مدل‌سازی و حل یک مسأله زمان‌بندی دو معیاره در یک سیستم تولید سلولی با در نظر گرفتن زمان‌های آماده‌سازی وابسته به توالی سلول‌ها

رضا توکلی مقدم[1 *]; آزاده نصری[2]

چکیده

در این مقاله، یک مدل برنامه‌ریزی ریاضی عدد صحیح مختلط دو معیاره برای زمان بندی سلول‌ها و قطعات در هر سلول به‌طور همزمان در یک سیستم تولید سلولی[1] (CMS) ارائه می‌شود. هدف این مدل، کمینه کردن زمان کل تکمیل ساخت قطعات و حرکت بین سلولی قطعات با در نظر گرفتن زمان‌های راه‌اندازی وابسته به توالی خانواده قطعه اختصاص یافته به سلول‌ها می‌باشد. در طراحی و برنامه‌ریزی سیستم تولید سلولی، سه مرحله اصلی می‌بایست مد نظر قرار گیرد که شامل تشکیل سلول، استقرار سلول‌ها و ماشین‌ها، و زمان بندی آنها است. از آنجا که این مدل پیشنهادی و مسأله زمانبندی CMS جزء مسایل چندجمله‌ای نامعین سخت[2] است، از الگوریتم ژنتیک[3] (GA) به‌عنوان یکی از روش‌های فراابتکاری کارامد برای حل مدل استفاده می‌شود. در پایان برای نشان دادن کارایی این الگوریتم پیشنهادی، چند مسأله نمونه حل شده، نتایج محاسباتی مربوطه با نتایج به‌دست آمده به کمک یک ابزار بهینه‌ساز مورد مقایسه قرار می‌گیرد.

کلمات کلیدی : زمان بندی، سیستم تولید سلولی، زمان‌های آماده‌سازی وابسته به توالی، الگوریتم ژنتیک.

Modeling and Solving of a Bi-Criteria Scheduling Problem for a Cellular Manufacturing System with Sequence-Dependent Cell Setup Times

R. Tavakkoli-Moghaddam; A. Nasri

ABSTRACT

This paper presents a new, bi-criteria mixed-integer programming model for scheduling cells (part families) and parts within each cell in a manufacturing cellular system. The objective of this model is to minimize the makespan and inter-cell movements simultaneously, while considering sequence-dependent cell setup times. In the CMS design and planning, three main steps must be considered, namely cell formation (i.e., part families and machine grouping), inter and intra-cell layouts, and scheduling issue. Due to the NP-hardness of the proposed model and the scheduling problem in the CMS, a genetic algorithm (GA) as an efficient meta-heuristic method is proposed to solve such a hard problem. Finally, a number of test problems are solved to show the efficiency of the proposed GA and the related computational results are compared with the results obtained by the use of an optimization tool.

KEYWORDS : Scheduling, Cellular Manufacturing System, Sequence-Dependent Setup Times, Genetic Algorithm.

تاریخ دریافت مقاله:۱۳۸۶/۱۲/۱۱

تاریخ اصلاحات مقاله: ۱۳۸۸/۵/۱۰

[1] * نویسنده مسئول و استاد گروه مهندسی صنایع، پردیس دانشکده‌های فنی، دانشگاه تهران: E-mail: tavakoli@ut.ac.ir

[2] دانش‌آموخته کارشناسی ارشد گروه مهندسی صنایع، پردیس دانشکده‌های فنی، دانشگاه تهران: E-mail: azadehnasri@yahoo.com

۱- مقدمه

یکی از تصمیم‌گیری‌های اساسی در تولید سلولی، زمان-بندی قطعات در هر سلول است. در زمان بندی CM مجموعه ای از قطعات باید در سلول‌ها به وسیله مجموعه ای از ماشین‌ها پردازش شوند. هدف یافتن توالی انجام قطعات در هر گروه و توالی انجام گروه‌های قطعات در سلول‌ها بر روی مجموعه ماشین‌هاست؛ به گونه‌ای که معیار مورد نظر در زمانبندی بهینه شود. معیارهای مختلفی می‌توانند برای یک مسأله زمانبندی تولید سلولی در نظر گرفته شوند که می‌توان به "حداقل نمودن زمان تکمیل"، "حداقل کل زمان تکمیل وزن‌دار"، "حداقل کردن حداکثر دیرکرد"، حداقل کردن کل تاخیر"، "حداقل کردن کل تأخیر وزن‌دار"، "حداقل کردن تعداد کارهای به تاخیر افتاده" و "حداقل کردن تعداد جابجایی‌های بین سلولی" اشاره کرد. همچنین در عمل محدودیت های عملیاتی مختلفی نیز در مسأله زمانبندی CM وجود دارند. برای نمونه زمان‌های راه‌اندازی می‌توانند به توالی انجام سلول‌ها وابسته باشند. در توضیح زمان‌های راه‌اندازی وابسته به توالی انجام سلول‌ها این نکته مورد توجه است که در سیستم تولید سلولی با توجه به توالی خانواده قطعات متعلق به هر سلول هزینه راه‌اندازی ماشین‌ها متفاوت است، برای مثال در صنایع شیمیایی و یا خطوط رنگ کاری اگر توالی پردازش هر سلول تغییر کند با توجه به رنگ مورد نیاز، زمان آماده‌سازی ماشین‌ها برای پردازش خانواده قطعه بعدی متفاوت خواهد بود. در این شرایط در صورت تغییر در توالی سلول‌ها زمان‌های راه‌اندازی متفاوت بوده، مدت زمان تکمیل کل کارها تغییر می‌کند.

بیشتر الگوریتم‌هایی که برای مسأله زمان‌بندی گروهی توسعه داده شده‌اند دارای دو مرحله‌اند: مرحله‌ی اول توالی قطعات را در گروه‌ها مشخص می‌کند و مرحله دوم توالی انجام گروه‌های قطعات در سلول‌ها را تعیین می‌نماید. از نظر تعداد سلول‌ها می‌توان مقالات را به دو دسته تقسیم نمود: مقالاتی که یک سلول و آنهایی که بیش از یک سلول را در نظر گرفته اند. سلیمانپور [۱] چندین سلول را در نظر گرفت و روشی ابتکاری برای کمینه کردن زمان تکمیل ارائه نمود. از نظر تعداد ماشین‌ها نیز مقالات را می توان به دو دسته تقسیم نمود: مقالاتی که دو ماشین در سلول تولیدی نظر گرفتند، یوشیدا و هیتومی [۲] الگوریتمی برای حل بهینه مسأله زمانبندی جریان کارگاهی دو ماشینه ارائه داده‌اند که در آن زمان راه‌اندازی در نظر گرفته شده است. سگیکوچی [۳] و بیکر [٤] کار یوشیدا و هیتومی را گسترش داده، مسأله جریان کارگاهی دو ماشینه را با زمان راه‌اندازی زمانبندی کردند. یانگ و چرن [٥] مسأله زمانبندی جریان کارگاهی را در نظر گرفتند که در آن هر گروه نیاز به زمان راه‌اندازی و زمان برداشت از ماشین مشابه دارند. هیتومی و حام [٦] حد پایینی برای بهینه کردن مدت زمان کل ساخت تعریف کرده، روش شاخه و حد را برای دستیابی به توالی بهینه قطعات و گروه‌ها پیشنهاد می‌کند. با توجه به اینکه زمانبندی سیستم‌های تولید سلولی یک مسأله چندجمله‌ای نامعین سخت است [۷]، تحقیق‌های مختلفی انجام شده‌اند که در آنها تلاش بر ایجاد الگوریتم‌های ابتکاری برای حل مسأله زمانبندی گروهی است.

