

مدل ریاضی و الگوریتم جدید برای مسئله زمان‌بندی اقتصادی انباشته در محیط جریان کارگاهی انعطاف‌پذیر

حسن زهلی، بهمن نادری* و محمد محمدی

گروه مهندسی صنایع، دانشکده فنی مهندسی، دانشگاه خوارزمی، تهران

(دریافت مقاله: ۱۳۹۴/۰۸/۲۵ - دریافت نسخه نهایی: ۱۳۹۵/۱۱/۰۳)

چکیده- در این مقاله مسئله تعیین اندازه انباشته و زمان‌بندی چند محصول در محیط جریان کارگاهی انعطاف‌پذیر با ماشین‌های موازی یکسان بررسی می‌شود. مراحل تولید به صورت سری بوده و بافر محدود بین مراحل متوالی در نظر گرفته شده است. تابع هدف مسئله، کمینه سازی مجموع هزینه‌های راه‌اندازی و نگهداری در واحد زمان است. مدل ریاضی موجود در ادبیات مسئله دارای پیچیدگی اندازه و پیچیدگی محاسباتی زیادی است. در این مقاله، یک مدل ریاضی خطی مختلط عدد صحیح جدید برای مسئله ارائه می‌شود. همچنین یک الگوریتم فرا ابتکاری کارآمد برای حل ابعاد بزرگ مسئله پیشنهاد می‌شود. نتایج حاصل از انجام آزمایشات عددی، بیانگر برتری قابل توجه مدل و الگوریتم پیشنهادی این مقاله در مقایسه با مدل و الگوریتم‌های موجود در ادبیات مسئله است.

واژه‌های کلیدی: تعیین اندازه انباشته، زمان‌بندی، جریان کارگاهی مختلط، مدل ریاضی، الگوریتم فرا ابتکاری.

New Mathematical Model and Algorithm for Economic Lot Scheduling Problem in Flexible Flow Shop

H. Zohali*, B. Naderi, and M. Mohammadi

Department of Industrial Engineering, Faculty of Engineering, Kharazmi University, Tehran, Iran

Abstract: This paper addresses the lot sizing and scheduling problem for a number of products in flexible flow shop with identical parallel machines. The production stages are in series, while separated by finite intermediate buffers. The objective is to minimize the sum of setup and inventory holding costs per unit of time. The available mathematical model of this problem in the literature suffers from huge complexity in terms of size and computation. In this paper, a new mixed integer linear program is developed for delay with the huge dimensions of the problem. Also, a new meta heuristic algorithm is developed for the problem. The results of the numerical experiments represent a significant advantage of the proposed model and algorithm compared with the available models and algorithms in the literature.

Keywords: lot sizing; scheduling; hybrid flow shop; mathematical model; meta heuristic.

*: مسئول مکاتبات، پست الکترونیکی: bahman.naderi@khu.ac.ir

n	تعداد محصولات	$s_{i,j}$	زمان راه‌اندازی محصول i در مرحله j
m	تعداد مراکز کاری	$h_{i,j}$	هزینه نگهداری محصول i در مرحله j
$\gamma m - 1$	تعداد کل مراحل اعم از مراکز کاری و بافرها	PH	طول افق برنامه‌ریزی
f_{max}	حداکثر تعداد سیکل‌های ممکن	M	یک عدد حقیقی مثبت بزرگ
k, i	شمارنده محصولات	$\sigma_{r,j}$	بردار توالی تولید روی $M_{r,j}$
j	شمارنده مراحل	$b_{i,j}$	زمان آغاز پردازش محصول i در مرحله j
u	شمارنده سیکل‌ها	T	سیکل تولید
M_j	تعداد ماشین‌های موازی در مرحله j	F	تعداد سیکل‌های تولید در طول افق برنامه‌ریزی
$M_{r,j}$	ماشین r ام در مرحله j	Q_i	اندازه انباشته محصول i ($Q_i = d_i.T$)
d_i	نرخ تقاضای محصول i	$t_{i,j}$	مدت زمان پردازش محصول i در مرحله j ($t_{i,j} = d_{i,j}.T / p_{i,j}$)
$d_{i,j}$	نرخ تقاضای محصول i در مرحله j ، اگر مرحله j مرکز کاری باشد $d_{i,j} = d_i$ و اگر مرحله j بافر باشد $d_{i,j} = 0$ خواهد بود.	$y_{i,r,j}$	متغیر تخصیص محصولات به ماشین‌ها، اگر محصول i توسط $M_{r,j}$ پردازش شود $y_{i,r,j} = 1$ و در غیر این صورت $y_{i,r,j} = 0$ شود.
$p_{i,j}$	نرخ تولید محصول i در مرحله j ، اگر مرحله j مرکز کاری باشد $p_{i,j} = p_{i,j}$ و اگر مرحله j بافر باشد $p_{i,j} = R$ خواهد بود. (R یک عدد حقیقی غیر صفر است).	$x_{i,j,k}$	متغیر تعیین توالی محصولات، اگر در مرحله j محصول i پس از محصول k پردازش شود $x_{i,j,k} = 1$ و در غیر این صورت $x_{i,j,k} = 0$ است.
sc_i	مجموع هزینه‌های راه‌اندازی محصول i در تمام مراحل	v_u	متغیر تعیین تعداد سیکل‌های تولید؛ اگر تعداد سیکل‌های تولید برابر u باشد $v_u = 1$ و در غیر این صورت $v_u = 0$ است.

۱- مقدمه

مسئله تعیین اندازه انباشته^۱ یکی از مهم‌ترین مسائل تصمیم‌گیری در بین مسائل برنامه‌ریزی تولید است که پرداختن به آن در هر سیستم تولیدی، موجب کاهش هزینه‌های تولید و در نتیجه بهبود عملکرد سیستم خواهد شد. هزینه‌های اصلی و مشترک تولید محصول در اکثر سیستم‌های تولیدی عبارتند از: هزینه ساخت^۲، هزینه راه‌اندازی^۳، هزینه نگهداری^۴ و هزینه کمبود^۵ که مجموع آنها هزینه کل را تشکیل می‌دهد. در شرایطی

که بحث افزایش قیمت، تخفیف و مواردی از این قبیل مطرح نباشد، هزینه ساخت مقدار ثابتی بوده و در تحلیل مدل‌ها می‌توان از آن صرف‌نظر کرد.

با فرض ثابت بودن هزینه ساخت، مجموع هزینه‌های راه‌اندازی، نگهداری و کمبود برابر هزینه کل تولید خواهد بود. کاهش دادن هر یک از این هزینه‌ها به تنهایی، لزوماً موجب کاهش هزینه کل نخواهد شد. به‌عنوان مثال فرض کنید مقدار مشخص تقاضا از یک محصول به سیستم برسد و سیستم

می‌توان به صنعت کاغذسازی، صنایع پلاستیک و نیز صنایع نساجی اشاره کرد (مرکادو و دیاز [۲]). در چنین صنایعی، محصولات دارای تقاضای ثابت و مشخصی بوده و در سیستم تولید پیوسته و نیز حجم نسبتاً بالایی تولید می‌شوند.

در دنیای واقعی تعداد کمی از محصولات هستند که تنها با یک مرحله پردازش قابل تولید باشند. به همین دلیل بررسی مسئله در سیستم‌های تولید چند مرحله‌ای موجب کاربردی‌تر شدن آن خواهد شد. یکی از سیستم‌های چند مرحله‌ای رایج در صنعت، محیط جریان کارگاهی^۴ (FS) است. در این سیستم تعداد مشخصی ماشین (مرحله) که هر کدام عملیات مختلفی را انجام می‌دهد، به صورت سری قرار گرفته‌اند. پردازش هر محصول از ماشین اول آغاز شده و پس از طی تمام ماشین‌ها (با همان ترتیب موجود ماشین‌ها) توسط ماشین آخر خاتمه می‌یابد. به منظور افزایش توان عملیاتی این سیستم، معمولاً به جای یک ماشین در هر مرحله، از چند ماشین به صورت موازی استفاده می‌شود که در این حالت آن را جریان کارگاهی مختلط^۵ (HFS) می‌نامند. به عبارت دیگر، سیستم HFS حالت کلی‌تر FS است.

در حالت کلاسیک FS و HFS ظرفیت بافرهای میانی^۱ به صورت نامحدود در نظر گرفته می‌شود. به عبارت دیگر، هنگامی که پردازش یک محصول در یکی از مراحل به پایان برسد، آن محصول مرحله جاری را ترک می‌کند و اگر ماشین موجود در مرحله بعدی توسط محصول دیگری اشغال نشده باشد، این محصول می‌تواند بلافاصله پردازش خود را در مرحله بعد آغاز کند. در غیر این صورت، محصول بین این دو مرحله منتظر خواهد بود تا ماشین موجود در مرحله بعدی آزاد شود. این شرایط تنها زمانی قابل اجرا است که محصولات از نظر فیزیکی بسیار کوچک بوده و فضای کمی را اشغال کنند اما در حالتی که محصولات از لحاظ فیزیکی دارای حجم بزرگی باشند، امکان‌پذیر نخواهد بود. به همین دلیل منطقی است که ظرفیت محدود برای بافرهای میانی در نظر گرفته شود.

تولیدی موظف به پاسخگویی نسبت به این تقاضای طی مدت زمان مشخص باشد. در حالتی که کل تقاضا در یک انباشته تولید شود، هزینه راه‌اندازی به حداقل مقدار ممکن خواهد رسید اما به دلیل وابستگی هزینه نگهداری به سطح موجودی‌ها، این هزینه افزایش خواهد یافت، در نتیجه ممکن است میزان افزایش هزینه نگهداری بیشتر از میزان کاهش هزینه راه‌اندازی باشد که در این صورت هزینه کل افزایش خواهد یافت. همچنین اگر تقاضا به صورت انباشته‌های کوچک تولید شود، هزینه نگهداری کاهش می‌یابد اما با توجه به بیشتر تعداد دفعات راه‌اندازی تولید، هزینه راه‌اندازی افزایش خواهد یافت و امکان دارد که میزان کاهش هزینه نگهداری کمتر از میزان افزایش هزینه راه‌اندازی باشد. این گونه اثر متقابل بین هزینه کمبود با سایر هزینه‌ها نیز برقرار است. مسئله تعیین اندازه انباشته با ایجاد نوعی توازن بین هزینه‌ها مشخص می‌کند که هر محصول در چه زمانی و به چه مقداری تولید شود تا هزینه کل کمینه شود.

تعیین توالی^۶ و زمان‌بندی^۷ نیز یکی از تصمیمات مرتبط با مسئله تعیین اندازه انباشته است که تخصیص بهینه منابع تولیدی به عملیات مورد نیاز تولید محصولات را مشخص می‌کند. در نظر گرفتن تصمیمات توالی و زمان‌بندی در مسئله تعیین اندازه انباشته موجب پیچیده‌تر شدن آن می‌شود اما در عوض باعث نزدیک‌تر شدن شرایط مسئله به دنیای واقعی خواهد شد. به همین دلیل، غالباً تصمیمات زمان‌بندی و تعیین اندازه انباشته به طور هم‌زمان مورد بررسی قرار می‌گیرد. کریمی و همکاران [۱] دسته‌بندی جامعی از انواع مسائل تعیین اندازه انباشته را ارائه کرده‌اند.

