

برنامه‌ریزی فرایند و زمان بندی تولید کارگاهی یکپارچه با استفاده از الگوریتم جستجوی ممنوع

سید حسام الدین ذگردی
دانشیار

بخش مهندسی صنایع، دانشکده فنی و مهندسی، دانشگاه تربیت مدرس

نرگس ایمانی پور
دانشجوی دکتری

چکیده

در این مقاله حاضر مسأله برنامه‌ریزی فرایند و زمان بندی یکپارچه در سیستم ساخت و تولید انعطاف پذیر کارگاهی بررسی می‌شود. برخلاف مسائل زمان بندی کلاسیک به دلیل انعطاف پذیری تولید، قطعات ممکن است دارای چندین برنامه فرایند یا چندین مسیر ساخت باشند. دو تابع برنامه‌ریزی فرایند و زمان بندی بسیار به هم مرتبط و بر یکدیگر تأثیر گذارند. بررسی ادبیات نشان می‌دهد که یکپارچه‌سازی برنامه ریزی فرایند و زمان بندی موجب بهبود قابل توجه اهداف زمان بندی و افزایش مطلوبیت مؤثر منابع ساخت و تولید می‌شود [۱]. در این مقاله پس از ارائه مدل ریاضی مسأله تحقیق، یک الگوریتم جستجوی ممنوع برای حل آن پیشنهاد می‌شود. نتایج آزمایش‌های عددی نشان می‌دهد که الگوریتم پیشنهادی نسبت به روش سلسله مراتبی مرسوم برای حل این نوع مسائل عملکرد برتری دارد.

کلمات کلیدی

برنامه‌ریزی فرایند و زمان بندی یکپارچه، تولید کارگاهی انعطاف پذیر، مدل سازی، الگوریتم جستجوی ممنوع

Integration Process Planning and Job Shop Scheduling using a Tabu Search Approach

N. Imanipour
Ph.D Candidate

S.H. Zegordi
Associate Professor

Department of Industrial Engineering,
School of Engineering, Tarbiat Modarres University

Abstract

This paper presents integrated process planning and scheduling in a job shop flexible manufacturing system. Due to production flexibility, each part may have alternate process plans or alternate routs. Two functions of process planning and scheduling are related to each other and optimality of scheduling depends on process planning. Literature review shows that typical scheduling objectives and efficient utilization of manufacturing resources, can be significantly improved as the result of integration of these two important manufacturing system functions [1]. In this paper a heuristic algorithm based on tabu search is presented for solving this problem. The experimental results show that the proposed algorithm outperforms the traditional hierarchical approach.

Keywords

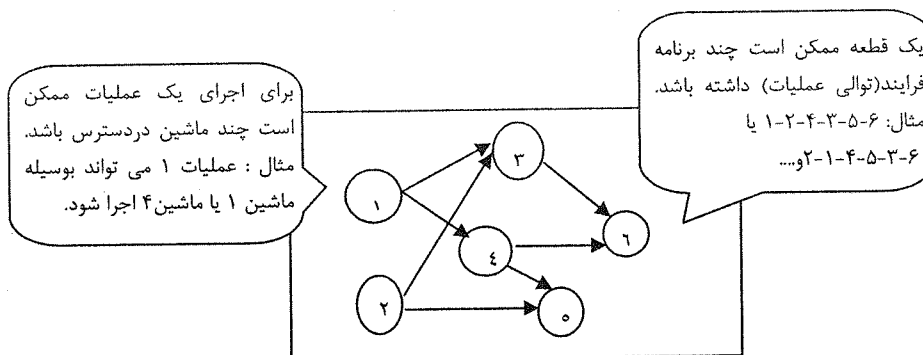
Integration, Process Planning, Scheduling, Modeling, Flexible Job Shop, Tabu Search

برنامه‌ریزی فرایند، فعالیتی است که نیازمندی‌های تکنولوژیک را برای تبدیل یک قطعه از فرم اولیه به فرم نهایی تعیین می‌کند در حالی که فعالیت زمان بندی با تخصیص و زمان بندی عملیات بر منابع بر اساس برنامه فرایند تعیین شده می‌خواهد معیارهای زمان بندی مانند بیشینه زمان تکمیل قطعات (C_{max}) یا متوسط زمان در جریان بودن قطعات (\bar{F}) را بهینه کند. در مقاله حاضر یکپارچه‌سازی این دو تابع مهم در سیستم ساخت و تولید کارگاهی مورد توجه واقع شده است.

در مسأله زمان بندی تولید کارگاهی^۳ (JS) کلاسیک هر قطعه فقط یک برنامه فرایند دارد و هر عملیات از هر قطعه نیز تنها با یک ماشین معلوم قابل اجرا است. امروزه در بسیاری از سیستم‌های ساخت و تولید، انعطاف‌پذیری در عملیات ساخت افزایش یافته است. مسأله تولید کارگاهی انعطاف پذیر^۴ (FJS) حالت توسعه یافته مسأله JS است که در آن برای اجرای هر عملیات ممکن است بیش از یک ماشین در دسترس باشد از این رو با توجه به تنوع تخصیص ماشین‌ها به عملیات هر قطعه ممکن است دارای چند مسیر^۵ باشد. فرض دیگر در JS این است که توالی عملیات هر قطعه معلوم و ثابت است. در مقاله حاضر این فرض نقض می‌شود و توالی عملیات قطعه ثابت نیست، بلکه با توجه به روابط پیش‌نیازی بین عملیات، یک قطعه ممکن است دارای چند توالی عملیات یا به عبارتی چند برنامه فرایند باشد. شکل (۱) این دو نوع انعطاف‌پذیری را برای قطعه‌ای با ۶ عملیات نشان می‌دهد. با توجه به این تعریف مسأله تحقیق حاضر شامل چند زیر مسأله می‌شود: زیرمسأله مسیریابی (تخصیص ماشین به عملیات)، زیر مسأله تعیین برنامه فرایند (تعیین توالی عملیات هر قطعه)، زیر مسأله زمانبندی. منظور از برنامه‌ریزی فرایند تعیین مسیر و برنامه فرایند هر قطعه است.

مسأله FJS اولین بار توسط برندی مارت (۱۹۹۳)^[۲] تعریف شد. در مقاله وی انعطاف‌پذیری تنها در مسیر فرض می‌شود اما برنامه فرایند هر قطعه یکتا و ثابت است. چمبرز و بارمن (۱۹۹۶ و ۱۹۹۸)^[۳ و ۴]، چن و همکاران (۱۹۹۹)^[۵] و کاسم و همکاران (۲۰۰۲)^[۶] هر یک با استفاده از الگوریتم‌های مختلف این مسأله را حل کرده‌اند. در همه این تحقیقات فرض می‌شود برنامه فرایند قطعه ثابت و معلوم است و مسأله تحقیق شامل دو زیرمسأله مسیریابی و زمانبندی است. لی و همکاران (۲۰۰۲)^[۷] در مقاله خود در زنجیره تأمین ساخت و تولید مسأله‌ای مشابه با مسأله تحقیق حاضر را تعریف و با استفاده از الگوریتم ژنتیک حل کرده‌اند. کیم و همکاران (۲۰۰۳)^[۸] نیز با تعریف سه نوع انعطاف‌پذیری در مسأله تولید کارگاهی از الگوریتم تکاملی برای حل این نوع مسأله استفاده کرده‌اند. بیک از اقلو (۲۰۰۲)^[۹] یک مدل بهینه‌سازی تلفیقی از قواعد اولویت بندی و الگوریتم شبیه سازی حرارتی^۶ و متغیرهای زبانی^۷ برای حل مسأله FJS پیشنهاد داده است.

