

یک الگوریتم فراابتکاری ترکیبی برای مسئله زمان بندی کار کارگاهی منعطف با منابع دوگانه محدود انسان و ماشین

مهدی یزدانی*، مصطفی زندیه**، رضا توکلی مقدم***

(تاریخ دریافت: ۹۲/۷/۲۰ - تاریخ پذیرش: ۹۲/۹/۳۰)

چکیده

در این مقاله مسئله زمان بندی کار کارگاهی منعطف با در نظر گرفتن منابع دوگانه محدود انسان و ماشین با هدف کمینه سازی معیار حداکثر زمان تکمیل کارها مورد بررسی قرار گرفته است. مسئله مورد مطالعه از گروه مسائل NP-hard است و از ۳ زیرمسئله تشکیل شده است. مسئله اول تخصیص هر عملیات به یک ماشین از میان ماشین های موجود برای انجام آن عملیات، مسئله دوم تخصیص هر عملیات به یک کارگر از میان کارگرهای قادر به انجام آن عملیات و مسئله دیگر تعیین توالی عملیات ها روی ماشین ها با توجه به کارگران در نظر گرفته شده به منظور بهینه سازی معیار عملکرد می باشد. ما در این مقاله مدل ریاضی مسئله مورد نظر را تهیه و در ادامه یک الگوریتم فراابتکاری ترکیبی را برای حل آن ارائه کرده ایم. الگوریتم ترکیبی توسعه داده شده از الگوریتم های جستجوی همسایگی متغیر و شبیه سازی تبرید برای جستجوی فضای جواب استفاده می کند. به منظور ارزیابی عملکرد الگوریتم ارائه شده، مطالعات محاسباتی با در نظر گرفتن مسائل نمونه ایجاد شده انجام خواهد شد. نتایج نشان می دهد که الگوریتم ارائه شده روشی مؤثر برای حل مسئله زمان بندی کار کارگاهی منعطف با منابع دوگانه محدود انسان و ماشین است.

کلمات کلیدی: زمان بندی، منابع دوگانه محدود انسان و ماشین، محیط کار کارگاهی منعطف، مدل سازی ریاضی، الگوریتم شبیه سازی تبرید، الگوریتم جستجوی همسایگی متغیر.

دانشگاه آزاد اسلامی، واحد علوم و تحقیقات، گروه مهندسی صنایع، تهران، ایران (نویسنده مسئول)

Mehdi_Yazdani2007@yahoo.com

** دانشیار، گروه مدیریت صنعتی، دانشکده مدیریت و حسابداری، دانشگاه شهید بهشتی، تهران، ایران

*** استاد، دانشکده مهندسی صنایع، پردیس دانشکده های فنی، دانشگاه تهران، تهران، ایران

مقدمه

زمان‌بندی تولید به‌عنوان یک جنبه مهم از مباحث برنامه‌ریزی تولید، چالشی پیچیده و با اهمیت در حوزه زمان‌بندی وظایف و تخصیص منابع است و همواره مسائل آن مورد توجه محققان بوده‌است. مسئله زمان‌بندی را می‌توان چنین تعریف نمود: تعیین توالی زمانی و تخصیص سفارشات مشتریان به منابع موجود تولید (اعم از پرسنل، ماشین‌آلات، ابزار و غیره) به‌منظور انجام مجموعه‌ای از عملیات مربوطه در یک زمان معین (پیندو، ۱۹۹۵). مسئله زمان‌بندی کار کارگاهی (*JSP*)^۲ یکی از شناخته‌شده‌ترین و پیچیده‌ترین مسائل زمان‌بندی است که در آن می‌بایست n کار روی m ماشین با حفظ توالی خاصی انجام شوند. هر کار در این محیط از عمل‌های مختلفی تشکیل شده‌است و مسیر پردازش آن‌ها بر روی ماشین‌ها از قبل مشخص است. بین عملیات مربوط به یک کار روابط پیش‌نیازی وجود دارد. هدف کلی در این مسئله تعیین زمان‌بندی و توالی مناسب مجموعه‌ای از کارها بر روی مجموعه‌ای از ماشین‌ها با توجه به معیار بهینه‌سازی در نظر گرفته‌شده. قابل ذکر است، این مسئله جزء دسته مسائل *NP-Hard* قرار دارد (گری و همکاران، ۱۹۷۶).

امروزه نه تنها بقاء هر سیستم تولیدی و خدماتی در گرو میزان قابلیت آن سیستم در سازگاری با نظر و سلیقه مشتری است، بلکه پیش‌رو بودن و هدایت سلیقه مشتری از ارکان مهم موفقیت سازمان‌ها به‌شمار می‌آید. در این میان برخی سیاست‌های استراتژیک در صورت استفاده می‌توانند منجر به افزایش قابلیت‌های سازگاری در سیستم‌ها شوند. از جمله این سیاست‌ها، بهره‌گیری از مزایای اقتصادی حاصل از پیامدهای استقرار سیستم‌های تولیدی پیشرفته و ایجاد توانایی در سیستم‌ها در انطباق با تحولات سریع بازار بر اثر تغییر سریع ذائقه مشتری است. با توجه به این موارد مطالعه در ارتباط با محیط‌های کار کارگاهی منعطف که از دسته سیستم‌های تولیدی پیشرفته است، می‌تواند برای بخش صنعت سودمند باشد. مسئله

1 - Pinedo

2 - Job Shop

3 - Garey

4 - Flexible Job Shop

زمان‌بندی کار کارگاهی منعطف^۱ (*FJSP*) حالت عمومی و بسط داده‌شده مسئله زمان‌بندی کار کارگاهی است، هنگامی که هر عملیات از یک کار می‌تواند روی یک ماشین از مجموعه ماشین‌های در دسترس برای پردازش آن، انجام شود (پزلا و همکاران، ۲۰۰۸). این مسئله را می‌توان به دو زیر مسئله تقسیم کرد. (۱) زیر مسئله تخصیص عملیات به ماشین‌ها (تخصیص هر عملیات به یک ماشین از میان ماشین‌های در دسترس) و (۲) زیر مسئله توالی عملیات روی ماشین‌ها (مشخص کردن ترتیب انجام عملیات‌های مربوط به کارهای مختلف روی ماشین‌ها). هدف مسئله *FJSP* تعیین تخصیص و توالی مناسب و در نهایت زمان‌بندی بهینه یا نزدیک به بهینه برای مسئله مورد نظر با توجه به معیار بهینه‌سازی مورد مطالعه می‌باشد. ماتی و ژی (۲۰۰۴) ثابت کرده‌اند که مسئله *FJSP* در گروه مسائل *NP-Hard* قرار دارد. پیچیدگی این مسئله به مراتب از پیچیدگی مسئله *JSP* بیشتر است.

در اکثر مسائل زمان‌بندی موجود در ادبیات تنها تجهیزات کارگاهی نظیر ماشین‌ها به عنوان منبع محدود در نظر گرفته شده‌اند و محدودیت‌هایی نظیر دسترس‌پذیری کارکنان و مهارت‌های مربوط به آن‌ها جهت اجرای فرایندها نادیده گرفته شده‌است. در حالی که در واقعیت، در نظر گرفتن هر دو منبع انسان و ماشین در کنار هم، در تنظیم برنامه زمان‌بندی و اجرای کارهای سیستم مؤثرند. سیستم‌هایی که در آن بحث کاربرد دو منبع نیروی انسانی و ماشین برای تهیه برنامه زمان‌بندی وجود دارد و منابع اشاره شده محدود در نظر گرفته می‌شوند، سیستم‌هایی با منابع دوگانه محدود^۴ (*DRC*) نامیده شده‌اند (المراغی و همکاران، ۲۰۰۰). در محیط این گونه سیستم‌ها علاوه بر این که ماشین‌ها همیشه در دسترس نیستند، کارگران نیز به علت مشغول بودن در ایستگاه‌های مختلف و حرکت در بین ایستگاه‌ها، ممکن است در دسترس نباشند. بنابراین تصمیمات زمان‌بندی در محیط *DRC* هم ظرفیت ماشین‌ها و هم دسترس‌پذیری نیروی انسانی را می‌بایست در نظر بگیرد (جینگیاو و همکاران، ۲۰۱۰).

1- Flexible Job Shop Scheduling

2- Pezzella

3 - Mati and Xie

4- Dual resource constrained

5 - ElMaraghy

6 - Jingyao

نلسون (۱۹۶۷) اولین کسی بود که مبحث زمان‌بندی در محیط *DRC* را مورد مطالعه قرار داد. با مطالعه تحقیقات انجام‌شده در ارتباط با مبحث زمان‌بندی در محیط *DRC*، مشخص می‌شود که تعداد تحقیقات انجام‌شده در این زمینه محدود است. همچنین اکثر این تحقیقات مربوط به محیط زمان‌بندی کار کارگاهی با منابع دوگانه محدود (*DRCJSP*) است که در ادامه به آن خواهیم پرداخت. ژيجون و جیانینگ (۲۰۰۵) یک روش بهینه‌سازی هوشمند را که براساس الگوریتم ژنتیک (*GA*) ایجاد شده‌بود، برای مسئله *DRCJSP* به کار گرفتند. ژینگانگ و همکاران (۲۰۰۷) الگوریتم اجتماع مورچگان^۶ (*ACO*) را برای همین مسئله به کار بردند. هویوآن و همکاران (۲۰۰۹) مسئله *DRCJSP* را با توجه به عوامل واقعی موجود در محیط صنایع قالب‌گیری تزریقی پلاستیک مورد مطالعه قرار دارند. دو منبع لحاظ‌شده در این تحقیق، ماشین و قالب در نظر گرفته شده‌بود. آن‌ها یک روش ابتکاری فعال ترکیب‌شده با قانون‌های اولویتی را برای حل این مسئله، با توجه به کمینه‌سازی معیار عملکرد حداکثر زمان تکمیل کارها^۷ به کار گرفتند. جابر و نیومن (۲۰۱۰) با در نظر گرفتن عامل خستگی و توجه به بازیابی و بهبودی نیروی کاری، یک مدل برنامه‌ریزی خطی عدد صحیح مختلط را برای مسئله *DRCJSP* به کار گرفتند. پیچیدگی مدل ارائه‌شده با توجه به در نظر گرفتن چهار وضعیت متفاوت کاهش پیدا کرد. جینگیانو و همکاران (۲۰۱۰) یک الگوریتم *GA* را برای مسئله *DRCJSP* دو هدفه، با فرض وجود کارکنان غیریکسان، ارائه کردند. بهره‌وری و شاخص میانگین وزنی خستگی به عنوان معیارهای عملکرد در نظر گرفته شد و با استفاده از قاعده جمع وزنی تابع هدف مسئله تشکیل شد. جینگیانو و همکاران (۲۰۱۱) یک الگوریتم تلفیقی را براساس ترکیب *ACO* و *SA* (جستجوگر محلی)، به منظور حل مسئله *DRCJSP* با هدف

1 - Nelson

2 - Dual resource constrained Job Shop Scheduling Problem

3 - Zhijun and Jianying

4- Genetic Algorithm

5 - ZhiGang

6- Ant Colony Optimization

7 - Huiyuan

8 - Makespan

9 - Jaber and Neumann

10 - Jingyao

کمیته‌سازی حداکثر زمان تکمیل کارها به کار گرفتند. لوبو و همکاران (۲۰۱۳) یک فرایند مؤثر را به منظور محاسبه حد پایین مسئله *DRCJSP* که در آن کمیته‌سازی حداکثر دیرکرد به عنوان تابع هدف تعیین شده بود، توسعه دادند.