مسئله زمان‌بندی اقتصادی انباشته^۸ (ELSP)، یکی از مدل‌های پایه‌ای و مهم در بین مسائل تعیین اندازه انباشته است. در این مسئله باید چند محصول توسط یک ماشین تولید شود. تابع هدف مسئله برابر با مجموع هزینه‌های راه‌اندازی و نگهداری محصولات است که باید حداقل شود. این مسئله در صنایع مختلف دنیای واقعی قابل مشاهده است که از میان آنها

مربوط به یافتن جواب شدنی برای آن است. به طور کلی، نشدنی بودن این مسئله می‌تواند ناشی از محدودیت ظرفیت ماشین و یا تداخل زمان‌بندی محصولات باشد [۷]. برای تولید چند محصول توسط یک ماشین، ظرفیت ماشین باید به اندازه‌ای باشد که با در نظر گرفتن زمان راه‌اندازی و زمان پردازش هر محصول، امکان تولید تمام محصولات در هر سیکل یا دوره^{۱۶} (منظور از دوره، فاصله زمانی بین دو بار آغاز تولید متوالی هر محصول است) فراهم شود. علاوه بر این، نباید بین زمان‌بندی تولید محصولات تداخل ایجاد شود، به عبارت دیگر زمان آغاز تولید هر محصول نباید زودتر از زمان تکمیل پردازش محصول قبل از آن در توالی باشد زیرا در این صورت تقاضای محصول قبلی با کمبود مواجه خواهد شد.

بررسی شرط کافی بودن ظرفیت ماشین با توجه به مشخص بودن زمان‌های راه‌اندازی و نرخ تولید و نرخ تقاضای محصولات امر بسیار ساده‌ای خواهد بود. در حقیقت یافتن یک زمان‌بندی شدنی برای مسئله موجب پیچیدگی زیاد آن شده است. به منظور کاهش پیچیدگی مسئله و اطمینان از یافتن زمان‌بندی شدنی محصولات از رویکردهای متعددی در مدل‌سازی مسئله استفاده شده اما توجه محققان معمولاً بر سه رویکرد سیکل مشترک^{۱۷}، دوره پایه^{۱۸} و انباشته‌های متغیر با زمان^{۱۹} است [۸]. این رویکردها شرط تعویض صفر^{۲۰} (ZLR) را برای محصولات در نظر می‌گیرند، بدین معنا که تولید هر محصول زمانی می‌تواند آغاز شود که سطح موجودی آن به صفر رسیده باشد. در مرجع [۹] مرور کاملی بر این مسئله و مطالعات انجام شده پیرامون آن ارائه شده است.

مسئله ELSP در محیط FS توسط محققان کمتری مورد بررسی قرار گرفته است که از میان آنها می‌توان به دابسون و بانو [۱۰]، نجداوی و کلیندورفر [۱۱]، نجداوی [۱۲] اشاره کرد. تمام محققانی که تا سال ۱۹۹۹ این مسئله را مورد بررسی قرار داده‌اند، تنها به تعیین متغیر اندازه انباشته پرداخته‌اند و اثر تغییر توالی محصولات روی هزینه کل را ناچیز تلقی می‌کردند.

در این مقاله مدل ELSP در محیط HFS با ماشین‌های موازی یکسان بررسی می‌شود. همچنین فرض افق برنامه‌ریزی محدود و بافرهای میانی محدود برای مسئله در نظر گرفته شده است. علاوه بر این، تمامی پارامترهای مسئله اعم از تقاضای محصولات، نرخ تولید ماشین‌ها و هزینه‌های راه‌اندازی و نگهداری به صورت قطعی در نظر گرفته شده است. این مسئله برای اولین بار توسط اکرمی و همکاران [۳] مورد بررسی قرار گرفته است. در این تحقیق ابتدا یک مدل غیرخطی مختلط عدد صحیح^{۱۱} (MINLP) برای مسئله توسعه داده می‌شود و سپس حالت خطی شده همین مدل ریاضی ارائه می‌شود. در نهایت، نتایج حاصل از مقایسه عملکرد مدل‌های ریاضی ارائه شده در این مقاله و مدل پیشنهادی توسط اکرمی و همکاران [۳] ارائه می‌شود.

اولین و ساده‌ترین مسئله تعیین اندازه انباشته، یک قرن پیش توسط تافت [۴] تحت عنوان مقدار تولید اقتصادی^{۱۳} (EPQ) معرفی می‌شود. در این مسئله، یک محصول با تقاضای قطعی و ایستا، توسط یک ماشین با نرخ تولید محدود و شرط مجاز نبودن کمبود باید تولید شود. همچنین تابع هدف این مسئله برابر با مجموع هزینه‌های راه‌اندازی و نگهداری است.

مدل ELSP در واقع بسط مسئله EPQ برای چند محصول است. راجرز [۵] را می‌توان از جمله نخستین محققانی دانست که به معرفی و مطالعه این مسئله پرداخته است. راجرز برای حل این مسئله از رابطه EPQ برای هر محصول به طور مستقل استفاده کرد و نتیجه گرفت که جواب‌های حاصل از این روش معمولاً نشدنی است. محققان بعدی، رویکرد راجرز را اصطلاحاً جواب‌های مستقل^{۱۴} نامیده و از آن بیشتر برای یافتن حد پایین تابع هدف مسئله اصلی استفاده کردند.

اگرچه مسئله EPQ به سادگی قابل حل است اما مدل ELSP بسیار پیچیده‌تر از آن است. از بین محققان بسیاری که به مطالعه مدل ELSP پرداخته‌اند، مرجع [۶] ثابت کرد که این مسئله از نظر پیچیدگی، در گروه مسائل غیر چندجمله‌ای^{۱۵} سخت (NP-hard) قرار دارد. در حقیقت پیچیدگی زیاد این مسئله،

مثال‌های مشترکی را توسط مدل پیشنهادی خود و دو الگوریتم فرا ابتکاری طراحی شده حل کردند. نتایج حاصل از حل این مثال‌ها نشان داد که کیفیت جواب‌های حاصل از الگوریتم‌های فرا ابتکاری به مراتب بهتر از نتایج حل دقیق مدل ریاضی است. به همین دلیل، آنها استفاده از الگوریتم‌های فرا ابتکاری را برای حل این مسئله توصیه کردند.

در مقاله دیگری جنابی و همکاران [۱۵] نیز مسئله معرفی شده توسط انیچه و همکاران [۱۳] را در محیط HFS با ماشین‌های موازی غیریکسان و افق برنامه‌ریزی محدود مورد بررسی قرار داده و یک مدل MINLP برای مسئله معرفی کردند. علاوه بر این، آنها از الگوریتم‌های شبیه‌سازی تبرید و ژنتیک ترکیبی برای حل مسئله استفاده کردند. جنابی و همکاران [۱۵] نیز پس از انجام آزمایشات عددی، به نتیجه‌گیری مشابهی با مراجع [۳ و ۱۴] رسیده و استفاده از الگوریتم‌های فرا ابتکاری به جای حل دقیق مدل ریاضی را توصیه کردند.

همان‌طور که بیان شد، محققانی که به بررسی مسئله ELSF در محیط HFS پرداخته‌اند، استفاده از الگوریتم‌های فرا ابتکاری را به جای حل مدل ریاضی مسئله را توصیه می‌کنند. دلیل اصلی چنین نتیجه‌گیری را می‌توان در غیرخطی بودن مدل ریاضی ارائه شده توسط آنها دانست. در حقیقت تمامی مدل‌های ریاضی توسعه داده شده در ادبیات مسئله به صورت غیرخطی بوده و حل کردن این مدل‌ها نیازمند زمان و بار محاسباتی بسیار زیادی است.

۲- مدل‌سازی مسئله

در مسئله مورد بررسی، مجموعه‌ای از محصولات باید در یک سیستم تولید چند مرحله‌ای تولید شوند. در این سیستم، مراحل به صورت سری قرار گرفته و هر مرحله (مرکز کاری) دارای حداقل یک ماشین است. در مرحله‌ای که دارای بیش از یک ماشین هستند، از ماشین‌های یکسان به صورت موازی استفاده شده است. پردازش هر محصول از مرحله اول آغاز شده و پس از طی تمام مراحل (با همان ترتیب موجود

به عبارت دیگر، هیچ یک از این محققان تصمیمات تعیین توالی و اندازه انباشته را به طور هم‌زمان بررسی نکرده است. برای مثال نجداوی و کلیندورفر [۱۱] از قاعده معروف کوتاه‌ترین زمان پردازش^{۲۱} برای توالی محصولات استفاده کرده‌اند و متغیری که توالی محصولات را به عنوان خروجی مسئله در نظر بگیرد، در تحقیق آنها وجود ندارد.

برای اولین بار انیچه و همکاران [۱۳] تصمیمات تعیین توالی و اندازه انباشته را به طور هم‌زمان در این مسئله بررسی کردند. آنها با استفاده از رویکرد سیکل مشترک، یک مدل ریاضی مختلط عدد صحیح برای مسئله ارائه دادند. متأسفانه به دلیل غیرخطی بودن مدل پیشنهادی، هیچ تضمینی بر بهینگی جواب‌های حاصل وجود نداشت. همچنین آنها به منظور حل مسئله از دو الگوریتم فرا ابتکاری معروف جستجوی ممنوعه^{۲۲} (TS) و شبیه‌سازی تبرید^{۲۳} (SA) استفاده کردند. نتایج حاصل از مطالعات آنها نشان داد که تغییر توالی محصولات، اثر قابل توجهی بر کاهش هزینه کل خواهد داشت.

ترابی و همکاران [۱۴] مسئله معرفی شده توسط انیچه و همکاران [۱۳] را برای یک تأمین کننده در زنجیره تأمین بررسی کردند. در این مسئله، تأمین کننده طی یک افق برنامه‌ریزی محدود و در محیط HFS به تولید چند محصول می‌پردازد. ترابی و همکاران [۱۴] یک مدل MINLP برای مسئله خود توسعه دادند. همچنین آنها از الگوریتم ژنتیک^{۲۴} (GA) نیز برای حل مسئله استفاده کردند. براساس مقایسه بین نتایج حاصل از حل دقیق مدل ریاضی و حل مسئله توسط الگوریتم ژنتیک، آنها نتیجه گرفتند که استفاده از الگوریتم فرا ابتکاری برای حل این مسئله بسیار مناسب‌تر است.

اکرمی و همکاران [۳] مسئله معرفی شده توسط انیچه و همکاران [۱۳] را با فرض افق برنامه‌ریزی محدود، در محیط HFS مطالعه کردند. همچنین فرض بافرهای میانی محدود برای این مسئله در نظر گرفته شده بود. آنها یک مدل MINLP برای مسئله خود پیشنهاد داده و نیز الگوریتم‌های جستجوی ممنوعه و ژنتیک را برای حل مسئله به کار گرفتند. اکرمی و همکاران

در یکی از مراحل به پایان برسد، آن محصول مرحله جاری را ترک می‌کند و اگر ماشین موجود در مرحله بعدی توسط محصول دیگری اشغال نشده باشد، این محصول می‌تواند بلافاصله پردازش خود را در مرحله بعد آغاز کند. در غیر این صورت، محصول در مرحله جاری منتظر خواهد بود تا ماشین موجود در مرحله بعدی آزاد شود. در این شرایط محصولات اجازه توقف بین مراحل را ندارند. بنابراین، زمان آغاز پردازش محصول در هر مرحله، معادل با زمان ترک کردن مرحله قبل است.