لگندران و نودتاسمبون (LN) [۸] الگوریتمی پیشنهاد کردند که برای حل مسأله زمانبندی داخل سلولی به روش LN ارجاع داده شد و بسیار مشابه الگوریتم پیشنهادی NEH (نواز، ان-اسکور و هم) [۹] بود. تنها تفاوت بین الگوریتم NEH و روش LN این است که در الگوریتم NEH کارهای اولیه بر حسب مدت کل پردازش به‌صورت نزولی مرتب می‌شود اما در روش LN میانگین زمان پردازش به‌صورت نزولی در نظر گرفته می‌شود. در مسایل زمانبندی تک سلولی با بیش از دو ماشین مقالات حاضر را می‌توان از نظر محیط تولیدی به دو گروه تولید کارگاهی، جریان کارگاهی و غیره تقسیم نمود. ومرلو و واکهاریا [۱۰] پیاده‌سازی زمانبندی هشت خانواده محصول در یک محیط تولید کارگاهی را با هم سنجیداند و بیان کردند که زمانبندی بر مبنای تشکیل خانواده در ارتباط با حداقل زمان جریان و تاخیر برتری دارد. محمودی و دولی [۱۱] یک سلول جریان کارگاهی با زمان‌های راه‌اندازی وابسته به توالی را در نظر گرفتند. برای یک مسأله ماشین موازی با زمان‌های راه‌اندازی وابسته به توالی، ائوم [۱۲] باهدف حداقل کردن کل دیرکرد وزنی، یک روش ابتکاری جستجوی ممنوع را توسعه داد. لگندران [۱۳] بر روی پیاده‌سازی روش‌های مختلف حاصل از ترکیب سه روش PT، LN و CDS (کمبل، دوداک و اسمیت) [١٤] بررسی‌هایی را انجام داد و در پایان بیان نمود که روش LN-PT که در آن در مرحله اول از روش LN و در مرحله بعدی از روش PT استفاده می‌شود، نسبت به روش‌های ترکیبی PT-LN، PT-CDS و CDS-PT برتری دارد. الگوریتم‌های PT و CDS به‌ترتیب الگوریتم‌های ابتکاری یگانه و چندگانه هستند که مسأله زمانبندی جریان کارگاهی با n کار و m ماشین را به مسأله‌ای با n کار و دو ماشین ساده می‌کنند و سپس از الگوریتم جانسون برای تعیین توالی کارها استفاده می‌کنند.

یانگ و لیااو [۱٥] مسأله زمانبندی سلولی را با دو سلول و حرکت بین سلولی در نظر گرفتند و آن را با استفاده از روش

شاخه و حد و یک الگوریتم ابتکاری حل نمودند. اسکالر [۱۶] حد پایین جدیدی که پایین تر از روش ارائه شده توسط هیتومی و هام [۲] که در آن از مقایسه جزئی توالی‌ها در رویه شاخه و حد استفاده می‌شود برای مسأله زمانبندی گروهی جریان کارگاهی ارائه داد.

اسکالر [۱۷]، فرانکا [۱۸] و هندیزاده [۱۹] مسأله زمانبندی سلولی را با یک سلول، چندین ماشین و زمان آماده‌سازی وابسته به توالی با هدف حداقل کردن زمان کل ساخت و چشم پوشی از حرکات بین سلولی در نظر گرفتند، اسکالر مسأله مورد نظر را با الگوریتم‌های ابتکاری بهبود یافته حل کرد. فرانکا مسأله مورد نظر را با الگوریتم‌های فراابتکاری ژنتیک و ممتیک حل نمود، و هندیزاده مسأله مورد نظر را با الگوریتم فرا ابتکاری جستجوی ممنوع حل نمود. توکلی مقدم و همکاران [۲۰] مسأله زمانبندی تولید سلولی را با هدف حداقل کردن زمان تکمیل قطعات و جابجایی‌های بین سلولی، با الگوریتم ممتیک حل نمودند.

تفاوت مدل ارئه شده در این مقاله با سایر تحقیقات انجام شده، در نظر گرفتن زمان‌های راه‌اندازی وابسته به توالی سلولی با دو هدف کمینه کردن زمان کل تکمیل قطعات و جابجایی‌های بین سلول‌ها برای تعیین توالی قطعات در هر سلول و توالی انجام گروه‌های قطعات در سلول‌ها به‌طور همزمان و ارائه یک مدل برنامه‌ریزی عدد صحیح مختلط چند معیاره[۴] است.

۲- مدل پیشنهادی و روش حل آن

طی فرایند زمانبندی سیستم تولید سلولی با توجه به مدل پیشنهادی مقاله، خروجی های مورد انتظار عبارتند از:

- تشکیل خانواده قطعات با توجه به استقرار ماشین‌ها در هر سلول،
- تعیین توالی قطعات برای تولید در هر سلول،
- تعیین توالی انجام گروه‌های قطعات در سلول‌ها.

۲-۱- مدل ریاضی

مدل پیشنهادی با تشریح مفروضات، پارامترها، متغیرها، تابع هدف و محدودیت‌ها در ادامه ارائه می‌گردد:

مفروضات مدل

فرضیات در نظر گرفته شده برای این مدل عبارتند از:

- زمان عملیات تمامی قطعات روی هر نوع ماشین قطعی است.
- تعداد قطعات مشخص و ثابت می‌باشد.
- تعداد ماشین‌های موجود در سیستم مشخص و ثابت است.
- تعداد سلول‌های موجود در سیستم مشخص و ثابت است.
- ماشین‌های موجود در هر سلول مشخص است.
- هر نوع ماشین می‌تواند تنها یک نوع عملیات را انجام دهد و هر عملیات نیز می‌تواند توسط یک نوع ماشین انجام شود.
- مسافت بین سلول‌ها یکسان است؛ یعنی زمان جابجایی بین سلولی برای تمامی حرکات ثابت است.
- زمان خرابی برای ماشین‌ها نخواهیم داشت.
- راندمان ماشین‌ها ۱۰۰٪ است.
- زمان‌های راه‌اندازی برای هر یک از ماشین‌ها مشخص و وابسته به توالی سلول‌ها است.

اهداف مدل

همان‌طور که می‌دانیم در یک رویکرد کلی باید چندین معیار را در زمانبندی مدنظر قرار داد؛ یعنی باید تمامی معیارهای مرتبط با زمانبندی تولید سلولی را به گونه‌ای در تابع هدف با هم ترکیب نمود. اما به دلیل پیچیدگی مسأله و مشکلات ناشی از زمان محاسباتی مورد نیاز آن؛ در نظر گرفتن همه آنها امکان پذیر نیست. بنابراین در این مقاله، تنها زمان کل ساخت (تکمیل قطعات یا کارها) و حرکت بین سلولی در نظر گرفته خواهد شد. هدف مدل، به حداقل رساندن مجموع کل زمان ساخت و زمان حرکت بین سلولی است.

- مدت زمان کل ساخت: زمانی که لازم است تا کل قطعات موجود در سیستم زمانبندی شده و تولید گردند.
- مدت زمان حرکت بین سلولی: زمان حمل و نقل قطعات بین سلول‌هاست. در مواقعی قطعات نمی‌توانند به‌طور کامل داخل یک سلول تولید گردند؛ زیرا که تمامی ماشین‌های مورد نیاز برای تولید قطعات در همان سلولی که قطعه قرار دارد وجود ندارد. باید توجه داشت که این حرکات بین سلولی باعث کاهش اثر بخشی سیستم‌های تولید سلولی از طریق افزایش حجم حمل و نقل قطعات و زمان جریان و همچنین پیچیدگی کنترل تولید می‌شوند.

زیر نویس‌ها

- i : نماد قطعات که متعلق به مجموعه $i = \{1,\ldots,P\}$ است و P تعداد قطعات است.
- j : نماد ماشین که متعلق به مجموعه $j = \{1,\ldots,M\}$ است و M تعداد ماشین‌ها است.

- c : نماد سلول که متعلق به مجموعه $c = \{1,\dots,f\}$ است و f تعداد سلول‌ها است.
- k : نماد توالی قطعات که متعلق به مجموعه $k = \{1,\dots,K\}$ می‌باشد و K تعداد توالی‌ها است.
- b : نماد توالی سلول که متعلق به مجموعه $b = \{1,\dots, KC\}$ می‌باشد و Kc تعداد توالی‌ها است.

نمادهای ورودی

مقادیر پارامترهای ورودی که باید در ابتدای حل مدل ریاضی مشخص شوند، عبارتند از: مسیر ساخت قطعات، زمان عملیات، استقرار ماشین در سلول، زمان حرکت بین سلول و زمان راه‌اندازی وابسته به توالی:

t_{ij} : زمان لازم برای پردازش قطعه i روی ماشین j ؛

$$a_{ij} = \begin{cases} ۱ & \text{اگر قطعه } i \text{ به ماشین } j \text{ نیاز داشته باشد؛} \\ ۰ & \text{در غیر این‌صورت.} \end{cases}$$

$$m_{jc} = \begin{cases} ۱ & \text{اگر ماشین } j \text{ به سلول } c \text{ تخصیص یابد؛} \\ ۰ & \text{در غیر این‌صورت.} \end{cases}$$

s_{ncj} : زمان راه‌اندازی روی ماشین j در سلول c وقتی بلافاصله پس از سلول n قرار دارد.

s_{ccj} : زمان راه‌اندازی روی ماشین j در سلول c .

