زمانبندی در محیط جریان کاری منعطف با محدودیت انجام همزمان چند کار

محمد علی سنکوکه
دانشجوی کارشناسی ارشد رشته صنایع دانشگاه تربیت مدرس
(sankokeh@modares.ac.ir)
محمد رضا امین ناصری
استادیار دانشگاه تربیت مدرس
(amin-nas@modares.ac.ir)
دانشگاه تربیت مدرس- تهران- تقاطع بزرگراه جلال آل احمد و شهید چمران

واژه های کلیدی

زمانبندی- جریان کاری منعطف- توالی عملیات- الگوریتم ابتکاری

چکیده

این مقاله به بررسی مساله زمانبندی در محیط جریان کاری منعطف[1] با محدودیت انجام همزمان چند کار روی یک ماشین می‌پردازد. در ادبیات موضوع تاکنون به این مساله پرداخته نشده است. این مساله در مواردی که با ایستگاههایی چون کوره اعم از کوره های پخت و عملیات حرارتی، سندبلاست[2]، شات بلاست[3] و ... که در آنها چند کار بطور همزمان می‌توانند روی یک ماشین مورد پردازش قرار بگیرند، کاربرد دارد. ابتدا نشان می‌دهیم مساله از نوع NP-Hard است و سپس سه الگوریتم ابتکاری بمنظور حل مساله و یک کران پایین بمنظور مقایسه الگوریتمها توسعه داده می‌شود. الگوریتمهای ابتکاری ارائه شده بر پایه الگوریتمهای مربوط به مساله ماشینهای موازی[4] ، تئوری محدودیتها[5] و قاعده جانسون[6] استوار هستند. در انتها نیز به مقایسه الگوریتمهای ارائه شده با یکدیگر پرداخته ایم. تعداد زیادی از مسائل که بصورت تصادفی ایجاد شده اند، توسط این سه الگوریتم حل شده و نتایج آنها با کران پایین توسعه داده شده مقایسه گردیده است. نتایج نشاندهنده برتری نسبی الگوریتم مبتنی بر مساله ماشینهای موازی است.

[1] **Flexible Flow Shop**
[2] **Sand blast**
[3] **Shot blast**
[4] **Parallel machines**
[5] **Theory of constraint**
[6] **Johnson rule**

۱- مقدمه

این مقاله به بررسی زمانبندی چند کار روی چند ماشین در محیط جریان کاری منعطف با محدودیت انجام همزمان چند کار روی یک ماشین می‌پردازد. امروزه در اکثر محیطهای تولیدی با ایستگاههایی مواجه می‌شویم که در آنها چند کار بطور همزمان می‌توانند روی یک ماشین مورد پردازش قرار بگیرند. ایستگاههایی مثل کوره اعم از کوره های پخت و عملیات حرارتی، سند بلاست، شات بلاست ، ریخته گری (قالبهای خوشه‌ای) و ... از جمله ایستگاههایی هستند که در آنها چند کار بطور همزمان می‌توانند روی یک ماشین مورد پردازش قرار بگیرند. همچنین در مواردی که نمی‌توان از زمان حمل و نقل بین ایستگاهها صرفنظر کرد و این حمل و نقل بصورت دسته‌ای صورت می‌پذیرد، می‌توان حمل و نقل را بصورت ایستگاهی در نظر گرفت که در آن چند کار بطور همزمان می‌توانند روی یک ماشین مورد پردازش قرار بگیرند.
Linn & Zhang [1] طی مقاله‌ای کارهایی که تا سال ۱۹۹۹ روی این مساله انجام شده بود را طبقه بندی کردند. اغلب این تحقیقات در زمینه مساله جریان کاری منعطف، بدون در نظر گرفتن هر گونه محدودیتی بود. بعد از این سال تحقیقات کمی روی مساله جریان کاری منعطف خالص صورت پذیرفت [2,3,4,5] و در اکثر تحقیقات، مساله مذکور با در نظر گرفتن محدودیتهایی مورد بررسی قرار می‌گرفت. Botta[6] به بررسی مساله جریان کاری منعطف، با در نظر گرفتن محدودیتهای تقدمی[7] و شکاف زمانی[8] بین فعالیتها و تابع هدف کمینه کردن حداکثر تاخیرها[9] پرداخت. Sawik[7] یک روش حل دقیق برای مساله جریان کاری منعطف، با در نظر گرفتن بافرهای[10] میانی محدود در بین مراحل ارائه داد. Sawik[8] مجددا یک روش دقیق برای حل مساله جریان کاری منعطف، با در نظر گرفتن بافرهای میانی محدود ارائه داد که علاوه بر محدودیتهای مساله قبل فرض زمانبندی گروهی[11] را نیز همراه خود داشت. Gupta et al[9] به بررسی مساله جریان کاری منعطف با محدودیت قابل کنترل بودن زمانهای فعالیت و موعدهای تحویل قابل تعیین پرداختند. آنها فرض کردند که توالی کارها در تمام مراحل یکسان است. تابع هدف مساله آنها کمینه کردن جمع وزنی زمانهای تکمیل ، تاریخهای تحویل[12]، زمانهای زودکرد [13]و زمانهای دیر کرد [14]کارها بود.
Kurz & Askin[10] به بررسی مساله جریان کاری منعطف، با در نظر گرفتن زمانهای آماده سازی[15] وابسته به توالی[16] و مجاز بودن پرش کارها از برخی مراحل پرداختند. Kurz & Askin[11] مجددا به بررسی همین مساله پرداختند و پس از مدلسازی عدد صحیح مختلط [17]مساله مذکور و ارائه یک کران پایین، یک الگوریتم ژنتیک[18]، بنام RKGA، برای آن ارائه نمودند. Bertel & Billaut [12] به بررسی مساله جریان کاری منعطف، با تابع هدف کمینه کردن جمع وزنی تعداد کارهای دیرکرددار[19] پرداختند. آنها فرض کردند که محدودیت متفاوت بودن سرعت ماشینها و مجاز بودن برگشت مجدد کارها [20]به مراحل قبلی نیز وجود دارد. Logendran & Carson [13] به بررسی مساله جریان کاری منعطف، با در نظر گرفتن زمانبندی گروهی و تابع هدف کمینه کردن حداکثر زمانهای تکمیل پرداختند. Oguz et al [14] به بررسی مساله جریان کاری منعطف با تابع هدف کمینه کردن حداکثر زمانهای تکمیل و این محدودیت که برای هر عملیات در هر مرحله چند ماشین مورد نیاز است، پرداختند. kyparisis & koulamas[15] به بررسی مساله جریان کاری منعطف، با تابع هدف کمینه کردن حداکثر زمان تکمیل کارها و با در نظر گرفتن محدودیت متفاوت بودن سرعت ماشینها پرداختند. Low[16] به حل مساله جریان کاری منعطف، با تابع هدف

