بکارگیری الگوریتم شبیه سازی تبرید در مسئله زمانبندی تک‌ماشین با موعد تحویل مشترک

محمد مهدی مبینی[1]
رضا توکلی مقدم[2]

چکیده:
در این مقاله مسئله زمانبندی تعداد مشخصی کار بر روی ماشین در حالتی که کارها دارای موعد تحویل مشترک هستند، مورد بررسی قرار گرفته است. در ابتدا شرحی بر مسئله ارائه شده است و بعد از آن مدل ریاضی مسئله ارائه شده است، سپس الگوریتم فراابتکاری مورد نظر برای حل این مسئله معرفی شده است و در انتها جواب های به دست آمده از طریق این الگوریتم با جوابهای بدست آمده از طریق حل مدل ارائه شده با نرم افزار Lingo 8 مقایسه شده.

کلمات کلیدی: زمان بندی، موعد تحویل مشترک، الگوریتم فرا ابتکاری، شبیه سازی تبرید

Using simulated annealing in single machine scheduling problem with common due date

M.M. Mobini
R. Tavakoli Moghadam

ABSTRACT
In this article, scheduling of predetermined jobs on machine respect to common due date is considered. Firstly, describe the problem is presented and after that mathematic model is described. Afterwards metaheuristic algorithm for this problem is described and finally achieved results are compared with results of Lingo8.

KEYWORDS
Scheduling, Common due date, Metaheuristic algorithm, Simulated annealing.

[1] گروه مهندسی صنایع، دانشکده فنی، دانشگاه تهران (MahdiMobini@ut.ac.ir)
[2] گروه مهندسی صنایع، دانشکده فنی، دانشگاه تهران (tavakoli@ut.acc.ir)

مقدمه

زمان بندی کارها در مقابل داشتن یک موعد تحویل کلی، برای اتمام کارها، بسیار مورد توجه قرار گرفته است. به علت فضای فشرده رقابتی امروز شرکت ها ناچار به تولید انواع گوناگونی از کالاها هستند، ضمن آنکه مشتری انتظار داردکه کالای او در موعد مقرر تحویل شود. در راستای تامین این خواسته مشتریان مفاهیم جدیدی همچون تولید ناب و تولید بهنگام بسط داده شدند و مورد توجه قرار گرفتند. به عنوان مثال فلسفه تولید بهنگام بیان می کند که: مقدار مشخصی از کالا باید دقیقا در زمان درست آن مورد پردازش قرار گیرند و دقیقا در زمان مشخصی تحویل داده شود. در فلسفه تولید بهنگام زمان تحویل کالا می تواند به صورت یک زمان تحویل کلی در نظر گرفته شود. به عنوان مثال این حالت موقعی اتفاق می افتد که چندین کالا برای مونتاژ نهایی بر روی یک قطعه مورد نیاز است. در این صورت تمام آن کالاها دارای یک زمان تحویل یکسان هستند که از آن به عنوان موعد تحویل مشترک نام برده می‌شود. ضمن آنکه در عمل موعد تحویل مشترک در موارد دیگری نیز کاربرد دارد به عنوان مثال فرض کنید که یک مشتری دسته‌ای از محصولات را سفارش داده است بطوریکه می‌خواهد تمام آن را در یک زمان تحویل بگیرد.

کانت در سال ۱۹۸۱ [۱] از اولین افرادی بود که مسئله موعد تحویل مشترک را مورد بررسی قرار داد. او مسئله را بصورت کمینه کردن کل انحراف از موعد تحویل مشترک در نظر گرفت و الگوریتمی برای حل آن آرائه داد. افراد دیگری نیز از قبیل بیسکاپ[۲]، هان و همکاران[۳]، چند و چنیبرگر[۴]، و غیره نیز این مسئله مورد توجه قرار دادند.