در این بخش، دو مدل ریاضی جدید برای مسئله مورد بررسی ارائه می‌شود. ابتدا یک مدل MINLP برای مسئله توسعه داده می‌شود. سپس دومین مدل ریاضی (M2) که حالت خطی شده مدل اول (M1) است معرفی خواهد شد.

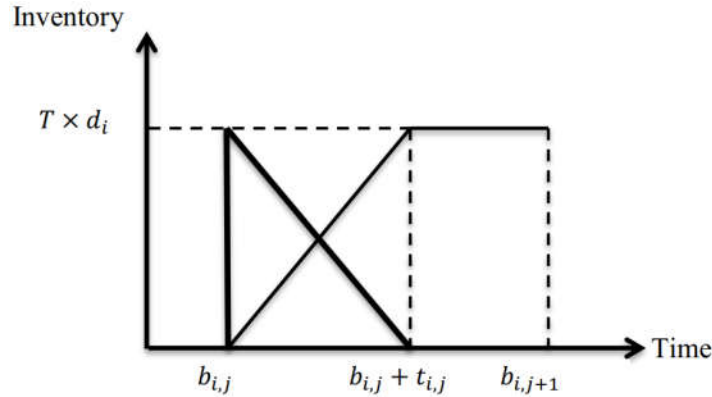
در عمل پس از اتمام پردازش یک محصول در هر مرحله، ارزش کیفی محصول افزایش خواهد شد به همین دلیل واحد هزینه نگهداری هر محصول در هر مرحله نسبت به مرحله قبل کوچک‌تر نخواهد بود. به عبارت دیگر:

$$h_{i,j-1} \leq h_{i,j}, \quad h_{i,j} < h_{i,2m-1}; \\ i = 1, \dots, n, \quad j = 2, \dots, 2m-1$$

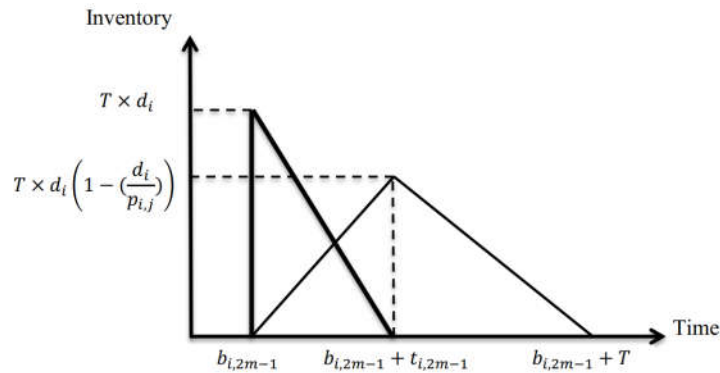
۲-۱- مدل ریاضی غیرخطی (M1)

تابع هدف مسئله برابر میانگین هزینه‌های راه‌اندازی و نگهداری در واحد زمان است. با توجه به اینکه ارسال محصولات نهایی برای مشتری به‌طور پیوسته انجام می‌شود، دو نوع موجودی برای هر محصول وجود خواهد داشت. نوع اول موجودی در جریان ساخت^{۲۷} (WIP) که در مراحل ماقبل آخر است و نوع دوم، موجودی محصول نهایی که در مرحله آخر وجود دارد. در نتیجه هزینه کل برابر مجموع هزینه‌های راه‌اندازی و نگهداری موجودی در جریان ساخت و نگهداری محصول نهایی است که این سه هزینه به ترتیب با TC_{setup} ، TC_{WIP} و TC_{final} در مدل‌سازی نشان داده شده‌اند. اولین قسمت از تابع هدف، میانگین هزینه راه‌اندازی در واحد زمان است که به صورت زیر محاسبه می‌شود:

مراحل) توسط مرحله آخر خاتمه می‌یابد. هر ماشین در هر زمان، قابلیت پردازش حداکثر یک محصول را دارد. تقاضا و تولید هر محصول دارای نرخ قطعی و ثابتی است. کمبود موجودی مجاز نبوده و تقاضای تمامی محصولات باید به‌طور کامل پاسخ داده شود. هر محصول در هر یک از مراحل تنها توسط یک ماشین پردازش می‌شود. پردازش هر محصول در هر مرحله زمانی می‌تواند آغاز شود که راه‌اندازی مربوط به آن انجام شده باشد. همچنین محصول نهایی به‌طور پیوسته برای مشتری ارسال می‌شود. زمان و هزینه راه‌اندازی هر محصول به‌صورت قطعی، ثابت و مستقل از توالی در نظر گرفته می‌شود. قطع کار^{۲۵} مجاز نیست؛ یعنی در هر مرحله پردازش انباشته هر محصول از لحظه آغاز پردازش تا لحظه تکمیل پردازش بدون وقفه انجام می‌گیرد. تقسیم انباشته^{۲۶} مجاز نیست؛ یعنی انباشته هر محصول به‌صورت یک جا و پس از اتمام پردازش کامل آن در مرحله جاری می‌تواند به مرحله بعد انتقال یابد. همچنین اندازه انباشته هر محصول در تمام مراحل تولید یکسان است. بافر محدود بین مراحل متوالی تولید در نظر گرفته شده است. از رویکرد سیکل مشترک برای زمان‌بندی محصولات استفاده می‌شود، به عبارت دیگر تمام محصولات دارای سیکل تولید برابر هستند. علاوه بر این، افق برنامه‌ریزی به‌صورت پیوسته و محدود بوده و سیکل تولید باید به تعداد صحیح تکرار شود تا افق برنامه‌ریزی به اتمام برسد. هدف از این مسئله، یافتن یک زمان‌بندی تولید است به‌طوری که هزینه کل کمینه شود. در این مقاله به منظور فرمول‌بندی مسئله، هر بافر به‌عنوان یک مرحله مجزا با زمان پردازش صفر در نظر گرفته شده است. به عبارت دیگر، اگر مسئله دارای m مرحله با زمان پردازش غیر صفر (مرکز کاری) باشد، تعداد تعداد مراحل با زمان پردازش صفر (بافر) برابر $m-1$ خواهد بود، در نتیجه تعداد کل مراحل برابر $2m-1$ است. همچنین ظرفیت هر بافر برابر تعداد ماشین‌های موازی در مرحله مربوط به آن است. در چنین سیستمی هنگامی که پردازش یک محصول



شکل ۱- نمودار موجودی بر حسب زمان محصول i در مرحله j ($j < 2m-1$)



شکل ۲- نمودار موجودی بر حسب زمان محصول i در مرحله آخر

اثر دادن واحد هزینه نگهداری، میزان متوسط هزینه نگهداری در واحد زمان برای تمام محصولات در تمام مراحل ماقبل آخر به صورت زیر محاسبه می شود:

$$TC_{WIP} = \sum_{i=1}^n \left(\sum_{j=2}^{2m-1} h_{i,j-1} \times d_i \left(\frac{T \times d_{i,j}}{2 \times p_{i,j}} \right) + \sum_{j=1}^{2m-1} h_{i,j} \times d_i \left(b_{i,j+1} - b_{i,j} - \frac{T \times d_{i,j}}{2 p_{i,j}} \right) \right) \quad (2)$$

همچنین میانگین موجودی و هزینه نگهداری موجودی محصول i در مرحله آخر به صورت زیر محاسبه می شود (اکرمی و همکاران [۲]):

$$AI_{final} = \frac{1}{T} \left(d_i \times T \left(\frac{t_{i,2m-1}}{2} \right) + d_i \times T \left(1 - \left(\frac{d_i}{p_{i,2m-1}} \right) \right) \times \left(\frac{T}{2} \right) \right)$$

$$TC_{setup} = \frac{\sum_{i=1}^n sc_i}{T}$$

هزینه های نگهداری از روی میانگین موجودی در زمان محاسبه می شوند. نمودار موجودی بر حسب زمان محصول در جریان ساخت و محصول نهایی به ترتیب در شکل های (۱) و (۲) نشان داده شده است. در این شکل ها، خطوط ضخیم بیانگر موجودی ورودی و خطوط باریک بیانگر موجودی خروجی است.

میانگین موجودی و هزینه نگهداری موجودی محصول i در مرحله j (مرحله j جزو مراحل ماقبل آخر است) به صورت زیر محاسبه می شود [۳]:

$$AI_{WIP} = \frac{1}{T} \left(d_i \times T \left(\frac{t_{i,j}}{2} \right) + d_i \times T \left(\left(\frac{t_{i,j}}{2} \right) + d_i \times T \left(b_{i,j+1} - b_{i,j} - \frac{t_{i,j}}{2} \right) \right) \right) \quad (1)$$

با جای گذاری عبارت $t_{i,j} = T \times d_{i,j} / p_{i,j}$ در عبارت فوق و

$$b_{i,\gamma m-1} + T \times \frac{d_i}{P_{i,\gamma m-1}} \leq T \quad ; i=1, \dots, n \quad (11)$$

$$b_{i,j} \geq s_{i,j} \quad ; i=1, \dots, n, j=1, \dots, \gamma m-1 \quad (12)$$

$$F \times T = PH \quad (13)$$

$$y_{i,r,j}, x_{i,j,k} \in \{0,1\} \quad ; \forall i,k,r,j \quad (14)$$

$$b_{i,j}, T \geq 0 \quad ; \forall i,j \quad (15)$$

$$F \geq 1 \text{ and integer} \quad (16)$$

مجموعه محدودیت (۵) الزام می‌کند که هر محصول در هر مرحله فقط توسط یک ماشین پردازش شود. براساس مجموعه محدودیت (۶) هر محصول زمانی می‌تواند یک مرحله را ترک کند که پردازش آن در این مرحله تکمیل شده باشد. مجموعه محدودیت‌های (۷) و (۸) به‌طور هم‌زمان تضمین می‌کند که در مراحل ماقبل آخر، پردازش هر محصول توسط هر ماشین زمانی می‌تواند آغاز شود که محصول قبلی آن در توالی، ماشین را ترک کرده باشد. همچنین مجموعه محدودیت‌های (۹) و (۱۰) به‌طور هم‌زمان تضمین می‌کند که در مرحله آخر، پردازش هر محصول توسط هر ماشین زمانی می‌تواند آغاز شود که محصول قبلی آن در توالی، ماشین را ترک کرده باشد. براساس محدودیت‌های (۱۱) زمان تکمیل پردازش هر محصول در مرحله نهایی نباید بیشتر از طول سیکل باشد. مجموعه محدودیت (۱۲) به لزوم انجام راه‌اندازی‌ها در هر مرحله اشاره می‌کند. محدودیت (۱۳) تضمین می‌کند که تعداد سیکل‌ها در افق برنامه‌ریزی برابر یک عدد صحیح باشد. متغیرهای تصمیم نیز توسط محدودیت‌های (۱۴)، (۱۵) و (۱۶) تعریف شده است.

