

## مدل‌سازی و زمان‌بندی مسائل جریان کارگاهی مونتاژ دو مرحله‌ای با ماشین‌های مونتاژ غیرهمسان

مهتاب باشی ورشوساز\*، بهمن نادری و محمد محمدی  
گروه مهندسی صنایع، دانشکده فنی مهندسی، دانشگاه خوارزمی، تهران

(دریافت مقاله: ۱۳۹۵/۰۸/۲۳ - دریافت نسخه نهایی: ۱۳۹۶/۱۲/۲۲)

چکیده- این پژوهش به بررسی مسئله زمان‌بندی جریان کارگاهی مونتاژ دو مرحله‌ای می‌پردازد. در این مسئله تعدادی محصول مشابه، هر یک متشکل از چندین قطعه مختلف سفارش داده شده است. ساخت هر قطعه شامل  $m$  عملیات است که در ایستگاه اول به وسیله  $m$  ماشین مختلف انجام می‌شوند. قطعات پس از تکمیل فرایند ساخت، در ایستگاه دوم به وسیله چندین ماشین ناهمسان به محصول نهایی مونتاژ می‌شوند. هدف مسئله، یافتن توالی بهینه قطعات در ایستگاه ساخت، تخصیص و توالی بهینه محصولات در ایستگاه مونتاژ است. یک مدل برنامه‌ریزی خطی عدد صحیح مختلط و دو الگوریتم فرا ابتکاری ازدحام ذرات با جستجوی محلی (MPSO) و شبیه‌سازی تبرید (SA) برای حل مسئله پیشنهاد شده است. برای ارزیابی عملکرد مدل و الگوریتم‌های پیشنهادی آزمایشات محاسباتی مطرح شده که نتایج نشان‌دهنده برتری الگوریتم MPSO نسبت به SA است.

واژه‌های کلیدی: مونتاژ جریان کارگاهی دو مرحله‌ای، ماشین‌های مونتاژ غیرهمسان، مدل‌سازی ریاضی، زمان‌های راه‌اندازی غیروابسته به توالی، الگوریتم‌های فرا ابتکاری.

## Modeling and Scheduling Two-stage Assembly Flow Shop Problems with Non-Identical Assembly Machines

M. Bashi Varshosaz\*, B. Naderi and M. Mohammadi

Department of Industrial Engineering, Faculty of Engineering, Kharazami University, Tehran, Iran.

**Abstract:** The purpose of this research is to deal with the problem of two-stage assembly flow shop scheduling. A number of single-item products (identical) each formed of several different parts are ordered. Each part has  $m$  operations done at the first

\*: مسئول مکاتبات، پست الکترونیکی: mahtab.varshosaz@gmail.com

stage with  $m$  different machines. After manufacturing the parts, they are assembled into a final product with some non-identical machines. The purpose of the problem is to find the optimal sequence of the parts in the manufacturing stage, allocation and the optimal sequence of the products in the assembly stage. A mixed integer linear programming model and two metaheuristic algorithms, which are particle swarm with local search (MPSO) and simulated annealing (SA), are presented to solve this problem. Computational experiments are conducted to evaluate the performance of the proposed model and algorithms. The results show that the MPSO algorithm performs better than the SA one.

**Keywords:** Two-stage assembly flow shop, Non-identical assembly machines, Mathematical modeling, Sequence independent setup times, Metaheuristics.

## فهرست علائم

احتمال اعمال هر عملگر	prob	ضریب کاهش دما	Alpha
زمان مونتاژ محصول $k$ روی ماشین $l$	$q_{k,l}$	ضریب کاهش دما در MIE	beta
انحراف درصدی نسبی	RPD	متغیر پیوسته برای زمان تکمیل قطعه $j$ کار $k$ روی ماشین $i$	$c_{k,j,i}$
زمان آماده‌سازی قطعه $j$ ام روی ماشین $i$	$s_{j,i}$	بیشترین زمان تکمیل کارها	$C_{max}$
دمای نهایی در MIE	Tf	متغیر پیوسته برای زمان تکمیل سفارش $k$ در مرحله دوم	$f_k$
تعداد ماشین‌های مرحله دوم (مونتاژ)	w	تعداد قطعات هر سفارش (محصول)	g
متغیر باینری است که اگر در مرحله اول قطعه $j'$ ام کار $k'$ بلافاصله بعد از قطعه $j$ کار $k$ روی همه ماشین‌ها قرارگیرد، یک و در غیر این صورت صفر است ( $k' < k$ ).	$x_{k',j,k,j'}$	اندیس برای ماشین‌های مرحله اول $\{1, 2, \dots, m\}$	i
متغیر باینری است که اگر در مرحله اول قطعه $j'$ کار $k$ بلافاصله بعد از قطعه $j$ کار باشد، یک و در غیر این صورت صفر است ( $j' \neq j$ ).	$y_{k,j,j'}$	تعداد تکرار در هر دما	It
متغیر باینری است که اگر در مرحله دوم کار $k'$ بلافاصله بعد از کار $k$ روی ماشین $l$ باشد، یک و در غیر این صورت صفر است ( $k' \neq k$ ).	$z_{k,k',l}$	اندیس آیتم قطعه‌ها $\{0, 1, 2, \dots, g\}$ ، $j \in \{1, 2, \dots, g\}$	$j, j'$
		اندیس برای سفارشات $\{0, 1, 2, \dots, n\}$ ، $k \in \{1, 2, \dots, n\}$	$k, k'$
		اندیس برای ماشین‌های مرحله دوم $\{1, 2, \dots, w\}$	l
		تعداد ماشین‌های مرحله اول (ساخت)	m
		عدد مثبت بسیار بزرگ	M
		تعداد سفارشات (محصولات)	n
		زمان پردازش قطعه $j$ روی ماشین $i$	$p_{j,i}$
		احتمال اعمال روش MIE روی هر ذره	pMIE
		اندازه جمعیت	popsiz

## ۱- مقدمه

زمان‌بندی مناسب برای ورود کارها به این مرحله، تخصیص مناسب کار و اپراتور به ماشین‌ها، در نظر گرفتن قابلیت و دسترس بودن ماشین‌ها و... می‌توان از منابع موجود و زمان، حداکثر بهره را برد. معمولاً زمان‌بندی برای ساخت قطعات و برنامه‌ریزی برای عملیات مونتاژ به صورت مجزا مورد مطالعه قرار گرفته است [۱ و ۲] که منجر به نتایج ایده‌آلی برای کل سیستم تولیدی نخواهد شد. بنابراین در تحقیقات اخیر در نظر

امروزه درصد عمده‌ای از تولیدات دارای حداقل یک مرحله مونتاژ هستند. بنابراین لازم است که در برنامه‌ریزی تولید و زمان‌بندی به فاز مونتاژ نیز توجه کرد. به طور کلی تولید کارآمد و بهره‌وری بالا وقتی امکان‌پذیر است که صرفه‌جویی در زمان و هزینه و توجه به کیفیت در تمام مراحل تولید وجود داشته باشد. مرحله مونتاژ هم یکی از مراحل تولید است که با

گرفتن هم‌زمان این دو ایستگاه در مسائل زمان‌بندی بیشتر مورد توجه است [۳].

اولین مطالعات در زمینه مسائل زمان‌بندی جریان کارگاهی نوع مونتاژ توسط لی و همکاران [۴] در سال ۱۹۹۳ انجام شده است. آنها یک مسئله زمان‌بندی فلوشاپ مونتاژ دو ایستگاه با تابع هدف میکسپن را مطالعه کردند که در این مسئله ساده هر محصول با مونتاژ دو نوع قطعه ساخته می‌شود. جزء اول هر محصول روی ماشین اول و جزء دوم روی ماشین دوم پردازش می‌شود. در نهایت ماشین سوم دو قطعه را به محصول نهایی مونتاژ می‌کند. آنها یک رویکرد حل شاخه و کران و یک روش حل تقریبی را ارائه دادند. پاتس و همکاران [۵] مدل [۴] را به مواردی با چندین ماشین ساخت گسترش دادند به طوری که  $m$  ماشین در مرحله اول و یک ماشین در مرحله دوم وجود داشت و یک الگوریتم هیورستیک برای حداقل کردن میکسپن پیشنهاد دادند. لی و پاتس در مقاله‌های خود مستقلاً اثبات کردند که مسئله جریان کارگاهی مونتاژ حتی با دو ماشین ساخت در ایستگاه اول نیز، NP-Hard است. هاری و پاتس [۶] مسئله‌ای مشابه با [۵] در نظر گرفتند و یک الگوریتم شاخه و کران ارائه دادند. چنگ و وانگ [۷] حداقل کردن میکسپن در یک زمان‌بندی فلوشاپ دو ماشین را با یک ساختار خاص در نظر گرفتند و برای چندین حالت خاص زمان‌بندی‌های بهینه‌ای به دست آوردند. سانگ و جان [۸] یک مسئله زمان‌بندی مونتاژ دو ایستگاه را با هدف حداقل‌سازی میکسپن مورد مطالعه قرار دادند. در مسئله آنها  $n$  کار وجود دارد که هر کار شامل دو جزء است. یکی از دو جزء با محدودیت زمان تحویل ساخته می‌شود.