مسئله *FJSP* یکی از مسائل مهم در محیط مسائل زمان‌بندی است که در تحقیقات انجام شده در ارتباط با آن توجه بسیار کمی به منابع انسانی صورت گرفته است. در حالی که برای انجام عملیات در چنین محیطی نیاز به نیروی انسانی تردیدناپذیر است. لذا مسئله زمان‌بندی کار کارگاهی منعطف با منابع دوگانه محدود انسان و ماشین (*DRCFJSP*) به عنوان یکی از پیچیده‌ترین و جدیدترین مسائل بهینه‌سازی ترکیباتی در محیط‌های *DRC* توسط محققان مطرح و مورد مطالعه قرار گرفته است. در این مسئله با توجه به اضافه شدن نیروی انسانی انعطاف‌پذیر و چندمهارته به محیط مسئله، حالت‌هایی پیش می‌آید که برای انجام یک عملیات، بیش از یک کارگر وجود دارد. بنابراین با یک مبحث ویژه مواجهیم و آن تخصیص کارگر مناسب برای انجام یک کار بر روی ماشین در دسترس است. مسئله *DRCFJSP* را می‌توان به سه زیرمسئله تقسیم کرد. مسئله اول تخصیص هر عملیات به یک ماشین از میان ماشین‌های موجود برای انجام آن عملیات، مسئله دوم تخصیص هر عملیات به یک کارگر از میان کارگرهای قادر به انجام آن عملیات و مسئله دیگر تعیین توالی عملیات روی ماشین‌ها با توجه به کارگران در نظر گرفته شده به منظور بهینه‌سازی معیار عملکرد مورد نظر است. با مطالعه ادبیات مشخص می‌شود که تعداد معدودی از مقالات موضوع *DRCFJSP* را به عنوان موضوع تحقیق تعیین کرده‌اند. ژیانزو و ژنهی (۲۰۱۱) یک الگوریتم ایمنی مصنوعی^۴ (*AIA*) و *GA* ترکیبی را به منظور حل مسئله زمان‌بندی کارگاهی منعطف با منابع دوگانه محدود انسان و ماشین (*DRCFJSP*) به کار گرفتند. الگوریتم ارائه شده به شکل مؤثری از هم‌گرایی زودرس اجتناب می‌کند که این عامل باعث بالارفتن فشار انتخاب می‌شود. لوبو و

1 - Lobo

2 - Dual resource constrained flexible Job Shop Scheduling Problem

3 - Xianzhou and Zhenhe

4- Artificial Immune Algorithm

5 - Liu

همکاران (۲۰۱۱) یک الگوریتم ژنتیک ترکیبی مبتنی بر رویکرد پارتورا برای مسئله *DRCFJSP* با اهداف دوگانه توسعه دادند. ما کزیم زمان تکمیل کارها و هزینه‌های تولید به عنوان اهداف مسئله معرفی شدند. الگوریتم مورد نظر از روش جایگاه ویژه و چهار نوع عملگر تقاطع به منظور بهبود کیفیت جواب‌ها استفاده کرده‌است.

با توجه به ادبیات موجود و تعداد اندک تحقیقات انجام شده، در این تحقیق مسئله *DRCFJSP* به عنوان موضوع اصلی تحقیق مورد مطالعه قرار گرفته‌است. عدم ارائه یک مدل برنامه‌ریزی ریاضی در حالت‌های مختلف خطی یا عدد صحیح برای مسئله *DRCFJSP* یکی از ضعف‌های موجود در ادبیات تحقیق تاکنون است. لذا در این تحقیق مدل برنامه‌ریزی ریاضی این مسئله تنظیم و همراه با توضیحات کامل توصیف خواهد شد. از آنجایی که مسئله *DRCFJSP* در گروه مسائل *NP-Hard* قرار دارد، ارائه یک روش دقیق که بتواند مسئله مورد نظر را در یک زمان منطقی مورد بهینه‌سازی قرار دهد، بسیار سخت و دور از ذهن است. اما در این حالت توسعه الگوریتم‌های فراابتکاری برای حل اینگونه مسائل در یک زمان معقول می‌تواند استراتژی بسیار مناسبی باشد. لذا در این مقاله یک الگوریتم فراابتکاری ترکیبی برای بهینه‌سازی مسئله *DRCFJSP* ارائه خواهد شد. روش مورد نظر از ترکیب الگوریتم‌های جستجوی همسایگی متغیر (*VNS*) و شبیه‌سازی تبرید (*SA*) به منظور جستجوی فضای جواب استفاده می‌کند. در ادامه به منظور ارزیابی عملکرد الگوریتم ارائه شده و مقایسه آن با الگوریتم‌های دیگر، مطالعات محاسباتی با در نظر گرفتن مسائل نمونه ایجاد شده انجام خواهد شد. همچنین روشی برای به دست آوردن حد پایین جواب مسئله‌ها ارائه می‌شود.

بخش‌های مختلف مقاله در ادامه به این صورت تنظیم شده‌است. در بخش دوم تعریف مسئله با شرح کامل ارائه خواهد شد. مدل ریاضی مسئله مورد نظر در بخش سوم شرح داده خواهد شد. الگوریتم ترکیبی ارائه شده برای حل مسئله *DRCFJSP* در بخش چهارم مورد مطالعه قرار خواهد گرفت. نتایج محاسباتی الگوریتم‌ها و مقایسات آن‌ها در بخش پنجم گزارش شده‌است. نتیجه‌گیری و تحقیقات آتی در بخش ششم بیان خواهد شد.

1 - Variable neighborhood search

2 - Simulated annealing

بیان مسئله

مسئله DRCFJSP عبارت است از برنامه‌ریزی اجرای n کار روی m ماشین توسط l کارگر. در این مسئله مجموعه‌ای از کارها $J = \{J_1, J_2, \dots, J_n\}$ ، ماشین‌ها $M = \{M_1, M_2, \dots, M_m\}$ و کارگران $W = \{W_1, W_2, \dots, W_h\}$ وجود دارند. پردازش کارها به دو منبع محدود شده است: ماشین و انسان. هر کار J_i از n_i عملیات $(O_{i,1}, O_{i,2}, \dots, O_{i,n_i})$ تشکیل شده است که این عملیات‌ها با توجه به روابط پیش‌نیازی که بین آن‌ها وجود دارد، اجرا می‌شوند. هر عملیات $O_{i,j}$ ، عملیات J_i را از کار نام، می‌تواند روی یک ماشین از میان مجموعه ماشین‌های قادر به پردازش آن عملیات $M_{i,j} = \{M_{i,j,1} \cup M_{i,j,2} \cup \dots \cup M_{i,j,l} \cup \dots \cup M_{i,j,h-1} \cup M_{i,j,h}\}$ اجرا شود جایی که $M_{i,j,l}$ نشان‌دهنده ماشینی است که می‌تواند عملیات $O_{i,j}$ را توسط کارگر l ام انجام دهد. کارگران از جنبه انعطاف‌پذیری در مسائل به دو دسته تقسیم می‌شوند. دسته اول به کارگرانی اشاره می‌کند که از لحاظ مهارت انجام یک عملیات روی تمام ماشین‌ها یکسان هستند (المراغی و همکاران، ۱۹۹۹؛ المرأغی و همکاران، ۲۰۰۰). اما دسته دوم کارگرانی هستند که مهارت‌های متفاوتی را از لحاظ انجام عملیات روی ماشین‌های مختلف دارا هستند (بخورست و گالمن، ۲۰۰۹). در اکثر مسائل موجود در ادبیات محیط DRC، به دلیل ساده کردن محاسبات از نوع اول کارگران استفاده شده است. اما در این تحقیق کارگران غیرهمگن یعنی دسته دوم کارگران در فضای مسئله مورد مطالعه قرار گرفته‌اند. همچنین در این مسئله مجموعه کارگرانی که قادر به انجام کار روی یک ماشین هستند می‌تواند با ماشینی دیگر متفاوت باشد. S_k نشان‌دهنده مجموعه کارگرانی است که توانایی انجام کار روی ماشین k را دارا هستند.

فرضیات زیر در مسئله DRCFJSP برقرار است:

- کلیه کارها می‌توانند از زمان صفر شروع شوند و زمان آزادسازی کارها برابر با صفر است ($r_i = 0$)

1- ElMaraghy

2 - Bokhorst and Gaalman

- کلیه ماشین‌ها و کارگران از زمان صفر در دسترس هستند.
 - یک ماشین در یک لحظه زمانی تنها می‌تواند یک عملیات را انجام دهد.
 - کارگر می‌تواند از ماشینی به ماشینی دیگر منتقل شود اما در طول پردازش یک عملیات نمی‌تواند ماشین را ترک کند.
 - بریدگی کارها مجاز نیست.
 - هر کارگر می‌تواند روی بیشتر از یک ماشین کار کند و هر ماشین می‌تواند توسط کارگران متفاوت کنترل شود.
 - امکان انجام هر عملیات توسط ماشین‌ها و کارگران متفاوت وجود دارد، در حالی که زمان پردازش متفاوت و قطعی خواهد بود.
 - هر پردازش احتیاج به هر دو منبع ماشین و کارگر دارد.
 - زمان آماده‌سازی و زمان جابه‌جایی بین ماشین‌ها نادیده گرفته شده‌است.
 - زمان خرابی ماشین‌آلات و نگهداری و تعمیرات آن‌ها نادیده گرفته شده‌است.
- هدف از مسئله *DRCFJSP* تخصیص هر عملیات به یک ماشین از میان ماشین‌های موجود برای انجام آن عملیات، تخصیص هر عملیات به یک کارگر از میان کارگرهای قادر به انجام آن عملیات و مسئله دیگر تعیین توالی عملیات‌ها روی ماشین‌ها با توجه به کارگران در نظر گرفته شده به منظور بهینه‌سازی معیار حداکثر زمان تکمیل کارها می‌باشد. میزان انعطاف‌پذیری مسئله *DRCFJSP* ارتباط مستقیم با میزان انعطاف‌پذیری ماشین‌ها و کارگران دارد. این مسئله انعطاف‌پذیری کامل منابع را دارا است اگر و فقط اگر $S_k = W$ و $M_{i,j} = M$ باشد. در غیر این صورت انعطاف‌پذیری مسئله ناکامل خواهد بود.
- اطلاعات مربوط به یک مثال عددی (۴ کار، ۳ ماشین و ۲ کارگر) با انعطاف‌پذیری ناکامل در شکل ۱ نشان داده شده‌است. در این جدول علامت ∞ به معنی آن است که عملیات مربوطه نمی‌تواند توسط ترکیب ماشین و کارگر مورد نظر انجام شود.