مسأله تحقیق حاضر شامل هر ۳ زیر مسأله مورد اشاره است. هدف تحقیق، یکپارچه سازی دو تابع برنامه‌ریزی فرایند و زمان بندی به منظور کمینه کردن اهداف زمان بندی شامل C_{max} یا \bar{F} است. این دوهدف از جمله اهداف مهم در حوزه مسائل زمان بندی است که کمینه کردن آنها افزایش سرعت فرایند ساخت و تولید، افزایش نرخ خروجی، کاهش سطح موجودی در گردش و افزایش مطلوبیت مؤثر منابع ساخت و تولید را به دنبال خواهد داشت. الگوریتم پیشنهادی بر پایه اصول الگوریتم جستجوی ممنوع^۸ است. از آنجا که این الگوریتم جزو ابتکاری‌های پارامتریک محسوب می‌شود، پس از تنظیم پارامترهای الگوریتم، عملکرد آن نسبت به روش سلسله مراتبی مرسوم برای حل این نوع مسائل ارزیابی می‌شود.



شکل (۱) شبکه ارتباط عملیات یک قطعه فرضی با ۶ عملیات.

۱- مدل سازی ریاضی

مسئله برنامه‌ریزی فرایند و زمان بندی یکپارچه در سیستم ساخت و تولید انعطاف‌پذیر کارگاهی^۴ (IPPSFJS) به این گونه تعریف می‌شود. تعدادی قطعه (کار) برای ساخت به کارگاه شامل مجموعه ماشین‌های مشخص وارد می‌شود. ساخت هر قطعه مستلزم مجموعه‌ای عملیات است که بین آنها روابط پیش‌نیازی مشخص وجود دارد. برای پردازش هر عملیات چند آلترناتیو ماشین ممکن است وجود داشته باشد. طول زمان پردازش هر عملیات روی هر ماشین معلوم است. برای رسیدن به اهداف زمان بندی امکان برون منبع‌یابی^۱ وجود دارد. اگر عملیاتی به منابع بیرونی ارجاع شود زمان حمل به طول زمان انجام عملیات اضافه می‌شود. هدف ارائه مدل و روش حلی است که اهداف زمان بندی را برآورده کند و خروجی آن موارد زیر باشد:

- تعیین بهترین برنامه فرایند هر قطعه

- تعیین بهترین مسیر برای هر قطعه

- بهترین برنامه زمانبندی عملیات

سایر فرضیات عبارتند از:

۱- قطع کردن عملیات مجاز نیست.

۲- هر قطعه در هر لحظه فقط می‌تواند روی یک ماشین پردازش شود.

۳- هر ماشین در هر لحظه فقط می‌تواند یک قطعه را پردازش کند.

۴- زمانهای آماده‌سازی مستقل از توالی و در زمان انجام عملیات لحاظ شده است.

۵- زمان حمل قطعات بین منابع داخلی قابل چشم‌پوشی است.

۶- زمان حمل قطعات از کارگاه به منابع بیرونی معلوم است.

به این ترتیب برای بیان مدل ریاضی لازم است مجموعه‌ها و متغیرهای زیر تعریف شود:

۱-۱- مجموعه‌ها

$i=1, \dots, N$: مجموعه قطعات (کارها)

$k=1, \dots, M$: مجموعه ماشین‌ها (شامل منابع داخلی و بیرونی)

$s_j = 1, \dots, P_j$: مجموعه برنامه‌های فرایند برای قطعه i

$j = 1, \dots, n_i$: تعداد عملیات قطعه i

۲-۱- پارامترها

$P_{t_{jk}}(i, s_j)$: مدت زمان انجام عملیات j ام از ترکیب (i, s_j) روی ماشین k

T_k : مدت زمان حمل قطعه از کارگاه به منبع بیرونی k (یا برعکس)

D_i : موعد تحویل قطعه i

α_i : جریمه هر واحد زودکرد قطعه i

β_i : جریمه هر واحد دیرکرد قطعه i

$$\left. \begin{array}{l} ۱ \text{ ماشین } k \text{ جزء منابع بیرونی است.} \\ ۰ \text{ در غیر این صورت.} \end{array} \right\} = Z_k$$

$$\left. \begin{array}{l} ۱ \text{ ماشین } k \text{ جزو آلترناتیو ماشین عملیات } j \text{ام از ترکیب } (i, s_j) \text{ است.} \\ ۰ \text{ در غیر این صورت.} \end{array} \right\} = a_{jk}(i, s_j)$$

L: یک مقدار ثابت بسیار بزرگ

۳-۱- متغیرها

$$Ft_{jk}^{(i,s_i)} = \text{زمان پایان عملیات زام از ترکیب } (i, s_i) \text{ روی ماشین } k$$

$$\left. \begin{array}{l} ۱ \text{ برنامه فرایند } s_i \text{ برای قطعه } i \text{ استفاده شود.} \\ ۰ \text{ در غیر این صورت.} \end{array} \right\} = Y_{is_i}$$

$$\left. \begin{array}{l} ۱ \text{ عملیات زام از ترکیب } (i, s_i) \text{ روی ماشین } k \text{ انجام شود.} \\ ۰ \text{ در غیر این صورت.} \end{array} \right\} = X_{jk}^{(i,s_i)}$$

$$\left. \begin{array}{l} ۱ \text{ عملیات زام از ترکیب } (i, s_i) \text{ قبل از عملیات } q \text{ از ترکیب } (r, s_r) \text{ روی ماشین } k \text{ انجام شود.} \\ ۰ \text{ در غیر این صورت.} \end{array} \right\} = R_{jqk}^{(i,s_i)(r,s_r)}$$

۴-۱- مدل ریاضی

$$\text{Min } C_{\max} = \text{Max } C_i \quad (۱)$$

or,

$$\text{Min } \bar{F} = \frac{1}{n} \sum_{i=1}^N (C_i - r_i) \quad (۲)$$

S.T.

