

حل مسئله‌ی زمان‌بندی پروژه در حالت چندگانه (MRCPSP) به کمک الگوریتم ژنتیک

محمدحسن سیب * (استاد)

محمدرضا افشار (کارشناس ارشد)

یعقوب علیپوری (دانشجوی دکتری)

دانشکده‌ی مهندسی عمران و محیط زیست، دانشگاه صنعتی امیرکبیر

مهندسی عمران شریف، بهار ۱۳۹۵ (۱۳۹۵)
دربی ۲ - ۳۲، شماره ۱/۱، ص. ۱۰۱-۱۱۲، (یادداشت شنی)

وجود محدودیت منابع و روابط پیش‌نیازی بین بعضی از فعالیت‌ها در یک پروژه، زمان‌بندی پروژه را به یک مسئله‌ی دشوار تبدیل کرده است. در این نوشتار، الگوریتم ژنتیک توسعه‌یافته‌ی برای حل مسئله‌ی زمان‌بندی پروژه در حالت چندگانه (MRCPSP) با هدف کمینه‌ساختن زمان پروژه، تحت محدودیت‌های پیش‌نیازی و منابع پیشنهاد شده است. جهت حل این مسئله، روش نمایش کلید تصادفی و روش نمایش فهرست حالات اجرایی مربوط، برای کدگذاری استفاده و به منظور رمزگشایی نیز از روش تولید زمان‌بندی سری چندحالتی کمک گرفته شده است. در این مطالعه، تابع تناسب جدیدی جهت کاهش زمان محاسبات برنامه ارائه شده است. همچنین عملگر جهش جدیدی جهت بهبود کیفیت راه‌حل‌ها پیشنهاد شده است. مجموعه‌های پایه و شناخته‌شده‌ی کتابخانه‌ی مسائل زمان‌بندی پروژه‌ها (PSBLIB)، جهت آزمایش الگوریتم ژنتیک پیشنهادی به کار گرفته شده‌اند، و نتایج محاسباتی حاصل از آن و مقایسه‌های انجام‌شده، کارآمدی الگوریتم پیشنهادی را نشان می‌دهد.

واژگان کلیدی: زمان‌بندی پروژه‌ها در حالت چندگانه، محدودیت‌های پیش‌نیازی، محدودیت منابع، الگوریتم ژنتیک، روش نمایش کلید تصادفی.

۱. مقدمه

زمان‌بندی پروژه^۱ به‌عنوان یکی از زیرشاخه‌های برنامه‌ریزی پروژه^۲، در سال‌های اخیر مورد توجه بسیاری از پژوهشگران قرار گرفته است. وجود محدودیت منابع و روابط پیش‌نیازی بین بعضی از فعالیت‌ها، زمان‌بندی پروژه را به یک مسئله‌ی دشوار تبدیل کرده است.^۱ زمان‌بندی پروژه تحت محدودیت منابع (RCPSP)^۳ و گسترش یافته‌ی آن، RCPSP در حالت چندگانه (MRCPSP)^۴، مهم‌ترین مسائل در این زمینه هستند که مسئله‌ی دوم (MRCPSP) به واقعیت نزدیک‌تر است. MRCPSP پیچیده‌تر از RCPSP است. پژوهشگران در پژوهشی در سال ۱۹۹۷ اثبات کرده‌اند که اگر بیش از یک منبع تجدیدناپذیر داشته باشیم، یافتن حتی یک راه‌حل ممکن برای MRCPSP، راه‌حل NP-complete است.^۱ هدف از حل MRCPSP، یافتن یک حالت اجرایی و یک زمان شروع برای هر فعالیت به‌گونه‌ی است که زمان پروژه، کمینه شود و محدودیت‌های پیش‌نیازی و منابع رعایت شوند.

سه روش جهت حل MRCPSP وجود دارد: ۱. روش‌های دقیق، ۲. روش‌های ابتکاری، ۳. روش‌های فراابتکاری. در روش‌های دقیق، اولین فرایند بهینه، برنامه‌ریزی خطی^۵ است، که در پژوهشی در سال ۱۹۸۰ ارائه شده است.^۳ در پژوهش دیگری در سال ۱۹۸۲، روش شمارشی پیشنهاد شده،^۴ و در سال ۱۹۸۹، روش مذکور بهبود یافته است.^۵ سپس در مطالعه‌ی در سال ۱۹۹۴، زمان محاسبات توسط ۴ معیار عمده و یک حد ممکن مرزی کاهش یافته است.^۶ همچنین پژوهشگران دیگری در سال ۱۹۹۳ یک الگوریتم شاخه و کران ارائه کرده‌اند،^۷ به طوری که در پژوهش دیگری در سال ۱۹۹۷ نشان داده شده است که الگوریتم مذکور برای نمونه‌هایی با دو یا بیش از دو منبع تجدیدناپذیر قادر نخواهد بود تا راه حل بهینه را بیابد.^۸ همچنین در سال ۱۹۹۷، شیوه‌ی شمارشی ارائه‌شده توسط برخی پژوهشگران،^۹ به حالت چندگانه گسترش یافته است.^۹ سپس در سال ۱۹۹۸، روش‌های مختلف شاخه و کران پیشنهاد شده است.^{۱۰} اگرچه روش‌های دقیق می‌توانند مسائل را به شکل بهینه حل کنند، اما قادر نیستند مسائلی با بیش از ۲۰ فعالیت را در زمان محاسباتی معقول حل کنند.^{۱۱} لذا، پژوهشگران مجبور به استفاده از روش‌های ابتکاری و فراابتکاری، برای دستیابی به زمان‌بندی معقول با زمان محاسبات کوتاه هستند.

برخی پژوهشگران در پژوهش‌های خود، با استفاده از اعمال یک محدودیت زمانی روی الگوریتم شاخه و کران، یک روش ابتکاری برای حل مسائل بزرگ پیشنهاد کرده‌اند.^{۱۱} همچنین در پژوهشی در سال ۱۹۹۳، یک فرایند زمان‌بندی اتفاقی

* نویسنده مسئول

تاریخ دریافت: ۱۳۹۲/۷/۹، اصلاحیه ۱۳۹۳/۳/۱۹، پذیرش ۱۳۹۳/۴/۱۶

پیشنهاد شده است.^[۱۲] در یک فرایند جستجوی محلی نیز فرایند اختصاص حالت اجرایی بهبود بخشیده شده است.^[۱] فرایند یک عبوره و چند عبوره نیز در پژوهشی در سال ۱۹۹۴ ارائه شده است.^[۱۳] همچنین برخی پژوهشگران، چندین روش ابتکاری چند عبوره، براساس قوانین اولویت پیشنهاد کرده‌اند.^[۱۴]

روش‌های فراابتکاری، نسل جدید روش‌های ابتکاری هستند و در سال‌های اخیر به طور موفقیت‌آمیزی در حل مسائل MRCPSP مورد استفاده قرار گرفته‌اند. برخی پژوهشگران (۱۹۸۹) نیز نشان داده‌اند که روش‌های فراابتکاری عملکرد بهتری نسبت به روش‌های ابتکاری دارند.^[۱۵] در روش‌های مذکور، پژوهشگران دیگری با استفاده از سیستم شبیه‌سازی ایمنی بدن^۶ به حل MRCPSP پرداخته‌اند،^[۱۶] و در MRCPSP با جدایش، الگوریتم ژنتیک جدیدی را با استفاده از روش تولید زمان‌بندی سری گسترش یافته به‌کار گرفته‌اند.^[۱۷] همچنین در پژوهش دیگری در سال ۲۰۰۱، از الگوریتم شبیه‌سازی تبرید (SA) استفاده شده است.^[۱۸] در سال ۲۰۰۵ نیز از الگوریتم دسته‌ی پرندگان (PSO)^۸ برای حل MRCPSP کمک گرفته شد،^[۱۹] و برخی پژوهشگران در سال ۲۰۰۹ نیز از یک جستجوی ترکیبی پراکنده برای این منظور استفاده کرده‌اند.^[۲۰] الگوریتم دسته‌ی مورچگان^۹ (ACO) نیز در سال ۲۰۱۲ ارائه شده است.^[۲۱] همچنین در پژوهش‌هایی در سال ۲۰۰۹ از یک الگوریتم دیفرانسیلی تکاملی استفاده شده است، که در آن تأثیر اندازه‌ی جمعیت روی بهبود کیفیت راه‌حل‌ها تحلیل؛^[۲۲] و نیز یک رویکرد آموزشی چند عامله^{۱۰} ارائه شده است.^[۲۳] در پژوهشی نیز در سال ۲۰۱۰، از یک الگوریتم ترکیبی مبتنی بر تبه‌بندی استفاده شده است، که MRCPSP را به یک مسئله‌ی دو هدفه تبدیل می‌کرده است.^[۲۴] همچنین برخی پژوهشگران در سال ۲۰۱۱، یک الگوریتم مؤثر جهش قورباغه (SFLA)^{۱۱} را ارائه کرده‌اند، که در آن از یک روش نمایش جدید جهت کدگذاری و نیز روش طراحی آزمایش تاگوچی (DOE)^{۱۲} برای تعیین پارامترهای مناسب آن استفاده شده است.^[۲۵] در پژوهش دیگری در سال ۲۰۱۲ یک الگوریتم توزیعی مؤثر (EDA)^{۱۳} پیشنهاد شده است، که در آن فهرست حالت اجرایی - فعالیت جهت کدگذاری اقتباس و تنظیم پارامترها براساس آزمایش DOE بررسی شده است.^[۲۶]

در این نوشتار، الگوریتم ژنتیک توسعه‌یافته‌ی جهت حل MRCPSP پیشنهاد شده است، که در آن روش نمایش کلید تصادفی و روش نمایش فهرست حالات اجرایی مربوطه برای کدگذاری و به منظور رمزگشایی از روش تولید زمان‌بندی سری چندحالتی استفاده شده است.

از دلایل انتخاب این الگوریتم جهت حل MRCPSP، می‌توان به توانایی آن در فرار از نقطه‌ی بهینه‌ی محلی (الگوریتم ژنتیک به‌طور هم‌زمان شمار زیادی از نقاط فضای پاسخ را به‌کار می‌گیرد و احتمال گرفتار شدن الگوریتم مذکور را در نقاط بهینه‌ی محلی تا اندازه‌ی زیادی کاهش می‌دهد) و سازگاری بی‌نظیر آن با مسئله‌ی MRCPSP اشاره کرد؛ به طوری که در بین الگوریتم‌های ارائه‌شده جهت حل MRCPSP در ادبیات موضوع، الگوریتم ژنتیک همواره جزء بهترین الگوریتم‌های موجود بوده است (به بخش ۲.۶. مراجعه شود).