یوکویاما [۹] و یوکویاما و سانتس [۱] یک زمان‌بندی ترکیبی را برای سیستم‌های تولیدی شامل ماشین‌کاری قطعات و عملیات مونتاژ مورد بررسی قرار دادند و یک روش حل براساس شاخه و کران توسعه دادند. در این مطالعه چندین محصول از انواع مختلف سفارش داده شده که قطعات مورد نیاز برای این محصولات در یک فلوشاپ ساخته می‌شوند و هر

محصول توسط عملیات مونتاژ قطعات به صورت سلسه مراتبی تولید می‌شود. قطعات به اولین زیر مونتاژ، مونتاژ می‌شوند و چندین قطعه دیگر و زیر مونتاژ اول به زیر مونتاژ دوم مونتاژ می‌شوند. این عملیات مونتاژ ادامه می‌یابد تا آخرین زیر مونتاژ که محصول نهایی است تولید شود. الله‌وردی و انزی [۳] یک مسئله زمان‌بندی مونتاژ دو ایستگاه را که  $m$  ماشین در مرحله اول و یک ماشین مونتاژ در مرحله دوم دارد با در نظر گرفتن زمان‌های راه‌اندازی مستقل از زمان‌های پردازش بررسی کردند. آنها یک رابطه غالب و سه الگوریتم هیورستیک یعنی الگوریتم جستجوی ممنوعه ترکیبی و الگوریتم تکاملی تفاضلی خود تطبیقی (SDE) و یک الگوریتم تکاملی تفاضلی جدید (NSDE) را ارائه دادند.

با بررسی ادبیات موضوع درمی‌یابیم که اغلب یک ماشین برای مرحله دوم در نظر گرفته شده، درحالی که با توجه به پیشرفت تکنولوژی و پیچیدگی محصولات، وجود یک ماشین مونتاژ با دنیای واقعی سازگاری چندانی ندارد. بنابراین در این پژوهش مسئله مونتاژ جریان کارگاهی دو مرحله‌ای را با وجود چندین ماشین موازی ناهمسان در مرحله دوم بررسی می‌کنیم. مسئله مورد بررسی در این تحقیق، مونتاژ در محیط جریان کارگاهی با در نظر گرفتن زمان‌های آماده‌سازی غیروابسته به توالی با هدف کاهش میکسپن است. در این مسئله تعدادی محصول یکسان وجود دارد که هر یک با مونتاژ چندین قطعه مختلف ساخته می‌شوند. ابتدا قطعات در ایستگاه اول با  $m$  ماشین مختلف سری، ساخته شده و سپس در یک ایستگاه مونتاژ با چندین ماشین ناهمسان به محصول نهایی مونتاژ می‌شوند. هنگامی که ماشین‌های ساخت شروع به پردازش قطعات می‌کنند و یا حین تغییر آیتم قطعات، عملیات راه‌اندازی غیروابسته به توالی مورد نیاز است. زمان‌های راه‌اندازی در مرحله دوم نادیده گرفته شده است.

ادامه مقاله به این صورت سازماندهی شده است: در بخش دوم به تعریف مسئله و بیان یک مثال عددی ساده

می پردازیم. در بخش سوم سعی بر این است که با ارائه مدل ریاضی عدد صحیح مختلط، ضمن ارائه مدلی کارا و ساده بتوان ابعاد کوچک مسئله را به طور دقیق حل کرد. در بخش چهارم با توجه به پیچیدگی مسئله در سائزهای بزرگ، الگوریتم‌های فرا ابتکاری برای حل به کار گرفته شده است. در بخش پنجم عملکرد مدل و الگوریتم‌ها مورد بررسی و مقایسه قرار می‌گیرد و در بخش آخر نتیجه‌گیری پژوهش بیان می‌شود.

## ۲- تعریف مسئله

مسئله مورد نظر این پژوهش، مونتاژ جریان کارگاهی دو مرحله‌ای است و به این صورت تعریف می‌شود:  $n$  کار مشابه در زمان صفر در دسترس هستند. هر کار از  $g$  قطعه تشکیل شده است که ساخت هر قطعه شامل  $m$  عملیات مختلف است که در مرحله اول روی  $m$  ماشین مختلف سری انجام می‌شوند. هر ماشین در هر لحظه قادر به پردازش یک قطعه است و قطع کار مجاز نیست؛ یعنی پس از اتمام عملیات هر قطعه روی هر ماشین، عملیات قطعه جدید آغاز می‌شود. همه ماشین‌های ساخت نیازمند آماده‌سازی غیروابسته به توالی هستند. پس از پردازش همه قطعات یک محصول در مرحله اول امکان شروع فرایند مونتاژ در مرحله دوم وجود دارد. در مرحله دوم  $W$  ماشین مونتاژ غیرهمسان<sup>۱</sup> وجود دارد که به آماده‌سازی نیاز ندارند. هر محصول تنها روی یکی از ماشین‌های مرحله دوم مونتاژ می‌شود. نکته قابل ذکر این است که ماشینی که به این کار اختصاص می‌یابد، لزوماً ماشین بیکار در آن لحظه نیست و ممکن است قطعات پس از تکمیل در مرحله اول منتظر خالی شدن یک ماشین مونتاژ بمانند. علت این انتظار، غیرهمسان بودن ماشین‌های مونتاژ است که یک کار یکسان را با زمان‌های متفاوت مونتاژ می‌کنند. تابع هدف در نظر گرفته شده برای این مسئله کمینه‌سازی میکسپن<sup>۲</sup> است که بیانگر کاهش حداکثر زمان تکمیل کارها در مرحله دوم است. در ادامه

مثالی برای توصیف فضای کلی مسئله بیان می‌شود.

## مثال عددی

مسئله‌ای با چهار محصول و هر محصول متشکل از سه قطعه مفروض است. دو ماشین ساخت در مرحله اول و دو ماشین مونتاژ در مرحله دوم در نظر گرفته شده است. توالی بهینه قطعات در ایستگاه اول به صورت  $\{1, 2, 1, 2, 3, 3, 3, 2, 3, 3, 1, 2\}$  است، که این ترتیب روی تمامی ماشین‌های ساخت یکسان است. در مرحله دوم کار سه روی ماشین دوم و کارهای  $\{4, 2, 1\}$  روی ماشین اول مونتاژ می‌شوند. پارامترهای مربوط به زمان‌های پردازش و آماده‌سازی در جدول‌های (۱) و (۲) آورده شده است. زمان مونتاژ محصولات روی ماشین‌های مرحله دوم  $p(M_{A1}) = 30, p(M_{A2}) = 50$  است. شکل (۱) نمودار گانت پردازش و مونتاژ کارها را نشان می‌دهد.

شکل (۱) چگونگی پردازش و مونتاژ کارها را نشان می‌دهد. عملیات ساخت مربوط به قطعات در مرحله اول به طور سری و به ترتیب روی ماشین اول و سپس ماشین دوم انجام می‌شود. پردازش یک قطعه در مرحله اول وقتی به پایان می‌رسد که همه عملیات مربوط به آن به اتمام رسیده باشد. پس از اتمام مرحله اول، قطعات برای ورود به مرحله دوم آماده هستند و یکی از سه حالت زیر اتفاق می‌افتد:

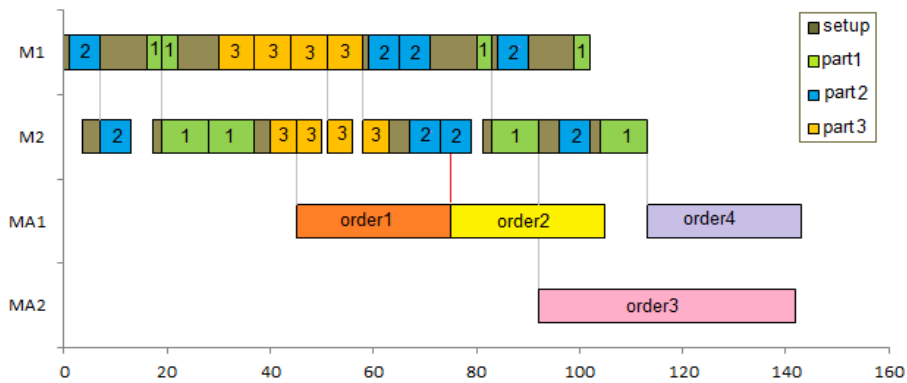
۱- زمانی که پردازش اولیه قطعات یک محصول تمام شده است، ماشین مونتاژ مورد نظر خالی باشد که در این حالت نه ماشین و نه محصول بیکار می‌ماند.

