فصلنامه علمی – پژوهشی مطالعات مدیریت صنعتی – سال پانزدهم، شماره ۴۷، زمستان ۹۶
صفحات ۱۵۱-۱۶۷

مدل‌سازی مساله زمانبندی تولید جریان کارگاهی چند حالته با منابع محدود

مهدی یزدانی*، بهمن نادری**

تاریخ دریافت:۹۴/۱۰/۱۱ - تاریخ پذیرش:۹۵/۱۲/۲۲

چکیده

در مسائل زمانبندی معمولا زمان پردازش عملیات‌های هر کار مشخص و ثابت در نظر گرفته می‌شود. در ادبیات مسائل زمانبندی پروژه بسیار تاکید شده است که زمان هر فعالیت/عملیات می‌تواند گاهی چند حالته باشد و با تخصیص مقداری بیشتری از منابع به یک فعالیت، زمان پردازش آن نیز کاهش یابد. در اینگونه مسائل علاوه بر زمانبندی فعالیت‌ها باید تخصیص منابع محدود در دسترس به فعالیت‌ها نیز انجام شود. این ضعف در ادبیات مسائل زمانبندی وجود دارد که زمان پردازش فعالیت‌ها ثابت فرض می‌شود. در این مقاله، مسئله جریان کارگاهی از حالت کلاسیک خود به مسئله جریان کارگاهی چند حالته با منابع محدود توسعه داده می‌شود. این مقاله به طور جامع در مورد مدلسازی ریاضی این مسئله بحث می‌کند. در این راستا دو مدل ریاضی به فرم برنامه‌ریزی ریاضی عدد صحیح مختلط خطی با دو مفهوم مختلف ارائه می‌شود. مدل اول، مکان محور و مدل دوم توالی محور است. برای ارزیابی عملکرد این دو مدل، پیچیدگی اندازه و پیچیدگی محاسباتی آنها تعیین و مقایسه می‌شود. در شاخص پیچیدگی اندازه، مدل اول تعداد متغیرهای بیشتر اما تعداد محدودیت‌های کمتری در مقایسه با مدل دوم نیاز دارد. در شاخص پیچیدگی محاسباتی، مدل اول عملکرد کاملا بهتری از مدل دوم ارائه می‌کند. همچنین مدل اول علاوه بر حل تعداد بیشتری از مسائل به صورت بهینه، زمان کمتری نیز برای حل در مقایسه با مدل دوم احتیاج دارد.

واژگان کلیدی: زمان‌بندی جریان کارگاهی، چند حالته، مدل‌سازی ریاضی، برنامه‌ریزی عددصحیح مختلط، پیچیدگی اندازه و محاسباتی.

* استادیار، گروه مهندسی صنایع، دانشکده مهندسی صنایع و مکانیک، واحد قزوین، دانشگاه آزاد اسلامی، قزوین، ایران
(نویسنده مسئول) m_yazdani@qiau.ac.ir
** دانشیار، گروه مهندسی صنایع، دانشکده فنی و مهندسی، دانشگاه خوارزمی، تهران، ایران

مقدمه

یک مسئله زمانبندی را بدین صورت می‌توان تعریف کرد. یک مجموعه از کارها وجود دارد که برای تکمیل آنها باید تعدادی عملیات روی آنها انجام شود. هر عملیات در یک ایستگاه کاری انجام می‌شود که در هر ایستگاه تنها یک پردازشگر (که اصطلاحا ماشین گفته می‌شود) وجود دارد. هدف تعیین توالی انجام کارها در ایستگاه‌های کاری به نحوی است که یک یا چند تابع هدف که همگی معمولا زمان محور هستند بهینه شوند(پیندو[1]، ۲۰۰۸).

بر اساس چگونگی مسیر پردازش کارها در ایستگاه‌ها مسئله‌های زمانبندی مختلفی تعریف می‌شود. منظور از مسیر پردازش، ترتیب عملیات‌های یک کار (یا به عبارتی ترتیب ایستگاه‌های کاری که یک کار باید ملاقات کند) است. اگر مسیر پردازش برای همه کارها مشخص و یکسان باشد بدین صورت که ابتدا در ایستگاه کاری اول، سپس در ایستگاه کاری دوم و در نهایت در ایستگاه کاری آخر پردازش شوند، به مسئله جریان کارگاهی[2] گفته می‌شود. اگر مسیر پردازش برای هر کار مشخص و متفاوت از سایر کارها باشد، مسئله کارکارگاهی[3] گفته می‌شود. در صورتی که مسیر پردازش کارها از قبل مشخص نباشد و باید توسط برنامه‌ریز مشخص شود، مسئله کارگاه باز[4] گفته می‌شود.

