

ارائه الگوریتم الکترومغناطیس تلفیقی برای مسأله زمان‌بندی پروژه با منابع محدود و فعالیت‌های چند حالت

محمد حسین صادقی^۱، رضا توکلی مقدم^{۲*}

۱- کارشناس ارشد مهندسی صنایع، پردیس دانشکده‌های فنی، دانشگاه تهران

۲- استاد گروه مهندسی صنایع، پردیس دانشکده‌های فنی، دانشگاه تهران

چکیده

در این مقاله، مسأله زمان‌بندی پروژه با منابع محدود و فعالیت‌های چند حالت (یعنی امکان انتخاب روش‌های اجرایی مختلف برای فعالیت‌ها)، برای حل به دو زیر مسأله تقسیم می‌شود: تخصیص روش‌های اجرایی به فعالیت‌ها و سپس زمان‌بندی فعالیت‌ها به منظور کمینه نمودن زمان اتمام پروژه. روش الکترومغناطیس^۱ با مسأله اول در ارتباط بوده و فهرست روش اجرای فعالیت‌ها را تولید می‌کند. پس از تعیین روش اجرایی هر فعالیت، زمان و مصرف منابع آن فعالیت بر اساس روش انتخاب شده برای اجرای آن تعیین و یک برنامه زمان‌بندی تصادفی به روش سری برای آن ایجاد می‌گردد. سپس یک روش جستجوی محلی نسبت به بهبود برنامه اقدام می‌کند. ضمناً در مقاله، یک تابع جریمه جدید برای فهرست‌های روش نشدنی از نظر منابع تجدیدناپذیر پیشنهاد می‌شود. عملکرد روش حل پیشنهادی با بهترین روش‌های حل پیشنهاد شده تاکنون برای این مسأله بر اساس معیارهای توقف زمان حل و تعداد برنامه‌های زمان‌بندی تولید شده مقایسه می‌گردد که نتایج گزارش شده، گویای عملکرد عالی این روش است.

واژه‌های کلیدی: زمان‌بندی پروژه با منابع محدود، منابع تجدیدناپذیر، حداقل نمودن طول زمان اجرای پروژه، روش الکترومغناطیس، جستجوی محلی.

۱- مقدمه

در مسأله کلاسیک زمانبندی پروژه با منابع محدود (RCPSP)^۲ هدف پیدا کردن بهترین برنامه زمانبندی فعالیت‌ها با حداقل زمان اتمام پروژه است. محدودیت‌های پیشین‌سازی بین فعالیت‌ها و حداکثر مقدار در دسترس هر یک از منابع تجدیدپذیر در دوره‌های زمانی اجرای پروژه دو دسته محدودیت در زمانبندی فعالیت‌ها در این مسأله هستند که باید در نظر گرفته شوند. هر فعالیت پروژه یک روش اجرایی دارد و زمان فعالیت و نیازمندی‌های لازم برای یک مجموعه از منابع ثابت فرض می‌شود. شبکه پروژه به صورت AON ارایه می‌شود که می‌توان آن را به صورت گراف $G = (V, E)$ نشان داد. در این گراف گره‌ها نشان دهنده فعالیت‌ها و کمان‌ها نمایانگر روابط پیشین‌سازی هستند. اگر کمان (i, j) در شبکه وجود داشته باشد، آنگاه فعالیت i باید قبل از شروع فعالیت j به پایان برسد.

مجموعه فعالیت‌های پروژه را می‌توان به صورت $V = \{1, 2, \dots, J\}$ تعریف کرد که در آن 1 و J فعالیت‌های مجازی شروع و اتمام پروژه هستند. این فعالیت‌ها دارای زمان اجرا و مصرف منابع صفر هستند و در واقع، نقش واقعه را دارند. در مسأله زمانبندی پروژه با منابع محدود و فعالیت‌های چند حالتی (یعنی امکان انتخاب روش‌های اجرایی مختلف برای فعالیت‌ها) (MRCPSP)^۳ (المقربسی^۴، ۱۹۷۷)، برای هر فعالیت مجموعه‌ای از روش‌های اجرایی قابل انتخاب می‌باشد که این موضوع تفاوت اساسی این مسأله با RCPSP است. ضمناً با توجه به وجود روش‌های مختلف اجرایی برای هر فعالیت امکان در نظر گرفتن منابع تجدید ناپذیر نیز در این مسأله