جدول ۱- زمان پردازش عملیات ها

M_3		M_2		M_1		عملیات
W_2	W_1	W_2	W_1	W_2	W_1	
۸	۱۲	۴	۶	۹	۱۰	O_{11}
۱۴	۱۵	∞	∞	۱۲	۹	O_{12}
∞	∞	۱۴	۱۹	۱۴	۱۶	O_{21}
۸	۷	۱۲	۱۰	∞	∞	O_{22}
۶	۱۳	۵	۸	۶	۱۰	O_{31}
۱۵	۸	∞	∞	۸	۱۱	O_{32}
∞	∞	۱۲	۷	۱۰	۸	O_{33}
۴	۶	۷	۹	∞	∞	O_{41}
∞	∞	۱۷	۱۱	۱۳	۱۴	O_{42}
۸	۱۰	∞	∞	۷	۳	O_{43}

مدل‌سازی ریاضی

در این بخش یک مدل برنامه‌ریزی خطی عدد صحیح مختلط ($MILP$) برای مسئله $DRCFJSP$ ارائه شده است. نمادهای استفاده‌شده به شرح زیر است.

پارامترها

n : تعداد کارها؛ m : تعداد ماشین‌ها؛ h : تعداد کارگران؛ n_i : تعداد عملیات هر کار؛
 $M_{i,j}$: مجموعه ماشین‌هایی که توانایی ماشین‌پردازی عملیات $O_{i,j}$ را دارند؛ $W_{i,j}$:
 مجموعه کارگرانی که مهارت فنی انجام عملیات $O_{i,j}$ را دارند؛ $M_{i,j,l}$: مجموعه
 ماشین‌هایی که روی آن‌ها عملیات $O_{i,j}$ توسط کارگر l قابل انجام است؛ S_k : مجموعه
 کارگرانی که توانایی انجام کار روی ماشین k را دارند؛ $p_{i,j,k,l}$: زمان پردازش عملیات
 $O_{i,j}$ روی ماشین k توسط کارگر l .

شاخص‌ها

i, r : شاخص شماره کارها؛ j, s : شاخص شماره عملیات کارها؛ k : شاخص شماره ماشین‌ها؛ l : شاخص شماره کارگرها.

متغیرهای تصمیم

$X_{i,j,r,s}$: متغیر باینری که اگر عملیات $O_{i,j}$ بعد از عملیات $O_{r,s}$ انجام گیرد برابر با ۱ و در غیر این صورت برابر با صفر است؛

$Y_{i,j,k,l}$: متغیر باینری که اگر عملیات $O_{i,j}$ بر روی ماشین k ام توسط کارگر l ام انجام گیرد برابر با ۱ و در غیر این صورت برابر با صفر است؛ $C_{i,j}$: متغیر پیوسته برای زمان تکمیل عملیات کارها؛ C_{\max} : متغیر پیوسته برای حداکثر زمان تکمیل کارها.

با توجه به پارامترها، شاخص‌ها و متغیرهای تصمیم تعریف شده، فرمول‌بندی مسئله به شرح زیر است:

$$\text{Min } C_{\max}$$

Subject to:

$$\sum_{k \in M_{i,j}} \sum_{l \in S_k} Y_{i,j,k,l} = 1 \quad \forall i, j \quad (1)$$

$$C_{i,j} \geq C_{i,j-1} + \sum_{k \in M_{i,j}} \sum_{l \in S_k} Y_{i,j,k,l} \cdot p_{i,j,k,l} \quad \forall i, j \quad (2)$$

$$C_{i,j} \geq C_{r,s} + \sum_{l \in S_k} Y_{i,j,k,l} \cdot p_{i,j,k,l} - M \cdot (1 - x_{i,j,r,s}) - M \cdot (2 - \sum_{l \in S_k} Y_{i,j,k,l} - \sum_{l \in S_k} Y_{r,s,k,l}) \quad (3)$$

$$\forall i < n, j; r > i, s; k \in \{M_{i,j} \cap M_{r,s}\}$$

(۴)

$$C_{r,s} \geq C_{i,j} + \sum_{l \in S_k} Y_{r,s,k,l} \cdot p_{r,s,k,l} - M \cdot x_{i,j,r,s} - M \cdot (2 - \sum_{l \in S_k} Y_{i,j,k,l} - \sum_{l \in S_k} Y_{r,s,k,l})$$

$$\forall i < n, j; r > i, s; k \in \{M_{i,j} \cap M_{r,s}\}$$

(۵)

$$C_{i,j} \geq C_{r,s} + \sum_{k \in M_{i,j,l}} Y_{i,j,k,l} \cdot p_{i,j,k,l} - M \cdot (1 - x_{i,j,r,s}) - M \cdot (2 - \sum_{k \in M_{i,j,l}} Y_{i,j,k,l} - \sum_{k \in M_{r,s,l}} Y_{r,s,k,l})$$

$$\forall i < n, j; r > i, s; l \in S_k$$

(۶)

$$C_{r,s} \geq C_{i,j} + \sum_{k \in M_{i,j,l}} Y_{r,s,k,l} \cdot p_{r,s,k,l} - M \cdot x_{i,j,r,s} - M \cdot (2 - \sum_{k \in M_{i,j,l}} Y_{i,j,k,l} - \sum_{k \in M_{r,s,l}} Y_{r,s,k,l})$$

$$\forall i < n, j; r > i, s; l \in S_k$$

(۷)

$$C_{\max} \geq C_{i,n_i} \quad \forall i$$

(۸)

$$C_{i,j} \geq 0 \quad \forall i, j$$

(۹)

$$X_{i,j,r,s}, Y_{i,j,k,l} \in \{0,1\} \quad \forall i, j, r, s, k, l$$

$$C_{i,0} = 0$$

جایی که محدودیت ۱ مشخص می‌کند که هر عملیات توسط چه ماشین و چه کارگری پردازش شود. محدودیت ۲ زمان تکمیل عملیات یک کار را در ارتباط با عملیات قبلی همان کار مشخص می‌کند تا در یک زمان حداکثر یک عملیات از هر کار پردازش شود. محدودیت ۳ و ۴ ارتباط زمانی عملیات دو کار متفاوت را با هم مشخص می‌کند اگر دو

عملیات توسط یک ماشین پردازش شوند. در این صورت یک ماشین در یک زمان حداکثر یک عملیات را پردازش می‌کند. محدودیت ۵ و ۶ ارتباط زمانی عملیات دو کار متفاوت را با هم مشخص می‌کند اگر دو عملیات توسط یک کارگر پردازش شوند. در این صورت یک کارگر در یک زمان حداکثر یک عملیات را پردازش می‌کند. محدودیت ۷ زمان تکمیل هر کار را در مقایسه با حداکثر زمان تکمیل کارها کنترل می‌کند. محدودیت‌های ۸ و ۹ متغیرهای تصمیم را مشخص می‌کند.

الگوریتم ترکیبی برای مسئله *DRCFJSP*

پیچیدگی بالای مسئله *DRCFJSP* دلیلی آشکار و روشن برای استفاده از الگوریتم‌های فراابتکاری به منظور بهینه‌سازی آن است. در این بخش الگوریتم ترکیبی جستجوی همسایگی متغیر-شبه‌سازی تبرید (*VNS-SA*) برای بهینه‌سازی مسئله *DRCFJSP* ارائه شده است. این بخش به سه زیربخش تقسیم شده است. در زیربخش اول نحوه‌ی نمایش جواب را ارائه می‌کنیم. در زیربخش دوم ساختارهای همسایگی استفاده شده را مورد بررسی قرار می‌دهیم. در زیربخش سوم جزئیات مربوط به الگوریتم ترکیبی ارائه شده را بیان خواهیم کرد. همچنین در زیربخش چهارم نحوه تنظیم پارامترهای الگوریتم گزارش شده است.

نحوه نمایش جواب

همانطور که در بخش گذشته بیان شد هدف از مسئله *DRCFJSP* تصمیم‌گیری روی تخصیص عملیات به کارگران و ماشین‌ها و تعیین توالی عملیات روی ماشین‌ها است. بنابراین برای بیان یک جواب در مسئله *DRCFJSP* باید هر ۳ تصمیم موردنظر در نمایش جواب لحاظ شود. در این مقاله از نحوه‌ی نمایش جواب ارائه شده توسط کاظم و همکاران (۲۰۰۲) استفاده شده است و با تعدادی تغییرات برای نمایش جواب مسئله *DRCFJSP* به کار گرفته شده است. در این روش نمایش جواب از یک رشته اطلاعاتی بهره گرفته شده که

برای هر عملیات یک خانه و در هر خانه یک شاخص چهارتایی (i, j, k, l) را برای نمایش اطلاعات مربوط به آن عملیات در نظر گرفته‌است که در آن i نشان‌دهنده شماره کار، j نشان‌دهنده شماره عملیات، k نشان‌دهنده شماره ماشین و l نشان‌دهنده شماره کارگر است. طول رشته (تعداد خانه‌ها) برابر با مجموع کل تعداد عملیات کارها است. محل قرارگیری عملیات در داخل رشته جواب نشان‌دهنده ترتیب توالی انجام آن‌ها روی ماشین‌ها است. اطلاعات مربوط به مسئله جدول ۱ را در نظر بگیرید. میزان انعطاف‌پذیری این مسئله با توجه به نسبت تعداد خانه‌های فعال جدول (مخالف با ∞) به کل خانه‌های جدول برابر با $0/73$ (نسبت ۴۴ به ۶۰) به دست می‌آید. یکی از جواب‌های مربوط به این مسئله با توجه به نحوه نمایش تعریف شده در شکل ۱ نشان داده شده‌است. نمودار زمانی انجام عملیات مربوط به این جواب در شکل ۲ نشان داده شده‌است. حداکثر زمان تکمیل کارها معادل با ۴۰ است.

