Research in Production and Operations Management
University of Isfahan E-ISSN: 2981-0329
Vol. 15, Issue 1, No. 36, Spring 2024
https://doi.org/10.22108/POM.2024.136793.1500

(Research paper)

Scheduling a Batch Processing Machine in a Just-in-Time Production System Considering a Tight Due Date

Taha Keshavarz*
Department of Industrial Engineering, Faculty of Engineering, Semnan University, Semnan, Iran,
taha_keshavarz@semnan.ac.ir
Hamidreza Zarabadipour
Department of Industrial Engineering, Faculty of Engineering, Yazd University, Yazd, Iran,
hamidreza.zarabadipour@gmail.com

Purpose: The development and complexity of new markets, on the one hand, and economic constraints, on the other hand, have made it an inevitable necessity to pay attention to the two principles of providing a desirable and reliable level of service to customers and reducing supply and maintenance costs. Therefore, the need to study the methods that enable the production system to deal with these issues is felt more than ever. Just-In-Time production strategy has been mentioned as one of the appropriate approaches to balance between the two principles. On the other hand, the issue of sequencing and scheduling of operations in batch processing systems has been widely considered in the last two decades. A batch processing machine can process a batch of jobs simultaneously, which reduces the machine's set-up time and facilitates material flow management. This study aims to minimize the total weighted earliness and tardiness penalties of jobs with non-identical sizes on the batch processing machine, considering that the due date is tight.

Design/methodology/approach: Mathematical programming has been used to model the problem. A mixed integer linear programming model has been proposed for the research problem. Since the problem is shown to be NP-hard, heuristic and meta-heuristic methods have been developed to find near-optimal solutions for industrial-sized instances. Also, a dynamic programming approach has been proposed to find the optimal scheduling of a predetermined batch of jobs.

* Corresponding author, Orcid: 0000-0002-7343-9677

Findings: The dynamic programming algorithm requires a high computational effort, and the solution time by this algorithm increases significantly when the number of jobs increases. However, the obtained results indicated that the proposed heuristic algorithms lead to good performance with less time and in practice, such algorithms can be used for real applications and large-size instances. The average relative deviation of the proposed particle swarm algorithm is less than 1%, and the value of this index for the proposed heuristic algorithm is 1.78%.

Research implications: Examining the two investigated methods for batching the jobs, one based on a heuristic algorithm and the other with the help of solving a mathematical model, indicated no significant difference between these two methods. Therefore, if necessary, the heuristic algorithm with less computational effort can be used without losing the quality of the solution.

Practical implications: According to the findings, developing efficient heuristic and meta-heuristic algorithms for batch processing machine scheduling in just-in-time production systems can reduce production costs.

Originality/value: For the first the heuristic and meta-heuristic algorithms were proposed for the problem of scheduling a batch processing machine considering a tight due date in a just-in-time production system. A dynamic programming approach was also proposed for the first time to find the optimal scheduling of a predetermined batch of jobs.

Keywords: Batch processing machine, Just-In-Time, Tight due date, Dynamic programming, Heuristic algorithm, Particle swarm optimization

پژوهش در مدیریت تولید و عملیات، دوره ۱۵، شماره ۱، پیاپی ۳۶، بهار ۱۴۰۳
دریافت: ۱۴۰۱/۱۱/۲۳ پذیرش: ۱۴۰۲/۱۲/۲۹ ص ۸۹-۱۱۲
https://doi.org/10.22108/POM.2024.136793.1500

(مقالهٔ پژوهشی)

زمان‌بندی یک ماشین پردازش انباشته در راستای تحقق تولید بهنگام با در نظر گرفتن موعد تحویل نزدیک

طاها کشاورز[1*]، حمید رضا زرآبادی پور[2]

۱- استادیار گروه مهندسی صنایع، دانشکده فنی مهندسی، دانشگاه سمنان، سمنان، ایران، taha_keshavarz@semnan.ac.ir
۲- کارشناس ارشد گروه مهندسی صنایع، دانشکده فنی مهندسی، دانشگاه یزد، یزد، ایران، hamidreza.zarabadipour@gmail.com

چکیده: توسعه و پیچیدگی بازارهای جدید از یک‌سو و محدودیت‌های اقتصادی از سوی دیگر، سبب شده‌اند تا توجه به دو اصل ارائهٔ خدمات مطلوب و کاهش هزینه‌ها به ضرورتی اجتناب‌ناپذیر تبدیل شوند. نگرش تولید بهنگام، ازجمله رویکردهای مناسب برای موازنهٔ میان دو اصل یادشده است. همچنین، طی دو دههٔ اخیر، به موضوع تعیین توالی و زمان‌بندی عملیات در سیستم‌های تولید انباشته‌ای به‌طور وسیعی توجه شده است. دستگاه پردازش انباشته‌ای، هم‌زمان یک انباشته را از کارها پردازش می‌کند و این امر سبب کاهش زمان تنظیم دستگاه و همچنین تسهیل در امر مدیریت جریان مواد می‌شود. هدف پژوهش حاضر، کمینه‌سازی مجموع وزنی تعجیل و تأخیر کارهایی با اندازهٔ غیر یکسان بر ماشین پردازش انباشته، با لحاظ‌کردن موعد تحویل نزدیک به زمان شروع زمان‌بندی و در راستای تحقق تولید بهنگام است. در این تحقیق دو رویکرد برای انباشته‌سازی کارها، یکی مبتنی بر یک روش ابتکاری و دیگری مبتنی بر حل یک مدل ریاضی، بررسی شده است؛ سپس توالی انباشته‌ها به کمک یک الگوریتم برنامه‌ریزی پویا برای تحقق تولید بهنگام، تعیین شده است. همچنین یک الگوریتم ابتکاری و یک الگوریتم فراابتکاری بر مبنای الگوریتم ازدحام ذرات برای حل کامل مسئله ارائه شده است. نتایج محاسباتی حاکی از آن است که متوسط انحراف نسبی الگوریتم ازدحام ذرات پیشنهادی، کمتر از ۱درصد و مقدار این شاخص برای الگوریتم ابتکاری ارائه‌شده ۱/۷۸درصد است.

واژه‌های کلیدی: ماشین پردازش انباشته، تولید بهنگام، موعد تحویل نزدیک، برنامه‌ریزی پویا، الگوریتم ابتکاری، الگوریتم ازدحام ذرات

* نویسنده مسئول، Orcid: 0000-0002-7343-9677

۱- مقدمه

تاکنون بسیاری از محققان به‌علت کاربرد وسیع مسائل مربوط به زمان‌بندی ماشین‌های پردازش انباشته، به این موارد توجه کرده‌اند. از همین روی، تحقیقات پیشین در بر گیرندهٔ طیف گسترده‌ای از مطالب‌اند. در این بخش سعی شده است تا چشم‌اندازی از تلاش‌های انجام‌شده در جهت شناخت و مطالعه دربارهٔ این موضوع ارائه شود. به‌طور کلی بسیاری از پژوهش‌های انجام‌شده در این زمینه، مربوط به ارائهٔ الگوریتم‌هایی برای یافتن مقدار بهینه یا نزدیک به بهینهٔ معیار عملکرد بوده است.

بیش از پنجاه سال از نخستین تحقیقات در حوزهٔ زمان‌بندی ماشین‌ها می‌گذرد. نخستین کوشش‌ها در این زمینه، بیشتر معطوف به کمینه‌سازی زمان تکمیل و هزینهٔ تأخیر بوده است. به‌کارگیری روش‌های ابتکاری به‌منظور یافتن جواب نزدیک به بهینه در زمان پذیرفتنی و اهمیت این تئوری در مدیریت زمان و منابع در صنایع، باعث شد تا این موضوع در اواخر دههٔ ۶۰ میلادی به موضوع مهمی برای پژوهشگران این حوزه بدل شود (سن و همکاران[۱]، ۲۰۰۳). نخستین بار کانت در پژوهشی مسئلهٔ حداقل‌سازی مجموع هزینه‌های تأخیر و تعجیل را در حالتی بررسی کرد که برای تمامی کارها موعد تحویل یکسان وجود دارد (بیکر و شودر[۲]، ۱۹۹۰). هوگوین و ون د ولد[۳] (۱۹۹۱) با لحاظ‌کردن موعد تحویل نزدیک در مسئلهٔ حداقل‌سازی تابع مجموع وزنی تعجیل و تأخیر، توالی مجموعه‌ای از کارها را تعیین کردند. در این مسئله، دستگاه تنها توانایی پردازش یک کار را داشت. آنها همچنین ثابت کردند که حتی اگر تمامی وزن‌ها یکسان باشد، مسئلهٔ بررسی‌شده NP-hard است. هال و همکاران[۴] (۱۹۹۱) نشان دادند مسئلهٔ حداقل‌سازی هزینه‌های تعجیل و تأخیر با موعد تحویل نزدیک به زمان شروع زمان‌بندی NP-hard است. نیرچو و امیرو[۵] (۲۰۱۳) با بهره‌گیری از الگوریتم ازدحام ذرات[۶] (PSO)، مجموع هزینه‌های تعجیل و تأخیر را برای n کار دارای موعد تحویل نزدیک، به حداقل‌سازی کردند. همچنین نشان دادند در ۸۲٪ موارد، الگوریتم پیشنهادی از عملکرد مناسبی برخوردار است. جایانتی و آنوسویا[۷] (۲۰۱۷) برای حل مسئلهٔ حداقل‌سازی مجموع وزنی تعجیل و تأخیر، از الگوریتم بهینه‌سازی ازدحام ذرات بهره جستند. در این الگوریتم، ذره با کمترین مقدار در ابتدای توالی قرار می‌گیرد و این روند برای دیگر کارها نیز ادامه دارد. در پایان تحلیلی بر نتایج انجام شده است که برآیند آن برتری الگوریتم PSO را در حل این مسئله نشان داده است.

اولین کوشش علمی برای بررسی زمان‌بندی مسئلهٔ پردازش انباشته را ایکورا و گیمبل[۸] (۱۹۸۶) انجام دادند. آنها الگوریتمی را برای حداقل‌سازی معیار زمان پایان عملیات (C_{max}) برای محیط تک ماشین با در نظر گرفتن زمان پردازش ثابت، در هنگامی ارائه کردند که موعدهای تحویل و زمان‌های دسترسی سازگارند ($r_i \le r_j \rightarrow d_i \le d_j$). لی و همکاران[۹] (۲۰۱۵) مسئلهٔ حداقل‌سازی تعجیل و تأخیر را با فرض اندازهٔ غیریکسان کارها بررسی کردند. آنها یک الگوریتم ابتکاری را برای انباشته‌سازی و یک الگوریتم ژنتیک ترکیبی را برای تعیین توالی انباشته‌ها ارائه کردند. پارسا و همکاران[۱۰] (۲۰۱۷) ضمن در نظر گرفتن موعدهای تحویل یکسان و در فاصلهٔ دور نسبت به زمان حال برای کارها در شرایط پردازش انباشته، مسئلهٔ حداقل‌سازی مجموع تعجیل و تأخیر را به‌صورت یک مدل برنامه‌ریزی عدد صحیح بیان و در ادامه از یک الگوریتم برنامه‌ریزی پویا برای یافتن توالی انباشته‌ها استفاده کردند. آنها در پایان نتیجه گرفتند الگوریتم‌های انباشته‌سازی مبتنی بر قاعدهٔ طولانی‌ترین زمان پردازش[۱۱] (LPT)، بهترین

عملکرد را در میان دیگر الگوریتم‌های ابتکاری به دست آورده است. ژانگ و همکاران[۱۲] (۲۰۲۱) مسئلهٔ زمان‌بندی روی تک ماشین پردازندهٔ انباشته را با در نظر گرفتن اهداف تولید بهنگام بررسی کردند. آنها ضمن استخراج ویژگی‌های جواب بهینه، یک الگوریتم ترکیبی مبتنی بر الگوریتم ژنتیک را پیشنهاد داده‌اند.