ازجمله فرضیات کلاسیک مسائل زمانبندی می‌توان به این موارد اشاره کرد. کارها مستقل از هم هستند و هیچ روابط پیشنیازی بین آنها وجود ندارد. همه کارها در زمان صفر در دسترس هستند. یک کار در یک زمان تنها در یک ایستگاه کاری می‌تواند باشد در نتیجه حداکثر می‌توان یک عملیات روی آن انجام شود. زمان پردازش عملیات‌های یک کار مشخص است. از سوی دیگر ماشین موجود در هر ایستگاه کاری همیشه در دسترس هستند و هیچ خرابی اتفاق نمی‌افتد. یک ماشین تنها قابلیت پردازش یک کار در یک زمان را دارد. در نتیجه یک ماشین نمی‌تواند چند کار را همزمان پردازش کند. زمان آماده‌سازی و همچنین زمان حمل

1- Pinedo
2- Flow shop
3- Job shop
4- Open shop

ونقل بین ایستگاه‌های کاری ناچیز است و می‌توان از آنها صرف‌نظر کرد.

یکی از فرضیاتی که در مسائل زمانبندی معمولا در نظر گرفته می‌شود مشخص و ثابت بودن زمان پردازش عملیات‌های هر کار است. این ضعف به طور کلی در ادبیات مسائل زمانبندی وجود دارد. برای نمونه می‌توان به مقاله‌های زندیه و همکاران[۱] (۲۰۰۶)، یزدانی و همکاران[۲] (۲۰۱۰) و لی و همکاران[۳] (۲۰۱۴) اشاره کرد. در حالیکه در ادبیات مسائل زمانبندی پروژه بسیار تاکید شده است زمان پردازش هر فعالیت/عملیات می‌تواند چند حالته[۴] باشد. در حقیقت با تخصیص مقداری بیشتری از منابع به یک فعالیت، زمان پردازش آن نیز کاهش می‌یابد. البته مقدار کل منبع در دسترس محدود است. در اینگونه مسائل هدف علاوه بر زمانبندی فعالیت‌ها، تخصیص منابع محدود در دسترس به فعالیت‌ها نیز است. دراین بین می‌توان به مقاله‌های وب و ویت[۵] (۲۰۰۷)، کریاکریس و همکاران[۶] (۲۰۱۲)، لین و همکاران[۷] (۲۰۱۳)، افشارنجفی[۸] (۲۰۱۴)، چنگ و همکاران[۹] (۲۰۱۴)، و ونگ و همکاران[۱۰] (۲۰۱۱) اشاره کرد. همچنین گارسیا[۱۱] (۲۰۱۶) و چنگ و همکاران[۱۲] (۲۰۱۲) مثالهای کاربردی از زمانبندی با فرض منابع محدود ارائه کرده اند.

در این مقاله، مسئله جریان کارگاهی از حالت کلاسیک خود به مسئله جریان کارگاهی چند حالته با منابع محدود توسعه داده می‌شود. این مقاله به طور جامع در مورد مدلسازی ریاضی مسئله بحث می‌کند. در این راستا دو مدل ریاضی به فرم برنامه‌ریزی ریاضی عدد صحیح مختلط خطی[۱۳] ارائه می‌شود. این دو مدل از نظر مفهومی با یکدیگر متفاوت هستند. مدل اول،

1- Zandieh et al.
2- Yazdani et al.
3- Li et al.
4- Multi-mode
5-Voß and Witt
6- Kyriakidis et al.
7- Lin et al.
8- Afshar-Nadjafi
9- Cheng et al.
10- Wang et al.
11- Garcia
12- Cheng et al.
13- Mixed integer linear programming

مکان محور[1] و مدل دوم توالی محور[2] است. عملکرد این دو مدل باهم مقایسه و ارزیابی می‌شوند. برای ارزیابی عملکرد مدل‌های برنامه‌ریزی ریاضی، پیچیدگی اندازه[3] و پیچیدگی محاسباتی[4] مدل‌ها بررسی می‌شود. برای محاسبه پیچیدگی اندازه، تعداد متغیرهای باینری[5] و پیوسته[6] و همچنین تعداد محدودیت‌ها[7] محاسبه می‌شود. برای محاسبه پیچیدگی محاسباتی، زمان مورد نیاز برای حل توسط مدل ارزیابی می‌شود. با استفاده از نرم‌افزار سیپلکس[8] مدل ریاضی حل می‌شود (۲۰۰۸).

ادامه مقاله بدین صورت سازماندهی شده است. در بخش دوم ابتدا مسئله جریان کارگاهی چندحالته با منابع محدود تعریف و مدل‌های ریاضی آن ارائه می‌شود. در بخش سوم، مدل‌های ریاضی ارزیابی و مقایسه می‌شوند. در نهایت، در بخش چهارم، مقاله جمع‌بندی و نتیجه‌گیری می‌شود.

تعریف و مدل‌سازی مسئله

در این بخش ابتدا مسئله جریان کارگاهی چندحالته با منابع محدود تعریف می‌شود. سپس دو مدل برنامه‌ریزی ریاضی برای مسئله توسعه داده می‌شود. در این مسئله، یک مجموعه n تایی از کارها وجود دارد که برای تکمیل آنها باید m فعالیت/علمیات روی هر یک انجام شود. هر عملیات در یک ایستگاه کاری انجام می‌شود که در هر ایستگاه یک پردازشگر وجود دارد. در نتیجه در یک کارگاه m پردازشگر وجود دارد. تمامی کارها مسیر پردازش یکسانی دارند. به عبارتی ابتدا توسط پردازشگر اول، سپس پردازشگر دوم و به همین ترتیب تا آخرین پردازشگر باید پردازش شوند.

