زمانبندی کارها در محیط کارگاه گردش کاری با معیار حداقل سازی مجموع دیرکرد و زود کرد کارها

فریبرز جولای
استادیار
گروه مهندسی صنایع، دانشکده فنی دانشگاه تهران

سید محمد معطر حسینی
دانشیار
دانشکده مهندسی صنایع، دانشگاه صنعتی امیرکبیر

سید محمد حسن حسینی
کارشناسی ارشد
دانشکده مهندسی صنایع دانشگاه صنعتی امیرکبیر

چکیده

در این مقاله مسئله زمانبندی n کار مستقل بر روی m ماشین در محیط کارگاه گردش کاری ترتیبی (Permutation flow shop) با زمانهای پردازش و موعد تحویل دلخواه برای کارها، بررسی و یک مدل با هدف کمینه کردن مجموع دیرکردها و زودکردها ($\sum E/T$) با استفاده ازتکنیک الگوریتمهای ژنتیک ارائه شده است. مسئله مفروض از نوع ایستا بوده و بجز محدودیت ماشین‌آلات (بعنوان منابع) محدودیت دیگری بر مسئله حاکم نیست. همچنین مسئله موردنظر از نقطه نظر اطلاعات در دسترس، معین میباشد. مدل ارائه شده به لحاظ بهینگی جواب نهایی و زمان حل مسئله ارزیابی و جوابهای آن با یکی از مدلهای موجود مقایسه شده است.

کلمات کلیدی

کارگاه گردش کاری، الگوریتم ژنتیک، معیار مجموع دیرکرد و زودکرد

Flow Shop Scheduling Problem with Minimizing Earliness/Tardiness Criteria

F. Jolai
Assistant professor
IE Department, Faculty of Engineering, University of Tehran

S. M. Moattar Hosseini
Associated professor
IE Department, Amir kabir University of Technology

S. M. Hassan Hosseini
MSc. Graduated student
IE Department, Amir kabir University of Technology

Abstract

In this paper, we consider a permutation flow shop scheduling problems with n jobs and m machines. The jobs have arbitrary processing times and due dates. Our objective is minimize the total earliness and tardiness of all jobs. We develop a genetic algorithm to solve this problem. The performance of the algorithm is evaluated through computational experiments.

Key words

Genetic algorithm, Permutation flow shop, Earliness/Tardiness criteria.

مقدمه

از زمان چاپ اولین مقاله جانسون [۱] درباره مسئله توالی عملیات کارگاه گردشکاری در سال ۱۹۵۴، این مسئله مورد توجه بسیاری از پژوهشگران قرار گرفته است. در مسئله کارگاه گردشکاری، m ماشین و n کار وجود دارد. هر کار نیازمند m عملیات میباشد و هر عملیات یک ماشین مختلف نیاز دارد. n کار با توالی یکسان روی m ماشین انجام میشوند. زمان فرآیند کار i روی ماشین j بصورت t_{ij} (i=1,2,...,n و j=1,2,...,m) بیان میگردد. هدف یافتن بهترین ترتیب انجام و تکمیل کارها میباشد. بهینگی توالی عملیات با درنظر گرفتن یک معیار کارایی مثلاً زمان تکمیل کل کارها، مجموع دیرکردها، و یا مجموع دیرکردها و زودکردها مشخص میشود. تنها تفاوت کارگاه گردش کاری ترتیبی با کارگاه گردشکاری در حالت عمومی اینستکه در حالت ترتیبی، کارها در مراحل مختلف فرآیند (ماشینها) از یکدیگر سبقت نمی‌گیرند. بعبارت دیگر ترتیب کارها روی ماشین اول، تعیین کننده ترتیب کارها روی تمام ماشینها میباشد. در حالت عمومی، کارها ممکن است در مراحل مختلف فرآیند از یکدیگر سبقت بگیرند. در عمل، اکثر مسائل کارگاه گردشکاری از نوع ترتیبی (Permutation) بوده و سبقت کارها از یکدیگر بندرت اتفاق میافتد. فرضهای اصلی مسئله عبارتند از:

۱- هر کار باید به ترتیب بوسیله تمام ماشینها پردازش شود.
۲- در یک زمان هر ماشین تنها یک کار را پردازش میکند.
۳- در یک زمان هر کار تنها بوسیله یک ماشین پردازش میشود.
۴- کارها بطور پیوسته انجام میشود و بریدگی مجاز نمی‌با شد.
۵- زمانهای آماده‌سازی (set-up) کارها مستقل از توالی آنهاست و بعنوان بخشی از زمان پردازش درنظر گرفته میشود.
۶- توالی عملیات کارها روی تمام ماشینها یکسان است و باید توالی عمومی تعیین شود.

در اکثر صنایع تولیدی بزرگ مانند خودروسازی و صنایع مونتاژ، عملیات پردازش قطعات و تکمیل کارها بصورت خطی و مرحلهای صورت میگیرد. لذا مسائل زمانبندی کارگاه گردشکاری طیف گسترده‌ای از مدلهای تولیدی و مونتاژ را پوشش میدهد. از طرف دیگر این مسئله با معیار حداقل کردن مجموع دیرکردها و زودکردها ($\sum E/T$) یک معیار تولید کننده ـ مشتری پسند بوده و در راستای اهداف سیستمهای تولید بموقع (JIT) میباشد و کمتر مورد توجه محققین بوده‌است.

اغلب مسائل توالی عملیات NP-Hard میباشند. لذا الگوریتمهای جستجوی دقیق و بهینه‌یاب برای حل اینگونه مسائل نیازمند زمان محاسباتی زیادی میباشند. بویژه این زمان با بزرگ شدن ابعاد مسئله بصورت نمایی افزایش مییابد و در برخی موارد نیز یافتن جواب بهینه عملاً امکانپذیر نمی‌باشد. به همین خاطر الگوریتمهای ابتکاری که درپی بدست آوردن جواب خوب در زمان کوتاه میباشند، در حل این مسائل کاربرد بسیاری پیدا کرده‌اند. در این مقاله یک الگوریتم ژنتیک برای مساله مورد نظر ارایه شده است. در بخش سوم مروری بر ادبیات موضوع صورت گرفته است. بخش چهارم به تشریح الگوریتم ژنتیک پرداخته است. در بخش پنجم به تعیین پارامترهای الگوریتم ژنتیک برای مسئله مورد بحث و تشریح مدل پرداخته میشود. در بخش ششم مدل ارائه شده ارزیابی میگردد. در بخش هفتم بحث و نتیجه‌گیری انجام میشود.