وقتی در یک مسئله موعد تحویل مشترک کالاها را زمانبندی می‌کنیم دو نوع هزینه متصور خواهد بود. اول برای کارهایی که قبل از زمان تحویل خاتمه می‌یابند،که از آن به عنوان جریمه زودکرد یاد می‌کنیم. دوم هزینه ناشی از کارهایی که بعد از زمان تحویل خاتمه یافته‌اند و در نتیجه بعد از موعد تحویل به دست مشتری خواهند رسید. از هزینه نوع دوم به عنوان جریمه دیرکرد یاد خواهیم کرد.

باید خاطر نشان کرد که بدست آوردن مقادیر دقیق این جریمه ها به خصوص در مورد جریمه دیرکرد بسیار مشکل خواهد بود زیرا این هزینه شامل هزینه هایی چون از دست دادن مشتری، از دست دادن حسن شهرت، عدم رضایت مندی مشتری و... خواهد بود. در مورد هزینه نوع اول (زودکرد) می توان هزینه نگهداری کالاهای آماده و هزینه سرمایه درگیر با این کالاها را نام برد. به هر حال در نظر گرفتن مقدار تقریبی این هزینه ها بسیار مهم خواهد بود.

در زمینه زمانبندی با در نظر گرفتن موعد تحویل مشترک دو نوع زمان تحویل رایج است که محدود کننده و غیر محدود کننده نامیده می شوند. موعد تحویل در صورتیکه در صدد یافتن مقدار بهینه آن باشیم و یا اینکه مقدار از پیش تعیین شده آن بیشتر یا مساوی مجموع زمان انجام فعالیتها باشد و تاثیری بر روی توالی کارها نداشته باشد، محدود کننده نخواهد بود. از سوی دیگر اگر موعد تحویل از پیش تعیین شده باشد و بر روی توالی بهینه تاثیر داشته باشد آنرا محدود کننده می‌نامیم.

در این مقاله زمان بندی بر روی یک ماشین در نظر گرفته شده است. می دانیم که زمانبندی یک ماشین حالت خاصی از زمانبندی در حالت چند ماشین است ولی آنچه در اینجا به عنوان دلیل در نظر گرفتن مسئله تک ماشین مورد توجه قرار می گیرد مواردی از قبیل موارد زیر است:

- آنالیز حالت تک ماشین می تواند به فهم بهتر حالت چند ماشین منجر شود.
- در بسیاری حالت های واقعی یک ماشین گلوگاه می تواند به عنوان یک ماشین مجزا مورد تحلیل قرار گیرد، به طوری که زمان بندی عملیات تولید باید حول این ماشین انجام شود و در صورتی می‌تواند زمانبندی مطلوب باشد که، زمانبندی ماشین گلوگاه در بهترین حالت ممکنه صورت گرفته باشد.

بنابراین در این مقاله فقط حالت تک ماشین در نظر گرفته شده است هر چند که ممکن است در نظر گرفتن حالت چند ماشین نیز مزیتهای بر حالت تک ماشین داشته باشد.

فرض کنیم که یک موعد تحویل محدود کننده وجود دارد و برای هر کار جریمه زودکرد و دیرکرد مشخص شده است. هدف پیدا کردن توالی عملیاتی است که دارای کمترین مجموع جریمه های زودکرد و دیرکرد باشد.

این یک مسئله NP-hard است. بنابراین الگوریتم های دقیق برای حل این مسئله دارای کارایی چندانی نخواهد بود مگر در مورد مسائل کوچک. به همین علت، محققین به ارائه الگوریتمهای ابتکاری و فراابتکاری برای این مسئله پرداخته اند. از کارهایی که در زمینه حل این مسئله با

استفاده از الگوریتم های فرا ابتکاری صورت گرفته است می توان به گلور و همکاران[۵] اشاره کرد که با استفاده از الگوریتم جستجوی ممنوع به حل این مسئله پرداخت. همچنین رابین و راگاتز الگوریتم ژنتیک را برای این مسئله را ارائه کردند.