1- Position-based
2- Sequence-based
3- Size complexity
4- Computational complexity
5- Binary variable
6- Continuous variable
7- Constraint
8- CPLEX

زمان پردازش هر کار در هر ایستگاه ثابت نیست بلکه وابسته به مقدار منبع تجدید ناپذیر تخصیص داده شده به آن است. هر کار در چند حالت می‌تواند انجام شود که اساس این حالت‌ها بدین صورت است که با افزایش مقدار هر نوع منبع تخصیص داده شده به یک کار، زمان پردازش فعالیت‌های آن نیز کاهش می‌یابد. تمام فعالیت‌های یک کار باید در یک حالت انجام شود. علاوه بر آن، یک فعالیت آماده‌سازی قبل از انجام هر فعالیت تولیدی باید انجام شود. هدف در این مسئله تخصیص منابع موجود به فعالیت‌ها و همچنین پیدا کردن توالی انجام آنها به نحوی است که زمان تکمیل آخرین فعالیت (اصطلاحا میکسپن[1]) مینیمم شود. همچنین لازم به ذکر است که توالی انجام کارها روی همه ایستگاه‌ها یکسان است. بدین معنی که توالی تعیین شده برای پردازش کارها در ایستگاه اول تا آخر ثابت خواهد ماند.

در ادامه دو مدل ریاضی برای مسئله توسعه داده می‌شود. مدل اول مکان محور است. بدین معنی که متغیرهای تصمیم مکان هر کار در توالی را مشخص می‌کنند. مدل دوم نیز توالی محور است. بدین معنی که متغیرهای تصمیم توالی نسبی کارها را نسبت به هم مشخص می‌کنند. قبل از ارائه مدل ریاضی، پارامترها و نمادهای زیر تعریف شده تا در مدل استفاده شود.

n	تعداد کارها
m	تعداد ایستگاه‌های کاری
r	تعداد انواع منابع استفاده شده
g	شمارنده منابع
j, k	شمارنده کارها
i	شمارنده ایستگاه‌های کاری
o_j	تعداد حالت‌های انجام کار jام
l	شمارنده حالت‌ها
$p_{j,i,l}$	زمان پردازش کار jدر ایستگاه کار i در حالت l

1- Makespan

$s_{j,i,l}$	زمان آماده‌سازی کار j در ایستگاه کار i در حالت l
$a_{j,l,g}$	مقدار منبع نوع g مورد نیاز برای اجرای کار j در حالت l
b_g	مقدار منبع نوع g در دسترس
M	یک عدد مثبت بزرگ

همچنین متغیرهای تصمیم تعریف شده در مدل اول نیز به صورت زیر است.

$X_{j,k,l}$	متغیر باینری که عدد یک می‌گیرد اگر کار j در مکان k توالی در حالت l پردازش شود، در غیر اینصورت عدد صفر خواهد داشت.
$C_{k,i}$	متغیرپیوسته برای زمان تکمیل کار در مکان k در ایستگاه i
C_{max}	متغیرپیوسته برای زمان تکمیل آخرین فعالیت

مدل ریاضی اول به صورت زیر است:

Min C_{max}

Subject to:

$$\sum_{k=1}^{n}\sum_{l=1}^{o_j} X_{j,k,l} = 1 \qquad \forall_j \qquad (۱)$$

$$\sum_{j=1}^{n}\sum_{l=1}^{o_j} X_{j,k,l} = 1 \qquad \forall_k \qquad (۲)$$

$$\sum_{j=1}^{n}\sum_{k=1}^{n}\sum_{l=1}^{o_j} \left(X_{j,k,l} \cdot a_{j,l,g}\right) \le b_g \qquad \forall_g \qquad (۳)$$

$$C_{k,i} \ge C_{k,i-1} + \sum_{j=1}^{n}\sum_{l=1}^{o_j} X_{j,k,l} \cdot \left(s_{j,i,l} + p_{j,i,l}\right) \qquad \forall_{k,i} \qquad (۴)$$

$$C_{k,i} \geq C_{k-1,i} + \sum_{j=1}^{n}\sum_{l=1}^{o_j} X_{j,k,l} \cdot (s_{j,i,l} + p_{j,i,l}) \qquad \forall_{k>1,i} \qquad (۵)$$

$$C_{max} \geq C_{n,m} \qquad (۶)$$

$$C_{k,i} \geq 0 \qquad \forall_{k,i} \qquad (۷)$$

$$X_{j,k,l} \in \{0,1\} \qquad \forall_{j,k,l} \qquad (۸)$$

جایی که $C_{k,0} = 0$.

