته می‌شود تا امکان تعویض تقریباً آزادجواب‌های همسایگی وجود داشته باشد. در غیر این‌صورت جواب نهایی به سمت جواب شروع نزدیک خواهد شد. ضابطه‌ی کاهش دما و حرکت به سمت سردشدن سیستم معمولاً به‌صورت تابعی به شکل زیر در نظر گرفته می‌شود:

$$T_{r+۱} = \alpha * T_r \qquad ۰/۵ \leq \alpha \leq ۰/۹۹ \qquad r = ۱, ۲, ..., n$$

چنانچه α مقدار بزرگی داشته باشد سیستم دیرتر سرد می‌شود و تعداد نقاط بیشتری از فضای جواب بررسی می‌شوند که خود سبب افزایش زمان حل مسئله خواهد شد.[۱۷]

۵.۱.۳. سازوکار پذیرش جواب‌های نامزدشده

فرض کنید جواب جاری منجر به تابع هدفی معادل E_c شده است و نیز جواب همسایگی ایجادشده مقدار تابع هدف E_N را داشته باشد در حالت کمینه‌سازی اگر:

۱. $\Delta E = E_N - E_c \leq ۰$، جواب همسایگی جایگزین جواب جاری می‌شود و چنانچه $E_N \prec E_B$ و $S_B = S_N$ قرار داده می‌شود، در غیر این‌صورت S_B و E_B بدون تغییر باقی می‌مانند.

۲. $\Delta E = E_N - E_c \succ ۰$، مقدار $y = \exp(\frac{-\Delta E}{T})$ با یک عدد تصادفی بین صفر و یک مقایسه می‌شود. اگر از عدد تصادفی بزرگ‌تر باشد، جواب همسایگی به‌عنوان S_c پذیرفته می‌شود؛ در غیر این‌صورت جواب همسایگی دیگری تولید می‌شود.

۶.۱.۳. معیارهای توقف الگوریتم SA

با توجه به اینکه الگوریتم‌های فراابتکاری هیچ‌گونه شناختی نسبت به نقطه‌ی بهینه کلی و به‌طورکلی درجه‌ی بهینه‌بودن جواب‌ها ندارند، معیاری برای توقف نیاز است. به‌طورکلی می‌توان در حالت ایستا شرط توقف را چنین بیان کرد: رسیدن به یک دمای نهایی از پیش تعیین‌شده، T_f که کسر بسیار کوچکی از دمای اولیه است ($T_f = \alpha * T_۰, \alpha \leq ۰/۰۱$) یا رسیدن به یک مقدار بیشینه‌ی از پیش‌تعیین‌شده برای کل تکرارها.[۱۸]

روش SA ابزاری فراهم می‌کند تا بتوان از نقاط بهینه‌ی محلی فرارکرد و به‌وسیله پذیرش جواب‌های بدتر با یک احتمال مشخص به نقطه‌ی بهینه‌ی کلی دست یافت. این موضع یکی از نقاط قوت رویکرد SA است.[۱۷] گام‌های لازم برای شبیه‌سازی SA در این مقاله به‌صورت زیر است:

گام ۱. پارامترهای ورودی

$$(T_۰, \alpha, \text{maxiteration}, \text{maxiter}\,۲, \text{set}\ T = T_۰)$$

گام ۲. انتخاب جواب اولیه و $S_۰$ محاسبه‌ی مقدار تابع هدف به ازای جواب اولیه‌ی $E_۰$.

گام ۳. جواب اولیه به‌عنوان بهترین جواب درنظرگرفته‌شود ($E_c = E_۰, E_B = E_۰$, $S_c = S_۰, S_B = S_۰$).

گام ۴. یک جواب همسایگی S_N با استفاده از عملگر تقاطع یا جهش به‌نحوی که متعاقباً ذکر می‌شود تولید شود. عدد تصادفی بین صفر و ۱ تولید شود. چنانچه این عدد بزرگتر از ۰/۵ باشد از عملگر تقاطع و چنانچه این عدد کوچکتر از ۰/۵ باشد از عملگر جهش و روش میان‌گذاری استفاده شود. مقدار تابع هدف E_N محاسبه شود.

گام ۵. مقدار $\Delta E = E_N - E_c$ محاسبه شود.

گام ۶. اگر $\Delta E \leq ۰$، جواب همسایگی کنونی به‌عنوان جواب کنونی پذیرفته شود ($S_c = S_N,\ E_c = E_N$) و الگوریتم به گام ۹ منتقل شود.

گام ۷. اگر $\Delta E \succ ۰$، یک عدد تصادفی در بازه‌ی [۰, ۱] تولید شود.

گام ۸. اگر $\exp(\frac{-\Delta E}{T}) > \text{rand}[۰, ۱]$، جواب همسایگی به‌عنوان جواب کنونی پذیرفته شود ($S_c = S_N, E_c = E_N$) و الگوریتم به گام ۹ منتقل شود؛ در

غیر این‌صورت E_c و S_c بدون تغییر باقی بمانند و الگوریتم به گام ۱۰ منتقل شود.

گام ۹. اگر $E_c < E_B$، $E_B = E_c$ و $S_B = S_c$ قرار داده شوند؛ در غیر این‌صورت E_B و S_B بدون تغییر باقی بمانند.

گام ۱۰. اگر تعدا تکرارها در این دما کمتر از ۲maxiter باشد، الگوریتم به گام ۴ برگردد؛ در غیر این‌صورت دما با استفاده از رابطه‌ی:

$$T_{r+۱} = \alpha * T_r \qquad ۰/۵ \leq \alpha \leq ۰/۹۹ \qquad r = ۱, ۲, ..., n$$

محاسبه شود و گام بعدی اجرا شود.