مروری بر ادبیات موضوعی

مسئله n قطعه و دو ماشین سری با فرضیات مدل پایه، توسط جانسون [۱] مورد بررسی قرار گرفته شد و جواب بهینه مسئله یعنی کمینه کردن دامنه عملیات ($C_{\max}$) را ارائه نمود. وی همچنین نشان داد ترتیب عملیات دو ماشین در برنامه بهینه یکسان است. بسیاری از محققین، کار جانسون را برای تعمیم به ماشینهای بیشتر و پیدا کردن الگوریتمهای بهینه ادامه دادند، ولیکن جز در حالات خاص موفقیت چندانی برای حل مسئله در حالت کلی بدست نیاوردند. دودک و همکاران[۲] نشان دادند مسئله با تابع هدف کمینه کردن $C_{\max}$ از مسائل NP-Hard است، این مسئله بصورت n/m/p/ $C_{\max}$ نمایش داده میشود. پس از آن روشهای ابتکاری سریعاً رشد یافتند.

ایگنال و شارج[۳]، اسمیت و دودک [۴]، و آشور[۵] به ارایه روش حل BB و قواعد حذف میپردازند. در این مقالات جواب بهینه برای مسائل با اندازه‌های کوچک بدست آورده شده و عملاً برای مسائل بزرگ کارایی ندارند.

پالمر[۶]، کمبل و همکاران [۷]، داننبرینگ[۸] به روشهای ابتکاری پرداخته‌اند. داننبرینگ[۸] و همچنین ذگردی و

همکاران[۹] روشهای ابتکاری را به دو دسته روشهای ساختنی و روشهای بهبود دادنی تقسیم کرده‌اند.

روش ابتکاری جالبی توسط ناواز و همکاران [۱۰] برای حل مسئله n/m/p/ C_{max} ارائه گردید. مبنای این روش براساس داشتن اولویت بالاتر برای قطعه با کل زمان پردازش بیشتر است. این روش به روش NEH مشهور است.

اشکال اساسی تمام روشهای بهبود دهنده با هر نوع تابع هدف، توقف در نقاط کمینه محلی است. برای رفع این مشکل، روشهای جدیدی از جمله SA (Simulated Annealing) ، GA (Genetic Algorithm) و TS (Tabu Search) توسعه داده شده‌اند. بزرگترین اشکال این روشها، طولانی بودن زمان حل مسائل نسبت به روشهای ابتکاری سنتی میباشد[۱۱].

کمینه کردن مجموع وزنی انحراف از موعد تحویل با وزنهای مشابه برای زودکرد و دیرکرد، یک تابع غیر منظم میباشد. در بررسیهای انجام شده برای حالت کلی مسئله Flow shop با معیار مجموع زودکرد و دیرکردها (مسئله n/m/p/ $\sum E/T$) مقاله‌ای پیدا نشد.

ساختار کلی الگوریتمهای ژنتیک

الگوریتمهای ژنتیک یک نوع از روشهای جستجوی تصادفی هستند که وابسته به مکانیزم انتخاب و ژنتیک طبیعی میباشند و برخلاف روشهای جستجوی متداول، با یک مجموعه اولیه از جوابهای تصادفی که آنرا جمعیت اولیه مینامیم شروع میکنند. هر عضو جمعیت را یک کروموزوم مینامیم که بیانگر یک جواب مسئله است. کروموزومها معمولاً بصورت رشتهای از اعداد یا نمادها میباشند. کروموزومها در طی تکرارهای متوالی الگوریتم تکامل مییابند. هر تکرار الگوریتم را یک نسل مینامیم. در هر نسل، صلاحیت کروموزومها براساس معیار عملکرد مورد ارزیابی قرار میگیرد. برای تولید نسل بعدی، کروموزومهای بعدی که آنها را فرزندان مینامیم، طی مراحل ذیل ایجاد میشوند:

۱ـ ادغام دو کروموزوم از نسل جاری با استفاده از یک عملگر ترکیب.

۲ـ تغییر دادن یک کروموزوم با استفاده از یک عملگر جهش.

انتخاب نسل جدید نیز طی مراحل ذیل انجام میگیرد:

۱ـ ارزیابی و تعیین میزان صلاحیت کلیه کروموزومها (شامل والدین و فرزندان) براساس معیار عملکرد.

۲ـ انتخاب تعداد ثابتی از بهترین کروموزومها (برابر با تعداد ثابت اعضاء جمعیت) و دور ریختن بقیه کروموزومها.

بعد از چند نسل (تکرار)، الگوریتم بسمت بهترین کروموزومها همگرا میشود. اگر P(t) و C(t) بترتیب بیانگر والدین و فرزندان در نسل t ام باشند، آنگاه ساختار کلی الگوریتمهای ژنتیک را میتوان بصورت رویه ذیل نشان داد:

```
Procedure: Genetic Algorithms
Begin
    t=0
    initialize  P(t)
    evaluate  P(t)
    while  (not termination condition) do
        recombine  P(t) by crossover and mutation
        to yield  C(t)
        evaluate  C(t)
        select  P(t+1) from  P(t) and  C(t)
        t=t+1
    loop
End
```

معمولاً جمعیت اولیه بصورت تصادفی از میان جوابهای مسئله ایجاد میشود. عملیات ترکیب، جهش و ارزیابی (انتخاب) عملیات اصلی در الگوریتمهای ژنتیک میباشند. الگوریتمهای ژنتیک بطور متناوب روی فضای جواب و فضای کد گذاری شده کار میکند. در واقع عملگرهای ژنتیک (ترکیب و جهش)، روی فضای کدْ گذاری شده کار میکنند و عملهای تعیین صلاحیت و انتخاب، روی فضای جواب کار میکنند.