متغیرهای تصمیم گیری

$$x_{ic} = \begin{cases} ۱ & \text{اگر قطعه } i \text{ به سلول } c \text{ تخصیص یابد؛} \\ ۰ & \text{در غیر این‌صورت.} \end{cases}$$

$$y_{cb} = \begin{cases} ۱ & \text{اگر سلول } c \text{ به توالی } b \text{ اختصاص یابد؛} \\ ۰ & \text{در غیر این‌صورت.} \end{cases}$$

$$z_{ikc} = \begin{cases} ۱ & \text{اگر قطعه } i \text{ به توالی } k \text{ سلول } c \text{ تخصیص یابد؛} \\ ۰ & \text{در غیر این‌صورت.} \end{cases}$$

C_{kjcb} = زمان تکمیل قطعه واقع در توالی k روی ماشین j در سلول c که در توالی b واقع است؛

C_{max} = مدت زمان کل ساخت.

مدل ریاضی

تابع هدف عبارت است از کمینه‌سازی زمان حرکت بین سلولی و مدت زمان کل ساخت. در واقع مدل یاد شده، مساله را هم از لحاظ زمانبندی و هم از لحاظ عملیاتی مدنظر قرار می‌دهد.

$$\min \sum_{i=1}^{P}\sum_{c=1}^{C}\left(\sum_{j=1}^{M} a_{ij} \times \left|a_{ij} - m_{jc}\right|\right) \times x_{ic} \times T_c + C_{\max} \qquad (۱)$$

رابطه (۲) هر یک از قطعات را به یک سلول تخصیص می‌دهد.

$$\sum_{c\in C} x_{ic} = 1 \qquad ;\forall i \in P \qquad (۲)$$

رابطه (۳) تضمین می‌کند که هر توالی فقط یک سلول اختصاص یابد.

$$\sum_{c\in C} y_{cb} = 1 \qquad ;\forall b \in Kc \qquad (۳)$$

رابطه (٤) تضمین می‌کند که هر سلول تنها به یک توالی اختصاص یابد.

$$\sum_{b\in K_c} y_{cb} = 1 \qquad ;\forall c \in C \qquad (۴)$$

رابطه (٥) هر یک از قطعات تخصیص داده شده به هر سلول را به یک توالی در آن سلول اختصاص می‌دهد.

$$\sum_{k\in K} z_{ikc} = x_{ic} \qquad ;\forall i \in P \;\; ,\forall c \in C \qquad (۵)$$

رابطه (٦) تضمین می کند که هر توالی در یک سلول حداکثر به یک قطعه اختصاص یابد.

$$\sum_{i\in P} z_{ikc} \le 1 \qquad ;\forall k \in K \;\; ,\forall c \in C \qquad (۶)$$

رابطه (٧) زمان تکمیل قطعه اختصاص داده شده به اولین توالی بر روی اولین ماشین در سلولی که در توالی اول قرار گرفته را برابر با مجموع زمان پردازش قطعه مشخص شده و زمان راه‌اندازی سلول مشخص شده بر روی ماشین اول قرار می‌دهد.

$$C_{11C1} = \max(\forall i \in p : (t_{i1} + s_{cc1} \times a_{i1}) \times z_{i1c} \times y_{c1}) \;;\forall \; c \in C \qquad (۷)$$

رابطه (٨) زمان تکمیل قطعه اختصاص داده شده به توالی kام روی ماشین اول در سلولی که در توالی اول قرار گرفته را در صورتی‌که پیش از قطعه مشخص شده، قطعه‌ای در توالی ما قبل سلول مشخص شده قرار گرفته باشد (برای ماشین اول راه‌اندازی در سلول مورد نظر صورت گرفته است) برابر با مجموع زمان تکمیل در توالی قبلی و زمان پردازش در توالی فعلی قرار می‌دهد؛ در غیر این‌صورت برابر با مجموع زمان پردازش توالی فعلی و زمان راه‌اندازی ماشین اول در سلول مشخص شده قرار می‌دهد.

$$C_{k1c1} = \begin{cases} C_{k-1,1,c,1} + \max(\forall i \in p : \\ \quad t_{i1} \times z_{ikc} \times y_{c1}) \quad \text{if } C_{k-1,1,c,1} > 0 \\ \max(\forall i \in P : (t_{i1} + s_{cc1} \times a_{i1}) \\ \quad \times z_{ikc} \times y_{c1}) \quad \text{if } C_{k-1,1,c,1} = 0 \end{cases} \;;\forall\,(k \ge 2) \in K \;,\; c \in C \qquad (۸)$$

رابطه (٩) زمان تکمیل قطعه اختصاص داده شده به توالی اول سلول‌هایی که در سایر توالی‌ها غیر از توالی اول قرار دارند روی ماشین اول را برابر با مجموع زمان تکمیل در توالی آخر سلول قبلی و زمان راه‌اندازی ماشین اول در سلول توالی فعلی و زمان پردازش قطعه توالی اول سلول فعلی قرار می‌دهد.

$$C_{1,1,C,b}=\begin{cases}\max(\forall n_1\in C,n_1\neq c\&i\in P:C_{K,1,n_1,b-1}\\ +(s_{n_1c1}\times a_{i1}\times y_{n_1,b-1}+t_{i1})\times z_{i1c}\times y_{cb})\\ \quad\text{if } C_{K,1,n_1,b-1}>C_{K,1,n_2,b-2}>0\\ \max(\forall n_1\in C,n_1\neq c\&i\in P:C_{K,1,n_1,b-1}\\ +(s_{n_2c1}\times a_{i1}\times y_{n_2,b-1}+t_{i1})\times z_{i1c}\times y_{cb})\\ \quad\text{if } C_{K,1,n_1,b-1}=C_{K,1,n_2,b-2}>C_{K,1,n_3,b-3}>0\\ \vdots\\ \max(\forall n_1\in C,n_1\neq c\&i\in P:C_{K,1,n_1,b-1}\\ +(s_{n_fc1}\times a_{i1}\times y_{n_f,b-f}+t_{i1})\times z_{i1c}\times y_{cb})\\ \quad\text{if } C_{K,1,n_1,b-1}=C_{K,1,n_2,b-2}=\\ \quad C_{K,1,n_3,b-3}=\cdots=C_{K,1,n_f,b-f}>0\\ \max(i\in P:\ (s_{cc1}\times a_{i1}+t_{i1})\times z_{i1c}\times y_{cb})\\ \quad\text{if } C_{K,1,n_1,b-1}=0\end{cases}$$
$$;\forall\,(b\geq 2)\in Kc\ \ \&\ \ c\in C$$

(۹)

رابطه (۱۰) زمان تکمیل قطعه اختصاص داده شده به توالی k روی ماشین اول را در سلولی که در سایر توالی‌ها غیر از توالی اول قرار دارند برابر با مجموع زمان تکمیل توالی قبلی و زمان پردازش قطعه مشخص شده در توالی فعلی روی ماشین اول و در صورتی‌که هیچ قطعه‌ای در توالی‌های ماقبل توالی فعلی در سلول مشخص شده قرار نداشته باشند، زمان آماده‌سازی ماشین اول در سلول مورد نظر قرار می‌دهد.

(۱۰)
$$C_{k1cb}=\begin{cases}C_{k-1,1,c,b}+\max(\forall n\in C,n\neq c\,\&\,i,o\in P\\ \quad,i\neq 0\,\&\,d\in K,d<k:(s_{nc1}\times a_{i1}\times y_{n,b-1}\\ \quad\times(1-a_{o1}\times z_{odc})+t_{i1})\times z_{ikc}\times y_{cb})\\ \quad\quad\text{if } C_{k\text{-}1,1,c,b}>0\\ \max(i\in P:\ (s_{cc1}\times a_{i1}+t_{i1})\times z_{ikc}\\ \quad\times y_{cb})\quad\quad\text{if } C_{k-1,1,c,b}=0\end{cases}$$
$$;\forall\,(b\geq 2)\in Kc,\ c\in C\ ,\ (k\geq 2)\in K$$

رابطه (۱۱) زمان تکمیل قطعه اختصاص داده شده به توالی اول روی ماشین j (غیر از ماشین اول) را در سلولی که به توالی اول اختصاص دارد برابر با مجموع زمان تکمیل این قطعه روی ماشین (j-1) در همان توالی و زمان راه اندازی ماشین j در سلول مشخص شده و زمان پردازش قطعه توالی k روی ماشین j قرار می‌دهد.

$$C_{1jc1}=C_{1,j-1,c,1}+\max(\forall i\in P:(t_{ij}+s_{ccj}\times a_{ij})\times z_{i1c}\times y_{c1})\quad ;\forall(j\geq 2)\in M\ \&\ \forall c\in C \tag{۱۱}$$