۲-۲- مدل ریاضی خطی (M2)

غیرخطی بودن مدل ریاضی M1 مربوط به هزینه راه‌اندازی موجود در تابع هدف و نیز محدودیت (۱۳) در این مدل است. با استفاده از محدودیت (۱۳) به‌سادگی می‌توان غیرخطی بودن هزینه راه‌اندازی را برطرف کرد. تابع هدف حاصل پس از خطی کردن هزینه راه‌اندازی، در رابطه (۱۷) ارائه شده است:

با جای‌گذاری عبارت $t_{i,\gamma m-1} = T \times d_{i,\gamma m-1} / P_{i,\gamma m-1}$ عبارت فوق و اثر دادن واحد هزینه نگهداری، میزان متوسط هزینه نگهداری در واحد زمان برای تمام محصولات در مرحله آخر به‌صورت رابطه (۳) محاسبه می‌شود:

$$TC_{\text{final}} = \sum_{i=1}^n T \left(h_{i,\gamma m-1} \left(\frac{d_i^r}{\gamma \times P_{i,\gamma m-1}} \right) + h_{i,\gamma m-1} \times d_i \left(1 - \frac{d_i}{\gamma \times P_{i,\gamma m-1}} \right) \right) \quad (3)$$

توجه شود که در روابط (۲) و (۳) برای موجودی ورودی از یک مرحله، واحد هزینه نگهداری همان مرحله در نظر گرفته شده است. در نهایت تابع هدف مسئله به‌صورت زیر بیان می‌شود:

$$Z1 = \sum_{i=1}^n \frac{sc_i}{T} + \sum_{i=1}^n \sum_{j=1}^{\gamma m-1} h_{i,j-1} \times d_i \left(\frac{T \times d_{i,j}}{\gamma \times P_{i,j}} \right) + h_{i,j} \times d_i \left(b_{i,j+1} - b_{i,j} - \frac{T \times d_{i,j}}{\gamma \times P_{i,j}} \right) + \sum_{i=1}^n T \left(h_{i,\gamma m-1} \left(\frac{d_i^r}{\gamma \times P_{i,\gamma m-1}} \right) + h_{i,\gamma m-1} \times d_i \left(1 - \frac{d_i}{\gamma \times P_{i,\gamma m-1}} \right) \right) \quad (4)$$

اکنون محدودیت‌های مدل M1 به‌طور کامل ارائه می‌شود:

$$\sum_{r=1}^{m_j} y_{i,r,j} = 1 \quad ; i=1, \dots, n, j=1, \dots, \gamma m-1 \quad (5)$$

$$b_{i,j} + T \times \frac{d_{i,j}}{P_{i,j}} \leq b_{i,j+1} \quad ; i=1, \dots, n, j=1, \dots, \gamma m-2 \quad (6)$$

$$b_{i,j} \geq b_{k,j+1} + s_{i,j} - M \left(\gamma - y_{i,r,j} - y_{k,r,j} - x_{i,j,k} \right) \quad ; i=1, \dots, n, k > i, j=1, \dots, \gamma m-2, r=1, \dots, m_j \quad (7)$$

$$b_{k,j} \geq b_{i,j+1} + s_{k,j} - M \left(\gamma - y_{i,r,j} - y_{k,r,j} + x_{i,j,k} \right) \quad ; i=1, \dots, n, k > i, j=1, \dots, \gamma m-2, r=1, \dots, m_j \quad (8)$$

$$b_{i,j} \geq b_{k,j} + T \times \frac{d_{k,j}}{P_{k,j}} + s_{i,j} - M \left(\gamma - y_{i,r,j} - y_{k,r,j} - x_{i,j,k} \right) \quad ; i=1, \dots, n, k > i, j=\gamma m-1, r=1, \dots, m_j \quad (9)$$

$$b_{k,j} \geq b_{i,j} + T \times \frac{d_{i,j}}{P_{i,j}} + s_{k,j} - M \left(\gamma - y_{i,r,j} - y_{k,r,j} + x_{i,j,k} \right) \quad ; i=1, \dots, n, k > i, j=\gamma m-1, r=1, \dots, m_j \quad (10)$$

رابطه (۲۵) تعیین کرد. (عبارت [a] بیانگر بزرگ‌ترین عدد صحیح کوچک‌تر یا مساوی a است):

$$f_{\max} = \left\lceil \frac{PH}{T_{\min}} \right\rceil \quad (25)$$

۳- الگوریتم فرا ابتکاری

با توجه به پیچیدگی بالای مسئله، حل کردن مدل ریاضی برای ابعاد بزرگ مسئله نیازمند زمان محاسباتی زیادی است. به همین دلیل استفاده از الگوریتم‌های فرا ابتکاری برای حل ابعاد بزرگ‌تر مسئله مورد اهمیت است. در این بخش یک الگوریتم فرا ابتکاری کارآمد مبتنی بر الگوریتم جستجوی همسایگی متغیر کاهش شده^{۲۸} (RVNS) و بهینه‌سازی حشره میوه^{۲۹} (FOA) برای حل مسئله پیشنهاد می‌شود.

۳-۱- الگوریتم جستجوی همسایگی متغیر کاهش شده

الگوریتم RVNS حالتی ساده شده از الگوریتم جستجوی همسایگی متغیر (VNS) است. الگوریتم VNS یک الگوریتم مبتنی بر جستجوی محلی^{۳۰} است که برای اولین بار توسط ملانودویک و هانسن [۱۶] معرفی شده است. سادگی و کارآمدی این الگوریتم موجب شده تا در سال‌های اخیر توسط محققان متعددی برای حل انواع مسائل زمان‌بندی به‌کار گرفته شود. لی [۱۷] الگوریتم VNS را برای حل نوعی از مسئله زمان‌بندی در محیط FS استفاده می‌کند. مصلحی و خراسانیان [۱۸]، حاتمی و همکاران [۱۹] از جمله سایر محققانی هستند که از این الگوریتم برای حل مسائل زمان‌بندی استفاده کرده‌اند.

الگوریتم VNS از دو فاز اصلی تحت عنوان آشفته‌سازی و جستجوی محلی به همراه تعداد مشخصی ساختار همسایگی مرتب شده تشکیل شده است. ابتدا در فاز آشفته‌سازی یک جواب تصادفی از اولین ساختار همسایگی انتخاب خواهد یافت و توسط فاز جستجوی محلی بهبود داده می‌شود. اگر بهبود حاصل شد جواب جدید جایگزین جواب فعلی می‌شود و مجدداً از اولین ساختار همسایگی الگوریتم ادامه خواهد شد و

$$Z_{\gamma} = \frac{F}{PH} \sum_{i=1}^n sc_i + \sum_{i=1}^n \sum_{j=1}^{\gamma m - \gamma} h_{i,j-1} \times d_i \left(\frac{T \times d_{i,j}}{\gamma \times p_{i,j}} \right) + h_{i,j} \times d_i \left(b_{i,j+1} - b_{i,j} - \frac{T \times d_{i,j}}{\gamma \times p_{i,j}} \right) + \sum_{i=1}^n \Gamma \left(h_{i,\gamma m - \gamma} \left(\frac{d_i^{\gamma}}{\gamma \times p_{i,\gamma m - 1}} \right) + h_{i,\gamma m - 1} \times d_i \left(1 - \frac{d_i}{\gamma \times p_{i,\gamma m - 1}} \right) \right) \quad (17)$$

علاوه بر این، با استفاده از مجموعه محدودیت‌های (۱۸) تا (۲۱) به جای محدودیت (۱۳) مدل ریاضی M1 به یک مدل خطی تبدیل خواهد شد. همچنین به دلیل استفاده از متغیر تصمیم جدید در مدل ریاضی M2 محدودیت (۲۲) جایگزین محدودیت (۱۴) می‌شود:

$$\sum_{u=1}^{f_{\max}} v_u = 1 \quad (18)$$

$$\sum_{u=1}^{f_{\max}} u \times v_u = F \quad (19)$$

$$PH \times v_u \leq u \times T \quad ; u = 1, \dots, f_{\max} \quad (20)$$

$$u \times T \leq PH + (PH \times u - PH) \times (1 - v_u) \quad ; u = 1, \dots, f_{\max} \quad (21)$$

$$y_{i,r,j}, x_{i,j,k}, v_u \in \{0,1\} \quad ; \forall i, k, r, j, u \quad (22)$$

پارامتر f_{\max} بیانگر حداکثر تعداد سیکل‌های ممکن در افق برنامه‌ریزی است. به منظور محاسبه مقدار این پارامتر می‌توان از شرط لازم برای شدنی بودن مسئله استفاده کرد. رابطه (۲۳) بیانگر شرط لازم برای شدنی بودن مسئله است:

$$\sum_{i=1}^n \left(\frac{d_{i,j}}{p_{i,j}} + \frac{s_{i,j}}{T} \right) \leq m_j \quad ; j = 1, \dots, \gamma m - 1 \quad (23)$$

با استفاده از رابطه (۲۳) به سادگی می‌توان یک حد پایین برای طول سیکل به صورت زیر محاسبه کرد:

$$T_{\min} = \max_j \left(\frac{\sum_{i=1}^n s_{i,j}}{m_j - \sum_{i=1}^n \frac{d_{i,j}}{p_{i,j}}} \right) \quad (24)$$

با مشخص بودن حد پایین طول سیکل، به سادگی می‌توان حداکثر تعداد سیکل‌های ممکن در افق برنامه‌ریزی را طبق

```

Procedure : RVNS algorithm
Input : initial solution; number of neighbor structures ( $k_{max}$ )
Output : best solution
 $\pi$ :=initial solution
 $\pi^*=\pi$ 
While stopping criterion is not met do
  set=1
  While  $k < k_{max}$  do
     $\pi'$ :=random neighbor from  $NS_k(\pi)$ 
    If  $\pi'$  is better than  $\pi$ 
       $\pi := \pi'$ 
      Update  $\pi^*$ 
       $k := 1$ 
    Elseif
       $k := k + 1$ 
    Endif
  Endwhile
Endwhile

```

شکل ۳- شبه کد کلی RVNS

پس از نزدیک شدن به منبع غذایی احتمالی، از حس بینایی خود برای ارزیابی ارزش منبع غذایی استفاده می‌کند. در نهایت مگس‌های میوه به سمت بهترین منبع غذایی یافت شده حرکت کرده و همین رویه تا برقراری شرط توقف الگوریتم تکرار می‌شود.

۳-۳- الگوریتم پیشنهادی

در این قسمت یک الگوریتم RVNS ساده و بسیار کارآمد برای مسئله مورد بررسی ارائه خواهد شد. الگوریتم پیشنهادی از ایده الگوریتم FOA در تولید جواب جدید استفاده می‌کند. در ادامه به‌طور مفصل به این بحث پرداخته خواهد شد.

۳-۳-۱- نمایش جواب و تابع برازش

مسئله مورد بررسی در این مقاله از دو نوع متغیر تصمیم تشکیل شده است. نوع اول متغیرهای گسسته برای تخصیص و تعیین توالی محصولات روی ماشین‌ها است. نوع دوم متغیرهای پیوسته برای زمان‌بندی و تعیین اندازه انباشته محصولات است. توجه شود که با محاسبه متغیر طول دوره (T) می‌توان مقدار انباشته محصولات را محاسبه کرد.