مجموعه محدودیت اول، برای تعیین مکان پردازش در توالی و حالت انجام هر کار است. باید توجه داشت که اگر n کار داشته باشیم، n مکان در توالی خواهیم داشت و هدف این مجموعه محدودیت تعیین مکان هر کار در توالی است. علاوه بر مکان، باید حالت/مد انجام آن کار نیز مشخص شود. مجموعه محدودیت دوم برای اطمینان از این موضوع است که در یک مکان از توالی دقیقا یک کار وجود داشته باشد. مجموعه محدودیت سوم برای اطمینان از رعایت محدودیت مصرف برای هر نوع از منابع است تا انتخاب حالت انجام کارها، به صورتی باشد که مجموع مصرف یک نوع منبع مورد نیاز بیشتر از کل منبع در دسترس برای آن نباشد. مجموعه محدودیت چهارم مشخص می‌کند که یک کار در یک زمان حداکثر می‌تواند در یک ایستگاه پردازش شود. در نتیجه تا پردازش آن در یک مرحله تکمیل نشده باشد امکان پردازش آن در مرحله بعدی وجود ندارد. پس زمان تکمیل بین دو مرحله متوالی همیشه از زمان پردازش آن کار بیشتر است.

مجموعه محدودیت پنجم نشان می‌دهد که هر ایستگاه در یک زمان می‌تواند حداکثر یک کار را پردازش کند. در نتیجه اختلاف بین زمان تکمیل دو کار متفاوت در یک مرحله باید حداقل برابر با زمان پردازش کار قبلی باشد. مجموعه محدودیت ششم برای محاسبه زمان تکمیل آخرین فعالیت است. این زمان تکمیل قاعدتا برابر با زمان تکمیل آخرین عملیات کاری است که در مکان آخر در توالی قرار دارد. مجموعه محدودیت‌های هفتم و هشتم برای تعریف متغیرهای تصمیم مدل است.

متغیرهای تصمیم مدل دوم نیز به صورت زیر است:

$X_{k,j}$ متغیر باینری که عدد یک می‌گیرد اگر کار j بعد از کار k پردازش شود، در غیر اینصورت عدد صفر خواهد داشت.

$Y_{j,l}$ متغیر باینری که عدد یک می‌گیرد اگر کار j در حالت l پردازش شود، در غیر اینصورت عدد صفر خواهد داشت.

$C_{j,i}$ متغیرپیوسته برای زمان تکمیل کار j در ایستگاه i

C_{max} متغیرپیوسته برای زمان تکمیل آخرین فعالیت

مدل ریاضی دوم به صورت زیر است.

Min C_{max}

Subject to:

$$\sum_{l=1}^{o_j} Y_{j,l} = 1 \qquad \forall_j \qquad (۹)$$

$$\sum_{j=1}^{n}\sum_{l=1}^{o_j}\left(Y_{j,l}\cdot a_{j,l,g}\right) \le b_g \qquad \forall_g \qquad (۱۰)$$

$$C_{j,i} \ge C_{j,i-1} + \sum_{l=1}^{o_j} Y_{j,l}\cdot\left(s_{j,i,l} + p_{j,i,l}\right) \qquad \forall_{j,i} \qquad (۱۱)$$

$$C_{j,i} \ge C_{k,i} + \sum_{l=1}^{o_j} Y_{j,l}\cdot\left(s_{j,i,l} + p_{j,i,l}\right) - M\cdot\left(1 - X_{k,j}\right) \qquad \forall_{j<n,k>j,i} \qquad (۱۲)$$

$$C_{k,i} \ge C_{j,i} + \sum_{l=1}^{o_k} Y_{k,l}\cdot\left(s_{j,i,l} + p_{j,i,l}\right) - M\cdot X_{k,j} \qquad \forall_{j<n,k>j,i} \qquad (۱۳)$$

$$C_{max} \ge C_{j,m} \qquad \forall_j \qquad (۱۴)$$

$$C_{j,i} \geq 0 \qquad \forall_{j,i} \qquad (۱۵)$$

$$X_{k,j} \in \{0,1\} \qquad \forall_{j<n,k>j} \qquad (۱۶)$$

$$Y_{j,l} \in \{0,1\} \qquad \forall_{j,l} \qquad (۱۷)$$

جایی که $C_{j,0} = 0$.

مجموعه محدودیت نهم، برای تعیین حالت انجام هر کار است. مجموعه محدودیت دهم برای اطمینان از رعایت محدودیت منابع است (مثل مجموع محدودیت سوم در مدل اول). مجموعه محدودیت یازدهم مشخص می‌کند که یک کار در یک زمان حداکثر می‌تواند در یک ایستگاه پردازش شود (مثل مجموع محدویت چهارم در مدل اول). مجموعه محدودیت‌های دوازدهم و سیزدهم با یکدیگر نشان می‌دهند که هر ایستگاه در یک زمان می‌تواند حداکثر یک کار را پردازش کند (مثل مجموعه محدودیت پنجم در مدل اول). مجموعه محدودیت چهاردهم برای محاسبه زمان تکمیل آخرین فعالیت است (مثل مجموع محدودیت ششم در مدل اول). مجموعه محدودیت‌های پانزدهم، شانزدهم و هفدهم برای تعریف متغیرهای تصمیم مدل است.