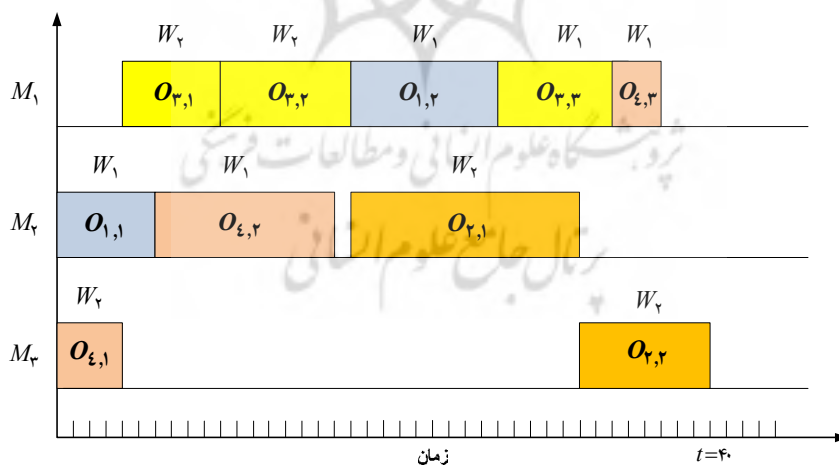
روابط پیش‌نیازی و وضعیت زمان انجام عملیات مختلف در جواب یک مسئله *DRCFJSP* را می‌توان با استفاده از یک شبکه گره‌ای و در قالب یک مدل گراف ترسیم کرد. گراف جهت‌دار مربوط به جواب نمایش داده در شکل ۱، در شکل ۳ ترسیم شده‌است. هر گره به معنای یک عملیات است. گره‌های مجازی E و K به ترتیب نقاط شروع و پایان می‌باشند. کمان‌هایی که با خطوط کامل مشخص شده‌اند، نشان‌دهنده روابط پیش‌نیازی بین عملیات مربوط به یک کار و کمان‌هایی که با خطوط نقطه‌چین مشخص شده‌اند، نشان‌دهنده روابط پیش‌نیازی میان عملیات هستند که به وسیله یک ماشین مشابه یا توسط یک کارگر یکسان اجرا می‌شوند. روابط پیش‌نیازی بین عملیات یک کار که توسط یک ماشین یا یک کارگر مشابه انجام می‌شوند، توسط خطوط کامل مشخص شده‌است (رابطه پیش‌نیازی بین عملیات O_{31} و O_{32}). مسیر بحرانی در این شکل با خطوط ضخیم‌تر مشخص شده‌است. مجموع زمان‌های انجام عملیات مسیر بحرانی برابر با مقدار حداکثر زمان تکمیل کارها در یک زمان‌بندی است. زمان‌بندی عملیات مربوط به این مسیر دارای زمان شناوری صفر است. کوچک‌ترین تأخیر در زمان انجام عملیات مسیر بحرانی باعث خواهد شد تا مقدار حداکثر زمان تکمیل کارها برای زمان‌بندی موردنظر افزایش یابد. جدول ۲ اطلاعات زمان‌بندی را با توجه به قواعد شبکه‌های گره‌ای برای گراف موردنظر گزارش کرده‌است.

(۴,۱,۳,۲)	(۳,۱,۱,۲)	(۱,۱,۲,۱)	(۳,۲,۱,۲)	(۴,۲,۲,۱)	(۱,۲,۱,۱)	(۳,۳,۱,۱)	(۲,۱,۲,۲)	(۲,۲,۳,۲)	(۴,۳,۱,۱)
-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------

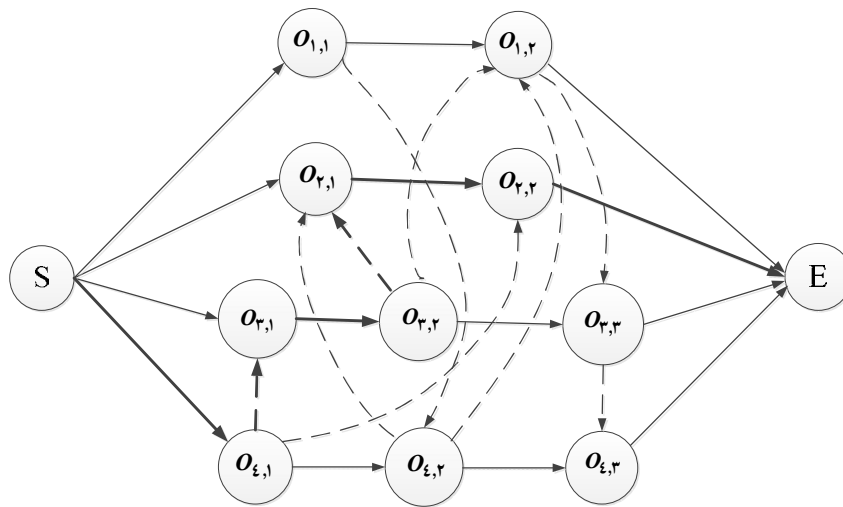
شکل ۱- نحوه نمایش جواب

جدول ۲- اطلاعات زمان بندی انجام عملیات

عملیات	زودترین زمان شروع	دیرترین زمان شروع	زودترین زمان پایان	دیرترین زمان پایان	زمان شناوری کل
$O_{1,1}$	۰	۱	۶	۷	۱
$O_{1,2}$	۱۸	۲۳	۲۷	۳۲	۵
$O_{۲,1}$	۱۸	۱۸	۳۲	۳۲	۰
$O_{۲,۲}$	۳۲	۳۲	۴۰	۴۰	۰
$O_{۳,1}$	۴	۴	۱۰	۱۰	۰
$O_{۳,۲}$	۱۰	۱۰	۱۸	۱۸	۰
$O_{۳,۳}$	۲۷	۳۲	۳۵	۴۰	۵
$O_{۴,1}$	۰	۳	۴	۷	۰
$O_{۴,۲}$	۶	۷	۱۷	۱۸	۷
$O_{۴,۳}$	۳۵	۳۷	۳۸	۴۰	۲



شکل ۲- نمودار زمانی انجام کارها



شکل ۳- گراف جهت دار جواب

ساختار همسایگی

هدف اصلی از کاربرد ساختار همسایگی در الگوریتم‌ها ایجاد جواب همسایگی از یک جواب موجود به وسیله انجام تغییراتی در آن می باشد (یزدانی و همکاران، ۲۰۱۰). ساختارهای همسایگی بسیاری در مسائل زمان‌بندی به کار گرفته شده‌اند. در این زیر بخش ۹ ساختار همسایگی ارائه شده برای مسئله مورد *DRCFJSP* مطالعه قرار خواهد گرفت. این ساختارهای همسایگی به گونه‌ای طراحی شده‌اند که از تولید جواب‌های غیرموجه جلوگیری می کنند.

ساختار همسایگی *SNS-1*

به منظور تغییر در توالی انجام عملیات جواب کاندید، ساختار همسایگی *SNS-1* به کار گرفته می شود به نحوی که تخصیص عملیات به ماشین‌ها و کارگران تغییر نکند. در این ساختار همسایگی با توجه به محدودیت‌های پیش‌نیازی تغییرات از طریق تعویض عملیات مجاور در جواب کاندید انجام می گیرد. ساختار همسایگی *SNS-1* می تواند با مقادیر

1 - Neighboring solution
2 - Yazdani

متفاوت تغییرات در جواب کاندید در مراحل مختلف یک الگوریتم معین به کار رود. تعداد تغییرات ایجادشده در جواب با علامت S_1 نمایش داده می شود.

ساختار همسایگی SNS-2

ساختار همسایگی SNS-2، برای تعویض عملیات مربوط به دو کار در جواب کاندید به کار می رود. در اجرای این ساختار تخصیص عملیات به ماشین ها و کارگران تغییری نمی کند.

ساختار همسایگی ANS-1

ساختار همسایگی ANS-1 برای تغییر در تخصیص عملیات به ماشین ها در جواب کاندید به کار می رود. به این صورت که کارگران تخصیص داده شده و توالی عملیات روی ماشین ها تغییر نمی کند. ساختار همسایگی ANS-1 می تواند با مقادیر متفاوت تغییرات در جواب کاندید در مراحل مختلف یک الگوریتم معین به کار رود. تعداد تغییرات ایجادشده در جواب با علامت A_1 نمایش داده می شود.

ساختار همسایگی ANS-2

ساختار همسایگی ANS-2 برای تغییر در تخصیص عملیات به کارگران در جواب کاندید به کار می رود به این صورت که ماشین های تخصیص داده شده و توالی عملیات روی ماشین ها تغییر نمی کند. ساختار همسایگی ANS-2 می تواند با مقادیر متفاوت تغییرات در جواب کاندید در مراحل مختلف یک الگوریتم معین به کار رود. تعداد تغییرات ایجادشده در جواب با علامت A_2 نمایش داده می شود.

ساختار همسایگی ترکیبی HNS

این ساختار همسایگی ترکیبی است از ساختارهای همسایگی SNS-1، ANS-1 و ANS-2. لذا به منظور تولید جواب همسایگی، تغییرات تصادفی بر روی رشته جواب کاندید

با استفاده از $SNS-1$ با مقدار S_1 تنظیم شده، $ANS-1$ با مقدار A_1 تنظیم شده و $ANS-2$ با مقدار A_2 تنظیم شده انجام می‌گیرد. جواب به دست آمده از اجرای این ۳ عملگر، به عنوان جواب همسایگی در نظر گرفته می‌شود.

ساختار همسایگی $INS-1$

ساختار همسایگی $INS-1$ بر اساس جهش هوشمند ارائه شده توسط پزلا و همکاران (۲۰۰۸) توسعه و تطبیق داده شده است. این ساختار همسایگی به این ترتیب عمل می‌کند: به منظور تولید جواب همسایگی از جواب کاندید توسط این ساختار همسایگی ابتدا ماشین‌های با بیشترین و کمترین حجم کاری در جواب کاندید مشخص می‌شوند. حجم کاری یک ماشین برابر با مجموع زمان پردازش عملیات انجام شده روی آن ماشین است. سپس یک عملیات به تصادف از ماشین با بیشترین حجم کاری انتخاب و به ماشین با کمترین حجم کاری تخصیص داده می‌شود و کارگر مربوطه حفظ می‌شود. این تخصیص تنها در صورتی ممکن است که ماشین با کمترین حجم کاری توانایی انجام این عملیات را توسط کارگر مربوطه داشته باشد.