متغیرهای گسسته موجب پیچیدگی زیاد مسئله شده است. به همین دلیل از الگوریتم فرا ابتکاری برای تعیین مقدار متغیرهای

در غیر این صورت یک جواب تصادفی از دومین ساختار همسایگی انتخاب شده و فاز جستجوی محلی روی آن اجرا خواهد شد.

الگوریتم RVNS حالتی خاص از VNS است که در آن فاز جستجوی محلی حذف شده است. در واقع برای مسائلی که بررسی تعداد زیادی جواب در فاز جستجوی محلی مستلزم صرف زمان محاسباتی بالایی باشد الگوریتم RVNS نسبت به VNS ارجحیت دارد. برای شناخت بیشتر این الگوریتم مرجع [۲۰] پیشنهاد می‌شود. در شکل (۳) شبه کد کلی الگوریتم RVNS نشان داده شده است.

۳-۲- الگوریتم مگس میوه

الگوریتم FOA یکی از الگوریتم‌های جدید است که توسط پان [۲۱] برای حل مسائل بهینه‌سازی معرفی شده است. این الگوریتم براساس رفتار جمعی نوع خاصی از حشرات به نام مگس میوه برای یافتن منبع غذا توسعه داده شده است. مگس‌های میوه دارای حس بویایی و بینایی بسیار قدرتمندتری نسبت به سایر حشرات هستند و همین خصوصیات در یافتن منبع غذایی به آنها کمک فراوانی می‌کند. در ابتدا هر مگس میوه با استفاده از حس بویایی می‌تواند مکان جغرافیایی منبع غذایی احتمالی را تشخیص داده و به سمت آن حرکت کند. مگس میوه

۳-۳-۳- شرح الگوریتم

همان‌طور که بیان شد برای یافتن تابع برازش هر جواب می‌بایست یک مدل MILP به صورت بهینه حل شود. واضح است که با افزایش ابعاد مسئله تعداد جواب‌های مورد بررسی نیز افزایش خواهد یافت و حل کردن یک مدل MILP به ازای هر جواب امر بسیار دشواری بوده و موجب صرف زمان محاسباتی بسیار زیادی خواهد شد. به عبارت دیگر ممکن است که الگوریتم فرصت کافی برای دستیابی به جواب‌های با کیفیت را نداشته باشد. در واقع این موضوع حساسیت بسیار زیادی در طراحی الگوریتم و کارایی آن دارد.

در همین بخش اشاره شد که الگوریتم RVNS برای مسائلی که بررسی تعداد زیاد جواب‌ها ممکن نباشد بسیار مفید است. با این وجود، همچنان مشکل جدی در انتخاب جواب‌های همسایه برای جواب فعلی الگوریتم وجود دارد. به عبارت دیگر در الگوریتم RVNS کلاسیک هر جواب جدید به صورت تصادفی توسط ساختار همسایگی تولید می‌شود و این امر مشابه با جستجوی کاملاً تصادفی در فضای جواب است. در این مقاله از رویکرد بسیار مفید الگوریتم FOA برای تشخیص جواب‌های احتمالی خوب استفاده می‌شود.

رویکرد جواب‌های مستقل در ادبیات مسئله توسط محققان متعددی برای محاسبه حد پایین تابع هدف مسئله به کار رفته است. طبق این رویکرد فرض می‌شود که هر محصول به‌طور مستقل از سایر محصولات و در زودترین زمان ممکن تولید شود. به عبارت دیگر تعداد ماشین‌ها در هر مرحله برابر تعداد محصولات و نیز زمان‌های راه‌اندازی برابر صفر باشند. در چنین شرایطی تنها متغیر تصمیم مسئله طول سیکل (T) خواهد بود. براساس اینچه و بکتور [۲۲] مقدار بهینه زمان سیکل به صورت زیر قابل محاسبه است:

گسسته استفاده می‌شود. پس از جای‌گذاری مقدار متغیرهای گسسته در مدل M2 و حل کردن آن به صورت بهینه متغیرهای پیوسته مسئله به دست خواهند آمد. در نتیجه هر جواب الگوریتم بیانگر چگونگی تخصیص و توالی محصولات روی ماشین‌ها است. در این مقاله همانند اکرمی و همکاران [۳] از بردار جایگشتی برای نمایش هر جواب الگوریتم استفاده شده است. این بردار جایگشتی حاوی n عضو بوده که بیانگر ترتیب پردازش محصولات در اولین مرحله است. همچنین از قاعده اولین ماشین موجود^{۳۱} (FAM) برای تعیین تخصیص و توالی محصولات در سایر مراحل استفاده می‌شود.

هر جواب در الگوریتم دارای یک مقدار عددی منحصر است که بیانگر میزان مطلوبیت آن نسبت به سایر جواب‌ها است. در مسئله مورد بررسی هر جواب با هزینه کل کمتر جواب بهتری محسوب می‌شود. در نتیجه تابع هدف مسئله به عنوان تابع برازش^{۳۲} الگوریتم در نظر گرفته شده است. در واقع میزان برازندگی هر جواب پس از جای‌گذاری متغیرهای گسسته (توالی و تخصیص) در مدل M2 و حل بهینه مدل حاصل به دست خواهد آمد.

۳-۳-۲- جواب اولیه

در الگوریتم RVNS با یک جواب اولیه، آغاز فرایند جستجو تنها به یک جواب اولیه نیاز دارد. در این مقاله از روشی که اکرمی و همکاران [۳] برای تولید جواب اولیه الگوریتم TS پیشنهادی خود معرفی کرده‌اند، استفاده خواهد شد. در این روش تمامی محصولات برحسب مدت زمان پردازش در مرحله اول به صورت صعودی مرتب می‌شوند. در بخش ۳ اشاره شد که زمان پردازش محصول i در مرحله j برابر $T \times d_{i,j} / p_{i,j}$ است. با توجه به این که مقدار متغیر T قبل از شروع الگوریتم مشخص نیست مقدار آن $T=1$ فرض شده و جواب اولیه به دست می‌آید.

$$t_{\min} = \sqrt{\frac{\sum_{i=1}^n s c_i}{\sum_{i=1}^n \sum_{j=1}^{r_m-1} d_i \left(\frac{d_{i,j}}{p_{i,j}} \right) (h_{i,j-1} + h_{i,j}) + \sum_{i=1}^n \left[\left(\frac{h_{i,r_m-1} \cdot d_i}{p_{i,r_m-1}} \right) + h_{i,r_m-1} \cdot d_i \left(1 - \frac{d_i}{p_{i,r_m-1}} \right) \right]}}$$

قبل از جای گذاری	1	2	3	4	5	6	7	8
------------------	---	---	---	---	---	---	---	---

پس از جای گذاری	1	2	4	5	6	3	7	8
-----------------	---	---	---	---	---	---	---	---

شکل ۴- ساختار همسایگی اول

قبل از جای گذاری	1	2	3	4	5	6	7	8
------------------	---	---	---	---	---	---	---	---

پس از جای گذاری	1	2	5	6	3	4	7	8
-----------------	---	---	---	---	---	---	---	---

شکل ۵- ساختار همسایگی دوم

قبل از جای گذاری	1	2	3	4	5	6	7	8
------------------	---	---	---	---	---	---	---	---

پس از جای گذاری	1	2	5	6	7	3	4	8
-----------------	---	---	---	---	---	---	---	---

شکل ۶- ساختار همسایگی سوم

بردار جواب جایگزین یک محصول از محصولات پس از خود می‌شوند. در NS_p هر دو محصول متوالی از بردار جواب جایگزین دو محصول متوالی از محصولات پس از خود می‌شوند.

گام‌های اجرایی الگوریتم پیشنهادی به صورت زیر است:

(۱) جواب اولیه تولید شده و به عنوان جواب فعلی و نیز بهترین جواب الگوریتم ثبت شود.

(۲) تمام جواب‌های ممکن از NS_1 را تولید شود. جواب‌های حاصل برحسب میزان تفاوت C_{max} آنها با t_{min} به صورت صعودی در یک لیست جدید مرتب شوند. مقدار تابع برازش برای جواب‌های ۱ تا A در این لیست محاسبه شود. هرگاه که اولین بهبود برای جواب فعلی حاصل شد، نیازی به محاسبه تابع برازش سایر جواب‌ها نیست و جواب جدید جایگزین جواب فعلی شود.

(۳) اگر جواب فعلی در گام ۲ بهبود پیدا کرد، گام ۲ مجدداً تکرار خواهد شد. در غیر این صورت گام ۴ اجرا می‌شود.

در واقع t_{min} کمترین مقدار برای طول سیکل بهینه مسئله است. با توجه به مدل ریاضی طول سیکل را می‌توان معادل کوچک‌ترین زمان تکمیل (C_{max}) تمام محصولات در دوره دانست. همچنین طول سیکل متغیر اصلی تابع هدف بوده بخش عمده از تابع هدف وابسته به همین متغیر است. در نتیجه هر جواب الگوریتم که طول سیکل آن کوچک‌تر (به t_{min} نزدیک‌تر) باشد به طور بالقوه یک جواب با کیفیت است.

در الگوریتم RVNS پیشنهادی سه ساختار همسایگی^{۳۳} مبتنی بر عملگر جای‌گذاری استفاده شده است. نحوه عملکرد اولین، دومین و سومین ساختار که به اختصار با NS_1 ، NS_2 و NS_3 نشان داده می‌شود به ترتیب در شکل (۴)، (۵) و (۶) آمده است. در NS_1 هر عضو از بردار جواب (هر محصول) با جایگزین اعضای بعد از خود می‌شود. به عنوان مثال در شکل (۴) محصول ۳ پس از محصول ۶ جای‌گذاری شده است. در NS_2 هر دو محصول متوالی از

```

Procedure : proposed RVNS
Input : initial solution
Output : best solution
 $\pi$  := initial solution
 $\pi^* := \pi$ 
While stopping criterion is not met do
  set  $k = 1$ 
  While  $k < 2$  do
    Generate all of the NS $k$  neighbors
    Calculation absolute gap between  $C_{max}$  and  $t_{min}$  for all neighbors
    Sort the neighbors ascending order of related gap value in the new list (sol_list)
    For  $i = 1$  to  $A$  do
       $\pi'$  :=  $i$  th solution in the sol_list
      If  $\pi'$  is better than  $\pi$  do
         $\pi := \pi'$ 
        update  $\pi^*$ 
         $k = 1$ 
        break
      Elseif
         $k = k + 1$ 
      Endif
    Endfor
  Endwhile
 $\pi := NS_3(\pi)$ 
Endwhile

```

شکل ۷- شبه کد الگوریتم RVNS پیشنهادی

همچنین زمان پردازش محصول i در مرحله j برابر $T \times d_{i,j} / p_{i,j}$ فرض می‌شود که T بیانگر طول سیکل جواب فعلی الگوریتم است. این الگوریتم دارای تنها یک پارامتر (A) است. شبه کد نهایی الگوریتم پیشنهادی در شکل (۷) ارائه شده است.