$$Ft_{(j+1)k}^{(i,s_i)} - Ft_{jm}^{(i,s_i)} + L(1 - a_{(j+1)k}^{(i,s_i)}) X_{(j+1)k}^{(i,s_i)} \geq Pt_{(j+1)k}^{(i,s_i)} + T_k Z_k + T_m Z_m$$

$$i=1, \dots, N; s_i=1, \dots, P_i; j=1, \dots, n_i-1; k=1, \dots, M; m=1, \dots, M \quad (۳)$$

$$Ft_{jk}^{(i,s_i)} - Ft_{qk}^{(r,s_r)} + L \times R_{jqk}^{(i,s_i)(r,s_r)} \geq Pt_{jk}^{(i,s_i)} X_{jk}^{(i,s_i)}$$

$$i \neq r=1, \dots, N; p_i=1, \dots, P_i; p_r=1, \dots, P_r; j=1, \dots, n_i; s=1, \dots, n_r; k=1, \dots, M \quad (۴)$$

$$Ft_{qk}^{(r,s_r)} - Ft_{jk}^{(i,s_i)} + L(1 - R_{jqk}^{(i,s_i)(r,s_r)}) \geq Pt_{qk}^{(r,s_r)} X_{qk}^{(r,s_r)} \quad (۵)$$

$$\sum_{s_i=1}^{P_i} Y_{is_i} = 1 \quad i=1, \dots, N \quad (6)$$

$$\sum_{k=1}^M a_{jk}^{(i, s_i)} X_{jk}^{(i, s_i)} = Y_{is_i} \quad i=1, \dots, N; s_i=1, \dots, P_i; j=1, \dots, n_i \quad (7)$$

$$F_{lk}^{(i, s_i)} \geq (r_i + T_k Z_k + P_{lk}^{(i, s_i)}) X_{lk}^{(i, s_i)} \quad i=1, \dots, N; s_i=1, \dots, P_i; k=1, \dots, M \quad (8)$$

$$F_{jk}^{(i, s_i)} \leq L \times X_{jk}^{(i, s_i)} \quad i=1, \dots, N; s_i=1, \dots, P_i; j=1, \dots, n_i; k=1, \dots, M \quad (9)$$

$$C_i \geq \sum_{k=1}^M \sum_{s_i=1}^{P_i} (F_{nk}^{(i, s_i)} + T_k Z_k) X_{nk}^{(i, s_i)} \quad i=1, \dots, N \quad (10)$$

$$F_{jk}^{(i, s_i)} \geq 0; Y_{is_i} = 0, 1; X_{jk}^{(i, s_i)} = 0, 1 \quad i=1, \dots, N; s_i=1, \dots, P_i; j=1, \dots, n_i; k=1, \dots, M \quad (11)$$

$$R_{jqk}^{(i, s_i)}(r, s_r) = 0, 1 \quad i=1, \dots, N-1; r=i+1, \dots, N; s_i=1, \dots, P_i; s_r=1, \dots, P_r; j=1, \dots, n_i; q=1, \dots, n_r; k=1, \dots, M \quad (12)$$

۱-۵- شرح مدل

روابط ۱ و ۲ اهداف زمان بندی را که به ترتیب کمینه کردن C_{max} یا \bar{F} است نشان می دهد. رابطه ۳ تضمین می کند مجموعه عملیات برنامه فرایند S_i برای قطعه i تداخل زمانی نداشته باشد و توالی آن رعایت شود و اگر انجام عملیاتی به منبع بیرونی واگذار می شود زمان حمل و نقل در نظر گرفته شود. رابطه ۴ و ۵ به طور همزمان تضمین می کند که مجموعه عملیاتی که روی یک ماشین انجام می شود، تداخل زمانی نداشته باشند. طبق رابطه ۶ ساخت هر قطعه فقط از طریق یک برنامه فرایند انجام می شود و بر اساس رابطه ۷ هر عملیات از یک برنامه فرایند منتخب تنها به یک ماشین تخصیص می یابد. رابطه ۸ تضمین می کند زمان پایان اولین عملیات از برنامه فرایند S_i برای قطعه i حداقل بزرگتر از مدت زمان انجام این عملیات به اضافه زمان در دسترس بودن این قطعه است. در صورتی که منبع k یک منبع بیرونی است زمان حمل به طول زمان انجام عملیات اضافه می شود. رابطه ۹ تضمین می کند اگر عملیات Z از ترکیب (i, S_i) روی ماشین k انجام نمی شود متغیر مربوط به زمان تکمیل آن روی ماشین k صفر منظور شود. رابطه ۱۰ زمان تکمیل هر قطعه را محاسبه و روابط ۱۱ و ۱۲ نوع متغیرها را تعیین می کند. همان طور که گفته شد مسأله IPPSFJS حالت توسعه یافته مسأله زمان بندی JS است. مسأله زمان بندی JS یکی از سخت ترین مسائل بهینه سازی ترکیبی است. این مسأله نه تنها دشوار است بلکه در میان این طبقه از مسائل جزو مشکل ترین هاست. به نقل از مرجع [۹] حل بهینه مسأله زمان بندی با ۲۴ قطعه و یک ماشین با استفاده از کامپیوتری با قدرت تولید و ارزیابی یک برنامه زمان بندی در هر میکروثانیه، حدود ۲۰ بلیون سال طول می کشد. به این ترتیب پیچیدگی محاسباتی مسأله IPPSFJS به مراتب بیشتر از مسأله زمان بندی JS است. با بررسی ادبیات موضوع، موارد زیادی از کاربرد روش های ابتکاری به خصوص الگوریتم های فراابتکاری^{۱۱} در حل موفقیت آمیز مسائل زمان بندی دیده می شود [۹]. در این مقاله با توجه به قابلیت های الگوریتم جستجوی ممنوع در حل مسائل پیچیده بهینه سازی ترکیبی، ساختمان الگوریتم پیشنهادی بر پایه اصول الگوریتم جستجوی ممنوع بنا شده است.

۲- ساختمان پایه الگوریتم جستجوی ممنوع (TS)

الگوریتم جستجوی ممنوع جزء روش های جستجوی فراابتکاری است که برای دور شدن از بهینه محلی از اهرم حافظه استفاده می کند. این روش اولین بار توسط گلاور (۱۹۷۷) معرفی شد و از آن زمان تا کنون به طور وسیع برای حل مسائل

- بهینه‌سازی ترکیبی واز جمله مسائل زمانبندی به کار رفته است [۱۰ و ۱۱ و ۱۲]. ساختمان پایه الگوریتم جستجوی ممنوع را می‌توان به صورت زیر تفکیک کرد:
- نمایش فضای جواب و تولید جواب اولیه
 - تعریف حرکت و تولید همسایگی
 - ساختار لیست ممنوع و طول آن
 - تعریف سطح آرمانی
 - معیار توقف