گام ۱۱. بررسی شرایط توقف: اگر تعداد تکرارهای کل کمتر از maxiteration باشد الگوریتم به گام ۴ برود؛ در غیر این‌صورت E_B و S_B به‌عنوان جواب بهینه گزارش و الگوریتم توقف شود.[۱۵] در این مقاله از تنظیم پارامتر الگوریتم SA برای مسائل کارگاه باز عمومی که توسط نادری و همکاران صورت گرفته است، استفاده می‌شود.[۱۹]

۴. نتایج محاسباتی

برای بررسی و مقایسه مدل‌های ارائه‌شده و الگوریتم SA از مکانیسم تولید عدد صحیح تصادفی برای پارامترهای مسئله به شرح زیر استفاده می‌شود. زمان پردازش p_i برابر عدد صحیح تصادفی بین بازه‌ی $[۱, ۹۹]$ و موعد تحویل کارها در بازه‌ی $[P * (۱ - T - R/۲), P * (۱ - T + R/۲)]$ انتخاب می‌شود که P برابر مجموع زمان پردازش کارها تقسیم بر تعداد ماشین‌ها $(P = \sum_{i=۱}^{n} \frac{P_i}{m})$ و T, R نیز به‌ترتیب مربوط به بازه‌ی $\{۰/۲, ۰/۶, ۱\}$ و $\{۰/۴, ۰/۶, ۰/۸\}$ هستند[۲۰] که در این مسئله $T = ۰/۴$ و $R = ۰/۶$ در نظر گرفته شده‌اند.

در این مقاله برای مدل اصلی از نرم‌افزار GAMS(ver.۲۴.۱) و برای مدل پیشنهادی و الگوریتم SA از نرم افزار MATLAB ۲۰۱۲a روی کامپیوتر با مشخصات (Intel® CPU ۲/۶ GHZ and ۴ GB of RAM memory) استفاده شده است.

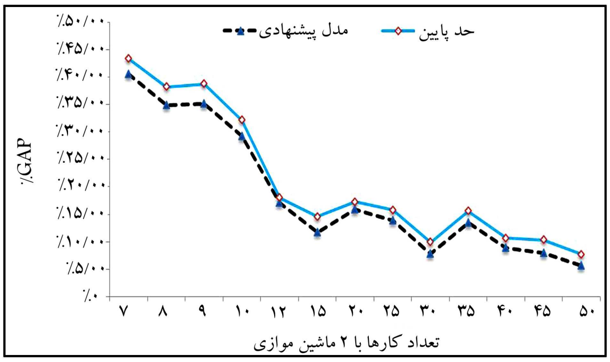
در ادامه نتایج حاصل از اجرای الگوریتم SA، که برای هر مسئله پنج بار اجرا شده است، با نتایج مدل برنامه‌ریزی عدد صحیح مختلط و مدل پیشنهادی که یک حد پایین برای مسئله است، مقایسه می‌شود. برای سنجش کارایی جواب مدل پیشنهادی به‌عنوان حد پایین و مقایسه با حد پایین موجود در تحقیقات پیشین و نیز الگوریتم SA معیار GAP که در رابطه‌ی زیر معرفی می‌شود، به‌کار رفته است.[۲۱]

$$\%\,\mathrm{GAP} = \frac{-(T_i - T^*)}{T^*} \times ۱۰۰ \qquad (۱۴)$$

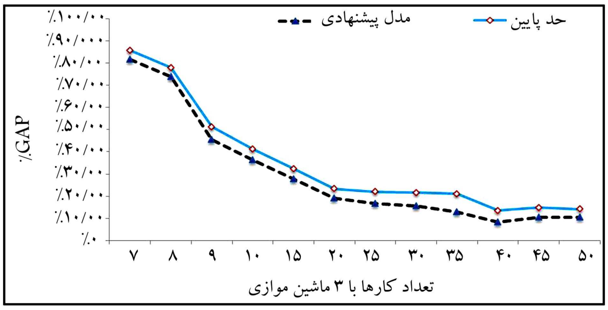
در رابطه‌ی ۱۴، T^* بیانگر مقدار بهینه در مسائل با اندازه‌ی کوچک یا جواب الگوریتم SA در مسائل با اندازه‌های متوسط و بزرگ است. مقدار T_i برای هر مسئله برابر مقدار به‌دست‌آمده برای هر یک از موارد مورد مقایسه در نظر گرفته می‌شود.

جدول ۱ نشان می‌دهد که جواب به‌دست‌آمده از مدل پیشنهادی همواره کوچک‌تر از جواب بهینه‌ی به‌دست‌آمده از مدل برنامه‌ریزی عدد صحیح مختلط است. به عبارت دیگر، می‌توان نتیجه گرفت جواب مدل پیشنهادی حد پایین مسئله‌ی مربوطه است. همچنین با توجه به نتایج حد پایین موجود در تحقیقات پیشین و مقایسه‌ی آن با جواب مدل پیشنهادی می‌توان دریافت که جواب مدل پیشنهادی اختلاف کمتری با جواب بهینه دارد. بنابراین، حد پایین به‌دست‌آمده از مدل پیشنهادی مهم‌تر از حد پایین موجود در پیشینه‌ی تحقیق است. معیار دیگری که اهمیت دارد زمان حل مسائل است که مدل برنامه‌ریزی عدد صحیح مختلط حتی برای مسائل با اندازه‌ی کوچک نیاز به زمان خیلی زیادی دارد در حالی‌که مدل پیشنهادی در زمان خیلی کمتری می‌تواند مسائل با اندازه‌ی بزرگ را هم حل کند. همچنین نتایج مربوط به مدل برنامه‌ریزی عدد صحیح مختلط و الگوریتم SA نشان می‌دهند که الگوریتم شبیه‌سازی تبرید جواب خیلی خوب را ارائه می‌دهد به‌گونه‌یی که در تمام مسائل مورد آزمایش با اندازه‌ی کوچک و متوسط به جواب بهینه دست می‌یابد و این امر بر کارایی خوب الگوریتم شبیه‌سازی تبرید دلالت دارد.