در تطبیق ساختار عمومی الگوریتمهای ژنتیک برای مسئله n/m/p/ $\sum E/T$ ، آزمایشهای متعددی انجام شد تا بهترین وضعیت برای پارامترها مشخص شود. نتایج در بخش بعدی آورده شده است.

بررسی پارامترهای الگوریتم ژنتیک برای مسئله n/m/p/$\sum E/T$

ابتدا لازم است مسئله در قالب الگوریتمهای ژنتیک نمایش داده شود. سپس مراحل و پارامترهای الگوریتم بصورت مناسب تعریف و مشخص شوند.

روش کد گذاری

در این نمایش هر کروموزوم نشاندهنده یک توالی است. واضح است که عملگرهای ترکیبی و جهشی باید طوری عمل کنند که توالی بودن (جواب ممکن) کروموزومها تضمین شود.

برای کدگذاری مسائل ترتیبی از قبیل مسئله زمانبندی Flow Shop عموماً از کد گذاری جایگشتی استفاده میشود. در این کد گذاری، هر کروموزوم رشته‌ای از اعداد است که ترتیب و اولویتها را مشخص میکند.

اندازه جمعیت و تعداد نسل

اندازه جمعیت (تعداد جوابهای ممکن در هر نسل) یکی از پارامترهای موثر در حجم و زمان انجام محاسبات میباشد. پس از انجام آزمایشهای متعدد در نهایت مشخص گردید اندازه جمعیت برابر ۲۰ مناسب میباشد. تعداد نسل (تعداد تکرار الگوریتم) نیز عامل موثری در تعیین حجم محاسبات و میزان بهینگی جواب نهایی میباشد. هرچه تعداد نسل بیشتر باشد زمان محاسبات بالا رفته و جواب نهایی نیز بهبود بیشتری خواهد یافت. درخصوص تعداد نسل، این نکته مورد توجه قرار گرفت که هر مسئله رفتار متفاوتی با سایر مسائل دارد و درنظر گرفتن یک تعداد نسل ثابت برای تمام مسائل در اندازههای مختلف، منطقی بنظر نمیرسد. اگر تعداد نسل یک عدد ثابت درنظر گرفته شود، مشاهده میگردد که برخی از مسائل در تعداد نسل کمی، جواب مناسبی را پیدا کرده‌اند ولی برای رسیدن به تعداد نسل مشخص شده باید تکرار را ادامه دهند. همچنین ممکن است در برخی مسائل بزرگتر، جواب بطور مرتب بهبود پیدا کند ولی چون تعداد نسل بیشتری مجاز نبوده، بهبود بیشتر امکان‌پذیر نیست. در الگوریتم ژنتیک پیشنهادی سعی شده است این عیب تا حدی برطرف گردد. برای این منظور تعداد نسل از قبل مشخص نیست و به میزان بهبود مسئله بستگی دارد. در این حالت شرط توقف، تولید تعداد نسل ثابتی بعد از آخرین بهبود است. مقدار این پارامتر با آزمایشهای مختلف برابر ۷۵ تکرار الگوریتم بعد از آخرین بهبود تعیین شد.

تشکیل جمعیت اولیه

بمنظور تشکیل جمعیت اولیه (۲۰ جواب ممکن اولیه) حالات زیر مورد بررسی قرار گرفت:

- تشکیل جمعیت اولیه بصورت کاملاً تصادفی.
- قراردادن نتیجه روش ابتکاری EDD (اجرای کارها به ترتیب صعودی موعدهای تحویل آنها) بعنوان یک جواب در جمعیت اولیه و بقیه جمعیت اولیه (۱۹ جواب) بصورت تصادفی.
- قراردادن نتیجه روش ابتکاری NEH تعدیل شده در جمعیت اولیه و بقیه جمعیت بصورت تصادفی.
- قراردادن نتیجه روشهای ابتکاری EDD و NEH تعدیل شده در جمعیت اولیه و بقیه جمعیت بصورت تصادفی.

با توجه به اینکه در روش NEH موعد تحویل کارها درنظر گرفته نمیشود، به منظور استفاده از آن برای تابع هدف $\sum E/T$، نیازمند انجام تغییراتی در آن هستیم. به همین خاطر ابتدا کارها را براساس ترتیب غیرنزولی موعد تحویل مرتب نموده و از فهرست حاصل در قدم ۲ الگوریتم NEH استفاده مینماییم.

روش NEH تعدیل شده

قدم ۱: کارها را براساس مقدار غیرنزولی موعد تحویل مرتب کنید.

قدم ۲: دو کار اول و دوم لیست قدم ۱ را برداشته و بهترین توالی این دو کار را با محاسبه مقدار $\sum E/T$ برای دو توالی ممکن، مشخص کنید. موقعیت نسبی این دو کار در بقیه قدمهای الگوریتم ثابت باقیمانده و تغییر نمیکند. قرار دهید $i = 3$.

قدم ۳: کار در موقعیت i ام لیست قدم ۱ را برداشته و بهترین توالی را با قرار دادن این کار در i موقعیت توالی جزئی قدم قبلی، بدون تغییر در قدمهای نسبی، مشخص کنید. موقعیت نسبی کارها در این توالی جزئی، در بقیه قدمهای الگوریتم ثابت باقی میماند.

قدم ۴: اگر $i = n$ باشد، توقف کنید. در غیر اینصورت قرار دهید $i = i + 1$ و به قدم ۳ بروید.

آزمایشهای متعدد نشان داد که حالت دوم (قراردادن نتیجه روش EDD در جمعیت اولیه) نسبت به سایر موارد بهتر عمل میکند. لذا برای محاسبات بعدی، نتیجه روش EDD بعنوان یک رشته در جمعیت اولیه قرارداده میشوند.

