

ارائه الگوریتم‌های کارآمد برای حل مسأله زمانبندی جریان کارگاهی انعطاف‌پذیر با ماشین‌های موازی غیرمرتبط و زمان‌های راه‌اندازی وابسته به توالی با هدف کمینه‌سازی مجموع زودکرد و دیرکرد

سعیده غلامی^{۱*}، فرزانه رجایی ایبانه^۲

۱_ استادیار دانشکده مهندسی صنایع، دانشگاه صنعتی خواجه نصیرالدین طوسی، تهران، ایران

۲_ کارشناسی ارشد دانشکده مهندسی صنایع، دانشگاه صنعتی خواجه نصیرالدین طوسی، تهران، ایران

چکیده

در این مقاله، یک مدل ریاضی مبتنی بر برنامه‌ریزی عدد صحیح آمیخته برای مسأله زمانبندی جریان کارگاهی انعطاف‌پذیر با ماشین‌های موازی نامرتبط و زمان‌های راه‌اندازی وابسته به توالی با هدف کمینه‌سازی مجموع زودکرد و دیرکرد، ارائه شده است. به علت پیچیدگی این مسأله، برای حل مسائل با ابعاد بزرگ، از الگوریتم‌های فراابتکاری استفاده شده است؛ در این پژوهش یک الگوریتم مبتنی بر شبیه‌سازی تبرید و الگوریتم دیگری مبتنی بر بهینه‌سازی ذرات ارائه شده است، و برای تنظیم پارامترهای الگوریتم‌های پیشنهادی از روش طراحی آزمایش‌های تاگوچی استفاده شده است. برای تحلیل عملکرد الگوریتم‌های حل، چهل‌ویک مسأله نمونه با ابعاد مختلف طراحی، و هرکدام ده مرتبه اجرا شده است. با توجه به تحلیل نتایج آزمایش‌های محاسباتی زمان حل الگوریتم مبتنی بر بهینه‌سازی ذرات کمتر بوده است، ولی کیفیت جواب حاصل از الگوریتم مبتنی بر شبیه‌سازی تبرید بهتر از الگوریتم مبتنی بر بهینه‌سازی ذرات بوده است؛ به طور متوسط میزان درصد انحراف نسبی، نتایج آزمایش‌های محاسباتی الگوریتم مبتنی بر بهینه‌سازی ذرات ۴/۴ درصد، و الگوریتم مبتنی بر شبیه‌سازی ۲/۳ درصد بوده است.

واژه‌های کلیدی: الگوریتم بهینه‌سازی انبوه ذرات، الگوریتم شبیه‌سازی تبرید، دیرکرد، زمان بندی جریان کارگاهی انعطاف‌پذیر، زمان‌های راه‌اندازی وابسته به توالی، زودکرد.

مقدمه

در مسأله جریان کارگاهی انعطاف‌پذیر کارها باید در چند مرحله پردازش شود و حداقل یکی از مراحل بیش از دو ماشین موازی دارد. ماشین‌های موازی می‌توانند یکسان^۱، یکنواخت^۲ و یا غیرمرتبط^۳ باشند. در ماشین‌های موازی یکسان سرعت پردازش کارها روی همه ماشین‌های یکسان است؛ اگر کار J_m ، روی هر یک از ماشین‌ها انجام شود، زمان پردازش آن برابر است با P_j . در ماشین‌های موازی یکنواخت، ماشین‌هایی مشابه با سرعت پردازش متفاوت، به کار گرفته می‌شود؛ سرعت پردازش ماشین i برابر V_i است، اگر کار J_m ، به طور کامل روی ماشین i انجام شود، زمان پردازش آن برابر است با $P_{ij} = P_j / V_i$ ، در ماشین‌های موازی غیر مرتبط ماشین‌هایی متفاوت به کار گرفته می‌شود و ماشین i کار J_m را با سرعت V_{ij} پردازش می‌کند؛ اگر کار J_m ، به طور کامل روی ماشین i انجام شود، زمان پردازش آن برابر است با $P_{ij} = P_j / V_{ij}$.

در بیشتر پژوهش‌های پیشین ماشین‌ها به صورت یکسان در نظر گرفته شده‌اند. در مسائل دنیای واقعی، معمولاً ماشین‌های جدید را به صورت موازی با ماشین‌های قدیمی قرار می‌دهند تا ظرفیت تولید افزایش یابد. در این نوشتار زمان‌های آماده‌سازی وابسته به توالی عملیات است، که در بسیاری از مسائل توالی عملیات در صنایع نساجی، تولید نیمه‌رساناها، تولید کاشی و سرامیک و صنایع رنگ کاربرد دارد. تابع هدف مجموع زودکرد و دیرکرد که هزینه‌های نگهداری و هزینه‌های عدم تحویل به موقع محصول به مشتری را کاهش می‌دهد.

مسأله جریان کارگاهی انعطاف‌پذیر با دو مرحله با تعدادی ماشین موازی یکسان در مرحله اول و یک ماشین در مرحله دوم و با تابع هدف کمینه‌سازی بیشترین زمان تکمیل را گاپتا و همکاران (۱۹۹۷) بررسی کردند. بوتانگولاز (۲۰۰۰) مسأله جریان کارگاهی انعطاف‌پذیر با محدودیت پیشین‌سازی بین کارها و محدودیت زمان‌های اتلاف بین مراحل و با تابع هدف کمینه‌سازی بیشترین زمان تاخیر بررسی و شش روش ابتکاری جدید ارائه کردند. تران و مینگ ننگ (۲۰۱۱) مسأله جریان کارگاهی انعطاف‌پذیر را با محدودیت ظرفیت انبارهای میانی و بدون در نظر گرفتن آن با هدف‌های کمینه‌سازی بیشترین زمان تکمیل، مجموع موزون زمان در جریان ساخت و مجموع موزون زمان‌های تاخیر بررسی و الگوریتم جریان آب^۴ را ارائه کردند. لیو و چانگ (۲۰۰۰) یک روش دقیق و یک روش ابتکاری برای حل مسأله جریان کارگاهی انعطاف‌پذیر با زمان‌های آماده‌سازی وابسته به توالی ارائه کردند و اسکین (۲۰۰۳) این مسأله را با هدف کمینه‌سازی بیشترین زمان تکمیل بررسی کرد، و یک مدل عدد صحیح، چند روش ابتکاری و یک روش الگوریتم ژنتیک با کلیدهای تصادفی ارائه کرد. نادری و همکاران (۲۰۰۹) برای مسأله جریان کارگاهی انعطاف‌پذیر چند هدفه با زمان‌های آماده‌سازی وابسته به توالی و زمان حمل و نقل، الگوریتمی ترکیبی مبتنی بر شبیه‌سازی تبرید و یک روش جستجوی ساده ارائه کردند.

مسأله با فرض ماشین‌های موازی غیرمرتبط، در مسأله جریان کارگاهی انعطاف‌پذیر، از مسائلی است که کمتر مطالعه شده است؛ البته می‌توان به تحقیقات جانگواتانیک و همکاران (۲۰۰۵، ۲۰۰۸، ۲۰۰۹)

می‌شود و در بخش چهارم نتایج محاسباتی الگوریتم‌های ارائه شده، گزارش می‌شود. در نهایت نتایج این پژوهش در بخش پنج جمع‌بندی می‌شود.