ترینداده و همکاران[۱۳] (۲۰۲۱) یک مدل‌سازی جدید مبتنی بر گراف برای مسئلهٔ کمینه‌کردن زمان تکمیل آخرین کار را در محیط پردازش انباشته ارائه کرده‌اند. آزمایش‌های محاسباتی آنها حاکی از برتری مدل پیشنهادی‌شان نسبت‌به مدل‌های موجود است. قیروگا و همکاران[۱۴] (۲۰۲۱) برای حل مسئلهٔ زمان‌بندی انباشته با هدف حداقل‌کردن مجموع وزن‌دار تأخیرها، یک الگوریتم ابتکاری مبتنی بر جست‌وجوی را محلی ارائه کرده‌اند. به‌تازگی پسوا و همکاران[۱۵] (۲۰۲۲) یک روش دقیق مبتنی بر مدل‌سازی زمان-گسسته[۱۶] را برای همین مسئله پیشنهاد داده‌اند. آنها به کمک روش پیشنهادی، مسائلی با حداکثر ۱۰۰ کار را به‌صورت بهینه حل کردند. یانگ و همکاران[۱۷] (۲۰۲۲) مسئلهٔ زمان‌بندی پردازش انباشته را با خانواده‌های ناسازگار و تابع هدف مجموع وزن‌دار زمان تکمیل کارها را در نظر گرفته‌اند. آنها برای حل مسئله، چندین مدل‌سازی و یک الگوریتم شاخه و ارزش[۱۸] را توسعه داده‌اند. نتایج آنها حاکی از آن است که مسائلی با حداکثر ۱۵۰ کار را الگوریتم شاخه و ارزش پیشنهادشده حل می‌شود. کونگ و همکاران[۱۹] (۲۰۲۳) موضوع کاهش انتشار کربن را در حوزهٔ مسائل زمان‌بندی تولید انباشته در محیط تولید نیمه‌هادی‌ها بررسی کرده‌اند. تیان و ژنگ[۲۰] (۲۰۲۴) مسئلهٔ زمان‌بندی یک ماشین پردازش انباشته را وقتی بررسی کرده‌اند که قیمت برق مصرفی طی ساعات مختلف روز متفاوت است. آنها یک نحوهٔ مدل‌سازی جدید را برای مسئله و چندین رویکرد حل را برای کاهش هزینه‌های برق مصرفی ارائه کرده‌اند. متعاقباً ژنگ و چن[۲۱] (۲۰۲۴) یک الگوریتم بهبودیافته را برای برقراری تعادل بین مصرف انرژی و بهره‌وری تولید در محیط تولید انباشته ارائه کرده‌اند. فاولر و مونخ[۲۲] (۲۰۲۲) طی یک مقالهٔ مروری، پیشینهٔ موضوع زمان‌بندی پردازش انباشته را به‌صورت جامع بررسی کرده‌اند. آنها پژوهش‌های گذشته را با توجه به محیط ماشین‌آلات و نوع تابع هدف دسته‌بندی و دربارهٔ روندهای تحقیقات آتی بحث کرده‌اند.

جدول ۱- مقایسهٔ تحقیق **حاضر** با دیگر تحقیقات شاخص در این حوزه

Table 1- Comparison of the current research with other leading research in this field

نویسندگان	انباشته	کار	موعد تحویل	روش حل	تابع هدف (min)
هووگوین و ون د ولد (۱۹۹۱)		✓	نزدیک	برنامه‌ریزی پویا	$\sum w_j\left(E_j+T_j\right)$
نیرچو و امیرو (۲۰۱۳)		✓	دور	الگوریتم ***PSO***	$\sum\left(\alpha_j E_j+\beta_j T_j\right)$
جایانتی و آنوسویا (۲۰۱۷)		✓	نزدیک	الگوریتم ***PSO***	$\sum\left(\alpha_j E_j+\beta_j T_j\right)$
لی و همکاران (۲۰۱۵)	✓		دور	الگوریتم ژنتیک	$\sum w_j\left(E_j+T_j\right)$
پارسا و همکاران (۲۰۱۷)	✓		دور	برنامه‌ریزی پویا و ابتکاری	$\sum w_j\left(E_j+T_j\right)$
تحقیق حاضر	✓		نزدیک	برنامه‌ریزی پویا و ابتکاری	$\sum w_j\left(E_j+T_j\right)$

با نگاهی هرچند اجمالی دربارهٔ پیشینهٔ موضوع، درمی‌یابیم که بررسی هم‌زمان مسئله‌های تولید بهنگام و پردازش انباشته تاکنون کمتر بررسی شده است. به‌تازگی پارسا و همکاران (۲۰۱۷) پژوهشی را در زمینهٔ این دو حوزه، با در نظر گرفتن موعد تحویل دور[۲۳] یا فرصت‌دار انجام دادند، حال آنکه موعد تحویل در عمل نزدیک به زمان آغاز زمان‌بندی است که در این صورت شرایط حل مسئله نیز دستخوش تغییر می‌شود. از همین روی در این پژوهش، زمان‌بندی پردازش انباشته برای تحقق اهداف تولید بهنگام، با لحاظ‌کردن موعد تحویل نزدیک به زمان حال، به‌عنوان شکاف تحقیقاتی موجود در پیشینهٔ موضوع بررسی می‌شود. شایان ذکر است که در نظر گرفتن موعد تحویل نزدیک به‌جای دور در مسائل زمان‌بندی، باعث پیچیدگی بسیار بیشتر مسئله خواهد شد. حتی در ساده‌ترین حالت که مسئلهٔ زمان‌بندی تک‌ماشینه است، وقتی موعد تحویل نزدیک به‌جای دور در نظر گرفته می‌شود، مسئله از حالت ساده به یک مسئلهٔ NP-hard تبدیل می‌شود. همچنین در نظر گرفتن موعدهای تحویل غیرمشترک برای کارها، مسئله را به‌شدت پیچیده‌تر می‌کند که خارج از چارچوب کار مطرح‌شده در این مقاله است.

۲- تعریف مسئله

در این بخش مشخصه‌ها، مفروضات و شرایط بهینگی مسئله بیان شده است. دستگاه پردازش انباشته، ماشینی با قابلیت پردازش هم‌زمان چندین کار و یا به‌عبارتی پردازش یک انباشته از کارهاست. در انباشته‌سازی موازی، زمان پردازش انباشته (p_b)، برابر زمان بزرگ‌ترین کار موجود در انباشته است. معیار عملکرد نیز در راستای تحقق تولید بهنگام، حداقل‌سازی مجموع وزنی تعجیل و تأخیر در نظر گرفته و با رابطهٔ $\sum w_j \left|C_j - d\right|$ مشخص شده است. در رابطهٔ مذکور C_j و w_j به ترتیب زمان تکمیل و وزن کار jام است و d موعد تحویل مشترک برای تمامی کارها قلمداد می‌شود. فرض می‌شود موعد تحویل به ابتدای افق برنامه‌ریزی نزدیک است. همچنین رابطهٔ $\sum w_j \left(E_j + T_j\right)$ شکل دیگری از بیان این معیار عملکرد است که در آن $E_j = max_j\left\{0, d - C_j\right\}$ و $T_j = max_j\left\{0,\ C_j - d\right\}$ است. دیگر مفروضات در مدل بررسی‌شده به شرح زیر است:

۱. تعداد n کار وجود دارد که همگی سازگار و متعلق به یک خانواده‌اند؛

۲. هر کار j برای پردازش به میزان s_j واحد از ظرفیت ماشین نیاز دارد. به پارامتر s_j اندازهٔ کار گفته می‌شود؛

۳. همهٔ کارها در لحظهٔ صفر در دسترس‌اند؛

۴. زمان مورد نیاز برای پردازش کار j معلوم و برابر p_j است؛

۵. ظرفیت ماشین برابر B است و تمامی کارها اندازه‌ای کوچک‌تر یا مساوی ظرفیت ماشین دارند؛

۶. توقف پردازش روی یک انباشته ممکن نیست و پس از شروع پردازش بر دستگاه، هیچ کاری به انباشته اضافه یا از آن کم نمی‌شود؛

۷. موعد تحویل کارها یکسان و برابر d است. موعد تحویل نزدیک[۲۴] در نظر گرفته می‌شود.

شایان ذکر است موعد تحویل نزدیک مواقعی مطرح می‌شود که فاصلهٔ زمانی تا تحویل سفارش‌ها به مشتری یا مشتریان به‌گونه‌ای است که حتی اگر کار در ابتدای افق برنامه‌ریزی تولید شروع شود، باز هم بخشی از سفارش‌ها با تأخیر مواجه خواهند شد. در مقابل اگر تا زمان موعد تحویل سفارش‌ها، به‌اندازهٔ کافی مهلت وجود داشته باشد که همهٔ سفارش‌ها بدون تأخیر تولید شود، آنگاه با حالت موعد تحویل دور یا مهلت‌دار مواجه خواهیم بود. براساس

نمادگذاری استاندارد مسائل زمان‌بندی (گراهام و همکاران[۲۵]، ۱۹۷۹)، مسئلهٔ مدنظر در این پژوهش با نماد $1\left|p-batch,\ s_i \le B,\ d_i = d\right|\sum w_j\left(E_j + T_j\right)$ مشخص و بیان می‌شود.

با توجه به پژوهش براکر و همکاران[۲۶] (۱۹۹۸)، تمامی مسائل پردازش انباشته‌ای که معیار عملکرد مبتنی بر موعد تحویل دارند، ازنظر پیچیدگی محاسباتی در طبقهٔ NP-hard قرار دارند. از همین روی مسئلهٔ بررسی‌شده در این پژوهش نیز NP-hard است.

برخی از شرایط و ویژگی‌های هر زمان‌بندی بهینه برای مسئله، با مفروضات ذکرشده در فوق به شرح زیر است (مونخ و همکاران[۲۷]، ۲۰۰۶):

۱. هیچ زمان بیکاری بین پردازش دو انباشتهٔ متوالی وجود ندارد. به عبارت دیگر با شروع پردازش انباشتهٔ اول، انباشته‌های بعدی بدون هیچ بیکاری بین پردازش انباشته‌ها پردازش خواهند شد؛

۲. انباشته‌هایی که قبل یا مقارن با موعد تحویل تکمیل می‌شوند (که آنها را با مجموعهٔ E نشان می‌دهیم و درواقع انباشته‌های دارای تعجیل‌اند) به ترتیب صعودی نسبت وزن به زمان پردازش انباشته‌ها (w_b/p_b) و انباشته هایی که بعد از موعد تحویل تکمیل می‌شوند (که آنها را با مجموعهٔ T نشان می‌دهیم و درواقع انباشته‌های دارای تأخیرند)، به ترتیب نزولی نسبت وزن به زمان پردازش انباشته‌ها (w_b/p_b) مرتب می‌شوند که در آن p_b بزرگ‌ترین زمان پردازش کارهای تخصیص‌یافته به انباشته bام و w_b مجموع وزن کارهای تخصیص داده شده به انباشته bام است. در صورتی که کارها وزن یکسانی داشته باشند، آنگاه w_b معادل تعداد کارهای درون هر انباشته خواهد بود؛

۳. حداقل یکی از این دو حالت برقرار است: یا اولین کار در زمان صفر شروع می‌شود، یا موعد تحویل مقارن با زمان شروع یا تکمیل انباشته‌ای خواهد بود که بزرگ‌ترین نسبت w_b/p_b را دارد.