رابطه (۱۲) زمان تکمیل قطعه اختصاص داده شده به توالی اول روی ماشین j (غیر از ماشین اول) را در سلولی که به توالی غیر از توالی اول اختصاص داده شده است برابر مجموع بیشترین زمان تکمیل این قطعه روی ماشین قبلی (j-1) در سلول فعلی و زمان تکمیل توالی آخر سلول قبلی (b-1) روی ماشین j با زمان راه‌اندازی ماشین j در سلول فعلی با زمان پردازش قطعه توالی اول در سلول فعلی (b) روی ماشین j قرار می‌دهد.

$$\begin{aligned}C_{1jcb}&=\max(\max(\forall n\in C:C_{K,j,n,b-1}),\\ &C_{1,j-1,c,b})+\max(\forall n\in C,n\neq c\,\&\,\mathrm{i}\in\mathrm{P}:\\ &(\mathrm{t_{ij}}+\mathrm{s}_{ncj}\times a_{ij}\times y_{n,b-1})\times z_{i1c}\times y_{cb})\\ &;\forall(\mathrm{j}\geq 2)\in\mathrm{M}\ ,\ \ \mathrm{c}\in\mathrm{C}\ ,\ (\mathrm{b}\geq 2)\in\mathrm{Kc}\end{aligned}\tag{۱۲}$$

رابطه (۱۳) زمان تکمیل قطعه اختصاص داده شده به توالی k (غیر از توالی اول) روی ماشین j (غیر از ماشین اول) را در سلولی که در توالی اول قرار گرفته در صورتی‌که پیش از قطعه مشخص شده؛ قطعه‌ای در توالی‌های ماقبل (d<k) سلول مشخص شده قرار گرفته باشد (برای ماشین j راه اندازی در سلول مورد نظر صورت گرفته است) برابر با مجموع بیشترین زمان تکمیل توالی قبلی (k-1) روی ماشین j و زمان تکمیل توالی فعلی (k) روی ماشین قبلی (j-1) با زمان پردازش در توالی فعلی روی ماشین j قرار می دهد و در غیر این‌صورت برابر با مجموع بیشترین زمان تکمیل توالی قبلی (k-1) روی ماشین j و زمان تکمیل توالی فعلی (k) روی ماشین قبلی (j-1) با زمان راه‌اندازی ماشین j در سلول توالی اول با زمان پردازش در توالی فعلی روی ماشین j قرار می‌دهد.

(۱۳)
$$C_{kjc1}=\begin{cases}\max(C_{k-1,j,c,1},C_{k,j-1,c,1})+\max(\forall i\in p:t_{ij}\times z_{ikc}\times y_{c1})\\ \quad\quad\text{if } C_{k\text{-}1,j,c,1}>0\\ \max(C_{k-1,j,c,1},C_{k,j-1,c,1})+\max(\forall i\in P:\\ \quad(t_{ij}+s_{ccj}\times a_{ij})\times z_{ikc}\times y_{c1})\quad\text{if } C_{k\text{-}1,j,c,1}=0\end{cases}$$
$$;\forall\,(k\geq 2)\in K\ ,\ \ (j\geq 2)\in M\ ,\ c\in C$$

رابطه (۱٤) زمان تکمیل قطعه اختصاص داده شده به توالی k (غیر از توالی اول) روی ماشین j (غیر از ماشین اول) را در سلول c که به توالی b (غیر از توالی اول) اختصاص دارد برابر با مجموع بیشترین زمان تکمیل توالی قبلی (k-1) روی ماشین j و زمان تکمیل توالی فعلی (k) روی ماشین قبلی(j-1) با زمان پردازش قطعه مشخص شده در توالی فعلی روی ماشین j و در صورتی‌که هیچ قطعه‌ای در توالی‌های ماقبل توالی فعلی (d<k) در سلول توالی فعلی قرار نداشته باشند، زمان راه‌اندازی ماشین j را در سلول مورد نظر قرار می‌دهد.

$$C_{kjcb} = \max(C_{k-1,j,c,b}, C_{k,j-1,c,b}) + \max(\forall n \in C, n \neq c \;\&\; i,o \in P, i \neq 0 \;\&\; d \in K \;,\; d \leq k\text{:}\; (s_{ncj} \times a_{ij} \times y_{n,b-1} \times (1 - a_{oj} \times z_{odc}) + t_{ij}) \times z_{ikc} \times y_{cb}) \quad ;\forall\, (j \geq 2) \in M \;,\; c \in C \;,\; (k \geq 2) \in K \;,\; (b \geq 2) \in K_c \tag{۱۴}$$

رابطه (۱۵) C_{max} برابر با بیشترین زمان تکمیل قرار می‌گیرد.

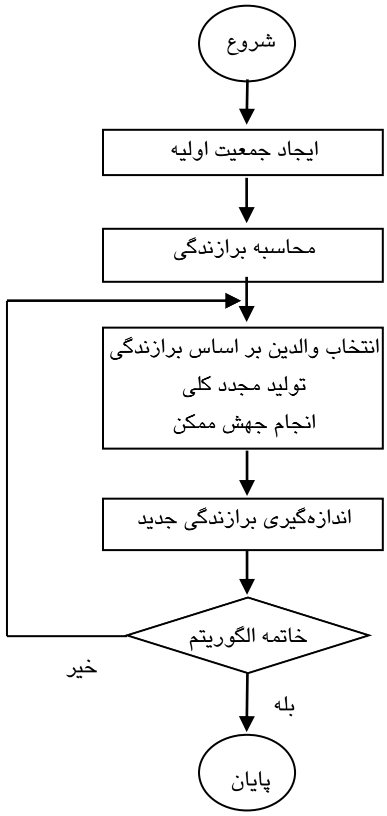
$$C_{\max} = \max(C_{k,j,c,b}) \quad ;\forall j \in M \;,\; k \in K ,\; c \in C \;\&\; b \in Kc \tag{۱۵}$$

در رابطه (۱۶) متغیرهای ۰ و ۱ معرفی می‌گردند،

$$x_{ic}, y_{cb}, z_{ikc} : \text{binary} \qquad ;\forall i \in P , k \in K, c \in C \;,\; b \in Kc \tag{۱۶}$$

۲-۲- الگوریتم ژنتیک پیشنهادی

الگوریتم ژنتیک یک رویکرد جستجوی تصادفی تطبیقی بر اساس نظریه "بقای انسب" داروین است. مناسب ترین عضو، بیشترین احتمال بقا را و در نتیجه ازدیاد تعداد را دارد؛ در حالی‌که نامناسب‌ترین‌ها می‌میرند. یکی از مهم‌ترین ویژگی‌های GA گرایش آن به سمت جواب‌های بهینه قطعی یا نزدیک به بهینه حتی در فضاهای جستجوی بزرگ یا پیچیده است [۲۱]. شکل (۱) روش کلی الگوریتم ژنتیک را نشان می‌دهد.



شکل ۱ : نمای کلی الگوریتم ژنتیک

به‌کارگیری الگوریتم ژنتیک نیاز دارد به :

- یک ساز و کار کدگذاری برای نمایش جواب به نام کروموزوم،
- توابع ارزیابی،
- یک رویکرد انتخاب برای تولید دوباره،
- یک عملگر تقاطعی[۵]،
- یک عملگر جهش[۶].

در ادامه به تشریح جزییات الگوریتم می‌پردازیم.

نمایش جواب

در این مسأله ماتریس A با ابعاد (c×p) طراحی شده که به عنوان یک حل برای تعیین توالی قطعات در سلول‌ها است. بدین ترتیب که در p عنصر این ماتریس اعداد ۱ تا p اختصاص می‌یابد و چگونگی طراحی این ماتریس و حل‌ها به گونه‌ای است که حل‌ها همگی به‌صورت شدنی ایجاد می‌گردند و مسأله در محیط نشدنی سیر نمی‌کند. همان‌طور که قبل از این بیان شد، برای کمینه کردن مدت زمان کل ساخت با دو نوع توالی روبه رو هستیم: تعیین توالی قطعات در هر سلول و تعیین توالی سلول‌ها، که به‌طور همزمان تعیین می‌گردند. پس متناظر با هر توالی قطعات درون سلول‌ها یک توالی برای سلول‌ها نیز خواهیم داشت. پس متناظر با ماتریس A که توالی قطعات در سلول‌ها را مشخص می‌کند، ماتریس B با ابعاد (۱×c) در نظر گرفته شده است که در هر سطر آن توالی سلول‌ها آورده می‌شود. پس در پایان هر ماتریس A و یک ماتریس B یک حل برای مدل ارایه شده در قسمت ۲ محسوب می‌گردد. اینک نمونه‌ای از کروموزوم برای ۱۰ قطعه و ۳ سلول آورده شده است.