۱- تشریح مسأله زمانبندی جریان کارگاهی انعطاف‌پذیر با زمان‌های آماده‌سازی وابسته به توالی

در این پژوهش مسأله زمانبندی در یک سیستم جریان تولید کارگاهی انعطاف‌پذیر بررسی می‌شود که در آن هر مرحله حداقل یک ماشین قرار دارد و به‌صورت موازی هستند و کارها در دو مرحله پردازش می‌شوند؛ زمان‌های آماده‌سازی، وابسته به توالی است، و بعد از اتمام پردازش هر کار در هر مرحله و شروع کار بعدی در نظر گرفته می‌شود که وابسته است به ترتیب این دو کار و ماشینی که پردازش روی آن انجام می‌شود. ماشین‌های هر مرحله، غیرمرتبط هستند. پارامترها و متغیرهای مسأله قطعی است و تمام داده‌ها از پیش تعیین شده و ثابت هستند. تابع هدف کمینه‌سازی مجموع زودکرد و دیرکرد است که باعث می‌شود هزینه‌های نگهداری و هزینه‌های عدم تحویل به موقع محصول به مشتری، کاهش یابد.

مفروضات مسأله به شرح ذیل است:

- وقفه مجاز نیست.
- زمانی که کارها پردازش خود را روی ماشین‌ها آغاز می‌کنند، قبل از این که به مرحله بعد بروند باید پردازش خود را به اتمام برسانند.
- پردازش یک کار در هر مرحله نمی‌تواند به چند قسمت تقسیم شود و هر کار باید فقط روی یک ماشین پردازش انجام شود.

اشاره کرد. همچنین، یواریما و همکاران (۲۰۰۹) مسأله جریان کارگاهی انعطاف‌پذیر با ماشین‌های موازی غیرمرتبط، زمان‌های راه‌اندازی وابسته به توالی، محدودیت زمان‌های در دسترس بودن ماشین‌ها و محدودیت انبارهای میانی را با تابع هدف کمینه‌سازی مجموع زمان‌های تکمیل بررسی و برای حل از الگوریتم ژنتیک استفاده کردند.

جانیاک و همکاران (۲۰۰۷) مسأله جریان کارگاهی انعطاف‌پذیر با تابع هدف مجموع زودکرد و دیرکرد با و زمان انتظار را در نظر گرفتند و سه الگوریتم فراابتکاری و سه الگوریتم سازنده را برای آن ارائه کردند. بهنامیان و زندیه (۲۰۱۱) مسأله جریان کارگاهی انعطاف‌پذیر با محدودیت زمان انتظار و تابع هدف مجموع زودکرد و دیرکرد را بررسی و الگوریتم رقابت استعماری گسسته را برای حل ارائه کردند.

تابع هدف مجموع زودکرد و دیرکرد تنها در مسائل جریان کارگاهی انعطاف‌پذیر با ماشین‌های یکسان در نظر گرفته شده است. در این پژوهش این تابع هدف در مسأله جریان کارگاهی با ماشین‌های موازی غیرمرتبط بررسی می‌شود.

در این نوشتار یک مسأله خاص از مسائل جریان کارگاهی انعطاف‌پذیر با زمان‌های راه‌اندازی وابسته به توالی، ماشین‌های غیرمرتبط و تابع هدف مجموع زودکرد و دیرکرد بررسی می‌شود، و ساختار آن به شرح ذیل است: در بخش دوم مسأله زمانبندی جریان کارگاهی انعطاف‌پذیر با زمان‌های آماده‌سازی وابسته به توالی تشریح می‌شود سپس در بخش سوم روش تحقیق و مدل‌سازی مسأله ارائه شده است که برای حل مسأله مورد نظر الگوریتم‌های مبتنی بر شبیه‌سازی تیرید و بهینه‌سازی انبوه ذرات ارائه

۲-۱- پارامترها و متغیرهای مدل

برای بیان مسأله و مدل‌سازی آن از پارامترها و متغیرهای زیر استفاده می‌شود:

J : مجموعه کارها
 S : مجموعه مراحل
 M_s : تعداد ماشین در مرحله S
 d_j : موعد تحویل
 P_{sjm} : زمان پردازش کار j روی ماشین m در مرحله S
 SUP_{skjm} : زمان آماده‌سازی کار k بعد از کار j روی ماشین m در مرحله S
 M : یک عدد بزرگ
 C_{sjm} : زمان تکمیل کار j روی ماشین m در مرحله S
 K_{sj} : زمان ترک کار j از مرحله S
 E_j : زودکرد کار j
 T_j : دیرکرد کار j

$$\left. \begin{array}{l} ۱ \text{ اگر کار } j \text{ روی ماشین } m \text{ در مرحله } S \text{ پردازش شود} \\ ۰ \text{ در غیر این صورت} \end{array} \right\} = X_{sjm}$$

$$\left. \begin{array}{l} ۱ \text{ اگر کار } j \text{ بلافاصله بعد از کار } k \text{ روی ماشین } m \text{ در} \\ \text{ } S \text{ پردازش شود.} \\ ۰ \text{ در غیر این صورت} \end{array} \right\} = W_{skjm}$$

- زمان در دسترس بودن کارها در مرحله اول برابر صفر است.
- قبل از شروع اولین کار در هر مرحله، زمانی به عنوان زمان آماده‌سازی وجود دارد.
- ماشین‌ها بدون هیچ خرابی در دسترس هستند.
- زمان پردازش هر مرحله از کارها از تقسیم زمان پردازش استاندارد بر سرعت پردازش هر یک از ماشین‌ها به دست می‌آید.

۲- روش تحقیق و مدل‌سازی مسأله

در این پژوهش برای تحلیل مناسب مسأله یک مدل برنامه‌ریزی عدد صحیح آمیخته ارائه می‌شود که با توجه به پیچیدگی محاسباتی آن، برای حل مسائل نمونه با ابعاد بزرگ، باید از روش‌های ابتکاری/فراابتکاری استفاده کرد؛ به همین علت در این پژوهش برای حل مسأله زمان‌بندی از الگوریتم‌های فراابتکاری مبتنی بر بهینه‌سازی انبوه ذرات و شبیه‌سازی تبرید استفاده شده است.

الگوریتم‌های پیشنهادی از دو بخش تشکیل شده است، در بخش اول هدف کمینه‌سازی بیشترین زمان تکمیل و دیرکرد است و در بخش دوم، هدف کمینه‌سازی مجموع زودکرد و دیرکرد، با تغییر زمان شروع و خاتمه کارها در مرحله آخر، است.

برای تعیین هزینه و زمان‌بندی کامل از الگوریتم پیشنهادی جانیاک (۲۰۰۷) استفاده می‌شود.

۲-۱-۱- مدل ریاضی

به علاوه زمان پردازش و زمان آماده‌سازی آن کار باشد.