---

[7] **Precedence constraints**
[8] **Time lag**
[9] **Lateness**
[10] **Buffer**
[11] **Group scheduling**
[12] **Due dates**
[13] **Earliness**
[14] **Tardiness**
[15] **Setup times**
[16] **Sequence dependent**
[17] **Mixed integer programming**
[18] **Genetic algorithm**
[19] **Tardy jobs**
[20] **Recirculation**

کمینه کردن جمع زمان جریان[21] کارها توسط الگوریتم شبیه سازی بازپخت[22] پرداخت. وی فرض کرده بود که زمانهای آماده سازی مستقل از توالی بوده ولی زمانهای تخلیه[23] وابسته به توالی هستند. مساله مورد بررسی در این مقاله تاکنون در ادبیات موضوع مورد بررسی قرار نگرفته است.

در ادامه این مقاله، در بخش دو به تعریف مساله می‌پردازیم. در بخش سوم به ارائه روشهای مختلف برای حل مساله مورد نظر پرداخته می‌شود. در بخش چهارم یک کران پایین برای مساله توسعه داده می‌شود. بخش پنجم به بیان نتایج محاسباتی بدست آمده از مقایسه و بررسی روشهای مختلف حل مساله اختصاص یافته است و در بخش پایانی نتیجه گیری و زمینه های مختلف برای کارهای آتی ارائه می‌گردد.

۲- تعریف مساله

مساله مورد نظر با زمانبندی در محیط جریان کاری منعطف سر و کار دارد. در این محیط فرض می‌شود که تعداد n کار وجود دارند که باید در m مرحله روی آنها پردازش انجام پذیرد. در هر مرحله مانند مرحله j تعداد l_j ماشین وجود دارد. مسیر انجام پردازش برای تمام کارها یکسان می‌باشد و هر کار باید از تمام مراحل بگذرد و در هر مرحله تنها باید روی یکی از ماشینها پردازش شود. همچنین زمان آماده بودن[24] ، برای تمام کارها برابر صفر فرض می‌شود.

در مساله مورد بررسی در این مقاله علاوه بر فرضیات محیط جریان کاری منعطف فرض می‌شود که ماشینهای هر مرحله کاملا مشابه هم بوده و فقط ممکن است از لحاظ سرعت انجام کار با یکدیگر تفاوت داشته باشند. همچنین فرض می‌شود که نسبت تفاوت سرعت ماشینها برای تمام کارها یکسان است. یعنی برای هر کار i داریم:

$$p_{ijk} = \frac{p_{ij}}{v_{jk}}$$

که p_{ij} بیانگر زمان پردازش کار i در مرحله j، v_{jk} بیانگر سرعت ماشین k در مرحله j و p_{ijk} بیانگر زمان پردازش کار i در مرحله j در صورتی که روی ماشین k مورد پردازش قرار گیرد، هستند. علاوه بر این فرض می‌شود که در برخی از مراحل ماشینها می‌توانند چند کار را همزمان مورد پردازش قرار دهند. در این مراحل ویژه ظرفیت ماشینها ممکن است متفاوت باشد (این ظرفیت برای ایستگاههای معمولی برابر یک فرض می‌شود). تابع هدف مساله کمینه کردن حداکثر زمانهای تکمیل کارها در مرحله آخر یا کمینه کردن $c_{\max}$ است.

در ادامه به تبیین برخی از نمادهای بکار رفته در این مقاله می‌پردازیم:

n: تعداد کارها

m: تعداد مراحل

i: شاخص کار، i=1,2,…,n

j: شاخص مرحله، j=1,2,…,m

lj: تعداد ماشینها در مرحله j

k: شاخص ماشین، $\forall j, k = 1,2,\dots,l_j$

v_{jk}: سرعت ماشین k ام در مرحله j

[21] **Flow time**
[22] **Simulated annealing**
[23] **Removal times**
[24] **Ready time**

p_{ij} : زمان پردازش کار i در مرحله j

p_{ijk} : زمان پردازش کار i در مرحله j، اگر روی ماشین kام در این مرحله مورد پردازش قرار بگیرد.

u_{jk}: ظرفیت پذیرش کارها برای ماشین kام در مرحله j

c_{ji}: زمان تکمیل مجازی کار i در مرحله j

$c_{\max}$: حداکثر زمانهای تکمیل کارها در مرحله آخر

$d_{j,k,b}$: در مرحله j بیانگر زمان تکمیل b امین دسته روی ماشین k می‌باشد.