این تذکر لازم است که در پژوهش‌های بسیاری از الگوریتم ژنتیک جهت حل MRCPSP کمک گرفته شده است.^[۲۷-۲۳] در الگوریتم‌های ژنتیک ارائه‌شده جهت حل MRCPSP، همواره یکی از مهم‌ترین دغدغه‌های عنوان‌شده‌ی پژوهشگران، بهبود تابع تناسب جهت کاهش زمان پروژه و زمان محاسبات برنامه بوده است. برخی پژوهشگران نیز در سال ۲۰۰۹، جزء آخرین پژوهشگرانی بوده‌اند که با ارائه‌ی تابع تناسب جدیدی، سعی در بهبود مدل پیشنهادی خود کرده‌اند و برای اولین بار فرایند بهبود پیشرو - پسروی چندحالتی را جهت بهبود الگوریتم ژنتیک ارائه کرده‌اند،^[۲۶]

که با توجه به کارایی بی‌نظیر آن جهت کاهش زمان پروژه در این پژوهش نیز از این روش کمک گرفته شده و تابع تناسب جدیدی پیشنهاد شده است که در ادامه به آن پرداخته شده است (به نواقص توابع تناسب پیشین در بخش ۳.۱.۵. اشاره شده است).

ضمن اینکه روش کدگذاری و عملگرهای انتخاب و تقاطع مورد استفاده در این پژوهش در الگوریتم ژنتیک، برای اولین بار در ادبیات موضوع جهت حل MRCPSP به‌کار گرفته شده است. یکی دیگر از نوآوری‌های این پژوهش، معرفی عملگر جهش جدیدی جهت بهبود کیفیت راه‌حل‌هاست.

در بخش دوم این نوشتار نیز MRCPSP توصیف شده است. سپس، در بخش سوم، الگوریتم ژنتیک پیشنهادی، جهت حل MRCPSP به تفصیل شرح داده شده است. در ادامه، در بخش چهارم، به نتایج آزمایش‌های انجام‌شده جهت اثبات کارآمدبودن الگوریتم پیشنهادی و در پایان در بخش پنجم، به بحث و نتیجه‌گیری پرداخته شده است.

۲. جایگاه موضوع و اهمیت آن

زمان‌بندی پروژه با ابداع روش مسیر بحرانی (CPM)^{۱۴} در اوایل قرن پنجاه آغاز شده است. روش CPM با استفاده از محدودیت‌های پیش‌نیازی و زمان مربوط به هر یک از فعالیت‌ها می‌تواند زودترین و دیرترین زمان شروع و پایان هر فعالیت و در نتیجه زودترین زمان پایان پروژه را محاسبه کند. در CPM محدودیت منابع در نظر گرفته نمی‌شوند؛ لذا یک فعالیت می‌تواند براساس محاسبات CPM بعد از اتمام فعالیت‌های پیش‌نیازی خود آغاز شود. این در حالی است که می‌دانیم وجود محدودیت منابع در تخصیص منابع و زمان‌بندی پروژه اثرگذار است.^[۲۶] این موضوع زمینه‌ساز پیدایش مسئله‌ی زمان‌بندی پروژه با محدودیت منابع (RCPSP) شده است. هدف RCPSP کمیته‌سازی زمان پروژه با در نظر گرفتن محدودیت منابع و محدودیت‌های پیش‌نیازی است. در RCPSP کلاسیک، برای هر فعالیت یک حالت اجرایی با زمان و منابع ثابت و مشخص در نظر گرفته می‌شود. با توجه به وجود محدودیت منابع، چنین شرایطی (یعنی وجود فقط یک حالت اجرایی برای هر فعالیت) ممکن است سبب تأخیر فعالیت‌ها و بی‌کاری منابع شود؛ لذا در شرایط عملی در تقسیم منابع محدود بین فعالیت‌های متعدد، برای هر فعالیت چندین حالت اجرایی با زمان و مقدار متفاوتی از منابع در نظر گرفته می‌شود، که جهت زمان‌بندی آن باید از مسئله‌ی زمان‌بندی پروژه با محدودیت منابع در حالت چندگانه (MRCPSP) استفاده شود. MRCPSP صنعت ساخت و ساز بسیار کاربردی است. برای مثال جهت فعالیت‌هایی نظیر بتن‌ریزی یا آرماتوربندی ممکن است به تجهیزات یا کارگران یکسان در یک زمان نیاز باشد. در ضمن بعضی از فعالیت‌های ساخت، عملیات تکراری هستند (نظیر فعالیت‌های یکسان برای طبقات مختلف در ساختمان‌های بلندمرتبه) که ممکن است به طور هم‌زمان آغاز شوند و به تجهیزات و کارگران یکسان نیاز داشته باشند، که با توجه به ناکافی بودن منابع موردنیاز در یک زمان، ترکیبات مختلفی از مقدار منابع، جهت شروع این فعالیت‌ها در نظر گرفته می‌شود، که هر ترکیب یک حالت اجرایی نامیده می‌شود و زمان مربوط به خود را دارد.^[۲۶]

هدف از حل MRCPSP، یافتن یک حالت اجرایی و یک زمان شروع برای هر فعالیت است؛ به گونه‌ی که زمان پروژه، کمیته و محدودیت‌های پیش‌نیازی و منابع

الگوریتم پیشنهادی، نتایج به دست آمده با نتایج بهترین الگوریتم‌های موجود در ادبیات موضوع مقایسه شده است.

۵. الگوریتم ژنتیک

الگوریتم GA بر پایه‌ی مکانیسم‌های تکاملی زیستی و ژنتیک‌های طبیعی قرار دارد. در واقع، ایده‌ی اولیه‌ی این روش از نظریه‌ی تکاملی داروین الهام گرفته شده و کاربرد آن بر پایه‌ی ژنتیک طبیعی استوار است. اصول اولیه‌ی الگوریتم GA استاندارد در سال ۱۹۶۲ در دانشگاه میشیگان ارائه شده است. سپس، در سال ۱۹۷۵ مبنای ریاضی آن در کتابی توسط هلند با نام «تطابق در سیستم‌های طبیعی و مصنوعی» منتشر شده است.^[۲۹] الگوریتم ژنتیک بیشتر اوقات جهت حل مسائل بهینه‌سازی ترکیبی به کار می‌رود و در این گونه مسائل، شمار راه‌حل‌ها به طور نمایی با اندازه‌ی مسئله رشد می‌کند.^[۲۹] در GA راه‌حل‌های اولیه‌ی مسئله در مجموعه‌ی از کروموزوم‌ها ایجاد می‌شوند که هر کدام از آن‌ها یک فرد از افراد جمعیت اولیه را تشکیل می‌دهد. GA این جمعیت را به صورت تصادفی یا از روش‌های ابتکاری ایجاد می‌کند و براساس سه عملگر متوالی و تکرارشونده‌ی: ۱. عملگر انتخاب؛ ۲. عملگر تقاطع؛ ۳. عملگر جهش، آن را تکامل می‌دهد و جمعیت جدید در پایان هر تکرار ایجاد می‌شود.

برای مسائل بهینه‌سازی بزرگ، جمعیت اولیه می‌تواند از دانش قبلی در مورد راه‌حل‌ها ایجاد شود. این فرایند نباید تنوع جمعیت را به شدت محدود کند، در غیر این صورت هم‌گرایی زودرس می‌تواند اتفاق بیفتد. اندازه‌ی جمعیت اولیه معمولاً به عنوان تابعی از طول کروموزوم تعیین می‌شود (این اندازه معمولاً بین ۳۰ تا ۲۰۰ انتخاب می‌شود).

۱.۱.۵. الگوریتم ژنتیک برای حل مسئله‌ی MRCPS

در الگوریتم ژنتیک پیشنهادی، جمعیت اولیه به طور تصادفی جهت توزیع‌بخشیدن به فضای جستجو و یافتن مناطق امیدوارکننده ایجاد می‌شود. در ادامه، الگوریتم ارائه شده با جزئیات بیشتری شرح داده شده است.

۱.۱.۵.۱. فرایند پیش‌پردازش

قبل از آغاز الگوریتم ژنتیک، فرایند پیش‌پردازش جهت کاهش فضای جستجو و محاسبات زمانی بر روی MRCPS اعمال می‌شود. براساس این فرایند، از یک طرف حالت‌های اجرایی ناکارآمد و غیرقابل اجرا^{۱۶} و از طرف دیگر، منابع تجدیدناپذیر اضافی^{۱۷} حذف می‌شوند. برای اولین بار در پژوهشی در سال ۱۹۹۷، از این فرایند در الگوریتم شاخه و کران استفاده شده است.^[۹]

۲.۱.۵. روش نمایش راه‌حل‌ها (کدگذاری)

مسئله‌ی انتخاب یک نمایش مناسب جهت جستجو ضروری است.^[۳۲] در ادبیات RCPSP، چندین روش نمایش وجود دارند که روش نمایش کلید تصادفی و فهرست تقدم، محبوبیت بیشتری دارند. در روش نمایش توالی فعالیت‌ها، جایگاه هر فعالیت اولویت نسبی آن را جهت زمان بندی تعیین می‌کند، اما در روش نمایش کلید تصادفی، این وظیفه را مقادیر اولویت تقبل می‌کنند. برخلاف روش نمایش توالی فعالیت‌ها (AL)، در روش نمایش کلید تصادفی (RK)، همه‌ی کروموزوم‌ها یک زمان بندی پیش‌نیازی شدنی را نشان می‌دهند و رعایت روابط پیش‌نیازی در مرحله‌ی ارزیابی تابع تناسب صورت می‌پذیرد. به علاوه در RK، تولید و تکامل جمعیت مستقل از

رعایت شود. در ادبیات MRCPCP سه روش کلی جهت حل MRCPCP پیشنهاد شده است: ۱. روش‌های دقیق، ۲. روش‌های ابتکاری، ۳. روش‌های فراابتکاری. در دهه‌ی اخیر، با توجه به عدم کارایی روش‌های دقیق در مسائل بزرگ، روش‌های ابتکاری و فراابتکاری مورد توجه بیشتری قرار گرفته‌اند.^[۱۱] در این پژوهش نیز سعی بر آن است که با بهبود روش فراابتکاری الگوریتم ژنتیک به حل MRCPS پرداخته شود.