ارزیابی عملکرد مدل‌ها

در این بخش، عملکرد دو مدل ریاضی ارائه شده در بخش قبل ارزیابی و مقایسه می‌شوند. برای ارزیابی عملکرد مدل‌های برنامه‌ریزی ریاضی، پیچیدگی اندازه و پیچیدگی محاسباتی مدل‌ها بررسی می‌شوند. برای اطلاعات بیشتر می‌توان خوانندگان را به مقاله‌های استفورد و همکاران[1] (۲۰۰۵)، تسنگ و استفورد[2] (۲۰۰۸) و نادری و سلماسی[3] (۲۰۱۲) ارجاع داد.

برای محاسبه پیچیدگی اندازه، تعداد متغیرهای باینری و پیوسته و همچنین تعداد محدودیت‌ها محاسبه می‌شوند. برای محاسبه پیچیدگی محاسباتی، زمان مورد نیاز برای حل توسط مدل ارزیابی می‌شود (تسنگ و استفورد، ۲۰۰۸). با استفاده از نرم‌افزار سیپلکس[4] هر دو مدل

1- Stafford et al.
2- Teseng and Stafford
3- Naderi and Salmasi
4- Cplex

ریاضی حل می‌شود. در زیربخش بعدی، پیچیدگی اندازه دو مدل محاسبه می‌شود. در ادامه آن، پیچیدگی محاسباتی دو مدل ریاضی ارزیابی و مقایسه می‌شوند.

پیچیدگی اندازه

یکی از معیارهای ارزیابی مدل ریاضی، پیچیدگی اندازه آن است. برای تعیین پیچیدگی اندازه، تعداد متغیرهای باینری و پیوسته و همچنین تعداد محدویت‌های مورد نیاز برای مدلسازی یک مسئله محاسبه می‌شود. برای شمردن تعداد متغیرها و محدودیت‌ها، یک مسئله با n کار، m ایستگاه، r نوع منبع و هر کار با h حالت اجرا فرض می‌شود.

مدل اول برای چنین مسئله نیاز به n^2h متغیرباینری، nm متغیر پیوسته و $2n+r+2nm-m+1$ محدودیت دارد. در صورتی که مدل دوم برای همین مسئله نیاز به $(n^2-n)/2+nh$ متغیرباینری، nm متغیر پیوسته و $2n+r+n^2m$ محدودیت دارد. جدول ۱ نتایج را نشان می‌دهد.

جدول ۱. پیچیدگی اندازه دو مدل ریاضی

مشخصه	مدل اول	مدل دوم
تعداد متغیرهای باینری	n^2h	$(n^2-n)/2+nh$
تعداد متغیرهای پیوسته	nm	nm
تعداد محدودیت‌ها	$2n+r+2nm-m+1$	$2n+r+n^2m$

از آنجایی که تعداد متغیرهای باینری وابستگی درجه دو به تعداد کارها و وابستگی خطی به تعداد محدودیت و حالت دارد، به نظر می‌رسد پیچیدگی مدل‌ها بیشتر وابسته به تعداد کار است. نکته دیگری که در این ارزیابی نباید از آن غافل شد، تعداد محدودیت‌هایی است که در آنها M بزرگ وجود دارد. ارزیابی وجود M در مدل‌های برنامه‌ریزی عددصحیح از اهمیت بالایی برخوردار است. مدل‌هایی که در آنها M وجود دارد معمولا کیفیت پایین‌تری دارند زیرا باعث آزادسازی خطی مدل ضعیف می‌شوند. در نتیجه باعث بالارفتن پیچیدگی محاسباتی می‌شوند (نادری و سلماسی، ۲۰۱۲). مدل اول از این حیث بهتر از مدل دوم است زیرا اصلا

محدودیتی که در آن از M استفاده شده باشد ندارد در صورتی که در مدل دوم در $(n^2\text{-}n)m$ عدد از محدودیت‌ها (مجموعه محدودیت‌های دوازدهم و سیزدهم) از M استفاده می‌شود.

پیچیدگی محاسباتی مدل‌ها

یکی دیگر از شاخص‌های مهم برای ارزیابی مدل‌های برنامه‌ریزی ریاضی، تخمین زمان محاسباتی مدل برای حل یک مسئله در یک نرم‌افزار تخصصی برنامه‌ریزی ریاضی است. بدیهی است هر مدلی که زمان محاسباتی کمتری نیاز داشته باشد، مدل بهتری محسوب می‌شود. لازم به یادآوری است که این شاخص پیچیدگی محاسباتی مکمل شاخص پیچیدگی اندازه است. باید دقت کرد مدلی که پیچیدگی اندازه کمتری دارد لزوما پیچیدگی محاسباتی کمتری ندارد. زیرا ممکن است با اضافه کردن محدودیت و یا متغیرهای کمکی بتوان پیچیدگی محاسباتی را کم کرد اما طبیعتا پیچیدگی اندازه بزرگتر می‌شود. همچنین همان‌طور که پیش‌تر مطرح شد وجود M بزرگ نیز تاثیرگذار است.