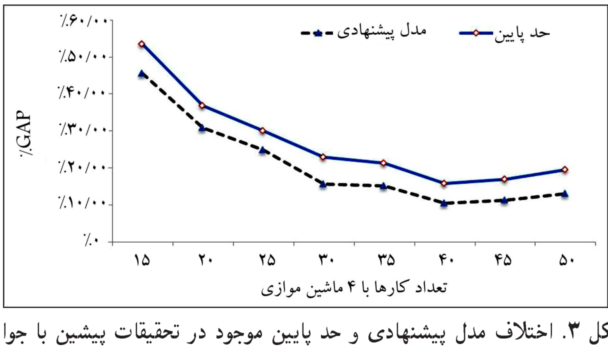
جدول ۲ و شکل‌های ۱ تا ۳ نتایج مربوط به مسائل مورد آزمایش در مدل برنامه‌ریزی عدد صحیح مختلط، مدل پیشنهادی، و حد پایین موجود در تحقیقات



شکل ۱. اختلاف مدل پیشنهادی و حد پایین موجود در تحقیقات پیشین با جواب بهینه‌ی مسئله‌ی ماشین‌های موازی با ۲ ماشین.



شکل ۲. اختلاف مدل پیشنهادی و حد پایین موجود در تحقیقات پیشین با جواب بهینه‌ی مسئله‌ی ماشین‌های موازی با ۳ ماشین.



شکل ۳. اختلاف مدل پیشنهادی و حد پایین موجود در تحقیقات پیشین با جواب بهینه‌ی مسئله‌ی ماشین‌های موازی با ۴ ماشین.

جدول ۱. نتایج مربوط به مدل برنامه‌ریزی ریاضی، مدل پیشنهادی، حد پایین موجود در تحقیقات پیشین و الگوریتم SA برای مسائل با اندازه‌ی کوچک.

تعداد		مدل برنامه‌ریزی عدد صحیح مختلط		مدل پیشنهادی			حد پایین مرور ادبیات		متوسط جواب الگوریتم شبیه‌سازی تبرید با ۵ اجرا	
ماشین‌ها	کارها	دیرکرد کل	زمان پردازش (ثانیه)	دیرکرد کل	زمان پردازش (ثانیه)	GAP (٪)	حد پایین دیرکرد کل	GAP (٪)	دیرکرد کل	GAP (٪)
۲	۵	۱۵۷	۰/۱۶	۹۶	۰/۸۲	۳۸/۸۵	۹۴/۵	۳۹/۸۱	۱۵۷	۰/۰۰
	۶	۶۲	۰/۱۷	۲۵	۰/۸۷	۵۹/۶۸	۲۳	۶۲/۹	۶۲	۰/۰۰
	۷	۱۵۸	۱/۰۸	۹۴	۰/۸۸	۴۰/۵۱	۸۹/۵	۴۳/۳۵	۱۵۸	۰/۰۰
	۸	۱۷۸	۱/۷	۱۱۶	۰/۹۲	۳۴/۸۳	۱۱۰	۳۸/۲	۱۷۸	۰/۰۰
	۹	۲۰۵	۹/۴۸	۱۳۳	۰/۹۴	۳۵/۱۲	۱۲۵/۵	۳۸/۷۸	۲۰۵	۰/۰۰
	۱۰	۲۱۶	۵۱/۰۱	۱۵۳	۰/۹۷	۲۹/۱۷	۱۴۶/۵	۳۲/۱۸	۲۱۶	۰/۰۰
	۱۲	۱۱۶۶	۱۰۳۷۵	۹۶۷	۱/۰۱	۱۷/۰۷	۹۵۶	۱۸/۰۱	۱۱۶۶	۰/۰۰
۳	۵	۷۳	۰/۱۴	۱۰	۰/۹۱	۸۶/۳	۷	۹۰/۴۱	۷۳	۰/۰۰
	۶	۶۶	۰/۳	۱۶	۰/۹۲	۷۵/۷۶	۱۴	۷۸/۷۹	۶۶	۰/۰۰
	۷	۵۴	۳/۱۵	۱۰	۰/۹۴	۸۱/۴۸	۷/۷	۸۵/۷۴	۵۴	۰/۰۰
	۸	۹۲	۶/۱	۲۴	۰/۹۷	۷۳/۹۱	۲۰/۳	۷۷/۹۳	۹۲	۰/۰۰
	۹	۲۱۱	۳۷۱/۶۶	۱۱۵	۰/۹۸	۴۵/۵۰	۱۰۳	۵۱/۱۸	۲۱۱	۰/۰۰
	۱۰	۳۵۲	۲۴۸۵	۲۲۴	۰/۹۹	۳۶/۳۶	۲۰۷	۴۱/۱۹	۳۵۲	۰/۰۰

جدول ۲. نتایج مربوط به مدل پیشنهادی، حد پایین موجود در تحقیقات پیشین و الگوریتم SA برای مسائل با اندازه‌ی متوسط و بزرگ.