۲- مونتاژ محصول قبلی روی ماشین مونتاژ، زودتر از پردازش اولیه کار فعلی به اتمام می‌رسد. بنابراین ماشین بیکار می‌ماند تا عملیات اولیه کار فعلی به پایان برسد. این شرایط برای اولین ماشین مونتاژ پس از تکمیل سفارش دوم ایجاد شده است.

۳- مراحل پردازش اولیه قطعات به اتمام رسیده ولی ماشین مونتاژ مربوطه مشغول مونتاژ محصول قبلی است. در این حالت

S <sub>ji</sub>	۱	۲
۱	۹	۲
۲	۱	۴
۳	۸	۳

P <sub>ji</sub>	۱	۲
۱	۳	۹
۲	۶	۶
۳	۷	۵



شکل ۱- نمودار گانت زمان پردازش و مونتاژکارها در مثال عددی

مدل ریاضی پیشنهادی و بیان محدودیت‌ها:

$$\text{Min } C_{\max} \quad (1)$$

Subject to:

$$\forall j, k' \quad \sum_{\substack{k=1, \\ k \neq k'}}^n \sum_{j=1}^g x_{k,j,k',j'} + \sum_{\substack{j=1, \\ j \neq j'}}^g y_{k',j,j'} = 1 \quad (2)$$

$$\forall j > 0, k > 0 \quad \sum_{\substack{k=1, \\ k \neq k'}}^n \sum_{j=1}^g x_{k,j,k',j'} + \sum_{\substack{j=1, \\ j \neq j'}}^g y_{k',j,j'} \leq 1 \quad (3)$$

$$\sum_{k=1}^n \sum_{j=1}^g x_{0,0,k',j'} = 1 \quad (4)$$

$$\forall k > 0, 0 < j < g, j < j' \quad y_{k,j,j} + y_{k',j',j} \leq 1 \quad (5)$$

$$\forall 0 < k < n, k' > k, j > 0 \quad x_{k,j,k',j} + x_{k',j,k,j} \leq 1 \quad (6)$$

$$\forall k' \neq k, j \neq j', k > 0, j > 0 \quad x_{k,j,k',j'} + x_{k',j',k,j} \leq 1 \quad (7)$$

قطعات تا اتمام مونتاژ محصول قبلی اجازه ورود به مرحله دوم را ندارند. این شرایط برای سفارش دوم ایجاد شده است.

### ۳- مدل ریاضی پیشنهادی

در این بخش برای مسئله مورد بررسی، یک مدل ریاضی برنامه‌ریزی خطی عدد صحیح مختلط ارائه شده است. این نوع از برنامه‌ریزی به دلیل سادگی و انعطاف‌پذیری، کارآمد است و همچنین در ابعاد کوچک توسط نرم‌افزارهای بهینه‌سازی قابل حل است. مدل پیشنهاد شده با به‌کارگیری سه متغیر باینری و دو متغیر پیوسته و ۲۴ محدودیت خطی، مسئله را مدل‌سازی می‌کند. این مدل مبتنی بر یافتن توالی نسبی<sup>۳</sup> است یعنی متغیرهای باینری استفاده شده، توالی نسبی کارهای مختلف روی ماشین‌ها را نشان می‌دهند. این نوع تعریف متغیر باینری اولین بار توسط ریوس-مرکادو و بارد ارائه شد [۱۰]. در این مدل یک کار و یک قطعه مجازی تعریف می‌شود که اولین قطعه و کاری خواهد بود که در توالی پردازش می‌شود. به‌همین جهت، احتیاج است که شماره‌های  $k$  و  $j$  از صفر شروع شوند.

$$\forall l, k, k', k' > k$$

$$z_{k,k',l} \in \{0,1\} \quad (25)$$

محدودیت (۱) نشان‌دهنده تابع هدف است. مجموعه محدودیت (۲) مشخص می‌کند که هر قطعه از هر کار تنها یک بار در توالی وجود داشته باشد. مجموعه محدودیت (۳) اشاره دارد که هر قطعه روی هر ماشین می‌تواند دقیقاً یک قطعه بعدی از کار دیگر یا همان کار داشته باشد (به جز آخرین قطعه). محدودیت (۴) اولین قطعه پردازش شده روی هر ماشین را تعیین می‌کند. مجموعه محدودیت‌های (۵)، (۶) و (۷) از ایجاد جواب‌های غیرموجه بدین صورت که یک کار هم‌زمان به‌عنوان کار بعدی و قبلی برای دیگری باشد، جلوگیری می‌کند. مجموعه محدودیت (۸) بیان می‌کند که در مرحله دوم اگر کار  $k'$  به‌عنوان کار بعدی یک کار روی ماشین  $l$  ظاهر شد، باید حداکثر یک‌بار به عنوان کار قبلی یک کار دیگر روی همان ماشین ظاهر شود. مجموعه محدودیت (۹) اشاره دارد به اینکه هر کار روی هر ماشین در مرحله دوم حداکثر یک‌بار به‌عنوان کار قبلی یک کار دیگر است. مجموعه محدودیت (۱۰) بیان می‌کند که هر کار روی هر ماشین در مرحله دوم حداکثر یک‌بار به‌عنوان کار بعدی کار دیگر باشد (به جز آخرین کار). مجموعه محدودیت (۱۱) اولین کار روی ماشین  $l$  در مرحله دوم را نشان می‌دهد. مجموعه محدودیت (۱۲) زمان تکمیل هر قطعه کار اول روی هر ماشین در مرحله اول را مشخص می‌کند. مجموعه محدودیت (۱۳) این اطمینان را حاصل می‌کند که بعد از پایان عملیات هر قطعه روی هر ماشین در مرحله اول، بلافاصله عملیاتش روی ماشین بعدی شروع شود. علاوه بر آن این مجموعه محدودیت مشخص می‌کند که یک قطعه هم‌زمان نمی‌تواند توسط دو ماشین پردازش شود. مجموعه محدودیت (۱۴) مشخص می‌کند که پس از اتمام پردازش هر قطعه از کاری، امکان شروع پردازش قطعه مشابه از کار بعدی وجود دارد. مجموعه محدودیت‌های (۱۵)، (۱۶) و (۱۷) بیان می‌کنند که تا زمانی که پردازش یک قطعه روی یک ماشین در مرحله اول تمام نشده، نمی‌توان پردازش قطعه بعدی را شروع کرد. یا به‌عبارتی یک ماشین نمی‌تواند بیش از یک قطعه در یک زمان پردازش کند. محدودیت‌های (۱۸) و (۱۹) و (۲۰) نشان‌دهنده زمان اتمام کارها

$$\forall k', l$$

$$\sum_{k=0}^n z_{k,k',l} \geq \sum_{k=0}^n z_{k',k,l} \quad (8)$$

$$\forall k'$$

$$\sum_{k=0, k \neq k'}^n \sum_{l=1}^w z_{k,k',l} = 1 \quad (9)$$

$$\forall k'$$

$$\sum_{k=1, k \neq k'}^n \sum_{l=1}^w z_{k',k,l} \leq 1 \quad (10)$$

$$\forall l$$

$$\sum_{k'=1}^n z_{0,k',l} = 1 \quad (11)$$

$$\forall i, j > 0$$

$$c_{1,j,i} \geq p_{j,i} + s_{j,i} \quad (12)$$

$$\forall k > 0, j > 0, i > 1$$

$$c_{k,j,i} \geq c_{k,j,i-1} + p_{j,i} \quad (13)$$

$$\forall k > 0, j > 0, i$$

$$c_{k,j,i} \geq c_{k-1,j,i} + p_{j,i} \quad (14)$$

$$\forall j > 0, i, k' \neq k, 0 < k$$

$$c_{k',j,i} \geq c_{k,j,i} + p_{j,i} + M(x_{k,j,k',j} - 1) \quad (15)$$

$$\forall i, j \neq j', k \neq k'$$

$$k > 0, j > 0$$

$$c_{k',j',i} \geq c_{k,j,i} + s_{j',i} + p_{j',i} + M(x_{k,j,k',j'} - 1) \quad (16)$$

$$\forall i, k > 0, j \neq j', j > 0$$

$$c_{k,j',i} \geq c_{k,j,i} + s_{j',i} + p_{j',i} + M(y_{k,j,j'} - 1) \quad (17)$$

$$\forall k > 0, k', k \neq k', l$$

$$f_{k'} \geq f_k + q_{k',l} + M(z_{k,k',l} - 1) \quad (18)$$

$$\forall k > 0, k', k \neq k', j > 0, i = m, l$$

$$f_{k'} \geq c_{k',j,i} + q_{k',l} + M(z_{k,k',l} - 1) \quad (19)$$

$$\forall k' = 1, k = 0, i = m, j > 0, l$$

$$f_{k'} \geq c_{k',j,i} + q_{k',l} + M(z_{k,k',l} - 1) \quad (20)$$

$$\forall k, j, i$$

$$c_{\max} \geq f_n \quad (21)$$

$$c_{k,j,i} \cdot c_{\max} \cdot f_k \geq 0 \quad (22)$$

$$\forall k' > k, j, j'$$

$$x_{k,j,k',j'} \in \{0,1\} \quad (23)$$

$$\forall k, j' \neq j$$

$$y_{k,j,j'} \in \{0,1\} \quad (24)$$

$g[n(1/\delta g + 0/\delta n g + 0/\delta n + 0/\delta) + g] + wn(n+1)/2$	تعداد متغیرهای باینری
$(n+1)(1+m(g+1))+1$	تعداد متغیرهای پیوسته
$g[m(n^2g+n+2)+g(1/\delta n^2+n+1)] + w[n(1/\delta n + 0/\delta) + g(n^2-n+1)+1] + m(n+1) + 3n + 4$	تعداد محدودیت‌های خطی