فرض کنید هدف کمینه کردن تابع $f(x)$ است و مجموعه X شامل همه جواب‌های ممکن^{۱۲} مسأله است. همسایگی $N(s)$ برای هر جواب $s \in X$ شامل همه جواب‌های ممکن است که با اعمال تغییر m روی s حاصل می‌شود. به هر تغییر m که s را به یک جواب در همسایگی آن تبدیل می‌کند، حرکت می‌گویند. فرایند حل مسأله با یک جواب ممکن اولیه شروع می‌شود و با تکرارهای متوالی تا برآورده شدن شرط خاتمه ادامه پیدا می‌کند. در هر تکرار به ازای هر جواب s مجموعه همسایگی آن تولید می‌شود. از بین مجموعه همسایگی بهترین همسایه (s^*) انتخاب می‌شود. انتخاب بهترین همسایه در هر همسایگی با این فرض صورت می‌گیرد که این جواب با احتمال بیشتری به سوی جواب بهینه حرکت می‌کند. اگر فرض شود که f^0 بهترین مقدار به دست آمده برای تابع هدف تا این مرحله است آنگاه لزوماً $f(s^*)$ از f^0 بهتر نیست. برای دور شدن از بهینه محلی جواب s^* پذیرفته می‌شود حتی اگر $f(s^*)$ از f^0 یا مقدار تابع هدف به ازای جواب فعلی بیشتر باشد. این استراتژی ممکن است الگوریتم را در حلقه بسته بیندازد. برای این منظور از حافظه کوتاه مدت استفاده می‌شود به این طریق که لیستی از تمامی حرکت‌های پذیرفته شده در طی L تکرار گذشته نگهداری می‌شود. چنانچه جواب s^* عضو این لیست باشد باید رد شود. این کار ممکن است موجب حذف تعدادی از جوابهای خوب شود. از این رو برای هر مسأله یک سطح آرمانی در نظر گرفته می‌شود. اگر جواب s^* این سطح آرمانی را راضی کند پذیرفته و در غیر این صورت رد می‌شود. در پایان هر تکرار با مکانیزم تعریف شده، لیست ممنوع و سطح آرمانی به هنگام می‌شود. و جواب s^* جانشین جواب فعلی می‌شود و به همین ترتیب الگوریتم تارسیدن به معیار توقف تکرار می‌شود.

۳- الگوریتم جستجوی ممنوع پیشنهادی برای مسأله IPPSFJS

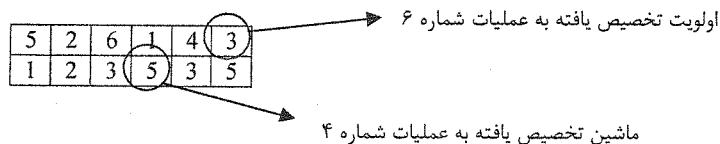
در ساختار الگوریتم پیشنهادی، دو حلقه جستجو پیش بینی شده است. حلقه جستجوی خارجی با استفاده از تابع تولید همسایگی پیش بینی شده بهترین ترتیب عدد اولویت تخصیص یافته به عملیاتها و متناظر با آن بهترین برنامه فرایند هر قطعه را جستجو می‌کند. در مقابل، حلقه جستجوی داخلی سعی می‌کند بهترین برنامه تخصیص ماشین به عملیات را متناظر با هر برنامه فرایند بیابد. در نهایت برنامه زمان بندی عملیات نیز با استفاده از رویکرد زمان بندی پیش‌بینی شده تهیه می‌شود. هر دو حلقه مبتنی بر اصول الگوریتم جستجوی ممنوع است. شکل (۲) pseudo code الگوریتم جستجوی ممنوع پیشنهادی را نشان می‌دهد. با استفاده از مثال شکل (۱) ساختمان الگوریتم پیشنهادی را می‌توان به صورت زیر تشریح کرد:

۳-۱- نمایش فضای جواب و تولید جواب اولیه

هر جواب مسأله به صورت یک آرایه دو بعدی نمایش داده می‌شود. طول آرایه برابر با مجموع تعداد عملیات همه قطعات (TOR) است. هر عنصر از سطر اول عددی بین ۱ و TOR است و عدد اولویت تخصیص یافته به هر عملیات را نشان می‌دهد دو عملیات مختلف نمی‌توانند عدد اولویت یکسان داشته باشد. بر اساس اولویت تخصیص یافته به عملیات‌های هر قطعه، برنامه فرایند آن قطعه با استفاده از الگوریتم Topological Sort [۷] تعیین می‌شود. به علاوه از این اعداد اولویت در ایجاد برنامه زمانبندی عملیات‌ها نیز استفاده می‌شود به این ترتیب که در هر تکرار رویکرد زمانبندی، از بین مجموعه عملیات آماده زمان بندی (مجموعه عملیاتی که پیش نیاز ندارند یا پیش نیازهای آنها انجام شده است) عملیاتی که شاخص اولویت آن بالاتر است برای زمان بندی انتخاب می‌شود. عناصر سطر دوم نشان‌دهنده شماره ماشین تخصیص یافته به هر عملیات است. به این

ترتیب سطر اول آرایه با ترتیبی تصادفی از اعداد ۱ تا TOR و عناصر سطر دوم با اختصاص تصادفی یک ماشین از مجموعه ماشینهای ممکن برای انجام هر عملیات پر می‌شود.

مثال:



Begin:

Randomly generate an initial feasible solution (s).

Calculate the objective functions for s.

$s \rightarrow s^{best}$ and $f(s) \rightarrow f(s^{best})$.

Initialize outer-TL and outer-AL.

Initialize inner-TL and inner-AL.

Outer loop

outer-niter-without-improvement=0

do {

Generate priority numbers neighborhood solutions by outer-neighboring function.

for (all priority numbers neighborhood solutions)

Recall **inner loop**.

Select the best priority numbers neighborhood solution, which is not tabu or satisfies outer-AL.

if (there is an improvement in the objective function) {

outer-niter-without-improvement=0

update the best solution.

}

else outer-niter-without-improvement= outer-niter-without-improvement+1

update outer-TL and outer-AL.

}while (outer-niter-without-improvement < outer-max-iter)

Report the best solution founds, which includes the best known process plan for each part, the best known assignment machine for each operation and the best known schedule)

Inner loop

inner-niter-without-improvement=0

do{

Generate machine assignment neighborhood solutions by inner-neighboring function.

for (all machine assignment neighborhood solutions)

Recall **scheduling**.

Select the best machine assignment neighborhood solution, which is not tabu or satisfies inner-AL.

if (there is an improvement in the objective function) {

inner-niter-without-improvement=0

update the best inner solution.

}

else inner-niter-without-improvement= inner-niter-without-improvement+1

Update inner-TL and inner-AL.

}while (inner-niter-without-improvement < inner-max-iter)

return the best inner solution founds.

scheduling

Initialize:

$S1 = \{O_i / O_i \text{ has no predecessors}\}$

$S2 = \{\text{all operations minus } S1\}$

do {

Select the operation (O^*) with higher priority numbers among members of S1.

Schedule O^* on its assignment machine in possible earliest time.

Delete O^* from S1.

if there is any operation in S2 for which all predecessors are scheduled, then delete it from S2 and add to S1.

}while (S1 is not empty)

Calculate the objective function for the obtained schedule.

return (the objective function)

شکل (۲) Pseudo code الگوریتم جستجوی ممنوع پیشنهادی.