ساختار همسایگی $INS-2$

این ساختار همسایگی به این ترتیب عمل می‌کند: به منظور تولید جواب همسایگی از جواب کاندید توسط این ساختار همسایگی ابتدا کارگرانی با بیشترین و کمترین حجم کاری در جواب کاندید مشخص می‌شوند. حجم کاری یک کارگر برابر با مجموع زمان پردازش عملیات انجام شده توسط آن کارگر است. سپس یک عملیات به تصادف از میان عملیات مربوط به کارگری با بیشترین حجم کاری انتخاب شده و به کارگری با کمترین حجم کاری تخصیص داده می‌شود و ماشین مربوطه حفظ می‌شود. این تخصیص تنها

در صورتی ممکن است که کارگری با کمترین حجم کاری توانایی انجام این عملیات را توسط ماشین مربوطه داشته باشد.

ساختار همسایگی *INS-3*

ساختار همسایگی *INS-3* به این ترتیب عمل می کند: به منظور تولید جواب همسایگی از جواب کاندید ابتدا ماشین های با بیشترین و کمترین زمان تکمیل کار در جواب کاندید مشخص می شوند. سپس یک عملیات به تصادف از ماشین با بیشینه زمان تکمیل کار انتخاب و به ماشین با کمینه زمان تکمیل کار تخصیص داده می شود.

ساختار همسایگی *INS-4*

ساختار همسایگی *INS-4* براساس جهش هوشمند ارائه شده توسط گائو و همکاران (۲۰۰۸) توسعه و تطبیق داده شده است. در این ساختار همسایگی به دنبال آن هستیم که با تغییرات هوشمند در مسیر بحرانی یک جواب، به زمان بندی با مقدار حداکثر زمان تکمیل کارهای کمتر دست یابیم. برای شرح این ساختار همسایگی پارامترهای زیر را در نظر بگیرید:

$C^E(r)$: زودترین زمان تکمیل یک عملیات؛ $S^L(r)$: دیرترین زمان شروع یک عملیات؛ $PJ(r)$: عملیات قبلی مربوط به عملیات r که هر دو مربوط به یک کار هستند؛ $SJ(r)$: عملیات بعدی مربوط به عملیات r که هر دو مربوط به یک کار هستند؛ $PM(r)$: عملیات قبلی مربوط به عملیات r که هر دو روی یک ماشین اجرا می شوند؛ $PW(r)$: عملیات قبلی مربوط به عملیات r که هر دو توسط یک کارگر اجرا می شوند؛ t_j^S : زمان شروع انجام عملیات روی ماشین j ام

t_j^E : زمان تکمیل انجام عملیات روی ماشین j ام؛ t_j^S : زمان شروع انجام عملیات توسط کارگر j ام؛ t_j^E : زمان تکمیل انجام عملیات توسط کارگر j ام.

ساختار همسایگی $INS-4$ به این ترتیب عمل می کند: به منظور تولید جواب همسایگی از جواب کاندید G ابتدا گراف جهت دار آن را رسم و مسیر بحرانی آن را مشخص می کنیم. سپس عملیات r را به تصادف از میان مسیر بحرانی انتخاب می کنیم و این عملیات را از جایگاه فعلی اش در مسیر بحرانی حذف می کنیم. جواب \bar{G} را به عنوان جوابی در نظر می گیریم که عملیات r از آن حذف شده است. علامت (-) نشان دهنده موقعیتی است که عملیات r از جواب کاندید حذف شده است. در مرحله بعد با توجه به جواب \bar{G} می بایست عملیات r را در بازه زمانی مناسب دیگری به ماشین l و به کارگر l تخصیص و انجام دهیم به طوری که مقدار حداکثر زمان تکمیل کارهای جواب جدید ایجاد شده نسبت به حداکثر زمان تکمیل کارهای جواب کاندید G مساوی یا کمتر باشد. در صورت پیدانکردن بازه زمانی مناسب برای عملیات r ، عملیات دیگری را از بین عملیات مسیر بحرانی انتخاب می کنیم. انتخاب بازه زمانی جدید شروع و پایان انجام عملیات r روی ماشین l توسط کارگر l می بایست با یک رویکرد زمان بندی فعال و با توجه به محدودیت های پیش نیازی عملیات انتخاب شود. لذا عملیات r روی ماشین l توسط کارگر l می بایست بعد از زودترین زمان تکمیل عملیات $PJ(r)$ یعنی $c^{E-}(PJ(r))$ شروع شود. و قبل از دیرترین زمان شروع عملیات $SJ(r)$ یعنی $s^{L-}(SJ(r))$ تکمیل شود. بنابراین برای تخصیص عملیات r در یک بازه زمانی منطقی رابطه زیر را خواهیم داشت:

(۱۰)

$$\max\{t_j^S, t_l^S, c^{E-}[PJ(r)]\} + p_{rjl} < \min\{t_j^E, t_l^E, s^{L-}[SJ(r)]\}$$

جایی که p_{rjl} برابر با زمان انجام عملیات r روی ماشین l توسط کارگر l است. اگر بخواهیم عملیات r و v را بر روی یک ماشین و عملیات r را قبل از عملیات v در زمان بندی وارد کنیم، برای بازه زمانی انتخابی رابطه زیر را خواهیم داشت:

(۱۱)

$$\max\{c^{E-}[PM(v)], c^{E-}[PJ(r)]\} + p_{rjl} < \min\{s^{L-}(v), s^{L-}[SJ(r)]\}$$

جایی که $c^{E-}[PM(v)]$ برابر با زودترین زمان تکمیل عملیات $PM(v)$ ، $s^{L-}(v)$ برابر با

زودترین زمان شروع عملیات v و p_{ijl} برابر با زمان انجام عملیات r روی ماشین l ام توسط کارگر l ام است. در حالتی دیگر اگر بخواهیم عملیات r و v را توسط یک کارگر و عملیات r را قبل از عملیات v وارد کنیم برای بازه زمانی انتخابی رابطه زیر را خواهیم داشت:

(۱۲)

$$\max\{c^{E-}[PW(v)], c^{E-}[PJ(r)]\} + p_{ijl} < \min\{s^{L-}(v), s^{L-}[SJ(r)]\}$$

جایی که $c^{E-}[PW(v)]$ برابر با زودترین زمان تکمیل عملیات $PW(v)$ است. همچنین در انتخاب بازه انجام عملیات r می‌بایست نکات زیر را رعایت کنیم:

- نمی‌توان عملیات r را قبل از عملیاتی که زودترین زمان تکمیل اش کمتر از $c^{E-}(PJ(r))$ است، وارد کرد.
- نمی‌توان عملیات r را بعد از عملیاتی که دیرترین زمان شروعش بیشتر $s^{L-}(SJ(r))$ است، وارد کرد.

به عبارت کامل‌تر نمی‌توان بازه انجام عملیات r را قبل از عملیاتی که زودترین زمان تکمیل‌شان کمتر از $c^{E-}(PJ(r))$ و یا بعد از عملیاتی که دیرترین زمان شروع‌شان بیشتر از $s^{L-}(SJ(r))$ است، تعیین کرد. ذکر این نکته لازم است که این ساختار همسایگی، قصد دارد با شکست مسیر بحرانی جواب‌کاندید و جابه‌جایی مؤثر یک عملیات در این مسیر، باعث ایجاد بهبود در مقدار هدف حداکثر زمان تکمیل کارها شود.

شرح الگوریتم

الگوریتم ترکیبی موردنظر از ترکیب دو الگوریتم VNS و SA بهره خواهد برد. الگوریتم VNS یکی از الگوریتم‌های فراالبتکاری بر پایه تغییرات سیستماتیک ساختار همسایگی است که برای جستجوی جواب بهینه در فضای مسائل بهینه‌سازی ترکیباتی به کار می‌رود (ملاذنیچ و هنسن، ۱۹۹۷؛ هنسن و ملاذنیچ، ۲۰۰۱). تغییرات سیستماتیک از طریق تغییر موقعیت از یک همسایگی به همسایگی دیگر جواب در حین جستجوی فضای جواب و به منظور فرار از نقطه

بهینه موضعی انجام می‌گیرد. ملادنویچ (۱۹۹۵) اولین کسی بود که الگوریتم جستجوی *VNS* را مطرح کرد. همچنین الگوریتم *SA* یکی از رویکردهای جستجوی تصادفی است که به دلیل پذیرش حرکت‌های غیر بهبوددهنده تابع هدف، برخلاف سایر روش‌های جستجوی موضعی به نقطه شروع (حل اولیه) وابسته نبوده و می‌تواند از دام بهینه‌های موضعی تا حد زیادی رهایی یابد. اساس این الگوریتم توسط متروپولیس و همکاران (۱۹۵۳) منتشر شد. آن‌ها یک الگوریتم برای شبیه‌سازی سرد شدن مواد در یک حمام گرما ارائه نمودند. اگر یک ماده جامد تا نقطه ذوب‌اش گرم شود و سپس تا حالت رسیدن به حالت جامد سرد شود، خواص ساختاری، پایداری و سطح انرژی ذرات جامد سرد شده، به سرعت سرد شدن بستگی دارد. الگوریتم ارائه‌شده تغییر در انرژی سیستم را تا زمانی که به یک حالت پایدار منجمد برسد، شبیه‌سازی می‌کرد. سی سال بعد کیکپاتریک و همکاران (۱۹۸۳) پیشنهاد کردند که این نوع شبیه‌سازی می‌تواند با عنوان الگوریتم *SA* برای جستجوی جواب‌های موجه یک مسئله بهینه‌سازی با هدف هم‌گرایی به یک جواب بهینه به کار رود.

شبه‌کد الگوریتم ترکیبی *VNS-SA* ارائه‌شده برای مسئله *DRCFJSP* در شکل ۴ نمایش داده شده است. همانطور که در شکل مشخص است این الگوریتم از دو بخش تفکیک‌شده *VNS* و *SA* تشکیل شده است. الگوریتم *VNS-SA* در هر تکرار از بخش *VNS* شروع می‌کند. دو دسته از ساختارهای همسایگی برای بخش *VNS* استفاده شده‌اند که عبارت‌اند از N_k^s و N_l^{ls} . ساختارهای همسایگی N_k^s در فرآیند ارتعاش به کار گرفته می‌شوند. جدول ۳ ترتیب ساختارهای همسایگی استفاده‌شده در فرآیند ارتعاش را نشان می‌دهد. تمامی ساختارهای همسایگی استفاده‌شده در این مرحله از نوع ساختار همسایگی ترکیبی است. اما تعداد تغییرات روی جواب کاندید با توجه به افزایش k (شمارنده تکرار حلقه ارتعاش بخش *VNS*) افزایش می‌یابد. همچنین ساختارهای همسایگی N_l^{ls} در فرآیند جستجوی محلی به کار گرفته می‌شوند. جدول ۴ ترتیب ساختارهای همسایگی استفاده‌شده

-
- 1 - Metropolis
 - 2 - Kirkpatrick
 - 3 - Shaking procedure
 - 4 - Local search

در فرآیند جستجوی محلی را نشان می‌دهد. تعداد تکرارهای فرآیند جستجوی محلی بخش VNS برابر با ۵۰۰ تنظیم شده‌است.