عبارت‌های (۸) و (۹) مقدار k_{sj} را برابر زمانی قرار می‌دهد که کار J مرحله را ترک می‌کند. عبارت (۱۰) نشان می‌دهد که در هر مرحله هر کار فقط روی یک ماشین پردازش می‌شود. عبارت (۱۱) نشان‌دهنده این است که هر کار باید به دنبال کار دیگری مثل کار I فقط روی یک ماشین پردازش شود. عبارت (۱۲) نشان‌دهنده این است که ممکن است کاری قبل از کار دیگر روی یک ماشین برنامه‌ریزی شده یا نشده باشد ولی آن کار حداکثر با یک کار دیگر باید دنبال شود. در عبارت‌های (۱۳) و (۱۴) برای محاسبه زمان آماده‌سازی، کار صفر به عنوان اولین کاری است که روی تمام ماشین‌ها پردازش می‌شود

تابع هدف در این مسأله کمینه‌سازی مجموع دیرکرد و زودکرد است که در عبارت (۱) بیان می‌شود، و برای تعیین مقدارهای زودکرد و دیرکرد به صورت عدد صحیح در مدل ریاضی، عبارت‌های (۳)، (۴)، (۱۵) و (۱۶) در نظر گرفته می‌شود. عبارت (۲) رابطه دیرکرد و زودکرد را با موعد تحویل مشخص می‌کند. عبارت (۵) تضمین می‌کند که زمان پردازش هر کار بزرگتر یا مساوی زمان پردازش و زمان آماده‌سازی کار J در مرحله اول است. با در نظر گرفتن عبارت (۶) به دنبال آن هستیم که پردازش هیچ کدام از کارها روی یک ماشین همزمان نشود. عبارت (۷) تضمین می‌کند که زمان تکمیل هر کار بزرگتر یا مساوی زمان تکمیل آن کار در مرحله قبل

$$\text{Min}Z = \sum_{j=1}^J (E_j + T_j) \quad ..$$

$$\text{s.t.} \quad (1)$$

$$K_{sj} + E_j - T_j = d_j \quad , \forall s, s = S; \quad (2)$$

$$E_j \geq d_j - K_{sj} \quad , \forall j, s = S; \quad (3)$$

$$T_j \geq K_{sj} - d_j \quad , \forall j, s = S; \quad (4)$$

$$C_{sjm} \geq P_{sjm} + W_{sijm} * \text{sup}_{sijm} \quad , \forall j, i \neq j, m = 1, \dots, M_s, s = 1; \quad (5)$$

$$C_{sjm} - C_{sim} + M * (1 - W_{sijm}) \geq P_{sjm} + W_{sijm} * \text{sup}_{sijm} \quad , \forall s \neq 1, j, i \neq j, m = 1, \dots, M_s; \quad (6)$$

$$C_{sjm} - K_{(s-1)j} \geq P_{sjm} - M * (1 - X_{sjm}) + W_{sijm} * \text{sup}_{sijm} \quad , \forall s \neq 1, j, i \neq j, m = 1, \dots, M_s; \quad (7)$$

$$C_{sjm} \leq K_{sj} \quad , \forall s, j, m = 1, \dots, M_s; \quad (8)$$

$$C_{sjm} \geq K_{sj} - M * (1 - X_{sjm}) \quad , \forall s, j, m = 1, \dots, M_s; \quad (9)$$

$$\sum_{m=1}^{M_s} X_{sjm} = 1 \quad , \forall s, j \neq 0 \quad (10)$$

$$X_{sjm} - \sum_{i=1}^J W_{sijm} = 0 \quad , i \neq j, \forall s, j \neq 0, m = 1, \dots, M_s; \quad (11)$$

$$X_{sim} - \sum_{j=1}^J W_{sijm} \geq 0 \quad , i \neq j, \forall s, i, m = 1, \dots, M_s; \quad (12)$$

$$\sum_{i=1}^J W_{si0m} = 0 \quad , i \neq j, \forall s, j, m = 1, \dots, M_s; \quad (13)$$

$$X_{s0m} = 1 \quad , \forall s, j, m = 1, \dots, M_s; \quad (14)$$

$$T_j \geq 0 \quad (15)$$

$$E_j \geq 0 \quad (16)$$

۲-۲- الگوریتم‌های پیشنهادی برای حل مسأله

در اینجا با توجه به پیچیدگی الگوریتم حل، دستیابی به جواب بهینه، حتی برای مسائل با ابعاد کوچک، مستلزم صرف وقت زیادی خواهد بود؛ بنابراین در این شرایط بهتر است که از روش‌های ابتکاری/فراابتکاری برای حل مسأله استفاده شود. در این نوشتار از دو الگوریتم مبتنی بر شبیه‌سازی تبرید و الگوریتم مبتنی بر بهینه‌سازی انبوه ذرات استفاده می‌شود، و برای تعیین مقدار تابع هدف از الگوریتم ایجاد زمان‌بندی کامل استفاده می‌شود.

۲-۲-۱- الگوریتم پیشنهادی ایجاد زمان‌بندی کامل

ابتدا زمان در دسترسی کارها در مرحله اول را برابر صفر است.

$$R(s, j) = 0, s = 1, j = 1, \dots, J$$

با فرض مشخص بودن توالی مرحله اول، در مرحله دوم با استفاده از قاعده FIFO توالی کارها تعیین می‌شود.

$$AV(m \cdot s) = 0, m = \{1, \dots, M(s)\} \quad (17)$$

C_{sjm} ، زمان تکمیل کار j روی ماشین m در مرحله s و \sup_{s^*jm} زمان آماده‌سازی کار j روی ماشین m در مرحله s است. در صورتیکه کار قبل از کار j به کار j روی ماشین m تغییر یابد، زمان تکمیل هر کار روی هر ماشین به ترتیب زیر محاسبه می‌شود و ماشینی انتخاب می‌شود که کمترین زمان تکمیل را داشته باشد:

$$C_{sjm} = \max \{AV(m, s), r(s, j)\} + p_{sjm} + \sup_{s^*jm}; \quad (18)$$

ماشین انتخاب شده را m^* بنامید؛ سپس زمان در دسترس بودن ماشین m^* در مرحله s و زمان در دسترسی کار j در مرحله s به روز می‌شود. زمان تکمیل کار j ، C_j را برابر زمان تکمیل کار j روی ماشین m^* در مرحله s (C_{sjm^*}) است.

برای مرحله آخر و برای هر ماشین π : توالی کارهایی است، که قرار است روی ماشین m پردازش شوند، و B_j مجموعه کارهایی است، که بدون بیکاری و وقفه به صورت متوالی بر روی ماشین m بعد از کار j پردازش می‌شوند و Lb_j جایگاه آخرین کار در گروه B_j است و $EB_j \subseteq B_j$ زیرمجموعه‌ای از گروه B_j است که زودکرد دارند و SEB_j برابر کمترین مقدار زودکرد در گروه B_j و π_{mj} نشان‌دهنده j امین کار در توالی روی ماشین m است. $ID(B_j)$ نشان‌دهنده زمان بیکاری بعد از گروه B_j است و از رابطه (۱۹) محاسبه می‌شود:

$$ID(B_j) = st(\pi_{Lb_j+1}) - C(\pi_{Lb_j}) \quad (19)$$

st نشان دهنده زمان شروع کار و C نشان دهنده زمان تکمیل است.