ادامه مقاله حاضر شامل بخش های زیر خواهد بود: در بخش دوم نمادهای مورد استفاده برای بیان ریاضی مسئله و فرموله کردن آن ذکر خواهد شد. سپس مدل ریاضی مورد استفاده شرح داده خواهد شد. در بخش سوم به معرفی الگوریتم SA و اقدامات صورت گرفته در این مقاله برای بهبود عملکرد آن می پردازیم. در بخش چهارم مقایسه‌ای بین جوابهای بدست آمده از الگوریتم ارائه شده و جواب های نرم فزار Lingo 8 صورت خواهد گرفت و در انتها نتیجه گیری ارائه خواهد شد.

معرفی مسئله

نمادها :

d : زمان تحویل مشترک

n : تعداد کارها

p_i : زمان عملیات کار i ام

α_i : جریمه زود کر کار i ام

β_i : جریمه دیر کرد کار i ام

C_i : زمان تکمیل کار i ام

E_i : زودکرد کار i ام

L_i : دیر کرد کار i ام

X_i : زمان فعالیت i ام

فرض شده است که همه کارها در زمان صفر آماده هستند. همه کارها باید دقیقا برروی یک ماشین عملیات شوند و هر کار دقیقا یکبار مورد عملیات قرار می گیرد. زمان عملیات هر کار P_i از قبل مشخص شده و قطعی می باشد. بین کارها اولویتی وجود ندارد و یک زمان تحویل مشترک، محدود کننده وجود دارد. یک کار دارای تاخیر است اگر عملیات آن پس از d تمام شود و دارای زودکرد است اگر زمان خاتمه عملیات آن قبل از d باشد. بدین ترتیب زودکرد و دیرکرد هر کار به صورت زیر تعریف می شود.

$$E_i = max\{0, d - c_i\} \qquad for\ all\ jobs \quad i = 1,2,...,n$$

$$T_i = max\{0, c_i - d_i\} \qquad for\ all\ jobs \quad i = 1,2,...,n$$

هدف پیدا کردن توالی است که دارای کمترین میزان جریمه باشد ؛ یعنی دارای کمترین میزان مجموع جریمه های زودکرد و دیرکرد باشد. بنابراین به دنبال یافتن توالی (s) هستیم به طوریکه مجموع زیر را کمینه کند.

$$f(s) = \sum_{i=1}^{n} (\alpha_i E_i + \beta_i T_i)$$

در ادامه دو خاصیت شناخته شده مسئله زمان بندی تک ماشین، که در ساختار مدل و الگوریتم مورد استفاده قرار گرفته‌اند ذکر می شود.

خاصیت ۱: در توالی بهینه زمان بیکاری بین فعالیتهای متوالی نخواهیم داشت.

خاصیت ۲: یک جواب بهینه وجود خواهد داشت اگر اولین کار در زمان صفر شروع شود و یا اینکه یک کار قبل از زمان d به اتمام رسیده باشد.

بنابر خاصیت ۲ در هنگام گشتن به دنبال جواب بهینه نباید محدودیت آغاز اولین کار در زمان صفر گنجانده شود.

مدل ریاضی

با توجه به نمادهای معرفی شده و توضیحات ارائه شده مدل ریاضی مورد استفاده در این مقاله به شرح زیر خواهد بود:

$$min\ f(s)=\sum_{i=1}^{n}(\alpha_i E_i+\beta_i L_i) \qquad (1)$$

$S.T.$

$E_i \geq d - c_i$ $\quad for\ i=1,...,n$ (2)

$T_i \geq c_i - d$ $\quad for\ i=1,...,n$ (3)

$c_i = x_i + p_i$ $\quad for\ i=1,...,n$ (4)

$y_{ij}(x_i + p_i) < x_j$ $\quad for\ i,j=1,...,n\ ;\ i \neq j$ (5)

$y_{ij} + y_{ji} = 1$ $\quad for\ i,j=1,...,n\ ;\ i \neq j$ (6)