تعداد		مدل پیشنهادی		حد پایین موجود در تحقیقات پیشین		متوسط جواب الگوریتم شبیه‌سازی تبرید با ۵ اجرا
ماشین‌ها	کارها	دیرکرد کل	GAP (٪)	حد پایین	GAP (٪)	دیرکرد کل
۲	۱۵	۱۱۴۴	۱۱/۷۳	۱۱۰۷	۱۴/۵۸	۱۲۹۶
	۲۰	۱۲۷۶	۱۵/۸۹	۱۲۵۵/۵	۱۷/۲۴	۱۵۱۷
	۲۵	۲۳۷۸	۱۳/۷۹	۲۳۲۴	۱۵/۷۵	۲۷۵۸/۴
	۳۰	۴۲۶۵	۷/۸	۴۱۶۶/۵	۹/۹۳	۴۶۲۶
	۳۵	۴۴۱۳	۱۳/۵۲	۴۳۰۹/۵	۱۵/۵۵	۵۱۰۲/۸
	۴۰	۸۲۲۹	۸/۸۹	۸۰۶۷	۱۰/۶۹	۹۰۳۲/۲
	۴۵	۹۸۹۰	۷/۹۸	۹۶۳۹/۵	۱۰/۳۱	۱۰۷۴۷/۴
	۵۰	۱۳۱۳۱	۵/۵۹	۱۲۸۳۴	۷/۷۲	۱۳۹۰۸/۴
۳	۱۵	۴۷۵	۲۷/۷	۴۴۴/۳	۳۲/۳۷	۶۵۷
	۲۰	۱۰۶۴	۱۹/۰۹	۱۰۰۷/۷	۲۳/۳۷	۱۳۱۵
	۲۵	۱۳۴۸	۱۶/۵۸	۱۲۶۰/۳	۲۲/۰۱	۱۶۱۶
	۳۰	۱۷۱۶	۱۵/۵۱	۱۵۹۴	۲۱/۵۲	۲۰۳۱
	۳۵	۲۰۶۴	۱۲/۹۸	۱۸۷۲/۳	۲۱/۰۷	۲۳۷۲
	۴۰	۵۲۴۶	۸/۲۷	۴۹۴۴	۱۳/۵۵	۵۷۱۹
	۴۵	۶۶۰۸	۱۰/۵۸	۶۲۹۰/۷	۱۴/۸۸	۷۳۹۰
	۵۰	۸۷۲۶	۱۰/۴۸	۸۳۶۷/۳	۱۴/۱۶	۹۷۴۸
۴	۱۵	۲۰۶	۴۵/۵	۱۷۶	۵۳/۴۴	۳۷۸
	۲۰	۶۶۴	۳۰/۸	۶۰۵/۵	۳۶/۹۰	۹۵۹/۶
	۲۵	۱۱۱۸	۲۴/۸۲	۱۰۴۰/۵	۳۰/۰۳	۱۴۸۷
	۳۰	۲۱۱۰	۱۵/۵۹	۱۹۲۷/۸	۲۲/۸۸	۲۴۹۹/۸
	۳۵	۲۳۷۵	۱۵/۲	۲۲۰۳/۳	۲۱/۳۳	۲۸۰۰/۸
	۴۰	۴۴۸۶	۱۰/۵	۴۲۱۷	۱۵/۸۷	۵۰۱۲/۲
	۴۵	۴۷۴۲	۱۱/۲۴	۴۴۳۲/۸	۱۷/۰۲	۵۳۴۲/۲
	۵۰	۵۴۳۶	۱۳/۰۲	۵۰۲۴/۳	۱۹/۶۱	۶۲۴۹/۸

پیشین، و الگوریتم اصلاح‌شده‌ی SA را با پارامترهای زیر نشان می‌دهد.

$$T_0 = 10, \quad \alpha = 0.98, \quad \text{maxiteration} = 600, \quad \text{maxiter2} = 20$$

۵. نتیجه‌گیری و پیشنهادات

۱. حد پایینی که توسط مدل ارائه می‌شود در مسائل با اندازه‌های بزرگ اختلاف زیادی با جواب الگوریتم شبیه‌سازی تبریدشده ندارد. بنابراین، این حد پایین کیفیت خیلی خوبی در مسائل با اندازه‌ی بزرگ دارد. از سوی دیگر، با توجه به اینکه این حد پایین از حد پایین موجود در تحقیقات پیشین بهتر است، از اهمیت ویژه‌یی برخوردار است.

۲. نوع نگاه به مسئله‌ی زمان‌بندی ماشین‌های موازی و ارائه‌ی مدل ریاضی عدد صحیح مختلط جدید برای آن و اضافه‌شدن یک مدل برنامه‌ریزی ریاضی کارا به پیشینه‌ی این موضوع اهمیت خاصی دارد.

۳. زمان حل مهم‌ترین عامل در مسائل زمان‌بندی است و این نوع مسائل در اندازه‌ی متوسط هم نیاز به زمان حل خیلی زیاد دارند. مدل پیشنهادی در اندازه‌های متوسط و حتی کوچک هم نسبت به مدل برنامه‌ریزی عدد صحیح مختلط به زمان حل کمتری نیاز دارد. با استفاده از مدل پیشنهادی و الگوریتم شبیه‌سازی معرفی‌شده می‌توان در زمان کمتری به جواب‌های خوبی دست یافت. در این مقاله حالت عمومی مسئله در نظر گرفته شده است می‌توان برای تحقیقات آتی فرض‌های دیگر مسئله مثل زمان قطع مجازکارها یا حتی خرابی ماشین و غیره را بررسی کرد.