۴- نتایج محاسباتی

در این بخش عملکرد مدل و الگوریتم پیشنهادی این مقاله با بهترین مدل و الگوریتم‌های موجود در ادبیات مسئله مقایسه می‌شود. مدل‌های ریاضی در نرم‌افزار GAMS حل شده‌اند به طوری که از موتور حل BARON برای مدل‌های غیرخطی و CPLEX برای مدل‌های خطی استفاده شده است. همچنین از نرم‌افزار MATLAB ۲۰۱۲ برای الگوریتم‌های فرا ابتکاری استفاده شده است. به منظور محاسبه تابع برآزش هر جواب از الگوریتم نرم‌افزار CPLEX 12.4 در نرم‌افزار MATLAB فراخوانی می‌شود. تمامی محاسبات توسط رایانه با مشخصات 2.50 GHz Intel Core i5 Duo و 4 GB حافظه انجام گرفته است.

(۴) تمام جواب‌های ممکن از NS_h را تولید شود. جواب‌های حاصل برحسب میزان تفاوت C_{max} آنها با t_{min} به صورت صعودی در یک لیست جدید مرتب شوند. مقدار تابع برآزش برای جواب‌های ۱ تا A در این لیست محاسبه شود. هرگاه که اولین بهبود برای جواب فعلی حاصل شد، نیازی به محاسبه تابع برآزش سایر جواب‌ها نیست و جواب جدید جایگزین جواب فعلی شود.

(۵) اگر جواب فعلی در گام ۴ بهبود پیدا کرد، الگوریتم از گام ۲ ادامه پیدا می‌کند. در غیر این صورت گام ۶ اجرا می‌شود.
 (۶) یک جواب تصادفی توسط NS_h تولید شده و جایگزین جواب فعلی الگوریتم می‌شود.
 (۷) در هر مرحله از الگوریتم که بهترین جواب موجود بهبود پیدا کرد، جواب جدید جایگزین بهترین جواب جاری می‌شود.
 (۸) گام‌های ۲ تا ۷ تا زمانی که شرط توقف برقرار نشده باشد ادامه خواهند داشت.
 برای محاسبه C_{max} هر جواب در گام‌های ۲ و ۴ از قاعده FAM با در نظرگیری زمان‌های آماده‌سازی استفاده می‌شود.

جدول ۱- مقایسه مدل‌ها از نظر پیچیدگی اندازه

مدل	NBV	NCV	NC
M1	$\cdot / \Delta n(n-1)(\gamma m-1) + n \sum_{j=1}^{\gamma m-1} m_j$	$n(\gamma m-1)+1$	$\gamma n(\gamma m-1) + n(n-1) \left(\sum_{j=1}^{\gamma m-1} m_j \right) + 1$
M2	$\cdot / \Delta n(n-1)(\gamma m-1) + f_{\max} + n \sum_{j=1}^{\gamma m-1} m_j$	$n(\gamma m-1)+1$	$\gamma n(\gamma m-1) + n(n-1) \left(\sum_{j=1}^{\gamma m-1} m_j \right) + \gamma f_{\max} + \gamma$
M3	$n^{\gamma} \sum_{j=1}^{\gamma m-1} m_j$	$n(\gamma m-3)+1$	$\gamma n(\gamma m-1) + n((n-1)^{\gamma} + 1) \left(\sum_{j=1}^{\gamma m-1} m_j \right) + (n-1) \left(\sum_{j \in \Phi} m_j \right) - n + 1$
M4	$n^{\gamma} \sum_{j=1}^{\gamma m-1} m_j + f_{\max}$	$n(\gamma m-3)+1$	$\gamma n(\gamma m-1) + n((n-1)^{\gamma} + 1) \left(\sum_{j=1}^{\gamma m-1} m_j \right) + (n-1) \left(\sum_{j \in \Phi} m_j \right) - n + \gamma f_{\max} + \gamma$

۴-۱- ارزیابی عملکرد مدل ریاضی

در بخش قبلی ابتدا یک مدل ریاضی غیرخطی جدید (M1) و سپس حالت خطی شده آن (M2) برای مسئله مورد بررسی معرفی شد. اکرمی و همکاران [۳] تنها مدل ریاضی موجود در ادبیات مسئله را به صورت یک مدل MINLP پیشنهاد داده‌اند که آن را به اختصار M3 نام گذاری می‌کنیم. با استفاده از رویکرد خطی‌سازی مدل ریاضی که در این مقاله به کار رفته است، می‌توان مدل M3 را به یک مدل خطی تبدیل کرد که آن را مدل M4 می‌نامیم.

در این بخش مدل‌های ریاضی موجود، از نظر پیچیدگی اندازه^{۳۵} و پیچیدگی محاسباتی^{۳۶} مورد ارزیابی قرار می‌گیرند. در بحث پیچیدگی اندازه، هر چهار مدل ریاضی از نظر تعداد متغیرهای جفتی^{۳۷} (NBV)، تعداد متغیرهای پیوسته^{۳۸} (NCV) و تعداد محدودیت‌ها^{۳۹} (NC) با یکدیگر مقایسه خواهند شد. در پیچیدگی محاسباتی، تعدادی مثال عددی از مسئله توسط مدل‌های ریاضی حل شده و نتایج حاصل ارائه می‌شود.

نتایج حاصل از مقایسه مدل‌ها از نظر پیچیدگی اندازه در جدول (۱) ارائه شده است. در این جدول، تعداد محصولات برابر n و تعداد کل مراحل برابر $\gamma m-1$ در نظر گرفته شده است. همچنین Φ بیانگر مجموعه مراحل است که در آنها بیش از یک ماشین وجود دارد. واضح است که مدل غیرخطی معرفی شده در این مقاله (M1) در مقایسه با مدل پیشنهادی اکرمی و

در ادبیات مسئله بزرگ‌ترین اندازه حل شده از مسئله شامل ۱۰ مرحله و ۱۰ محصول است. در این مقاله دو گروه مثال عددی طراحی شده است. گروه اول که ابعاد کوچک و متوسط مسئله فرض شده است شامل ۸ سایز مختلف بوده و تعداد محصولات و مراکز کاری به صورت $m = \{2, 5\}$ و $n = \{4, 6, 8, 10\}$ است. گروه دوم که ابعاد بزرگ مسئله فرض شده است شامل ۶ سایز مختلف بوده و تعداد محصولات و مراکز کاری به صورت $m = \{1, 15\}$ و $n = \{15, 20, 25\}$ است. تعداد ۵ مثال تصادفی برای هر یک از ابعاد در گروه اول تولید شده است. تعداد ۱۰ مثال تصادفی برای هر یک از ابعاد در گروه دوم تولید شده است. تعداد ماشین‌ها در مراکز کاری فرد برابر ۱ و در سایر مراحل برابر ۲ است. همچنین واحد زمان معادل یک روز کاری بوده و افق برنامه‌ریزی برابر ۳۰۰ روز کاری در نظر گرفته شده است. برای تولید سایر پارامترهای مسئله از توزیع احتمالی یکنواخت^{۳۴} (U) به صورت زیر استفاده شده است:

$$d_i \sim U(100, 1000)$$

$$p_{i,j} \sim U(5000, 15000)$$

$$h_{i,1} \sim U(1, 10), h_{i,j} = h_{i,j-1} + U(1, 5)$$

$$s_{i,j} \sim U(0/0, 10/25)$$

$$sc_i = 10000 \left(\sum_{j=1}^{\gamma m-1} s_{i,j} \right) + 10000 \times U(0, 1)$$

$$M = 1000$$

است. ارزیابی الگوریتم‌ها براساس معیار درصد انحراف نسبی^{۴۰} (RPD) و به صورت زیر است. عبارت Sol_{best} بیانگر بهترین جواب الگوریتم‌ها برای مثال عددی مفروض است. عبارت $Sol_{Algorithm}$ بیانگر جواب الگوریتم مورد نظر برای مثال عددی مفروض است:

$$RPD = \frac{Sol_{Algorithm} - Sol_{best}}{Sol_{best}} \times 100$$

تک پارامتر A در الگوریتم پیشنهادی بیانگر تعداد جواب‌های همسایه جدید بررسی شده است که برای آن چهار سطح مختلف برحسب n به صورت $A = \left\{ 1, \frac{n}{4}, n, 2n \right\}$ در نظر گرفته شده است. در نتیجه چهار الگوریتم متفاوت RVNS برای شدن مقدار مناسب پارامتر A تشکیل می‌شود. به همین منظور ۶۰ مثال عددی از گروه دوم تولید شده و توسط هر چهار الگوریتم RVNS حل شده است. نتایج حاصل بیانگر اختلاف معنادار الگوریتم سوم ($A=n$) با سایر الگوریتم‌ها است. در نتیجه مقدار $A=n$ برای الگوریتم RVNS پیشنهادی استفاده می‌شود. نمودار حداقل اختلاف معنادار^{۴۱} (LSD) مربوط به تنظیم پارامتر الگوریتم پیشنهادی در شکل (۸) ارائه شده است.

در این قسمت تعداد ۶۰ مثال عددی از گروه دوم برای انجام مقایسه بین RVNS پیشنهادی و الگوریتم‌های GA و TS اکرمی و همکاران [۳] حل شده است. نتایج حاصل برحسب بهترین (B)، میانگین (Av) و بدترین (W) مقدار RPD های میانگین در جدول گزارش شده است. به منظور تحلیل دقیق‌تر از آزمون تحلیل واریانس^{۴۲} (ANOVA) استفاده شده است. نتایج به دست آمده از آزمون ANOVA در جدول بیانگر اختلاف معنادار بین سه الگوریتم است. الگوریتم RVNS با $RPD = 0/674$ دارای بهترین عملکرد است. الگوریتم TS با $RPD = 3/012$ در جایگاه دوم قرار دارد. همچنین الگوریتم GA با $RPD = 4/921$ دارای بدترین عملکرد است.

۵- نتیجه‌گیری و پیشنهادات

در این مقاله به بررسی مسئله تعیین اندازه انباشته و زمان‌بندی

همکاران (M3) دارای NBV، NCV و NC کمتری است. همچنین مدل‌های خطی M2 و M4 در مقایسه با حالت غیرخطی خودشان یعنی M1 و M3، دارای تعداد محدودیت و تعداد متغیرهای جفتی بیشتری هستند.

نتایج حاصل از حل مثال‌های عددی گروه اول در جدول (۲) ارائه شده است. در ستون اول از این جدول، مقادیر تابع هدف به دست آمده از حل هر مثال گزارش شده است. همچنین زمان‌های محاسباتی حل هر مثال برحسب ثانیه، در ستون دوم (ستون Time) ارائه شده است. همچنین در ستون سوم (ستون Gap)، مقادیر اعلام شده توسط نرم‌افزار پس از متوقف شدن حل هر مثال، برحسب درصد ارائه شده است. به عنوان نمونه در مثال ۱۹، مدل‌های M1 و M2 به جواب بهینه مسئله دست یافته‌اند، در حالی که مدل M4 پس از ۳۶۰۰ ثانیه زمان محاسباتی هنوز به جواب بهینه نرسیده است. همچنین در این مثال، مدل M4 پس از ۳۶۰۰ ثانیه زمان محاسباتی، به هیچ جواب شدنی از مسئله دست نیافته است.