برای ارزیابی و مقایسه دو مدل پیشنهادی، مدل‌ها در نرم‌افزار سیپلکس[1] کد و با کامپیوتری با 2GB RAM و CPU Core2due حل می‌شوند. یک مجموعه مثال عددی شامل ۱۶ مثال از مسئله در اندازه‌های مختلف تولید می‌شود تا توسط مدل‌ها حل شوند. در این مثال‌ها تعداد کارها را برابر با ۵، ۱۰، ۱۵ و ۲۰ در نظر می‌گیریم. تعداد ایستگاه‌ها برابر با ۴ و ۶ خواهد بود. همچنین منابع استفاده شده نیز از ۱ نوع در نظر می‌گرفته می‌شود. از هر ترکیب تعداد کار و ماشین نیز ۲ مثال عددی تولید می‌شود. لازم به ذکر است که زمان‌های پردازش و آماده‌سازی به ترتیب از یک توزیع یکنواخت بین ۱ تا ۷۰ و ۰ تا ۱۹ تولید می‌شوند. تعداد حالت نیز در تمامی مسائل ۲ در نظر گرفته می‌شود. تعداد منابع مورد نیاز برای حالت اول ۱ و برای حالت دوم ۲ است. همچنین زمان پردازش حالت دوم نیز بین ۱۰٪ الی ۶۰٪ کمتر از زمان پردازش حالت اول لحاظ می‌شود. حداکثر تعداد منابع نیز بین ۲۰٪ تا ۴۰٪ بیشتر از تعداد کارها است. در نتیجه مجموعا ۱۶ مثال تولید می‌شود. برای هر مثال حداکثر ۳۶۰۰ ثانیه به

1- CPLEX

مدل‌ها زمان داده می‌شود.

جدول ۲ نتایج حاصل از حل این مثال‌ها در نرم‌افزار توسط هر دو مدل را نشان می‌دهد. اطلاعات این جدول شامل موارد زیر است.

- زمان مورد نیاز برای مثال‌هایی که در کمتر از ۳۶۰۰ ثانیه به صورت بهینه حل شده‌اند.
- شکاف بهینگی (فاصله بین حد پایین و بالا) برای مثال‌هایی که در ۳۶۰۰ ثانیه به صورت بهینه حل نشده‌اند.
- تعداد گره‌هایی که الگوریتم شاخه و کران نرم‌افزار جستجو کرده است.

مدل اول ۱۴ مثال را به صورت بهینه حل نمود در صورتی که مدل دوم ۷ عدد از مثال‌ها را به صورت بهینه حل می‌کند. در اصل، به نظر می‌رسد مدل دوم مسائل با اندازه بزرگی ۱۰ کار را می‌تواند حل کند و مسائل بزرگتر از این اندازه را نمی‌تواند به صورت بهینه حل کند. متوسط شکاف بهینگی مدل دوم برای اندازه ۱۵ کار برابر است ۴۱٪. مدل اول برای حل مسائل با اندازه ۱۰ کار و کمتر زمانی زیر ۸ ثانیه احتیاج دارند در صورتی که مدل دوم فقط مسائل با اندازه ۵ کار را در زیر ۸ ثانیه حل می‌کند. مدل اول برای حل مسائل با اندازه ۱۵ کار، زمانی کمتر از ۲۰۰ ثانیه به طور متوسط احتیاج دارد. در صورتی که مدل دوم در ۳۶۰۰ ثانیه قادر به یافتن جواب بهینه نیست. لازم به ذکر است که هیچ کدام مسائلی با سایز ۲۰ کار و ۶ ماشین را حل نکردند.

همچنین برای درک بهتر پیچیدگی اندازه مدل‌ها، برای مثال‌های این بخش با فرض دو حالت و یک نوع منبع تعداد متغیرها و محدودیت‌ها در جدول ۳ ارائه شده است. همان‌طور که نتایج نشان می‌دهد مدل اول تعداد متغیر باینری بیشتر اما تعداد محدودیت کمتری نیاز دارد، در صورتی که مدل دوم تعداد متغیر باینری کمتر اما تعداد محدودیت بیشتری دارد. از آنجایی که تعداد متغیرهای پیوسته برای هر دو مدل برابر با nm است از ارائه آنها صرف‌نظر می‌شود.