۳. توصیف مسئله

MRCPS به دنبال سیاست تخصیص منابع به گونه‌ی است که زمان پروژه کمینه شود. در MRCPS، هر پروژه $J + 2$ فعالیت دارد که جهت هر فعالیت چندین حالت اجرا پیش‌بینی شده است. بین بعضی از فعالیت‌ها روابط پیش‌نیازی وجود دارد. در این پژوهش، این روابط از نوع پایان - شروع و بدون وقفه در نظر گرفته شده‌اند. هر فعالیت j ($j = 0, \dots, J + 1$) که با حالت اجرایی m ($m \in 1, \dots, M_j$) آغاز می‌شود، باید در همان حالت اجرایی m ، بدون جدایش پایان پذیرد. هر یک از حالات اجرایی به یک یا چند نوع از منابع تجدیدپذیر و تجدیدناپذیر نیاز دارند. در واقع منابع به طور کلی به دو دسته‌ی منابع کاری که به آن منابع تجدیدپذیر نیز می‌گویند و منابع مصرفی یا منابع تجدیدناپذیر تقسیم می‌شوند. منابع کاری شامل نیروی انسانی، ماشین‌آلات و تجهیزات می‌شوند که به هر فعالیت تعداد مشخصی از این منابع تخصیص می‌یابد و انجام فعالیت به عهده‌ی آن‌هاست. معمولاً ماهیت منابع کاری به گونه‌ی است که با اتمام یک فعالیت یا بسته‌ی کاری منابع تخصیص داده شده به آن آزاد یا بی‌کار می‌شوند و می‌توان آن‌ها را به سایر فعالیت‌ها اختصاص داد. منابع مصرفی مثل گچ، سیمان، آجر، مواد خام و... هستند. ماهیت این گونه منابع به گونه‌ی است که با اتمام یک فعالیت یا بسته‌ی کاری، مقدار منابع مصرفی پیش‌بینی شده برای انجام آن فعالیت مصرف می‌شود و به اتمام می‌رسد. موجودیت منابع تجدیدپذیر k ($k = 1, \dots, R$) و موجودیت منابع تجدیدناپذیر l ($l = 1, \dots, N$) به ترتیب R_k و N_l هستند. اگر فعالیت j با حالت اجرایی m انجام شود، زمان مقدار منابع تجدیدپذیر مورد نیاز آن از نوع k و مقدار منابع تجدیدناپذیر مورد نیاز آن از نوع l به ترتیب d_{jmk} و r_{jml} نمایش داده می‌شود. فعالیت‌های شروع و پایان پروژه، فعالیت‌های مجازی با زمان صفر و بدون احتیاج به منابع در نظر گرفته می‌شوند. این فعالیت‌ها به ترتیب صفر و $J + 1$ آمین فعالیت پروژه محسوب می‌شوند. هدف MRCPS تحت شرایط عنوان شده، اختصاص یک زمان شروع و یک حالت اجرایی به هر فعالیت به گونه‌ی است که زمان پروژه، کمینه شود.

۴. متدولوژی تحقیق

در این پژوهش، پس از بررسی انواع روش‌های فراابتکاری که جهت حل MRCPS به کار گرفته شده است، انواع روش‌های کدگذاری و رمزگشایی راه‌حل‌های یک مسئله‌ی MRCPS شناسایی و در ادامه، با توسعه‌ی روش فراابتکاری الگوریتم ژنتیک به حل MRCPS پرداخته شده است.

الگوریتم پیشنهادی با استفاده از برنامه‌ی مطلب^{۱۵} ۲۰۱۲ کدنویسی و با استفاده از مجموعه‌های پایه و شناخته شده‌ی کتابخانه، مسائل زمان بندی پروژه‌ها (PSPLIB) آزمایش شده است. و در پایان جهت اعتبارسنجی نتایج حاصل از

که در آن، $mak(I)$ زمان پروژهی مربوط به راه حل I است، $\max_mak(P)$ بیشترین زمان پروژه مربوط به افراد جمعیت جاری است، $SFT(\mu)$ میزان تجاوز از منابع تجدیدناپذیر راه حل I را نمایش می دهد، و \min_CP مسیر بحرانی پروژه است که با توجه به کمینه ی زمان هر یک از فعالیت ها محاسبه می شود.

در سال ۲۰۰۹، لوا و همکاران با نرمالایز کردن تابع تناسب آلکارز و همکاران (۲۰۰۳)، یک تابع تناسب (معادله ی ۴) را پیشنهاد کرده اند: [۲۱]

$$f(I) = \begin{cases} 1 - \frac{\max_mak(P) - mak(I)}{\max_mak(P)}, & \text{if } ERR(\mu) = 0 \\ 1 + \frac{mak(I) - \min_CP}{mak(I)} + ERR(\mu), & \text{otherwise} \end{cases} \quad (4)$$

$$ERR(\mu) = \sum_{i=1}^N \max \left\{ 0, \frac{\sum_{j=1}^J n_{jmi} - N_i}{N_i} \right\} \quad (5)$$

که در آن، $ERR(\mu)$ درجه ی نشدنی بودن راه حل I را نمایش می دهد (معادله ی ۵) و سایر پارامترها مطابق توضیحات عنوان شده در معادلات قبلی هستند (پارامترهای $ERR(\mu)$ در بخش توصیف مسئله معرفی شده اند).

لوا و همکاران (۲۰۰۹) اثبات کرده اند که با استفاده از شرایط یکسان محاسباتی، تابع تناسب پیشنهادی ایشان، در مقابل توابع تناسب هارتمن و آلکارز (سایر توابع تناسب موجود در ادبیات موضوع)، انحراف متوسط بهتری دارد. [۲۱] به هر جهت واضح است که با توجه به محاسبه ی مسیر بحرانی شبکه ی پروژه در تابع تناسب لوا و همکاران (۲۰۰۹)، زمان محاسبات برنامه افزایش خواهد یافت. لذا، در این پژوهش، یک تابع تناسب دیگر (معادله ی ۶) جهت رفع نقص ذکر شده ارائه شده است:

$$f(I) = \begin{cases} 1 - \frac{T - mak(I)}{T}, & \text{if } ERR(\mu) = 0 \\ 1 + \frac{mak(I)}{T} + ERR(\mu), & \text{otherwise} \end{cases} \quad (6)$$

که در آن، T جمع بیشترین زمان همه ی فعالیت هاست و سایر پارامترها همانند رابطه ی ۱ هستند. انتظار می رود که زمان محاسبات مربوط به تابع تناسب پیشنهادی نسبت به تابع تناسب لوا و همکاران (۲۰۰۹) کمتر باشد؛ چرا که از طرفی تعداد محاسبات تابع تناسب پیشنهادی یکی کمتر از تابع تناسب لوا و همکاران است و از طرف دیگر، زمان محاسبات \min_CP بیشتر از T است.

در ادامه ی این بخش، از انحراف متوسط و زمان محاسبات برنامه جهت اثبات برتری تابع تناسب پیشنهادی استفاده شده است. برای این منظور، نمونه های استاندارد $J10, J12, J14, J16, J18$ و $J20$ از کتابخانه ی مسائل زمان بندی پروژه ها [۱۸] انتخاب شده اند. معیار توقف لحاظ شده، 5000 زمان بندی تولیدی است.

همان طور که انتظار می رفت و جدول ۱ نیز نشان می دهد، زمان اجرای برنامه با استفاده از تابع تناسب پیشنهادی کوچک تر و یا مساوی نسبت به زمانی است که از تابع تناسب لوا و همکاران استفاده می شود.

زمان های عنوان شده در جدول ۱، جهت تولید یک و فقط یک زمان بندی است (یعنی زمان مربوط به یک کروموزوم آن هم فقط جهت یک تکرار) و این در صورتی است که در پروژه های پنج مارک PSBLIB جهت یافتن زمان بهینه (در پروژه هایی که فقط بین ۱۰ تا ۲۰ فعالیت دارند)، کمینه ی $5000 \times 0.701 = 5000$ زمان بندی باید تولید شوند؛ لذا، با استفاده از تابع تناسب پیشنهادی $5000 \times 0.701 = 5000$ ثانیه، زمان اجرای برنامه بهبود می یابد. بالتبع اگر تعداد فعالیت ها بیشتر شوند (مطابق پروژه های واقعی)، از

روش تولید زمان بندی است؛ در حالی که در فهرست توالی فعالیت ها، باید روابط بین فعالیت ها در حین تولید و تکامل جمعیت کنترل شوند.

اما ضعف RK، تولید زمان بندی یکسان به دلیل تغییرات اندک مقادیر اولویت در فرایند تکامل در بعضی از نمایش هاست. پس از در نظر گرفتن ملاحظات عنوان شده در این پژوهش تصمیم گرفته شد از شیوه ی نمایش کلید تصادفی، جهت تعیین اولویت فعالیت ها برای زمان بندی استفاده شود. همچنین جهت اختصاص حالت های اجرایی به هر یک از فعالیت ها، از فهرست حالت های اجرایی (ML) کمک گرفته شده است؛ لذا هر راه حل I با دو بردار λ و μ ($I = (\lambda, \mu)$)، که μ به ترتیب RK و ML را نمایش می دهند، نشان داده شده است.

۳.۱.۵. تابع تناسب

ارزیابی یک کروموزوم برای تکامل یک جمعیت لازم و ضروری است، که این ارزیابی توسط تابع تناسب انجام می شود. تعیین تابع تناسب مناسب جهت عملکرد صحیح الگوریتم ژنتیک ضروری است. [۲۲] عملگرهای تقاطع و جهش به عنوان عملگرهای عمومی الگوریتم ژنتیک، در هنگام تولید جمعیت جدید، شدنی شدن راه حل ها را در نظر نمی گیرند؛ لذا تولید فرزندان نشدنی امری بدهی خواهد بود، که باید برای این فرزندان مجازاتی در نظر گرفته شود.

در ادبیات MRCPSP، تاکنون سه تابع تناسب تعریف شده اند. هارتمن (۲۰۰۱) این تابع تناسب (معادله ی ۱) را ارائه کرده است: [۲۱]

$$f(I) = \begin{cases} mak(I), & \text{if } SFT(\mu) = 0 \\ T + SFT(\mu), & \text{otherwise} \end{cases} \quad (1)$$

که در آن، $mak(I)$ زمان پروژه ی مربوط به راه حل I است، T جمع بیشترین زمان همه ی فعالیت هاست و $SFT(\mu)$ میزان تجاوز از منابع تجدیدناپذیر راه حل I را نمایش می دهد، که مطابق معادله ی ۲ به دست می آید:

$$SFT(\mu) = \sum_{i=1}^L \max \left\{ 0, \sum_{j=1}^J n_{jmi} - N \right\} \quad (2)$$

روشن است که اگر $SFT(\mu)$ برابر صفر شود، راه حل I شدنی است (پارامترهای $SFT(\mu)$ در بخش توصیف مسئله معرفی شده اند). نقاط ضعف این تابع تناسب عبارت است از:

۱. برای هر راه حل نشدنی، زمان پروژه ی مربوط به آن لحاظ نشده و فقط میزان تجاوز از منابع تجدیدناپذیر لحاظ شده است. لذا، دو راه حل با میزان تجاوز از منابع تجدیدناپذیر یکسان (بدون لحاظ زمان پروژه)، مقادیر تناسب یکسانی خواهد داشت که درست نیست.