$y_{ij} \in \{0,1\}$ $\quad for\ i,j=1,...,n$ (7)

$c_i \geq 0\ ,\ c_i\ integer$ $\quad for\ i=1,...,n$

$E_i \geq 0$ $\quad for\ i=1,...,n$ (8)

$T_i \geq 0$ $\quad for\ i=1,...,n$

$x_i \geq 0$ $\quad for\ i=1,...,n$

معادله شماره ۱ عبارتست از مجموع جریمه های دیرکرد و زودکرد و همان تابع هدف است که به دنبال کمینه کردن آن هستیم. محدودیت های شماره ۲و۳ تضمین می کنند که هر کار دارای دیرکرد است یا دارای زودکرد و یا اینکه دقیقا در زمان d خاتمه می یابد. محدودیت ۴ نشان دهنده زمان تکمیل کار i ام است. در محدودیت ۵ برای نشان دادن اینکه توالی کار i قبل از کار j است و یا بالعکس از متغیر صفر و یک y_{ij} استفاده شده که به صورت رابطه (۹) تعریف می شود.

$$y_{ij} = \begin{cases} 1 & , \ if\ job\ i\ is\ scheduled\ before\ job\ j \\ 0 & , \ otherwise \end{cases} \qquad (9)$$

به این ترتیب محدودیت های ۵ و ۶ تضمین می کنند که ماشین در هر لحظه فقط و فقط یک کار انجام می دهدو محدودیت ۷و۸ علامت متغیرهاست. این مدل به وسیله نرم افزار Lingo 8 حل شده است که در ادامه نتایج آن مورد بررسی قرار می گیرد در ضمن مقایسه‌ای مابین جواب های به دست آمده از این نرم افزار و الگوریتم SA تقویت شده در این مقاله ارائه خواهد شد. به این صورت دو جنبه مورد نظر ما برای پذیرش یا عدم پذیرش یک جواب نامطلوب که همان δ و t بودند تامین می‌شود.در الگوریتم ارائه شده در این مقاله از هر دو این شرایط همزمان استفاده شده است. الگوریتم مورد استفاده در بخش بعدی مطرح می‌شود.

الگوریتم شبیه سازی تبرید:

در بخش قبلی مختصری در باره الگوریتم شبیه سازی تبرید توضیح داده شد. در این بخش به معرفی مشخصات الگوریتمی که در این مقاله مورد استفاده قرار گرفته است می‌پردازیم. در الگوریتم ارائه شده احتمال پذیرش جواب تولید شده، در حالتیکه بهترین جواب یافت شده در مسیر الگوریتم نباشد، از رابطه زیر بدست می آید.

$p(\text{پذیرش جواب}) = Exp(-\delta/t)$

در هر دما آنقدر به جستجو ادامه می دهیم علاوه بر مشخصات مطرح شده برای مرحله فریز شده شرط تعادل دما را نیز مطرح کرد. بعبارت دیگر شرطی می گذاریم که اگر به آن برسیم دما را کاهش می دهیم. این شرط به این صورت است که رابطه زیر برقرار شود، به معنی آنست که دمای جاری هنوز به تعادل نرسیده است و باید بازهم مورد جستجو قرار گیرد.

$$\frac{\left|f_e - f_e^{'}\right|}{f_e^{'}} < \varepsilon$$

$f_e^{'}$ *میانگین مقدار تابع هدف در یک دوره :*

f_e : *میانگین مقدار تابع هدف از شروع حل*

ε : *مقدار آستانه برای تشخیص تعادل یا عدم تعادل*

نحوه نمایش جواب:

از مهمترین شاخصه های هر الگوریتم نحوه نمایش جوابهای آن است. این نحوه نمایش باید به صورتی باشد که امکان نمایش تمام جوابهای موجود در فضای جواب مسئله را داشته باشد و همچنین امکان تولید راحت جوابهای همسایه را نیز فراهم کند. به همین منظور برای حل مسئله تک ماشین در این مقاله نحوه نمایش هر جواب مسئله به صورت یک کروموزوم با تعداد n *ژن* (n=تعداد کارها) در نظر گرفته شده است. شکل ۱ نشان دهنده یک جواب برای یک مسئله فرضی با ۱۰ کار است.