در این مساله هر کاری که متعلق به یک دسته دلخواه می‌باشد، دارای دو زمان تکمیل مجازی و واقعی است. زمان تکمیل مجازی زمانی است که پردازش روی کار مزبور به اتمام می‌رسد. اما بعلت اینکه کار تنها زمانی آزاد می‌شود که پردازش تمام کارهای متعلق به آن دسته تمام شود، لذا یک زمان تکمیل دیگر به نام زمان تکمیل واقعی مطرح می‌شود.

Garey et al [17] نشان دادند که پیچیدگی مساله $F3//\ c_{\max}$ از نوع NP-Hard است و از آنجا که مساله مورد نظر ما تعمیم یافته مساله فوق الذکر است، لذا حداقل از نوع NP-Hard خواهد بود. در نتیجه پیدا کردن جواب بهینه در زمان معقول برای مساله غیر ممکن است و لذا باید از راه حلهای ابتکاری برای حل مساله استفاده نمود. همین نتیجه را می‌توان از NP-Hard بودن پیچیدگی مساله $P//c_{\max}$ که توسط Garey & Johnson [18] بیان شده است، بدست آورد، زیرا مساله ما به نوعی تعمیمی از مساله ماشینهای موازی نیز است. در این مقاله برای حل این مساله سه الگوریتم ابتکاری ارائه شده است.

۳- توسعه الگوریتمهای ابتکاری برای حل مساله

برای حل مساله می‌توان مساله را به دو زیر مساله تعیین توالی کارها در هر مرحله و تخصیص کارها بر اساس توالی بدست آمده، به ماشینهای آن مرحله، ویا بعبارت دیگر تعیین توالی کارها در هر مرحله و زمانبندی بر اساس توالی بدست آمده تقسیم نمود. ما در تمامی الگوریتمهای ارائه شده در این مقاله از رویه های زمانبندی پیشرو[۲۵] و یا پسرو[۲۶] ارائه شده توسط kyparisis & koulamas [14] استفاده می‌کنیم.

۳-۱- الگوریتم ابتکاری اول

این الگوریتم، تعمیم الگوریتم جانسون برای زمانبندی در محیط جریان کاری است. این الگوریتم ابتدا برای هر مرحله یک شاخص را محاسبه می‌کند که برابر مجموع حاصل ضرب سرعت در ظرفیت ماشینهای آن مرحله است.

$$\rho(j) = \sum_{k=1}^{l_j} v_{jk} * u_{jk}, \forall j = 1,2,\dots,m$$

در هر مرحله فرض می‌شود که با یک مساله زمانبندی با دو ماشین مجازی سر و کار داریم. در هر مرحله زمانهای پردازش روی ماشین مجازی اول و دوم برای کار i، که i=1,2,…, n به ترتیب با vp1(i) و vp2(i) نامگذاری می‌شوند. این زمانهای مجازی برای مرحله‌ای چون j از روابط زیر بدست می‌آیند:

[25] Forward
[26] Backward

$$vp1(i)=\sum_{s=1}^{j}\frac{p_{is}}{\rho(s)}$$
$$vp2(i)=\sum_{s=(m-j+1)}^{m}\frac{p_{is}}{\rho(s)}$$

سپس توالی کارها در هر مرحله (بجز مرحله آخر) بر اساس قاعده جانسون و زمانهای پردازش محاسبه شده روی دو ماشین مجازی بدست می-آید. توالی کارها در مرحله آخر مانند توالی کارها در مرحله ماقبل آخر فرض می‌شود.
در نهایت الگوریتم به انجام زمانبندی بر اساس توالیهای بدست آمده و زمانهای پردازش واقعی می‌پردازد. نحوه تخصیص کارها به ماشینها به قسمی است که در هر مرحله کار به ماشینی تخصیص می‌یابد که زودترین زمان تکمیل را برای آن کار در آن مرحله، مستقل از سایر کارها، بوجود آورد. این امر با توجه به زمان آماده بودن ماشینها در آن مرحله بدست می‌آید.

۳-۲- الگوریتم ابتکاری دوم

الگوریتم ابتکاری دوم، که از ایده مربوط به تئوری محدودیتها الهام گرفته شده است، ابتدا برای هر مرحله شاخص ρ را براساس رابطه‌ای که در الگوریتم قبل به آن اشاره شد، محاسبه می‌کند. سپس برای هر مرحله شاخص جدیدی محاسبه می‌کند و بر اساس آن ایستگاه گلوگاه را شناسایی می‌کند. نحوه محاسبه این شاخص جدید برای مرحله‌ای مانند j که با indicator(j) نشان داده می‌شود، بصورت زیر است:

$$Indicator(j)=\sum_{i=1}^{n}\frac{p_{ij}}{\rho(j)},\ \forall j=1,2,\ldots,m$$

هر ایستگاهی که دارای بزرگترین شاخص بود، بعنوان ایستگاه گلوگاه در نظر گرفته می‌شود. سپس با شروع از ایستگاه ما قبل ایستگاه گلوگاه زمانبندی رو به عقب صورت می‌پذیرد. بدین منظور برای هر کار یک زمان پردازش مجازی که برابرمجموع زمانهای پردازش کار مربوطه در ایستگاه گلوگاه و ایستگاههای بعد از آن است، محاسبه می‌گردد. بر اساس این زمانهای پردازش مجازی و بر مبنای قاعده SPT [27] یک توالی پایه[۲۸] برای ایستگاه ماقبل ایستگاه گلوگاه، بمنظور زمانبندی رو به عقب، بدست می‌آید.
در زمانبندی رو به عقب، کارها بترتیب توالی پایه و با شروع از یک عدد بزرگ مانند T، بصورت رو به عقب، به ماشینی اختصاص می‌یابند که، مستقل از سایر کارها، دیرترین زمان شروع را برای آنها موجب شود. توالی پایه برای سایر ایستگاههای ما قبل ایستگاه گلوگاه، با مرتب کردن کارها بر اساس زمان شروعشان در مرحله بعدی و بر مبنای قاعده LPT[۲۹] بدست می‌آید. یعنی هر چه زمان شروع کار در مرحله‌ای بیشتر باشد، از اولویت بیشتری در توالی پایه برخوردار است.
پس از اتمام زمانبندی رو به عقب، به زمانبندی رو به جلو پرداخته می‌شود که دقیقا بصورت فرایند زمانبندی در الگوریتمهای ابتکاری قبلی است.توالی کارها برای ایستگاهی خاص از ایستگاههای قبل از ایستگاه گلوگاه، بر اساس مرتب کردن کارها به ترتیب صعودی از زمان شروعشان در آن ایستگاه، که توسط فرایند زمانبندی رو به عقب تعیین شده‌اند، بدست می‌آید. این توالی برای ایستگاه گلوگاه و ایستگاههای بعد از آن، بترتیب صعودی از زمان تکمیل کارها در ایستگاه بلافاصله قبل از آنها، بدست می‌آید.

۳-۳- الگوریتم ابتکاری سوم

---
[27] Shortest Procees Time
[28] Basic Sequence
[29] Latest Process Time

این الگوریتم مساله را به m مساله ماشینهای موازی تبدیل می‌کند. این الگوریتم ابتدا برای هر مانند مرحله j شاخص ρ(j) را مانند دو الگوریتم قبل بدست می‌آورد و بر اساس آن برای هر کار یک زمان پردازش مجازی را، از طریق رابطه $\widetilde{p}_i = \sum_{j=1}^{m} \frac{p_{ij}}{\rho(j)}$ ، محاسبه می‌کند.
در ادامه الگوریتم، توالی کارها در مرحله اول را بر مبنای قاعده SPT و با استفاده از زمانهای پروسه مجازی تعیین می‌کند. الگوریتم سپس بر مبنای توالی بدست آمده به انجام زمانبندی می‌پردازد. توالی کارها برای سایر مراحل بدین ترتیب است که کاری در اولویت قرار می‌گیرد که زمان تکمیل کمتری را در مرحله قبل داشته باشد. زمانبندی مانند سایر الگوریتمها و برای تمام مراحل با توجه به زمانهای حقیقی پردازشها صورت می‌پذیرد.

۴- توسعه یک کران پایین

از آنجاییکه مدل مطرح شده در این مقاله تا کنون در ادبیات موضوع مطرح نشده است و از طرفی بعلت NP-Hard بودن مساله پیدا کردن جواب بهینه در زمان معقول عملا غیر ممکن است، لذا مبنایی جهت مقایسه کارایی الگوریتمهای ارائه شده در دست نیست. در نتیجه برای مقایسه کارایی الگوریتمها، داشتن یک کران پایین می‌تواند مبنای مقایسه خوبی باشد. لذا در این بخش یک کران پایین بدین منظور توسعه داده شده است که ذیلا به بیان آن می‌پردازیم.

$$LB = \max_{\substack{\forall i=1,..,n \\ \forall j=1,...,m}} \{lb(j), , lb'(i)\}$$

که

$$lb(j) = \sum_{h=1}^{j-1} \frac{\min_{i=1,..,n}\{p_{ih}\}}{\max_{k=1,..,l_h}\{v_{hk}\}} + \max\left\{ \frac{\sum_{i=1}^{n} p_{ij}}{\sum_{k=1}^{l_j} v_{jk} * u_{jk}}, \frac{\max_{i=1,..,n}\{p_{ij}\}}{\max_{k=1,..,l_j}\{v_{jk}\}} \right\} + \sum_{h=j+1}^{m} \frac{\min_{i=1,..,n}\{p_{ih}\}}{\max_{k=1,..,l_h}\{v_{hk}\}}, \forall j = 1,.., m$$

و

$$lb'(i) = \sum_{j=1}^{m} \frac{p_{ij}}{\max_{k=1,..,l_j}\{v_{jk}\}}, \forall\, i = 1,.., n$$

برای اثبات کران پایین ارائه شده ابتدا به بیان چند لم و اثبات آنها می‌پردازیم. در تمام حالات فرض می‌شود که زمانبندیها بصورت نیمه فعال[۳۰] هستند و کارها در زودترین زمان ممکن به ماشینها تخصیص می‌یابند.

لم ۱– فرض کنید یک ماشین با ظرفیت u موجود است و باید n کار روی آن مورد پردازش قرار بگیرند. همچنین فرض کنید s یک زمانبندی دلخواه روی این محیط و با تابع هدف کمینه کردن حداکثر زمان تکمیل کارها باشد.