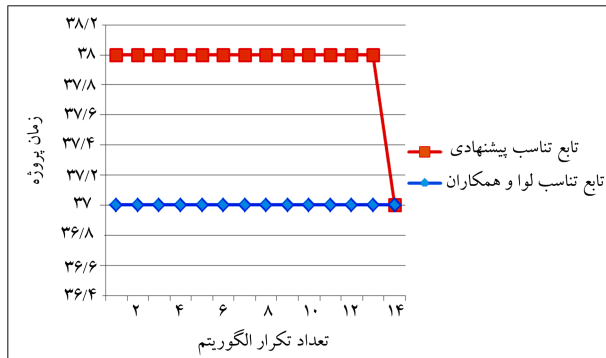
۲. عدد T بسیار بزرگ است، لذا احتمال حضور راه حل نشدنی در مرحله ی بعدی تکامل، نزدیک به صفر می شود. البته این نقص با برگزیدن عملگر انتخاب مناسب قابل جبران است.

۳. برای راه حل نشدنی، واحد زمان با منابع جمع شده است. جهت حل دو مشکل ابتدایی، آلکارز و همکاران (۲۰۰۳) تابع تناسب جدیدی ارائه کرده اند (معادله ۳): [۲۰]

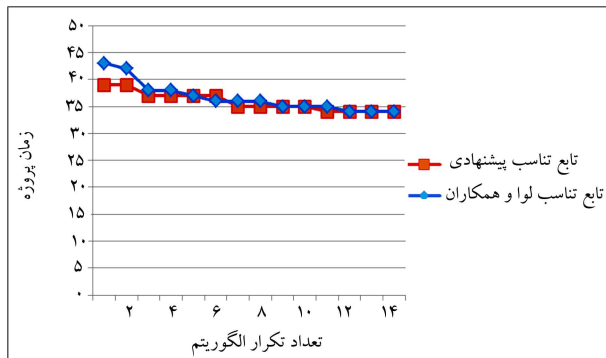
$$f(I) = \begin{cases} mak(I), & \text{If } ERR(\mu) \\ \max_mak(P) + mak(I) & \\ - \min_CP + SFT(\mu), & \text{otherwise} \end{cases} \quad (3)$$

جدول ۱. عملکرد توابع تناسب (معیار توقف لحاظ شده، ۵۰۰۰ زمان بندی تولیدی است).

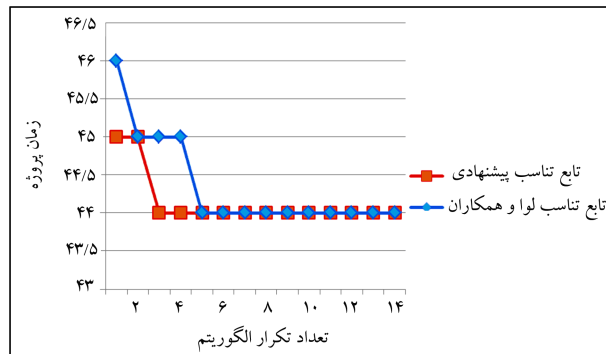
تابع تناسب	J۲۰	J۱۸	J۱۶	J۱۴	J۱۲	J۱۰
انحراف متوسط (%)	پژوهش حاضر	۰٫۵۷	۰٫۳۶	۰٫۳۵	۰٫۱۶	۰٫۰۲
	لوا و همکاران	۰٫۶۰	۰٫۳۸	۰٫۳۸	۰٫۱۸	۰٫۰۴
نرخ بهینه (%)	پژوهش حاضر	۶۸	۷۵	۷۵	۸۷	۹۸
	لوا و همکاران	۶۶	۷۴	۷۴	۸۶	۹۶
زمان جهت تولید یک زمان بندی (ثانیه)	پژوهش حاضر	۰٫۱۹	۰٫۱۸	۰٫۱۷	۰٫۱۵	۰٫۱۰
	لوا و همکاران	۰٫۲۰	۰٫۱۸	۰٫۱۷	۰٫۱۶	۰٫۱۰



شکل ۱. مقایسه‌ی سرعت هم‌گرایی تابع تناسب پیشنهادی و تابع تناسب لوا و همکاران بر روی نمونه‌ی ۲-۳۰۲۹.J



شکل ۲. مقایسه‌ی سرعت هم‌گرایی تابع تناسب پیشنهادی و تابع تناسب لوا و همکاران بر روی نمونه‌ی ۵-۳۰۴۶.J



شکل ۳. مقایسه‌ی سرعت هم‌گرایی تابع تناسب پیشنهادی و تابع تناسب لوا و همکاران بر روی نمونه‌ی ۸-۳۰۵۳.J

آنجایی که لازم است تعداد زمان بندی بیشتری (جهت یافتن زمان بهینه) تولید شود، لذا با استفاده از تابع تناسب پیشنهادی، زمان اجرای برنامه بهبود بیشتری می‌یابد. به علاوه انحراف متوسط‌های به دست آمده نشان می‌دهند که در همه‌ی موارد (به جز J۱۲) تابع تناسب پیشنهادی این پژوهش، عملکرد بهتری نسبت به تابع تناسب لوا و همکاران (۲۰۰۹) و در نتیجه، عملکرد بهتری نسبت به همه‌ی توابع تناسب پیشنهادی در ادبیات موضوع دارد.

جهت مقایسه‌ی سرعت هم‌گرایی، چندین نمونه از سخت‌ترین مسائل موجود در کتابخانه‌ی مسائل زمان بندی پروژه‌ها انتخاب شده‌اند و پس از انجام آزمایش‌های لازم مشخص شده است که در برخی از نمونه‌ها، سرعت هم‌گرایی الگوریتم با استفاده از تابع تناسب لوا و همکاران بهتر است (شکل ۱)، در برخی دیگر نیز سرعت هم‌گرایی الگوریتم با استفاده از هر دو تابع تناسب تقریباً یکسان است (شکل ۲)، و در مواردی نیز سرعت هم‌گرایی الگوریتم با استفاده از تابع تناسب پیشنهادی این پژوهش بهتر است (شکل ۳). لذا، نمی‌توان عنوان کرد که سرعت هم‌گرایی الگوریتم با استفاده از کدامین تابع تناسب بهتر خواهد بود.

پس از ارزیابی هر یک از کروموزوم‌ها به وسیله تابع تناسب، یک مقدار تناسب به هر کدام از آن‌ها اختصاص داده می‌شود.

۴.۱.۵. عملگر انتخاب

عملگر انتخاب بیشتر از عملگرهای تقاطع و جهش، در تعیین مشخصه‌های هم‌گرایی الگوریتم ژنتیک اثرگذار است. وظیفه‌ی عملگر انتخاب، علاوه بر تولید بخشی از جمعیت جدید، ایجاد جمعیت متوسط (استخر جفت‌گیری) است. [۲۲] در این پژوهش عملگر انتخاب به این ترتیب عمل می‌کند:

بر اساس تناسب کروموزوم‌ها، برخی از بهترین کروموزوم‌ها (افراد بالا^{۲۰} در شکل ۴) از نسل جاری به نسل بعدی انتقال داده می‌شوند. این راهبرد، راهبرد نخبه‌گرا^{۲۱} نامیده می‌شود. نقطه ضعف راهبرد عنوان‌شده، هم‌گرایی به نقطه‌ی بهینه‌ی محلی است که در این مطالعه، با عملگر جهش معرفی شده می‌تواند کنترل شود.

در انتخاب والدین، پدر از بهترین کروموزوم‌ها (افراد بالا در شکل ۴) و مادر از میان همه کروموزوم‌های جمعیت جاری به طور تصادفی انتخاب می‌شوند. این نوع از عملگر انتخاب، در الگوریتم ژنتیک مندوز و همکاران (۲۰۰۹) جهت حل RCPSP به کار گرفته شده و نتایج قابل قبولی حاصل شده است. در عملگر انتخاب مندوز و همکاران (۲۰۰۹)، بخشی از جمعیت جدید به طور تصادفی تولید شده است که در این پژوهش با توجه به اینکه جمعیت اولیه به طور تصادفی ایجاد شده است، از این قسمت عملگر انتخاب ایشان صرف‌نظر شده است. [۲۳]

شدنی و نشدنی متفاوت است. در تغییر حالت اجرایی یک فعالیت (که انتخاب فعالیت و تغییر حالت اجرایی آن به‌طور تصادفی رخ می‌دهد)، اگر ژن مربوط به راه‌حل شدنی باشد، تغییر در حالت اجرایی آن فعالیت به شرطی اتفاق می‌افتد که میزان مصرف منابع تجدیدناپذیر افزایش یابد (یعنی مقدار η ، در معادله‌ی ۷ افزایش یابد) و راه‌حل شدنی باقی بماند؛ و اگر ژن مربوط به راه‌حل نشدنی باشد، این تغییر به شرطی رخ می‌دهد که میزان ERR فهرست حالات اجرایی جدید کاهش یابد:

$$\eta = \frac{\sum_{l=1}^N \sum_{j=1}^J n_{jml} - N_l}{N_l} \quad (7)$$

عمل‌گر جهش پیشنهادی، تا زمانی که به یک نرخ جهش مشخص برسیم، ادامه می‌یابد.

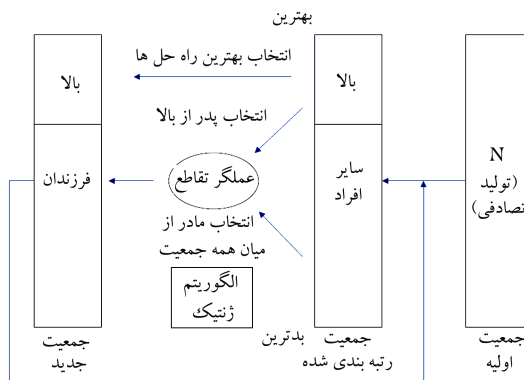
۷.۱.۵. روش تولید زمان‌بندی (فرایند رمزگشایی)

روش تولید زمان‌بندی، روشی مؤثر جهت تبدیل یک راه‌حل RCPSP به زمان‌بندی است. در ادبیات موضوع، دو نوع روش تولید زمان‌بندی وجود دارد: ۱. روش تولید زمان‌بندی سری، ۲. روش تولید زمان‌بندی موازی.

از آنجا که کولیش اثبات کرده است که با استفاده از روش تولید زمان‌بندی موازی، گاهی اوقات رسیدن به جواب بهینه غیرممکن است؛ در این پژوهش، از روش تولید زمان‌بندی سری استفاده شده است.^[۳۵] در MRCPSP، به دلیل وجود منابع تجدیدناپذیر، روش تولید زمان‌بندی سری باید اصلاح شود. در واقع، دو مسئله‌ی مهم باید در نظر گرفته شوند: ۱. تبدیل راه‌حل‌های نشدنی به شدنی تا جایی که ممکن باشد؛ ۲. استفاده‌ی کامل از منابع تجدیدناپذیر در راه‌حل‌های شدنی است. جهت پیاده‌سازی این دو هدف، از فرایند مقابله با راه‌حل‌های شدنی و نشدنی^{۲۲} کمک گرفته می‌شود. این فرایندها، روش تولید زمان‌بندی سری را به روش تولید زمان‌بندی سری چند حالتی (MSSGS)^{۲۳} گسترش می‌دهند.