۳-۲- تعریف حرکت و تولید همسایگی

حلقه خارجی: هر همسایگی یک جواب در حلقه خارجی با جا به جایی عدد اولویت تخصیص یافته به دو عملیات مختلف به طور تصادفی، تولید می‌شود. اندازه فضای همسایگی هر جواب برابر با مجموع تعداد عملیات هاست.
مثال: تغییر اولویت عملیات شماره ۳ و ۵



5	2	6	1	4	3
1	2	3	5	3	5

5	2	4	1	6	3
1	2	3	5	3	5

حلقه داخلی: با هر تغییر تصادفی ماشین تخصیص یافته به هر عملیات (در مواردی که امکان انجام عملیات با بیش از یک ماشین وجود دارد) یک جواب همسایه تولید می‌شود. بنابراین اندازه همسایگی هر جواب برابر با مجموع تعداد عملیات است که با بیش از یک ماشین می‌تواند انجام شود.

مثال: تغییر ماشین انجام دهنده عملیات شماره ۴ از ماشین ۵ به ماشین ۱

5	2	6	1	4	3
1	2	3	5	3	5



5	2	6	1	4	3
1	2	3	1	3	5

۳-۳- ساختار فهرست ممنوع و طول آن

حلقه خارجی: مطابق با تعریف حرکت در این حلقه هر گاه تعویض عدد اولویت دو عملیات به عنوان بهترین حرکت پذیرفته شود، شماره این دو عملیات به صورت یک زوج شماره وارد لیست ممنوع می‌شود از این پس تعویض عدد اولویت این دو عملیات تنها به شرطی قابل قبول است که در اثر این حرکت سطح رضایت مندی راضی شود.

حلقه داخلی: هر گاه ماشین تخصیص یافته به یک عملیات تغییر یابد، شماره عملیات و شماره ماشین قبلی به صورت یک زوج عدد وارد لیست ممنوع می‌شود. از این پس تخصیص این عملیات به این ماشین تنها با شرط راضی شدن سطح رضایت مندی پذیرفته است. نتایج آزمایش‌های عددی مقدماتی نشان می‌دهد که بهترین مقدار برای طول لیست ممنوع در دو حلقه برابر با نصف مجموع تعداد عملیات هاست.

۳-۴- تعریف سطح آرمانی

در هر تکرار هرگاه بهترین مقدار به دست آمده برای تابع هدف بهبود یابد، سطح آرمانی به هنگام شده و برابر با این مقدار قرار داده می‌شود.

۳-۵- معیار توقف (Termination Criteria)

از آنجا که مقدار بهینه تابع هدف یا حد پایین مناسب برای آن شناخته شده نیست؛ معیار توقف هر حلقه، تعداد دفعات تکرار متوالی آن حلقه بدون بهبود در تابع هدف فرض می‌شود. این مقدار برای مسائل مختلف متفاوت و به وسعت فضای جستجو بستگی دارد. نتایج آزمایش‌های عددی مقدماتی نشان می‌دهد استفاده از روابط زیر موجب عملکرد بهتر الگوریتم می‌شود.

Termination criteria for outer-loop = TOR

$$\text{Termination criteria for inner-loop} = \text{int} \left[\frac{\text{TOR} \times \text{MNAM}}{3} \right] \quad (13)$$

MNAM: بیشترین تعداد الترناتیو ماشین برای انجام یک عملیات

TOR: مجموع تعداد عملیات همه قطعات

۴- طراحی آزمایش‌های عددی

۴-۱- روش‌های مورد مقایسه

در این مقاله توانایی جستجو و عملکرد الگوریتم پیشنهادی با روش سلسله‌مراتبی مقایسه می‌شود. روش سلسله‌مراتبی رویکردی است که به طور وسیع در حل مسائلی نظیر مسأله تحقیق حاضر- مسائلی که از چند زیرمسأله مرتبط با هم تشکیل شده‌اند. استفاده شده است. در این روش ابتدا زیرمسأله مسیریابی حل می‌شود سپس زیرمسأله برنامه‌ریزی فرایند و زمان بندی تحت محدودیت جواب زیرمسأله مسیریابی حل می‌شود. در این مقاله مشابه مرجع [۸] برای حل مسأله مسیریابی کمینه کردن $\sum_k |W_k - \bar{W}|$ به عنوان تابع هدف استفاده می‌شود. W_k بار کاری ماشین k و \bar{W} متوسط بار کاری ماشینهاست. تخصیص عملیات به ماشین‌ها با هدف متوازن کردن بار کاری ماشین‌ها می‌تواند منجر به عملکرد خوب در مطلوبیت استفاده از منابع شود. از طرفی هر دو تابع کمینه‌سازی C_{max} و کمینه‌سازی \bar{F} با افزایش مطلوبیت استفاده از منابع ارتباط مستقیم دارند. روش سلسله‌مراتبی مورد استفاده در این تحقیق از ساختار الگوریتم جستجوی ممنوع پیروی می‌کند. برای انجام مقایسه منصفانه پارامترهای روش سلسله‌مراتبی به طور مشابه با پارامترهای الگوریتم پیشنهادی تنظیم می‌شوند. لازم به یادآوری است که مقایسه دو روش فوق فقط در حالتی که تمامی منابع داخلی است و زمان حمل به منابع بیرونی وجود ندارد، منطقی است. در حالتیکه منابع بیرونی وجود دارد عملکرد الگوریتم پیشنهادی با الگوریتم ژنتیک [۷] در دو مسأله نمونه برگرفته از مرجع [۷] مقایسه می‌شود.

جدول (۱) مشخصات مسائل نمونه.

Problem	N	TOR	MNAM	Min	Max
1	5	20	3	4	4
2	8	20	3	1	5
3	10	40	3	1	4
4	16	40	3	1	5
5	20	60	6	2	5
6	32	80	3	2	5
7	50	100	6	1	5

۴-۲- طراحی مسائل نمونه

از آنجا که مسائل استاندارد یا مشابهی در ادبیات یافت نشد مسائل نمونه به طور تصادفی تولید می‌شوند. زمان پردازش هر عملیات به طور تصادفی از توزیع یکنواخت $U[1, 10]$ تولید می‌شود. تعداد عملیات هر قطعه یک عدد تصادفی که در بازه $[\text{Min}, \text{Max}]$ قرار دارد. Max و Min به ترتیب حداقل و حداکثر عملیات در قطعه است - تعداد ماشین‌ها در کارگاه ۶ تاست. به این ترتیب مسائل نمونه در ۷ اندازه مختلف مطابق جدول (۱) تولید شده‌اند. ابعاد مسائل با توجه به عرف ادبیات انتخاب شده است.