جدول ۲. نتایج محاسبه پیچیدگی محاسباتی

مثال			مدل اول				مدل دوم			
n	m		زمان (ثانیه)	شکاف بهینگی	تابع هدف	تعداد گره	زمان (ثانیه)	شکاف بهینگی	تابع هدف	تعداد گره
۵	۴		۰/۰۸	٪۰	۳۵۵	۱۵	۰/۱۱	٪۰	۳۵۵	۳۷
۵	۴		۰/۰۵	٪۰	۲۴۹	۱۵	۰/۰۹	٪۰	۲۴۹	۲۷
۵	۶		۰/۱۹	٪۰	۵۲۰	۱۵۴	۰/۲۳	٪۰	۵۲۰	۱۵۴
۵	۶		۰/۰۹	٪۰	۴۱۵	۳۱	۰/۱۳	٪۰	۴۱۵	۶۶
۱۰	۴		۰/۴۸	٪۰	۵۱۱	۳۹۰	۲۶۰	٪۰	۵۱۱	۸۸۵۵۷۵
۱۰	۴		۱/۱۷	٪۰	۵۵۸	۶۰۷	۴۸۰	٪۰	۵۵۸	۱۲۰۴۹۳۸
۱۰	۶		۲/۶۷	٪۰	۷۰۳	۲۶۲۶	۶۵۲	٪۰	۷۰۳	۸۸۲۰۵۶
۱۰	۶		۷/۹۶	٪۰	۸۰۹	۱۰۴۸۷	۳۶۰۰	٪۱۱	۸۱۱	۴۰۳۲۰۰۳
۱۵	۴		۳/۲	٪۰	۷۹۲	۹۲۰	۳۶۰۰	٪۳۳	۷۹۴	۵۳۵۴۳۵۹
۱۵	۴		۴۳/۹	٪۰	۶۶۶	۶۲۵۳۰	۳۶۰۰	٪۴۵	۶۷۴	۲۶۱۵۵۵۷
۱۵	۶		۳۱۵	٪۰	۸۲۳	۳۶۷۰۹۵	۳۶۰۰	٪۴۳	۸۴۳	۱۱۱۰۲۰۲
۱۵	۶		۱۷۰	٪۰	۹۴۹	۱۵۵۷۹۱	۳۶۰۰	٪۴۴	۹۷۳	۱۰۲۷۱۳۸
۲۰	۴		۴/۴۱	٪۰	۱۰۰۳	۵۲۱	۳۶۰۰	٪۵۸	۱۰۰۵	۸۳۷۴۸۸
۲۰	۴		۱۰۴	٪۰	۱۰۷۴	۷۵۵۲۳	۳۶۰۰	٪۶۰	۱۰۷۴	۵۹۶۳۰۲
۲۰	۶		۳۶۰۰	٪۲	۱۱۴۸	۷۹۶۷۹۵	۳۶۰۰	٪۶۱	۱۱۸۳	۲۷۴۳۶۹
۲۰	۶		۳۶۰۰	٪۵	۱۱۵۶	۴۴۲۶۰۷	۳۶۰۰	٪۶۰	۱۲۱۳	۲۱۰۳۳۹

جدول ۳. نتایج عددی پیچیدگی اندازه

مثال		مدل اول			مدل دوم	
n	m	متغیر باینری	محدودیت‌ها		متغیر باینری	محدودیت‌ها
۵	۴	۵۰	۴۸		۲۰	۱۱۱
۵	۶	۵۰	۶۶		۲۰	۱۶۱
۱۰	۴	۲۰۰	۹۸		۶۵	۴۲۱
۱۰	۶	۲۰۰	۱۳۶		۶۵	۶۲۱
۱۵	۴	۴۵۰	۱۴۸		۱۳۵	۹۳۱

۱۵	۶	۴۵۰	۲۰۶		۱۳۵	۱۳۸۱
۲۰	۴	۸۰۰	۱۹۸		۲۳۰	۱۶۴۱
۲۰	۶	۸۰۰	۲۷۶		۲۳۰	۲۴۴۱

نتیجه‌گیری و مطالعات آتی

در این مقاله، مسئله جریان کارگاهی از حالت کلاسیک خود به مسئله جریان کارگاهی چند حالته با منابع محدود توسعه داده شد. تابع هدف این مسئله کمینه سازی زمان تکمیل آخرین کار است. معمولا در مسائل زمانبندی، زمان پردازش عملیات‌های هر کار مشخص و ثابت فرض می‌شود. در صورتی که در ادبیات زمانبندی پروژه بر این نکته تاکید شده است که زمان هر فعالیت/عملیات می‌تواند گاهی چند حالته باشد. بدین صورت که با تخصیص مقداری بیشتری از منابع به یک فعالیت، زمان پردازش آن نیز کاهش می‌یابد. در این دسته از مسائل باید تخصیص منابع محدود در دسترس به فعالیت‌ها علاوه بر زمانبندی فعالیت‌ها انجام شود. این شکاف تحقیقاتی در ادبیات مسائل زمانبندی وجود دارد که زمان پردازش فعالیت‌ها ثابت فرض می‌شود.

این مقاله به طور جامع مدلسازی ریاضی این مسئله را مطالعه می‌کند. در این راستا، دو مدل ریاضی به فرم برنامه‌ریزی ریاضی عدد صحیح مختلط خطی با دو مفهوم مختلف ارائه می‌شود. در مدل اول، از متغیرهایی که مکان هر کار در توالی را مشخص می‌کردند استفاده می‌شد در حالی که در مدل دوم متغیرهای تصمیم توالی کارها را مشخص می‌کردند. برای ارزیابی عملکرد این دو مدل، پیچیدگی اندازه و پیچیدگی محاسباتی آنها تعیین و مقایسه می‌شود. در شاخص پیچیدگی اندازه مدل اول تعداد متغیرهای بیشتر اما تعداد محدودیت‌های کمتری در مقایسه با مدل دوم نیاز دارد. در شاخص پیچیدگی محاسباتی، مدل اول عملکرد کاملا بهتری از مدل دوم ارائه می‌کند. علاوه بر حل تعداد بیشتری از مسائل به صورت بهینه، زمان کمتری نیز برای حل در مقایسه با مدل دوم احتیاج دارد. مدل اول، مسائل با اندازه ۵ کار را با زمانی در کمتر از ۱ ثانیه، مسائل با اندازه ۱۰ کار را با زمانی کمتر از ۱۰ ثانیه و در نهایت برخی مسائل با اندازه ۲۰ کار را با زمانی حدود ۱۰۰ ثانیه می‌تواند حل کند.