فرایند مقابله با راه‌حل‌های نشدنی: بهترین شیوه در مواجهه با راه‌حل‌های نشدنی، کاربرد فرایند ترمیم جهت بهبود برخی از آن‌هاست.^[۳۲] از این رو در این نوشتار، بعد از تولید جمعیت اولیه و اعمال فرایند پیش‌پردازش، فرایند مقابله با راه‌حل‌های نشدنی جهت تبدیل راه‌حل‌های نشدنی به شدنی آغاز شده است. این فرایند به هر کدام از راه‌حل‌های نشدنی به‌طور جداگانه اعمال شده است. در این فرایند، یک فعالیت به‌طور تصادفی انتخاب شده و حالت اجرایی آن به حالت اجرایی جدیدی تغییر یافته است؛ اگر با این تغییر، ERR کاهش یافت، حالت اجرایی جدید جایگزین قبلی می‌شود. این فرایند تا زمانی که به یک مقدار مشخص از پیش تعیین شده برسیم، ادامه می‌یابد (در این پژوهش، این مقدار مشابه سایر مطالعات موجود در این زمینه، برابر تعداد فعالیت‌های پروژه در نظر گرفته شده است). پس از انجام فرایند عنوان‌شده، فقط راه‌حل‌های شدنی اجازه‌ی ورود به تکرار پیشرو - پس‌روی چند حالتی (MM-FBI)^{۲۴} (که در ادامه شرح داده شده است) را خواهند داشت. فرایند مقابله با راه‌حل‌های شدنی در MM-FBI بخش بعد شرح داده شده است.

فرایند تکرار پیشرو - پس‌روی چند حالتی (MM-FBI): در RCPSP، تکرار پیشرو - پس‌روی (FBI)^{۲۵} یک شیوه‌ی مؤثر جهت بهبود کیفیت راه‌حل‌هاست.^[۳۶] این ایده، توسط لوا و همکاران (۲۰۰۹) جهت MRCPSP، به نام فرایند تکرار پیشرو - پس‌روی چند حالتی (MM-FBI) گسترش یافته است.^[۳۱] این روش با استفاده از فرایند بهبود حالت اجرایی^{۲۶}، زمان پایان هر فعالیت را در هر نقطه‌ی تصمیم از زمان‌بندی جزئی، بدون تغییر حالات اجرایی و تأخیر سایر فعالیت‌ها کاهش می‌دهد.^[۳۰]



شکل ۴. عمل‌گر انتخاب الگوریتم ژنتیک پیشنهادی.

کروموزم ۱	۰/۷۵	۰/۴۳	۰/۵۵	۰/۸	۰/۶
کروموزم ۲	۰/۵۵	۰/۷۷	۰/۶۸	۰/۳۷	۰/۷۲
بردار تصادفی	۰/۴۴	۰/۷۱	۰/۴۵	۰/۹۲	۰/۶۷
رابطه با احتمال تقاطع ۰/۷	<	>	<	>	<
فرزند	۰/۷۵	۰/۷۷	۰/۵۵	۰/۳۷	۰/۶

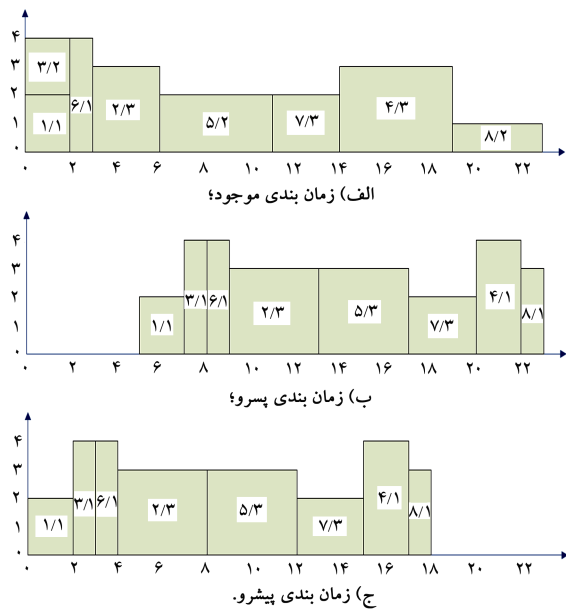
شکل ۵. یک مثال برای احتمال تقاطع در الگوریتم ژنتیک پیشنهادی (احتمال تقاطع = ۰/۷).

۵.۱.۵. عمل‌گر تقاطع

اسپیرس و دیزانگ (۱۹۹۱) با بررسی انواع عمل‌گرهای تقاطع نشان دادند که عمل‌گر تقاطع یکنواخت عملکرد بهتری نسبت به سایرین دارد. براساس مطالعه‌ی ایشان، عمل‌گر عمل‌گر تقاطع یکنواخت، مستقل از طول کروموزوم است.^[۳۴] با توجه به این موضوع، در این پژوهش از عمل‌گر تقاطع یکنواخت استفاده شده است. در عمل‌گر تقاطع یکنواخت یک عدد تصادفی در بازه‌ی ۰ و ۱، به عنوان احتمال تقاطع در نظر گرفته می‌شود (در این پژوهش این عدد ۰/۷ لحاظ شده است). سپس یک بردار تصادفی حاوی اعدادی در بازه‌ی ۰ و ۱ تولید می‌شود. طول این بردار برابر طول راه‌حل است. پس از مقایسه‌ی احتمال تقاطع و اعداد بردار تصادفی، ژن‌هایی که باید از پدر و مادر به فرزند انتقال یابند، تعیین می‌شوند. زمانی که عدد تصادفی کمتر از احتمال تقاطع باشد، ژن مربوط و متناظر با عدد تصادفی از پدر به فرزند انتقال می‌یابد؛ در غیر این صورت، این ژن از مادر به فرزند انتقال می‌یابد (شکل ۵). در واقع، این مسئله سبب می‌شود که ۷۰٪ ژن‌های فرزند تولیدشده، مشابه بهترین راه‌حل‌های موجود شود (چون احتمال تقاطع ۰/۷ در نظر گرفته شده است). یک نمونه مثال از این عمل‌گر در شکل ۵ مشاهده می‌شود.

۶.۱.۵. عمل‌گر جهش

عمل‌گر جهش، ابزاری مهم در تکرارهای پایانی الگوریتم ژنتیک است، که وظیفه‌ی آن حفظ تنوع در فضای جستجو است. در این پژوهش، جهت اعمال عمل‌گر جهش بر روی اولویت فعالیت‌ها، چند فعالیت از راه‌حل‌های متفاوت و یا یک راه‌حل، به‌طور تصادفی انتخاب شده و اولویت آن‌ها به مقادیر جدید دیگری (که این مقادیر نیز به‌طور تصادفی تعیین می‌شوند) تغییر یافته است. جهت تغییر فهرست حالت‌های اجرایی، عمل‌گر جهش جدیدی تعریف شده است که رفتار آن در مواجهه با راه‌حل‌های



شکل ۷. یک تکرار MM-FBI و میزان مصرف منبع تجدیدناپذیر (NRC).

در این قسمت پژوهش، یک گام از MM-FBI پیاده‌سازی شده است. فرض شود که فهرست اولویت فعالیت‌ها (λ) و فهرست حالات اجرایی (μ), یک راه‌حل شدنی جهت پروژه‌ی فوق‌الذکر بدین صورت است (جدول ۳).

گراف منابع مربوط به راه‌حل ذکر شده، با استفاده از فرایند SSGS در شکل ۷ الف، نمایش داده شده است (۳.۲). به معنای فعالیت ۳ با حالت اجرایی ۲ است. جهت ترسیم گراف منابع مربوط به راه‌حل ذکر شده (شکل ۷ الف) با استفاده از فرایند MSSGS به این ترتیب عمل شده است: این ترسیم در J مرحله باید انجام شود که پس از حذف فعالیت‌های غیرمجازی (فعالیت‌های ابتدایی و انتهایی به شماره‌ی ۰ و ۹ که نشان‌دهنده‌ی آغاز و پایان پروژه هستند)، در هر مرحله یک فعالیت انتخاب و زمان‌بندی می‌شود. دو مجموعه‌ی فعالیت در هر مرحله وجود دارد:

۱. مجموعه‌ی زمان‌بندی شده‌ی S_j ($j = 1, \dots, J$);

۲. مجموعه‌ی تصمیم D_j ($j = 1, \dots, J$); شامل فعالیت‌هایی که پیش‌نیازهایشان به انجام رسیده است؛ یعنی این پیش‌نیازها در گراف منابع زمان‌بندی شده‌اند.

در هر مرحله، فعالیت‌هایی از مجموعه‌ی تصمیم جهت زمان‌بندی انتخاب می‌شود که اولویت بیشتری دارد (مقدار اولویت هر فعالیت در فهرست λ مربوط به هر راه‌حل مشخص شده است) و این فعالیت با رعایت محدودیت منابع تجدیدناپذیر در گراف منابع زمان‌بندی می‌شود. سپس فعالیت مذکور از مجموعه‌ی تصمیم به مجموعه‌ی زمان‌بندی شده انتقال می‌یابد.

برای مثال جهت زمان‌بندی اولین فعالیت در راه‌حل ذکر شده به این ترتیب عمل می‌شود:

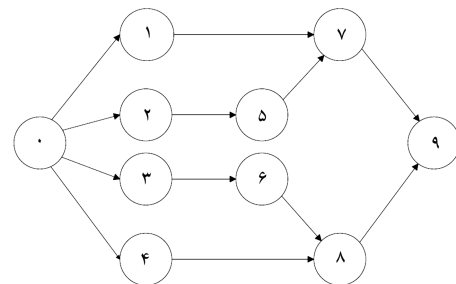
پس از تشکیل مجموعه‌ی تصمیم که شامل فعالیت‌های ۱، ۲، ۳ و ۴ است (عملت قرارگیری این ۴ فعالیت در مجموعه‌ی تصمیم این است که ۴ فعالیت مذکور، پیش‌نیازی ندارند (شکل ۶ را ببینید))، فعالیت ۱ از مجموعه‌ی تصمیم، جهت زمان‌بندی در نقطه‌ی صفر پروژه انتخاب می‌شود (چون فهرست λ مقدار اولویت بیشتری را برای این فعالیت نمایش می‌دهد). جهت زمان‌بندی سایر فعالیت‌های این راه‌حل نیز به همین ترتیب عمل می‌شود، تا گراف شکل ۷ الف کاملاً تولید شود.