چنانچه در توالی ارائه‌شده برای انباشته‌ها، انباشته‌ای وجود داشته باشد که قبل از موعد تحویل شروع و بعد از موعد تحویل تکمیل شود، انباشتهٔ میانی[۲۸] نامیده می‌شود. در صورت وجود انباشتهٔ میانی، اولین انباشته در زمان‌بندی بهینه حتماً در زمان صفر شروع می‌شود و بازهٔ زمان‌بندی به‌صورت $[0, \sum p_b]$ است.

۳- روش‌شناسی پژوهش

با توجه به NP-hard بودن مسئلهٔ بررسی‌شده در این پژوهش، در این بخش الگوریتم‌هایی را برای حل کارایی مسئله بررسی می‌کنیم. به‌طور کلی حل مسئلهٔ بررسی‌شده دو مرحلهٔ اصلی دارد:

مرحلهٔ اول: تشکیل انباشته و تعیین زمان پردازش هر انباشته براساس بزرگ‌ترین زمان پردازش کارهای موجود در آن؛

مرحلهٔ دوم: تعیین توالی انباشته‌های تشکیل‌شده برای پردازش بر دستگاه.

در ادامه با توجه به مبانی نظری و مفروضات مطرح‌شده در بخش قبل، ابتدا الگوریتم‌هایی را برای ایجاد انباشته‌ها بررسی می‌کنیم و سپس رویکردهایی برای تعیین توالی انباشته‌های تشکیل‌شده ارائه می‌کنیم. در انتها یک الگوریتم ابتکاری مبتنی بر روش بهبود فردی و همچنین الگوریتم ترکیبی ازدحام ذرات را برای حل کامل مسئله پیشنهاد می‌کنیم.

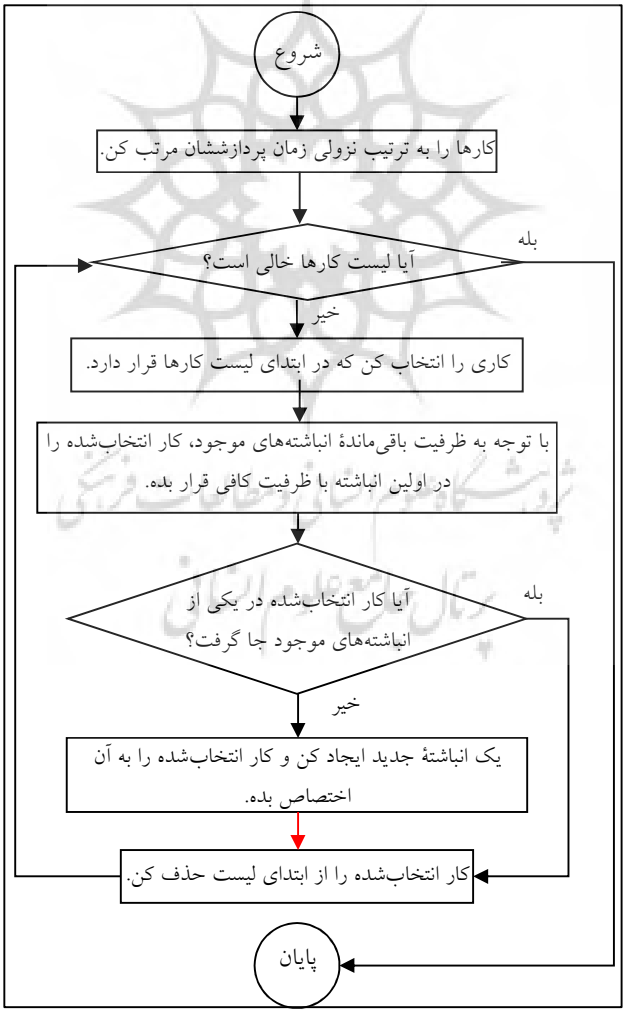
۳-۱- الگوریتم ابتکاری برای انباشته‌سازی

برای یافتن یک زمان‌بندی موجه، ابتدا باید کارها را درون انباشته‌ها قرار داد و سپس توالی قرارگیری انباشته‌ها را روی ماشین تعیین کرد. با توجه به نتایج به دست آمده از پژوهش پارسا و همکاران (۲۰۱۷)، به‌طور کلی الگوریتم های ابتکاری مبتنی بر قاعدهٔ طولانی‌ترین زمان پردازش برای مسئلهٔ حداقل‌سازی مجموع وزنی تعجیل و تأخیر انباشته‌ها جواب بهتری نسبت‌به دیگر الگوریتم‌های ابتکاری ایجاد می‌کنند. از همین روی برای تولید انباشته، از الگوریتم ابتکاری مبتنی بر قاعدهٔ LPT به شرح زیر استفاده می‌شود:

گام ۱. کارها به ترتیب نزولی زمان پردازش مرتب می‌شوند. در صورتی که دو کار دارای زمان پردازش یکسان باشند، براساس ترتیب نزولی اندازه‌شان مرتب می‌شوند؛

گام ۲. اولین کار در ابتدای لیست، انتخاب و در اولین انباشته با ظرفیت خالی قرار می‌گیرد. اگر اندازهٔ کار منتخب به‌گونه‌ای بود که در هیچ انباشته‌ای قرار نمی‌گرفت، به یک انباشتهٔ جدید تخصیص داده می‌شد. به این روش، قاعدهٔ First Fit گفته می‌شود. گام ۲ تا اختصاص تمام کارها به انباشته‌ها تکرار می‌شود.

مراحل الگوریتم ابتکاری انباشته‌سازی در شکل ۱ نشان داده شده است.



شکل ۱- مراحل اجرای الگوریتم ابتکاری انباشته‌سازی

Fig. 1- Flowchart of batch formation algorithm

۳-۲- انباشته‌سازی به روش حل مدل ریاضی با هدف کمینه‌سازی زمان تکمیل آخرین انباشته

ازجمله عوامل مؤثر در رسیدن به جواب بهینه، تشکیل انباشته‌های مناسب از کارهاست. روش انباشته‌سازی در الگوریتم ابتکاری این پژوهش، مبتنی بر قاعدهٔ LPT است. اما در این بخش، انباشته‌سازی ازطریق حل مدل ریاضی زیر انجام می‌شود. کاهش تعداد متغیرها در ازای ثابت‌کردن جای انباشته‌ها در توالی، اصلی‌ترین ویژگی این مدل است. با توجه به اینکه کمینه‌سازی معیار عملکرد زمان تکمیل آخرین انباشته (C_{max}) که معادل $\sum p_b$ است، وابسته به ترتیب قرارگیری انباشته‌ها نیست، ثابت‌کردن جای انباشته در توالی، سبب بروز مشکل نمی‌شود. به عبارت دیگر در مسائل زمان‌بندی انباشته با تابع هدف C_{max}، تنها کافی است دربارهٔ نحوهٔ تشکیل انباشته‌ها تصمیم‌گیری شود و پس از تشکیل انباشته‌ها، هر ترتیبی از آنها دارای C_{max} یکسانی برابر مجموع زمان پردازش انباشته‌ها خواهد بود.

در مسئلهٔ بررسی‌شده در این تحقیق ($\sum w_j (E_j + T_j)$)، تشکیل انباشتهٔ مناسب و جابه‌جایی انباشته‌ها در توالی، عوامل اصلی کمینه‌سازی معیار عملکردند. به همین دلیل امکان حل مدل ریاضی مسئله با معیار عملکرد $\sum w_j (E_j + T_j)$ از این طریق وجود ندارد. بنابراین بعد از تعیین انباشته‌ها از این روش که به حداقل‌شدن زمان تکمیل آخرین انباشته منجر می‌شود، مجموع وزن‌دار تأخیر و تعجیل‌ها با استفاده از الگوریتم‌های پیشنهادی کمینه می‌شود.

در این بخش برای انباشته‌سازی، از مدل ریاضی ارائه‌شدهٔ ترینداده و همکاران (۲۰۱۸) استفاده شده است. به کمک این مدل، انباشته‌ها به‌نحوی تشکیل می‌شوند که $\sum p_b$ کمینه شود. برای بیان مدل ریاضی، فرض کنید کارها به‌گونه‌ای مرتب شده‌اند که $p_1 \le p_2 \le \ldots \le p_n$. اگر کار j به انباشتهٔ k اختصاص یابد، متغیر تصمیم صفر و یک x_{jk} مقدار یک خواهد گرفت و در غیر این صورت، صفر خواهد بود. متغیر x_{jk} تنها به ازای $j \le k$ تعریف می‌شود و به عبارت دیگر کار j تنها در صورتی به انباشتهٔ k اختصاص می‌یابد که $j \le k$ باشد. انباشتهٔ k هم در صورتی تشکیل می‌شود که کار k به آن تخصیص یابد. با این نحوهٔ تعریف متغیر تصمیم، هم تقارن[۲۹] در مدل ریاضی از بین می‌رود و هم تعداد متغیرهای صفر و یک از n^2 به $n(n+1)/2$ کاهش می‌یابد. نکتهٔ مهم‌تر این است که این مفروضات به جواب‌هایی منجر می‌شود که در آن زمان پردازش انباشتهٔ k، در صورت استفاده، برابر با p_k خواهد بود؛ زیرا کار k به‌طور حتم به انباشتهٔ k اختصاص داده شده است و همچنین کاری است که طولانی‌ترین زمان پردازش را بین کارهای اختصاص‌یافته به آن انباشته دارد. به کمک این متغیرهای تصمیم، مدل ریاضی تشکیل انباشته‌ها به‌صورت زیر خواهد بود.

$$\text{Min} \sum_{k=1}^{n} p_k x_{kk} \qquad (۱)$$

s.t:

$$\sum_{k \ge j} x_{jk} = 1 \qquad \forall j = 1,2,\ldots,n \qquad (۲)$$

$$\sum_{j \le k} x_{jk} \le B x_{kk} \qquad \forall k = 1,2,\ldots,n \qquad (۳)$$

$$x_{jk} \le x_{kk} \qquad \forall k = 1,2,\ldots,n ; j \le k \;\; \forall j = 1,2,\ldots,n; \qquad (۴)$$

$$x_{jk} \in \{0, 1\} \qquad \forall k = 1,2,\ldots,n ; j \le k \;\; \forall j = 1,2,\ldots,n; \qquad (۵)$$

عبارت (۱) بیان‌کنندهٔ تابع هدف و معادل کمینه‌سازی زمان تکمیل آخرین انباشته است. دستهٔ محدودیت (۲) برای اطمینان از اینکه هر کار دقیقاً به یک انباشته اختصاص می‌یابد، به مدل اضافه شده است. این موضوع که اندازهٔ انباشته‌ها نباید از ظرفیت ماشین تجاوز کند، به کمک دسته محدودیت (۳) در نظر گرفته شده است. دسته محدودیت (۴) تضمین می‌کند که تنها در صورت تشکیل انباشتهٔ k، این امکان وجود خواهد داشت که کاری به آن تخصیص یابد. عبارت (۵) نیز نوع متغیرهای تصمیم مدل را بیان می‌کند. آزمایش‌های محاسباتی نشان می‌دهد به کمک مدل ریاضی فوق، مسائلی با ۱۰۰ کار یا کمتر روی کامپیوترهای شخصی امروزی به‌راحتی حل می‌شود.