فراهم شده است. در صورت انتخاب هر یک از روش‌های اجرایی، میزان مصرف منابع تجدید پذیر، منابع تجدید ناپذیر و طول زمان اجرا برای هر فعالیت متفاوت خواهد بود. برای مثال، یک کارگر ممکن است کاری را در ۱۰ ساعت انجام دهد (حالت ۱) و در مقابل، دو کارگر همان فعالیت را در ۵ ساعت به انجام برسانند (حالت ۲). شایان ذکر است که این نسبت لزوماً خطی تغییر نمی‌کند و حتی ممکن است منابعی که برای انجام دو روش متفاوت یک فعالیت صرف می‌شوند، با یکدیگر فرق داشته باشند، مثلاً استفاده از ماشین آلات خاصی بتواند زمان انجام فعالیت را حتی تا یک ساعت کاهش دهد (حالت ۳). فرض می‌شود که زمان اجرای فعالیت‌ها پیوسته است؛ یعنی قطع فعالیت پس از شروع و پیش از اتمام آن امکان‌پذیر نیست. ضمناً پس از انتخاب روش اجرایی فعالیت، نمی‌توان هنگام انجام فعالیت آن را تغییر داد.

به زبان ریاضی در مسأله زمانبندی پروژه با منابع محدود و فعالیت‌های چند حالتی، برای هر فعالیت $j \in V$ از شبکه $G = (V, E)$ یک مجموعه M_j از روش‌های اجرایی در نظر گرفته می‌شود. به محض اینکه فعالیت در روش $m \in M_j$ آغاز شد، بدون وقفه و بدون تغییر آن باید پایان یابد. مدت زمان انجام فعالیت j طبق روش m است و برحسب کوچکترین واحد زمانی پروژه تعریف می‌شود. یک مجموعه R^p از منابع تجدید پذیر و یک مجموعه R^v از منابع تجدید ناپذیر، آماده به کارگیری به منظور اجرای فعالیت‌های پروژه هستند. T حد بالای زمان اتمام پروژه بوده، از جمع حداکثر زمان اجرای فعالیت‌ها به دست می‌آید. در واقع، به منظور

حد ارایه نمودند. ژو و همکاران^{۱۱} (۲۰۰۶) یک روش شاخه و برش ارایه نمود. به هر حال هیچ‌کدام از این روش‌ها نمی‌تواند برای حل مسایل بزرگ استفاده شود زیرا قادر به یافتن جواب بهینه در یک مدت زمان منطقی نیستند. به همین خاطر، روش‌های ابتکاری و فرا ابتکاری مورد توجه قرار گرفته‌اند.

روش‌های ابتکاری

تالبوت^{۱۲} (۱۹۸۲) و اسپرچر و درکسل^{۱۳} (۱۹۹۸) پیشنهاد نمودند که برای روش شاخه و حدشان یک حد زمانی قرار دهند. بوکتور^{۱۴} (۱۹۹۳) ۲۱ قاعده ابتکاری برای زمان‌بندی را آزمود و تلفیقی از ۵ روش که احتمال بیشتری برای تولید بهترین جواب را داشت، پیشنهاد نمود. اوزدامار و اولوسوی^{۱۵} (۱۹۹۴) یک روش بررسی محلی بر مبنای محدودیت ارایه نمود. بوکتور^{۱۶} (۱۹۹۶a) یک روش ابتکاری بر مبنای روش محاسبه مسیر بحرانی^{۱۷} ارایه کرد. کولیش و درکسل^{۱۸} (۱۹۹۷) یک روش جستجوی محلی با سه فاز پیشنهاد نمود. لووا و همکاران^{۱۹} (۲۰۰۶) چندین روش چند مسیره بر اساس قواعد اولویت طراحی نمودند.

روش‌های فرا ابتکاری

اسلووینسکی و همکاران^{۲۰} (۱۹۹۴) یک روش تک مسیره، یک روش چند مسیره و یک روش شبیه سازی تبرید^{۲۱} ارایه نمودند. بوکتور^{۲۲} (۱۹۹۶b) همچنین توسعه روش شبیه سازی تبرید خود برای حالت تک روش اجرایی (حالت) را برای حالت چند روش اجرایی، ولی بدون در نظر گرفتن منابع تجدید ناپذیر ارایه نمود. بولیمن و لکوک^{۲۳} (۲۰۰۳) نیز روشی را که به منظور حل مسایل تک روش اجرایی