نقطه ضعف فرایند MM-FBI این است که فعالیت‌هایی که زودتر زمان‌بندی می‌شوند، شانس بیشتری جهت استفاده از منابع تجدیدناپذیر دارند. جهت غلبه بر این نقطه ضعف در ادبیات موضوع، فرایند بهبود حالت اجرایی به‌طور تصادفی فقط به برخی از فعالیت‌ها اعمال شده است. از این رو تضمین می‌شود که همه‌ی فعالیت‌ها، شانس برابری جهت استفاده از منابع تجدیدناپذیر داشته باشند.

این فرایند تصادفی بدین صورت پیاده‌سازی می‌شود: ابتدا، برای تک‌تک فعالیت‌های یک راه‌حل، یک عدد تصادفی بین ۰ و ۱ تولید می‌شود. اگر عدد تصادفی مربوط به هر فعالیت از یک مرز SGS_Pper که معمولاً ۰٫۷ در نظر گرفته می‌شود) کمتر شود، فرایند بهبود حالت اجرایی به آن فعالیت اعمال می‌شود. بدین ترتیب به ۷۰٪ فعالیت‌ها (در $SGS_Pper = 0.7$) این فرایند اعمال خواهد شد. در واقع وانگ و فانگ (۲۰۱۱) نشان داده‌اند که SGS_Pper مهم‌ترین عامل در تعیین کیفیت راه‌حل‌ها و بهترین مقدار برای آن عدد ۰٫۷ است.^[۲۵] جهت روشن‌تر شدن مسئله به این مثال توجه شود: یک پروژه با ۱۰ فعالیت (شامل ۲ فعالیت مجازی) ۱ نوع منبع تجدیدناپذیر و ۱ نوع منبع تجدیدناپذیر را در نظر بگیرید. در این پروژه هر فعالیت، سه حالت اجرایی دارد. فعالیت‌های شروع و پایان، فعالیت‌های مجازی با یک حالت اجرایی زمان صفر و بدون احتیاج به منابع هستند. در پروژه مذکور، موجودیت منابع تجدیدناپذیر و تجدیدناپذیر، به ترتیب ۴ و ۲۵ است. اطلاعات تکمیلی و محدودیت‌های پیش‌نیازی این پروژه به ترتیب در جدول ۲ و شکل ۶ ارائه شده‌اند.

جدول ۲. اطلاعات فعالیت‌های پروژه مثال MRCPS

شماره‌ی فعالیت	حالت اجرایی ۱			حالت اجرایی ۲			حالت اجرایی ۳		
	r_{j2k}	n_{j2l}	d_{j2}	r_{j2k}	n_{j2l}	d_{j2}	r_{j2k}	n_{j2l}	d_{j2}
۰	۰	۰	۰	۰	۰	۰	۰	۰	۰
۱	۲	۵	۲	۲	۲	۴	۱	۲	۵
۲	۲	۴	۲	۲	۱	۵	۳	۳	۳
۳	۲	۱	۲	۲	۱	۲	۱	۱	۳
۴	۲	۵	۲	۳	۳	۴	۳	۳	۵
۵	۲	۶	۲	۳	۳	۵	۳	۳	۴
۶	۲	۱	۲	۴	۱	۲	۳	۱	۳
۷	۲	۴	۲	۲	۲	۲	۲	۲	۳
۸	۱	۳	۱	۳	۱	۴	۲	۲	۲
۹	۰	۰	۰	۰	۰	۰	۰	۰	۰



شکل ۶. شبکه‌ی پروژه برای مثال MRCPS

جدول ۳. فهرست اولویت فعالیت‌ها و حالات اجرایی.

فعالیت‌ها	۰	۱	۲	۳	۴	۵	۶	۷	۸	۹
اولویت فعالیت (λ)	۰	۰٫۹۵	۰٫۷۵	۰٫۶	۰٫۴	۰٫۵۵	۰٫۷۵	۰٫۸	۰٫۷	۰٫۰۷
حالت اجرایی فعالیت (μ)	۱	۱	۳	۲	۳	۲	۱	۳	۲	۱

جدول ۴. نتایج آماری الگوریتم ژنتیک پیشنهادی جهت حل مجموعه‌ی مسائل پایه (۵۰۰۰ زمان‌بندی تولید شده به‌عنوان معیار توقف لحاظ شده است).

مجموعه‌های پایه	راه‌حل‌های شدنی	انحراف متوسط (%)	نرخ بهینه (%)	زمان اجرای برنامه (ثانیه)
J10	۵۳۶	۰٫۰۲	۹۷٫۷۶	۰٫۱۰
J12	۵۴۷	۰٫۰۹	۹۱٫۹۲	۰٫۱۳
J14	۵۵۱	۰٫۱۶	۸۶٫۷۱	۰٫۱۵
J16	۵۵۰	۰٫۳۵	۷۴٫۹۸	۰٫۱۷
J18	۵۵۲	۰٫۳۶	۷۴٫۵۱	۰٫۱۸
J20	۵۵۴	۰٫۵۷	۶۸٫۰۱	۰٫۱۹

آزمایش الگوریتم ژنتیک پیشنهادی مورد استفاده قرار گرفته‌اند. این مجموعه‌ها، به ترتیب شامل ۱۰، ۱۲، ۱۴، ۱۶، ۱۸، ۲۰ و ۳۰ فعالیت غیرمجازی هستند. هرکدام از مجموعه‌های ذکرشده، ۶۴۰ نمونه دارند که برخی از آن‌ها نشدنی هستند. بنابراین، آن‌ها از آزمایش‌های این پژوهش حذف شده‌اند. ستون دوم در جدول ۴، تعداد راه‌حل‌های شدنی هر مجموعه را نشان می‌دهد. در هر نمونه، یک فعالیت می‌تواند در یکی از ۳ حالت اجرایی تعریف‌شده با زمان مشخصی بین ۱ تا ۱۰ واحد زمانی به اجرا درآید.

هر نمونه شامل دو نوع منبع تجدیدپذیر و دو نوع منبع تجدیدناپذیر است. جهت همه‌ی مجموعه‌های عنوان‌شده (به جزء J30)، زمان بهینه‌ی نمونه‌های شدنی شناخته شده‌اند. در واقع، زمان بهینه، جهت J30 تاکنون یافت نشده است.

۱.۶. تنظیم پارامترها

چون الگوریتم پیشنهادی جهت حل MRCPSPP، مشابه الگوریتم ژنتیک مندرج و همکاران (۲۰۰۹) است که جهت حل RCPSPP به کار گرفته شده (مشابهت در عمل‌گرهای کدگذاری، تقاطع و انتخاب) و نتایج امیدوارکننده توسط ایشان حاصل شده است، در این پژوهش نیز پارامترهای مندرج و همکاران (۲۰۰۹) اقتباس شده‌اند. از این رو، $J \times 5$ (تعداد فعالیت‌ها = J) به‌عنوان جمعیت اولیه و درصد افراد بالا، ۲۰ و نرخ جهش، ۱۰٪ تنظیم شده و مقدار SGS_Pper نیز، ۰٫۷ در نظر گرفته شده است.

۲.۶. مقایسه با سایر الگوریتم‌های موجود

نتایج آماری الگوریتم پیشنهادی در جدول ۴ نمایش داده شده است. انحراف متوسط از زمان بهینه (در مورد J30 انحراف از کمترین زمان یافت‌شده)، نرخ بهینه (درصد تعداد راه‌حل‌هایی که دقیقاً به جواب بهینه دست یافته‌اند) و زمان اجرای برنامه جهت تولید یک زمان‌بندی (به ثانیه) جهت مقایسه استفاده شده‌اند. برای مجموعه‌های J20 - J10 مقادیر بهینه و برای مجموعه‌ی J30 بهترین نتایج موجود (BKS) 27 در کتابخانه‌ی مسائل زمان‌بندی پروژه‌ها (تا ۲ می ۲۰۱۳) به‌عنوان پایه‌ی جهت محاسبه‌ی انحراف متوسط در نظر گرفته شده‌اند.

MM-FBI با زمان‌بندی پسروروی زمان‌بندی موجود آغاز می‌شود. برای این منظور، زمان پایان فعالیت‌ها در زمان‌بندی موجود به‌عنوان اولویت فعالیت‌ها در نظر گرفته می‌شود و فرایند بهبود حالت اجرایی در طی تولید این زمان‌بندی اعمال می‌شود. فرض کنید که در راه‌حل ذکرشده، فعالیت‌های ۳، ۴، ۵ و ۸ به‌طور تصادفی جهت اعمال فرایند بهبود حالت اجرایی در زمان‌بندی پسروروی انتخاب شده‌اند. همان‌طور که در شکل ۷ مشاهده می‌شود، فرایند بهبود حالت اجرایی، زمان انجام فعالیت‌های ۸ و ۴ را سه روز و زمان انجام فعالیت‌های ۵ و ۳ را یک روز کاهش می‌دهد. با این تغییرات میزان مصرف منبع تجدیدناپذیر از ۲۰ به ۲۵ (یعنی موجودیت منبع تجدیدناپذیر) افزایش می‌یابد. یک تکرار MM-FBI با زمان‌بندی پیشرو تکمیل می‌شود. جهت زمان‌بندی پیشرو، زمان آغاز فعالیت‌ها در زمان‌بندی پسروروی، به‌عنوان اولویت فعالیت‌ها در نظر گرفته می‌شود. اگر موجودیت منبع تجدیدناپذیر کاملاً استفاده نشده باشد، فرایند بهبود حالت اجرایی در این جهت نیز اعمال می‌شود. در این مثال، چون موجودیت منبع تجدیدناپذیر در فرایند پسروروی ذکرشده کاملاً استفاده شده است، فرایند بهبود حالت اجرایی در فرایند پیشرو اعمال نمی‌شود. پس از اتمام یک تکرار فرایند MM-FBI، زمان پروژه‌ی مثال مطرح شده از ۲۳ به ۱۸ کاهش می‌یابد (شکل ۷ ج). لازم به تذکر است که کارایی روش MM-FBI در کاهش زمان پروژه، توسط لوا و همکاران به اثبات رسیده است.^[۳۱]

۶. نتایج به‌دست‌آمده از الگوریتم ژنتیک پیشنهادی

در این بخش، با استفاده از نتایج حاصله، عملکرد الگوریتم ژنتیک پیشنهادی جهت حل MRCPSPP مورد بررسی قرار گرفته است. الگوریتم ژنتیک پیشنهادی با استفاده از برنامه‌ی مطلب کدنویسی شده و آزمایش‌ها بر روی رایانه‌ی ۲ هسته‌ی ۲۹۳۰۰ با پردازشگر ۲٫۷ گیگاهرتزی صورت پذیرفته است.

در ادبیات موضوع، جهت مقایسه‌ی الگوریتم‌های مختلف، زمان اجرای برنامه جهت تولید یک زمان‌بندی و یا تعداد زمان‌بندی تولیدشده، به‌عنوان معیار توقف در نظر گرفته می‌شوند. با توجه به اینکه پژوهشگران، بیشتر از ۵۰۰۰ زمان‌بندی تولیدشده جهت معیار توقف استفاده کرده‌اند، در این پژوهش نیز از معیار توقف مذکور استفاده شده است و حسن استفاده از معیار توقف ذکرشده این است که نتایج وابسته به قدرت پردازش رایانه‌ها نیستند.