جدول ۳- ساختارهای همسایگی به‌کار گرفته‌شده در فرآیند ارتعاش

مرحله	$k=1$	$k=2$	$k=3$	$k=4$
ساختار همسایگی	$A_1 = 1$ $HNS \quad A_2 = 1$ $S_1 = 1$	$A_1 = 2$ $HNS \quad A_2 = 2$ $S_1 = 2$	$A_1 = 3$ $HNS \quad A_2 = 3$ $S_1 = 3$	$A_1 = 4$ $HNS \quad A_2 = 4$ $S_1 = 4$

جدول ۴- ساختارهای همسایگی به‌کار گرفته‌شده در فرآیند جستجوی محلی

مرحله	$l=1$	$l=2$	$l=3$	$l=4$
ساختار همسایگی	$ANS-1 \quad A_1 = 1$	$ANS-2 \quad A_2 = 1$	$SNS-1 \quad S_1 = 2$	$SNS-2$
مرحله	$l=5$	$l=6$	$l=7$	$l=8$
ساختار همسایگی	$ANS-1 \quad A_1 = 2$	$ANS-2 \quad A_2 = 2$	$SNS-1 \quad S_1 = 4$	$INS-4$
مرحله	$l=9$	$l=10$	$l=11$	$l=12$
ساختار همسایگی	$INS-1$	$INS-2$	$INS-3$	$INS-4$

اگر در هر تکرار از حلقه ارتعاش و پس از طی فرآیند جستجوی محلی به جوابی بهتر از جواب فعلی (\bar{x}) دست یابیم، جواب به‌دست آمده جایگزین جواب فعلی خواهد شد، حلقه تکرار فرآیند ارتعاش قطع و تکرار دیگری از الگوریتم شروع خواهد شد. اما اگر در کل تکرارهای حلقه ارتعاش بخش VNS به جوابی بهتر از جواب فعلی دست نیابیم، آن‌گاه بدون تغییر شمارنده تکرار الگوریتم، جواب فعلی به‌عنوان جواب ورودی بخش SA قرار خواهد گرفت و جستجوی روی جواب‌ها توسط الگوریتم SA در همان تکرار دنبال خواهد شد. الگوریتم ارائه‌شده از یک تابع هندسی به‌منظور کاهش دمای بخش SA استفاده کرده‌است. تابع موردنظر به‌صورت زیر است:

$$T_{i+1} = \alpha T_i \quad (13)$$

جایی که α پارامتر کاهش دهنده دما می‌باشد. پس از بررسی‌های انجام‌شده و آزمایشات عددی در مسائل مختلف، مقدار ضریب α برابر با $0/9$ تعیین شده‌است. یعنی در هر مرحله درجه حرارت به میزان ۱۰ درصد کاهش پیدا می‌کند. دمای اولیه الگوریتم SA (T_0) همواره باید در یک حد متوسط و کافی تعیین‌شود تا جستجو در فضای جواب به شکل مناسبی حاصل شود. پارامتر دما برای بخش SA الگوریتم پیشنهادی برای هر مسئله متفاوت است و قبل از اجرای الگوریتم می‌بایست تنظیم شود. برای دستیابی به دمای اولیه، در ابتدا یک جواب موجه را به صورت تصادفی با توجه به مسئله ایجاد می‌کنیم و سپس توسط ساختارهای همسایگی موجود جواب همسایگی را از جواب موجود ایجاد می‌کنیم. این عمل را ۵۰۰ بار به صورت متوالی انجام می‌دهیم و میزان تغییرات در تابع هدف به‌ازای دو جواب متوالی را محاسبه می‌کنیم. سپس محاسبات زیر را انجام می‌دهیم:

$$T_0 = \left[\overline{\Delta_f} \right] \quad (14)$$

جایی که $\overline{\Delta_f}$ برابر با میانگین تغییرات مثبت و منفی به‌وجودآمده در توابع هدف جواب‌های کاندید پس از تولید جواب‌های همسایگی (تغییرات جواب‌هایی که مقدار تابع هدف آن‌ها بدتر و بهتر شده‌است در محاسبات وارد خواهد شد و تغییراتی با مقادیر صفر در محاسبه دمای اولیه وارد نخواهد شد) در ۵۰۰ تکرار می‌باشد و علامت $[]$ نشان‌دهنده مقدار جزء صحیح میانگین به‌دست‌آمده می‌باشد. همانطور که در بالا ذکر شد برای کاربرد الگوریتم برای هر مسئله خاص می‌بایست یک دمای اولیه را محاسبه کرد. درجه حرارت نهایی برابر با $0/1$ تعیین شده‌است. در هر دما از بخش SA جستجو در فضای جواب با توجه به ساختارهای همسایگی موجود صورت می‌گیرد. ترتیب ساختارهای همسایگی استفاده‌شده مشابه فرآیند جستجوی محلی بخش VNS است. تعداد جستجو در هر دما از بخش SA برابر با ۱۰۰ تعیین شده‌است. در انتهای بخش SA ، جواب به‌دست‌آمده از این بخش به‌عنوان جواب ورودی تکرار بعدی الگوریتم قرار خواهد گرفت. الگوریتم $VNS-SA$ تا زمانی ادامه

می‌یابد که شرط توقف الگوریتم ارضا شود. شرط توقف الگوریتم براساس زمان و برابر با زمان محاسباتی مورد نیاز تعیین شده است که با توجه به ابعاد هر مسئله تعیین خواهد شد.

Initialization

- Select a set of neighborhood structures that will be used in VNS and SA parts of hybrid algorithm, number of neighborhood search in each iteration of VNS local search (n_{l-VNS}), initial temperature of SA (T_0), function of reduce temperature, number of neighborhood search in each temperature of SA (n_{l-SA}).
- Choose stopping condition (maximum number of iterations (n_{max})).
- Generate randomly a population of solutions and the best solution among them (x) is selected for initial solution.
- Set initial solution as current solution $\bar{x} \leftarrow x$.

Repeat

VNS part of algorithm:

for $k = 1: k_{max}$ ($k_{max} = 4$)

Shaking:

Generate random point $x_{VNS} \in N_k^i(\bar{x})$ with N_k^i ;

Local search:

Get solution x_{VNS} ;

Set $l \leftarrow 1$

for $i = 1: n_{l-VNS}$

Generate random point $x'_{VNS} \in N_l^i(x_{VNS})$ with N_l^i ;

If $f(x'_{VNS}) < f(x_{VNS})$ then $x_{VNS} \leftarrow x'_{VNS}$ and $l \leftarrow l + 1$;

else select the random integer number L in the range $[1, l_2]$ and set $l \leftarrow L$;

endif

endfor

Updating:

If $f(x_{VNS}) < f(\bar{x})$ then $\bar{x} \leftarrow x_{VNS}$ break loop and move to next iteration of hybrid algorithm;

else $k \leftarrow k + 1$;

end if

endfor

SA part of algorithm:

Set $T \leftarrow T_0$; $x_{SA} \leftarrow \bar{x}$ and $x_{SA}^* \leftarrow \bar{x}$;

While $T > T_f$

Set $l \leftarrow 1$

for $j = 1: n_{l-SA}$

Generate random point $x'_{SA} \in N_l^j(x_{SA})$ with N_l^j ;

if $\Delta_{x_{SA}, x'_{SA}} < 0$ then $x_{SA} \leftarrow x'_{SA}$; $x_{SA}^* \leftarrow x'_{SA}$ and $l \leftarrow l + 1$;

elseif $\Delta_{x_{SA}, x'_{SA}} = 0$ then generate random number R uniformly in the range $(0,1)$:

if $R \leq 0.5$ then $x_{SA} \leftarrow x'_{SA}$; $x_{SA}^* \leftarrow x'_{SA}$ and $l \leftarrow l + 1$;

endif

elseif $\Delta_{x_{SA}, x'_{SA}} > 0$ then generate random number R uniformly in the range $(0,1)$:

if $R \leq \exp(-\Delta_{x_{SA}, x'_{SA}} / T)$ then $x_{SA} \leftarrow x'_{SA}$ and $l \leftarrow l + 1$;

endif

else $l \leftarrow l + 1$;

endif

if $l > 12$ then $l \leftarrow 1$;

endif

endfor

Update T : Decrease of temperature according to linear low;

endwhile

if $f(x_{SA}^*) \leq f(\bar{x})$ then $\bar{x} \leftarrow x_{SA}^*$;

endif

Until the stopping condition is met.

شکل ۴- شبه کد الگوریتم VNS-SA

ارزیابی عددی

در این بخش نتایج محاسباتی به دست آمده از اجرای الگوریتم *VNS-SA* به منظور بهینه سازی مسائل *DRCFJSP* مورد ارزیابی و مطالعه قرار گرفته است. برای برنامه نویسی الگوریتم از نرم افزار *MATLAB-2012B* بهره گرفته ایم. همچنین برای اجرای الگوریتم از یک کامپیوتر با مشخصات ۳/۲ گیگاهرتز (*Core i7*) و کارت حافظه ۴ گیگابایت استفاده نموده ایم. کمینه سازی حداکثر زمان تکمیل کارها به عنوان تابع هدف در نظر گرفته شده است. برای انجام مطالعات محاسباتی ۲۰ مسئله را در ابعاد کوچک، متوسط و بزرگ ایجاد کرده ایم. زمان های پردازش مربوط به مسائل ایجاد شده به طور تصادفی در بازه [۱،۹۹] تعیین شده است. همچنین مجموعه های $M_{i,j}$ و S_k برای مسائل مورد نظر به شکل تصادفی انتخاب شده است. نتایج حاصل از اجرای الگوریتم برای ۲۰ مسئله ایجاد شده در جدول ۵ نمایش داده شده است. ستون اول تا پنجم مربوط به اطلاعات مسائل است جایی که n نشان دهنده تعداد کارها، m نماد تعداد ماشین ها، w نشان دهنده تعداد کارگران و TO بیانگر مجموع تعداد عملیات است. مسائل دارای انعطاف پذیری کامل با علامت ستاره (*) مشخص شده اند. در این مقاله برای به دست آوردن حد پایین جواب مسائل ایجاد شده، رابطه ارائه شده توسط لی و جیوئو (۲۰۱۴) به عنوان الگو در نظر گرفته شده است و با انجام تعدادی تغییرات و اضافه کردن دو تابع جدید و مؤثر دیگر، به صورت یک قانون حد پایین جدید مورد استفاده قرار گرفته است. برای به دست آوردن حد پایین جواب یک مسئله ابتدا می بایست زودترین زمان شروع هر عملیات را به دست آوریم که به صورت زیر به دست می آید:

$$r_{i(j+1)} = r_{ij} + \tilde{t}_{ij} \quad \forall 1 \leq i \leq n; 1 \leq j \leq n_i - 1 \quad (15)$$