محاسبه این مقدار فرض می‌شود برای کلیه فعالیت‌ها روش اجرایی انتخاب می‌شود که دارای طولانی‌ترین زمان بوده و هیچ دو فعالیتی همزمان در حال اجرا نبوده باشد و همه فعالیت‌ها در امتداد هم اجرا می‌شوند. فرض می‌شود که از منبع تجدیدپذیر k در هر یک از پیوندهای $t = 1, \dots, T$ به میزان R_k^p واحد موجود است. همچنین R_k^v واحد از منبع تجدید ناپذیر k در کل زمان پروژه در دسترس است. اگر فعالیت j ام به روش m ام انجام شود، نیاز به r_{jkm}^p واحد از منبع تجدیدپذیر k در پیوندهای $t = 0, 1, \dots, T$ و r_{jkm}^v واحد از منبع تجدید ناپذیر k نیاز دارد. هدف مدل MRCPSPP انجام کلیه فعالیت‌های پروژه در فقط و فقط یکی از روش‌های ممکن است، به گونه‌ای که محدودیت منابع و روابط پیشنهادی ارضا گردیده و کوتاهترین زمان اتمام پروژه حاصل شود.

تاکنون روش‌های متفاوتی برای حل مسأله زمان‌بندی پروژه با منابع محدود و فعالیت‌های چند حالتی پیشنهاد شده است که می‌توان آنها را در سه دسته کلی روش‌های دقیق، ابتکاری و فرا ابتکاری دسته‌بندی نمود.

روش‌های دقیق

نخستین روش حل برای این مسأله که رویکرد برنامه ریزی خطی یک مرحله‌ای و دو مرحله‌ای بود، توسط اسلووینسکی^۵ (۱۹۸۰) ارایه شد. تالبوت^۶ (۱۹۸۲) و پترسون و همکاران^۷ (۱۹۸۹) یک روش شمارشی برای حل مدل پیشنهاد نمودند. اسپرچر و همکاران^۸ (۱۹۹۷)، هارتمن و درکسل^۹ (۱۹۹۸) و اسپرچر و درکسل^{۱۰} (۱۹۹۸) الگوریتم‌های شاخه و

دارد، ارایه نمودند. آنها در این الگوریتم دو جمعیت را در نظر گرفتند. آنها همچنین، با اعمال تغییراتی در روش تولید برنامه زمانبندی سری، فرآیند انتخاب روش اجرایی برای فعالیت را با انتخاب آن روش شدنی که زمان پایان فعالیت را کمینه می‌کند، گسترش دادند.

ژانگ و همکاران^{۳۲} (۲۰۰۶) یک الگوریتم با استفاده از ویژگی های روش دسته ذرات^{۳۳} برای حل مسأله ارایه نمودند. جاربوی و همکاران^{۳۴} (۲۰۰۸) الگوریتم اصلی را توسعه داده، آن را به منظور حل مسایل بهینه‌سازی ترکیبیاتی^{۳۵} با مقادیر صحیح ارایه نمودند که در مواردی با روش اصلی متفاوت بود. آنها کیفیت بالاتر روش پیشنهادی خود را نسبت به روش ژانگ و همکاران^{۳۶} (۲۰۰۶) با مقایسه نتایج نشان دادند.

رنجبر و همکاران^{۳۷} (۲۰۰۹) یک روش جستجوی پراکنده تلفیقی^{۳۸} را برای حل مسأله DTRTP^{۳۹} با چندین نوع منبع (MDTRTP)^{۴۰} توسعه دادند. این مسأله حالت خاصی از مسأله MRCPSP است و تنها تفاوت آنها در وجود منابع تجدید ناپذیر است. آنها همچنین، به منظور حل مسأله MRCPSP تغییرات کوچکی نیز در روش حل خود ایجاد نمودند. در روش حل آنها جواب هایی که از نظر منابع تجدید ناپذیر، نشدنی بودند، با استفاده از تابع ارایه شده توسط الکاراز و همکاران^{۴۱} (۲۰۰۳) جریمه می‌شوند.