۴-۳- طراحی روش انجام آزمایش‌ها

از هر اندازه مسأله یک دسته ۵ نمونه‌ای به طور تصادفی تولید و هر نمونه با ۵ جواب اولیه تصادفی متفاوت و به ازای هر جواب ۵ بار اجرا می‌شود. به عبارتی هر مسأله ۲۵ بار حل می‌شود. برنامه هر دو الگوریتم به زبان Borland C کد شده، با یک کامپیوتر شخصی (Pentium 3 (CPU 800 MHz) اجرا می‌شود. در آزمایش‌های مقدماتی برای دو دسته مسأله نمونه ۱ و ۳ پارامترهای طول لیست ممنوع و معیار توقف در ۱۸ وضعیت مختلف تنظیم شدند که بر اساس نتایج بدست آمده مناسبترین مقدار برای طول لیست ممنوع در دو حلقه برابر با نصف مجموع تعداد عملیات و معیار توقف در دو حلقه مطابق رابطه (۱۲) برای هر مسأله تعیین می‌شود. پس از تنظیم مناسب پارامترها آزمایش‌های نهایی روی ۷ دسته مسأله نمونه انجام شد.

۵- نتایج آزمایش‌ها

آزمایش‌ها مطابق روش بیان شده در قسمت ۴-۳ با استفاده از دو الگوریتم مورد مقایسه (قسمت ۴-۱) روی $(7 \times 5) = 35$ مسأله نمونه انجام شد. هر الگوریتم برای هر مسأله $(5 \times 5) = 25$ بار اجرا شده و در هر اجرا بهترین مقدار بدست آمده برای هدف

نگهداری شد. جداول (۲) و (۳) نتایج حاصل از الگوریتم جستجوی ممنوع پیشنهادی و روش سلسله مراتبی (HA) را به ترتیب برای توابع هدف C_{max} و \bar{F} نشان می دهد. ستون اول و دوم از هر جدول به ترتیب شماره دسته مسأله و شماره مسأله نمونه در هر دسته را نشان می دهد. در ستون سوم میانگین مقدار اولیه تابع هدف آمده است. بهترین (Best)، میانگین (Mean) و انحراف معیار (s.d) ۲۵ بار اجرای هر مسأله با دو روش به ترتیب در ستونهای ۴-۶ و ۷-۹ نشان داده شده است. ستون ۱۰ نرخ بهبود جواب حاصل از الگوریتم جستجوی ممنوع پیشنهادی را نسبت به جواب حاصل از روش سلسله مراتبی، و ستون ۱۱ زمان اجرا را برای الگوریتم جستجوی ممنوع پیشنهادی نشان می دهند.

جدول (۲) مقایسه C_{max} برای الگوریتم جستجوی ممنوع پیشنهادی و روش سلسله مراتبی.

Prob.	Sample no	Mean init.	Hierarchical approach			Proposed TS			Improved rate(%)	Cput.(s)
			Best	Mean	s.d.	Best	Mean	s.d.		
1	1	41.4	16	21.0	2.4	14	16.3	1.3	22.1	0.2
	2	33.8	14	19.2	2.4	13	14.6	1.2	23.9	0.1
	3	29.8	17	20.2	1.5	15	16.6	0.8	18.0	0.2
	4	41.6	18	21.5	3.1	18	20.9	1.3	3.0	0.2
	5	31.0	14	16.7	2.5	15	16.4	1.4	2.2	0.1
	average				2.4			1.2	13.8	0.2
2	1	49.6	22	26.2	3.2	22	22.3	0.6	15.0	0.2
	2	53.8	19	24.1	3.3	18	20.5	2.0	15.0	0.2
	3	44.2	19	20.7	1.5	19	20.2	0.9	2.7	0.2
	4	43.2	16	19.8	3.8	16	17.4	1.9	12.3	0.2
	5	46.8	20	26.2	5.0	15	17.4	1.4	33.6	0.2
	average				3.4			1.4	15.7	0.2
3	1	59.2	27	31.5	2.6	26	29.6	2.1	6.1	2.3
	2	59.2	24	30.7	5.2	22	26.4	2.2	14.2	3.0
	3	57.0	24	29.9	3.7	24	27.3	1.5	8.8	2.6
	4	63.6	24	28.8	4.8	24	26.4	1.4	8.6	3.1
	5	57.8	23	26.9	2.9	23	26.5	1.5	1.5	2.9
	average				3.8			1.7	7.8	2.8
4	1	95.2	41	45.8	3.3	42	45.4	3.3	1.0	3.0
	2	100.0	47	55.3	5.6	47	52.4	4.1	5.2	2.6
	3	93.4	42	51.7	13.1	39	45.4	3.5	12.1	3.1
	4	103.4	42	53.7	8.3	41	44.7	2.4	16.7	3.2
	5	83.4	43	50.3	7.4	41	44.9	2.8	10.7	2.9
	average				7.5			3.2	9.1	3.0
5	1	141.0	57	78.0	18.6	49	53.0	2.0	32.1	22.0
	2	137.4	60	75.8	16.4	48	52.4	2.5	30.9	22.7
	3	140.8	60	69.8	11.2	48	52.3	2.3	25.1	21.7
	4	138.4	58	69.7	11.8	50	53.5	2.3	23.2	21.5
	5	125.4	54	61.8	6.5	47	49.9	1.4	19.3	22.8
	average								26.1	22.1
6	1	150.0	78	95.9	16.2	72	78.4	3.0	18.3	64.8
	2	165.4	78	88.1	10.2	73	78.5	3.7	10.9	62.8
	3	143.0	82	88.8	5.8	77	83.4	4.0	6.0	57.4
	4	169.0	79	92.7	11.1	71	78.8	3.3	15.1	66.6
	5	151.0	78	90.2	8.5	72	77.9	3.4	13.7	66.4
	average				10.3			3.5	12.8	63.6
7	1	195.6	95	103.6	12.4	75	79.8	3.5	23.0	247.1
	2	187.4	89	110.0	27.3	75	78.3	3.2	28.8	245.0
	3	180.0	92	102.5	8.6	75	81.2	3.9	20.8	230.6
	4	182.4	96	107.7	14.4	74	76.5	1.5	28.9	279.2
	5	185.8	103	121.2	17.6	82	86.6	3.4	28.5	258.9
	average				16.1			3.1	26.0	252.1

جدول (۳) مقایسه متوسط زمان تکمیل قطعات برای الگوریتم جستجوی ممنوع پیشنهادی و روش سلسله مراتبی.