یکی از زمینه‌های جالب تحقیقاتی آتی، در نظر گرفتن تابع مطلوبیت[1] برای هرکار در هر حالت است. بدین ترتیب که جدا از محدویت منابع، حالت هر کار باید به نحوی تعیین شود که نه تنها زمان تکمیل آخرین فعالیت مینیمم شود بلکه مجموع مطلوبیت کارها نیز حداکثر شود. در نتیجه مسئله، یک تصمیم گیری دوهدفه خواهد شد. علاوه بر آن طراحی یک الگوریتم فرابتکاری نیز برای حل مسئله در اندازه‌های بزرگتر می تواند برای ادبیات مسئله مفید باشد.

بعد از ارائه مدل‌های ریاضی برای این مسأله، به نظر می‌رسد ارائه یک الگوریتم مبتنی بر تجزیه برای این مسأله امکان پذیر است زیرا مدل ریاضی آن قابل تفکیک به مسأله‌های کوچکتر است. در نتیجه ارائه یک الگوریتم تجزیه، به طور مثال الگوریتم تجزیه بندرز، می‌تواند یک تحقیق جذاب برای مسأله محسوب شود. همچنین، متاسفانه از آنجایی که مدل‌های ریاضی توسعه داده شده دارای تعداد زیادی محدودیت M بزرگ هستند، به عنوان یک تحقیق آتی دیگر می‌تواند روی ارائه مدل‌هایی که تعداد اینگونه محدودیت‌های آنها کمتر است، بحث و مطالعه انجام شود.

1- Utility function

منابع

Afshar-Nadjafi, B. (2014) A solution procedure for preemptive multi-mode project scheduling problem with mode changeability to resumption, *Applied Computing and Informatics*, DOI: 10.1016/j.aci.2014.02.003.

Cheng, J., J. Fowler, K. Kempf, S. Mason, (2014) Multi-mode resource-constrained project scheduling problems with non-preemptive activity splitting, *Computers and Operations Research*, DOI: 10.1016/j.cor.2014.04.018.

Cheng, J., Fowler, J., Kempf, K., (2012) Simulation-based multi-mode resource-constrained project scheduling of semiconductor equipment installation and qualification, *Proceedings of the Winter Simulation Conference*, 1-12.

Garcia, C., (2016) Resource-constrained scheduling with hard due windows and rejection penalties, *Engineering optimization*, 48(9), 1515-1528.

Kyriakidis, T.S., G.M. Kopanos, M.C. Georgiadis, (2012) MILP formulations for single- and multi-mode resource-constrained project scheduling problems, *Computers and Chemical Engineering*, 36, 369–385.

Lee C.H., Liao, C.J., Chung, T.P., (2014) Scheduling with multi-attribute setup times on two identical parallel machines, *International Journal of Production Economics*, 153, 130-138.

Lin, D., Lee, C.K.M., Ho, W. (2013) Multi-level genetic algorithm for the resource-constrained re-entrant scheduling problem in the flow shop, *Engineering Applications of Artificial Intelligence*, 26(4), 1282–1290.

Naderi, B., N. Salmasi, (2012) Permutation flowshops in group scheduling with sequence-dependent setup times, *European Journal of Industrial Engineering*, 6(2), 177-199.

Pinedo, M.L. (2008) Theory, algorithms, and systems, 3rd edn. Springer Science+Business Media, New York.

Stafford, E.F., F.T. Tseng, J.N.D. Gupta, (2005) Comparative evaluation of MILP flowshop models, *Journal of Operational Research Society*, 56, 88–101.

Tseng, F.T., E.F. Stafford, (2008) New MILP models for the permutation flowshop problem, *Journal of the Operational Research Society*, 59, 1373–1386.

Voß, S., Witt, A., (2007) Hybrid flow shop scheduling as a multi-mode multi-project scheduling problem with batching requirements: A real-world application, *International Journal of Production Economics*, 105(2), 445–458.

Wang, L., C. Fang, (2011) An effective shuffled frog-leaping algorithm for multi-mode resource-constrained project scheduling problem, *Information Sciences*, 181(20), 4804-4822.

Yazdani, M., M. Amiri, M. Zandieh, (2010) Flexible job-shop scheduling with parallel variable neighborhood search algorithm, *Expert Systems with Applications*, 37(1), 678-687.

Zandieh, M., S.M.T. Fatemi Ghomi, S.M. Moattar Husseini, (2006) An immune algorithm approach to hybrid flow shops scheduling with sequence-dependent setup times, *Applied Mathematics and Computation*, 180(1), 111-127.