۱	.	.	۴	.	.	۹	.	.	.
.	.	۳	.	.	۱۰	.	.	.	۶
.	.	.	۲	.	.	۷	۵	.	۸

ماتریس A

۲	۳	۱

ماتریس B

در مثال یاد شده، دو ماتریس A و B با هم یک کروموزوم به شمار می روند که نشان می‌دهد قطعات (۱،۴،۹) به ترتیب به سلول اول، قطعات (۳،۱۰،۶) به ترتیب به سلول دوم و قطعات (۲،۷،۵،۸) به ترتیب به سلول سوم اختصاص یافته‌اند و همچنین توالی سلول‌ها به‌صورت ۲، ۳ و ۱ است: یعنی ابتدا قطعات موجود در سلول دوم با توالی تعیین شده پردازش می‌شوند؛ سپس قطعات موجود در سلول سوم و در پایان پردازش قطعات مربوط به سلول اول انجام می‌گیرد.

جمعیت اولیه

اولین مرحله ای بعد از تعیین روشی برای تبدیل هر جواب به یک کروموزوم ایجاد یک جمعیت اولیه از کروموزوم هاست.

در این مرحله جواب اولیه به‌صورت تصادفی تولید می‌شود. البته در بعضی موارد با توجه به نوع مساله و برای بالا بردن سرعت همگرایی الگوریتم از روش‌های ابتکاری نیز استفاده گردیده است. در این مقاله جمعیت اولیه به‌صورت تصادفی ایجاد شده و ساختار ماتریسی دارد، یعنی به اندازه جمعیت کروموزوم به‌صورت تصادفی ایجاد می‌گردد. در این تحقیق اندازه جمعیت از رابطه (۱۸) محاسبه می‌گردد:

$$pop-size=\begin{cases}2\times(m\times p\times c) & \text{if } (m\times p\times c)\le 40\\ 80 & \text{if } (m\times p\times c)>40\end{cases} \qquad (۱۸)$$

تابع برازش

اگر g مقدار تابع هدفی باشد که از رابطه (۱) به دست می‌آید، با انجام تبدیل (۱۹) مقدار تابع برازش برای هر یک از کروموزوم‌های تولید شده در مرحله قبل به دست می‌آوریم:

$$f_i=\begin{cases}C_{max}-g_i & \text{if } g_i<C_{max}\\ 0 & \text{otherwise}\end{cases} \quad ; \quad i=1,\dots,n \qquad (۱۹)$$

که در آن n = pop-size است. برای انتخاب C_{max} راه‌های مختلفی وجود دارد؛ بزرگترین مقدار g_i که تا این لحظه دیده شده، بزرگ‌ترین مقدار در جمعیت فعلی و یا بزرگ‌ترین مقدار در k نسل قبلی [۲۲]. در اینجا مقدار C_{max} برابر با بزرگ‌ترین مقدار g که در جمعیت فعلی وجود دارد قرار داده می‌شود.

مکانیزم نمونه گیری

ساز و کار نمونه‌گیری به چگونگی انتخاب کروموزوم‌ها از فضای نمونه گیری مربوط می‌شود که در این مقاله از روش نمونه‌برداری تصادفی با جایگذاری[۷] استفاده شد. در این روش احتمال انتخاب متناظر با هر کروموزوم، بر اساس مقدار برازندگی آن محاسبه می‌شود؛ به‌طوریکه اگر f_k مقدار برازندگی کروموزوم k باشد احتمال بقای متناظر با آن کروموزوم عبارت است از:

$$p_k=f_k\Big/\sum_{i=1}^{n}f_i \qquad (n=\text{pop-size}) \qquad (۲۰)$$

روش SSR روش انتخاب اصلی چرخ رولت[۸] است. در این روش به این صورت عمل می شود که برای انتخاب هر کروموزوم، ابتدا یک عدد تصادفی بین صفر و یک تولید شده و سپس عدد گفته شده در هر بازه ای که قرار گرفت، کروموزوم متناظر با آن انتخاب می شود. کروموزوم انتخاب شده دوباره به گردونه شانس بازگردانده می‌شود و این چرخه تا انتخاب اندازه جمعیت مورد نیاز ادامه می‌یابد.

عملگر تقاطع

احتمال یا نرخ تقاطعی (P_c) عبارت است از احتمال رخ دادن عملگر تقاطع در هر یک از کروموزوم‌ها. برای هر یک از کروموزوم‌ها یک عدد تصادفی بین صفر و یک تولید کرده، اگر این عدد کوچک‌تر از P_c باشد کروموزوم مورد نظر برای تقاطع انتخاب می‌گردد که در این صورت به‌صورتی که در ادامه تشریح می‌گردد فرزند جدیدی تولید می‌کند؛ در غیر این صورت کروموزوم مورد نظر برای عمل تقاطع انتخاب نمی‌گردد.

عملگر تقاطع عملگرهایی هستند که یک یا چند نقطه از دو یا چند جواب را انتخاب و مقادیر آنها را تعویض می‌کنند. این عملگرها یک جواب را در نظر گرفته، محل‌هایی از جواب را با جواب‌های دیگر جایگزین می‌کنند و جواب‌های جدید بوجود می‌آورند. در این مقاله، برای تولید فرزندان از عملگر یک نقطه‌ای استفاده شده که به تصادف نقطه‌ای برای برش مشخص می‌گردد و از این نقطه عمل برش صورت گرفته، سمت راست کروموزوم‌های والد با هم تعویض می شوند؛ به این ترتیب دو فرزند جدید ایجاد می‌شود. ممکن است فرزندان ایجاد شده حلی غیر موجه باشند که در این صورت باید دو فرزند ایجاد شده به عنوان دو حل از مسأله موجه گردند. با یک مثال چگونگی عملکرد عملگر تقاطعی طراحی شده نشان داده می‌شود، که ماتریس A با ۵ قطعه و ۲ سلول را در نظر می‌گیرد. دو کروموزوم P_1 و P_2 به عنوان والد هستند که بعد از عمل برش دو فرزند C_1 و C_2 تولید می‌گردند:

نقطه برش

$$P_1\ \begin{array}{ccc|cc}0&2&5&0&0\\3&0&1&0&4\end{array} \rightarrow C_1\ \begin{array}{ccccc}0&2&5&2&1\\3&0&1&0&4\end{array}$$

$$P_2\ \begin{array}{ccc|cc}5&0&0&2&1\\0&0&3&0&4\end{array} \rightarrow C_2\ \begin{array}{ccccc}5&0&0&0&0\\0&0&3&0&4\end{array}$$

همانطور که مشخص است، کروموزوم‌های تولید شده غیر موجه هستند؛ کروموزوم C_1 دارای قطعات مضاعف است (قطعات ۱ و ۲) ودر کروموزوم C_2 بعضی قطعات نادیده گرفته شده‌اند (قطعات ۱ و ۲). برای موجه کردن این کروموزوم ها از استراتژی اصلاحی استفاده می‌شود که در بخش‌های بعدی به تشریح آن پرداخته می‌شود.

عملگر جهش

عملگر جهش برای جلوگیری از جستجو از واگرایی در یک بهینه محلی مورد استفاده قرار می گیرد. احتمال یا نرخ جهش (P_m) در کروموزوم برابر با احتمال تغییر کردن هر کدام از ژن‌ها است که مقدار کوچکی است. فرایند جهش استفاده شده در این مقاله بدین صورت است که ابتدا برای هر کروموزوم عددی بصورت تصادفی بین صفر و یک تولید می شود؛ اگر این عدد کوچکتر از P_m باشد، عمل جهش به‌صورتی که در ادامه

تشریح می‌گردد، انجام می‌شود؛ در غیر این‌صورت بدون تغییر باقی می‌ماند. در کروموزومی که برای جهش انتخاب شده است، دو عدد تصادفی بین یک و تعداد کل ژن‌ها انتخاب می‌گردد و مقادیر ژن‌های متناظر با این اعداد با یکدیگر تعویض می‌گردند. ممکن است فرزندان ایجاد شده حلی غیر موجه باشند که در این صورت باید دو فرزند ایجاد شده به عنوان دو حل از مسأله موجه گردند.