Prob.	Sample no	Mean init.	Hierarchical approach			Proposed TS			Improved rate(%)	Cput.(s)
			Best	Mean	s.d.	Best	Mean	s.d.		
1	1	30.04	13.60	16.87	1.9	11.80	13.16	0.7	22.0	1.3
	2	23.40	12.80	14.78	1.3	11.20	11.78	0.3	20.3	0.3
	3	21.80	13.60	14.83	0.8	12.00	12.90	0.5	13.0	0.2
	4	28.88	14.60	16.44	1.6	14.40	15.58	0.6	5.3	0.3
	5	23.20	12.80	13.55	0.6	12.20	12.91	0.4	4.7	0.3
	average				1.2			0.5	13.1	0.5
2	1	25.68	12.12	14.02	1.2	11.62	12.16	0.4	13.3	0.3
	2	31.35	13.25	14.58	0.8	12.25	13.14	0.6	9.9	0.3
	3	25.97	10.12	11.56	1.0	9.12	9.57	0.3	17.2	0.3
	4	22.73	10.00	11.32	1.7	7.38	7.83	0.3	30.9	0.4
	5	25.00	11.25	15.65	1.9	10.62	11.24	0.4	28.2	0.3
	average				1.3			0.4	19.9	0.3
3	1	37.06	18.20	20.78	1.4	16.70	18.20	0.8	12.4	3.7
	2	38.44	17.50	21.00	1.7	16.20	18.24	1.1	13.1	4.1
	3	36.78	17.70	19.82	1.2	16.80	18.12	0.8	8.6	3.8
	4	44.52	18.10	20.08	1.3	15.20	17.08	0.7	14.9	4.7
	5	41.72	15.80	17.54	1.2	15.50	17.17	0.9	2.1	4.0
	average				1.4			0.8	10.2	4.0
4	1	46.49	23.38	27.12	2.1	21.75	23.43	1.2	13.6	4.5
	2	51.89	25.69	27.43	1.3	25.81	27.30	1.0	0.5	4.4
	3	51.63	24.69	26.92	1.6	23.06	25.07	1.2	6.9	4.9
	4	54.60	24.12	26.58	2.1	21.19	22.87	1.2	13.9	5.0
	5	46.17	23.12	24.95	1.4	20.25	21.81	1.2	12.6	5.0
	average				1.7			1.2	9.5	4.8
5	1	85.36	31.05	36.95	3.8	26.75	29.08	1.2	21.3	38.3
	2	79.14	36.30	39.80	3.8	27.95	30.83	1.8	22.5	37.8
	3	74.35	32.65	37.13	2.8	25.60	27.41	1.8	26.2	37.0
	4	79.94	31.50	37.13	4.1	25.60	28.60	2.1	23.0	35.9
	5	69.32	33.95	35.61	1.2	24.55	27.40	1.7	23.0	39.8
	average				3.1			1.7	23.2	37.8
6	1	79.74	42.66	45.16	2.6	36.41	40.22	2.1	10.9	109.1
	2	96.36	42.56	46.80	2.4	36.59	40.33	1.7	13.8	111.5
	3	80.06	40.78	46.69	3.2	39.12	41.95	1.6	10.1	110.6
	4	94.36	42.06	46.27	2.2	37.03	39.88	1.5	13.8	116.0
	5	83.76	40.69	44.46	2.0	36.53	39.89	1.9	10.3	118.9
	average				2.5			1.8	11.8	113.2
7	1	96.78	38.58	42.21	2.5	29.30	30.90	1.1	26.8	608.3
	2	86.57	36.78	41.30	5.2	28.40	30.13	1.1	27.1	563.7
	3	91.18	38.12	40.58	5.2	28.34	30.63	1.1	24.5	585.7
	4	92.66	38.16	40.90	1.9	27.50	29.58	1.1	27.7	656.5
	5	88.73	39.42	43.43	2.7	28.96	31.69	1.4	27.0	692.7
	average				3.5			1.2	26.6	621.4

بر اساس نتایج گزارش شده در جداول (۲) و (۳) در تمامی مسائل نمونه نرخ بهبود الگوریتم جستجوی ممنوع پیشنهادی نسبت به روش سلسله مراتبی (ستون ۱۰) عددی مثبت است. بر اساس این معیار الگوریتم جستجوی ممنوع پیشنهادی نسبت به روش سلسله مراتبی عملکرد بهتر دارد. بهتر بودن نتایج الگوریتم جستجوی ممنوع پیشنهادی به این دلیل است که در این الگوریتم در هر تکرار سعی می‌شود بهترین ترکیب برنامه فرایند، برنامه مسیر و برنامه زمانبندی جستجو شود به عبارتی سعی

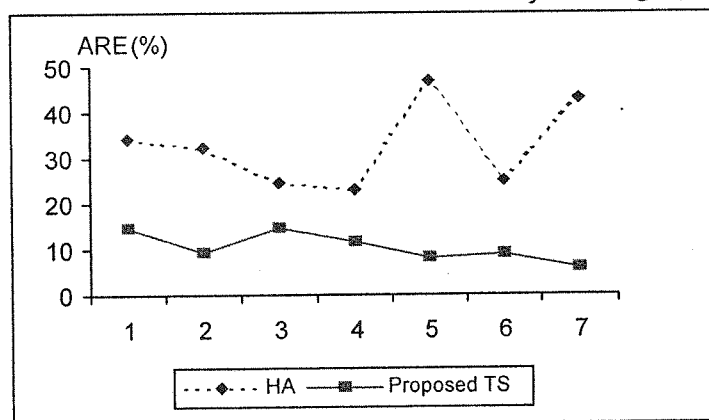
می‌شود ۳ زیر مسأله به طور همزمان بهبود یابند. در حالی که در روش سلسله مراتبی با ثابت کردن مسیر هر قطعه از طریق حل زیر مسأله مسیریابی، بخشی از فضای جواب حذف می‌شود و بنابراین ادامه جستجو در فضای باقیمانده صورت می‌گیرد. البته در الگوریتم جستجوی ممنوع پیشنهادی در برابر این بهبود کیفیت جواب، معیار زمان تاحدودی تخریب می‌شود. از آن جایی که زمان اجرا در بدترین حالت حدود ۶ دقیقه است، این افزایش زمان قابل قبول است.

نمودارهای (۱) و (۲) عملکرد دو الگوریتم را به ترتیب برای دو هدف \bar{F} و C_{max} بر اساس معیار درصد متوسط خطای نسبی ARE^{13} نشان می‌دهد. این معیار در هر مسأله، فاصله جواب از بهترین مقدار شناخته شده تابع هدف را مبنای مقایسه قرار می‌دهد. متوسط خطای نسبی برای هر دسته مسأله با استفاده از رابطه (۱۴) محاسبه می‌شود که در آن p تعداد مسأله نمونه در هر دسته (در اینجا ۵) است.

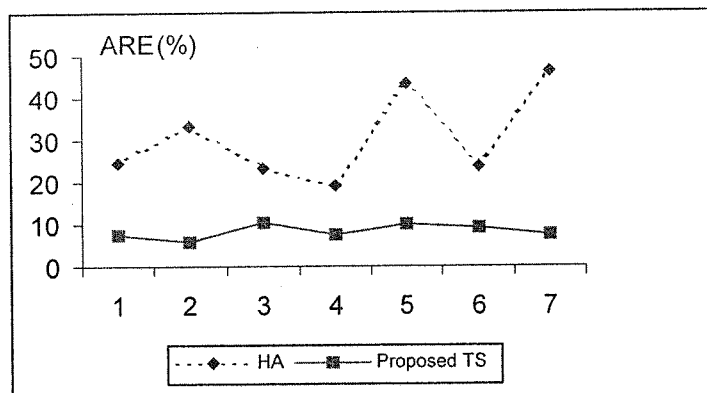
$$ARE = \frac{1}{p} \sum_{i=1}^p \frac{Mean_i - Best_i}{Best_i} \times 100 \quad (14)$$

با بررسی این دو نمودار مشاهده می‌شود که:

- ۱- الگوریتم جستجوی ممنوع پیشنهادی متوسط خطای نسبی کمتری نسبت به روش سلسله مراتبی دارد.
- ۲- رفتار الگوریتم جستجوی ممنوع با بزرگ شدن ابعاد مسأله یک رفتار پایدار است.
- ۳- در مسائل نمونه ۵ و ۷ که دارای انعطاف‌پذیری بالاتری نسبت به مسائل دیگر هستند متوسط خطای نسبی روش سلسله مراتبی به شدت افزایش پیدا کرده است که نشان می‌دهد با افزایش انعطاف‌پذیری و در نتیجه بزرگ شدن فضای جواب این روش به هیچ وجه قابل اعتماد نیست. از سوی دیگر مقایسه متوسط مقدار اولیه تابع هدف و متوسط بهترین مقدار بدست آمده (در هر مسأله) تأیید کننده این موضوع است که یکپارچه کردن تابع برنامه‌ریزی فرایند و تابع زمانبندی موجب بهبود معنی‌دار در اهداف زمان بندی می‌شود که البته الگوریتم جستجوی پیشنهادی نیز با طراحی مناسب، به خوبی از شانس انعطاف‌پذیری در راستای بهبود اهداف زمانبندی استفاده کرده است.



نمودار (۱) مقایسه متوسط خطای نسبی بین الگوریتم TS پیشنهادی و روش سلسله مراتبی (تابع هدف C_{max}).



نمودار (۲) مقایسه متوسط خطای نسبی بین الگوریتم TS پیشنهادی و روش سلسله مراتبی (تابع هدف \bar{F}).

جدول (۴) مقایسه عملکرد الگوریتم جستجوی ممنوع پیشنهادی را با عملکرد الگوریتم ژنتیک ارائه شده توسط لی و همکاران [۷] بر روی دو مسأله نمونه از مرجع [۷] نشان می دهد (تنها مسائل در دسترس برای مقایسه). در این دو مسأله هدف زمان بندی کمینه کردن C_{max} است. در مسأله اول الگوریتم پیشنهادی توانسته جواب بهینه را بیابد و در مسأله دوم با زمان اجرایی به مراتب کمتر از الگوریتم ژنتیک مقدار هدف را حدود ۲۰ درصد بهبود دهد.

جدول (۴) مقایسه نتایج الگوریتم جستجوی ممنوع پیشنهادی با نتایج مرجع [۷].

Problem	Parts	Operations	Machines	GA		Proposed TS	
				C_{max}	Cpu.t(s)	C_{max}	Cpu.t(s)
1	4	7	5	13	4.5	13	0.01
2	8	20	5	34	13.5	27	0.12

۶- نتیجه گیری

در این مقاله، یکپارچگی برنامه ریزی فرایند و زمان بندی و تأثیر آن بر بهبود اهداف زمانبندی در سیستم ساخت و تولید انعطاف پذیر کارگاهی مورد بررسی واقع شده است. پس از مدل سازی مسأله یک الگوریتم جستجوی ممنوع دو حلقه ای برای حل این مسأله پیشنهاد شد. نتایج حاصل از آزمایش های عددی برای دو هدف کمینه کردن C_{max} و \bar{F} ، نشان دهنده عملکرد برتر روش پیشنهادی نسبت به روش سلسله مراتبی مرسوم برای حل این مسأله است. همچنین نتایج نشان می دهد که یکپارچه شدن دو تابع برنامه ریزی فرایند و زمان بندی موجب بهبود معنی دار اهداف زمان بندی می شود. علاوه بر موضوع انعطاف پذیری در محیط های واقعی تولید کارگاهی، موضوعات کاربردی دیگری مانند سطوح فرعی مونتاژ، محدودیتهای مربوط به قید وبستها و ابزار و زمان های آماده سازی وجود دارد که توجه به آنها باعث غنی شدن مدل می شود. همچنین با ظهور فلسفه مدیریت زنجیره تأمین لازم است که واحد ساخت و تولید به عنوان یک عضو زنجیره، اهداف سایر اجزا از جمله مشتری را نیز ببیند که این موضوع می تواند بر تعریف اهداف تأثیر بگذارد.

زیر نویس ها

- | | |
|-----------------------------|--|
| 1- Makespan | 8- Tabu Search (TS) |
| 2- Mean flow time | 9- Integrated Process Planning and Scheduling in FJS |
| 3- Job Shop Scheduling (JS) | 10- Outsourcing |
| 4- Flexible Job Shop (FJS) | 11- Meta heuristic |
| 5- Rout | 12- Feasible |
| 6- Simulated Annealing | 13- Average Relative Error |
| 7- Linguistic | |

مراجع

- [1] Weltan and B. Khoshnevis, Integration of process planning and scheduling- a review, Journal of Intelligent Manufacturing, Vol. 11, 51-63, (2000).
- [2] Brandimarte, P., Routing and scheduling in a flexible job shop by tabu search, Annals of Operations Research, Vol. 41, 157-183, (1993).
- [3] Chambers, J.B., and Barnes, J.W., Flexible job shop scheduling by tabu search, Graduate program in operations research and industrial engineering, The university of Texas at Austin, Technical report series, ORP96-09. Available from <<http://www.cs.utexas.edu/users/jbc/>>, (1996).
- [4] Chambers, J.B., and Barnes J.W., Reactive Search for flexible job shop scheduling, Graduate program in operations research and industrial engineering, The university of Texas at Austin, Technical report series, ORP98-**. Available from <<http://www.cs.utexas.edu/users/jbc/>>, (1998).
- [5] Chen, H., Hlow, J. and Lehman C., A genetic algorithm for flexible job-shop scheduling, 0-7803-5180-0-5/99 1999 IEEE, 1120-1125, (1999).
- [6] Kacem, I., Hammadi, S. and Borne, P., Pareto-optimality approach for flexible job-shop scheduling problems: hybridization of evolutionary algorithms and fuzzy logic, Mathematics and Computers in Simulation, Vol. 60, No. 3-5, 245-276, (2002).
- [7] Lee, Y.H., Jeong, C.S. and Moon, C., Advanced planning and scheduling with outsourcing in manufacturing supply chain, Computers & Industrial Engineering, Vol. 43, 351-374, (2002).
- [8] Kim, Y.K., Park, K. and Ko, J., A Symbiotic evolutionary algorithm for the integration of process planning and job shop scheduling, Computers & Operations Research, Vol. 30, 1151-1171, (2003).
- [9] Baykasoglu, A., Linguistic-based meta heuristic optimization model for flexible job shop scheduling, International Journal of Production Research, Vol. 40, 4523-4543, (2002).
- [10] Glover, F., Tabu search part I. ORSA Journal on Computing, Vol. 1, No.13, 190-206, (1989).
- [11] Glover, F., Tabu search part II. ORSA Journal on Computing, Vol. 2, No.1, 4-32, (1990).
- [12] Glover, F., Tabu search: A Tutorial. Interfaces, Vol. 20, No. 4, 74-94, (1990)