۱	۲	۵	۸	۳	۶	۴	۹	۷	۱۰

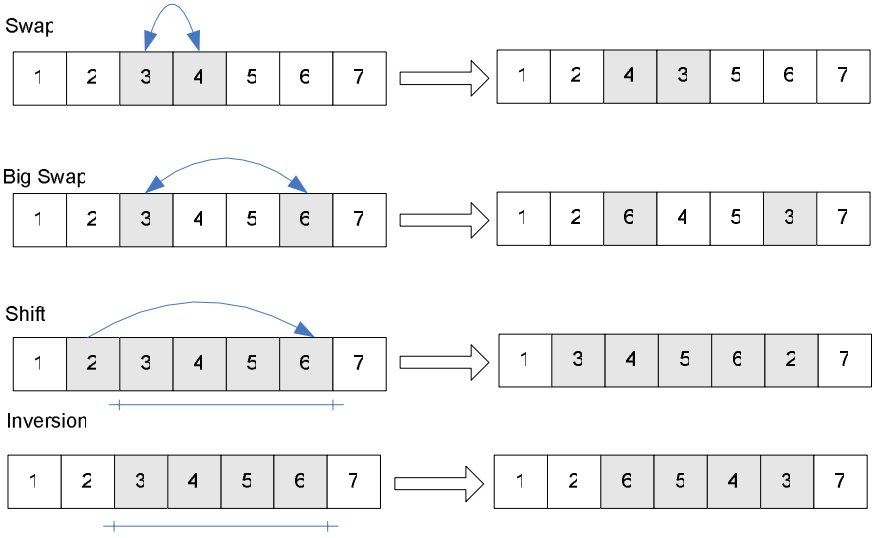
شکل۱: نمایش جواب مسئله

نحوه تولید جوابهای همسایه:

در این مقاله برای تولید جواب همسایه از عملگرهای زیر استفاده شده است.

- Swap : جانشینی
- Big Swap : جانشینی بزرگ
- Shift : انتقال
- Inversion : معکوس

عملکرد هر یک از این عملگرها در شکل۲ نشان داده شده است.



شکل۲ : عملکرد عملگرهای تولید جواب همسایه

الگوریتم شبیه سازی تبرید، برای بهبود یک جواب اولیه برای بدست آوردن حل نزدیک به بهینه و یا بعضا بهینه در مسائل NP-complete استفاده میشود. در این الگوریتم و به طور کلی در الگوریتم های فرا ابتکاری بوسیله گشتن هوشمندانه فضای حل به دنبال بدست آوردن جواب‌های بهینه یا نزدیک به بهینه هستیم. در الگوریتم SA تعدادی پارامتر وجود دارد . مقدار این پارامترها اغلب با استفاده از تجربه و تکرار ممتد آزمایشات تخمین زده می‌شوند . این پارامترها باید به نحوی مقدار دهی شوند که بهترین نرخ نزدیکی به جواب انتهایی را بدست دهند. این پارامترها عبارتند از :

- دمای اولیه
- ضریب کاهش دما (cooling factor)