[30] **Semiactive**

حال فرض کنید با محیط دیگری که یک محیط ماشینهای موازی با u ماشین، که ظرفیت همه آنها برابر یک است، سروکار داریم. آنگاه به ازای هر زمانبندی s روی محیط اول، حداقل یک زمانبندی s' در محیط دوم وجود دارد که مجموعه نامساویهای زیر در مورد آنها برقرار است.

$$c_i' \le c_i, \forall i = 1,\ldots,n$$

که c_i بیانگر زمان تکمیل مجازی کار iام تحت زمانبندی s، در محیط اول، و c_i' بیانگر زمان تکمیل واقعی کارiام تحت زمانبندی s'، در محیط دوم، است.

برهان: در محیط اول و تحت زمانبندی s روابط زیر برقرارند:

(۱)

$$d_z \ge \max_{i\in B_z}\{c_i\}, z = 1,\ldots,\left[\frac{n}{u}\right]+1$$

(۲)

$$d_z \le st_w, \forall w \in B_{z+1}, z = 1,\ldots,\left[\frac{n}{u}\right]$$

که st_w بیانگر زمان شروع کار w ، c_i بیانگر زمان تکمیل کار i، B_z نشاندهنده مجموعه کارهایی که به دسته z ام اختصاص یافته‌اند و d_z بیانگر زمان تکمیل دسته z ام و یا بعبارت دیگر زمان تکمیل واقعی کارهایی که به دسته z ام اختصاص یافته‌اند، است.

از (۱) و (۲) نتیجه می‌شود که

$$\max_{i\in B_z}\{c_i\} \le st_w, \forall w \in B_{z+1}, z = 1,\ldots,\left[\frac{n}{u}\right]$$

یا

(۳)

$$c_i \le st_w, \forall i \in B_z, \forall w \in B_{z+1}, z = 1,\ldots,\left[\frac{n}{u}\right]$$

حال در محیط دوم فرض کنید زمانبندی s' را داریم که دارای این شرایط است:

اولا: هر کار تعلق یافته به هر دسته، بر اساس زمانبندی s، در زمانبندی s' فقط و فقط به یک ماشین تخصیص می‌یابد.

ثانیا: در زمانبندی s کاری که متعلق به دسته‌ای با اندیس کوچکتر است، در زمانبندی s' زودتر از کاری که متعلق به دسته‌ای با اندیس بزرگتر است مورد پردازش قرار می‌گیرد (به شرطی که در زمانبندی s' هر دو به یک ماشین تخصیص یابند و قرار باشد روی آن پردازش شوند).

در این حالت واضح است که دیگر لازم نیست تمام مجموعه محدودیتهای (۳) با هم برقرار باشند و چون تعداد محدودیتها کمتر می‌شود، لذا داریم:

$$st'_w \le st_w, \forall w \in B_{z+1}, z = 1,\ldots,\left[\frac{n}{u}\right]$$

یا

$$c'_w \leq c_w, \forall w \in B_{z+1}, z = 1,\ldots,\left[\frac{n}{u}\right]$$

که st'_w بیانگر زمان شروع کار w تحت زمانبندی s' است.

از طرفی برای مرحله اول نیز داریم $c'_w = c_w, \forall w \in B_1$ و لذا اثبات کامل است.

نتیجه– فر ض کنید در محیط اول با محیط ماشینهای موازی با L ماشین و با ظرفیتهای $u_1, u_2, \ldots, u_L$ سروکار داریم که باید n کار روی آنها پردازش شوند. همچنین فرض کنید s یک زمانبندی دلخواه روی این محیط و با تابع هدف کمینه کردن حداکثر زمان تکمیل کارها باشد. حال فرض کنید با محیط دیگری نیزکه یک محیط ماشینهای موازی است و دارای $\sum_{k=1,..,L} u_k$ ماشین، با ظرفیت برابر یک، است، سروکار داریم. آنگاه به ازای هر زمانبندی s روی محیط اول حداقل یک زمانبندی s' در محیط دوم وجود دارد که در آن داریم

$$c'_{\max} \leq c_{\max}$$

که $c_{\max}$ و $c'_{\max}$ به ترتیب بیانگر حداکثر زمان تکمیل کارها در زمانبندیهای Sو s' هستند.

لم ۲– فرض کنید با زمانبندی در یک محیط ماشینهای موازی که در آن L ماشین با ظرفیت برابر یک و سرعتهایی برابر با $v_1, v_2, \ldots, v_L$ وجود دارند، سروکار داریم که n کار باید به آنها تخصیص یابند. یک کران پایین برای تابع هدف بهینه کمینه کردن حداکثر زمان تکمیل کارها که آنرا با $c^*_{\max}$ نشان می‌دهیم، از رابطه زیر بدست می‌آید.

$$c^*_{\max} \geq \frac{\sum_{i=1}^{n} p_i}{\sum_{k=1}^{L} v_k}$$

که در آن p_i بیانگر زمان تکمیل کار i ام و v_k بیانگر سرعت ماشین k ام است.