۱	۱	۲	۲	۳	۳	۱	۲	۳
---	---	---	---	---	---	---	---	---

شکل ۲- نحوه نمایش یک جواب

جستجوی محلی در الگوریتم PSO یک روش کارآمد برای گرفتن یک جواب بهتر است [۱۲، ۱۳، ۱۴ و ۱۵]. یک روش بهبود چندگانه ذرات و پذیرش ذرات جدید براساس الگوریتم SA برای افزایش توانایی جستجوی محلی الگوریتم PSO به کار گرفته شده است که در ادامه توضیح داده خواهد شد.

#### ۴-۱-۱- تولید جمعیت اولیه

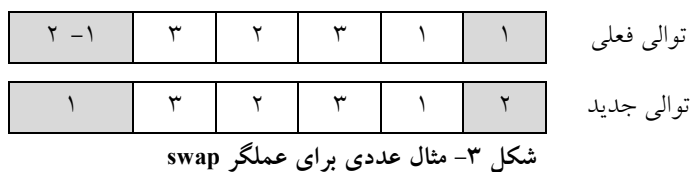
گام نخست در استفاده از الگوریتم‌های فرا ابتکاری، شیوه مناسب نمایش جواب‌ها است که باعث شناساندن یک جواب کاندید برای الگوریتم می‌شود و در جستجوی مؤثر هر الگوریتم نقشی مهم ایفا می‌کند. دو نوع شیوه نمایش جواب در ادبیات برای مسائل زمان‌بندی وجود دارد، نوع اول نمایش مبتنی بر عملیات [۱۶] است که مناسب فضای جواب گسسته است و نوع دیگر بردار اعداد حقیقی است که به راحتی می‌تواند از یک موقعیت در فضای پیوسته به فضای گسسته انتقال یابد. با توجه به گستردگی استفاده از شیوه نمایش مبتنی بر عملیات و سادگی پیاده‌سازی آن، برای الگوریتم‌های به کار رفته در این پژوهش از این روش استفاده می‌شود. برای مثال مسئله‌ای با سه کار و هر کار سه قطعه را در نظر بگیرید، در نتیجه ۹ قطعه برای پردازش وجود دارد. توالی شکل (۲) می‌تواند یک جواب اولیه برای الگوریتم باشد. عدد شماره هر قطعه به تعداد کارها تکرار می‌شود. به طور مثال هفتمین بلوک، اولین قطعه کار سوم را

روی ماشین مونتاژ است. محدودیت (۲۱) مقدار میکسپن را محاسبه می‌کند. متغیرهای مدل توسط مجموعه محدودیت‌های (۲۲-۲۵) تعریف می‌شوند. جدول (۳) تعداد متغیرها و محدودیت‌های مورد نیاز برای مدل‌سازی یک مسئله با  $n$  محصول و  $g$  قطعه و  $m$  ماشین ساخت و  $w$  ماشین مونتاژ را نشان می‌دهد.

#### ۴- الگوریتم‌های فرا ابتکاری پیشنهادی برای حل مسائل

##### ۴-۱- الگوریتم ازدحام ذرات

PSO<sup>۴</sup> یک الگوریتم بهینه‌سازی مبتنی بر جمعیت است که توسط کندی و ابرهات در سال ۱۹۹۵ پیشنهاد شد [۱۱] و از رفتار اجتماعی دسته پرندگان و یا دسته ماهی‌ها الهام گرفته شده است. این الگوریتم با یک جمعیت از جواب‌های تصادفی که ذره نامیده می‌شوند، آغاز می‌شود و به وسیله تولید نسل‌های جدید برای رسیدن به جواب بهینه جستجو می‌کند. در هر تکرار، هر ذره توسط دو جواب خوب به روز می‌شود. یکی از این دو جواب، بهترین جواب کل جمعیت و دیگری بهترین جوابی است که تاکنون آن ذره به آن دست یافته است. الگوریتم PSO برای بهینه‌سازی مسائل غیرخطی با فضای پیوسته ارائه شده است، اما مسئله زمان‌بندی فلوشاپ مونتاژ یک مسئله ترکیبی با فضای حل گسسته است که با به کار بردن عملگرهای تقاطع برای ایجاد جمعیت جدید می‌توان این الگوریتم را برای مسائل فضای گسسته مناسب کرد. همچنین افزایش توانایی



نشان می‌دهد چون عدد یک، سه بار تکرار شده است.

#### ۴-۱-۲- جستجوی محلی

لین و همکاران [۱۷] یک روش جستجوی محلی جدید با نام طرح بهبود چندگانه افراد<sup>۵</sup> (MIE) را برای مسئله زمان‌بندی کار کارگاهی توسعه دادند. این روش شامل چهار ساختار جستجوی همسایگی (swap, insertion, inversion and or-opt) است که مکرراً برای بهبود یک جواب مورد نظر به‌کار برده می‌شوند. جستجوی محلی به‌کار رفته در این پژوهش براساس MIE است. جستجوی محلی به‌شرح زیر است. برای هر یک از عملگرها احتمالی بین صفر و یک در نظر می‌گیریم به طوری که مجموع احتمال‌ها یک باشد سپس یک عدد تصادفی تولید می‌شود که اگر کوچک‌تر از احتمال عملگر اول بود این عملگر انتخاب می‌شود و اگر از احتمال عملگر اول بزرگ‌تر و از مجموع احتمال عملگر اول و دوم کوچک‌تر بود عملگر دوم انتخاب می‌شود و به همین ترتیب برای عملگر سوم و چهارم پس از اعمال یکی از چهار عملگر روی ذره مورد نظر، اگر ذره جدید بهتر از ذره قبلی باشد جایگزین می‌شود. در غیر این صورت ذره جدید احتمالاً پذیرفته می‌شود. احتمال پذیرش برای یک جواب بد به‌وسیله مکانیزمی مشابه الگوریتم

شبه‌سازی تبرید تعیین می‌شود. اگر ذره جدید پذیرفته شود، جستجو با آن ادامه می‌یابد و در غیر این صورت با ذره قبلی سپس ساختار همسایگی دیگری انتخاب می‌شود و این رویه تا رسیدن به شرط توقف تکرار می‌شود. ساختارهای همسایگی به‌کار رفته در جستجوی محلی به‌صورت زیر هستند:

۱. ساختار همسایگی اول (swap): در این ساختار موقعیت دو قطعه که به‌صورت تصادفی انتخاب شده‌اند با یکدیگر جابه‌جا می‌شود. چگونگی اجرای این همسایگی با مثالی در شکل (۳) نشان داده شده است. در این مثال دو سفارش از یک محصول با سه قطعه وجود دارد. موقعیت‌های ۱ و ۶ به تصادف برای نقاط جابه‌جایی انتخاب شده‌اند. عدد قطعات در این دو موقعیت را با هم جابه‌جا می‌کنیم.
۲. ساختار همسایگی دوم (insertion): در این ساختار موقعیت یک قطعه در توالی مجدداً تعیین می‌شود. دو موقعیت a, b به تصادف انتخاب می‌شوند و قطعه موجود در موقعیت a به موقعیت جدید b انتقال می‌یابد. در مثال (شکل ۴) موقعیت‌های ۱ و ۴ به تصادف انتخاب شده‌اند.
۳. ساختار همسایگی سوم (inversion): این ساختار، قطعات بین دو موقعیت که به‌صورت تصادفی انتخاب شده‌اند را معکوس می‌کند. در مثال شکل (۵) موقعیت‌های ۲ و ۵ به



۱	۳	۲	۱	۳	۲	توالی فعلی
۱	۱	۳	۲	۳	۲	توالی جدید

شکل ۶- مثال عددی برای عملگر or-opt

۳.۱	۲	۱	۱	۲	۳	ذره اول
۲	۱	۳	۳	۱	۲	ذره دوم
۳	۲	۱	۱	۳	۲	ذره جدید

شکل ۷- مثال عددی برای عملگر تقاطع تک نقطه‌ای

۱	۳	۲	۲	۳	۱	ذره اول
۱	۲	۳	۱	۲	۳	ذره دوم
۱	۳	۲	۲	۳	۱	ذره جدید

شکل ۸- مثال عددی برای عملگر تقاطع دو نقطه‌ای

موقعیت جدید هر ذره را به وجود می‌آورند. عملگرهای

به کار رفته به شرح زیر هستند:

۱. عملگر تقاطع تک نقطه‌ای: در اجرای این عملگر یک موقعیت تصادفی به عنوان نقطه جداسازی انتخاب می‌شود. قطعات در موقعیت‌های قبل از این نقطه به‌طور کامل از ذره اول به ذره جدید می‌رسد و از چپ به راست از ذره دوم حذف می‌شوند. موقعیت‌های خالی در جواب جدید، با قطعات باقی مانده در ذره دوم با همان ترتیب پر می‌شوند. مثال (شکل ۷) چگونگی کار عملگر تک نقطه‌ای را نشان می‌دهد. فرض می‌کنیم نقطه جداسازی ۴ است.