عملگرهای ژنتیک و احتمال وقوع آنها

عملگر ترکیب، مهمترین عملگر ژنتیک است که روی دو کروموزوم اعمال شده و فرزندان را با استفاده از تلفیق خصوصیات کروموزومهای والد ایجاد میکند. در بررسیهای انجام شده، چندین نوع عملگر ترکیبی مورد بررسی قرار گرفتند. نتایج نشان داد عملگر ترکیب یک نقطه‌ای در حل مسئله مورد بررسی (n/m/p/ $\sum E/T$) بهتر عمل میکند. در این عملگر، نقطه ترکیب در هر بار تکرار، بصورت تصادفی انتخاب میشود.

در بین انواع عملگرهای جهشی مورد آزمایش، کارایی عملگر جهش تعویض بیشتر از سایر عملگرها بود. انتخاب ژنها برای تعویض نیز در هربار تکرار بصورت تصادفی انجام میشود. عملگر جهش به صورت تصادفی ژنهای کروموزومها را تغییر میدهد. اثر این کار در روند حل مسئله اینست که اولاً از محدود شدن تمام جوابهای جمعیت در یک نقطه بهینه موضعی جلوگیری میکند. ثانیاً با تغییر تصادفی ژنهای کروموزومهای بد، امکان بهبود آنها را فراهم میکند. روش انجام جهش روی کروموزومها نیز همانند ترکیب وابسته به نحوه کد گذاری و نوع مسئله است.

بمنظور تعیین احتمال وقوع هرکدام از عملگرهای ترکیب و جهش برای یک رشته، روشها و راهکارهای مختلفی مورد آزمایش و بررسی قرار گرفت تا پارامترهای مناسب تشخیص داده شود. ابتدا این دو احتمال را بصورت زیر تعریف میکنیم:

P_{c_i}: احتمال استفاده از عملگر ترکیب برای دو رشته i و i+1 ام.

P_{m_i} : احتمال استفاده از عملگر جهش برای رشته i ام.

بطور کلی در الگوریتمهای ژنتیک بهتر است که عملگر ترکیب روی کروموزومهای خوب اعمال شود. چرا که ترکیب دو کروموزوم خوب معمولاً باعث پیدایش کروموزوم بهتر میشود هرچند که این موضوع صد در صد نمیباشد. عملگر جهش بهتر است روی کروموزومهای بد اعمال شود. آزمایشهای متعدد و دقیقی در زمینه احتمال وقوع عملگرهای ترکیب و جهش روی هرکدام از کروموزومها انجام شد. نتیجه نهایی این آزمایشها، نشان داد که بهترین احتمال برای وقوع عملگرهای ترکیب و جهش روی کروموزوم i ام میبایستی بصورت زیر تعریف گردد:

$$P_{c_i} = \frac{N-i}{N} \qquad P_{m_i} = \frac{i}{N}$$

تعریف دو احتمال فوق از آنجا ناشی شد که در مدل ارائه شده، بعد از هربار اعمال عملگرهای ترکیب و جهش و تولید تعدادی فرزند، مجموعه کل کروموزومها (مرکب از والدها و فرزندهای ایجاد شده) مطابق معیار عملکرد ($Min\sum E/T$) ارزیابی و بترتیب از بهترین به بدترین، مرتب میشوند. سپس ۲۰ کروموزوم بهتر انتخاب میشوند بطوریکه اولین کروموزوم، بهترین کروموزوم بوده و بترتیب الی آخر. آخرین کروموزوم بدترین کروموزوم میباشد. حال در تکرار بعدی کروموزومهای ابتدایی که بهتر میباشند مطابق تعریف دو احتمال فوق، دارای شانس بیشتری برای ترکیب و شانس کمتری برای جهش میباشند. بالعکس، کروموزومهای انتهایی که بد میباشند، شانس کمتری برای ترکیب و شانس بیشتری برای جهش دارند.

باتوجه به مباحث قبلی، الگوریتم حل مسئله در قدمهای زیر خلاصه میشود:

قدم ۰ : تعریف مسئله و ورود مقادیر اولیه

- تعداد کارها =: n
- تعداد ماشینها =: m
- ماتریس زمان پردازش کارها =: t_{ij}
- بردار موعد تحویل کارها =: d_i
- تعداد جمعیت ، ۲۰ =: N
- قراردادن مقدار اولیه صفر برای سایر متغیرها و ماتریسها

قدم ۱: تشکیل جمعیت اولیه

ـ تشکیل N-1 رشته (توالی) بصورت تصادفی، بطوریکه هیچ دو رشته‌ای یکسان نباشد.

ـ تشکیل رشته N ام با استفاده از جواب الگوریتم EDD.

قدم ۲: ارزیابی و مرتب کردن جمعیت اولیه

ـ محاسبه مقدار $\sum E/T$ برای کلیه رشته‌ها.

ـ مرتب کردن رشته‌ها بترتیب صعودی مقدار $\sum E/T$ مربوط به آنها.

قدم ۳: ترکیب

ـ انجام عملیات ذیل بترتیب از رشته اول تا رشته N-1 ام.

ـ انتخاب دو رشته i و i+1 ام با احتمال (N-i)/N .

ـ ایجاد عددی تصادفی بین ۱ تا n-1 (مثلاً g).

ـ تشکیل رشتهای جدید که ژنهای ۱ تا g آن از رشته i ام و بقیه بترتیب از رشته i+1 ام انتخاب میشوند.

قدم ۴: جهش

انجام عملیات ذیل بترتیب از اولین تا آخرین رشته.

ـ انتخاب رشته i ام با احتمال i/N .

ـ ایجاد دو عدد تصادفی نامساوی بین ۱ تا n (مثلاً g1 و g2).

ـ تعویض موقعیت دو ژن g1 و g2 با یکدیگر و تشکیل رشته جدید.

قدم ۵: ارزیابی و انتخاب

ـ محاسبه مقدار $\sum E/T$ برای کلیه رشته‌ها.

ـ مرتب کردن رشته‌ها بترتیب صعودی مقدار $\sum E/T$ مربوط به آنها.