- تعداد تکرار کل
- تعداد تکرار در هر دما
- دمای نهایی

مسلما این پارامتر ها و مقادیر آنها نسبت به مسئله تغییر می‌کنند. نکته‌ای که باید به آن توجه شود آنست که تعیین نامناسب هر یک از این پارامترها میتواند باعث پایین آمدن کیفیت جواب‌های بدست آمده از الگوریتم شود. به عنوان مثال اگر دمای اولیه بیش از حد بالا یا پایین در نظر گرفته شود ممکنست باعث جستجوهای بیهوده در حالت اول و یا جلوگیری از صورت گرفتن جستجوهای مطلوب و با احتمال بالا برای ارائه جواب بهبود دهنده، شود . به طور مشابه مثلا در مورد ضریب کاهش دما و تعداد تکرارها در هر دما و... . در نتیجه سعی بر آنست که با ترتیب دادن آزمایشهای متوالی و با استفاده از منطق تصادفی بودن الگوریتم شبیه سازی تبرید برای این مسئله خاص به دنبال مجموعه ای از پارامترها باشیم که بهترین یا یکی از بهترین ترکیبات ممکنه باشد.

ارائه جوابهای الگوریتم

در این بخش از مقاله جوابهای بدست آمده از الگوریتم SA تقویت شده که در بخش قبل معرفی شد، ارائه خواهد شد. همچنین در ادامه مقایسه‌ای بین جوابهای بدست آمده از طریق این الگوریتم با آنچه از طریق حل مدل ارائه شده در بخش ۲ بدست آمده، صورت خواهد گرفت تا کارایی الگوریتم ارائه شده مشخص گردد. برای شروع فرض می‌کنیم که مسئله‌ای با داده های جدول ۱ را در اختیار داریم.

جدول ۱: اطلاعات مسئله

کار	۱	۲	۳	۴
P_i	۱	۴۹	۱۰	۲۷
α_i	۲	۰	۳	۲
β_i	۳	۹	۵	۳
d	۸۰	۸۰	۸۰	۸۰

جدول ۲: مجموع هزینه های زودکرد و دیرکرد برای توالی ۴-۳-۲-۱

کار	زمان آغاز	زمان اتمام	جریمه زودکرد	جریمه دیرکرد
۱	۰	۱	۱۵۸	۰
۲	۱	۵۰	۰	۰
۳	۵۰	۶۰	۶۰	۰
۴	۶۰	۸۷	۰	۲۱
مجموع جریمه ها				۲۳۹

چون مجموع زمان عملیات ها، از d بزرگتر است، با یک موعد تحویل محدود کننده روبرو هستیم و مسئله دارای شرایط مورد نظر است. فرض کنید کارها را به ترتیب ۴-۳-۲-۱ در توالی قرار دهیم. مقدار بدست آمده برای مجموع هزینه های زود کرد و دیرکرد برابر با ۲۳۹ خواهد بود. جدول ۲ محاسبه این جریمه را نشان میدهد.

حال اگر توالی کارهای ۱ و ۲ را تعویض کنیم، مقدار مجموع جریمه ها به ۱۴۱ واحد کاهش میابد. به این صورت ما به دنبال پیدا کردن بهترین توالی هستیم به طوریکه این مجموع را به کمترین حالت ممکنه برساند. برای مثال ارائه شده، به علت کوچک بودن ابعاد مسئله جواب بهینه به راحتی قابل محاسبه است. این توالی بهینه عبارتست از ۱-۳-۴-۲ با مقدار جریمه ۲۳ واحد. البته باید توجه داشت که از آنجا که این مسئله دارای ابعاد کوچکی بود و در نتیجه فضای حل آن کوچک بود توانستیم جواب را به عنوان جواب بهینه مطلق معرفی کنیم، ولی در حالتی که ابعاد مسئله بزرگ باشد، به علت اینکه نمیتوان تمام فضای جواب را جستجو کرد، نمی‌توانیم جواب بهینه را با سهولت و قاطعیت مشخص کنیم.

برای نمایش جوابهای الگوریتم، مسائلی با ابعاد مختلف حل شده و در جدول ۳ نتایج ارائه شده اند.

در جدول ۳ مسائلی که بوسیله الگوریتم SA تقویت شده حل شده اند، بوسیله مدل ارائه شده و حل آن بوسیله نرم افزار Lingo 8 آورده شده است.