۲. عملگر تقاطع دو نقطه‌ای: به دو موقعیت تصادفی به عنوان نقاط جداسازی نیاز داریم. قطعات بینین نقاط به‌طور کامل از ذره اول به ذره جدید می‌رسد، قطعات باقی مانده در جواب جدید، به ترتیبی که در ذره دوم وجود دارد جایگزین می‌شوند. مثال (شکل ۸) رویه این عملگر را نشان می‌دهد. اعداد ۳ و ۵ به تصادف برای نقاط جداسازی تولید شده‌اند.

نحوه پیاده سازی الگوریتم ازدحام ذرات با جستجوی محلی که MPSO نامیده می‌شود در شکل (۹) آورده شده است.

تصادف انتخاب شده‌اند.

۴. ساختار همسایگی چهارم (or-opt): مشابه ساختار همسایگی دوم است با این تفاوت که قطعه در موقعیت  $a+1, a$  به دو موقعیت جدید انتقال می‌یابند. در مثال شکل (۶) موقعیت‌های ۲ و ۵ به تصادف انتخاب شده‌اند. قطعات در موقعیت ۲ و ۳ به موقعیت ۵ و ۶ منتقل می‌شوند.

#### ۴-۱-۳- فاز تجدید جمعیت

این فاز دومین قسمت الگوریتم است که در آن ذرات از بهترین موقعیت گذشته خود و بهترین موقعیت کل ذرات تأثیر می‌گیرند. از جمعیت اولیه تولید شده، جوابی که بهترین مقدار تابع هدف را داشته باشد، به عنوان بهترین موقعیت کل انتخاب می‌شود. در ابتدا عملگر تقاطع تک نقطه‌ای با ترکیب موقعیت فعلی هر ذره و بهترین موقعیت آن در گذشته یک جواب جدید تولید می‌کند، همچنین با ترکیب موقعیت فعلی هر ذره و بهترین موقعیت کل ذرات یک جواب جدید دیگر تولید می‌کند سپس به وسیله عملگر تقاطع دو نقطه‌ای این دو ذره جدید با هم ترکیب شده و

### MPSO Algorithm

```
Set MPSO parameters (popsize ,pMIE , betta , Tf , prob) ;
Generate initial population
  While computational_time < n do
    Do "local search"
    Do the "population updating phase"
  End while
```

شکل ۹. شبه کد پیاده‌سازی الگوریتم MPSO

## ۲-۴- الگوریتم انجماد تدریجی

روش شبیه‌سازی تبرید<sup>۷</sup> که توانایی خروج از بهینه محلی را دارد در بسیاری از مواقع الگوریتم انجماد تدریجی نیز نامیده می‌شود، از پدیده انجماد فیزیکی مواد الگو گرفته شده است. این الگوریتم شبیه به پدیده آنیلینگ، از یک پارامتر دما برای کنترل الگوریتم در طول بهبود جواب مسئله استفاده می‌کند. شانس جابه‌جایی یک جواب خوب با یک جواب بدتر در دماهای بالا، خروج الگوریتم از بهینه محلی را تضمین می‌کند و از طرف دیگر، کاهش احتمال پذیرش جواب بدتر با کاهش دما، موجب تضمین همگرایی SA است. در ابتدا یک جواب اولیه ایجاد و مقدار تابع هدف آن محاسبه می‌شود. سپس در هر دما، یک تعداد تکرار مشخص انجام می‌شود. برای هر جواب جاری، به وسیله عملگر insertion یک جواب جدید ( $x_c'$ ) پیدا شده و مقدار تابع هدف آن  $F(x_c')$  محاسبه می‌شود. در صورتی که جواب جدید مقدار تابع هدف جاری را بهبود دهد یا برابر آن باشد، به‌عنوان جواب جاری پذیرفته می‌شود. در غیر این صورت عدد تصادفی  $P$  تولید شده و با تابع احتمال  $\exp(-\Delta F/T_c)$  مقایسه می‌شود که  $\Delta F = F(x_c') - F(x_c)$ . در صورتی که  $P$  کمتر از تابع احتمال باشد، جواب همسایه به‌عنوان جواب جاری پذیرفته می‌شود. در انتها بعد از اتمام تعداد تکرارها در یک دمای ثابت، کاهش دما اتفاق خواهد افتاد. شرط توقف، رسیدن دمای الگوریتم به یک دمای انتهایی (دمای انجماد) است.

## ۵- ارزیابی عملکرد

در این بخش عملکرد مدل و الگوریتم‌ها مورد ارزیابی قرار می‌گیرد. برای این منظور آزمایش‌هایی برای مقایسه جواب‌های حاصل از مدل و الگوریتم‌ها در اندازه کوچک و بزرگ طراحی می‌شود. در آزمایش‌ها سایز کوچک پاسخ الگوریتم‌ها با جواب بهینه حاصل از مدل و در سایز بزرگ عملکرد دو الگوریتم با هم مقایسه می‌شود. برای ارزیابی مدل ریاضی پیشنهادی، از نرم‌افزار بهینه‌سازی GAMS 23.5 استفاده می‌شود. الگوریتم‌های فرا ابتکاری در محیط نرم‌افزار MATLAB7.12.0(R2011a) و به‌وسیله یک کامپیوتر شخصی با قدرت پردازش 2.2 GHz و RAM 3GB و Intel Core2 Duo اجرا می‌شوند.

برای از بین بردن تأثیر اندازه مسئله از یک معیار عملکردی معمول به نام انحراف درصدی نسبی استفاده می‌شود. بهترین پاسخ را  $Min_{sol}$  و جواب حاصل از هریک از الگوریتم‌ها را  $A Ig_{sol}$  نامیده و معیار عملکرد<sup>۸</sup> RPD به شکل رابطه (۲۶) محاسبه می‌شود:

$$RPD = \frac{A Ig_{sol} - Min_{sol}}{Min_{sol}} \times 100 \quad (26)$$

زمان‌های پردازش، مونتاژ و آماده‌سازی از بازه یکنواخت ۱ تا ۱۰۰ تولید می‌شوند. با توجه به اینکه تعیین درست پارامترها نقشی بسزا در عملکرد مطلوب یک الگوریتم دارد، در ابتدا، بهترین مقدار برای پارامترهای تأثیرگذار بر این الگوریتم‌ها تعیین می‌شود.

## ۵-۱- تنظیم پارامتر<sup>۹</sup>

برای تعیین سطح مناسب پارامترها از رویکردهای آماری

جدول ۴- پارامترها و سطوح مختلف آنها برای الگوریتم MPSO

prob	Tf	beta	P <sub>MIE</sub>	popsize	Level
[۰,۱ ۰,۱ ۰,۴ ۰,۴]	۰/۱	۰/۵۵	۰/۱	۵۰	۱
[۰,۲ ۰,۲ ۰,۳ ۰,۳]	۰/۲	۰/۷۵	۰/۳	۱۰۰	۲
[۰,۴ ۰,۴ ۰,۱ ۰,۱]	۰/۳	۰/۹	۰/۵	۱۵۰	۳

تعداد آزمایش‌ها مورد نیاز برای روش آزمایش فاکتوریلی کامل (۳۵) است ولی برای روش تاگوچی آرایه ارتوگونال I<sub>۲۷</sub> که تنها دارای ۲۷ آزمایش است مورد استفاده قرار می‌گیرد. برای اجرای هر یک از آزمایش‌ها سه مثال مختلف از ترکیب تعداد سفارش، قطعه و ماشین در نظر گرفته شده است. نتایج حاصل به وسیله نرم‌افزار minitab16 ارزیابی می‌شود. همان‌طور که از شکل (۱۰) مشخص است، سطوح مطلوب برای فاکتورها بدین ترتیب است:  
 $\text{Popsiz} = 50, \text{pMIE} = 0/1, \text{beta} = 0/55, \text{Tf} = 0/2,$   
 $\text{prob} = [0/1 \ 0/1 \ 0/4 \ 0/4]$