کروموزوم C_1 در مثال قبل را در نظر گیرید؛ فرض کنید مکان‌های ۳ و ٤ به عنوان محل‌های جهش انتخاب می‌شوند؛ در نتیجه بعد از انجام جهش تبدیل به C_1' می‌شود.

$$C_1 \begin{matrix} 0 & 2 & \overset{\downarrow}{5} & \overset{\downarrow}{2} & 1 \\ 3 & 0 & 1 & 0 & 4 \end{matrix} \rightarrow C_1' \begin{matrix} 0 & 2 & 2 & 5 & 1 \\ 3 & 0 & 0 & 1 & 4 \end{matrix}$$

استراتژی برخورد با محدودیت‌ها

بحث دیگری که در اجرای الگوریتم ژنتیک وجود دارد چگونگی برخورد با محدودیت‌های مسأله است؛ زیرا عملگرهای ژنتیک مورد استفاده در الگوریتم باعث تولید کروموزوم های غیر موجه می‌شود. در این مقاله از استراتژی اصلاحی استفاده شده که در این روش به جای اینکه کروموزوم غیر موجه حذف گردد، تبدیل به یک کروموزوم موجه می‌شود. از این عمل بعد از انجام عمل ترکیب که امکان ایجاد کروموزوم‌های غیر موجه وجود دارد، استفاده می‌شود.

متناظر با هر یک از کروموزوم‌های غیر موجه دو ماتریس انتقال تعریف می‌کنیم که تمام درایه‌های آن در شروع صفر است. ژن‌های سمت چپ کروموزوم غیر موجه را با ژن‌های سمت راست نقطه برش مقایسه می‌کنیم؛ به ازای هر تساوی که مشاهده می‌کنیم، در کروموزوم مربوطه ژن مربوطه را صفر و متناظر با آن در ماتریس انتقال مقدار آن را قرار می‌دهیم. این کار را برای تمام ژن‌های سمت راست نقطه برش انجام می‌دهیم. در پایان برای تولید فرزند موجه والد اول از ماتریس انتقال والد دوم و بالعکس استفاده می‌کنیم؛ به این ترتیب که به ازای ژن‌های سمت راست نقطه برش از مقادیر ماتریس انتقال مربوطه استفاده می‌کنیم. به این ترتیب سعی می‌شود که فرزندان خصوصیات والدین خود را ارث ببرند، به جای آنکه این خصوصیات را با حذف کروموزوم‌های غیر موجه از بین ببریم. برای روشن شدن مطلب مثال قبلی را در نظر بگیرید؛ به ازای دو کروموزوم C_1 و C_2 دو ماتریس I_1 و I_2 را داریم. ادامه الگوریتم به شرح زیر است:

$$C_1 \begin{matrix} 0 & \underline{0} & 5 & 2 & 1 \\ 3 & 0 & \underline{0} & 0 & 4 \end{matrix} \qquad I_1 \begin{matrix} 0 & \underline{2} & 0 & 0 & 0 \\ 0 & 0 & \underline{1} & 0 & 0 \end{matrix}$$

$$C_2 \begin{matrix} 5 & 0 & 0 & 0 & 0 \\ 0 & 0 & 3 & 0 & 4 \end{matrix} \qquad I_2 \begin{matrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{matrix}$$

و در پایـان، بـا جایگـذاری مـاتریس I_1 در کرومـوزوم C_2 و ماتریس I_2 در کرومـوزوم C_1 دو فرزنـد موجـه C_1' و C_2' تولیـد می‌گردند.

$$C_1' \begin{matrix} 0 & 0 & 5 & 2 & 1 \\ 3 & 0 & 0 & 0 & 4 \end{matrix} \qquad C_2' \begin{matrix} 5 & 0 & 2 & 0 & 0 \\ 1 & 0 & 3 & 0 & 4 \end{matrix}$$

معیار توقف

حرکت الگوریتم ژنتیک از یک نسل به نسل دیگر، یعنی فرایند انتخاب کروموزوم‌های اصلح و تولید مجدد تا جایی ادامه می‌یابد که معیار توقف ارضاء می‌گردد. چندین معیار توقف از جمله تعداد ثابت تولید نسل، زمان محاسباتی و همگرایی توابع برازش را می‌توان انتخاب کرد. در این مقاله از تعداد ثابت تولید نسل به عنوان معیار خاتمه الگوریتم پیشنهادی استفاده شده است.

۲-۳- تحلیل نتایج محاسباتی

با ارائه نمونه مسایل انتخابی، جواب‌های الگوریتم ژنتیک با یک حد نزدیک به بهینه که توسط نرم افزار لینگو ۸ ، حاصل شده، مقایسه می‌گردد. ماهیت الگوریتم‌های حل مسأله رسیدن به بهترین جواب در یک زمان قابل قبول است. از این دو پارامتر زمان محاسباتی و مقدار بهینه جواب به عنوان عوامل اصلی قیاس و نتیجه‌گیری مطرح خواهند بود. همچنین با توجه به اینکه زمان کل عملیات در این مدل مورد توجه است، زمان نیز مورد اندازه‌گیری و قیاس قرار می‌گیرد.

طراحی آزمایش

نحوه انتخاب ابعاد مسایل انتخابی با توجه به تاثیر سه پارامتر قطعه، ماشین و سلول پیچیده خواهد بود. تعداد سلول‌ها بین ۲ و ۳، تعداد قطعات به‌طور تصادفی بین ۳ و ۱۰ و تعداد ماشین‌ها بین ۲ و ۸ تغییر می‌کنند. برای مقایسه‌های هدف‌مندتر، ۱۲ مساله که به نوبت تغییرات یک پارامتر را نشان می‌دهد تعیین شده است که در جدول (۱) نشان داده شده است.

در هر یک از ۱۲ مسأله انتخاب شده ورودی های مسأله به این‌صورت تعیین می‌گردند: ماتریس مسیر ساخت قطعات؛ یعنی نیاز قطعات به ماشین‌ها، به‌طور تصادفی تعیین می‌گردند؛ زمان پردازش قطعات بر روی ماشین‌هایی که در ماتریس مسیر قطعه درایه متناظر با آن عدد ۱ قرار دارد اعداد صحیحی هستند که به‌طور تصادفی از توزیع یکنواخت U[1,10] انتخاب می گردند، ماتریس تعیین شرایط تعلق ماشین‌ها به سلول‌ها نیز به‌طور

تصادفی از اعداد ۰ و ۱ جایگذاری می‌گردند و زمان‌های راه‌اندازی وابسته به توالی سلول‌ها بدون در نظر گرفتن شرایط

جدول (۱): ابعاد مسایل انتخابی برای حل

ردیف	تعداد قطعه	تعداد ماشین	تعداد سلول	ابعاد($M\times P\times C$)
۱	۳	۲	۲	۱۲
۲	٤	۲	۲	١٦
۳	۳	٤	۲	۲٤
٤	٤	٤	۲	۳۲
٥	٤	٥	۲	٤۰
٦	٤	۷	۲	٥٦
۷	٥	۷	۲	۷۰
۸	٥	۸	۲	۸۰
۹	٦	۸	۲	۹٦
۱۰	۷	۸	۳	١٦۸
۱۱	۸	۸	۳	۱۹۲
۱۲	۱۰	۸	۳	۲٤۰
۱۳	۲۰	۱۰	٤	800
١٤	۳۰	۱۲	٥	1800
١٥	٤۰	١٥	٦	3600

تعلق ماشین‌ها در سلول‌ها، اعداد صحیحی هستند که از توزیع یکنواخت U[1,20] انتخاب می‌شوند. این زمان راه‌اندازی در دسته کلاس زمان راه‌اندازی کوچک[۹] قرار می‌گیرند [۱۷]. توزیع زمان راه‌اندازی سلولی U[1,20] نشان می‌دهد که نسبت میانگین متوسط زمان راه‌اندازی سلول به متوسط زمان پردازش قطعات موجود در هر سلول تقریباً ۲ به ۱ است (۱۰/٥ به ٥/٥).