همانطور که در جدول ۳ نمایان است جوابهای ارائه شده توسط الگوریتم ارائه شده، نسبت به آنچه از نرم افزار Lingo بدست آمده است، برتری مطلق دارد. آنچه در اینجا حائز اهمیت است، آنست که جوابهای ارائه شده در دو روش حل، طی چه مدت زمانی بدست آمده اند؟

برای نشان دادن این موضوع در جدول ۴ این مقایسه مابین الگوریتم ارائه شده و نرم افزار Lingo صورت گرفته است.

جدول۴: مقایسه زمان حل الگوریتم SA و نرم‌افزار

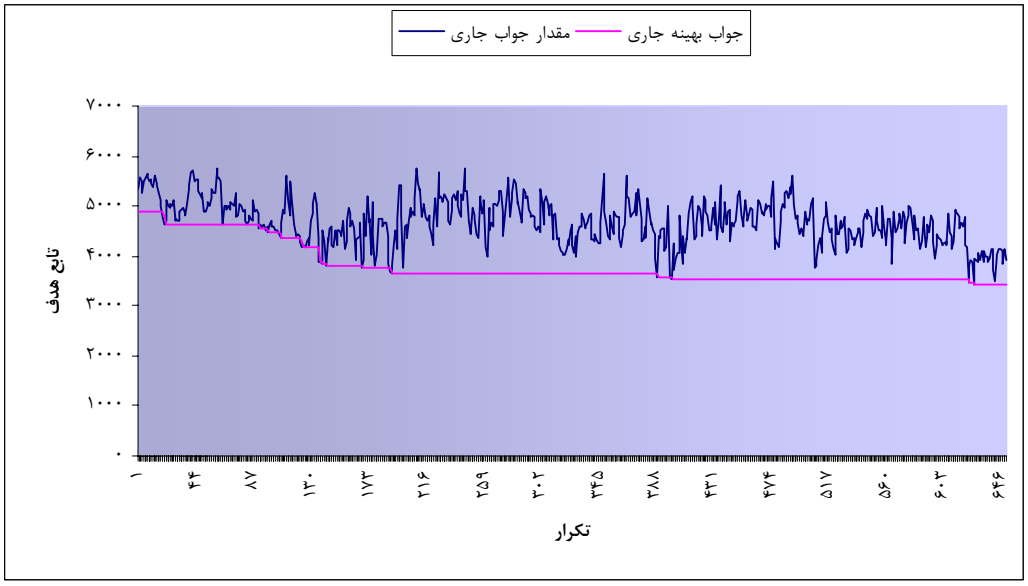
مسئله	۱	۲	۳	۴	۵	۶	۷
n	۴	۵	۸	۹	۱۰	۱۵	۲۰
SA	<۰۰:۰۱	<۰۰:۰۱	۰۰:۰۱	۰۰:۲۵	۰۰:۳۸	۰۰:۵۸	۰۱:۰۳
Lingo	<۰۰:۰۱	<۰۰:۰۱	۱۴:۰۷	۰۳:۴۴:۵۰	---	---	---

همانطور که در جدول ۴ مشخص است حتی برای تعداد کارهای به نسبت کوچک نیز زمان حل نرم افزار Lingo بسیار بزرگتر از الگوریتم SA تقویت شده است. این نشانی از کارایی بسیار خوب این الگوریتم است. در شکل ۱ روند حرکت الگوریتم ارائه شده در فضای حل یک مسئله با ۱۰

جدول۳: مقایسه بین جوابهای بدست آمده

مسئله	۱	۲	۳	۴	۵	۶	۷
n	۴	۵	۸	۹	۱۰	۱۵	۲۰
SA	۲۳	۱۲۵	۱۳۲۶	۲۱۶۳	۳۳۰۰	۱۴۸۷۲	۴۲۷۳۰
Lingo	۲۳	۱۲۵	۱۳۲۶	۲۱۶۳	---	---	---

کار نشان داده شده است. نکته‌ای که در این نمودار جالب توجه است اینست که الگوریتم با سرعت بسیار خوب و قابل قبولی به سمت جواب بهینه حرکت میکند.