۳-۳- تعیین توالی انباشته‌ها به کمک الگوریتم برنامه‌ریزی پویا

برنامه‌ریزی پویا الگوریتمی بسیار قدرتمند است که در آن مسئلهٔ اصلی ازطریق تقسیم‌شدن، به مجموعه‌ای از زیرمسئله‌ها حل می‌شود. حل مسئله با شروع از کوچک‌ترین زیرمسئله شروع می‌شود و با یافتن جواب‌هایی برای مسائل کوچک در رسیدن ساده‌تر به جواب مسائل بزرگ‌تر ادامه می‌یابد تا زمانی که کل مسئله حل شود. در این بخش یک الگوریتم برنامه‌ریزی پویا را برای تعیین توالی بهینهٔ پردازش انباشته‌ها روی ماشین ارائه می‌کنیم. فرض کنید کارها به روشی به انباشته‌ها اختصاص یافته‌اند و تعداد m انباشته در اختیار است که توالی پردازش آنها باید مشخص شود. مدت پردازش انباشتهٔ b را p_b و وزن آن را w_b در نظر بگیرید. با توجه به اینکه موعد تحویل نزدیک است، اولین انباشته در زمان‌بندی بهینه، حتماً در زمان صفر شروع می‌شود و انباشته‌ها در فاصلهٔ زمانی $[0, \sum p_b]$ قرار می‌گیرند. همچنین ممکن است انباشته‌ای با شرایط انباشتهٔ میانی در زمان‌بندی بهینه وجود داشته باشد. لم زیر، جزییات بیشتری را دربارهٔ مشخصات زمان‌بندی بهینه بیان می‌کند.

لم ۱. در زمان‌بندی بهینه، انباشته‌هایی که پیش از موعد تحویل یا هم‌زمان با آن تکمیل می‌شوند، به ترتیب غیر نزولی w_b/p_b تنظیم شده‌اند و انباشته‌هایی که پس از موعد تحویل یا هم‌زمان با آن شروع می‌شوند، به ترتیب غیر صعودی w_b/p_b مرتب شده‌اند. به عبارت دیگر، بدون در نظر گرفتن انباشتهٔ میانی، انباشته‌هایی که نسبت وزن به زمان پردازش بیشتری دارند، نزدیک‌تر به موعد تحویل قرار می‌گیرند و هر چقدر از موعد تحویل دور می‌شویم (چه قبل و چه پس از آن)، نسبت وزن به زمان پردازش انباشته‌ها کاهش می‌یابد.

اثبات: با فرض خلف و به کمک انجام جابه‌جایی جفتی دو انباشتهٔ مجاور، لم ۱ اثبات می‌شود. دو انباشتهٔ مجاور k و l را در نظر بگیرید که در زمان‌بندی بهینهٔ پیش از موعد تحویل تکمیل می‌شوند. فرض کنید در زمان‌بندی بهینهٔ S_1، ابتدا انباشتهٔ k و سپس بلافاصله پس از آن انباشتهٔ l قرار دارد. t_0 را زمان تکمیل انباشتهٔ l در زمان‌بندی S_1 در نظر بگیرید. واضح است که در این صورت زمان تکمیل انباشتهٔ k برابر $t_0 - p_l$ خواهد بود. اگر $w_k/p_k < w_l/p_l$ باشد (فرض خلف)، آنگاه با جابه‌جایی جفتی این دو انباشتهٔ مجاور و ثابت نگه داشتن توالی بقیهٔ انباشته‌ها، زمان‌بندی دیگری به نام S_2 به دست می‌آید که مقدار تابع هدف آن کمتر از تابع هدف زمان‌بندی S_1 خواهد بود؛ زیرا هزینهٔ دو انباشتهٔ k و l در زمان‌بندی S_1 معادل $w_k(d - t_0 + p_l) + w_l(d - t_0)$ و این هزینه در زمان‌بندی S_2 برابر $w_l(d - t_0 + p_k) + w_k(d - t_0)$ است. با توجه به اینکه هزینهٔ دیگر انباشته‌ها در هر دو

زمان‌بندی S_1 و S_2 یکسان است، بنابراین تفاوت هزینهٔ آنها با $w_k p_l - w_l p_k$ برابر است که با توجه به فرض خلف مقداری مثبت است؛ درنتیجه زمان‌بندی S_1 آن بهینه این نیست که با فرض ما در تناقض است. اثبات برای انباشته‌های پس از موعد تحویل به‌صورت مشابه انجام‌شدنی است.

با توجه به لم ۱ و برای تشریح الگوریتم برنامه‌ریزی پویا، فرض کنید انباشته‌ها به ترتیب غیر کاهشی نسبت وزن به زمان پردازششان مرتب شده‌اند؛ به عبارت دیگر $w_1/p_1 \le w_2/p_2 \le \ldots \le w_m/p_m$. مطابق لم ۱ انتظار داریم انباشته‌های با اندیس کوچک‌تر، دورتر از موعد تحویل و انباشته‌های با اندیس بزرگ‌تر، نزدیک‌تر به موعد تحویل قرار بگیرند. انباشتهٔ h را انباشتهٔ میانی در نظر بگیرید و $F_c^h(t)$ را معادل مقدار بهینهٔ هزینهٔ زمان‌بندی c انباشتهٔ اول تحت این شرایط قرار دهید که در آن باید مجموع زمان پردازش انباشته‌هایی برابر t باشد که پیش از موعد تحویل قرار می‌گیرند. به عبارت دیگر این c انباشته باید در بازهٔ $[0, t]$ و $\left[\sum_{b=c+1}^{m} p_b + t, \sum_{b=1}^{m} p_b\right]$ قرار بگیرند. انباشته‌هایی که در بازهٔ $[0, t]$ قرار می‌گیرند، دارای تعجیل و انباشته‌های بازهٔ دوم دارای تأخیر خواهند بود. رابطهٔ بازگشتی برنامه‌ریزی پویا به‌صورت زیر بیان می‌شود:

$$F_c^h(t) = \begin{cases} F_{c-1}^h(t) & \text{اگر } c = h \quad (۶) \\ F_c^h(t) + w_c\left(\sum_{b=c}^{m} p_b + t - d\right) & \text{اگر } d - p_c \le t \le d \quad (۷) \\ F_{c-1}^h(t - p_c) + w_c(d - t) & \text{اگر } \sum_{b=c+1}^{m} p_b + t \le d \quad (۸) \\ \min\{F_{c-1}^h(t - p_c) + w_c(d - t),\ F_c^h(t) + w_c(\sum_{b=c}^{m} p_b + t - d)\} & \text{در غیر این صورت} \quad (۹) \end{cases}$$

با فرض آنکه $c-1$ انباشتهٔ اول زمان‌بندی‌شده باشند و با توجه به لم ۱، انباشتهٔ cام یا باید در انتهای بازهٔ اول، یعنی $[0, t]$ یا در ابتدای بازهٔ دوم قرار بگیرد. رابطهٔ بازگشتی (۹) هر دو حالت را بررسی و کمترین مقدار را انتخاب می‌کند. رابطهٔ (۷) همان رابطهٔ (۹) است، در حالتی که بازهٔ انباشته‌های دارای تعجیل، یعنی بازهٔ $[0, t]$ به‌اندازهٔ کافی فضا برای جای‌دادن انباشتهٔ c را نداشته و درنتیجه فقط بررسی حالت دوم ضروری خواهد باشد. رابطهٔ (۸) نیز به همین صورت به حالتی مربوط است که بازهٔ انباشته‌های دارای تأخیر، یعنی بازهٔ $\left[\sum_{b=c+1}^{m} p_b + t, \sum_{b=1}^{m} p_b\right]$ فضای کافی نداشته باشد و انباشتهٔ c فقط قبل از موعد تحویل قرار بگیرد. رابطهٔ (۶) هم به انباشتهٔ میانی مربوط است که هزینهٔ آن بعداً محاسبه خواهد شد.

روابط بازگشتی فوق ماشین را در بازهٔ $[t, t + p_h]$ بیکار نگه می‌دارند تا انباشتهٔ میانی در این بازه قرار بگیرد. هزینهٔ نهایی با افزودن هزینهٔ انباشتهٔ میانی ($G_m^h(t)$) از رابطهٔ زیر محاسبه می‌شود:

$$G_m^h(t) = \begin{cases} F_m^h(t) + w_h(t + p_h - d) & \text{اگر } d - p_h \le t \le d \quad (۱۰) \\ \infty & \text{در غیر این صورت} \quad (۱۱) \end{cases}$$

جواب بهینه ازطریق رابطهٔ $\min_{1 \le h \le m} \min_{d - p_h \le t \le d} G_m^h(t)$ به دست می‌آید. شرایط اولیه نیز به‌صورت زیر است:

$$F_c^h(t)=\begin{cases} 0 & t=0,\ c=0 \quad (۱۲)\\ \infty & t=0,\ c\neq 0 \quad (۱۳)\\ \infty & t\neq 0,\ c=0 \quad (۱۴)\end{cases}$$

با توجه به روابط بازگشتی الگوریتم برنامه‌ریزی پویا، تعداد محاسبات لازم و درنتیجه زمان لازم برای یافتن جواب بهینه از ردهٔ $O(m^2d)$ است. پس از تشکیل انباشته‌ها، از الگوریتم برنامه‌ریزی پویای شرح داده شده در این بخش برای تعیین توالی بهینهٔ انباشته‌ها استفاده می‌شود؛ برای مثال یک روش ابتکاری مناسب برای تشکیل انباشته‌ها به کار برده و سپس از برنامه‌ریزی پویا برای تعیین بهترین توالی انباشته‌های تشکیل‌شده استفاده می‌شود.

۳-۴- ارائهٔ یک الگوریتم ابتکاری برای حل مسئله

در این بخش یک الگوریتم ابتکاری را برای حل مسئله ارائه می‌کنیم. در این الگوریتم ابتدا انباشته‌ها به روشی ابتکاری تشکیل و توالی آنها مشخص می‌شود، سپس جواب تولیدشده به کمک یک روش بهبوددهنده تا جای ممکن، بهبود می‌یابد. در این پژوهش از روش بهبود فردی[۳۰] (IE) استفاده شده است که یک روش ساده و پرکاربرد در جست‌وجوی همسایگی‌های یک جواب برای بهبود آن است.

۳-۴-۱- روش بهبود فردی

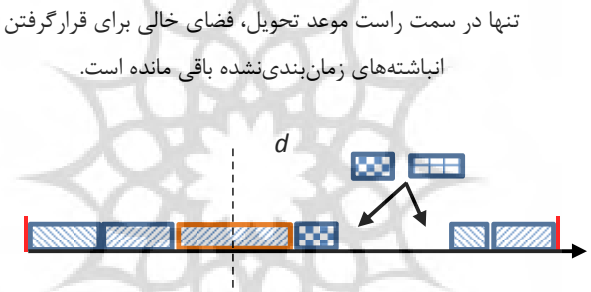
روش بهبود فردی یک روش جست‌وجوی همسایگی است که در آن مجموعه همسایه‌های جواب فعلی، شامل همسایه‌های به طول دو است (کو و همکاران[۳۱]، ۲۰۰۹). فرض کنید π_1 و π_2 دو بردار مربوط به توالی به‌صورت $\pi_1=(\pi_1(1),\pi_1(2),\ \dots,\pi_1(n))$ و $\pi_2=(\pi_2(1),\pi_2(2),\ \dots,\pi_2(n))$ باشند، چنانچه تعداد مؤلفه‌های متفاوت در این دو جواب k باشد، گفته می‌شود π_2 در یک همسایگی π_1 به طول k قرار دارد و برعکس.