ـ انتخاب N رشته اول بعنوان نسل جدید.

قدم ۶: مقایسه جواب

ـ چنانچه بهترین جواب نسل جدید از بهترین جواب نسل قدیم بهتر است، قرار دهید ITER = 0 و به قدم ۷ بروید.

ـ چنانچه بهترین جواب نسل جدید از بهترین جواب نسل قدیم بهتر نیست، قرار دهید ITER = ITER + 1 و به قدم ۷ بروید.

قدم ۷: شرط توقف

ـ چنانچه ITER < ۷۵ است، به قدم ۳ بروید.

ـ چنانچه ITER > = ۷۵ است، بهترین جواب را نمایش داده و الگوریتم پایان مییابد.

ارزیابی روش

در این بخش کارایی مدل ارائه شده را مورد بررسی و ارزیابی قرار میدهیم تا توانایی و نقاط ضعف آن مشخص شود. روش ارائه شده از دو جنبه بهینگی جواب و زمان اجرا تا دستیابی به جواب نهایی، مورد ارزیابی و آزمون قرار میگیرد. یکی از مشکلات اساسی در انجام این تحقیق این بود که در مطالعات انجام شده متاسفانه هیچگونه مدل مشابهی در زمینه کمینه کردن مجموع دیرکردها و زودکردها در مسائل Flow Shop مشاهده نشد تا بتوان مقایسه‌ای بین آنها انجام داد. بهمین علت دو مدل دیگر نیز با معیار کمینه کردن مجموع دیرکردها و زودکردها تهیه شد که یکی از آنها مدل انشعاب و تحدید (BB) و دیگری روش NEH تعدیل شده برای معیار $\sum E/T$ میباشد. لازم به توضیح است که تمامی برنامه‌های کامپیوتری در محیط MATLAB برنامه‌نویسی و کد شده‌اند اجرای کلیه برنامه‌ها و نرم‌افزارها نیز توسط یک دستگاه رایانه شخصی پنتیوم III با مشخصات CPU:800 MHz و RAM:256 MB انجام شده است.

طراحی آزمایش

ورودیهای اصلی مسئله عبارتند از موعد تحویل کارها و زمان پردازش آنها. زمان پردازش کارها بطور یکنواخت بین ۰ تا ۱۰۰ درنظر گرفته شده است. این توزیع دارای واریانس بالا بوده و اجازه خواهد داد تا مدل مورد بررسی تحت شرایط مختلف که برخی از آنها نامساعد است، ارزیابی شود.

بمنظور تعیین موعد تحویل کارها باید درنظر داشت که این پارامتر وابسته به مدت زمان پردازش کارها بوده و باید موعد تحویل کارها بصورت تصادفی طوری تعیین شود که مسئله تحت شرایط مختلف و بعضاً نامساعد قرار گرفته تا مدل در حالتهای مختلف ارزیابی شود. محققین مختلف دو عامل دیرکرد (τ) و دامنه موعد تحویل (R) را مهم دانسته و براساس آنها مسائل را بصورت تصادفی تولید کرده‌اند. برای این منظور از روابط ذیل برای بدست آوردن توزیع یکنواخت موعد تحویل کارها استفاده میشود:

$$\bar{d} = (1-\tau) * \sum_{j=1}^{m} \sum_{i=1}^{n} P_{ij} \qquad (۱-۴)$$

توزیع یکنواخت موعد تحویل کارها:

$$[\bar{d} * (1-\frac{R}{2}) \text{ و } \bar{d} * (1+\frac{R}{2})] \qquad (۲-۴)$$

اکثر محققان ازجمله او و مورتون [۱۲] و همچنین ذگردی و همکاران [۹] ، مقدار عامل دیرکرد (τ) را برابر ۰،۲ و ۰،۶، و مقدار دامنه موعد تحویل (R) را برابر ۰،۶ و ۱،۲ فرض کرده‌اند. این اعداد در تحقیقات استاندارد شده و محققین از این اعداد برای تولید مسائل تصادفی استفاده مینمایند.

در عمل، رابطه (۴-۱) برای مسائل Flow Shop مناسب نیست. چرا که مجموع زمانهای پردازش ($\sum_{j=1}^{m} \sum_{i=1}^{n} P_{ij}$) تمام کارها روی تمام ماشینها عدد بزرگی است و معمولاً اکثر کارها دارای زودکرد میباشد. برای رفع این مشکل با بیان دیگر میتوان رابطه (۴-۱) را بصورت زیر نوشت:

$$\bar{d} = (1-\tau) * M \qquad (۳-۴)$$

که در آن M زمان ختم کلیه کارهاست. در مسئله تک ماشین، مقدار M برابر مجموع زمان پردازش کارهاست. در مسئله دو ماشین، مقدار بهینه M توسط ترتیب جانسون بدست میآید. ولی در حالت کلی Flow Shop ، روشی برای محاسبه مقدار بهینه M وجود ندارد. لذا معمولاً مقدار M را از یک توالی تصادفی مسئله بدست میآورند. در این حالت همواره مقدار M از مجموع زمانهای پردازش تمام کارها روی تمام ماشینها کمتر است. رابطه (۴-۳) برای مسائل Flow Shop بهتر از رابطه (۴-۱) عمل میکند. ما نیز در این پروژه بمنظور تعیین توزیع یکنواخت موعد تحویل کارها، بترتیب از دو رابطه (۴-۳) و (۴-۲) استفاده میکنیم.

از ترکیب دو عامل دیرکرد (τ) و دامنه موعد تحویل کارها (R) ، چهار دسته مسئله ایجاد میشود. این چهار دسته در جدول شماره ۱ مشخص شده‌اند.

جدول شماره (۱) دسته‌بندی مسائل.

شماره دسته	مقدار τ	مقدار R
دسته اول	۰،۲	۰،۶
دسته دوم	۰،۲	۱،۶
دسته سوم	۰،۶	۰،۶
دسته چهارم	۰،۶	۱،۶

اندازه مسائل بنحوی طراحی میشود که متنوع بوده و اندازه‌های کوچک، متوسط و بزرگ در آن وجود داشته باشد. این مسائل مطابق جدول شماره ۲ در ۲۰ گروه طبقه‌بندی شده‌اند.