شکل۳: روند حرکت الگوریتم در فضای جواب

در شکل۳ "مقدار جواب جاری " نشان دهنده روند حرکت جوابهای بررسی شده است و جواب بهینه جاری نشان دهنده حرکت الگوریتم به سمت جواب بهینه میباشد. در این مثال، کل جوابهای بهبود دهنده و غیر بهبود دهنده که پذیرفته شده اند حدود ۶۵۰ مرتبه از میان ۱۰۰۰۰ مرتبه تکرار الگوریتم بوده است.

نتیجه گیری:

در این مقاله با توجه به اهمیت زیادی که مسئله زمانبندی کارها بر روی یک ماشین دارد، و اینکه این مسئله به عنوان NP-Hard شناخته شده است، الگوریتمی برای حل آن ارائه شده است. الگوریتم ارائه شده تلفیقی از الگوریتم شبیه سازی تبرید با عملگرهای الگوریتم ژنتیک است. همچنین یک مدل برنامه ریزی عدد صحیح غیر خطی برای این مسئله ارائه شد و بوسیله نرم افزار Lingo8 حل شد. در انتها جوابهای بدست آمده از الگوریتم با آنچه توسط نرم افزار بدست آمد، مورد مقایسه قرار گرفت. مقایسات صورت گرفته از دو جنبه زمان حل و کیفیت جواب بدست آمده برتری مطلق الگوریتم پیشنهادی بر نرم افزار را نشان میدهد.

منابع :

[1] Kanet, J. J. (1981). Minimizing the average deviation of job completion times about a common due date. Naval Research Logistics Quarterly, 28, 643–651.
[2] Biskup, D., & Feldmann, M. (2001). Benchmarks for scheduling on a single-machine against restrictive and unrestrictive common due dates. Computers and Operations Research, 28, 787–801.
[3] Han S., Ishii H. and Fujii S. (1994), "One machine scheduling problem with fuzzy due dates", European Journal of Operational Research, Vol. 79(1), 1-12
[4] Chand, S., & Schneeberger, H. (1988). Single machine scheduling to minimize weighted earliness subject to no tardy jobs. European Journal of Operational Research, 34, 221–230.
[5] Glover, F., 1986. Future path for integer programming and links to artificial intelligence. Computers and Operations Research 5, 533– 549.
[6] Rubin, P.A., Ragatz, G.L., 1995. Scheduling in a sequence dependent setup environment with genetic search. Computers Operations Research 22, 85–99.
[7] Biskup, D., & Cheng, T. C. E. (1999). Multiple-machine scheduling with earliness, tardiness and completion time penalties. Computers and Operations Research, 26, 45–57.

[8] Yeung, W. K., Oguz, K. C., & Cheng, T. C. E. (2001). Single-machine scheduling with a common due window. Computers and Operations Research, 28, 157–175.
[9] K.R. Baker, G.D. Scudder, Sequencing with earliness and tardiness penalties: A review, Operations Research 38 (1)(1990) 22-36.
[10] Yeung, W. K., Oguz, K. C., & Cheng, T. C. E. (2001). Single-machine scheduling with a common due window. Computers and Operations Research, 28, 157–175.
[11] Ji-Bo Wang, Single machine scheduling with common due date and control-able processing times, Applied Mathematics and Computation 74(2006)1245-1254.
[12] Rinnooy Kan AHG. Machine scheduling problem: classification, complexity and computations. Dordrecht: Martinus Niihof, 1976.
[13] Smith WE. Various optimizers for single stage production. Naval Research Logistics Quarterly 1956;3:59-66.
[14] Holsenback JE, Russell RM,Markland RE, Philipoom PR. An improved heuristic for the single-machine, weighted tardiness problem. Omega, The International Journal of Management Science 1999; 27:485–95.