در روش IE، ابتدا یک جواب اولیه تولید و سپس در هر مرحله تعدادی جواب همسایه به طول دو ایجاد و مقدار تابع هدف، به ازای جواب‌های همسایهٔ جواب جاری محاسبه و مقایسه می‌شود، در صورتی که بهبودی حاصل شد، جواب همسایه جایگزین جواب فعلی مسئله می‌شود.

با توجه به اینکه که در مسئلهٔ زمان‌بندی ماشین‌های پردازش انباشته، با جابه‌جایی کارهای درون یک انباشته، زمان پردازش انباشته تغییری نمی‌کند، از همسایه‌های ایجادشده به این وسیله صرف‌نظر می‌شود. از طرفی اگر کارهای انباشته‌های غیریکسان با یک دیگر جابه‌جا شوند، مجموعه‌ای بزرگ از جواب‌ها ایجاد می‌شود که بسیاری از این جابه‌جایی‌ها به جواب مشابه منجر می‌شود، از همین روی این نوع جابه‌جایی نیز نادیده گرفته می‌شود. بنابراین در روش پیشنهادی این پژوهش، تنها معاوضهٔ مکان انباشته‌ها در توالی با هم لحاظ می‌شود. الگوریتم IE، برای حل مسئله از یک جواب اولیه شروع می‌کند. مطابق الگوریتم پیشنهادی در این پژوهش، ابتدا یک الگوریتم ابتکاری، جواب اولیهٔ مناسبی را برای الگوریتم IE تولید می‌کند و سپس الگوریتم IE با جابه‌جایی در توالی انباشته‌ها، کیفیت جواب اولیه را تا حد امکان ارتقا می‌دهد. مزیت الگوریتم IE، برای حل مسئلهٔ بررسی‌شده در این پژوهش، سهولت در اجرا و قدرت بالای این الگوریتم در بررسی جواب‌های ممکن است.

۳-۴-۲- الگوریتم ابتکاری پیشنهادی برای تولید جواب اولیه (HA_IE)

در هر زمان‌بندی بهینه، انباشته‌هایی که قبل یا مقارن با موعد تحویل تکمیل می‌شوند (مجموعهٔ E)، به ترتیب صعودی نسبت w_b/p_b هستند و انباشته‌هایی که پس از موعد تحویل شروع و تکمیل می‌شوند (مجموعه T)، به ترتیب نزولی نسبت w_b/p_b قرار دارند. بر این اساس اگر انباشته‌ها به ترتیب صعودی نسبت w_b/p_b مرتب شوند، آنگاه اولین انباشته با کمترین مقدار نسبت w_b/p_b، باید در یکی از موقعیت‌های اول یا آخر برنامهٔ زمان بندی بهینه قرار گیرد. به بیان بهتر، پس از آنکه ترتیب اولیه برای انباشته‌ها براساس افزایش نسبت w_b/p_b مشخص شد، اولین انباشته انتخاب و بررسی می‌شود. آیا قرارگرفتن آن در اولین موقعیت، هزینهٔ کمتری ایجاد می‌کند یا در آخرین موقعیت؟ سپس انباشتهٔ اول در موقعیتی قرار می‌گیرد که هزینهٔ کمتری ایجاد کند. پس از اختصاص انباشتهٔ اول، روند تعیین توالی برای دیگر انباشته‌ها نیز بر همین اساس خواهد بود. به‌دلیل آنکه موعد تحویل نزدیک (Tight) است و بعد از چند تکرار ممکن است فضای خالی کافی قبل از آن وجود نداشته باشد، بنابراین برای تکرارهایی که در این شرایط صدق می‌کنند، فقط از فضای خالی سمت راست موعد تحویل استفاده می‌شود (شکل ۲). در پایان شکل بردار توالی، براساس نسبت w_b/p_b، مشابه ۸ خواهد بود.



شکل ۲- نحوهٔ قرارگیری انباشته‌ها پس از موعد تحویل

Fig. 2- Arrangement of the batches after the due date

۳-۴-۳- بهبود جواب به کمک الگوریتم IE

در این بخش، چگونگی بهبود کیفیت توالی اولیه به کمک الگوریتم IE تشریح می‌شود. ایجاد همسایگی در الگوریتم IE، مبتنی بر این شرط است که در زمان‌بندی بهینه حتماً شکل بردار توالی براساس نسبت w_b/p_b انباشته ها، مشابه ۸ است. بنابراین جابه‌جایی در بردار توالی تنها متناسب با حفظ این شرط انجام می‌شود و اگر جابه‌جایی چنین شرطی نداشت، اصلاً بررسی نمی‌شود. در ابتدا اولین و دومین انباشته انتخاب می‌شود، جای آنها در بردار توالی جابه‌جا و در صورت نیاز مقدار معیار عملکرد به ازای این تغییر محاسبه می‌شود. در مرحلهٔ بعد معاوضهٔ میان انباشتهٔ اول و سوم انجام می‌شود. روند جابه‌جایی در بردار توالی و محاسبهٔ مقدار عملکرد تا آخرین انباشتهٔ بردار توالی ادامه می‌یابد. در مرحلهٔ بعد از میان کلیه جابه‌جایی‌ها، بهترین جابه‌جایی شناسایی و جابه‌جایی مربوط به آن در بردار توالی اعمال می‌شود. این روند بعد از این مرحله برای دیگر انباشته‌ها تکرار خواهد شد.

مراحل اجرای الگوریتم پیشنهادی به‌صورت زیر است:

۱. کارها به ترتیب نزولی زمان پردازش مربوط به آنها و سپس به ترتیب نزولی اندازه‌شان مرتب شده‌اند. در صورتی که دو کار دارای زمان پردازش برابر باشند، براساس نزولی‌بودن اندازه‌شان مرتب می‌شوند؛

۲. اولین کار در ابتدای لیست انتخاب و در اولین انباشته با ظرفیت خالی قرار می‌گیرد. اگر اندازهٔ کار منتخب به‌گونه‌ای بود که در هیچ انباشته‌ای قرار نمی‌گرفت، به یک انباشتهٔ جدید تخصیص داده می‌شود. گام دو تا اختصاص تمام کارها به انباشته‌ها تکرار می‌شود؛

۳. انباشته‌های تولیدشده در مراحل قبل، به ترتیب صعودی‌بودن نسبت w_b/p_b مرتب می‌شوند؛

۳-۱. انباشته‌ای انتخاب می‌شود که در ابتدای لیست قرار دارد؛

۳-۲. اولین مکان خالی سمت چپ و آخرین مکان خالی سمت راست موعد تحویل برای تخصیص انباشته بررسی می‌شود؛ سپس انباشته به موقعیتی اختصاص می یابد که هزینهٔ کمتری داشته باشد. انباشتهٔ اختصاص یافته از لیست حذف می‌شود.

۳-۳. اگر هنوز به انتهای لیست نرسیده‌ایم، به گام ۳-۱ برمی‌گردیم.

۴. تا برقراری شرط توقف با استفاده از الگوریتم IE، جواب بهبود داده می‌شود:

۴-۱. توالی اولیهٔ σ_{ET} و مقدار معیار عملکرد به ازای آن محاسبه و برابر ET قرار داده می‌شود؛

۴-۲. در توالی به دست آمده از مرحلهٔ قبل، مکان دو انباشته معاوضه و در صورت برقراری شرایط لم ۱، توالی مربوطه برابر σ' و مقدار معیار عملکرد در این توالی برابر ET' قرار داده می‌شود.

۴-۳. اگر رابطهٔ $ET \geq ET'$ برقرار باشد، آنگاه: $\sigma_{ET} \leftarrow \sigma'$ و $ET \leftarrow ET'$.

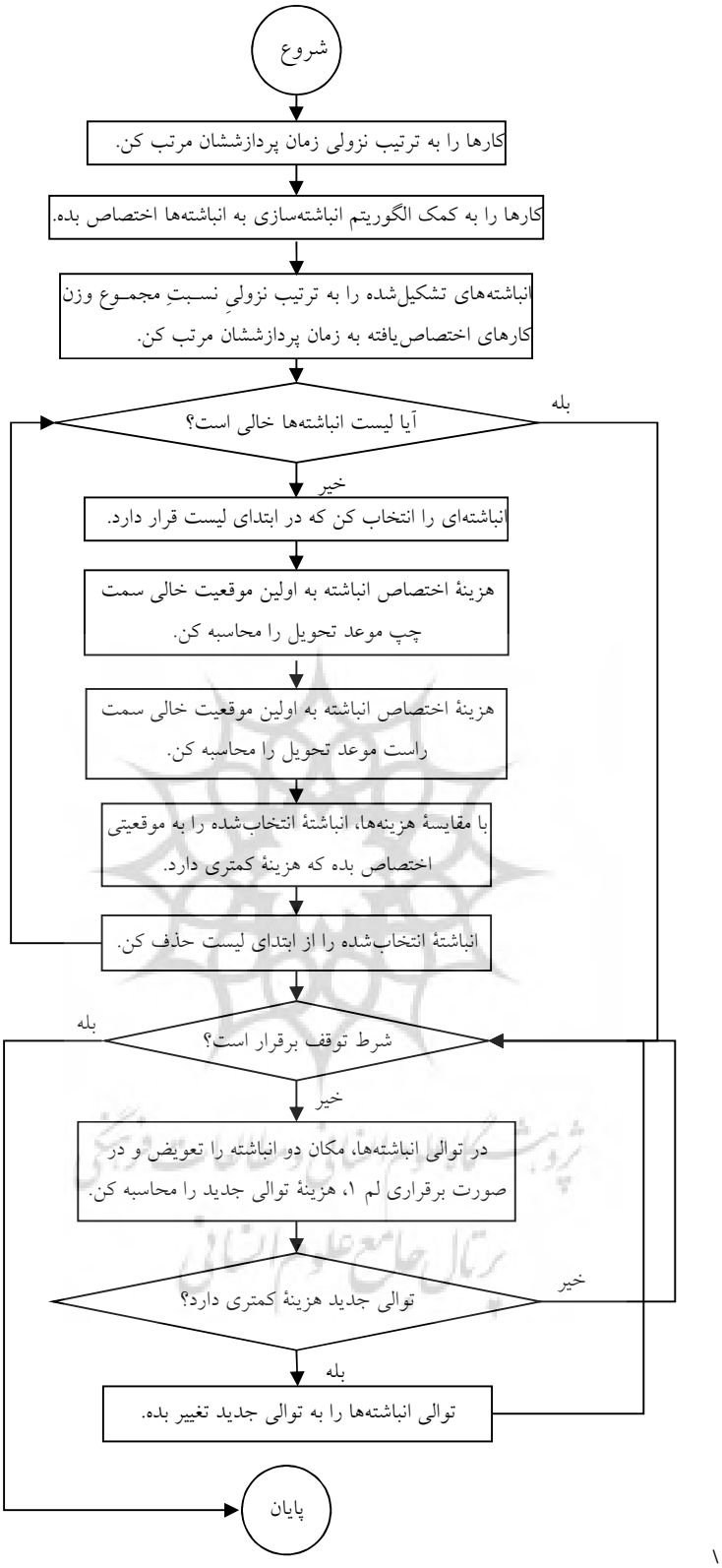
مراحل اجرای الگوریتم در شکل ۳ نشان داده شده است.