در صورتی که دو شرط زیر برقرار باشد:

$$E(\pi_{mj}) > 0 \quad (1)$$

(۲) تعداد کارهای دارای زودکرد بیش از تعداد کارهای با دیرکرد باشد.

کار j ام و گروه B_j به اندازه t به سمت راست جابه‌جا می‌شود، که از رابطه (۲۰) به دست می‌آید:

$$t = \min \{SEB_j, E(\pi_{mj}), ID(B_j)\} \quad (20)$$

سپس گروه B_j ، مقادیر E ، SEB_j ، $ID(B_j)$ ، مجموع زودکرد و دیرکرد محاسبه شود. این روند تا زمانی که یکی از شرطها نقض شود، تکرار می‌شود. به این

د) مکانیزم کاهش دما و پذیرش جواب

مکانیزم کاهش دما با استفاده از عبارت (۲۱) تعیین می‌شود: که در آن α ضریب کاهش دما و ثابتی کمتر از یک است. n نشان دهنده تعداد دفعات کاهش دما است مکانیزم پذیرش جواب‌های تولید شده به این صورت است که اگر جواب بهتر شود آن جواب پذیرفته شود، در غیر این صورت با استفاده از عبارت (۲۲) مقدار y محاسبه شود.

$$y = e^{-\frac{\Delta Z}{T}} \quad (22)$$

سپس یک مقدار تصادفی تولید می‌شود و با مقدار y مقایسه می‌شود، اگر مقدار عدد تصادفی به دست آمده کمتر از y باشد جواب پذیرفته شود، در غیر این صورت جواب رد می‌شود.

در این عبارت ΔZ برابر اختلاف بین جواب جاری و جواب همسایگی است و T بیانگر دمای جاری سیستم است. الگوریتم با مقادیر ۱۰۰۰، ۱۵۰۰ و ۱۸۰۰ برای دمای اولیه و نرخ کاهش دما با مقادیر ۰/۱، ۰/۲ و $\left(\frac{T_f}{T_0}\right)^{\frac{1}{MaxIt}}$ تحلیل می‌شود. که $MaxIt$ حداکثر تعداد تکرارها است.

ه) جستجو محلی

در الگوریتم پیشنهادی برای ارتقاء جواب‌های به دست آمده از این الگوریتم از جستجوهای محلی تصادفی استفاده می‌شود. به این منظور از دو نوع جستجوی همسایگی بر روی بهترین جواب به دست آمده از الگوریتم شبیه سازی تبرید استفاده می‌شود. یک جستجو بر روی مرحله اول و سپس جستجو بر روی مرحله آخر. جستجو بر روی مرحله اول به این علت اهمیت دارد که کوچکترین تغییر در زمان‌بندی

ترتیب یک برنامه زمان‌بندی کامل که متناسب با تابع هدف است، به دست می‌آید.

$$T_i = \alpha * T_{i-1} \quad i = 1, \dots, n \quad (21)$$

۲-۲-۲- الگوریتم مبتنی بر شبیه‌سازی تبرید

برای حل مدل پیشنهادی از الگوریتم مبتنی بر شبیه‌سازی تبرید استفاده می‌شود (جانیاک و همکاران، ۲۰۰۷). در الگوریتم پیشنهادی، مراحل تعریف نحوه نمایش جواب، تولید جمعیت اولیه، جستجوی همسایگی، مکانیزم کاهش دما و پذیرش جواب و جستجوهای محلی، باید اجرا شوند:

الف) نحوه نمایش جواب

در اینجا برای نمایش جواب، توالی کارها در مرحله اول در نظر گرفته می‌شود.

ب) تولید جمعیت اولیه

برای تولید جمعیت اولیه سه روش تصادفی، مبتنی بر قاعده^۱ EDD مبتنی بر الگوریتم NEH بررسی می‌شود.

ج) جستجوی همسایگی

برای پیمایش در فضای شدنی باید جواب شدنی دیگری با تغییر جواب فعلی ایجاد شود که جواب همسایه نامیده می‌شود. برای به دست آوردن جواب همسایه روش‌های جابجایی^۷ و جاسازی^۸ بررسی می‌شود. طول جستجوی همسایگی برای مقادیر ۵، ۱۰ و ۱۵ بررسی می‌شود.

روش تولید جمعیت اولیه مبتنی بر EDD، حداکثر تکرار ها ۲۰، تعداد همسایگی ۱۰، دمای اولیه ۱۵۰۰، نرخ کاهش دما $(\frac{T_f}{T_0})^1/MaxIt$. در شکل ۱ شبه کد الگوریتم پیشنهادی مبتنی بر شبیه‌سازی تبرید ارائه شده است.

۲-۲-۳- الگوریتم پیشنهادی مبتنی بر بهینه‌سازی انبوه

ذرات

کندی و ابره‌ارت (۱۹۹۵) الگوریتمی برای بهینه‌سازی مسائل با تابع هدف غیر خطی و متغیر تصمیم پیوسته ارائه کردند. این الگوریتم بر این اصل استوار است که در هر لحظه هر ذره مکان خود را در فضای جستجو با توجه به بهترین مکانی که تا کنون در آن قرار گرفته است و بهترین مکانی که در کل همسایگی‌اش وجود دارد، تنظیم می‌کند. در این الگوریتم هر ذره در فضای جستجو به دنبال نقطه بهینه حرکت می‌کند و دارای سرعت است. مسائل زمان‌بندی از نوع مسائل گسسته هستند و الگوریتم بهینه‌سازی انبوه ذرات ماهیتاً پیوسته است؛ بنابراین، از روش ارائه شده در تاسگیتايرن و همکاران (۲۰۰۴) استفاده می‌شود. برای به کارگیری الگوریتم پیشنهادی، مراحل زیر باید اجرا شوند:

- تعریف نحوه نمایش جواب
- تولید جمعیت اولیه
- جستجو، تعیین موقعیت و سرعت ذرات
- عملگرهای جهش و تقاطع

مرحله اول می‌تواند تغییر قابل توجهی در زمان‌بندی کلی و مقدار تابع هزینه ایجاد کند. به این منظور در هر یک از جستجوهای محلی تصادفی، چهار همسایگی در نظر گرفته شده است:

یک ماشین و دو کار که بر روی آن پردازش می‌شوند به صورت تصادفی انتخاب شده و با هم جابه‌جا می‌شوند.

یک ماشین و یک کار که روی آن پردازش می‌شود انتخاب شده و کار در موقعیت دیگری به صورت تصادفی قرار می‌گیرد.

دو ماشین و دو کار که بر روی آن‌ها پردازش می‌شوند به صورت تصادفی انتخاب و دو کار با هم جابه‌جا می‌شوند.

یک کار از ماشین اول به طور تصادفی انتخاب شده و در موقعیتی که به طور تصادفی روی ماشین دوم انتخاب شده است، جایگزین می‌شود.