پانوشت‌ها

1. crossover
2. mutation

منابع (References)

1. Pinedo, M.L., *Scheduling: Theory, Algorithms, and Systems*, Springer Science & Business Media (2012).
2. Graham, R.L., Lawler, E.L., Lenstra, J.K., and Rinnooy Kan, A.H.G. "Optimization and approximation in deterministic sequencing and scheduling: A survey", *Annals of Discrete Mathematics*, **5**, pp. 287-326 (1979).
3. Bilge, Ü., Kıraç, F., Kurtulan, M. and Pekgün, P. "A tabu search algorithm for parallel machine total tardiness problem", *Computers & Operations Research*, **31**(3), pp. 397-414 (2004).
4. Anghinolfi, D. and Paolucci, M. "Parallel machine total tardiness scheduling with a new hybrid metaheuristic approach", *Computers & Operations Research*, **34**(11), pp. 3471-3490 (2007).
5. Azizoglu, M. and Kirca, O. "Tardiness minimization on parallel machines", *International Journal of Production Economics,* **55**(2), pp. 163-168 (1998).
6. Tanaka, S. and Araki, M. "A branch-and-bound algorithm with Lagrangian relaxation to minimize total tardiness on identical parallel machines", *International Journal of Production Economics,* **113**(1), pp. 446-458 (2008).
7. Shim, S.-O. and Kim, Y.-D. "Scheduling on parallel identical machines to minimize total tardiness", *European Journal of Operational Research,* **177**(1), pp. 135-146 (2007).
8. Mensendiek, A., Gupta, J.N. and Herrmann, J. "Scheduling identical parallel machines with fixed delivery dates to minimize total tardiness", *European Journal of Operational Research*, **243**(2), pp. 514-522 (2015).
9. Kovalyov, M.Y. and Werner, F. "Approximation schemes for scheduling jobs with common due date on parallel machines to minimize total tardiness", *Journal of Heuristics,* **8**(4), pp. 415-428 (2002).
10. Elmaghraby, S.E. and Park, S.H. "Scheduling jobs on a number of identical machines", *AIIE transactions,* **6**(1), pp. 1-13 (1974).
11. Arkin, E.M. and Roundy, R.O. "Weighted-tardiness scheduling on parallel machines with proportional weights", *Operations Research,* **39**(1), pp. 64-81 (1991).
12. Unlu, Y. and Mason, S.J. "Evaluation of mixed integer programming formulations for non-preemptive parallel machine scheduling problems", *Computers & Industrial Engineering*, **58**(4), pp. 785-800 (2010).
13. Metropolis, N., Rosenbluth, A.W., Rosenbluth, M.N. and Teller A.H. "Equation of state calculations by fast computing machines", *The Journal of Chemical Physics*, **21**(6), pp. 1087-1092 (1953).
14. Kirkpatrick, S., Gelatt, C.D. and Vecchi, M.P. "Optimization by simmulated annealing", *Science*, **220**(4598), pp. 671-680 (1983).
15. Jeong, S.-J., Kim, K.-S. and Lee, Y.-H. "The efficient search method of simulated annealing using fuzzy logic controller", *Expert Systems with Applications*, **36**(3), pp. 7099-7103 (2009).
16. Özcan, U. "Balancing stochastic two-sided assembly lines: A chance-constrained, piecewise-linear, mixed integer program and a simulated annealing algorithm", *European Journal of Operational Research,* **205**(1), pp. 81-97 (2010).
17. Vidal, R.V.V., *Applied Simulated Annealing*, First ed. Springer (1993).
18. Vasan, A. and Raju, K.S. "Comparative analysis of simulated annealing, simulated quenching and genetic algorithms for optimal reservoir operation", *Applied Soft Computing*, **9**(1), pp. 274-281 (2009).

19. Naderi, B., Fatemi Ghomi, S.M.T., Aminnayeri, M. and Zandieh, M. "Scheduling open shops with parallel machines to minimize total completion time", *Journal of Computational and Applied Mathematics*, **235**(5), pp. 1275-1287 (2011).

20. Loukil, T., Teghem, J. and Tuyttens, D. "Solving multi-objective production scheduling problems using metaheuristics", *European Journal of Operational Research*, **161**(1), pp. 42-61 (2005).

21. Karimi, N., Zandieh, M. and Najafi, A. "Group scheduling in flexible flow shops: a hybridised approach of imperialist competitive algorithm and electromagnetic-like mechanism", *International Journal of Production Research*, **49**(16), pp. 4965-4977 (2011).

پیوست

قضیه: مدل پیشنهادی جوابی حد پایین برای دیرکرد کل کارها را می‌دهد.

برای اثبات اینکه مدل پیشنهادی حد پایین دیرکرد کل را می‌دهد یک مسئله با ۳ کار در حالت‌های مختلف برای موعد تحویل در نظر گرفته می‌شود، نشان داده خواهد شد که مقداری که برای مسئله به‌دست خواهد آمد کمتر از مقدار بهینه‌ی توالی یا مساوی با آن خواهد بود.

حالت اول: فرض کنید یک مسئله با ۳ کار به طوری‌که $p_2 < p_1 < p_3$ و $d_3 < d_1 < d_2$ و $p_i < d_i$، داشته باشیم در این صورت مقدار دیرکرد کل با استفاده از مدل پیشنهادی به‌صورت زیر محاسبه می‌شود.

$$T_{ji} = \begin{bmatrix} 0 & p_2 + p_1 - d_1 & p_2 + p_1 + p_3 - d_1 \\ 0 & p_2 + p_1 - d_2 & p_2 + p_1 + p_3 - d_2 \\ 0 & p_2 + p_3 - d_3 & p_2 + p_1 + p_3 - d_3 \end{bmatrix}$$

برای حل این ماتریس به روش مجارستانی عمل می‌شود که برای این‌کار با توجه به ماتریس بالا باید در ستون ۲ و ۳ نیز صفر ایجاد شود که برای این کار باید کمینه‌ی هر ستون را یافت و از درایه‌های آن ستون کم کرد.