نتایج محاسباتی ارائه شده در جدول (۲) نشان می‌دهد که مدل M3 در اکثر مثال‌ها قادر به یافتن جواب شدنی نیست، در حالی که مدل M1 عملکرد بهتری دارد. مدل‌های خطی (M2 و M4) دارای عملکرد بسیار بهتری نسبت به حالت غیرخطی خودشان (M1 و M3) هستند. همچنین مدل M2 در مقایسه با مدل M4 به مراتب عملکرد بهتری داشته است.

۴-۲- ارزیابی عملکرد الگوریتم فرا ابتکاری

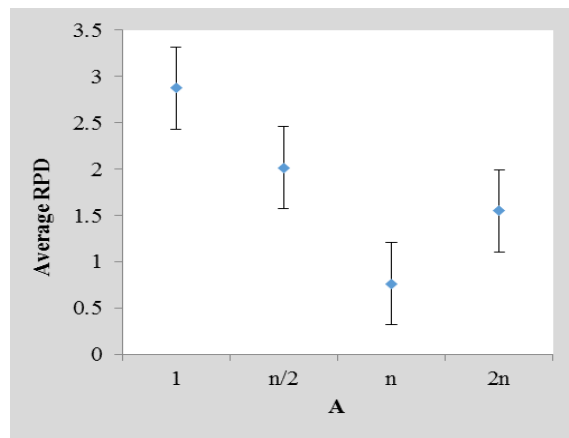
در این بخش الگوریتم پیشنهادی این مقاله با بهترین الگوریتم‌های موجود در ادبیات (GA و TS) ارائه شده توسط اکرمی و همکاران [۳] مورد ارزیابی قرار می‌گیرد. ابتدا تنظیم پارامتر برای الگوریتم RVNS پیشنهادی انجام می‌شود. سپس عملکرد الگوریتم RVNS در مقایسه با GA و TS ارائه شده توسط اکرمی و همکاران [۳] ارزیابی خواهد شد. شرط توقف الگوریتم‌ها برحسب زمان محاسباتی و برابر ۵ nm ثانیه در نظر گرفته شده است. هر مثال ۵ بار توسط هر الگوریتم حل شده

جدول ۲- مقایسه مدل‌ها از نظر پیچیدگی محاسباتی

Gap (%)	Time (s)	Objective	مدل	m	n	مثال	Gap (%)	Time (s)	Objective	مدل	m	n	مثال
۵۵	۳۶۰۰	۱۰۸۴۹۰/۶	M1	۵	۴	۲۱	۰	۱۵/۸	۱۸۰۷۹/۶	M1	۲	۴	۱
۰	۳/۲	۹۲۲۰۶/۶	M2				۰	۲/۸	۱۸۰۷۹/۶	M2			
		—	M3				۰	۱۲۵	۱۸۰۷۹/۶	M3			
۰	۹۶۱	۹۲۲۰۶/۶	M4				۰	۲/۹	۱۸۰۷۹/۶	M4			
۰	۳۱۲۵	۷۳۷۶۴/۹	M1	۵	۴	۲۲	۰	۳۰/۱	۳۲۳۳۵/۶	M1	۲	۴	۲
۰	۳/۱	۷۳۷۶۴/۹	M2				۰	۲/۷	۳۲۳۳۵/۶	M2			
۴۲۰	۳۶۰۰	۲۳۳۹۹۹	M3				۰	۱۶۲	۳۲۳۳۵/۶	M3			
۰	۹/۵	۷۳۷۶۴/۹	M4				۰	۳/۶	۳۲۳۳۵/۶	M4			
۱۱۴	۳۶۰۰	۶۴۴۲۹/۷	M1	۵	۴	۲۳	۰	۸/۵	۲۵۶۱۱/۱	M1	۲	۴	۳
۰	۳/۹	۶۲۳۴۰/۲	M2				۰	۴/۱	۲۵۶۱۱/۱	M2			
		—	M3				۰	۵/۳	۲۵۶۱۱/۱	M3			
۰	۱۵۳۸	۶۲۳۴۰/۲	M4				۰	۲۰/۷	۲۵۶۱۱/۱	M4			
۷۵/۳	۳۶۰۰	۴۸۳۵۵/۴	M1	۵	۴	۲۴	۰	۴/۲	۱۵۹۶۸/۳	M1	۲	۴	۴
۰	۲/۹	۴۱۱۹۶/۸	M2				۰	۲/۸	۱۵۹۶۸/۳	M2			
		—	M3				۰	۱۴/۷	۱۵۹۶۸/۳	M3			
۰	۳۳/۲	۴۱۱۹۶/۸	M4				۰	۸/۶	۱۵۹۶۸/۳	M4			
۱۹/۴	۳۶۰۰	۳۱۹۵۲/۲	M1	۵	۴	۲۵	۰	۲۰/۱	۴۰۰۳۲/۷	M1	۲	۴	۵
۰	۴/۷	۳۱۶۲۹/۸	M2				۰	۵/۹	۴۰۰۳۲/۷	M2			
		—	M3				۰	۱۲/۲	۴۰۰۳۲/۷	M3			
۰	۲۵/۱	۳۱۶۲۹/۸	M4				۰	۹/۳	۴۰۰۳۲/۷	M4			
		—	M1	۵	۶	۲۶	۰	۱۵/۸	۳۸۸۰۷۹	M1	۲	۶	۶
۰	۸/۷	۱۳۶۲۵۴۹۳	M2				۰	۴/۵	۳۸۸۰۷۹	M2			
		—	M3				۰	۱۴۲۶	۳۸۸۰۷۹	M3			
		—	M4				۰	۲۴/۷	۳۸۸۰۷۹	M4			
۱۰۰	۳۶۰۰	۱۸۷۳۲۷۶	M1	۵	۶	۲۷	۰	۱۲/۹	۶۱۲۸۰	M1	۲	۶	۷
۰	۹/۸	۵۳۱۹۴۹/۵	M2				۰	۵	۶۱۲۸۰	M2			
		—	M3				۰	۹۵/۴	۶۱۲۸۰	M3			
		—	M4				۰	۱۵۳۷	۶۱۲۸۰	M4			
		—	M1	۵	۶	۲۸	۰	۱۳۶۲	۵۳۹۴۴/۱	M1	۲	۶	۸
۰	۳۴/۱	۹۳۷۷۲۸۱	M2				۰	۲۷/۵	۵۳۹۴۴/۱	M2			
		—	M3				۱۷/۹	۳۶۰۰	۵۵۲۶۴/۵	M3			
		—	M4				۰	۵۵۸	۵۳۹۴۴/۱	M4			
		—	M1	۵	۶	۲۹	۰	۱۶۶	۲۹۹۳۷	M1	۲	۶	۹
۰	۹۲/۵	۶۵۳۷۴۲۴	M2				۰	۲/۸	۲۹۹۳۷	M2			
		—	M3				۰	۱۰۹۳	۲۹۹۳۷	M3			
		—	M4				۰	۶/۴	۲۹۹۳۷	M4			
		—	M1	۵	۶	۳۰	۰	۲۹/۴	۷۳۲۸۴/۳	M1	۲	۶	۱۰
۰	۱۶/۴	۴۲۲۹۶۳۱	M2				۰	۸/۵	۷۳۲۸۴/۳	M2			
		—	M3				۹/۹	۳۶۰۰	۷۳۳۶۹/۴	M3			
۰	۲۳۸	۴۲۲۹۶۳۱	M4				۰	۴۴۹	۷۳۲۸۴/۳	M4			
		—	M1	۵	۸	۳۱	۰	۲۵۴۹	۴۳۳۰۶۸/۹	M1	۲	۸	۱۱
۰	۱۶۲۸	۴۷۷۲۰۴/۶	M2				۰	۶۱/۸	۴۳۳۰۶۸/۹	M2			
		—	M3						—	M3			

جدول (۲) - ادامه

Gap (%)	Time (s)	Objective	مدل	m	n	مثال	Gap (%)	Time (s)	Objective	مدل	m	n	مثال
۷۴	۳۶۰۰	۱۵۲۱۲۷۷	M4				۰	۱۸۳۶	۴۳۰-۶۸/۹	M4			
۴۷۶	۳۶۰۰	۱۰۶۲۸۳۹	M1	۵	۸	۳۲	۰	۳۱۰۲	۱۴۷۶۲۹/۵	M1	۲	۸	۱۲
۰	۲۱۵۰	۳۸۴۸۲۶۷	M2				۰	۲۵/۸	۱۴۷۶۲۹/۵	M2			
		—	M3						—	M3			
۸۷	۳۶۰۰	۱۴۵۶۰۷۵	M4				۰	۱۶۳۱	۱۴۷۶۲۹/۵	M4			
		—	M1	۵	۸	۳۳	۶/۲	۳۶۰۰	۱۵۱۳۹۸۳	M1	۲	۸	۱۳
۰	۲۹۴۴	۵۴۹۳۱۲۲	M2				۰	۵۱/۷	۱۵۱۳۷۸۸	M2			
		—	M3						—	M3			
۶۲	۳۶۰۰	۵۵۲۴۸۸۳	M4				۰	۲۰۹۳	۱۵۱۳۷۸۸	M4			
		—	M1	۵	۸	۳۴	۱۸/۱	۳۶۰۰	۸۶۲۴۲۵۳	M1	۲	۸	۱۴
۰	۱۴۶۶	۶۰۸۱۲۳۳	M2				۰	۱۱۴	۸۶۲۰۶۷۹	M2			
		—	M3						—	M3			
۳۷	۳۶۰۰	۶۰۹۵۲۱۸	M4				۰	۲۳۵۹	۸۶۲۰۶۷۹	M4			
		—	M1	۵	۸	۳۵	۰	۳۰۹۴	۵۷۱۶۴۹۲	M1	۲	۸	۱۵
۰	۲۲۷۸	۴۹۳۷۲۵۰	M2				۰	۸۳	۵۷۱۶۴۹۲	M2			
		—	M3						—	M3			
۹۱	۳۶۰۰	۵۰۰۴۷۱۲	M4				۰	۱۹۰۴	۵۷۱۶۴۹۲	M4			
		—	M1	۵	۱۰	۳۶			—	M1	۲	۱۰	۱۶
۲۵	۳۶۰۰	۹۱۲۳۸۴۵	M2				۰	۳۱۰۵	۶۴۹۹۲۸	M2			
		—	M3						—	M3			
۷۲	۳۶۰۰	۹۲۳۹۱۵۷	M4				۲۶/۸	۳۶۰۰	۶۹۳۷۹۱۱	M4			
		—	M1	۵	۱۰	۳۷	۰	۳۲۵۹	۲۵۴۸۳۳۰	M1	۲	۱۰	۱۷
۱۹	۳۶۰۰	۵۲۶۴۹۷۳	M2				۰	۲۱۹۵	۲۵۴۸۳۳۰	M2			
		—	M3						—	M3			
۳۴	۳۶۰۰	۵۲۷۹۴۰۸	M4						—	M4			
		—	M1	۵	۱۰	۳۸			—	M1	۲	۱۰	۱۸
۸,۲	۳۶۰۰	۷۷۱۲۵۶۰	M2				۶/۸	۳۶۰۰	۲۱۵۳۶۸۵	M2			
		—	M3						—	M3			
۳۱	۳۶۰۰	۷۸۰۱۱۶۴	M4				۱۹/۴	۳۶۰۰	۲۴۱۲۲۹۱	M4			
		—	M1	۵	۱۰	۳۹	۰	۳۵۱۲	۱۸۴۵۷۶/۱	M1	۲	۱۰	۱۹
۳۵	۳۶۰۰	۶۸۲۶۰۶/۶	M2				۰	۲۴۲۱	۱۸۴۵۷۶/۱	M2			
		—	M3						—	M3			
۸۴	۳۶۰۰	۲۳۲۸۴۵۰	M4				۱۸	۳۶۰۰	۱۹۷۷۰۵/۹	M4			
		—	M1	۵	۱۰	۴۰			—	M1	۲	۱۰	۲۰
۵۷	۳۶۰۰	۹۹۷۴۴۲	M2				۰	۲۰۴۷	۲۴۵۱۵۵۸	M2			
		—	M3						—	M3			
۸۲	۳۶۰۰	۲۲۴۱۸۰۴	M4						—	M4			