جدول شماره (۲) گروه‌بندی مسائل.

شماره گروه	تعداد کارها	تعداد ماشینها
۱	۴	۵
۲	۴	۱۰
۳	۴	۲۰
۴	۶	۵
۵	۶	۱۵
۶	۶	۲۰
۷	۹	۷
۸	۹	۲۰
۹	۹	۲۵
۱۰	۱۵	۱۰
۱۱	۱۵	۲۵
۱۲	۱۵	۳۰
۱۳	۲۵	۲۰
۱۴	۲۵	۳۰
۱۵	۲۵	۳۵
۱۶	۴۰	۱۰
۱۷	۴۰	۲۰
۱۸	۴۰	۴۵
۱۹	۵۰	۲۰
۲۰	۵۰	۵۰

بمنظور آزمون و ارزیابی مدل اصلی ارائه شده در این مقاله ابتدا از طریق روش انشعاب و تحدید برای مسائل مختلف جواب بهینه را بدست آورده و جوابهای بدست آمده از طریق الگوریتم ژنتیک را با آن مقایسه نماییم. لذا بدین طریق مشخص میشود که در چند درصد اوقات (چند درصد مسائل)، مدل به جواب بهینه رسیده است.

با توجه به اینکه در مطالعات انجام شده مدل مشابهی که بتوان جواب و زمان حل مدل ارائه شده در این پروژه را با آن مقایسه نمود یافت نشد. بهمین دلیل در الگوریتم NEH که برای حل مسائل Flow Shop از نظر بهینگی جواب و زمان حل بسیار موفق بوده است، تغییر مختصری بعمل آورده تا بتوان جواب حاصل از مدل الگوریتم ژنتیک را با آن مقایسه نمود. به این تغییر در بخشهای قبل اشاره و نام روش NEH تعدیل شده را برای آن انتخاب کردیم.

برای مقایسه روش NEH تعدیل شده با الگوریتم ژنتیک، تعداد ۵ مسئله در هر گروه (مجموعاً ۱۰۰ مسئله در ۲۰ گروه) تولید شده است. این مسائل برای ۴ دسته (جدول شماره ۱) طراحی و در مجموع ۴۰۰ مسئله تولید شده است. برای مقایسه الگوریتم ژنتیک با نتایج روش انشعاب و تحدید ۱۰ مسئله در هر گروه حل شده است.

حل مسائل و نتایج حاصل

جدول ۳ نشاندهنده تعداد و درصد جواب بهینه در الگوریتم ژنتیک میباشد. از آنجائیکه در الگوریتم انشعاب و تحدیدبرای دست یافتن به جواب بهینه میبایستی تمام توالیها تعیین و تک تک مورد بررسی و ارزیابی واقع شوند، لذا این الگوریتم بسیار زمان بر بوده و همچنین بدلیل تعداد زیاد توالیهای ممکن در ابعاد بزرگ مسئله، تنها برای مسائلی که دارای ۹ کار یا کمتر میباشند قادر به یافتن جواب بهینه میباشد. الگوریتم ژنتیک در ۹۷ درصد اوقات موفق به یافتن جواب بهینه شده است.

جدول ۴ مقایسه بین مدل GA و NEH در دستیابی به جواب بهتر را نشان میدهد. این جدول بیانگر برتری مطلق مدل GA در هر ۲ گروه مسایل است. جدول ۵ زمان اجرای مدل GA تا رسیدن به جواب نهایی را نشان میدهد. اختلاف نتایج بین چهار دسته مسائل (که از تغییر پارامترهای τ و R ایجاد شده‌اند) قابل بررسی میباشد.

جدول شماره (۳) تعداد و درصد جواب بهینه بدست آمده از مدل.

گروه	اندازه مسئله		تعداد مسئله بررسی شده در هر دسته	تعداد جواب بهینه در هر دسته				مجموع	
	تعداد کار	تعداد ماشین		دسته اول	دسته دوم	دسته سوم	دسته چهارم	تعداد	درصد
۱	۴	۵	۱۰	۱۰	۱۰	۱۰	۱۰	۴۰	٪۱۰۰
۲	۴	۱۰	۱۰	۱۰	۱۰	۱۰	۱۰	۴۰	٪۱۰۰
۳	۴	۲۰	۱۰	۱۰	۱۰	۱۰	۱۰	۴۰	٪۱۰۰
۴	۶	۵	۱۰	۱۰	۱۰	۱۰	۱۰	۴۰	٪۱۰۰
۵	۶	۱۵	۱۰	۱۰	۱۰	۱۰	۱۰	۴۰	٪۱۰۰
۶	۶	۲۰	۱۰	۱۰	۱۰	۱۰	۹	۳۹	٪۹۸
۷	۹	۷	۱۰	۱۰	۷	۱۰	۸	۳۵	٪۸۸
۸	۹	۲۰	۱۰	۱۰	۹	۸	۹	۳۶	٪۹۰
۹	۹	۲۵	۱۰	۱۰	۱۰	۹	۹	۳۸	٪۹۵
مجموع	تعداد		۹۰	۹۰	۸۶	۸۷	۸۵	۳۴۸	٪۹۷
	درصد		ــــ	٪۱۰۰	٪۹۶	٪۹۷	٪۹۴	٪۹۷	ــــ

جدول شماره (٤) مقایسه جوابهای دو مدل GA و NEH.