کمینه‌ی ستون ۲: $p_2 + p_1 - d_2$ زیرا:

$$p_2 + p_1 - d_2 < p_2 + p_1 - d_1 \Rightarrow -d_2 < -d_1 \Rightarrow d_2 > d_1$$
$$p_2 + p_1 - d_2 < p_2 + p_3 - d_3 \Rightarrow d_3 - d_2 < p_3 - p_1 \Rightarrow - < +$$

کمینه‌ی ستون سوم درایه دوم آن است؛ چون:

$$p_2 + p_1 + p_3 - d_2 < p_2 + p_1 + p_3 - d_1 \Rightarrow -d_2 < -d_1 \Rightarrow d_2 > d_1$$
$$p_2 + p_1 + p_3 - d_2 < p_2 + p_1 + p_3 - d_3 \Rightarrow -d_2 < -d_3 \Rightarrow d_2 > d_3$$

حال طبق روال الگوریتم مجارستانی:

$$T_{ji} = \begin{bmatrix} 0 & d_2 - d_1 & d_2 - d_1 \\ 0 & 0 & 0 \\ 0 & p_3 - d_3 - p_1 + d_2 & d_2 - d_3 \end{bmatrix}$$

در ماتریس بالا طبق خاصیت روش مجارستانی تا زمانی‌که کمینه‌ی تعداد خطوط پوشش‌دهنده‌ی صفرها برابر با تعداد سطر یا ستون نشود روش مجارستانی ادامه می‌یابد. که ماتریس بالا طی این روش به‌صورت زیر در می‌آید:

$$T_{ji} = \begin{bmatrix} 0 & 0 & 0 \\ d_2 - d_1 & 0 & 0 \\ 0 & p_3 - d_3 - p_1 + d_1 & d_1 - d_3 \end{bmatrix}$$

که این ماتریس دارای حداقل تعداد خطوط پوشش‌دهنده برابر سطر و ستون است پس دیرکرد کل طبق ماتریس بالا برابر مقدار زیر خواهد بود:

$$\text{Total Tardiness (Proposed Model)} = 2 * p_1 + 2 * p_2 + p_3 - (d_1 + d_2)$$

اکنون برای مقایسه‌ی مقدار تابع هدف مدل پیشنهادی با تمام توالی‌های ممکن مسئله نشان داده می‌شود که این مقدار کمتر از تمام مقادیر مربوط به هر یک از توالی‌ها یا مساوی با آنها خواهد بود. جدول پیوست ۱ نشان‌دهنده‌ی این توضیح است.

دیرکرد کل برای توالی ۳ − ۲ − ۱ به‌صورت زیر محاسبه می‌شود که سایر توالی‌ها نیز به همین منوال است.

$$T_3 = p_3 - d_3 = 0, \quad T_2 = p_3 + p_2 - d_2, \quad T_1 = p_3 + p_2 + p_1 - d_1$$
$$\text{Total Tardiness} = T_1 + T_2 + T_3 = 2 * p_3 + 2 * p_1 + p_2 - (d_1 + d_2)$$

حالت دوم: این‌بار فرض می‌شود که یک مسئله با ۳ کار به طوری‌که $p_2 < p_1 < p_3$ و $d_2 < d_1 < d_3$ و $p_i < d_i$، وجود دارد. در این صورت مقدار دیرکرد کل با استفاده از مدل پیشنهادی به‌صورت زیر محاسبه می‌شود.

$$T_{ji} = \begin{bmatrix} 0 & p_2 + p_1 - d_1 & p_2 + p_1 + p_3 - d_1 \\ 0 & p_2 + p_1 - d_2 & p_2 + p_1 + p_3 - d_2 \\ 0 & p_2 + p_3 - d_3 & p_2 + p_1 + p_3 - d_3 \end{bmatrix}$$

برای حل این ماتریس مانند حالت اول:

کمینه‌ی ستون ۲: $p_2 + p_3 - d_3$ زیرا:

$$p_2 + p_3 - d_3 < p_2 + p_1 - d_1 \Rightarrow p_3 - p_1 < d_3 - d_1 \Rightarrow \blacksquare$$
$$p_2 + p_3 - d_3 < p_2 + p_1 - d_2 \Rightarrow p_3 - p_1 < d_3 - d_2 \Rightarrow \blacksquare$$

کمینه‌ی ستون سوم درایه‌ی سوم آن است؛ چون:

$$p_2 + p_1 + p_3 - d_3 < p_2 + p_1 + p_3 - d_1 \Rightarrow -d_3 < -d_1 \Rightarrow d_3 > d_1$$
$$p_2 + p_1 + p_3 - d_3 < p_2 + p_1 + p_3 - d_2 \Rightarrow -d_3 < -d_2 \Rightarrow d_3 > d_2$$

اکنون طبق روال الگوریتم مجارستانی:

$$T_{ji} = \begin{bmatrix} 0 & p_1 - d_1 - p_3 + d_3 & d_3 - d_1 \\ 0 & p_1 - d_2 - p_3 + d_3 & d_3 - d_2 \\ 0 & 0 & 0 \end{bmatrix}$$

جدول پیوست ۱. حالت اول.

مدل ارائه شده برای دیرکرد کل	مقایسه	مدل پایه برای دیرکرد کل	توالی
$2p_1 + 2p_2 + p_3 - (d_1 + d_2)$	$>$	$2p_1 + 2p_2 + p_3 - (d_2 + d_3)$	$1 - 2 - 3$
$2p_1 + 2p_2 + p_3 - (d_1 + d_2)$	$>$	$2p_1 + 2p_3 + p_2 - (d_2 + d_3)$	$1 - 3 - 2$
$2p_1 + 2p_2 + p_3 - (d_1 + d_2)$	$>$	$2p_2 + 2p_1 + p_3 - (d_1 + d_3)$	$2 - 1 - 3$
$2p_1 + 2p_2 + p_3 - (d_1 + d_2)$	$>$	$2p_2 + 2p_3 + p_1 - (d_1 + d_3)$	$2 - 3 - 1$
$2p_1 + 2p_2 + p_3 - (d_1 + d_2)$	$>$	$2p_3 + 2p_1 + p_2 - (d_1 + d_2)$	$3 - 1 - 2$
$2p_1 + 2p_2 + p_3 - (d_1 + d_2)$	$>$	$2p_3 + 2p_2 + p_1 - (d_1 + d_2)$	$3 - 2 - 1$

جدول پیوست ۲. حالت دوم.