۳-۵- الگوریتم ترکیبی ازدحام ذرات و بهبود فردی (PSO_IE)

در این بخش الگوریتم پیشنهادی مبتنی بر ترکیب الگوریتم‌های بهینه‌سازی ازدحام ذرات (PSO) و بهبود فردی (IE) را معرفی می‌کنیم. با توجه به عملکرد مناسب الگوریتم PSO به‌عنوان یک الگوریتم فراابتکاریِ جمعیت‌محور در حل مسائل زمان‌بندی پردازش انباشته (فاولر و مونخ، ۲۰۲۲)، در این پژوهش از این الگوریتم استفاده شده است. در روش پیشنهادی این بخش، ابتدا الگوریتم PSO، جواب اولیه‌ای برای الگوریتم IE تولید می‌کند و سپس الگوریتم IE با جابه‌جایی در توالی انباشته‌ها، کیفیت جواب‌ها را تا حد امکان ارتقا می‌دهد.

در الگوریتم بهینه‌سازی ازدحام ذرات، بردار d بعدی X_i^k، موقعیت ذرهٔ iام در تکرار kام را مشخص می‌کند و به‌صورت بردار $X_i^k = \left(x_{i1}^k, x_{i2}^k, x_{i3}^k, \ldots, x_{id}^k\right)$ تعریف می‌شود. مؤلفه‌های بردار موقعیت x_{ij}^k نیز، بیانگر مقدار مؤلفهٔ jام در ذرهٔ iام و تکرار kام است. الگوریتم ازدحام ذرات با دریافت تعدادی جواب اولیه، شروع به جست وجو در فضای شدنی مسئله و جواب‌هایی را به همان شکل اولیه تولید می‌کند. مقادیر اولیه برای مؤلفه‌های بردار موقعیت، از رابطهٔ (۱۵) به دست می‌آیند:

$$x_{ij}^0 = x_{min} + rand\left(x_{max} - x_{min}\right) \qquad (۱۵)$$



شکل ۳ – مراحل اجرای الگوریتم پیشنهادی

Fig. 2- Flowchart of the proposed algorithm

در رابطهٔ (۱۵)، عبارات x_{max} و x_{min} حدود بالا و پایین اولیه‌ای‌اند که به‌صورت پارامتر ورودی الگوریتم تعریف می‌شوند. همچنین عبارت $rand$ یک مقدار تصادفی دارای توزیع یکنواخت را در بازهٔ $[0, 1]$ اختیار می‌کند. هر ذره برای حرکت‌کردن در فضا، به یک سرعت نیاز دارد. بردار d بعدی V_i^k سرعت ذرهٔ iام در تکرار kام را مشخص می‌کند و به‌صورت $V_i^k = \left(v_{i1}^k, v_{i2}^k, v_{i3}^k, \ldots, v_{id}^k\right)$ تعریف می‌شود. مؤلفه‌های بردار سرعت v_{ij}^k نیز بیانگر مقدار مؤلفهٔ jام در ذرهٔ iام و تکرار kام است. با افزودن سرعت به موقعیت ذرات، موقعیت‌های جدیدی برای هر ذره در نظر گرفته می‌شود. برای مقداردهی اولیهٔ مؤلفه‌های بردار سرعت از رابطهٔ (۱۶) استفاده می‌شود.

$$v_{ij}^0 = v_{min} + rand\left(v_{max} - v_{min}\right) \quad (۱۶)$$

در رابطهٔ (۱۶) عبارات v_{max} و v_{min} حدود بالا و پایین اولیه‌ای‌اند که برای سرعت ذرات تعریف می‌شوند. اصولاً الگوریتم ازدحام ذرات برای مسائلی استفاده می‌شود که فضای حل آنها پیوسته باشد؛ اما در مسئلهٔ زمان‌بندی تک ماشین پردازش انباشته با معیار عملکرد حداقل‌سازی تأخیر و تعجیل، همانند بیشتر مسائل زمان‌بندی فضای حل گسسته است. مهم‌ترین مسئله در اجرای الگوریتم PSO برای مسائل زمان‌بندی، پیداکردن یک رابطهٔ مناسب بین موقعیت ذرات در الگوریتم PSO و توالی کارها در مسئلهٔ زمان‌بندی است. برای انجام این امر، بردار موقعیت ذرات، یک بردار n (تعداد کارها) بعدی در نظر گرفته می‌شود و سپس کارها به ترتیب صعودی مقدار موقعیتشان مرتب می‌شوند. این قاعده کمترین مقدار موقعیت[۳۲] (SPV) نامیده می‌شود (جایانتی و آنوسویا، ۲۰۱۷). پس از مشخص‌شدن ترتیب کارها، با توجه به اولویت کارها از قاعدهٔ First Fit برای تخصیص کارها به انباشته‌ها استفاده می‌شود و سپس به کمک الگوریتم IE و با جابه‌جایی توالی انباشته‌ها، کیفیت جواب‌ها تا حد امکان ارتقا می‌یابد.

در الگوریتم PSO استاندارد جواب‌های اولیه به‌صورت تصادفی ایجاد می‌شود؛ اما برای تضمین کیفیت جواب های اولیه، در این پژوهش از جواب الگوریتم ابتکاری استفاده می‌شود که در بخش ۳-۴ نیز به‌عنوان یکی از جواب‌ها توضیح داده شده است. در این زمینه نگاشت‌هایی وجود دارند که برای تبدیل توالی کارها به مقادیر پیوستهٔ موقعیت ذرات استفاده می‌شوند. نگاشت زیر برای این تبدیل به کار می‌رود و به کمک آن کاری که در اولویت jام قرار دارد، به مؤلفهٔ مکانی x_{0j}^0 نگاشت می‌شود:

$$x_{0j}^0 = x_{min} + \left(\frac{x_{max} - x_{min}}{n}\right) * \left(j + rand - 1\right) \quad (۱۷)$$

دیگر جزییات الگوریتم PSO_IE پیشنهادی شامل نحوهٔ به‌روزکردن سرعت و موقعیت ذرات همانند یک الگوریتم ازدحام ذرات در حالت عمومی آن است که به‌جهت اختصار از ذکر آنها پرهیز می‌شود.

۴- یافته‌ها

قدم اول برای تولید مسائل نمونه، تعیین مقادیر برای پارامترهای مورد نیاز این مسائل است. برای ایجاد این مسائل از روش به کار گرفته شدهٔ پارسا و همکاران (۲۰۱۷) استفاده شده است. در جدول ۲ نحوهٔ تولید مقادیر استفاده‌شده در این پژوهش نشان داده شده است. موعد تحویل مشترک کارها نیز با توجه به روش پیشنهادی هووگوین و ون د ولد (۱۹۹۱) به‌صورت تصادفی یکنواخت در بازهٔ $\left[0.2\sum p_j, 0.3\sum p_j\right]$ تولید شده است.

جدول ۲- پارامترهای مسائل نمونه

Table 2- Parameters of the instance problems

پارامتر	مقدار	کل مقادیر
تعداد کارها (n)	۱۰، ۲۰، ۴۰، ۶۰، ۱۰۰، ۲۰۰	۶
اندازهٔ کارها (s_j)	توزیع یکنواخت $[1, 40]$ توزیع یکنواخت $[10, 20]$ توزیع یکنواخت $[10, 30]$ توزیع یکنواخت $[1, 10]$	۴
زمان پردازش (p_j)	توزیع یکنواخت $[10, 50]$	۱
ظرفیت انباشته (B)	۴۰	۱

برای اعتبارسنجی الگوریتم برنامه‌ریزی پویا، جواب‌های به دست آمده از این روش با نتایج حل مدل ریاضی ارائه‌شدهٔ پارسا و همکاران (۲۰۱۷) مقایسه شده است. نکتهٔ حائز اهمیت آن است که احتمالاً مدل ریاضی در هر مرحله از تکرارهای خود، انباشته‌های جدیدی را برای بهینه‌سازی معیار عملکرد تولید می‌کند، اما انباشته‌ها در الگوریتم برنامه‌ریزی پویا ثابت‌اند و تنها یک مرتبه ساخته می‌شوند. از همین روی در جدول ۳ ظرفیت انباشته و اندازهٔ کارها یکسان فرض شده است تا مقایسه در حالت کاملاً برابر ارزیابی شود که البته این فرض تناقضی با هیچ‌یک از مفروضات مطرح‌شده ندارد و تنها حالت خاصی از مسئله است. در تمامی این نمونه‌ها، اندازهٔ انباشته و کارها برابر ۴۰ فرض شده است.

جدول ۳- اعتبارسنجی الگوریتم برنامه‌ریزی پویا

Table 3- Validation of the dynamic programming algorithm

اندازهٔ کارها	تعداد کارها	موعد تحویل	زمان پردازش کارها	مدل ریاضی	برنامه‌ریزی پویا
۴۰	۱۰	۵۰	۱۰, ۲۰,۱۱,۱۵,۱۲,۱۳,۱۸,۱۷,۱۶,۲۰	۳۹۴	۳۹۴
	۹	۳۵	۱۰, ۱۵,۱۳,۱۴,۱۴,۱۱,۱۰,۱۷,۱۶	۳۰۶	۳۰۶
	۸	۳۵	۲۰, ۲۰,۱۱,۱۲,۱۵,۱۴,۱۰,۱۲	۲۴۴	۲۴۴
	۷	۳۰	۱۵, ۱۳,۱۲,۱۱,۱۰,۱۹,۱۷	۱۸۹	۱۸۹
	۶	۲۱	۱۸, ۱۷,۱۳,۱۲,۱۲,۱۹	۱۸۲	۱۸۲

در نتایج حاصل از جدول ۳ نشان داده شده است که در صورت یکسان‌بودن انباشته‌ها، جواب الگوریتم برنامه ریزی پویا دقیقاً با جواب حاصل از حل مدل ریاضی برابر است.

در ادامه نتایج حاصل از حل مسائل نمونه و مقایسهٔ جواب‌های به دست آمده از الگوریتم‌های پیشنهادی با الگوریتم برنامه‌ریزی پویا ارائه می‌شود. به‌منظور ارزیابی الگوریتم‌های پیشنهادی از شاخص درصد انحراف نسبی (RPD) مطابق رابطهٔ (۱۸) استفاده شده است.

$$\mathrm{RPD} = \frac{\boldsymbol{ET_{Sol}} - \boldsymbol{ET_{DP}}}{\boldsymbol{ET_{DP}}} \times 100 \qquad (۱۸)$$

در رابطهٔ فوق، $\boldsymbol{ET_{Sol}}$ نشان‌دهندهٔ مقدار تابع هدف جواب حاصل از الگوریتم پیشنهادی و $\boldsymbol{ET_{DP}}$ بیانگر مقدار تابع هدف به دست آمده از الگوریتم برنامه‌ریزی پویاست. در هر رده دو نمونه و درمجموع ۶۰ نمونه مسئله به‌صورت تصادفی تولید و به کمک الگوریتم‌های پیشنهادی حل شده است. عملکرد الگوریتم ابتکاری (HA_IE) و الگوریتم ترکیبی ازدحام ذرات (PSO_IE) به‌صورت مجزا، نسبت‌به الگوریتم برنامه‌ریزی پویا (DP) مقایسه و نتایج در جدول ۴ گزارش شده است.