برای به دست آوردن برنامه زمان‌بندی کامل و تابع هدف از الگوریتم بخش ۳-۲-۱ استفاده می‌شود. به منظور تعیین مقدار مناسب پارامترهای الگوریتم‌های پیشنهادی از روش طراحی آزمایش‌های تاگوچی استفاده می‌شود؛ مزیت این روش این است که می‌توان با انجام آزمایش‌های کمتر، مقدار پارامترهای الگوریتم را طوری تنظیم کرد که میزان تغییرپذیری جواب به دست آمده از الگوریتم (متغیر پاسخ) کمینه شود.

با توجه به نتایج حاصل از طراحی آزمایش‌های تاگوچی پارامترهای الگوریتم پیشنهادی مبتنی بر شبیه‌سازی تبرید به شرح زیر است:

```

Initialize parameters  $T_0, MaxIt, MaxIterPerTemp, \alpha, T_f$ 
Set counter  $n = 0, k = 0$ 
Do(outside loop)
Generate initial solution  $X_0$ 
Do (inside loop)
Generate neighboring solution  $X_{n+1}$  by operation  $(X_n \xrightarrow{\text{neighborhood search}} X_{n+1})$ 
p
If  $F(X_{n+1}) < F(X_n)$  then  $X_{n+1} = X_n$ 
Else
Generate random  $y \sim u(0,1)$ 
If  $y < e^{\frac{-\Delta Z}{T_k}}$  then  $X_{n+1} = X_n$ 
End
Update  $X_{best}$ 
Set  $n = n+1$ 
Loop until  $(n \leq MaxIterPerTemp)$ 
 $T_{k+1} = \alpha T_k$ 
Loop until  $(k \leq MaxIt)$ 
Apply local search on  $X_{best}$ 
Update  $X_{best}$ 
    
```

شکل ۱- شبه کد الگوریتم مبتنی بر شبیه سازی تبرید.

الف) تعریف نحوه نمایش جواب ها

به منظور تبدیل مقادیر پیوسته به مقادیر گسسته از روش کوچکتین ارزش مکانی (SPV) استفاده می‌شود. هر ذره با یک بردار بیان می‌شود، که تعداد آرایه های این بردار برابر با تعداد کارها است. هر ذره بیانگر یک توالی از کارهاست و برای تعیین آن مقادیر تصادفی تولید شده برای هر یک از آرایه‌های این بردار به ترتیب صعودی مرتب می‌شوند. در

جدول ۱، مثالی از نحوه نمایش جواب برای پنج کار ارائه شده است؛ در این مثال به ازای هر کار یک مقدار تصادفی تولید شده است، که کوچکتین آنها مربوط به کار ۵ است، لذا این کار در اولین موقعیت برای پردازش قرار می‌گیرد؛ موقعیت سایر کارها به همین ترتیب تعیین می‌شود.

ذرات، و با مقادیر $(-1,1)$ ، $(-2j, 2j)$ و $(-4,4)$ برای اختلاف بین حداقل و حداکثر سرعت ذرات، تحلیل می‌شود.

C_1 و C_2 ضرایب یادگیری هستند. برای ایجاد توازن بین جستجوی محلی و جستجوی کلی، در محاسبه بردار سرعت از ضریب وزنی اینرسی (w) استفاده می‌شود. در این پژوهش این ضریب به صورت یک معادله خطی کاهشی به صورت $w(it+1) = w_{damp} * w(it)$ در نظر گرفته شده است.

الگوریتم با مقادیر ۱، $1/25$ و ۲ برای ضرایب یادگیری C_1 ، C_2 و مقادیر $0/85$ ، $0/9$ و $0/99$ برای w_{damp} و تعیین ضریب وزنی اینرسی، تحلیل می‌شود.

د) عملگرهای جهش و تقاطع

در الگوریتم پیشنهادی برای پیشگیری از قرار گرفتن در بهینه محلی از عملگر جهش و تقاطع که در الگوریتم ژنتیک به کار گرفته می‌شود، استفاده شده است.

یکی از انواع عملگرهای تقاطع، عملگر OPX است (جانگواتاناکیت؛ ۲۰۰۹). در این عملگر دو والد انتخاب می‌شود و دو نقطه به صورت تصادفی بر روی آنها انتخاب می‌شود، ژن‌های موجود بین این دو نقطه از والد اول به فرزند اول منتقل می‌شود، و مقدار بقیه موقعیت‌ها از والد دوم به ترتیب از چپ به راست، برای فرزند اول در نظر گرفته می‌شود؛ برای فرزند دوم نیز به صورت مشابه تعیین می‌شود (شکل ۲).

جدول ۱- نمایش جواب در الگوریتم مبتنی PSO

شماره کار	۱	۲	۳	۴	۵
	۱/۹	-۰/۲۲	۴/۱۰	۰/۰,۲۳	-۱/۲۳
توالی حاصل	۵	۲	۴	۳	۱

ب) تولید جمعیت اولیه

در این مرحله، جمعیت اولیه به صورت تصادفی تولید می‌شود. الگوریتم با مقادیر ۲۰، ۳۰ و ۴۰ برای اندازه جمعیت اولیه تحلیل می‌شود.

ج) جستجو، تعیین موقعیت و سرعت ذرات

هر ذره در جمعیت اولیه دارای سرعت صفر است. سرعت و موقعیت هر ذره با استفاده از عبارات‌های (۲۳) و (۲۴) در هر تکرار به روز می‌شود.

$$V(i, it+1) = w * V(i, it) + c1 * rand * (Pbest(i, it) - X(i, it)) + c2 * rand * (Gbest(i, it) - X(i, it)) \quad (23)$$

$$X(i, it+1) = X(i, it) + V(i, it+1) \quad (24)$$

مقادیر V و X در i و it به ترتیب نشان‌دهنده شماره ذره، شماره تکرار، بردار موقعیت ذره و بردار سرعت است. $rand$ یک مقدار تصادفی بین صفر و یک است. $Pbest$ نشان‌دهنده بهترین موقعیتی است که ذره تاکنون یافته است و $Gbest$ بهترین موقعیتی است که ذرات موجود تاکنون به آن دست یافته‌اند. هر دوی این مقادیر بر اساس تابع برازندگی تعیین می‌شوند.