#### ۵-۱-۲- الگوریتم SA

سطوح مختلف فاکتورهای کنترل الگوریتم SA در جدول (۵) آورده شده است. الگوریتم شبیه‌سازی تبرید سه پارامتر دارد و برای هر یک سه سطح مختلف در نظر گرفته شده است. تعداد آزمایش‌های مورد نیاز برای روش آزمایش فاکتوریلی کامل (۳۳) و آرایه ارتوگونال متناسب با آن L<sub>۹</sub> است. با توجه به شکل (۱۱) مقادیر پارامترها به شرح زیر است:  
 $\text{Temperature} = 150, \text{alpha} = 0/7, \text{it} = 25$

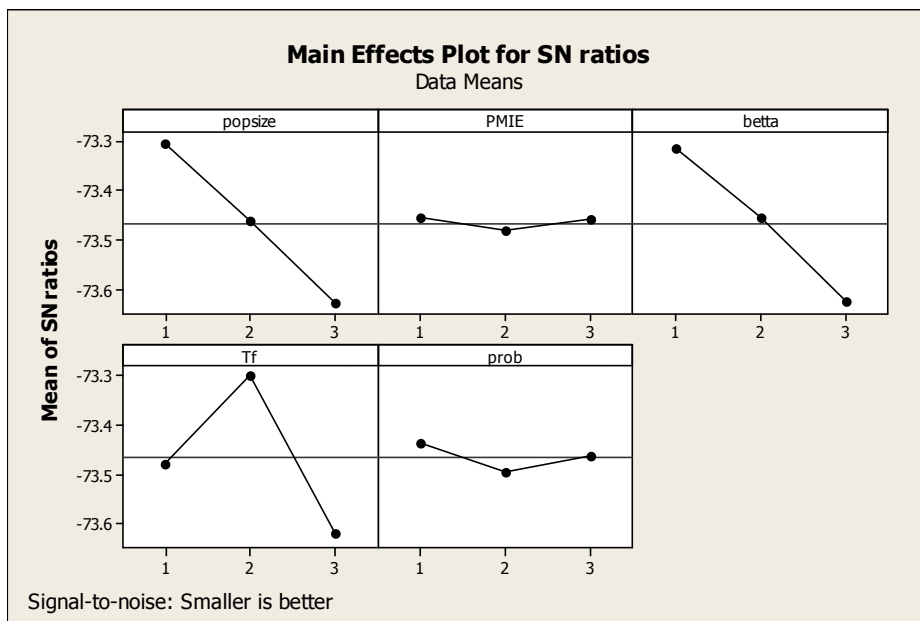
#### ۵-۲- ارزیابی عملکرد مدل و الگوریتم‌ها در ابعاد کوچک

در این بخش ابتدا به ارزیابی عملکرد مدل ریاضی معرفی شده در مقیاس کوچک می‌پردازیم. ۲۷ مسئله از ترکیب تعداد متفاوت سفارش و قطعه و ماشین‌های ساخت و مونتاژ و پاسخ بهینه آنها به وسیله نرم‌افزار GAMS23.5 به دست می‌آید. برای تعیین روند افزایش زمان با افزایش تعداد کار و قطعه، زمان رسیدن به جواب را ثبت می‌کنیم. به این ترتیب می‌توان اندازه مسائل قابل حل توسط مدل را ارزیابی کرد. بیشترین

متفاوتی بر مبنای طراحی آزمایش‌ها<sup>۱۰</sup> استفاده می‌شود که هر یک بسته به نوع آزمایش ممکن است مفید واقع شود. کامل‌ترین رویکرد که به طور مکرر مورد استفاده است، روش آزمایش فاکتوریلی کامل<sup>۱۱</sup> است. اما کارایی این روش با افزایش تعداد پارامترها کاهش می‌یابد. برای کاهش تعداد آزمایش‌های مورد نیاز، روش آزمایش فاکتوریل جزئی<sup>۱۲</sup> (FFE) توسعه یافته است [۱۸]. این روش تنها بخشی از کل ترکیبات ممکن پارامترها و برخی از تداخل‌های آنها را در نظر می‌گیرد که منجر به کاهش تعداد آزمایش‌ها می‌شود. تاگوچی [۱۹] یک گروه از ماتریس‌های FFE را توسعه داد که تعداد آزمایش‌ها را با حفظ اطلاعات کافی کاهش می‌دهد. در روش تاگوچی آرایه‌های ارتوگونال<sup>۱۳</sup> (OA) برای بررسی تعداد زیادی از پارامترها با تعداد کمی از آزمایش‌ها مورد استفاده قرار می‌گیرند. تاگوچی داده‌های تکراری را به ارزش دیگری که اندازه واریانس است تبدیل می‌کند. این تبدیل نسبت سیگنال به نویز<sup>۱۴</sup> (S/N) است. در اینجا سیگنال ارزش مطلوب (متغیر پاسخ) و نویز ارزش نامطلوب (انحراف استاندارد) را مشخص می‌کند. بنابراین نسبت S/N مقدار واریانس موجود در متغیرهای پاسخ را نشان می‌دهد. هدف ماکزیمم کردن نسبت S/N است. در این پژوهش روش تاگوچی که یک رویکرد مؤثر طراحی آزمایش است، برای تنظیم پارامتر الگوریتم‌های فرا ابتکاری پیشنهادی به کار برده می‌شود.

#### ۵-۱-۱- تنظیم پارامترهای الگوریتم MPSO

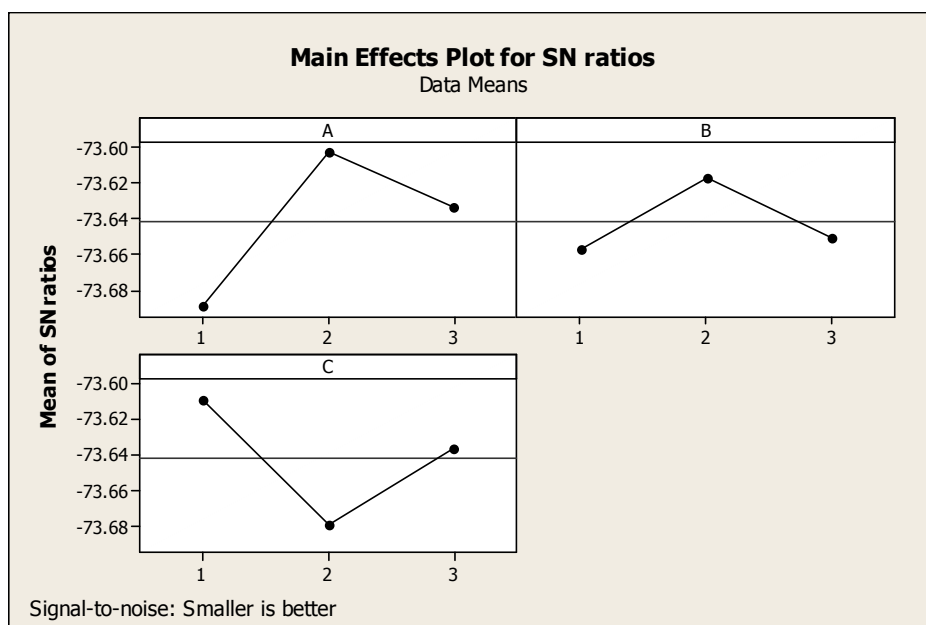
سطوح مختلف فاکتورهای قابل کنترل الگوریتم MPSO در جدول (۴) آورده شده است. الگوریتم ازدحام ذرات پنج پارامتر دارد و برای هر یک سه سطح مختلف در نظر گرفته شده است.