مسأله نمونه

در یک شرکت تولیدی که شامل واحد های مختلفی است، در واحد رنگ شرکت، چهار نوع قطعه تولیدی این شرکت را رنگ می‌کند. این واحد دارای دو سلول تولیدی است که پنج ماشین آن در این دو سلول مستقر شده‌اند. در این دو سلول، زمان راه‌اندازی هر ماشین با توجه به این که هر یک از خانواده محصولات دارای چه توالی در پردازش هستند متفاوت است. این تفاوت بدین علت است که برای پردازش هر خانواده قطعه به‌وسیله ماشین های مستقر در واحد رنگ با توجه به این که خانواده قطعه موجود در توالی قبلی نیاز به چه نوع رنگی دارد زمان راه اندازی ماشین‌ها وابسته به توالی سلول‌هاست و متفاوت است. برای مثال زمان راه‌اندازی ماشینی که در رنگ آمیزی خانواده قطعه توالی اول از رنگ سفید استفاده کرد، جهت آماده‌سازی برای رنگ آمیزی خانواده قطعه موجود در توالی دوم که از رنگ قرمز استفاده می کند، اگر توالی خانواده قطعات که همان توالی سلول‌هاست تغییر کند، زمان راه‌اندازی نیز متفاوت خواهد بود. زمان‌های راه‌اندازی در این مثال نمونه در جدول شماره ٥ آمده است. برای مثال در این جدول زمان آماده سازی ماشین اول زمانی که ابتدا قطعات مربوط به سلول اول پردازش گردد؛ سپس قطعات مربوط به سلول دوم ٥ است. جدول (۲) ماتریس مسیر ساخت قطعات را نشان می‌دهد که نشان دهنده ماشین‌های مورد نیاز جهت پردازش بر روی قطعات است. همان‌طور که می‌بینیم، قطعه ۱ به ماشین های ۱، ۲ و ۳ نیاز دارد. جدول (۳) زمان پردازش هر یک از قطعات را نشان می‌دهد. همان‌طور که می بینید زمان پردازش قطعه ۱ روی ماشین ۱، ۲ و ۳ به ترتیب ٤، ٥ و ۱۲ است. در نهایت تعلق ماشین های واحد رنگ شرکت به هر یک از دو سلول در جدول (٤) آمده است.

جدول (۲): ماتریس مسیر ساخت قطعات (a_{ij})

قطعه \ ماشین	۱	۲	۳	۴	۵
۱	۱	۱	۱	۰	۰
۲	۱	۱	۰	۰	۰
۳	۰	۰	۱	۱	۱
۴	۰	۰	۱	۱	۰

جدول (۳): زمان پردازش قطعات بر روی ماشین‌ها (t_{ij})

قطعه \ ماشین	۱	۲	۳	۴	۵
۱	۴	۵	۱۲	۰	۰
۲	۶	۴	۰	۰	۰
۳	۰	۰	۷	۶	۱۴
۴	۰	۰	۵	۳	۰

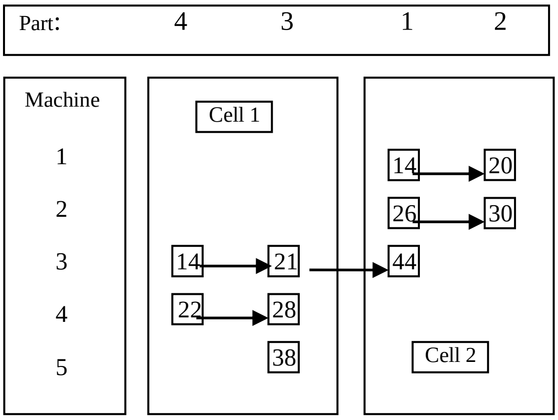
جدول (۴): ماتریس تخصیص ماشین‌ها به سلول‌ها (m_{cj})

سلول \ ماشین	۱	۲	۳	۴	۵
۱	۱	۱	۰	۰	۰
۲	۰	۰	۱	۱	۱

جدول (۵): زمان آماده سازی وابسته به توالی سلول‌ها (S_{ncj})

سلول-سلول \ ماشین	۱	۲	۳	۴	۵
۱-۱	۱۰	۷	۸	۳	۱۷
۲-۱	۵	۱۴	۹	۱۰	۶
۱-۲	۸	۳	۱۰	۱۰	۱۰
۲-۲	۱۲	۱۸	۹	۵	۹

با حل مدل ریاضی مربوط به مثال فوق با استفاده از نرم افزار لینگو ۸ جواب بهینه به‌صورت زیر به دست آمد:



شکل ۲: شمایی از توالی قطعات و سلول‌ها در آزمون ارائه شده

مقدار تابع هدف برای این مثال ۹٤ محاسبه شد که مدت زمان کل ساخت ٤٤ بوده (C_{max} = ٤٤) و با توجه به تخصیص قطعات، عدد حرکت بین سلولی وجود دارد. با توجه به نتایج حل، قطعات ۱ و ۲ به سلول ۱ و قطعات ۳ و ٤ به سلول ۲ اختصاص یافته‌اند. مدت زمان کل حل توسط لینگو ۸، ٥ ساعت و ٥٤ دقیقه بود. سایر اطلاعات مربوط به حل بهینه در شکل (۲) آمده است. در این شکل زمان تکمیل هر یک از قطعات را روی هر ماشین مشاهده می‌کنید.

جواب بهینه حاصل از حل مدل توسط نرم افزار لینگو بدین ترتیب است که در سلول ۱، ابتدا قطعه ۱ سپس قطعه ۲ و در سلول ۲، ابتدا قطعه ٤ سپس قطعه ۳ می‌بایست پردازش گردد. توالی سلول‌ها نیز بدین ترتیب است که ابتدا قطعات موجود در سلول ۲ با در نظر گرفتن توالی آنها پردازش می‌شوند و سپس قطعات موجود در سلول ۱.

تحلیل نتایج

برای پیاده‌سازی الگوریتم ژنتیک از برنامه ویژوال بیسیک (VB) استفاده شده است. پردازشگر اجرا کننده الگوریتم ژنتیک و مدل ریاضی لینگو یک کامپیوتر با مشخصات 2.4 GHz و 512 MB RAM است. با توجه به پیچیدگی حل مدل زمانبندی سیستم تولید سلولی به خصوص در ابعاد بزرگ محاسبه مقدار بهینه بسیار مشکل است؛ بنابراین معیار قضاوت، جواب نرم افزار لینگو ۸ به عنوان یک جواب نزدیک به بهینه است.

مقایسه جواب سایر مسایل در جدول ()٦ نشان داده شده است. با توجه به نتایج به دست آمده از حل مسأله توسط الگوریتم ژنتیک و نرم‌افزار لینگو ۸ دیده می‌شود که در ابعاد کوچک حل به‌دست امده از الگوریتم و نرم‌افزار مساوی هستند. این مطلب کارایی الگوریتم‌ها را نشان می‌دهد. با افزایش ابعاد مسأله (بزرگ‌تر از ٤٠) نرم افزار لینگو ۸ نمی‌تواند در زمانی قابل قبول مسأله را حل کند و الگوریتم ژنتیک در زمانی بسیار مناسب حلی بهینه یا نزدیک به بهینه ایجاد را می‌کند.

۳- نتیجه‌گیری

مدلی که در این مقاله پیشنهاد شده است، یک مدل قطعی برای حداقل کردن مدت زمان بیکاری و اختصاص قطعات به سلول‌ها با در راستای گرفتن زمان‌های راه‌اندازی وابسته به توالی سلول‌ها در جهت حداقل کردن زمان مورد نیاز برای تکمیل قطعات و حرکت بین سلولی در نظر گرفته شده است. بر اساس مطالعات انجام شده، این مسأله از نوع چندجمله‌ای نامعین سخت است است که با افزایش تعداد قطعات و ماشین‌ها حل آن از طریق نرم‌افزارهای بهینه‌سازی غیر ممکن می‌شود. رویکردهایی از قبیل شاخه و حد و برنامه‌ریزی پویا نیز با افزایش تعداد کارها دارای محدودیت زمان محاسبات و محدودیت ذخیره‌سازی در کامپیوتر است؛ از این رو استفاده از الگوریتم‌های ابتکاری و فراابتکاری می‌تواند موثر باشد. الگوریتم ژنتیک یکی از مهم‌ترین الگوریتم‌های فراابتکاری است که از آن برای بهینه‌سازی توابع مختلف استفاده می‌شود. الگوریتم ژنتیک، الگوریتم جستجویی است که در یک منطقه از جواب جستجو می‌کند و از فرایندهای ارزیابی زیست شناسی تقلید می‌کند. الگوریتم ژنتیک با توجه به مشخصات شرح داده شده برای حل مسأله مورد نظر این مقاله مورد استفاده قرار گرفت.

نتایج به دست آمده را می‌توان بصورت زیر خلاصه کرد:

- نتایج حاصل نشان می دهد که الگوریتم ژنتیک عملکرد قابل قبولی دارد و زمان رسیدن به حل بهینه را بسیار کاهش می‌دهد.
- با توسعه و گسترش مسایل، زمان محاسباتی توسط لینگو افزایش می‌یابد؛ در حالی که در حالت استفاده از الگوریتم فراابتکاری کم و افزایش نامحسوسی در مقابل زمان محاسباتی توسط لینگو دارد.
- با توجه به افزایش تنوع محصولات و حرکت صنایع به سمت استفاده از تولید سلولی در جهت بهره‌مندی از منافع حاصل از آن، استفاده از روش‌های معمولی تعیین توالی عملیات و زمانبندی تولید سلولی از عملکرد خوبی برخوردار نیست و استفاده از روش‌های فراابتکاری بسیار مورد توجه خواهد بود.