مطابق تعریف لوا و همکاران (۲۰۰۹) و پتگهم و ونهوک (۲۰۱۰)، یک زمان‌بندی تولیدی این‌گونه تعریف می‌شود که برای هر فعالیت، فقط یک زمان شروع‌شدنی محاسبه شود.^[۳۱] لذا، یک زمان‌بندی تولیدی، توسط فرایند MSSGS، مطابق معادله‌ی ۸ محاسبه می‌شود:

$$\frac{\sum_{j=1}^J (SGS_Pper * M_j + (1 - SGS_Pper))}{J} = 1 + 2 * SGS_Pper \quad (8)$$

مجموعه‌ی مشهور و پایه‌ی: J10، J12، J14، J16، J18، J20 و J30 که در سامانه‌ی <http://129.187.106.231/psplib/> موجود هستند، جهت

جدول ۵. انحراف متوسط از راه حل بهینه (۵۰۰۰ زمان بندی تولید شده به عنوان معیار توقف لحاظ شده است).

رتبه	الگوریتم	J۲۰	J۱۸	J۱۶	J۱۴	J۱۲	J۱۰
۱	الگوریتم ژنتیک پژوهش حاضر	۰٫۵۷	۰٫۳۶	۰٫۳۵	۰٫۱۶	۰٫۰۹	۰٫۰۲
۲	VPVGA	۰٫۵۷	۰٫۴۲	۰٫۳۲	۰٫۲۲	۰٫۰۹	۰٫۰۱
۳	VPVAIS	۰٫۷۰	۰٫۵۲	۰٫۳۹	۰٫۲۰	۰٫۰۷	۰٫۰۲
۴	LHGA	۰٫۸۷	۰٫۶۳	۰٫۴۴	۰٫۳۲	۰٫۱۷	۰٫۰۶
۵	LCEDA	۱٫۲۸	۰٫۹۰	۰٫۵۹	۰٫۴۳	۰٫۱۴	۰٫۱۲
۶	LCSFLA	۱٫۴۰	۰٫۹۴	۰٫۵۸	۰٫۴۶	۰٫۲۱	۰٫۱۰
۷	EFEA	۱٫۶۲	۱٫۳۰	۰٫۹۱	۰٫۷۷	۰٫۲۴	۰٫۱۴
۸	RSS	۱٫۶۴	۱٫۲۱	۰٫۹۵	۰٫۸۹	۰٫۶۵	۰٫۱۸
۹	TCGLS	۱٫۶۹	۱٫۳۲	۱٫۰۸	۰٫۹۳	۰٫۵۲	۰٫۳۳
۱۰	AGA	۱٫۹۱	۱٫۴۳	۱٫۱۲	۱٫۰۰	۰٫۷۳	۰٫۲۴
۱۱	JSA	۶٫۷۴	۵٫۵۲	۴٫۰۷	۲٫۶۰	۱٫۷۳	۱٫۱۶

جدول ۶. انحراف متوسط از بهترین نتایج موجود جهت مجموعه‌ی J۳۰ (۵۰۰۰ زمان بندی تولید شده به عنوان معیار توقف لحاظ شده است).

الگوریتم	انحراف متوسط (%)	زمان اجرا (ثانیه)	برابر	بدر
الگوریتم ژنتیک پژوهش حاضر ^a	۰٫۹۷	۰٫۲۷ ^c	۵۶٫۲	۴۳٫۸
VPVGA ^b	۱٫۰۸	۰٫۲۴ ^d	۷۱٫۰	۲۹

^a نتایج موجود در کتابخانه‌ی مسائل زمان بندی پروژه‌ها در می ۲۰۱۳.

^b نتایج موجود در کتابخانه‌ی مسائل زمان بندی پروژه‌ها در فوریه ۲۰۰۹.

^c T۹۳۰۰ ۲٫۵ GHz.

^d Pentium ۲٫۸ GHz

بر اساس ملاحظات عنوان شده، الگوریتم پیشنهادی با بهترین الگوریتم‌های موجود در زمینه‌ی MRCPSP، نظیر: الگوریتم ژنتیک و سیستم هوش مصنوعی پتنگم و ون هوک (۲۰۱۰ و ۲۰۰۹) که به ترتیب با VPVGA و VPVAIS نمایش داده شده‌اند،^[۱۷،۱۶] و EDA و SFLA ارائه شده توسط وانگ و فانگ (۲۰۱۲) و (۲۰۱۱) که به ترتیب با LCEDA و LCSFLA نمایش داده شده‌اند،^[۲۵،۲۶] الگوریتم ژنتیک ترکیبی لوا و همکاران (۲۰۰۹) که با LHGA نمایش داده شده است،^[۲۱] الگوریتم تکاملی ترکیبی الوم و فرتمیس (۲۰۱۰) که با EFEA نمایش داده شده است،^[۲۲] جستجوی ترکیبی پراکنده‌ی رنجبر و همکاران (۲۰۰۹) که با RSS نمایش داده شده است،^[۲۳] الگوریتم ژنتیک ارائه شده توسط آلکارز و همکاران (۲۰۰۳) که با AGA نمایش داده شده است،^[۲۰] الگوریتم دو فازه‌ی ژنتیک جستجوی محلی تسنگ و چن (۲۰۰۹) که با TCGLS نمایش داده شده است،^[۲۷] و الگوریتم شبیه‌سازی تبرید جوزفسکا و همکاران (۲۰۰۱) که با JSA نمایش داده شده است،^[۱۸] مقایسه شده است.

مقایسه‌ی مربوط به انحراف از متوسط در جدول ۵ انجام و الگوریتم‌ها با توجه به عملکردشان رتبه بندی شده‌اند. همان‌طور که مشاهده می‌شود، در J۱۰ و J۱۶ فقط VPVGA و در J۱۲ فقط VVAIS عملکرد بهتری نسبت به الگوریتم ژنتیک پیشنهادی دارد. در ضمن، با افزایش شمار فعالیت‌ها (به‌جز J۱۶) در مجموعه‌های J۱۴، J۱۸ و J۲۰ الگوریتم پیشنهادی عملکرد بهتری نسبت به سایر روش‌ها دارد. این مسئله نشان می‌دهد که الگوریتم پیشنهادی با افزایش پیچیدگی مسئله، عملکرد بهتری از خود نمایش می‌دهد. مقایسه‌ی الگوریتم پیشنهادی با VPVGA در جدول ۶ نیز این ادعا را تأیید می‌کند. در جدول ۶، اگرچه انحراف متوسط برای VPVGA با استفاده از بهترین نتایج موجود سال

جدول ۸، مقایسه‌ی مربوط به نرخ بهینه را نمایش می‌دهد. همان‌طور که مشاهده می‌شود، نرخ بهینه‌ی الگوریتم ژنتیک پیشنهادی، نسبت به سایر الگوریتم‌ها مناسب نیست. اما، علی‌رغم نرخ بهینه‌ی نامناسب الگوریتم پیشنهادی، نتایج این الگوریتم نشان می‌دهد که بیشتر راه‌حل‌های یافت شده، نزدیک به راه‌حل بهینه هستند. برای مثال در مجموعه‌ی J۱۰، ۵۲۴ مسئله از ۵۳۶ مسئله به طور بهینه حل شده‌اند و در ۱۱ نمونه اختلاف راه‌حل یافت شده نسبت به راه‌حل بهینه‌ی یک دوره (روز) است و فقط در یک نمونه، این اختلاف برابر ۲ دوره (روز) است. بنابراین الگوریتم پیشنهادی روی همه‌ی مسائل J۱۰، به خوبی عمل کرده است. همچنین، این الگوریتم روی ۱۰۰٪ مسائل J۱۲، ۹۹٪ مسائل J۱۴، ۹۷٪ مسائل J۱۸ و ۹۳٪ مسائل J۲۰، به خوبی عمل کرده است. شایان ذکر است که در این پژوهش، عملکرد خوب،

جدول ۷. مقایسه با سایر الگوریتم‌های پیشنهادی، معیار مقایسه = زمان اجرای برنامه (۵۰۰۰ زمان‌بندی تولید شده به‌عنوان معیار توقف لحاظ شده است).

الگوریتم	J۲۰	J۱۸	J۱۶	J۱۴	J۱۲	J۱۰
الگوریتم ژنتیک پژوهش حاضر ^a	۰٫۱۹	۰٫۱۸	۰٫۱۷	۰٫۱۵	۰٫۱۳	۰٫۱۰
VPVGA ^b	۰٫۱۷	۰٫۱۶	۰٫۱۵	۰٫۱۴	۰٫۱۳	۰٫۱۲
LHGA ^c	۰٫۱۵	۰٫۱۳	۰٫۱۲	۰٫۱۱	۰٫۱۰	۰٫۰۸
LCSFLA ^d	۰٫۲۷	۰٫۱۹	۰٫۱۵	۰٫۱۳	۰٫۰۹	۰٫۰۷

^a T۹۳۰۰ ۲٫۵ GHz.

^b Pentium ۲٫۸ GHz.

^c Pentium ۳ GHz.

^d T۷۵۰۰ ۲٫۲ GHz.

جدول ۸. مقایسه با سایر الگوریتم‌های پیشنهادی، معیار مقایسه: نرخ بهینه (۵۰۰۰ زمان‌بندی تولید شده به‌عنوان معیار توقف لحاظ شده است).

الگوریتم	J۲۰	J۱۸	J۱۶	J۱۴	J۱۲	J۱۰
الگوریتم ژنتیک پژوهش حاضر	۶۸٫۰۱	۷۴٫۵۱	۷۴٫۹۸	۸۶٫۷۱	۹۱٫۹۲	۹۷٫۷۶
VPVGA	۸۵٫۷۴	۸۸٫۹۵	۹۲٫۰۰	۹۴٫۵۶	۹۸٫۱۷	۹۹٫۶۳
LHGA	۸۰٫۳۲	۸۴٫۹۶	۹۰٫۰۰	۹۲٫۹۲	۹۶٫۵۳	۹۸٫۵۱
LCSFLA	۷۲٫۸۴	۷۹٫۴۴	۸۶٫۴۹	۹۰٫۸۶	۹۵٫۹۸	۹۷٫۹۳
JSA	۳۵٫۷۰	۴۳٫۵۰	۵۴٫۷۰	۶۶٫۴۰	۸۰٫۳۰	۸۵٫۶۰

آزمایش الگوریتم پیشنهادی، مجموعه‌های مشهور و پایه‌ی J۱۰، J۱۲، J۱۴، J۱۶، J۱۸، J۲۰ و J۳۰ که در کتابخانه‌ی مسائل زمان‌بندی پروژه‌ها (PSBLIB) موجود هستند، مورد استفاده قرار گرفته‌اند. مقایسه‌های انجام‌شده نشان داده است که الگوریتم ژنتیک پیشنهادی بسیار کارآمد است.