گروه	اندازه مسئله		تعداد مسئله در هر دسته	میانگین جواب GA و NEH در هر دسته						میانگین کل	
	تعداد کار	تعداد ماشین		دسته اول		دسته دوم		دسته سوم		دسته چهارم			
				مدل GA	مدل NEH	مدل GA	مدل NEH	مدل GA	مدل NEH	مدل GA	مدل NEH	مدل GA	مدل NEH
۱	٤	٥	٥	٦٢٤	٧٠٨	٥٨١	٦٠٢	٣٠٦	٤٠٩	٦٠٨	٦٥٧	٥٣٠	٥٩٤
۲	٤	١٠	٥	٩٦٠	٩٦٢	٦٨١	٧٤٠	٧٢٤	٧٣٧	٩٩٥	١١٣٥	٨٤٠	٨٩٤
۳	٤	٢٠	٥	١٢٢٣	١٢٢٣	٢١٥٩	٢٢٣٧	١٥٥١	١٦١٢	٢١١٢	٢٢٦٤	١٧٦١	١٨٣٦
٤	٦	٥	٥	١٢٣٤	١٣٩٩	٨٨٥	١٠١٠	٧٠١	٨٥٢	٨٢٤	١٠٧٠	٩١١	١٠٨٣
٥	٦	١٥	٥	١٩٥٣	٢٢٤٤	٢٢٤٤	٢٤٨٧	١٢٨٥	١٧٨٤	٢٤٠٤	٢٦٢٦	١٩٨٤	٢٢٨٥
٦	٦	٢٠	٥	٢٢٣٧	٢٤٤٠	٢٦٩٣	٣١٤٨	٢٣٦٧	٢٥٩٧	٢٩٨٨	٣٣٣٣	٢٥٧١	٢٨٧٩
٧	٩	٧	٥	٢٦٦٧	٣٦٢٨	١٢٢٠	١٨٥٤	١٢٠٢	١٦٤٦	١٦٠٨	١٩٩٩	١٦٧٤	٢٢٨٢
٨	٩	٢٠	٥	٤٩٨٧	٦٢٩٥	٤٠٨٣	٤٧٧٠	٣٠٨٨	٤٣٣٥	٥٥٦١	٦٥٩٠	٤٤٣٠	٥٦٢٢
٩	٩	٢٥	٥	٤٦٣٧	٦١١٥	٥٥٢١	٨٣٦٣	٤٣١٣	٥٦٨٠	٦٦٥١	٧٣٩٠	٥٢٨٠	٦٨٨٧
١٠	١٥	١٠	٥	٨٩١٢	١٠٠٥٧	٣٩٤٦	٥٨٣٨	٢٦٦٣	٥١٥٤	٤١٥٨	٥٥٩٨	٤٩١٩	٦٦٦٢
١١	١٥	٢٥	٥	١٢٤٣٧	١٤٨٣٢	١٠٣٨٠	١٢٦٩٤	٦٥٤٨	٨٦٤٢	١٢٠٨٩	١٥٢١٠	١٠٣٦٣	١٢٨٤٤
١٢	١٥	٣٠	٥	٩٢٣٨	١٣٧٣٤	٩٤٨٩	١١٤٠٢	٦٥٦٠	٨٥٥٤	١١٦٨١	١٣٨٤٠	٩٢٤٢	١١٨٨٢
١٣	٢٥	٢٠	٥	٢٨٦٧٨	٣٣٦١٤	١٤٥١١	١٨٤٣٩	٥١٧١	٩٥٩٩	١٣٣٥٧	١٨٠٣٣	١٥٤٢٩	١٩٩٢١
١٤	٢٥	٣٠	٥	٢٣٦٤٩	٢٧٨٢٣	١٥٠١٦	١٩١٢٦	٩١٨٨	١٣٩٩١	٢٤٤٦٦	٣١٧١٧	١٨٠٨٠	٢٣١٦٤
١٥	٢٥	٣٥	٥	٢٤٥٧٧	٣٣٧٨٩	٢٠٣٩١	٢٣٢٧٩	١٦٥٠٥	٢٢٧٣٩	٢٧١٦٨	٣٣٨٢٩	٢٢١٦٠	٢٨٤٠٩
١٦	٤٠	١٠	٥	٤٢٤٥٦	٥٢٧٠١	٢٩٤١٥	٤٢٨٠٤	٦٤٨٩	١٠٧١٣	١٤٤٦٧	٢٥٠٢١	٢٣٢٠٧	٣٢٨١٠
١٧	٤٠	٢٠	٥	٥٦٠٢٦	٦٦١٧٠	٤٣٥٨٦	٥٧٦٣٤	٣٨٧١٠	٥٣٤١٣	٥٧١٢٧	٧٥٣٢٧	٤٨٨٦٢	٦٣١٣٦
١٨	٤٠	٤٥	٥	٦٧٠٥٦	٨٩٣٣٣	٤٩٤٤٠	٦٠٠٠٨	٥٣٧٠٦	٧٣٨٩٤	٧٢٨٣٨	١٠٢٩٣٢	٦١٠١٠	٨١٥٤٢
١٩	٥٠	٢٠	٥	٨٣٠٧٩	١٠٤٩٨٥	٤١٩٩٠	٦٦٢٤٦	١٤٥٣١	٢٧٤٦٩	٣٢٧٥٥	٥٠٢٥١	٤٣٠٨٨	٦٢٢٣٨
٢٠	٥٠	٥٠	٥	٩٥٢٣٥	١١٥٨٩٦	٦٦٥٨٢	٨٧٥٨٩	٤٤١٧١	٦٣٥٦٣	٨٢٦٧٩	١٠٥٢٤٣	٧٢١٦٧	٩٣٠٧٣
میانگین کل				٢٣٥٩٣	٢٩٤٢٣	١٦٢٤٠	٢١٥١٣	١٠٩٨٩	١٥٨٦٩	١٨٨٧٩	٢٥٢٠٣	١٧٤٢٥	٢٣٠٠٢

جدول شماره (۵) زمان حل مدل GA تا رسیدن به جواب نهایی (ثانیه).