شکل ۸- نمودار LSD تنظیم پارامتر الگوریتم

جدول ۳- نتایج حاصل از حل الگوریتم‌ها

		RPD								
		B			Av			W		
n	m	GA	TS	RVNS	GA	TS	RVNS	GA	TS	RVNS
۱۵	۱۰	۰/۱۰۹	۰/۷۱۸	۰/۰۷۴	۳/۲۲۵	۲/۵۰۲	۰/۴۵۳	۶/۱۹۳	۵/۸۹۹	۰/۸۲۱
۲۰	۱۰	۲/۲۴۱	۰/۱۲۴	۰/۲۱۵	۳/۹۰۲	۲/۴۸۷	۰/۶۳۸	۷/۶۶۱	۴/۳۱۸	۱/۰۰۵
۲۵	۱۰	۱/۹۱۴	۰/۶۷۲	۰/۳۰۱	۵/۷۶۶	۳/۵۳۲	۰/۸۵۱	۹/۴۰۸	۵/۷۵۳	۱/۴۳۹
۱۵	۱۵	۲/۰۲۴	۰/۱۰۶	۰/۱۸۹	۵/۲۹۰	۲/۱۶۵	۰/۴۳۸	۸/۱۴۲	۶/۴۷۹	۰/۷۱۲
۲۰	۱۵	۰/۳۰۵	۰/۱۸۶	۰/۲۱۹	۶/۴۱۹	۴/۰۷۹	۰/۶۲۵	۱۲/۳۳۵	۱۰/۵۲۳	۲/۲۴۱
۲۵	۱۵	۰/۲۹۹	۰/۱۷۰	۰/۰۸۳	۴/۹۲۵	۳/۳۱۰	۱/۰۴۱	۱۰/۸۱۷	۹/۶۲۱	۳/۶۷۴

جدول ۴- نتایج حاصل از آزمون تحلیل واریانس

P_value	F	MS	SS	df	Factor
۰/۰۰۰	۶۲/۴۳	۴۱۰/۷۶	۸۲۱/۵۲	۲	Algorithm
		۶/۵۸	۱۱۶۵/۲۹	۱۷۷	Error
			۱۹۸۶/۸۱	۱۷۹	Total

پیچیدگی اندازه و پیچیدگی محاسباتی مقایسه شده‌اند. در پیچیدگی اندازه، مدل‌های ریاضی از نظر تعداد محدودیت و تعداد متغیرها مورد بررسی قرار گرفته‌اند. در پیچیدگی محاسباتی، تعدادی مثال عددی در ابعاد کوچک و متوسط به‌طور مشترک توسط مدل‌ها حل شده است. نتایج حاصل از این ارزیابی‌ها، بیانگر برتری قابل توجه مدل پیشنهادی نسبت به مدل موجود در ادبیات مسئله است. علاوه بر این، یک الگوریتم

چند محصول در محیط جریان کارگاهی مختلط با در نظر گرفتن بافرهای میانی محدود پرداخته شد. به دلیل غیرخطی بودن مدل ریاضی موجود در ادبیات مسئله، دستیابی به جواب بهینه یا حتی جواب شدنی طی یک زمان محاسباتی معقول در اکثر موارد امکان‌پذیر نیست. در این تحقیق، یک مدل ریاضی خطی عدد صحیح مختلط برای مسئله ارائه شده است. برای ارزیابی عملکرد، مدل پیشنهادی و مدل موجود در ادبیات مسئله از نظر

به منظور انجام تحقیقات آتی می توان از الگوریتم های فرا ابتکاری کارآمد برای حل مسئله در ابعاد بزرگ استفاده کرد. همچنین مدل سازی مسئله تحت رویکرد انباشته های متغیر با زمان که در ادبیات مسئله کمتر مورد توجه قرار گرفته می تواند پژوهش ارزشمندی باشد.

فرا ابتکاری کارآمد براساس الگوریتم جستجوی همسایگی متغیر برای حل مسئله در ابعاد بزرگ تر ارائه شد. پس از تنظیم پارامتر الگوریتم پیشنهادی عملکرد آن با بهترین الگوریتم های موجود مورد ارزیابی قرار گرفت. نتایج حاصل بیانگر برتری معنادار الگوریتم پیشنهادی است.

واژه نامه

- | | | |
|--|---|---|
| 1. lot sizing | 15. non polynomial | 30. local search |
| 2. production cost | 16. period | 31. first available machine (FAM) |
| 3. setup cost | 17. common cycle | 32. fitness function |
| 4. holding cost | 18. basic period | 33. neighbor structure |
| 5. shortage cost | 19. time varying lot sizes | 34. uniform distribution |
| 6. sequencing | 20. zero switch rule | 35. size complexity |
| 7. scheduling | 21. shortest processing time | 36. computing complexity |
| 8. economic lot scheduling problem (ELSP) | 22. Tabu Search | 37. number of binary variable (NBV) |
| 9. flow shop (FS) | 23. simulated annealing | 38. number of continuous variable (NCV) |
| 10. hybrid flow shop (HFS) | 24. genetic algorithm | 39. number of constraint (NC) |
| 11. intermediate buffers | 25. preemption | 40. relative percentage deviation (RPD) |
| 12. mixed integer non linear programming (MINLP) | 26. lot splitting | 41. least significant difference (LSD) |
| 13. economic production quantity (EPQ) | 27. work in process (WIP) | 42. analyze of variance (ANOVA) |
| 14. independent solutions | 28. reduced variable neighbor search (RVNS) | |
| | 29. fly optimization algorithm (FOA) | |

مراجع

- Karimi, B., Ghomi, S. F., and Wilson, J. M., "The Capacitated Lot Sizing Problem: A Review of Models and Algorithms", *Omega*, Vol. 31, pp. 365-378, 2003.
- Santander-Mercado, A., and Jubiz-Diaz, M., "The Economic Lot Scheduling Problem: A Survey", *International Journal of Production Research*, pp. 1-20, 2016.
- Akrami, B., Karimi, B., and Hosseini, S. M., "Two Metaheuristic Methods for the Common Cycle Economic Lot Sizing and Scheduling in Flexible Flow Shops with Limited Intermediate Buffers: The Finite Horizon Case", *Applied Mathematics and Computation*, Vol. 183, pp. 634-645, 2006.
- Taft, E. W., "The Most Economical Production Lot", *Iron Age*, Vol. 101, pp. 1410-1412, 1918.
- Rogers, J., "A Computational Approach to the Economic Lot Scheduling Problem", *Management Science*, Vol. 4, pp. 264-291, 1958.
- Gallego, G., and Shaw, D. X., "Complexity of the ELSP with General Cyclic Schedules", *IIE Transactions*, Vol. 29, pp. 109-113, 1997.
- Davis, S. G., "Scheduling Economic Lot Size Production Runs", *Management Science*, Vol. 36, pp. 985-998, 1990.
- Moon, I., Silver, E. A., and Choi, S., "Hybrid Genetic Algorithm for the Economic Lot-Scheduling Problem", *International Journal of Production Research*, Vol. 40, pp. 809-824, 2002.
- Holmbom, M., and Segerstedt, A., "Economic Order Quantities in Production: From Harris to Economic Lot Scheduling Problems", *International Journal of Production Economics*, Vol. 155, pp. 82-90, 2014.
- Dobson, G., and Yano, C. A., "Cyclic Scheduling to Minimize Inventory in a Batch Flow Line", *European Journal of Operational Research*, Vol. 75, pp. 441-461, 1994.
- El-Najdawi, M. K., and Kleindorfer, P. R., "Common Cycle Lot-Size Scheduling for Multi-Product, Multi-Stage Production", *Management Science*, Vol. 39, pp. 872-885, 1993.
- El-Najdawi, M. K., "A Job-Splitting Heuristic for Lot-Size Scheduling in Multi-Stage, Multi-Product Production Processes", *European Journal of Operational Research*, Vol. 75, pp. 365-377, 1994.
- Ouenniche, J., Boctor, F. F., and Martel, A., "The Impact of Sequencing Decisions on Multi-Item Lot Sizing and Scheduling in Flow Shops", *International*

Journal of Production Research, Vol. 37, pp. 2253-2270, 1999.

14. Torabi, S. A., Fatemi Ghomi, S. M. T., and Karimi, B., "A Hybrid Genetic Algorithm for the Finite Horizon Economic Lot and Delivery Scheduling in Supply Chains", *European Journal of Operational Research*, Vol. 173, pp. 173-189, 2006.
15. Jenabi, M., Fatemi Ghomi, S. M. T., Torabi, S. A., and Karimi, B., "Two Hybrid Meta- Heuristics for the Finite Horizon ELSP in Flexible Flow Lines with Unrelated Parallel Machines", *Applied Mathematics and Computation*, Vol. 186, pp. 230-245, 2007.
16. Mladenović, N., and Hansen, P., "Variable Neighborhood Search", *Computers & Operations Research*, Vol. 24, No. 11, pp. 197-1100, 1997.
17. Deming, L., "Variable Neighborhood Search for Two-Agent Flow Shop Scheduling Problem", *Computers & Industrial Engineering*, Vol. 80, pp. 125-131, 2015.
18. Moslehi, Gh., and Khorasanian. D., "A Hybrid Variable Neighborhood Search Algorithm for Solving the Limited-Buffer Permutation Flow Shop Scheduling Problem with the Makespan Criterion", *Computers & Operations Research*, Vol. 52, pp. 260-268, 2014.
19. Hatami, S., Ruiz, R., and Andrés-Romano, C., "The Distributed Assembly Permutation Flowshop Scheduling Problem", *International Journal of Production Research* Vol. 51, No. 17, pp. 5292-5308, 2013.
20. Hansen, P., and Mladenovic, N., *A Tutorial on Variable Neighborhood Search*. Groupe d'études et de recherche en analyse des décisions, HEC Montréal, 2003.
21. Wen-Tsao, P. "A New Fruit Fly Optimization Algorithm: Taking the Financial Distress Model as an Example", *Knowledge-Based Systems*, Vol. 26, pp. 69-74, 2012.
22. Ouenniche, J., and Bector. F., "Sequencing, Lot Sizing and Scheduling of Several Products in Job Shops: The Common Cycle Approach", *International Journal of Production Research*, Vol. 36, No. 4, pp. 1125-1140, 1998.