جدول ۴- سنجش عملکرد الگوریتم‌های پیشنهادی

Table 4- Evaluating the performance of the proposed algorithms

اندازهٔ کارها	تعداد کارها	متوسط نتایج			درصد انحراف نسبی (RPD)	
		DP	HA_IE	PSO_IE	HA_IE	PSO_IE
$[1, 40]$	۲۰	۱۴۷۶	۱۴۸۴	۱۴۷۹	۰/۵	۰/۲
	۴۰	۷۳۷۵	۷۶۵۸	۷۴۳۲	۳/۸	۰/۷
	۶۰	۱۶۰۱۲	۱۶۴۵۸	۱۶۲۵۰	۲/۷	۱/۴
	۱۰۰	۱۹۸۵۲	۲۰۰۲۶	۱۹۹۳۷	۰/۸	۰/۴
	۲۰۰	۲۰۷۳۳۹	۲۰۸۹۲۲	۲۰۸۵۵۰	۰/۷	۰/۵
$[10, 20]$	۲۰	۶۹۳	۷۱۶	۶۹۳	۳/۳	۰/۰
	۴۰	۴۵۰۵	۴۵۲۸	۴۵۱۰	۰/۵	۰/۱
	۶۰	۱۰۹۵۰	۱۱۱۹۲	۱۱۰۴۲	۲/۲	۰/۸
	۱۰۰	۲۸۸۹۰	۲۹۲۲۹	۲۹۱۰۳	۱/۱	۰/۳
	۲۰۰	۱۳۴۶۲۳	۱۳۵۸۸۸	۱۳۵۳۹۵	۰/۹	۰/۵
$[10, 30]$	۲۰	۲۱۹۵	۲۲۷۹	۲۲۳۶	۳/۸	۱/۸
	۴۰	۸۸۸۹	۹۰۹۶	۸۹۹۸	۲/۳	۱/۲
	۶۰	۲۰۳۴۳	۲۰۷۴۹	۲۰۵۸۹	۱/۹	۱/۲
	۱۰۰	۴۵۷۷۳	۴۶۵۳۶	۴۶۲۳۴	۱/۶	۱/۰
	۲۰۰	۲۰۶۸۶۱	۲۱۹۵۵۱	۲۱۸۸۳۰	۶/۱	۵/۷
$[1, 10]$	۲۰	۸۹۰	۸۹۰	۸۹۰	۰/۰	۰/۰
	۴۰	۲۲۴۶	۲۲۵۰	۲۲۴۶	۰/۱	۰/۰
	۶۰	۴۹۵۵	۴۹۷۵	۴۹۶۳	۰/۴	۰/۱
	۱۰۰	۸۶۶۴	۸۷۱۱	۸۶۸۱	۰/۵	۰/۱
	۲۰۰	۱۰۵۷۴	۱۰۸۴۴	۱۰۷۷۸	۲/۵	۲/۰

زمان حل الگوریتم‌های پیشنهادی برحسب ثانیه، به تفکیک در جدول ۵ ارائه شده است.

جدول ۵- زمان انجام محاسبات با الگوریتم‌ها برحسب ثانیه

Table 5- Computational time of the algorithms (s)

اندازهٔ کارها	تعداد کارها	الگوریتم HA_IE	الگوریتم PSO_IE	الگوریتم برنامه‌ریزی پویا
$[1, 40]$	۲۰	۰/۰۲۸	۰/۰۳	۱/۲۴۳
	۴۰	۰/۰۵۷	۰/۰۶۴	۲/۵۰۷
	۶۰	۰/۵۴	۰/۶	۳/۸۵۲
	۱۰۰	۰/۶	۰/۷	۸/۹۶۷
	۲۰۰	۱/۶۲	۲	۷۰/۵۵۱
$[10, 20]$	۲۰	۰/۰۲۶	۰/۰۳۵	۰/۸۸
	۴۰	۰/۰۴۷	۰/۰۵۲	۰/۹۲۸
	۶۰	۰/۰۸	۰/۰۹۴	۱/۲۷۴
	۱۰۰	۰/۱۶۴	۰/۱۸۲	۶/۷۶
	۲۰۰	۰/۸۷۴	۰/۹	۴۸/۵۸۵
$[10, 30]$	۲۰	۰/۰۳۵	۰/۰۵	۰/۳۷۷
	۴۰	۰/۶۰۳	۰/۰۷۱	۷/۵۴۶
	۶۰	۰/۱۵۶	۰/۲۱۲	۱۶/۰۳۱
	۱۰۰	۰/۲۱۸	۰/۲۷	۲۳/۴۸۵
	۲۰۰	۱/۵۷۸	۱/۸۱	۶۶
$[1, 10]$	۲۰	۰/۰۰۷	۰/۰۰۸۳	۰/۲۴۸
	۴۰	۰/۰۱۶	۰/۰۲۱	۰/۳۴۳
	۶۰	۰/۰۳۱	۰/۰۴۶	۰/۷۸
	۱۰۰	۰/۰۷	۰/۰۷۸	۲/۸۴۲
	۲۰۰	۰/۲۳۳	۰/۲۷۴	۷/۲۹

۵- بحث

برای مقایسهٔ آماری نتایج به دست آمده از حل مسائل مختلف به‌وسیلهٔ سه الگوریتم پیشنهادی، که در جدول ۴ نشان داده شده است، از آزمون تحلیل واریانس یک طرفه استفاده می‌کنیم. بر این اساس، چنانچه مقدار عبارت P-value، برای تک عامل بررسی‌شده (هر الگوریتم یک سطح از عامل فرض می‌شود) از سطح معناداری آزمون ($\alpha = ۰/۰۵$) کمتر باشد، فرض صفر آزمون یعنی برابری میانگین نتایج به دست آمده از حل مسئله به سه روش، رد می‌شود و در غیر این صورت دلیلی بر تفاوت نتایج الگوریتم‌ها وجود نخواهد داشت. برای انجام این آزمون، نرم‌افزار Minitab 17 به کار گرفته و نتایج حاصل در جدول ۶ گزارش شده است.

جدول ۶- تحلیل واریانس نتایج الگوریتم‌ها به ازای $\alpha = ۰/۰۵$

Table 6- Analysis of variance based on results of algorithms ($\alpha = 0.05$)

منبع تغییرات	درجهٔ آزادی (DF)	مجموع مربعات خطا (Adj SS)	میانگین مربعات خطا (Adj MS)	F-Value	P-Value
الگوریتم	۲	۱۰۲۲۸۳۸۰	۵۱۱۴۱۹۰	۰/۰۰	۰/۹۹۹
خطا	۵۷	۲/۵۲۲۱۲E+۱۱	۴۴۲۴۷۶۸۰۳۵		
کل	۵۹	۲/۵۲۲۲۲E+۱۱			

با توجه به نتایج آماری به دست آمده از جدول ۶، به‌وضوح مشخص می‌شود که فرض برابری میانگین حاصل از حل مسائل با استفاده از سه الگوریتم ردشدنی نیست. بر همین اساس الگوریتم‌های پیشنهادی، عملکرد مشابهی با الگوریتم برنامه‌ریزی پویا دارند.

تحلیل نتایج مربوط به زمان حل الگوریتم‌ها، که در جدول ۵ ارائه شد، نیز حاکی از آن است که حل مسئله به‌وسیلهٔ الگوریتم‌های HA_IE و PSO_IE در بیشتر موارد در کسری از ثانیه انجام‌شدنی است. هرچند الگوریتم برنامه‌ریزی پویا به تلاش محاسباتی بیشتری برای یافتن جواب بهینه نیاز دارد و زمان حل با این الگوریتم هنگامی که تعداد انباشته‌ها زیاد شود، به‌صورت چشمگیری افزایش می‌یابد. بر همین اساس و با توجه به نتایج آماری جدول ۶، الگوریتم‌های ابتکاری پیشنهادی با صرف زمان کم‌تر از عملکرد مناسبی برخوردارند و در عمل از این الگوریتم‌ها برای کاربردهای واقعی و در ابعاد بزرگ استفاده می‌شود.

در ادامه تأثیر روش انباشته‌سازی را تحلیل می‌کنیم. روش استفاده‌شده به این شرح است که ابتدا مدل ریاضی بخش ۳-۲ حل می‌شود و انباشته‌هایی با حداقل زمان تکمیل آخرین انباشته ($\sum p_b$) تعیین می‌شوند، سپس با استفاده از الگوریتم برنامه‌ریزی پویا، توالی بهینهٔ این انباشته‌ها به دست می‌آید. در مقابل از الگوریتم ابتکاری ارائه‌شده در بخش ۳-۱ برای تشکیل انباشته‌ها استفاده و نتایج با حالت قبل مقایسه می‌شود تا میزان تفاوت این دو رویکرد مشخص شود. برای مقایسهٔ تأثیر نحوهٔ انباشته‌سازی و همچنین تغییر اندازهٔ کارها بر جواب مسئله، آزمایشی با شرایط جدول ۷ انجام شده است. در این آزمایش، تعداد ۲۰ مسئلهٔ تصادفی به‌وسیلهٔ الگوریتم برنامه‌ریزی پویا حل شده است. برای سنجش اثربخشی این عوامل و برهم‌کنش میان آنها از تحلیل واریانس دو طرفه استفاده شده است.

جدول ۷- مقادیر پارامترهای مسائل نمونه

Table 7- Parameters value of the instance problems

پارامتر	مقدار
تعداد کل کار (n)	۲۰
اندازهٔ کارها (s_j)	توزیع یکنواخت $[۱, ۱۰]$، توزیع یکنواخت $[۱۰, ۲۰]$
زمان پردازش کارها (p_j)	توزیع یکنواخت $[۱, ۱۰]$
ظرفیت انباشته (B)	۴۰
نحوهٔ انباشته‌سازی	روش ۱. استفاده از مدل ریاضی، روش ۲. استفاده از الگوریتم ابتکاری انباشته‌سازی
سطح معناداری (α)	۰/۰۵

نتایج حاصل از تحلیل آماری، در جدول ۸ نشان داده شده است. در خروجی نرم‌افزار، اگر مقدار مشخص‌شده با P-Value برای هر عامل و اثر متقابل آنها کمتر از مقدار معناداری ($\alpha = ۰/۰۵$) باشد، آنگاه فرض صفر آزمون یعنی اثر اصلی حاصل از تغییر عوامل (نحوهٔ انباشته‌سازی و اندازهٔ کارها) و اثر متقابل آنها بر مقدار پاسخ (مقدار تابع هدف) در آزمون رد می‌شود.

براساس نتایج به دست آمده از تحلیل واریانس دوطرفه، تغییر نحوهٔ انباشته‌سازی با دو روش یادشده، دارای تأثیر کمی در جواب به دست آمده است. به بیان دیگر در نمونه‌های آزمایش‌شده، انباشته‌های ساخته‌شده براساس قاعدهٔ

LPT، تفاوت چشم‌گیری با انباشته‌های تولیدشده به کمک مدل ریاضی معرفی‌شده ندارند؛ اما در مقابل تغییر اندازهٔ کارها قویاً بر جواب مؤثر است. همچنین اثر متقابل میان این دو عامل نیز وجود ندارد. بنابراین الگوریتم ابتکاری پیشنهادی برای انباشته‌سازی از کارایی لازم برخوردار است و در ابعاد بزرگ مسئله، که امکان حل دقیق آن وجود ندارد، به‌عنوان یک جایگزین مناسب از آن استفاده می‌شود.