گروه	اندازه مسئله		تعداد مسئله در هر دسته	دسته اول	دسته دوم	دسته سوم	دسته چهارم	میانگین کل
	تعداد کار	تعداد ماشین						
۱	۴	۵	۵	۳،۱	۳،۱	۳،۲	۳،۱	۳،۱
۲	۴	۱۰	۵	۵،۲	۵،۲	۵،۲	۵،۲	۵،۲
۳	۴	۲۰	۵	۷،۳	۶،۲	۷،۵	۷،۷	۷،۲
۴	۶	۵	۵	۴،۶	۴،۵	۴،۵	۴،۵	۴،۵
۵	۶	۱۵	۵	۱۰،۸	۱۰،۸	۱۰،۷	۱۰،۹	۱۰،۸
۶	۶	۲۰	۵	۹،۶	۱۱،۷	۱۱،۲	۱۲،۴	۱۱،۲
۷	۹	۷	۵	۱۰،۷	۷،۶	۶،۵	۱۷،۲	۱۰،۵
۸	۹	۲۰	۵	۲۴،۶	۲۵،۳	۲۳،۰	۲۱،۰	۲۳،۵
۹	۹	۲۵	۵	۲۲،۹	۳۰،۳	۲۴،۹	۲۰،۸	۲۴،۷
۱۰	۱۵	۱۰	۵	۳۶،۷	۲۲،۲	۲۰،۲	۲۴،۸	۲۵،۹
۱۱	۱۵	۲۵	۵	۵۰،۷	۴۸،۸	۵۴،۰	۶۸،۹	۵۵،۶
۱۲	۱۵	۳۰	۵	۵۱،۹	۶۲،۵	۵۶،۲	۷۲،۷	۶۰،۸
۱۳	۲۵	۲۰	۵	۱۲۷،۸	۱۴۶،۱	۲۱۹،۹	۹۹،۶	۱۴۸،۳
۱۴	۲۵	۳۰	۵	۲۱۱،۱	۱۵۲،۴	۱۶۰،۲	۱۸۷،۶	۱۷۷،۸
۱۵	۲۵	۳۵	۵	۲۲۸،۰	۱۷۹،۶	۱۸۶،۹	۲۰۶،۰	۲۰۰،۱
۱۶	۴۰	۱۰	۵	۳۱۰،۱	۲۱۸،۷	۱۷۸،۲	۱۹۱،۶	۲۲۴،۷
۱۷	۴۰	۲۰	۵	۶۲۳،۶	۴۸۲،۵	۳۵۳،۰	۵۲۱،۵	۴۹۵،۱
۱۸	۴۰	۴۵	۵	۱۲۱۱،۰	۹۰۵،۷	۱۱۱۵،۷	۹۹۶،۷	۱۰۵۷،۳
۱۹	۵۰	۲۰	۵	۸۷۱،۱	۷۵۶،۴	۵۹۲،۵	۶۹۲،۷	۷۲۸،۲
۲۰	۵۰	۵۰	۵	۱۶۷۹،۹	۱۸۹۵،۶	۱۳۱۶،۴	۱۵۹۸،۶	۱۶۲۲،۷
میانگین کل				۲۷۵،۰	۲۴۸،۷	۲۱۷،۵	۲۳۸،۲	۲۴۴،۹

نتیجه و ارایه پیشنهادهای پژوهشهای آتی

در این مقاله مدلی براساس الگوریتمهای ژنتیک بمنظور زمانبندی مسئله Permutation flow shop شامل n کار و m ماشین با معیار $\sum E/T$ ارائه و جوابهای حاصل به لحاظ بهینگی و زمان دستیابی به جواب نهایی ارزیابی شدند.

این مدل برای مسائل تا ۹ کار و ۲۵ ماشین در ۹۷٪ حالات به جواب بهینه دست یافته است. همچنین جوابهای این مدل با جوابهای یکی از مدلهای سنتی در حل ۴۰۰ مسئله مختلف مقایسه شده است.

توسعه روش های تجربی سازنده بر اساس خواص برجستگی مسئله یکی از محورهای مهم تحقیقات آتی میباشد. همچنین بهبود عملکرد الگوریتم ژنتیک و مقایسه‌ی آن با دیگر روشهای بهبود دهنده زمینه مناسبی برای پژوهشهای بعدی فراهم میسازد.

مراجع

[1] S.M. Johnson, (1954), Optimal two-and three-stage production schedules, Naval Research logistic Quarterly, 1, 61-68.

[2] Dudek, R. Q., S. S. Panwalker, and M. L. Smith (1992), The lessons of flow shop scheduling research, Operation Research 40, 7-13.

[3] Ignall, E., and L. E. Scharge (1965) Application of the branch and bound technique to some flow shop scheduling problems, Operations Research 13 ,400-412.

[4] Smith , R. D., and R. A. Dudek (1967), A general algorithm for solution of the n-job m-machine sequencing problem of the flow shop. Operations Research, 15, 71-82 and Errata 17, 756 (1969).

[5] Ashour, S. (1970), An experimental investigation and comparative evaluation of flow shop scheduling techniques, Operations research 18, 541-548.

[6] Palmer, D.S. (1965), Sequencing Jobs through a multi stage process in the minimum total time- A quick method ob obtaini9ng a Near Optimum. Operational Research Quarterly, 16, 101-107.

[7] Campbell, H. G., R. A. Dudek, and M. L. Smith (1970), A heuristic algorithm for the n job m machine sequencing problem. Management Science 16, 630-637.

[8] Dannenbring, D. G. (1977), An evaluation of flow shop sequencing heuristics. Management Science 23, 1174-1182.

[9] Zegordy, S. H., and K. Itoh,, T. Enkawa, (1995), Minimizing makespan for flow shop scheduling by combining simulated annealing with sequencing knowledge. European Journal of Operational research 85, 515-531.

[10] Nawaz, M., Enscore, E.E., and Ham. I. (1988), A heuristic algorithm for the m-machine, n-job flow shop sequencing problem. OMEGA, 11, 91-95.

[11] Ogbu, F.A. and Smith D.K. (1990), The application of the simulated annealing algorithm to the solution of the n/m/Cmax flow shop problem. Computers and Operations Research, 17, 243-253

[12] Ow, P. S., and Morton, T.E., (1989), The single machine Early/Tardy problem, Management Science, 35, 177-191.