مدل ارائه شده برای دیرکرد کل	مقایسه	مدل پایه برای دیرکرد کل	توالی
$2p_1 + 2p_2 + p_3 - (d_1 + d_3)$	$>$	$2p_1 + 2p_2 + p_3 - (d_2 + d_3)$	$1 - 2 - 3$
$2p_1 + 2p_2 + p_3 - (d_1 + d_3)$	$>$	$2p_1 + 2p_3 + p_2 - (d_2 + d_3)$	$1 - 3 - 2$
$2p_1 + 2p_2 + p_3 - (d_1 + d_3)$	$>$	$2p_2 + 2p_1 + p_3 - (d_1 + d_3)$	$2 - 1 - 3$
$2p_1 + 2p_2 + p_3 - (d_1 + d_3)$	$>$	$2p_2 + 2p_3 + p_1 - (d_1 + d_3)$	$2 - 3 - 1$
$2p_1 + 2p_2 + p_3 - (d_1 + d_3)$	$>$	$2p_3 + 2p_1 + p_2 - (d_1 + d_2)$	$3 - 1 - 2$
$2p_1 + 2p_2 + p_3 - (d_1 + d_3)$	$>$	$2p_3 + 2p_2 + p_1 - (d_1 + d_2)$	$3 - 2 - 1$

ماتریس بالا طی روش مجارستانی به‌صورت زیر در می‌آید:

$$T_{ji} = \begin{bmatrix} 0 & 0 & p_3 - p_1 \\ 0 & d_1 - d_2 & p_3 - d_2 - p_1 + d_1 \\ p_1 - d_1 - p_3 + d_3 & 0 & 0 \end{bmatrix}$$

که این ماتریس دارای حداقل تعداد خطوط پوشش‌دهنده برابر سطر و ستون است. پس دیرکرد کل طبق ماتریس بالا برابر مقدار زیر است.

$$\text{Total Tardiness (Proposed Model)} = 2 * p_1 + 2 * p_2 + p_3 - (d_1 + d_2)$$

اکنون برای مقایسه‌ی مقدار تابع هدف مدل پیشنهادی با تمام توالی‌های ممکن مسئله نشان داده می‌شود که این مقدار کمتر از تمام مقادیر مربوط به هر یک از توالی‌ها یا مساوی با آنها خواهد بود. جدول پیوست ۲ نشان‌دهنده‌ی این توضیح است.

حالت سوم: این‌بار فرض می‌شود که یک مسئله با ۳ کار به‌طوری‌که $p_2 < p_1 < p_3$ و $d_i = d$ و $p_i < d_i$، وجود دارد. در این صورت مقدار دیرکرد کل با استفاده از مدل پیشنهادی به‌صورت زیر محاسبه می‌شود:

$$T_{ji} = \begin{bmatrix} 0 & p_2 + p_1 - d & p_2 + p_1 + p_3 - d \\ 0 & p_2 + p_1 - d & p_2 + p_1 + p_3 - d \\ 0 & p_2 + p_3 - d & p_2 + p_1 + p_3 - d \end{bmatrix}$$

برای حل این ماتریس مانند حالت اول:

کمینه‌ی ستون ۲: $p_2 + p_1 - d$. زیرا:

$$p_2 + p_1 - d < p_2 + p_3 - d \Rightarrow p_1 < p_3 \Rightarrow \blacksquare$$

همه درایه‌های ستون سوم برابرند. پس طبق روال الگوریتم مجارستانی:

$$T_{ji} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & p_3 - p_1 & 0 \end{bmatrix} \quad \text{or} \quad T_{ji} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & p_3 - p_1 & 0 \end{bmatrix}$$

$$\text{or} \quad T_{ji} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & p_3 - p_1 & 0 \end{bmatrix}$$

که این ماتریس دارای کمینه‌ی تعداد خطوط پوشش‌دهنده برابر سطر و ستون است و جواب بهینه‌ی چندگانه دارد. در همه‌ی حالت‌ها دیرکرد کل کارها برابر مقدار زیر است:

$$\text{Total Tardiness (Proposed Model)} = 2 * p_1 + 2 * p_2 + p_3 - 2 * d$$

اکنون برای مقایسه‌ی مقدار تابع هدف مدل پیشنهادی با تمام توالی‌های ممکن مسئله نشان داده می‌شود که این مقدار کمتر از تمام مقادیر مربوط به هر یک از توالی‌ها یا مساوی با آنها خواهد بود. جدول پیوست ۳ نشان‌دهنده‌ی این توضیح است.

حالت چهارم: این‌بار فرض می‌شود که یک مسئله با ۳ کار به‌طوری‌که $p_2 < p_1 < p_3$ و $d_1 < d_3 < d_2$ و $p_i < d_i$، وجود دارد. در این صورت مقدار دیرکرد کل با استفاده از مدل پیشنهادی به‌صورت زیر محاسبه می‌شود:

$$T_{ji} = \begin{bmatrix} 0 & p_2 + p_1 - d_1 & p_2 + p_1 + p_3 - d_1 \\ 0 & p_2 + p_1 - d_2 & p_2 + p_1 + p_3 - d_2 \\ 0 & p_2 + p_3 - d_3 & p_2 + p_1 + p_3 - d_3 \end{bmatrix}$$

برای حل این ماتریس مانند روش در حالت اول:

کمینه‌ی ستون ۲: $p_2 + p_1 - d_2$، زیرا:

$$p_2 + p_1 - d_2 < p_2 + p_1 - d_1 \Rightarrow -d_2 < -d_1 \Rightarrow d_2 > d_1 \Rightarrow \blacksquare$$

$$p_2 + p_1 - d_2 < p_2 + p_3 - d_3 \Rightarrow p_3 - p_1 > d_3 - d_2 \Rightarrow \blacksquare$$

جدول پیوست ۳. حالت سوم.