شکل ۱۰- نمودار میانگین نسبت S/N برای پارامترهای الگوریتم MPSO

جدول ۵- پارامترها و سطوح مختلف آنها برای الگوریتم SA

It(C)	Alpha(B)	Temperature(A)	Level
۲۵	۰,۵	۱۰۰	۱
۵۰	۰,۷	۱۵۰	۲
۷۵	۰,۹	۲۰۰	۳



شکل ۱۱- نمودار میانگین نسبت S/N برای پارامترهای الگوریتم SA

جدول ۶- نتایج حاصل از حل مسئله در سایز کوچک توسط مدل و الگوریتم‌ها

n	g	m	L	Cmax	الگوریتم‌ها				
					مدل	Gap	MPSO	SA	Time
۳	۲	۲	۲	۶۸۶	۱<	۰/۰۲۳	۶۸۶	۶۸۶	
۳	۲	۳	۲	۵۷۹	۱<	۰/۰۵۳	۵۷۹	۵۷۹	
۳	۲	۴	۲	۶۷۵	۱<	۰/۰۹۲	۶۷۵	۶۷۵	
۳	۳	۲	۲	۷۱۱	۵	۰/۱	۷۱۱	۷۱۱	
۳	۳	۳	۲	۸۵۰	۱<	۰/۱	۸۵۰	۸۵۰	
۳	۳	۴	۲	۹۴۸	۱<	۰/۰۹۹	۹۴۸	۹۴۸	
۳	۴	۲	۲	۷۹۹	۱۰۰۰	۰/۴۹۵	۸۰۴	۸۰۴	
۳	۴	۳	۲	۱۰۴۸	۱۰۰۰	۰/۴۰۴	۱۰۴۸	۱۱۳۱	
۳	۴	۴	۲	۱۱۹۰	۱۰۰۰	۰/۴۰۳	۱۲۴۴	۱۲۸۰	
۴	۲	۲	۲	۴۸۵	۱<	۰/۰۹۵	۴۸۵	۴۹۱	
۴	۲	۳	۲	۶۵۸	۱<	۰/۰۸۲	۶۵۸	۶۵۸	
۴	۲	۴	۲	۱۰۳۳	۱<	۰/۰۹۵	۱۰۳۳	۱۰۳۳	
۴	۳	۲	۲	۱۱۵۳	۱۰۰۰	۰/۱	۱۱۵۳	۱۱۵۳	
۴	۳	۳	۲	۱۰۸۸	۷۸۰	۰/۱	۱۰۸۸	۱۱۳۶	
۴	۳	۴	۲	۱۱۹۹	۱۰۰۰	۰/۳۴۱	۱۱۹۹	۱۲۰۵	
۴	۴	۲	۲	۹۴۳	۱۰۰۰	۰/۴۱۸	۹۵۷	۹۹۳	
۴	۴	۳	۲	۱۲۸۴	۱۰۰۰	۰/۵۲۳	۱۳۱۰	۱۳۲۰	
۴	۴	۴	۲	۱۴۴۹	۱۰۰۰	۰/۵۷۲	۱۴۴۹	۱۶۲۷	
۴	۲	۲	۳	۶۶۰	۱<	۰/۰۹۵	۶۶۰	۶۶۰	
۴	۲	۳	۳	۷۶۹	۱<	۰/۰۹۹	۷۶۹	۷۶۹	
۴	۲	۴	۳	۸۵۴	۱<	۰/۰۹۹	۸۵۴	۸۵۴	
۴	۳	۲	۳	۸۴۰	۵۴۰	۰/۱	۸۹۲	۸۵۴	
۴	۳	۳	۳	۹۳۷	۳۱۶	۰/۱	۹۳۷	۱۰۰۲	
۴	۳	۴	۳	۱۱۶۲	۳۰۰	۰/۱	۱۱۶۲	۱۲۱۴	
۴	۴	۲	۳	۱۲۳۱	۱۰۰۱	۰/۵۹۹	۱۲۳۱	۱۳۲۲	
۴	۴	۳	۳	۱۲۸۰	۱۰۰۰	۰/۵۹۹	۱۲۸۰	۱۳۰۰	
۴	۴	۴	۳	۱۳۶۲	۱۰۰۰	۰/۵۳۳	۱۳۶۲	۱۴۱۹	

### ۵-۳- عملکرد الگوریتم‌های فرا ابتکاری در سایز بزرگ

در این بخش عملکرد الگوریتم‌ها در مقیاس بزرگ مورد ارزیابی قرار می‌گیرد. با ترکیب مقادیر زیر ۷۲ مسئله برای این آزمایش در نظر گرفته شده است.

$n \in \{10, 30, 50, 70\}$ ,  $g \in \{5, 7, 9\}$ ,  $m \in \{3, 4, 5\}$ ,  $l \in \{4, 6\}$   
 نتایج آزمایش‌ها در جدول (۷) آورده شده است و حاکی از آن است که الگوریتم MPSO در سایز بزرگ، عملکردی بهتر از الگوریتم SA دارد. شکل (۱۲) حساسیت الگوریتم‌ها نسبت به تعداد کارها را نشان می‌دهد. الگوریتم MPSO کمترین

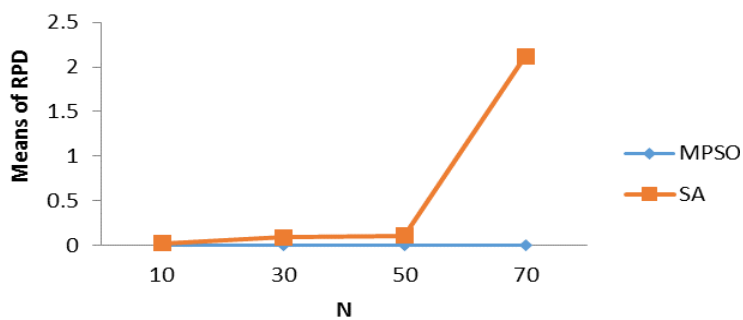
زمان مجاز برای محاسبات نرم‌افزار بهینه‌سازی ۱۰۰۰ ثانیه است که در این محدوده زمانی، نرم‌افزار قادر به حل بهینه مسائل با حداکثر چهار کار، چهار قطعه، چهار ماشین ساخت و چهار ماشین مونتاژ است. سپس همین مسائل را با الگوریتم‌های فرا ابتکاری حل کرده و نتایج حاصل از آنها را با جواب بهینه حاصل از حل مدل مقایسه می‌کنیم. نتایج حاصل در جدول (۶) ارائه شده است. جواب الگوریتم MPSO نسبت به الگوریتم SA در سایز کوچک، به جواب بهینه حاصل از حل دقیق، نزدیک‌تر است.

جدول ۷. مقایسه عملکرد الگوریتم‌ها در ساینز بزرگ با معیار RPD

n	G	m	l	MPSO	SA
۱۰	۵	۳	۴	۰	۰/۰۱۱
۱۰	۵	۳	۶	۰	۰/۰۱۳
۱۰	۵	۴	۴	۰	۰/۰۱۸
۱۰	۵	۴	۶	۰	۰/۰۲۴
۱۰	۵	۵	۴	۰	۰/۰۷۹
۱۰	۵	۵	۶	۰	۰/۱۶۴
۱۰	۷	۳	۴	۰	۰/۰۶۱
۱۰	۷	۳	۶	۰/۰۰۰۰۴	۰
۱۰	۷	۴	۴	۰	۰/۰۱۳
۱۰	۷	۴	۶	۰	۰/۰۱
۱۰	۷	۵	۴	۰	۰/۰۰۳
۱۰	۷	۵	۶	۰	۰/۰۰۳
۱۰	۹	۳	۴	۰/۰۰۰۲	۰
۱۰	۹	۳	۶	۰	۰/۰۰۷
۱۰	۹	۴	۴	۰	۰/۰۰۷
۱۰	۹	۴	۶	۰	۰/۰۲۳
۱۰	۹	۵	۴	۰	۰/۰۱۲
۱۰	۹	۵	۶	۰	۰/۰۰۵
۳۰	۵	۳	۴	۰	۰/۰۰۴
۳۰	۵	۳	۶	۰	۰/۱۹۷
۳۰	۵	۴	۴	۰	۰/۰۹۶
۳۰	۵	۴	۶	۰	۰/۰۱۶
۳۰	۵	۵	۴	۰	۰/۲۰۳
۳۰	۵	۵	۶	۰	۰/۲۰۲
۳۰	۷	۳	۴	۰	۰/۱۳۴
۳۰	۷	۳	۶	۰	۰/۰۰۲
۳۰	۷	۴	۴	۰	۰/۰۲۳
۳۰	۷	۴	۶	۰	۰/۱۰۴
۳۰	۷	۵	۴	۰	۰/۰۴۲
۳۰	۷	۵	۶	۰	۰/۲۲۰
۳۰	۹	۳	۴	۰	۰/۰۰۲
۳۰	۹	۳	۶	۰	۰/۰۸۷
۳۰	۹	۴	۴	۰	۰/۰۲۵
۳۰	۹	۴	۶	۰	۰/۱۷۶
۳۰	۹	۵	۴	۰	۰/۰۰۵
۳۰	۹	۵	۶	۰	۰/۰۰۳
۵۰	۵	۳	۴	۰	۰/۰۰۳
۵۰	۵	۳	۶	۰	۰/۲۱۸
۵۰	۵	۴	۴	۰	۰/۱۲۶
۵۰	۵	۴	۶	۰	۰/۰۲۶
۵۰	۵	۵	۴	۰	۰/۲۷۳
۵۰	۵	۵	۶	۰	۰/۲۰۸