جایی که $r_{i1} = r_i$ و $\tilde{t}_{ij} = \min_{k,l} \{p_{ijkl}\}$ در این مقاله مقدار r_j (زمان آزادسازی اولین عملیات هر کار) برای تمامی کارها صفر در نظر گرفته شده است. سپس مقادیر r_{ij} به دست آمده برای کلیه عملیات را به ترتیب صعودی مرتب می کنیم و در

مجموعه $R = \{r_{i_1 j_1}, r_{i_2 j_2}, \dots, r_{i_N j_N}\}$ قرار می‌دهیم (N : مجموع تعداد کل عملیات). سپس مقدار حد پایین را از رابطه زیر به دست می‌آوریم:

(۱۶)

$$C_{\max}^{low} = \max \left(\max_i (r_i + \sum_j \tilde{t}_{ij}), \tilde{E} \left(\frac{R_M + \sum_i \sum_j \tilde{t}_{ij}}{m} \right), \tilde{E} \left(\frac{R_W + \sum_i \sum_j \tilde{t}_{ij}}{h} \right), T_{k_0, \tilde{m}}, T_{l_0, \tilde{h}} \right)$$

جایی که $\tilde{E}(X)$ تابعی است که اگر X عدد صحیح باشد $\tilde{E}(X) = X$ و در غیر این صورت $\tilde{E}(X) = X + 1$ ؛ m برابر با تعداد ماشین‌ها ($k = \{1, 2, \dots, m\}$)؛ h برابر است با تعداد کارگران ($l = \{1, 2, \dots, l\}$)؛ R_M برابر با مجموع m عدد از زودترین زمان‌های شروع عملیات مطابق با مجموعه R ($R_M = \sum_{l=1}^m r_{i_l j_l}$)؛ R_W برابر با مجموع h عدد از زودترین زمان‌های شروع عملیات مطابق با مجموعه R ($R_W = \sum_{l=1}^h r_{i_l j_l}$)؛ $\tilde{m} = \tilde{E}(N_l / m)$ ؛ $\tilde{h} = \tilde{E}(N_l / h)$ ؛ همچنین

$$T_{k_0, \tilde{m}} = \min_k [\min_{1 \leq y \leq N - \tilde{m} + 1} (r_{i_y j_y} + \min_l (p_{i_y, j_y, k, l})) + \min_{C'_{z, \tilde{m}} \in E'_{z, \tilde{m}}} [S^k(C'_{y, \tilde{m}})]]$$

؛ $S^k(C'_{y, \tilde{m}})$ برابر است با مجموع کمینه زمان‌های پردازش عملیات مجموعه $C'_{y, \tilde{m}}$ روی ماشین k ؛ $C'_{y, \tilde{m}}$ یک بخش از مجموعه $E'_{y, \tilde{m}}$ ؛ $E'_{y, \tilde{m}}$ مجموعه‌ای متشکل از تمام ترکیبات ممکن $\tilde{m} - 1$ عملیاتی از میان $N - y$ عملیات مجموعه V_y ؛ مجموعه V_y برابر است با مجموعه مرتب‌شده عملیات از نظر زودترین زمان شروع از رتبه $y + 1$ تا

$$N(V_y = \{O_{i_{y+1}, j_{y+1}}, O_{i_{y+2}, j_{y+2}}, \dots, O_{i_N, j_N}\})$$

در دیگر روابط

$$T_{l_0, \tilde{h}} = \min_l [\min_{1 \leq z \leq N - \tilde{h} + 1} (r_{i_z j_z} + \min_k (p_{i_z, j_z, k, l})) + \min_{C'_{z, \tilde{h}} \in E'_{z, \tilde{h}}} [S^l(C'_{z, \tilde{h}})]]$$

؛ $S^l(C'_{z, \tilde{h}})$ برابر است با مجموع کمینه زمان‌های پردازش عملیات مجموعه $C'_{z, \tilde{h}}$ توسط کارگر l ؛ $C'_{z, \tilde{h}}$ یک بخش از مجموعه $E'_{z, \tilde{h}}$ ؛ $E'_{z, \tilde{h}}$ مجموعه‌ای متشکل از تمام ترکیبات ممکن $\tilde{h} - 1$ عملیاتی از میان $N - z$ عملیات مجموعه V_z ؛ مجموعه V_z برابر است با مجموعه مرتب‌شده عملیات از نظر زودترین زمان شروع از رتبه $z + 1$ تا

$$N(V_z = \{O_{i_{z+1}, j_{z+1}}, O_{i_{z+2}, j_{z+2}}, \dots, O_{i_N, j_N}\})$$

رتبه

مقادیر حد پایین به دست آمده برای مسائل در ستون ششم قرار داده شده است. با توجه به خاصیت تصادفی الگوریتم‌های ابتکاری، برای دستیابی بهترین جواب هر مسئله، الگوریتم توسعه داده شده را ۱۰ بار برای هر مسئله اجرا می‌کنیم. سپس بهترین جواب به دست آمده از این اجراها را به عنوان بهترین جواب الگوریتم در نظر می‌گیریم. بهترین جواب به دست آمده از الگوریتم *VNS-SA* و همچنین میانگین جواب‌ها به ترتیب در ستون هفتم و هشتم قرار داده شده است. همچنین برای نشان دادن کارایی الگوریتم ترکیبی، نتایج دو الگوریتم مستقل *VNS* و *SA* را با الگوریتم *VNS-SA* مقایسه کرده‌ایم. نتایج دو الگوریتم *VNS* و *SA* در ستون‌های ۹ تا ۱۲ قرار داده شده است. (لازم به ذکر است دو الگوریتم *VNS* و *SA* همان الگوریتم‌های بخش *SA* و بخش *VNS* الگوریتم ترکیبی ارائه شده می‌باشند که در این قسمت به صورت مستقل و تفکیک شده با الگوریتم ترکیبی *VNS-SA* مورد مقایسه قرار گرفته‌اند). نتایج نشان می‌دهد که الگوریتم *VNS-SA* به نتایج بهتری نسبت به دو الگوریتم دیگر بخصوص در مسائل بزرگ دست یافته است. در این مقاله برای مقایسات دقیق الگوریتم‌های موجود از یک معیار منطقی با عنوان معیار درصد انحراف نسبی (*RPD*) بهره گرفته‌ایم (رویز و اله وردی، ۲۰۰۷) این معیار به صورت زیر به دست می‌آید:

$$RPD = \frac{A \lg_{sol} - LB}{LB} \times 100\% \quad (17)$$

جایی که $A \lg_{sol}$ نشان‌دهنده بهترین جواب به دست آمده از الگوریتم مورد نظر و LB نشان‌دهنده حد پایین جواب‌های الگوریتم است. مقادیر میانگین درصد انحراف نسبی (*RPD*) برای الگوریتم‌های *SA*، *VDO* و *VNS-SA* به ترتیب برابر با ۲۵/۸۷، ۱۶/۷۵ و ۱۰/۲۵ به دست آمده است. نتایج بیانگر این است که بهترین مقدار (*RPD*) متعلق به الگوریتم *VNS-SA* است. همچنین به منظور اثبات آماری نتایج، آزمون تحلیل واریانس با توجه به مقادیر *RPD* با استفاده از نرم‌افزار *MINITAB* صورت گرفته است. نتایج این آزمون تفاوت معنی‌دار را بین مقادیر میانگین *RPD* سه الگوریتم *SA*، *VNS* و *VNS-SA*

1 - Relative percentage deviation

2 - Ruiz and Allahverdi

نشان می‌دهد. نمودار فاصله اطمینان ۹۵٪ برای مقادیر *RPD* در شکل ۵ ترسیم شده است. نتایج محاسباتی نشان‌دهنده این است که ترکیب دو الگوریتم *VNS* و *SA* و ارائه الگوریتم *VNS-SA* در بهبود کیفیت جواب‌ها مؤثر بوده است.

نتیجه‌گیری و تحقیقات آتی

در این مقاله مسئله *DRCFJSP* با هدف کمینه‌سازی معیار بیشینه زمان تکمیل کارها مورد مطالعه قرار گرفت و مدل ریاضی مسئله اشاره شده برای اولین بار ارائه شد. همچنین الگوریتم ترکیبی *VNS-SA* برای حل مسئله *DRCFJSP* توسعه داده شد. الگوریتم مورد نظر از ترکیب دو الگوریتم *VNS* و *SA* استفاده می‌کرد به این صورت که اگر جستجو در بخش *VNS* مؤثر واقع نمی‌شد، جستجو را به سمت بخش *SA* هدایت می‌کرد. جهت جستجوی هر چه بهتر فضای حل مسئله، یک روش جدید برای نمایش جواب این مسئله ارائه شد. همچنین ۹ ساختار همسایگی با توجه به ویژگی‌های مسئله برای ایجاد جواب‌های همسایگی مورد استفاده قرار گرفت. در بخش مطالعات محاسباتی روشی کارا برای دستیابی به حد پایین جواب‌های مسائل ایجاد شده مورد استفاده قرار گرفت. در ادامه نتایج الگوریتم *VNS-SA* توسعه داده شده با نتایج الگوریتم‌های *VNS* و *SA* مقایسه شد. نتایج به دست آمده عملکرد مؤثر و کارای الگوریتم ترکیبی ارائه شده را در حل مسائل *DRCFJSP* نشان می‌داد. به منظور تحقیقات آتی پیشنهادات زیر ارائه می‌گردد:

- در نظر گرفتن فرض زمان‌های آماده‌سازی وابسته به توالی در مسئله *DRCFJSP*
- در نظر گرفتن محدودیت‌های فرایندی نظیر خرابی ماشین‌آلات و تعمیرات و نگهداری در