برای تنظیم پارامترهای مربوط به موقعیت و سرعت ذرات الگوریتم با مقادیر $(0,1)$ ، $(j,-j)$ و $(0,4)$ برای اختلاف بین حداقل و حداکثر موقعیت

جدول ۲- جهش احیاء

توالی کارها	۵	۲	۴	۳	۱
	۱/۹	-۰/۲۲	۴/۱۰	۰/۲۳	-۱/۲۳
	۰/۳۲	-۰/۲۲	۴/۰۱	۱/۶۳	-۱/۲۳
توالی کارها بعد از جهش	۳	۲	۵	۴	۱

همچنین، الگوریتم با در نظر گرفتن مقادیر ۰/۱، ۰/۲ و ۰/۳ برای نرخ جهش و حالت‌های زیر برای تعیین نوع عملگر جهش تحلیل می‌شود:

- (۱) جهش معکوس / جابجایی
- (۲) جهش احیاء / معکوس
- (۳) جهش احیاء / جابجایی

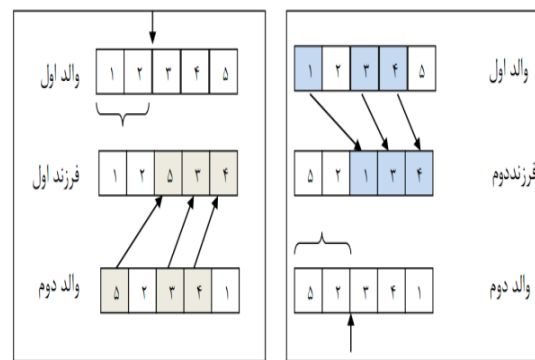
به منظور محاسبه تابع هزینه و به دست آوردن زمان‌بندی کامل از روش ابتکاری ارائه شده در بخش ۱، ۲، ۳ استفاده می‌شود. با توجه به این که بعد از چندین تکرار جواب‌ها به یک مقدار همگرا می‌شوند، حداکثر تعداد تکرارها به عنوان شرط توقف الگوریتم تعیین می‌شود. برای مسائل کوچک، متوسط و بزرگ به ترتیب حداکثر تعداد تکرارها ۱۰۰، ۲۰۰ و ۳۵۰ در نظر گرفته شده است.

با توجه به نتایج حاصل از طراحی آزمایش‌های تاگوچی پارامترهای الگوریتم پیشنهادی مبتنی بر بهینه‌سازی انبوه ذرات به شرح زیر است:

اندازه جمعیت اولیه ۴۰، عملگر جهش معکوس/احیاء، عملگر تقاطع OPX، نرخ جهش ۰/۳، نرخ تقاطع ۰/۷، C_1, C_2 برابر ۲، W_{damp} برابر ۰/۹۹، $X_{min} - X_{max}$ برابر $(-j, j)$ ، $V_{min} - V_{max}$ برابر $(-2j, 2j)$.

الگوریتم با در نظر گرفتن مقادیر ۰/۶، ۰/۷ و ۰/۸ برای نرخ تقاطع و حالت‌های زیر برای تعیین نوع عملگر تقاطع تحلیل می‌شود:

- (۱) OPX
- (۲) تک نقطه‌ای
- (۳) تک نقطه‌ای / OPX



شکل ۲- عملگر تقاطع تک نقطه‌ای اصلاح شده

در این الگوریتم عملگرهای جهش جابجایی، جهش معکوس و جهش احیاء (جدول ۲)، نیز بررسی می‌شود. در جهش جابجایی ابتدا دو کار به صورت تصادفی انتخاب می‌شوند و با هم جابجا می‌شوند؛ در جهش معکوس دو نقطه در طول کروموزوم به صورت تصادفی انتخاب می‌شوند و توالی کارها بین این دو نقطه معکوس می‌شود در جهش احیاء دو ژن به صورت تصادفی انتخاب می‌شود و دو عدد تصادفی جدید تولید شده و جایگزین اعداد قبلی می‌شود (جدول ۲). در این الگوریتم نیز از روش‌های جستجوی محلی به کار رفته در الگوریتم مبتنی بر شبیه‌سازی تبرید استفاده می‌شود.

مجموعه مسائل متوسط و بزرگ شامل ۱۲ ترکیب از تعداد کارها و تعداد مراحل است، که برای هر ترکیب ۲ مسأله نمونه ایجاد شده است. و هر مسأله تولید شده ۱۰ بار اجرا شده است. الگوریتم های پیشنهادی با متلب، کدنویسی شده، و همه مسائل بر روی رایانه پنتیوم آر دو هسته‌ای، با ۲،۶ گیگاهرتز و ۲ گیگابایت حافظه اجرا شده‌اند.

مسائل نمونه بیان شده در جدول ۳ ابتدا با نرم‌افزار GAMS به صورت دقیق، و سپس با الگوریتم‌های پیشنهادی حل شده‌اند و نتایج محاسباتی در جدول ۴ ارائه شده‌اند. در ستون‌های اول و دوم جدول ۴ شماره مسائل نمونه و اندازه مسأله (تعداد مرحله * تعداد کار) بیان شده است. به عنوان مثال اندازه مسأله نمونه ۱ معادل با (۳ کار * ۵ مرحله) است. زمان (ثانیه) و مقدار تابع هدف به دست آمده از حل مسائل نمونه با نرم‌افزار GAMS در ستون‌های سوم و چهارم، زمان (ثانیه) و مقدار تابع هدف به دست آمده از حل مسائل نمونه با استفاده از الگوریتم پیشنهادی مبتنی بر شبیه سازی تبرید در ستون‌های پنجم و ششم، ارائه شده است که ستون‌های مذکور با عنوان HSA مشخص شده‌اند؛ زمان (ثانیه) و مقدار تابع هدف به دست آمده از حل مسائل نمونه با استفاده از الگوریتم پیشنهادی مبتنی بر بهینه‌سازی گروه ذرات در ستون‌های هفتم و هشتم، ارائه شده است، و ستون‌های مذکور با عنوان HPSO مشخص شده‌اند.

در شکل ۳ شبه کد الگوریتم پیشنهادی مبتنی بر بهینه‌سازی انبوه ذرات ارائه شده است.

Initialize parameters

$C_1, C_2, W_{damp}, n_{pop}, P_c, P_m, crossover, mutation$

Generate initial particles with feasible random number

Calculate all fitness values

Set $n=0$

Do (loop)

Find Pbest

Find Gbest

Update velocity

Find sequence

Calculate fitness values

Apply Crossover and mutation

Update Gbest

Set $n = n+1$

Loop until (n ≤ MaxIt)

Apply local search on Gbest

Update Gbest

شکل ۳- شبه کد الگوریتم مبتنی بر شبیه سازی تبرید

۳- نتایج محاسباتی

به منظور ارزیابی عملکرد الگوریتم های ارائه شده ۲۰ مسأله کوچک و ۲۴ مسأله متوسط و بزرگ با پارامترهای مختلف تولید شده است. پارامترهای مسائل با استفاده از روابطی که جانگواتاناکیت و همکاران (۲۰۰۹) ارائه کرده‌اند، و از مقادیر تولید شده است.