ادامه جدول ۷

n	G	m	l	MPSO	SA
۵۰	۷	۳	۴	۰	۰/۱۷۹
۵۰	۷	۳	۶	۰	۰/۰۰۳
۵۰	۷	۴	۴	۰	۰/۰۳۱
۵۰	۷	۴	۶	۰	۰/۱۲۰
۵۰	۷	۵	۴	۰	۰/۰۴۱
۵۰	۷	۵	۶	۰	۰/۲۰۳
۵۰	۹	۳	۴	۰	۰/۰۰۰۴
۵۰	۹	۳	۶	۰	۰/۱۰۴
۵۰	۹	۴	۴	۰	۰/۰۲۵
۵۰	۹	۴	۶	۰	۰/۲۵۴
۵۰	۹	۵	۴	۰	۰/۰۶۴
۵۰	۹	۵	۶	۰	۰/۰۱۵
۷۰	۵	۳	۴	۰	۰/۰۰۰۲
۷۰	۵	۳	۶	۰	۰/۲۵۲
۷۰	۵	۴	۴	۰	۰/۱۴۱
۷۰	۵	۴	۶	۰	۰/۰۲۵
۷۰	۵	۵	۴	۰	۰/۲۸۲
۷۰	۵	۵	۶	۰	۰/۲۵۳
۷۰	۷	۳	۴	۰	۰/۱۹۴
۷۰	۷	۳	۶	۰	۰/۰۰۰۲
۷۰	۷	۴	۴	۰	۰/۰۳۵
۷۰	۷	۴	۶	۰	۰/۱۶۲
۷۰	۷	۵	۴	۰	۰/۰۵۲
۷۰	۷	۵	۶	۰	۰/۲۲۶
۷۰	۹	۳	۴	۰	۰/۰۰۰۱
۷۰	۹	۳	۶	۰	۰/۱۱۱
۷۰	۹	۴	۴	۰	۰/۰۲۵
۷۰	۹	۴	۶	۰	۰/۲۴۶
۷۰	۹	۵	۴	۰	۰/۰۷۰
۷۰	۹	۵	۶	۰	۰/۰۴۲
میانگین				۰	۰/۰۴۱



شکل ۱۲- تعیین حساسیت هریک از الگوریتم‌های فرا ابتکاری نسبت به تعداد کارها

بودن زمان مونتاژ روی این ماشین‌ها احتمال متناظر ماندن محصول برای خالی شدن ماشین مورد نظر با وجود خالی بودن ماشین دیگر وجود دارد. هنگامی که ماشین‌های ساخت شروع به پردازش قطعات می‌کنند و یا در هنگام تغییر آیتم قطعات، عملیات و زمان‌های آماده‌سازی غیروابسته به توالی مورد نیاز است. یک مدل برنامه‌ریزی خطی عدد صحیح مختلط و دو الگوریتم فرا ابتکاری برای حل مسئله پیشنهاد شد. سپس آزمایش‌هایی برای ارزیابی مدل و الگوریتم‌های پیشنهادی ارائه شد. نتایج نشان‌دهنده برتری الگوریتم MPSO نسبت به SA در مسائل با ابعاد کوچک و بزرگ بود. بررسی مسئله در سایر محیط‌های تولیدی، افزودن محدودیت منابع انسانی و در دسترس بودن اپراتور، حل مسئله با پارامترهای غیرقطعی و توسعه مدل چند هدفه بخشی از مسیرهای ادامه پژوهش هستند.

حساسیت را نسبت به تعداد کارها دارد و در واقع می‌توان گفت افزایش یا کاهش تعداد کارها عملکرد این الگوریتم را تحت تأثیر قرار نمی‌دهد ولی عملکرد SA با افزایش تعداد کارها بدتر می‌شود.

## ۶- نتیجه‌گیری و پیشنهادات

در این پژوهش، یک مسئله زمان‌بندی جریان کارگاهی مونتاژ با زمان‌های راه‌اندازی غیروابسته به توالی بررسی شد. تعدادی محصول یکسان سفارش داده شده است که هر محصول از مونتاژ چندین قطعه مختلف ساخته می‌شود. ابتدا قطعات در یک ایستگاه جریان کارگاهی متشکل از چندین ماشین مختلف تولید می‌شوند و سپس در یک ایستگاه با چندین ماشین ناهمسان به محصول نهایی مونتاژ می‌شوند. هر محصول تنها روی یکی از ماشین‌های مونتاژ پردازش می‌شود. به دلیل متفاوت

## واژه‌نامه

- |  |  |   |
|--|--|---|
| 1. non-identical                               | 6. single-point crossover              | 11. full factorial experiment             |
| 2. makespan                                    | 7. simulated annealing                 | 12. fractional factorial experiment (FFE) |
| 3. sequence-based                              | 8. relative percentage deviation (RPD) | 13. orthogonal                            |
| 4. particle swarm optimization                 | 9. tuning parameter                    | 14. signal-to-noise ratio                 |
| 5. multiple-type individual enhancement scheme | 10. design of experiments (DOE)        |   |

## مراجع

1. Yokoyama, M., and Santos, D. L., "Three-stage Flow-shop Scheduling with Assembly Operations to Minimize the Weighted Sum of Product Completion Times", *European Journal of Operational Research*, Vol. 161, pp. 754-770, 2005.
2. Kan, A. H. G. R., *Machine Scheduling Problems*, Martinus Nijhoff, Hague, 1976.
3. Allahverdi, A., and Al-Anzi, F. S., "The Two-stage Assembly Scheduling Problem to Minimize Total Completion Time with Setup Times", *Computers and Operations Research*, Vol. 36, pp. 2740-2747, 2009.
4. Lee, C. Y., Cheng, T. C. E., and Lin, B. M. T., "Minimizing the Makespan in the 3-machine Assembly-type Flowshop Scheduling Problem", *Management Science*, Vol. 39, pp. 616-625, 1993.
5. Potts, C. N., Sevast Janov, S. V., Strusevich, V. A., Van Wassenhove, L. N., and Zwaneveld, C. M., "The Two-Stage Assembly Scheduling Problem: Complexity and Approximation", *Operations Research*, Vol. 43, pp. 346-355, 1995.
6. Hariri, A. M. A., and Potts, C. N., "A Branch and Bound Algorithm for the Two-stage Assembly Scheduling Problem", *European Journal of Operational Research*, Vol. 103, pp. 547-556, 1997.
7. Cheng, T. C. E., and Wang, G., "Scheduling the Fabrication and Assembly of Components in a Two-machine Flowshop", *IIE Transactions*, Vol. 31, pp. 135-143, 1999.
8. Sung, C. S., and John, J., "Makespan Minimization for a 2-stage Assembly Scheduling Problem Subject to Component Available Time Constraint", *International Journal of Production*, Vol. 119, pp. 392-401, 2009.
9. Yokoyama, M., "Hybrid Flow-shop Scheduling with



- Assembly Operations”, *International Journal of Production Economics*, Vol. 73, pp. 103-116, 2001.
10. Rios-Mercado, R. Z., and Bard, J. F., “The Flowshop Scheduling Polyhedron with Setup Times”, *Journal of Combinatorial Optimization*, Vol. 7, pp. 291-318, 2003.
  11. Kennedy, J., and Eberhart, R., “Particle Swarm Optimization”, *Proceedings of IEEE International Conference on Neural Networks*, Vol. 4, pp. 1942-1948, 1995.
  12. Ge, H. W., Sun, L., Liang, Y. C., and Qian, F., “An Effective PSO and AIS-based Hybrid Intelligent Algorithm for Job-shop Scheduling”, *IEEE Transactions on Systems, Man and Cybernetics, Part A*, Vol. 38, pp. 358-368, 2008.
  13. Kuo, I. H., Horng, S. J., Kao, T. W., Lin, T. L., Lee, C. L., and Terano, T., “An Efficient Flow-shop Scheduling Algorithm Based on a Hybrid Particle Swarm Optimization Model”, *Expert Systems with Applications*, Vol. 36, pp. 7027-7032, 2009.
  14. Liao, C. J., Tseng, C. T., and Luarn, P., “A Discrete Version of Particle Swarm Optimization for Flowshop Scheduling Problems”, *Computers and Operations Research*, Vol. 34, pp. 3099-3111, 2007.
  15. Xia, W., and Wu, Z., “A Hybrid Particle Swarm Optimization Approach for the Jobshop Scheduling Problem”, *The International Journal of Advanced Manufacturing Technology*, Vol. 29, pp. 360-366, 2005.
  16. Gen, M., Tsujimura, Y., and Kubota, E., “Solving Job Shop Scheduling Problem using Genetic Algorithms”, *Proceedings of the 16th International Conference on Computer and Industrial Engineering*, Ashikaga, Japan, pp. 576-579, 1994.
  17. Lin, T. L., Hong, S. J., Kao, T. W., Chen, Y. H., Run, R. S., Chen, R. J., Lai, J. L., and Kuo, I. H., “An Efficient Job-shop Scheduling Algorithm Based on Particle Swarm Optimization”, *Expert Systems with Applications*, Vol. 37, pp. 2629-2636, 2010.
  18. Cochran, W. G., and Cox, G. M., *Experimental Designs*, 2nd ed., Wiley, USA, 1992.
  19. Ross, R. J., *Taguchi Techniques for Quality Engineering*, McGraw-Hill, USA, 1989.