همان‌طور که در بخش مربوط به فرایند تکرار پیشرو پسروی چند حالت (MM-FBI) اشاره شده است، برای اینکه به همه‌ی فعالیت‌ها فرصت برابر جهت کاهش زمان پروژه داده شود (به کمک تغییر حالت اجرایی آن فعالیت)، از یک فرایند تصادفی استفاده شده است؛ اما پژوهشگران این پژوهش معتقدند که می‌توان با ارائه‌ی الگوریتمی، فعالیت‌هایی با پتانسیل بیشتر جهت کاهش زمان پروژه را انتخاب کرد تا بدین ترتیب هم زمان اجرای پروژه و هم زمان اجرای برنامه بهبود یابد. مورد عنوان‌شده جهت مطالعات آینده در این زمینه پیشنهاد می‌شود.

از طرفی دیگر، مدل ارائه‌شده، مسئله‌ی تک هدفه‌ی MRCPSP را با هدف کاهش زمان پروژه مورد بررسی قرار داده است؛ لذا پیشنهاد می‌شود در مطالعات آینده، مدل ارائه‌شده جهت حل مسائل چندهدفه‌ی MRCPSP به‌کار گرفته شود.

راه‌حلی را شامل می‌شود که بیشینه‌ی اختلاف آن‌ها از راه حل بهینه ۲ دوره (روز) باشد.

در پایان، با توجه به مقایسه‌های صورت‌گرفته، می‌توان نتیجه گرفت که الگوریتم ژنتیک پیشنهادی بسیار کارآمد است.

۷. نتیجه‌گیری

در این نوشتار، الگوریتم ژنتیک توسعه‌یافته‌ی جهت حل MRCPSP به‌کار گرفته شده است، که در آن روش نمایش کلید تصادفی و فهرست حالات اجرایی مربوط به‌عنوان روش رمزگشایی استفاده و روش تولید زمان‌بندی سری چندحالتی جهت رمزگشایی در نظر گرفته شده است. یکی از نوآوری‌های این پژوهش، تعریف عمل‌گر جهش جدیدی برای راه‌حل‌های شدنی و نشدنی بوده است. همچنین تابع تناسب جدیدی جهت کاهش زمان اجرای برنامه و انحراف از متوسط ارائه شده است. جهت

پانوشته‌ها

1. project scheduling
2. project planning
3. resource constrained project scheduling problem
4. multi-mode resource constrained project scheduling problem
5. linear programming approach
6. artificial immune system
7. simulated annealing

8. particle swarm optimization
9. ant colony
10. a multi-agent learning approach
11. shuffled frog-leaping algorithm
12. Taguchi method of design of experiment
13. estimation of distribution algorithm
14. critical path method
15. Matlab
16. non-efficient and non-excusable modes
17. redundant non-renewable resources

18. project scheduling problem library (PSPLIB)
19. mating pool
20. top individuals
21. elicit strategy
22. infeasible tackling procedure and feasible tackling procedure
23. multi-mode Serial schedule generation scheme
24. multi-mode forward backward iteration
25. forward-backward iteration
26. mode improvement procedure
27. best known solutions

منابع (References)

1. Hartman, S. and Briskorn, D. "A survey of variants and extensions of the resource-constrained project scheduling problem", *European Journal of Operational Research*, **207**(1), pp. 1-14 (2009).
2. Kolisch, R. and Drexel, A. "Local search for non-preemptive multi-mode resource constrained project scheduling", *IIE Transactions*, **29**(11), pp. 987-999 (1997).
3. Slowinski, R. "Two approaches to problems of resource allocation among project activities: A comparative study", *Journal of Operational Research*, **31**(8), pp. 711-723 (1980).
4. Talbot, F.B. "Resource-constrained project scheduling with time-resource tradeoffs: The non-preemptive case", *Management Science*, **28**(10), pp. 1197-1210 (1982).
5. Patterson, J.H., Slowinski, R., Talbot, F. and Weglarz, J. "An algorithm for a general class of precedence and resource constrained scheduling problems", In: Slowin' ski, R., Weglarz, J. (Eds.), *Advances in Project Scheduling*, pp. 3-28 (1989).
6. Sprecher, A. "Resource-constrained project scheduling: Exact methods for the multi-mode case", *Lecture Notes in Economics and Mathematical Systems*, **409**, Springer, Berlin (1994).
7. Speranza, M.G. and Vercellis, C. "Hierarchical models for multi-project planning and scheduling", *European Journal of Operational Research*, **64**(2), pp. 312-325 (1993).
8. Sprecher, A., Hartmann, S. and Drexel, A. "An exact algorithm for project scheduling with multiple modes", *OR Spektrum: Organ der Deutschen Gesellschaft für Operations Research*, **19**(3), pp. 195-203 (1997).
9. Demeulemeester, E. and Herroelen, W.S. "A branch-and-bound procedure for the multiple resource-constrained project scheduling problems", *Management Science*, **38**(12), pp. 1803-1818 (1992).
10. Hartmann, S. and Drexel, A. "Project scheduling with multiple modes: A comparison of exact algorithms", *Networks*, **32**(4), pp. 283-297 (1998).
11. Sprecher, A. and Drexel, A. "Solving multi-mode resource-constrained project scheduling problems by a simple, general and powerful sequencing algorithm", *European Journal of Operational Research*, **107**, pp. 431-450 (1998).
12. Drexel, A. and Grünewald, J. "Nonpreemptive multi-mode resource-constrained project scheduling", *IIE Transactions*, **25**, pp. 74-81 (1993).
13. Slowinski, R., Soniewicki, B. and Weglarz, J. "DSS for multiobjective project scheduling", *European Journal of Operational Research*, **79**(2), pp. 220-229 (1994).
14. Lova, A., Tormos, P. and Barber, F. "Multi-mode resource constrained project scheduling: Scheduling schemes, priority rules and mode selection rules", *Inteligencia Artificial*, **10**(30), pp. 69-86 (2006).
15. Glover, F. and Greenberg, H.J. "New approaches for heuristic search: A bilateral linkage with artificial intelligence", *European Journal of Operational Research*, **39**(2), pp. 119-130 (1989).
16. Peteghne, V.V. and Vanhoucke, M. "An artificial immune system for the multi-mode resource-constrained project scheduling problem", in: *EvoCOP*, Springer, Berlin, Tübingen, Germany (2009).
17. Peteghne, V.V. and Vanhoucke, M. "A genetic algorithm for the preemptive and non-preemptive multi-mode resource-constrained project scheduling problem", *European Journal of Operational Research*, **201**(2), pp. 409-418 (2010).
18. Jo'zefowska, J., Mika, M., Rozycki, R., Waligora, G. and Weglarz, J. "Simulated annealing for multi-mode resource-constrained project scheduling", *Annals of Operations Research*, **102**(1), pp. 137-155 (2001).
19. Zhang, H., Li, X., Li, H. and Huang, F. "Particle swarm optimization-based schemes for resource-constrained project scheduling", *Automation in Construction*, **14**(3), pp. 393-404 (2005).
20. Ranjbar, M., Reyck, B. and De Kianfar, F. "A hybrid scatter search for the discrete time/resource trade-off problem in project scheduling", *European Journal of Operational Research*, **193**(1), pp. 35-48 (2009).
21. Zhang, H. "Ant colony optimization for multimode resource-constrained project scheduling", *American Society of Civil Engineers*, **28**(2), pp. 150-159 (2012).
22. Damak, N., Jarboui, B., Siarry, P. and Loukil, T. "Differential evolution for solving multi-mode resource constrained project scheduling problems", *Computers and Operations Research*, **36**(3), pp. 2653-2659 (2009).
23. Wauters, T., Verbeeck, K., Berghe, G. and De Causmaecker, P. "A multi-agent learning approach for the multi-mode resource-constrained project scheduling problem", In: *Proceedings of the 8th International Conference on Autonomous Agents and Multi Agent Systems*, pp. 1-8 (2009).
24. Elloumi, S. and Fortemps, P. "A hybrid rank-based evolutionary algorithm applied to multi-mode resource-constrained project scheduling problem", *European Journal of Operational Research*, **205**(1), pp. 31-41 (2010).
25. Wang, L. and Fang, C. "An effective shuffled frog-leaping algorithm for multi-mode resource-constrained project scheduling problem", *Information Sciences*, **181**(20), pp. 4804-4822 (2011).
26. Wang, L. and Fang, C. "An effective estimation of distribution algorithm for the multi-mode resource-constrained project scheduling problem", *Computers and Operations Research*, **39**(2), pp. 449-460 (2012).

27. Mori, M. and Tseng, C. "A genetic algorithm for multi-mode resource constrained project scheduling problem", *European Journal of Operational Research*, **100**(1), pp. 134-141 (1997).
28. Özdamar, L. "A genetic algorithm approach to a general category project scheduling problem", *IEEE Transactions on Systems, Man, and Cybernetics, Part C, Applications and Reviews*, **29**(1), pp. 44-59 (1999).
29. Hartman, S. "Project scheduling with multiple modes: A genetic algorithm", *Annals of Operations Research*, **102**(1), pp. 111-135 (2001).
30. Alcaraz, J., Maroto, C. and Ruiz, R. "Solving the multi-mode resource-constrained project scheduling problem with genetic algorithms", *Journal of the Operational Research Society*, **54**, pp. 614-626 (2003).
31. Lova, A., Tormos, P., Cervantes, M. and Barber, F. "An efficient hybrid genetic algorithm for scheduling projects with resource constraints and multiple execution modes", *International Journal of Production Economics*, **117**(2), pp. 302-316 (2009).
32. Lee, K. and El-Sharkawi, M., *Modern Heuristic Optimization Techniques*, Hoboken, New Jersey, John Wiley and Sons (2008).
33. Mendes, J., Gonçalves, J. and Resende, M. "A random key based genetic algorithm for the resource constrained project scheduling problem", *Computers and Operations Research*, **36**(1), pp. 92-109 (2009).
34. Spears, W. and Dejong, K. "On the virtues of parameterized uniform crossover", *In: Proceedings of the Fourth International Conference on Genetic Algorithms*, pp. 230-6 (1991).
35. Kolisch, R. "Serial and parallel resource-constrained project scheduling methods revisited: Theory and computation", *European Journal of Operational Research*, **90**(2), pp. 320-333 (1996).
36. Tormos, P. and Lova, A. "A competitive heuristic solution technique for resource-constrained project scheduling", *Annals of Operations Research*, **102**(1), pp. 65-81 (2001).
37. Tseng, L.Y. and Chen, S.C. "Two-phase genetic local search algorithm for the multi-mode resource-constrained project scheduling problem", *IEEE Transactions on Evolutionary Computation*, **13**(4), pp. 848-57 (2009).