جدول ۳- نحوه تولید پارامترهای مسائل نمونه

پارامترهای مسأله	مسائل کوچک	مسائل متوسط و بزرگ
تعداد کارها	۳؛ ۴؛ ۵	۵؛ ۱۰؛ ۳۰؛ ۵۰
تعداد مراحل	۲؛ ۳؛ ۴؛ ۵	۵؛ ۱۰؛ ۲۰
تعداد ماشین‌های غیرمرتبط در هر مرحله	۲؛ ۳؛ ۴؛ ۵	[۳ ۱]
زمان پردازش استاندارد	[۱, ۵۰]	[۱ ۱۰۰]
سرعت پردازش نسبی برای هر ماشین در هر مرحله برای هر کار	$U[0/7, 1/3]$	$U[0/7, 1/3]$
زمان راه اندازی وابسته به توالی	[۰ ۱۰]	[۰ ۱۰]
موعد تحویل	*	**
* میانگین زمان‌های راه اندازی کار در کل مراحل + مجموع زمان‌های پردازش استاندارد آن کار در تمامی مراحل + میانگین زمان پردازش آن کار بر روی یک ماشین * (J-1) * $U[0,1]$		
** میانگین زمان‌های راه اندازی کار در کل مراحل + مجموع زمان‌های پردازش استاندارد آن کار در تمامی مراحل + میانگین زمان پردازش آن کار بر روی یک ماشین * (J-1) * $U[0,1]$		

برای مقایسه نتایج الگوریتم‌ها از نظر زمان حل و کیفیت جواب، آزمون مقایسه زوجی با کمک نرم‌افزار مینی‌ت‌ب ۱۱۶ استفاده شد.

برای صحت انجام آزمون مذکور لازم است که داده‌های نامتعارف در نظر گرفته نشوند، از نرمال بودن داده‌ها اطمینان حاصل شود، و اندازه نمونه تصادفی باید به حدی باشد که بتوان اختلاف معنی دار بین میانگین مقادیر زمان حل و کیفیت جواب‌ها را تشخیص داد.

همانطور که در جدول ۵ مشاهده می‌شود در مقادیر مربوط به زمان‌های حل و همچنین مقادیر درصد انحراف نسبی (RPD) داده‌های نامتعارف وجود ندارد؛ همچنین، چون تعداد مسائل نمونه بیشتر از ۲۰ است نیازی به انجام آزمون نرمال بودن نیست و داده‌های جمع‌آوری شده برای شناسایی اختلاف معنی دار بین میانگین زمان‌های حل و میانگین کیفیت جواب‌های حاصل از دو الگوریتم، کافی است.

به منظور مقایسه الگوریتم‌ها با یکدیگر، درصد انحراف نسبی ('RPD) محاسبه می‌شود. این معیار در بسیاری از پژوهش‌های پیشین از جمله پژوهش بهنامیان و زندیه (۲۰۱۱)، از این معیار استفاده شده است. هرچه میزان درصد انحراف نسبی حاصل از اجرای الگوریتم کمتر باشد، عملکرد الگوریتم و کیفیت جواب بهتر است.

میزان درصد انحراف نسبی (RPD) از رابطه (۲۵) محاسبه می‌شود:

$$RPD = \frac{sol - \min_{sol}}{\min_{sol}} \quad (25)$$

که در آن sol مقدار جواب در یک اجرا است و \min_{sol} کمترین مقدار جواب، بین همه اجراهای مربوط به یک مسأله است.

جدول ۴ - نتایج حاصل از اجرای الگوریتم ها و نرم افزار GAMS

problem		GAMS		HSA		HPSO	
No.	size	obj	time	obj	time	Obj	time
۱	۳*۵	۰	۰/۱۶	۰	<۰/۴۴	۰	<۰/۱۸
۲	۴*۴	۱۷۹	۵/۲۶	۱۸۱	<۰/۳۴	۱۸۱	<۰/۱۴۵
۳	۴*۳	۹	۲۱۴/۸۲	۱۶	<۰/۳۸	۱۶	<۰/۱۶۶
۴	۳*۵	۹۵	۳/۰۹۹	۹۵	<۲۷/۹	۱۰۰	<۰/۱۲
۵	۴*۲	۰	۰/۲۴۹	۰	<۰/۳۵	۰	<۰/۱۴۷
۶	۳*۴	۸	۰/۷۳۳	۱۰	<۰/۳۶	۱۰	<۰/۱۴۵
۷	۵*۵	۰	۵/۴۱۷	۰	<۰/۷۶	۰	<۰/۳۳
۸	۳*۵	۶	۱۶/۶۶۰	۱۴	<۰/۴۲	۱۴	<۰/۱۷۶
۹	۴*۵	۰	۱۳/۲۲۱	۰	<۵۷/۴	۰	<۰/۲۴
۱۰	۳*۵	۰	۰/۴۵۲	۰	<۴۱/۹	۰	<۰/۱۷
۱۱	۴*۴	۱۹	۱/۸۲۹	۲۰	<۴۷	۲۰	<۰/۲
۱۲	۴*۴	۱۹	۳/۶۷۴	۱۹	<۰/۴۲	۱۹	<۰/۱۸۵
۱۳	۵*۳	۸	۱/۷۵۵	۸	<۰/۴۲	۸	<۰/۱۹
۱۴	۳*۴	۰	۰/۷۴۹	۰	<۰/۳۷	۰	<۰/۱۵
۱۵	۵*۵	۲۵۹	۳۱۹/۶۳۱	۲۶۳	<۵۷/۴۸	۲۶۳	<۰/۲۲
۱۶	۵*۵	*۳۹	۲۰۷/۷۵	۲۴	<۰/۷۲	۲۴	<۰/۳۲
۱۷	۵*۵	*۱۲	۲۰۶۳/۷	۱۲	<۰/۶۴	۱۲	<۰/۲۸

*این مسائل بعد از ۲۰۰۰ ثانیه به جواب بهینه نرسیده و اجرا متوقف شده و مقدار تابع هدف تا آن زمان ارائه شده است.

گروه ذرات از میانگین درصد انحراف نسبی (RPD) الگوریتم مبتنی بر شبیه‌سازی تبرید بزرگتر است است؛ به عبارت دیگر کیفیت جواب الگوریتم مبتنی بر شبیه‌سازی تبرید بهتر از کیفیت جواب الگوریتم مبتنی بر بهینه‌سازی گروه ذرات است.

با توجه به نتایج تحلیل آماری انجام شده؛ با سطح اطمینان ۹۵ درصد، میانگین زمان حل الگوریتم مبتنی بر بهینه‌سازی گروه ذرات از میانگین زمان حل الگوریتم مبتنی بر شبیه‌سازی تبرید کمتر است، و با سطح اطمینان ۹۵ درصد میانگین درصد انحراف نسبی (RPD) الگوریتم مبتنی بر بهینه‌سازی

نتیجه‌گیری

گرفتن زمان‌های آماده‌سازی وابسته به توالی و تابع

هدف مجموع زودکرد و دیرکرد، ارائه شد.