برهان: فرض کنید که به یک زمانبندی مانند S دست یافته‌ایم که در آن هیچ یک از ماشینها زمان بیکاری نداشته و تمام ماشینها از ابتدا تا زمان تکمیل آخرین کار مشغول پردازش کارها باشند. در این حالت زمان آزاد شدن تمام ماشینها یکسان و برابر $c_{\max}$ می‌باشد. بدیهی است که $c_{\max}$ در این حالت کوچکتر یا مساوی با هر $c_{\max}$ دیگری، مربوط به هر زمان بندی دیگری غیر از زمانبندی S، می‌باشد و لذا ما آنرا با $c^*_{\max}$ نشان می‌دهیم. حال فرض کنید مجموع زمان پردازش کارهایی که به ماشین k ام اختصاص یافته‌اند را با Q_k نشان دهیم. در این صورت برای زمانبندی S روابط زیر برقرارند:

(۴)

$$Q_1 + Q_2 + \ldots + Q_L = \sum_{i=1}^{n} p_i$$

(۵)

$$\frac{Q_1}{v_1} = \frac{Q_2}{v_2} = \ldots = \frac{Q_L}{v_L} = c^*_{\max}$$

که v_k بیانگر سرعت ماشین kام است. از (۵) نتیجه می‌شود

(۶)

$$Q_k = \frac{Q_1}{v_1} * v_k, \forall k = 1,\ldots,L$$

با جایگذاری (۶) در (۴) داریم:

$$\sum_{k=1}^{L}\frac{Q_1}{v_1}*v_k=\frac{Q_1}{v_1}*\sum_{k=1}^{L}v_k=\sum_{i=1}^{n}p_i \Rightarrow \frac{Q_1}{v_1}=\frac{\sum_{i=1}^{n}p_i}{\sum_{k=1}^{L}v_k}$$

که با توجه به (۵) داریم:

$$c^*_{\max}=\frac{\sum_{i=1}^{n}p_i}{\sum_{k=1}^{L}v_k}$$

که اثبات کامل است.

لم ۳: فرض کنید با یک محیط ماشینهای موازی با L ماشین و با ظرفیتهای $u_1,u_2,\dots,u_L$ و سرعتهای $v_1,v_2,\dots,v_L$ سروکار داریم که باید n کار روی آنها پردازش شوند. یک کران پایین برای تابع هدف بهینه کمینه کردن حداکثر زمان تکمیل کارها، که آنرا با $c^*_{\max}$ نمایش می‌دهیم، بصورت زیر بدست می‌آید.

$$c^*_{\max}\ge\frac{\sum_{i=1}^{n}p_i}{\sum_{k=1}^{L}u_k*v_k}$$

برهان: فرض کنید برای مساله اصلی یک زمانبندی مانند s داریم که دارای تابع هدف بهینه $c^*_{\max}$ است. طبق نتیجه لم ۱ اگر فرض کنیم u_1 ماشین با سرعت v_1، u_2 ماشین با سرعت v_2 و بهمین ترتیب u_L ماشین با سرعت v_L داریم، آنگاه حداقل یک زمانبندی مانند s'' وجود دارد که بر اساس آن $c''_{\max}\le c^*_{\max}$ که $c''_{\max}$ مقدار تابع هدف مربوط به زمانبندی s''، مربوط به محیط جدید، است.
از طرفی بر اساس لم ۲ به ازای هر زمانبندی s'' داریم:

$$c^*_{\max}\ge c''_{\max}\ge\frac{\sum_{i=1}^{n}p_i}{\sum_{k=1}^{L}u_k*v_k}\Rightarrow c^*_{\max}\ge\frac{\sum_{i=1}^{n}p_i}{\sum_{k=1}^{L}u_k*v_k}$$

که اثبات کامل است.

قضیه: یک محیط جریان کاری منعطف با m مرحله که در مرحله‌ای مانند j تعداد l_j ماشین وجود داشته باشد (j=1,…,m) را در نظر بگیرید. فرض کنید سرعت ماشین kام در مرحله j ام برابر v_{jk} و ظرفیت آن برابر u_{jk} باشد. اگر زمان پردازش کار i در مرحله j برابر p_{ij} باشد آنگاه یک کران پایین، که آنرا با LB نشان می‌دهیم، برای زمانبندی با تابع هدف کمینه کردن حداکثر زمان تکمیل کارها، از رابطه زیر بدست می‌آید.

$$LB=\max_{\substack{\forall i=1,..,n\\ \forall j=1,...,m}}\{lb(j),,lb'(i)\}$$

که

(۷)

$$lb(j)=\sum_{h=1}^{j-1}\frac{\min_{i=1,..,n}\{p_{ih}\}}{\max_{k=1,..,l_h}\{v_{hk}\}}+\max\left\{\frac{\sum_{i=1}^{n}p_{ij}}{\sum_{k=1}^{l_j}v_{jk}*u_{jk}},\frac{\max_{i=1,..,n}\{p_{ij}\}}{\max_{k=1,..,l_j}\{v_{jk}\}}\right\}+\sum_{h=j+1}^{m}\frac{\min_{i=1,..,n}\{p_{ih}\}}{\max_{k=1,..,l_h}\{v_{hk}\}},\forall j=1,..,m$$

و

(۸)

$$lb'(i) = \sum_{j=1}^{m} \frac{p_{ij}}{\max\limits_{k=1,..,l_j} \{v_{jk}\}}, \forall\, i = 1,..,n$$