جدول ۸- نتایج آزمون آماری برای سنجش تأثیر نحوهٔ انباشته‌سازی و اندازهٔ کارها به ازای α = ۰/۰۵

Table 8- Results of statistical test for evaluating the effect of batching and job size ($\alpha = 0.05$)

منبع تغییرات	درجهٔ آزادی (DF)	مجموع مربعات خطا (SS)	میانگین مربعات خطا (MS)	F-Value	P-Value
نحوهٔ انباشته‌سازی	۱	۴۶۱	۴۶۱	۰/۴۱	۰/۵۳۱
اندازهٔ کارها	۱	۹۲۴۸۰	۹۲۴۸۰	۸۲/۴۸	۰/۰۰۰
نحوهٔ انباشته‌سازی*اندازهٔ کارها	۱	۳	۳	۰/۰۰	۰/۹۵۸
خطا	۱۶	۱۷۹۴۰	۱۱۲۱		
کل	۱۹	۱۱۰۸۸۴			

۶- نتیجه‌گیری

در این پژوهش، مسئلهٔ زمان‌بندی یک ماشین پردازندهٔ انباشته با اندازهٔ کارهای غیریکسان و معیار کمینه‌سازی مجموع وزنی تأخیر و تعجیل کارها با در نظر گرفتن موعد تحویل نزدیک بررسی شد. همچنین الگوریتمی مبتنی بر رویکرد برنامه‌ریزی پویا توسعه و الگوریتم‌های ابتکاری مبتنی بر بهبود فردی برای حل مسئله پیشنهاد شد. همچنین روش ترکیبی مبتنی بر الگوریتم ازدحام ذرات برای حل مسئله در ابعاد بزرگ توسعه داد شد. در پایان با بررسی نمونهٔ مسئله‌های متفاوت، عملکرد الگوریتم‌ها با یکدیگر مقایسه شد. بررسی نتایج به دست آمده از حل مسائل نمونه، عملکرد خوبی برای الگوریتم‌های پیشنهادی در یافتن جواب مسئله، در زمان پذیرفتنی را نشان می‌دهد. همچنین بررسی دو روش بررسی‌شده برای انباشته‌سازی کارها، یکی مبتنی بر یک روش ابتکاری و دیگری به کمک حل یک مدل ریاضی، نشان داد که تفاوت چشمگیری بین این دو روش وجود ندارد. بنابراین در صورت لزوم بدون از دست رفتن کیفیت جواب، از روش ابتکاری استفاده می‌شود که زمان حل کوتاه‌تری دارد. همچنین یک الگوریتم برنامه‌ریزی پویا برای تعیین توالی انباشته‌ها با هدف تحقق تولید بهنگام ارائه شد. در ادامه یک الگوریتم ابتکاری و یک الگوریتم فراابتکاری بر مبنای الگوریتم ازدحام ذرات برای حل کامل مسئله ارائه شد. الگوریتم برنامه‌ریزی پویا به تلاش محاسباتی زیادی نیاز دارد و زمان حل با این الگوریتم هنگامی که تعداد انباشته‌ها زیاد شود، به‌صورت چشمگیری افزایش می‌یابد. هرچند نتایج به دست آمده حاکی از آن است که الگوریتم‌های ابتکاری پیشنهادی با صرف زمان کم‌تر از عملکرد مناسبی برخوردارند و در عمل از این الگوریتم‌ها برای کاربردهای واقعی و در ابعاد بزرگ استفاده می‌شود. متوسط انحراف نسبی الگوریتم ازدحام ذرات پیشنهادی کمتر از ۱درصد و مقدار این شاخص برای الگوریتم ابتکاری ارائه‌شده ۱/۷۸درصد است.

حوزهٔ زمان‌بندی دربارهٔ ماشین‌های پردازندهٔ انباشته و ادغام آن با موضوع تولید بهنگام، با توجه به نیاز روزافزون صنایع به افزایش سرعت و کاهش هزینه ها ازجمله حوزه‌های مطالعاتی کاربردی و جذاب است. بر همین اساس زمینه‌های مطالعاتی برای انجام، با گسترش ایده‌هایی چون تغییر مفروضات اساسی مسئله نظیر وجود زمان دسترسی

برای کارها، تغییر نوع ماشین پردازندهٔ انباشته به حالت سری و موارد دیگر، پیشنهاد می‌شود. توسعهٔ روش‌هایی برای یافتن حد پایین مسئله و سپس استفاده از این روش‌ها در بدنهٔ الگوریتم‌های حل دقیق مسئله، یکی دیگر از زمینه‌های مطالعاتی برای آینده است. در نظر گرفتن تعرفه‌های متفاوت برای هزینه‌های انرژی طی ساعات مختلف روز، پیشنهاد کاربردی دیگری است که جای مطالعه دارد. همچنین بررسی مسئله با در نظر گرفتن الزامات زیست‌محیطی مانند کاهش انتشار دی اکسید کربن، ازنظر کاربردهای عملی جذاب است.

References

Baker, K. R., & Scudder, G. D. (1990). Sequencing with earliness and tardiness penalties: a review. ***Operations Research, 38***(1), 22-36. http://www.jstor.org/stable/171295

Brucker, P., Gladky, A., Hoogeveen, H., Kovalyov, M. Y., Potts, C. N., Tautenhahn, T., & Van De Velde, S. L. (1998). Scheduling a batching machine. ***Journal of scheduling***, ***1***(1), 31-54. https://doi.org/10.1002/(SICI)1099-1425(199806)1:1%3C31::AID-JOS4%3E3.0.CO;2-R

Fowler, J. W., & Mönch, L. (2022). A survey of scheduling with parallel batch (p-batch) processing. ***European Journal of Operational Research***, ***298***(1), 1-24. https://doi.org/10.1016/j.ejor.2021.06.012

Graham, R. L., Lawler, E. L., Lenstra, J. K., & Kan, A. R. (1979). Optimization and approximation in deterministic sequencing and scheduling: a survey. ***Annals of discrete mathematics***, ***5***(1), 287-326. https://doi.org/10.1016/S0167-5060(08)70356-X

Hall, N. G., Kubiak, W., & Sethi, S. P. (1991). Earliness–tardiness scheduling problems, II: Deviation of completion times about a restrictive common due date. ***Operations Research***, ***39***(5), 847-856. https://doi.org/10.1287/opre.39.5.847

Hoogeveen, H., & van de Velde, S. (1991). Scheduling around a small common due date. ***European Journal of Operational Research***, ***55***(2), 237-242. https://doi.org/10.1016/0377-2217(91)90228-N

Ikura, Y., & Gimple, M. (1986). Efficient scheduling algorithms for a single batch processing machine. ***Operations Research Letters, 5***(2), 61-65. https://doi.org/10.1016/0167-6377(86)90104-5

Jayanthi, S., & Anusuya, S. (2017). Minimization of Total Weighted Earliness and Tardiness using PSO for One Machine Scheduling. I***nternational Journal of Pure and Applied Mathematical Sciences***, ***10***(1) , 35-44.

Kong, M., Wang, W., Deveci, M., Zhang, Y., Wu, X., & Coffman, D. M. A. (2023). Novel carbon reduction engineering method-based deep Q-learning algorithm for energy-efficient scheduling on a single batch-processing machine in semiconductor manufacturing. ***International Journal of Production Research***, 1-24. https://doi.org/10.1080/00207543.2023.2252932

Kuo, I.-H., Horng, S.-J., Kao, T.-W., Lin, T.-L., Lee, C.-L., Terano, T., & Pan, Y. (2009). An efficient flow-shop scheduling algorithm based on a hybrid particle swarm optimization model. ***Expert systems with applications, 36***(3), 7027-7032. https://doi.org/10.1016/j.eswa.2008.08.054

Li, Z., Chen, H., Xu, R., & Li, X. (2015). Earliness–tardiness minimization on scheduling a batch processing machine with non-identical job sizes. ***Computers & Industrial Engineering***, ***87***(1), 590-599. https://doi.org/10.1016/j.cie.2015.06.008

Mönch, L., Unbehaun, R., & Choung, Y. I. (2006). Minimizing earliness–tardiness on a single burn-in oven with a common due date and maximum allowable tardiness constraint. ***OR Spectrum***, ***28***(2), 177-198. https://doi.org/10.1007/s00291-005-0013-4

Nearchou, A. C., & Omirou, S. L. (2013). A particle swarm optimization algorithm for scheduling against restrictive common due dates. I***nternational Journal of Computational Intelligence Systems, 6***(4), 684-699. https://doi.org/10.1080/18756891.2013.802874

Parsa, N. R., Karimi, B., & Husseini, S. M. (2017). Exact and heuristic algorithms for the just-in-time scheduling problem in a batch processing system. ***Computers & Operations Research***, ***80***(1), 173-183. https://doi.org/10.1016/j.cor.2016.12.001

Pessoa, A. A., Bulhões, T., Nesello, V., & Subramanian, A. (2022). Exact approaches for single machine total weighted tardiness batch scheduling. ***INFORMS Journal on Computing, 34***(3), 1512-1530. https://doi.org/10.1287/ijoc.2021.1133

Queiroga, E., Pinheiro, R. G. S., Christ, Q., Subramanian, A., & Pessoa, A. A. (2021). Iterated local search for single machine total weighted tardiness batch scheduling. ***Journal of Heuristics, 27***(3), 353-438. https://doi.org/10.1007/s10732-020-09461-x

Sen, T., Sulek, J. M., & Dileepan, P. (2003). Static scheduling research to minimize weighted and unweighted tardiness: a state-of-the-art survey. ***International journal of production economics, 83***(1), 1-12. https://doi.org/10.1016/S0925-5273(02)00265-7

Tian, Z., & Zheng, L. (2024). Single machine parallel-batch scheduling under time-of-use electricity prices: New formulations and optimisation approaches. ***European Journal of Operational Research, 312***(2), 512-524. https://doi.org/10.1016/j.ejor.2023.07.012

Trindade, R. S., de Araújo, O. C. B., & Fampa, M. (2021). Arc-flow approach for single batch-processing machine scheduling. ***Computers & Operations Research***, ***134***, 1-16. https://doi.org/10.1016/j.cor.2021.105394

Trindade, R. S., de Araújo, O. C. B., Fampa, M. H. C., & Müller, F. M. (2018). Modelling and symmetry breaking in scheduling problems on batch processing machines. I***nternational journal of production research***, ***56***(22), 7031-7048. https://doi.org/10.1080/00207543.2018.1424371

Yang, F., Davari, M., Wei, W., Hermans, B., & Leus, R. (2022). Scheduling a single parallel-batching machine with non-identical job sizes and incompatible job families. ***European Journal of Operational Research, 303***(2), 602-615. https://doi.org/10.1016/j.ejor.2022.03.027

Zhang, H., Wu, F., & Yang, Z. (2021). Hybrid approach for a single-batch-processing machine scheduling problem with a just-in-time objective and consideration of non-identical due dates of jobs. ***Computers & Operations Research***, ***128***, 105194. https://doi.org/10.1016/j.cor.2020.105194

Zheng, X., & Chen, Z. (2024). An improved deep Q-learning algorithm for a trade-off between energy consumption and productivity in batch scheduling. ***Computers & Industrial Engineering***, ***188***, 109925. https://doi.org/10.1016/j.cie.2024.109925

[1] Sen et al.
[2] Baker and Scudder
[3] Hoogeveen and van de Velde
[4] Hall et al.
[5] Nearchou and Omirou
[6] Particle Swarm Optimization (PSO)
[7] Jayanthi and Anusuya
[8] Ikura and Gimple
[9] Li et al.
[10] Parsa et al.
[11] Longest Processing Time (LPT)
[12] Zhang et al.
[13] Trindade et al.
[14] Queiroga et al.
[15] Pessoa et al.
[16] Time-indexed formulation
[17] Yang et al.
[18] Branch and price
[19] Kong et al.
[20] Tian and Zheng
[21] Zheng and Chen
[22] Fowler and Mönch
[23] Loose due date
[24] Tight due date
[25] Graham et al.
[26] Brucker et al.

[27] Mönch et al.
[28] Straddling Batch
[29] Symmetry
[30] Individual Enhancement
[31] Kuo et al.
[32] Shortest Position Value (SPV)