در این مقاله، ابتدا یک مدل ریاضی، برای مسأله

زمانبندی جریان کارگاهی انعطاف‌پذیر با در نظر

جدول ۵- نتایج اجرای الگوریتم‌های پیشنهادی

Problem		HPSO		HAS	
Size	No	RPD	time	RPD	Time
5*5	1	0.003	31.5336	0.003	406.2074
	2	0	30.1062	0	383.9697
5*10	1	0.011	47.0788	0.011	600.5915
	2	0	46.1717	0	704.538
5*20	1	0.003	929565	0.0087	23935
	2	0.095	1089163	0.095	28843
10*5	1	0.013	515376	0	665.6209
	2	0.066	59.8586	0.03	756.4746
10*10	1	0.08	102.7365	0.039	1342.5
	2	0.064	96.0059	0.033	1236.2
10*20	1	0.026	173.0194	0.019	2224.1
	2	0.032	225.1084	0.005	2531.4
30*5	1	0.096	322.5112	0.046	3910.7
	2	0.057	344.8548	0.031	4145.2
30*10	1	0.072	720.8854	0.024	8531.3
	2	0.039	677.4780	0.016	8085
30*20	1	0.071	1213.5	0.027	24432
	2	0.052	1756.4	0.027	16387
50*5	1	0.053	607.1966	0.026	7504.8
	2	0.044	659.0809	0.028	7973.6
50*10	1	0.033	1182.6	0.018	14201
	2	0.051	1061.6	0.039	12782
50*20	1	0.033	331.8146	0.017	23935
	2	0.063	2428.9	0.022	2884.3
Mean		0.044	515.492467	0.023	8266.73

منابع

- Alaykyran, K., Engin, O., & Doyen, A. (2007). Using ant colony optimization to solve hybrid flow shop scheduling problems. *Int J Adv Manuf Techno*, 35, 541-550.
- Behnamian, J., & Zandieh, M. (2011). A discrete colonial competitive algorithm for hybrid flowshop scheduling to minimize earliness and quadratic tardiness penalties. *Expert Systems with Applications*, 38, 14490-14498.
- Botta-Genoulaz, V. (2000). Hybrid flow shop scheduling with precedence constraints and time lags to minimize maximum lateness. *Int. J. Production Economics*, 64, 101-111.
- Crowder, B. (2006). Minimizing the Makespan in a Flexible Flowshop with Sequence Dependent Setup Times, Uniform Machines, and Limited Buffers. West Virginia University.
- Gupta, J. N. D. (1998). Two-stage hybrid flow shop scheduling problem. *Journal of the Operational Research Society*, 39(4), 359-364.
- Janiak, A., Kozan, E., Lichtenstein, M., & Oguz, C. (2007). Metaheuristic approaches to the hybrid flow shop scheduling problem with a cost-related criterion. *International Journal of Production Economics*, 105(2), 407-424.
- Jenabi, M., Fatemi-Ghomi, S. M. T., Torabi, S. A., & Karimi, B. (2007). Two hybrid metaheuristics for the finite horizon elsp in flexible flow lines with unrelated parallel machines. *Applied Mathematics and Computation*, 186(1), 230-245.
- Jungwattanakit, J., Reodecha, M., Chaovallitwongse, P., & Werner, F. (2005). An Evaluation of Sequencing Heuristics for Flexible Flowshop Scheduling Problems with Unrelated Parallel Machines and Dual Criteria. *Otto-von-Guericke-Universitat Magdeburg*, 28(5), 1-23.
- Jungwattanakit, J., Reodecha, M., Chaovallitwongse, P., Werner, F. (2009). A
- این مسأله از نظر پیچیدگی محاسباتی در دسته مسائل نامعین سخت (NP-Hard) قرار می‌گیرد. لذا برای حل مسأله مورد نظر، الگوریتم‌های مبتنی بر بهینه‌سازی گروه ذرات و شبیه‌سازی تبرید ارائه شد. ابتدا چندین مسأله در ابعاد کوچک و بزرگ طراحی و سپس نتایج حاصل از اجرای روش‌های پیشنهادی مقایسه شد.
- نتایج محاسباتی نشان می‌دهد که گرچه الگوریتم مبتنی بر بهینه‌سازی انبوه از الگوریتم مبتنی بر شبیه‌سازی تبرید زمان حل کمتری دارد ولی از نظر کیفیت جواب الگوریتم مبتنی بر شبیه‌سازی تبرید نسبت به الگوریتم مبتنی بر بهینه‌سازی انبوه ذرات بهتر عمل می‌کند. به طور متوسط میزان درصد انحراف نسبی، نتایج آزمایش‌های محاسباتی الگوریتم مبتنی بر بهینه‌سازی ذرات ۴,۴ درصد و الگوریتم مبتنی بر شبیه‌سازی ۲,۳ درصد است.
- در نهایت پیشنهاد می‌شود به منظور بهبود کیفیت جواب حاصل از الگوریتم‌های پیشنهادی و کاهش زمان حل روش‌های بهتری برای تعیین برنامه زمان‌بندی کامل توسعه یابد، و یا از الگوریتم‌های ابتکاری و فراابتکاری کارآمدتری برای حل مسأله استفاده شود. همچنین، برای دستیابی به نتایج واقعی‌تر توصیه می‌شود که محدودیت‌ها و مفروضات دیگر که در مسائل دنیای واقعی مطرح است، مانند در نظر گرفتن محدودیت ظرفیت انبارهای میانی، در نظر گرفته شود.

- on *Robotics and Automation*, 16(4), 408–419.
- Tasgetiren, M. F., Sevkli, M., Liang, Y.-C., & Gencyilmaz, G. (2004). Particle Swarm Optimization Algorithm for Single Machine Total Weighted Tardiness Problem. Proceeding of IEEE Evolutionary Computation Congress.
- Tran, T. H., & Ming ng, K. (2011). A water-flow algorithm for flexible flow shop scheduling with intermediate buffers. *J Sched*, 14, 483-500.
- Urlings, T., and Ruiz, R., & Şerifoğlu, F. S. (2010). Genetic algorithms with different representation schemes for complex hybrid flexible flow line problems. *Int. J. Metaheuristics*, 1(1), 30-54.
- Yaurima, V., Burtseva, L., & Tchernykh, A. (2009). Hybrid flowshop with unrelated machines, sequence-dependent setup time, availability constraints and limited buffers. *Computers and Industrial Engineering*, 56(4), 1452–1463.
- comparison of scheduling algorithms for flexible flow shop problems with unrelated parallel machines, setup times, and dual criteria. *Computers and Operations Research* 36(2), 358–378.
- Kennedy, J., Eberhart, R., (1995). Particle Swarm Optimization. *Proceedings of IEEE International Conference on Neural Networks IV* 1942–1948
- Khalouli, S., Ghedjati, F., Hamzaoui, A., (2010). A meta-heuristic approach to solve a JIT scheduling problem in hybrid flow shop. *Engineering Applications of Artificial Intelligence*, 23, 765-771.
- Liao, C.-J., Tsengb, C.-T., & Luarn, P. (2007). A discrete version of particle swarm optimization for flowshop scheduling problems. *Computers & Operations Research*, 34, 3099-3111.
- Liu, C. Y., & Chang, S. C. (2000). Scheduling flexible flow shops with sequence-dependent setup effects. *IEEE Transactions*

پی‌نوشت

- 1 Identical Machines in Parallel
- 2 Parallel Machines with different speeds
- 3 Unrelated Machines in Prallel
- 4 Water Flow Algorithm (WFA)
- 5 Unrelated Parallel Machines
- 6 Earliest Due Date
- 7 interchange
- 8 Insertion
- 9 Regeneration mutation
- 10 Relative Percentage Deviation
- 11 Minitab16