برهان: فرض کنید در مرحله‌ای مانند j قرار داریم. اولین کاری که در این مرحله تکمیل می‌شود را k1 و آخرین کار را kn می‌نامیم. اولین جمله از سمت چپ معادله مربوط به lb(j) یک کران پایین برای حداقل زمان ممکنی است که طول می‌کشد تا کار k1 به مرحله j برسد. جمله دوم آن بیانگر یک کران پایین برای زمان تکمیل کار kn، در مرحله j ام، است. اولین عبارت این جمله بر اساس لم ۳ اثبات شد و دومین عبارت آن نیز بدیهی است. جمله سوم معادله مربوط به lb(j) نیز بیانگر یک کران پایین برای مدت زمانی است که طول می‌کشد تا کار kn، پس از خروج از مرحله jام، کاملا تکمیل شود. در نتیجه مجموع این سه جمله یک کران پایین برای تابع هدف کمینه کردن حداکثر زمان تکمیل کارها را می دهد. اما چون می‌توان به جای مرحله j هر مرحله دلخواه دیگر را در نظر گرفت، لذا m کران پایین بدست می‌آید.
اما هر یک از مجموعه کرانهای (۸) بیانگر یک کران پایین برای مدت زمان تکمیل یک کار دلخواه می‌باشد و بدیهی است که حداکثر زمان تکمیل کارها یا $C_{\max}$، بزرگتر یا مساوی زمان تکمیل هر یک از کارها است. در نتیجه n کران پایین دیگر نیز بدست می‌آید. لذا تعداد (m+n) کران پایین برای مساله داریم و بدیهی است که بزرگترین آنها، بعنوان بهترین کران پایین برای مساله انتخاب شود.

۵- نتایج محاسباتی

در بخش نتایج محاسباتی ابتدا مسائل مختلف با اندازه های مختلف را مورد بررسی قرار می‌دهیم. شش پارامتر از مساله را انتخاب کرده و برای هر کدام از پارامترها سطوحی چون بالا، پایین و متوسط را بصورتی که در جدول ۱ نشان داده شده است را تعریف کرده‌ایم.
در مورد ظرفیت ماشینها در هر مرحله، فرض شده است که در هر ایستگاه با احتمال ۰٫۷ تمام ماشینها دارای ظرفیت برابر یک و با احتمال ۰٫۳ دارای ظرفیتهای متفاوت هستند، که این ظرفیتهای متفاوت بر اساس متغیرهای تصادفی معرفی شده در جدول یک، مشخص می‌گردند.

پارامتر مساله	پایین	متوسط	بالا
تعداد کارها	۱۰	۵۰	۱۰۰
تعداد ایستگاهها	۲	۱۰	۲۰
تعداد ماشینها در هر مرحله	۱	U[1,4]	U[2,8]
زمانهای پردازش	U[12,18]		U[8,22]
سرعت ماشینها		U[1,4]	
ظرفیت ماشینها	U[1,5]		U[5,10]

جدول ۱- پارامترهای مختلف برای مسائل مختلف

تعداد ترکیبات مختلفی که توسط مقادیر مختلف این ۶ پارامتر می‌توان بدست آورد برابر ۱۰۸(۲*۱*۲*۳*۳*۳) نوع مساله است. از هر نوع مساله یک مساله را ایجاد کردیم و هر یک از الگوریتمهای ارائه شده را به ازای تمام مسائل ایجاد شده، اجرا کردیم. کلیه برنامه های کامپیوتری این تحقیق توسط زبان برنامه نویسی Visual Basic6 نوشته شده و توسط یک کامپیوتر P IV با پردازنده ۲۴۰۰ مگا هرتز اجرا شده‌اند.
نتایج بدست آمده از اجرای این سه الگوریتم ابتکاری در جدول ۲ نشان داده شده است. معیار انحراف نسبی برای هر یک از جوابها، از رابطه زیر بدست می‌آید.

$$\text{انحراف نسبی} = \frac{\text{(کران پایین} - \text{جواب بدست آمده از الگوریتم ابتکاری)}}{\text{کران پایین}}$$

الگوریتم	میانگین جوابها	بالاترین تابع هدف	پایین ترین تابع هدف	میانگین زمان حل (ثانیه)	میانگین انحراف نسبی
الگوریتم ابتکاری اول	۱۱۴۳,۴۲۵۹	۷۴۷۶,۱۷	۱۱,۵	۰,۰۱۸۵۱۹	۱,۰۰۵
الگوریتم ابتکاری دوم	۶۰۷,۲۱	۲۹۱۶,۴۲	۱۲,۵	۰,۰۲۷۷۷۸	۰,۲۸۲
الگوریتم ابتکاری سوم	۵۸۰,۹	۲۲۴۹,۷۵	۱۲,۲۵	۰,۰۱۸۵۱۹	۰,۲۴۸

جدول ۲-نتایج بدست آمده از اجرای سه الگوریتم ابتکاری

نتایج نشان می‌دهند که الگوریتم ابتکاری، سوم از نظر معیارهای میانگین جوابها، میانگین انحراف نسبی و حداکثر جوابهای بدست آمده، از حل مسائل مختلف، نتایج بهتری را نسبت به سایر الگوریتمها می‌دهد. این در حالی است که الگوریتم ابتکاری اول از لحاظ این معیارها بدترین نتایج را می‌دهد. همچنین الگوریتم ابتکاری اول از لحاظ معیار حداقل زمان حل نتایج نسبتا بهتری را نشان می‌دهد. شاید این امر نشاندهنده این واقعیت باشد که این الگوریتم برای مسائل با اندازه کوچک نتایج نسبتا بهتری را بدهد. از نظر معیار میانگین زمانهای حل، الگوریتمهای ارائه شده تفاوت زیادی نسبت به هم ندارند.