برای تحقیقات آتی پیشنهادهای زیر ارائه می گردد:

- از آنجا که مسأله به‌صورت ایستا در نظر گرفته شده است، بررسی آن در حالت پویا می‌تواند مفید باشد.
- در این مسأله شکست کارها مجاز نیست. در نظر گرفتن شکست در بعضی از صنایع ضروری است، لذا بررسی آن می‌تواند ارزشمند باشد.
- زمان‌های پردازش و موعدهای تحویل به‌صورت قطعی در نظر گرفته شده است که در برخی از صنایع می‌تواند به‌صورت احتمالی باشد.
- برای حل مسأله مورد نظر می‌توان از سایر الگوریتم‌های فراابتکاری مانند ممتیک، جستجوی پراکنده، شبیه‌سازی تبرید و غیره استفاده کرد.

جدول ۶: جواب‌های مسایل انتخابی

ردیف	تعداد قطعه	تعداد ماشین	تعداد سلول	روش حل	مقدار تابع هدف	زمان کل ساخت	زمان اجرای برنامه (ثانیه)
۱	۳	۲	۲	GA	۲۹	۲۹	۱۱
				لینگو	۲۹	۲۹	۰
۲	٤	۲	۲	GA	۲۸	۲۸	٥٦
				لینگو	۲۸	۲۸	۱۸٦
۳	۳	٤	۲	GA	۱۳۹	۳۹	۹۰
				لینگو	۱۳۹	۳۹	۷۲٥
٤	٤	٤	۲	GA	۱۳۷	۳۷	۱۷۸
				لینگو	۱۳۷	۳۷	۱۹۹٥
٥	٤	٥	۲	GA	۱۹۹	٤۹	۳٤٥
				لینگو	۱۹۹	٤۹	۱۷۷۲۳
٦	٤	۷	۲	GA	۲٥٦	٥٦	٥۰۱
				لینگو	--	--	۲۱۲۷٤
۷	٥	۷	۲	GA	۲۳٦	۸٦	٥۱۰
				لینگو	--	--	۲۸۸۰۰*
۸	٥	۸	۲	GA	۳٤۲	۹۲	٦۰٦
				لینگو	--	--	۲۸۸۰۰*
۹	٦	۸	۲	GA	۳۷۷	۷۷	۷۷٦
				Lingo	--	--	۲۸۸۰۰*
۱۰	۷	۸	۳	GA	۳۹۱	۹۱	۱۰٤۳
				Lingo	--	--	۲۸۸۰۰*
۱۱	۸	۸	۳	GA	٥٥٥	۱۰٥	۱۱۰۰
				لینگو	--	--	۲۸۸۰۰*
۱۲	۱۰	۸	۳	GA	۸٤۱	۹۱	۱۲۳۳
				لینگو	--	--	۲۸۸۰۰*
۱۳	۲۰	۱۰	٤	GA		۲٤۰	۱٥٤٥
				لینگو	--	--	۲۸۸۰۰*
۱٤	۳۰	۱۲	٥	GA		۳۸۷	۲۱۲۷
				لینگو	--	--	۲۸۸۰۰*
۱٥	٤۰	۱٥	٦	GA		٥۷۸	۲۷۳٦
				لینگو	--	--	۲۸۸۰۰*

* تا این زمان، جواب بهینه‌ای برای مسأله توسط نرم‌افزار لینگو ۸ به‌دست نیامده است.

۴- مراجع

[۱] Soleymanpour, M., Vrat, P., Shankar, R., "A transiently chaotic neural network approach to the design of cellular manufacturing", International Journal of Production Research, Vol. 40 (10), p.p. 2225–2244, 2002.

[۲] Yoshida, T., Hitomi, K., "Optimal two-stage production scheduling with setup times separated", AIIE Transactions, Vol. 11, p.p. 261–263, 1979.

[۳] Sekiguchi, Y., "Optimal schedule in a GT-type flow-shop under series-parallel precedence constraints", Journal of the Operations Research Society of Japan, Vol. 26, p.p. 226–251, 1983.

[٤] Baker, K.R., "Scheduling groups of jobs in the two machine flow shop", Mathematical and Computer Modeling, Vol. 13, p.p. 29-36, 1990.

[٥] Yang, D.L., Chern, M..S., "Two-machine flow shop group scheduling problem", Computers & Operations Research, Vol. 27, p.p. 975–985, 2000.

[٦] Hitomi, K., Ham, I., "Operations scheduling for group technology applications", Annals of the CIRP, Vol. 25 (1), p.p. 419–422, 1976.

[۷] Logendran, R., Sriskandarajah, C., "Two-machine group scheduling problem with blocking and anticipatory setups", European Journal of Operational Research, Vol. 69 (3), p.p., 467–481., 1993.

[۸] Logendran, R., Nudtasomboon, N., "Minimizing the makespan of a group scheduling problem: A new heuristic", International Journal of Production Economics, Vol. 22, p.p. 217–230, 1991.

[۹] Nawaz, M., Enscore, J.E.E., Ham, I., "A heuristic algorithm for the m-machine, n-job flow shop sequencing problem", OMEGA: International Journal of Management Science, Vol. 11 (1), p.p. 91–95., 1983.

[۱۰] Wemmerl. ov, U., Vakharia, A.J., "Job and family scheduling of a flow-line manufacturing cell: A simulation study", IIE Transactions, Vol. 23 (4), p.p. 383–393, 1991.

[۱۱] Mahmoodi, F., Dooley, K.J., "Group scheduling and order releasing: Review and foundations for research", Production Planning & Control, Vol. 3 (1), p.p. 70–80, 1992.

[۱۲] Eom, D.H., Shin, H.J., Kwun, I.H., Shim, J.K., Kim, S.S., "Scheduling jobs on parallel machines with sequence-dependent family setup times". Advanced Manufacturing Technology, Vol. 19, p.p. 926–932, 2002.

[۱۳] Logendran, R., Mai, L., Talkington, D., "Combined heuristics for bi-level group scheduling problems", International Journal of Production Economics, Vol. 38, p.p. 133–145, 1995.

[۱٤] Campbell, H.G., Dudek, R.A., Smith, M.L., "A heuristic algorithm for the n job, m machine sequencing problem", Management Science, Vol. 16, p.p. 630-637. 1970.

[۱٥] Yang, W. H., Liao, C. J., "Group scheduling on two cells with inter-cell movement", Computers & Operations Research, Vol. 23 (10), p.p. 997–1006, 1996.

[۱٦] Schaller, J., "A new lower bound for the flow shop group scheduling problem", Computers & Industrial Engineering, Vol. 41, p.p. 151–161, 2001.

[۱۷] Schaller, J.E., Gupta, J.N.D., Vakharia, A.J., "Scheduling a flowline manufacturing cell with sequence dependent family setup times". European Journal of Operational Research, Vol. 125, p.p. 324–339, 2000.

[۱۸] Franca, P.M., Gupta, J.N.D., Mendes, A.S., Moscato, P., Veltink, K.J., "Evolutionary algorithms for scheduling a flowshop manufacturing cell with sequence dependent family setups". Computers & Industrial Engineering, Vol. 48 (3), p.p. 491–506, 2005.

[۱۹] Hendizadeh, S.H., Faramarzi, H., Mansouri, S.A., "Meta-heuristics for scheduling a flowline manufacturing cell with sequence dependent family setup times". International Journal of Production Economics, Vol. 111 (2), p.p. 593-605, 2008.

[۲۰] Tavakkoli-Moghaddam, R., Gholipour-Kanani, Y., Cheraghalizadeh, R., "A genetic algorithm and memetic algorithm to sequencing and scheduling of cellular manufacturing systems", International Journal of Management Science and Engineering Management, Vol. 3 (2), pp. 119-130, 2008.

[۲۱] Holland, J.; Adaptation natural and artificial systems, University of Michigan press, An Arbor, MI, MIT Press, Cambridge, Ma, 1975.

[۲۲] Gen, M., Cheng, R.; Genetic algorithms & engineering design, New York, A Wiley Interscience Publication, 1997.

۵- زیرنویس‌ها

[1] Cellular Manufacturing Systems

[2] Non–Deterministic Polynomial Hard (NP-Hard)

[3] Genetic Algorithm

[5] Multi-Criteria Mixed-Integer Programming Model

[5] Crossover

[6] Mutation

[7] Stochastic Sampling with Replacement (SSR)

[9] Roulette Wheel Selection

[10] Small Setup Time (SSU)