مدل ارائه شده برای دیرکرد کل	مقایسه	مدل پایه برای دیرکرد کل	توالی
$۲p_{۱} + ۲p_{۲} + p_{۳} - ۲ * d$	$=$	$۲p_{۱} + ۲p_{۲} + p_{۳} - ۲ * d$	۱ − ۲ − ۳
$۲p_{۱} + ۲p_{۲} + p_{۳} - ۲ * d$	$>$	$۲p_{۱} + ۲p_{۳} + p_{۲} - ۲ * d$	۱ − ۳ − ۲
$۲p_{۱} + ۲p_{۲} + p_{۳} - ۲ * d$	$>$	$۲p_{۲} + ۲p_{۱} + p_{۳} - ۲ * d$	۲ − ۱ − ۳
$۲p_{۱} + ۲p_{۲} + p_{۳} - ۲ * d$	$>$	$۲p_{۲} + ۲p_{۳} + p_{۱} - ۲ * d$	۲ − ۳ − ۱
$۲p_{۱} + ۲p_{۲} + p_{۳} - ۲ * d$	$>$	$۲p_{۳} + ۲p_{۱} + p_{۲} - ۲ * d$	۳ − ۱ − ۲
$۲p_{۱} + ۲p_{۲} + p_{۳} - ۲ * d$	$>$	$۲p_{۳} + ۲p_{۲} + p_{۱} - ۲ * d$	۳ − ۲ − ۱

جدول پیوست ۴. حالت چهارم.

مدل ارائه شده برای دیرکرد کل	مقایسه	مدل پایه برای دیرکرد کل	توالی
$۲p_{۱} + ۲p_{۲} + p_{۳} - (d_{۲} + d_{۳})$	$=$	$۲p_{۱} + ۲p_{۲} + p_{۳} - (d_{۲} + d_{۳})$	۱ − ۲ − ۳
$۲p_{۱} + ۲p_{۲} + p_{۳} - (d_{۲} + d_{۳})$	$>$	$۲p_{۱} + ۲p_{۳} + p_{۲} - (d_{۲} + d_{۳})$	۱ − ۳ − ۲
$۲p_{۱} + ۲p_{۲} + p_{۳} - (d_{۲} + d_{۳})$	$>$	$۲p_{۲} + ۲p_{۱} + p_{۳} - (d_{۱} + d_{۳})$	۲ − ۱ − ۳
$۲p_{۱} + ۲p_{۲} + p_{۳} - (d_{۲} + d_{۳})$	$>$	$۲p_{۲} + ۲p_{۳} + p_{۱} - (d_{۱} + d_{۳})$	۲ − ۳ − ۱
$۲p_{۱} + ۲p_{۲} + p_{۳} - (d_{۲} + d_{۳})$	$>$	$۲p_{۳} + ۲p_{۱} + p_{۲} - (d_{۱} + d_{۲})$	۳ − ۱ − ۲
$۲p_{۱} + ۲p_{۲} + p_{۳} - (d_{۲} + d_{۳})$	$>$	$۲p_{۳} + ۲p_{۲} + p_{۱} - (d_{۱} + d_{۲})$	۳ − ۲ − ۱

کمینه‌ی ستون سوم درایه‌ی دوم آن است؛ چون:

$$p_{۲} + p_{۱} + p_{۳} - d_{۲} < p_{۲} + p_{۱} + p_{۳} - d_{۱} \Rightarrow -d_{۲} < -d_{۱} \Rightarrow d_{۲} > d_{۱}$$

$$p_{۲} + p_{۱} + p_{۳} - d_{۲} < p_{۲} + p_{۱} + p_{۳} - d_{۳} \Rightarrow -d_{۲} < -d_{۳} \Rightarrow d_{۲} > d_{۳}$$

حال طبق روال الگوریتم مجارستانی:

$$T_{ji} = \begin{bmatrix} ۰ & d_{۲} - d_{۱} & d_{۲} - d_{۱} \\ ۰ & ۰ & ۰ \\ ۰ & p_{۳} - d_{۳} - p_{۱} + d_{۲} & d_{۲} - d_{۳} \end{bmatrix}$$

ماتریس بالا طی روش مجارستانی به‌صورت زیر در می‌آید:

$$T_{ji} = \begin{bmatrix} ۰ & d_{۳} - d_{۱} & d_{۳} - d_{۱} \\ d_{۲} - d_{۳} & ۰ & ۰ \\ ۰ & p_{۳} - p_{۱} & ۰ \end{bmatrix}$$

که این ماتریس دارای کمینه تعداد خطوط پوشش دهنده برابر سطر و ستون است. پس دیرکرد کل طبق ماتریس بالا برابر مقدار زیر خواهد بود:

$$\text{Total Tardiness (Proposed Model)} = ۲ * p_{۱} + ۲ * p_{۲} + p_{۳} - (d_{۲} + d_{۳})$$

حال برای مقایسه‌ی مقدار تابع هدف مدل پیشنهادی با تمام توالی‌های ممکن مسئله نشان داده می‌شود که این مقدار کمتر از تمام مقادیر مربوط به هر یک از توالی‌ها یا مساوی با آنها خواهد بود. جدول پیوست ۴ نشان‌دهنده‌ی این توضیح است.

با توجه به چهار حالت بیان‌شده که تمام حالت‌های مختلف مسائل را در نظر می‌گیرد، می‌توان نتیجه گرفت که مدل پیشنهادی همیشه حد پایین یا جواب بهینه‌ی مسئله را می‌دهد.