۶-خلاصه و نتیجه گیری

در این مقاله به حل مساله زمانبندی در محیط جریان کاری منعطف با محدودیت قابلیت انجام همزمان چند کار توسط ماشینها و متفاوت بودن سرعت ماشینها با تابع هدف کمینه کردن حداکثر زمان تکمیل کارها پرداخته شد. ایستگاههایی مثل کوره اعم از کوره های پخت و عملیات حرارتی، سند بلاست، شات بلاست، ریخته گری (قالبهای خوشه‌ای) و ... از جمله ایستگاههایی هستند که در آنها چند کار بطور همزمان می‌توانند روی یک ماشین مورد پردازش قرار بگیرند. ابتدا نشان داده شد که مساله مذکور حداقل از نوع NP-Hard است. برای حل مساله، آنرا به دو زیر مساله تعیین توالی کارها در هر مرحله و تخصیص کارها به ماشینها بر اساس توالی آن مرحله تقسیم کردیم. سپس سه الگوریتم ابتکاری الهام گرفته شده از قاعده جانسون، تئوری محدودیتها و الگوریتمهای زمانبندی محیط ماشینهای موازی ارائه گردید. بمنظور مقایسه کارایی الگوریتمها کران پایینی برای مساله مذکور توسعه داده شد. هر یک از الگوریتمها روی تعداد زیادی از مسائل، که بصورت تصادفی ایجاد شده بودند، اجرا شدند که نتایج محاسباتی بدست آمده نشاندهنده برتری الگوریتم ابتکاری سوم، از نظر معیارهای میانگین جوابها، میانگین انحراف نسبی از کران پایین و بزرگترین مقدار برای تابع هدف، و برتری الگوریتم ابتکاری اول، از نظر معیار کمترین مقدار تابع هدف، نسبت به سایر الگوریتمها بود.

ارائه الگوریتمهای ابتکاری و فوق ابتکاری[31] مختلف که مساله اصلی را به دو زیر مساله تعیین توالی کارها و تخصیص کارها به ماشینها تقسیم می‌کنند، می‌تواند بعنوان زمینه‌ای برای انجام تحقیقات بیشتر روی این مساله در نظر گرفته شود. همچنین اضافه کردن محدودیتهایی چون وجود حالت مونتاژ و یا ترکیب محیطهای زمانبندی مختلف، که باعث عملی ترشدن و نزدیکتر شدن به مسائل زمانبندی واقعی شوند، می‌تواند بعنوان زمینه‌ای دیگر برای تحقیقات آتی در نظر گرفته شود.

مراجع

1- Linn, R., Zhang, W., Hybrid flow shop scheduling: a survey, Computers & Industrial Engineering 37, p57-61, 1999.

2- Engin, O., Döyen, A., A new approach to solve hybrid flow shop scheduling problems by artificial immune system, Future Generation Computer Systems 20, p1083–1095, 2004.

3- Neron, E., Baptisteb, P., Guptac J.N.D., Solving hybrid flow shop problem using energetic reasoninand global operationsg, Omega 29, p501–511, 2001.

4- Acero-Domínguez, M.J., Paternina-Arboleda, C.D., Scheduling jobs on a K-stage flexible flow shop using a TOC-based (bottleneck) procedure, Systems and Information Engineering Design Symposium, p295-298, 2004.

5- Kyparisis G. J., Koulamas, C., A note on weighted completion time minimization in a fexible flow shop, Operations Research Letters 29, p5–11, 2001.

6- Botta-Genoulaz, V., Hybrid fow shop scheduling with precedence constraints and time lags to minimize maximum lateness, Int. J. Production Economics 64, p101-111, 2000.

7- Sawik, T., Mixed integer programming for scheduling flexible flow lines with limited intermediate buffers, Mathematical and Computer Modelling 31, p39-52, 2000.

8- Sawik, T., An exact approach for batch scheduling in flexible flow lines with limited intermediate buffers, Mathematical and Computer hlodelling 36, p461-471, 2002.

9- Gupta, J.N.D., KruK, K., Lau, V.,Werner, F, Sotskov, Y. N., Heuristics for hybrid flow shops with controllable processing times and assignable due dates, Computers & Operations Research 29, 1417-1439, 2002.

10- Kurz, M. E., Askin, R. G, Comparing scheduling rules for flexible flow lines, Int. J. Production Economics 85, p371–388, 2003.

11- Kurz, M. E., Askin, R. G.,Scheduling flexible flow lines with sequence-dependent setup times, European Journal of Operational Research 159, p66–82, 2004.

12- Bertel, S., Billaut, J.C., A genetic algorithm for an industrial multiprocessor flow shop scheduling problem with recirculation, European Journal of Operational Research, xxx, p1–12, 2003.

[31] **Meta heuristic**

13- Logendran R., Carson ,S., Hanson, E., Group scheduling in flexible flow shops, Int. J. Production Economics xx, p1-13, 2004.

14- Oguz, C., Zinder, Y., Do, V. H., Janiak, A., Lichtenstein,M., Hybrid flow shop scheduling problems with multiprocessor task systems, European Journal of Operational Research 152, p115–131, 2004.

15- Kyparisis G.J., Koulamas, C., Flexible flow shop scheduling with uniform parallel machines, European Journal of Operational Research xxx, p1-13, 2004.

16- Low, C., Simulated annealing heuristic for flow shop schedulinproblems with unrelated parallel machines, Computers & Operations Researchxx, p1-13, 2004.

17- Garey, M.R., Johnson, D.S., Sethi, R., The complexity of flowshop and jobshop scheduling, Math. Oper. Res. 1, p117–129, 1976.

18- Garey, M.R., Johnson, D.S., Strongly NP-completeness results: motivation examples and implication, Journal of Asociation for Computing Machinery 25, p499–508, 1978.