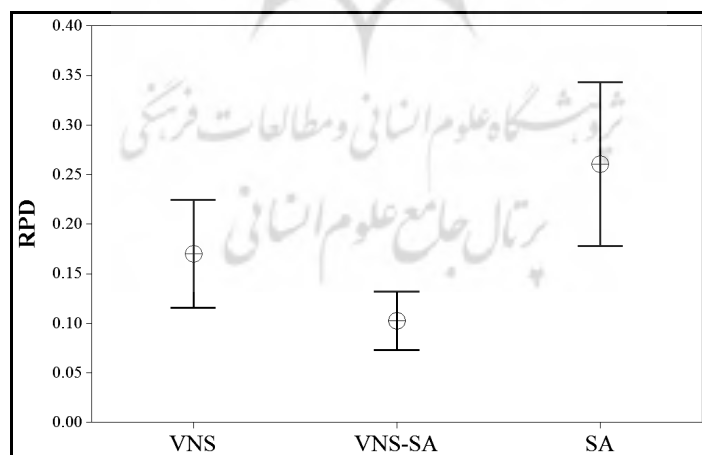
مسئله *DRCFJSP*

- توسعه مدل‌های ریاضی و الگوریتم‌های فراابتکاری برای مسئله *DRCFJSP* در حالت

چندهدفه

جدول ۵- نتایج الگوریتم‌ها

شماره مسئله	اطلاعات مسئله				حد پایین	نتایج VNS-SA		نتایج VNS		نتایج SA	
	n	m	w	TO		بهترین جواب	میانگین جواب‌ها	بهترین جواب	میانگین جواب‌ها	بهترین جواب	میانگین جواب‌ها
	P 1	۳	۲	۲		۸	۱۵۵	۱۶۷	۱۶۷	۱۶۷	۱۶۷
P 2	۴	۳	۲	۱۰	۱۷۴	۱۸۶	۱۸۶	۱۸۶	۱۸۶	۱۸۶	۱۸۶
⊗P 3	۴	۳	۲	۱۲	۲۷۱	۲۷۳	۲۷۳	۲۷۳	۲۷۳	۲۷۳	۲۷۳
P 4	۶	۴	۲	۱۸	۳۵۲	۳۵۷	۳۵۷	۳۵۷	۳۵۷	۳۵۷	۳۵۷/۲
P 5	۶	۴	۳	۲۵	۴۳۸	۴۶۳	۴۶۳	۴۶۳	۴۶۳/۵	۴۶۶	۴۶۶
⊗P 6	۸	۴	۳	۳۵	۵۱۴	۵۴۳	۵۴۳	۵۴۹	۵۵۲/۲	۵۶۴	۵۶۶/۳
P 7	۸	۴	۴	۴۰	۴۵۵	۴۹۵	۵۰۳/۲	۴۹۵	۵۰۷/۳	۵۳۳	۵۳۵/۵
P 8	۱۰	۵	۳	۴۵	۷۳۲	۷۷۳	۷۸۸/۴	۷۸۴	۷۹۴/۸	۸۲۹	۸۴۲/۵
⊗P 9	۱۰	۵	۴	۵۰	۵۸۵	۶۳۶	۶۳۹/۲	۶۶۳	۶۶۷/۵	۷۱۷	۷۱۷
P 10	۱۰	۵	۴	۵۵	۶۷۰	۷۱۸	۷۲۵/۴	۷۵۹	۷۷۴/۷	۸۴۱	۸۵۲/۶
⊗P 11	۱۲	۵	۴	۶۰	۷۲۲	۷۹۶	۸۰۵/۵	۸۸۱	۸۸۶/۱	۹۳۷	۹۴۵/۲
P 12	۱۲	۶	۳	۷۰	۱۰۶۲	۱۱۳۳	۱۱۴۱/۴	۱۱۹۶	۱۲۱۱/۶	۱۲۸۷	۱۲۸۷
⊗P 13	۱۲	۶	۵	۸۰	۸۵۸	۹۶۹	۹۸۰/۵	۱۰۶۹	۱۰۹۲/۹	۱۱۹۶	۱۲۰۳/۲
P 14	۱۲	۶	۶	۹۰	۷۵۹	۹۶۶	۹۸۵/۳	۱۱۰۹	۱۱۳۰/۶	۱۲۱۶	۱۲۳۶/۵
⊗P 15	۱۵	۸	۵	۱۰۰	۹۸۱	۱۰۸۸	۱۱۱۹/۵	۱۲۳۲	۱۲۵۸/۴	۱۳۱۸	۱۳۲۸/۴
P 16	۱۵	۸	۶	۱۰۰	۷۶۲	۹۰۷	۹۰۷	۹۹۵	۹۹۵	۱۰۹۴	۱۱۲۷/۲
⊗P 17	۲۰	۸	۶	۱۲۰	۸۷۳	۹۱۷	۹۹۴/۲	۱۰۵۳	۱۰۶۴/۴	۱۲۳۹	۱۲۵۴/۵
P 18	۲۰	۸	۸	۱۵۰	۸۹۴	۱۰۳۶	۱۰۵۹/۷	۱۱۲۵	۱۱۵۱/۶	۱۲۷۲	۱۲۹۴/۵
P 19	۳۰	۱۰	۸	۱۵۰	۹۳۴	۱۰۶۰	۱۱۷۴/۶	۱۲۰۲	۱۲۳۴/۳	۱۳۴۳	۱۳۶۸/۸
⊗P 20	۴۰	۱۰	۱۰	۳۰۰	۸۶۵	۱۰۲۳	۱۰۴۱/۲	۱۰۹۵	۱۱۰۷/۵	۱۲۹۵	۱۳۰۴/۵



شکل ۵- نمودار فاصله اطمینان ۹۵٪ برای مقادیر RPD الگوریتم‌ها

منابع

- Bokhorst, J.A.C. and Gaalman, G.J.C., (2009). Cross-training workers in dual resource constrained systems with heterogeneous processing times. *International Journal of Production Research*, Volume 47, Issue 22, Pages 6333-6356.
- ElMaraghy, H., Patel, V., and Ben Abdallah, I., (1999). *Genetic algorithm based approach for scheduling of dual-resource constrained manufacturing systems*. CIRP Annals Manufacturing Technology; Volume 48, Issue 1, Pages 369-372.
- ElMaraghy, H., Patel, V., and Ben Abdallah, I., (2000). Scheduling of manufacturing systems under dual-resource constraints using genetic algorithms. *Journal of Manufacturing Systems*; Volume 19 Issue 3, Pages 186-198.
- Gao, J., Sun, L., and Gen, M., (2008). A hybrid genetic and variable neighborhood descent algorithm for flexible job shop scheduling problems. *Computers & Operations Research*, Volume 35, Issue 9, Pages 2892-2907.
- Garey, M.R., Johnson, D.S., and Sethi, R., (1976). The complexity of flowshop and jobshop scheduling scheduling. *Mathematical Operation Research*; Volume 1, Issue 2, Pages 117-129.
- Hansen, P. and Mladenovic, N., (2001). Variable neighborhood search: principles and applications. *European Journal of Operational Research*, Volume 130, Issue 3, Pages 449-467.
- Huiyuan, R., Lili, J., Xiaoying, X., and Muzhi, L., (2009). Heuristic optimization for dual-resource constrained job shop scheduling. *International Asia Conference on Informatics in Control, Automation and Robotics*, IEEE Explore; Pages 485-488.
- Jaber, M.Y. and Neumann, W.P., (2010). Modelling worker fatigue and recovery in dual-resource constrained systems. *Modelling worker fatigue and recovery in dual-resource constrained systems*; Volume 59, Issue 1, Pages 75-84.
- Jingyao, L., Shudong, S., and Yuan, H., Ganggang, N., (2010). *Research on Double-Objective Optimal Scheduling Algorithm for Dual Resource Constrained Job Shop*. Artificial Intelligence and Computational Intelligence; Lecture Notes in Computer Science, Volume 6319/2010, Pages 222-229.

Jingyao, L., Shudong, S., and Yuan, H., (2011). Adaptive Hybrid Ant Colony Optimization for Solving Dual Resource Constrained Job Shop Scheduling Problem, *Journal OF Software*; Volume 6, Issue 4, Pages 584-594.

Kacem, I., Hammadi, S., and Borne, P., (2002). Approach by localization and Multi-objective evolutionary optimization for flexible job-shop scheduling problems. *IEEE Transactions on Systems, Man, and Cybernetics, Part C*, Volume 32, Issue 1, Pages 1-13.

Kirkpatrick, S., Gelatt, C.D., and Vecchi, M.P., (1983). *Optimization by simulated annealing*. *Science*, Volume 220, Pages 671–680.

Lei, D. and Guo, X., (2014). Variable neighbourhood search for dual-resource constrained flexible job shop scheduling”, *International Journal of Production Research*, 52(9), pp. 2519-2529.

Liu, X.X., Lio, Ch., and Tao, Z., (2011). Research on Bi-Objective Scheduling of Dual-Resource Constrained Flexible Job Shop, *Advanced Materials Research*; Volume 211-212, Pages 1091-1095.

Lobo, B.J., Hodgson, T.J., King, R.E., and Thoney, K.A., and Wilson, J.R., (2013). An effective lower bound max L in a worker-constrained job shop. *Computers and Operations Research*, Volume 40, Issue 1, Pages 328-343.

Mati, Y. and Xie, X., (2004). The complexity of two-job shop problems with multi-purpose unrelated machines. *European Journal of Operational Research*, Volume 152, Pages 159-169.

Metropolis, N., Rosenbluth, A.W., Rosenbluth, M.N., Teller, A. H., & Teller, E. (1953). Equation of state calculation by fast computing machines. *Journal of Chemical Physics*, 21.

Mladenovic, N. (1995). A variable neighborhood algorithm—a new metaheuristic for combinatorial optimization. *Abstracts of papers presented at Optimization Days: Montre’ al*, 112–112.

Mladenovic, N. and Hansen, P. (1997). Variable neighborhood search. *Computers and Operations Research*, Volume 24, Issue 11, Pages 1097–1100.

Nelson, R.T., (1967). Labor and machine limited production systems, *Management Science*, Volume 13, Issue 9, Pages 648–671.

Pezzella F., Morganti G., and Ciaschetti G., (2008). A genetic algorithm for the Flexible Job-shop Scheduling Problem. *Computers & Operations Research*; Volume 35, Issue 10, Pages 3202-3212.

Pinedo, M., (1995). *Scheduling: Theory, Algorithms and Systems*, Prentice-Hall. New Jersey, Englewood Cliffs.

Ruiz, R. and Allahverdi, A. (2007). Some effective heuristics for no-wait flowshops with setup times to minimize total completion time. *Annals of Operations Research*, Volume 156, Issue 1, Pages 143-171.

Xianzhou, C. and Zhenhe, Y., (2011). An Improved Genetic Algorithm for Dual-Resource Constrained Flexible Job Shop Scheduling, *Intelligent Computation Technology and Automation (ICICTA)*; Volume 1, Pages 42-45.

Yazdani, M., Amiri, M., and Zandieh, M., (2010). Flexible job-shop scheduling with parallel variable neighborhood search algorithm. *Expert Systems with Applications*, 37(1), Pages 678–687.

ZhiGang, L. Yan, L., and Shujuan, L., (2007). Multi-resource constrained job shop optimization scheduling based on ant colony algorithm. *Journal of System Simulation*, Volume 19, Pages 200-216.

Zhijun S. and Jianying, Zh., (2005). Intelligent optimization for job shop scheduling of dual-resources. *Journal of Southeast University (Natural Science Edition)*, Volume 35, Pages 376-381.