اکثر روش‌های تکاملی از نظر تشکیل جمعیت اولیه و استفاده از عملگرهای تقاطع^{۴۲} و جهش^{۴۳} به الگوریتم ژنتیک شباهت دارند. داماک و همکاران^{۴۴} (۲۰۰۹) یک روش تکاملی^{۴۵} وابسته به تفاوت ارایه نمودند. این روش عملگرهای حسابی را با

ارایه نموده بودند، با به کارگیری یک روش دو فازی برای حالت چند روش اجرایی گسترش دادند. آنها با استفاده از عملگرهای معرفی شده، در فاز اول، برای فهرستهای روش، همسایه پیدا کرد و سپس در فاز دوم همسایه‌هایی برای فهرست فعالیت‌های مرتبط با آن فهرست روش ایجاد می‌کنند. جوزفوسکا و همکاران^{۴۶} (۲۰۰۱) فرآیند ایجاد جواب همسایه یا مجاور با جواب فعلی را تنها در یک فاز دنبال نمودند. در روش پیشنهادی آنها سه روش تولید جواب همسایه؛ یعنی ایجاد فهرست روش مجاور، فهرست فعالیت مجاور و تلفیقی از دو روش قبلی به صورت تصادفی با احتمال معلوم انتخاب می‌شود.

تا کنون اشکال مختلفی از الگوریتم ژنتیک^{۴۵} به منظور حل مسأله برنامه ریزی پروژه با منابع محدود و فعالیت‌های چند حالتی (وجود چند روش اجرایی برای فعالیت‌ها) ارایه شده است. اوزدامار^{۴۶} (۱۹۹۹) یک الگوریتم ژنتیک بر اساس قواعد اولویت ارایه کرد. هارتمن^{۴۷} (۲۰۰۱) الگوریتم را بر اساس روش نمایش برنامه زمانبندی به شکل فهرست فعالیت‌ها و فهرست روش‌ها ارایه نمود. الکاراز و همکاران^{۴۸} (۲۰۰۳) با افزودن یک ژن به منظور نمایش روبه جلو و یا روبه عقب بودن برنامه ریزی، این شیوه را گسترش داد. لووا و جواب‌ها و استفاده همکاران^{۴۹} (۲۰۰۹) یک الگوریتم ژنتیکی تلفیقی ارایه نمودند. اصلی ترین نوآوری‌ها در این روش پیشنهادی، نحوه تخصیص روش‌ها به فعالیت‌ها، محاسبه تابع هدف برای از یک روش بهبود جواب بود. ون پتقم و ونهوک^{۳۰} (۲۰۱۰) یک الگوریتم ژنتیک به منظور حل مسأله برنامه‌ریزی پروژه در حالت چند روشی و توسعه آن در حالتی که امکان قطع^{۳۱} فعالیت‌ها وجود

این الگوریتم شامل چهار فاز اصلی راه اندازی، جستجوی محلی، محاسبه نیروی وارد بر هر ذره و حرکت در برای نیروست که در ادامه به بررسی کامل هر یک از آنها می‌پردازیم.

۲-۱- راه اندازی

در آغاز الگوریتم الکترومغناطیس، مشابه سایر روش‌های مبتنی بر جمعیت، m نقطه تصادفی از فضای شدنی مسأله انتخاب می‌شود. به منظور تولید یک نقطه تصادفی فرض می‌شود هر مؤلفه نقطه، یک متغیر تصادفی با تابع توزیع یکنواخت بین حد پایین و بالای خود بوده، سپس برای هر مؤلفه یک مقدار تصادفی انتخاب می‌گردد. پس از ایجاد این مجموعه، مقدار تابع هدف برای هر نقطه محاسبه و بهترین نقطه به عنوان x^{best} نشان داده می‌شود.

۲-۲- جستجوی محلی

این مرحله به منظور بررسی فضای اطراف هر یک از نقاط ایجاد شده صورت می‌پذیرد. بیربیل و فنگ^{۴۸} (۲۰۰۳) به ارایه یک الگوریتم ساده به منظور جستجوی محلی پرداختند. در روش ارایه شده از سوی آنها، برای هر یک از ابعاد نقاط ایجاد شده، قدمی تصادفی در برای افزایش و یا کاهش بعد مورد نظر برداشته می‌شود. برای این قدم به شکل تصادفی انتخاب می‌شود. همچنین، این قدم حداکثر می‌تواند با توجه به برای قدم تا حد بالا و یا پایین بعد مورد نظر برداشته شود. پس از اجرای این مرحله، اگر نقطه جدید ایجاد شده تابع هدف بهتری را فراهم آورد، با نقطه قبلی جایگزین شده، فرآیند جستجو برای بعد بعدی دنبال می‌گردد، در غیر این صورت، نقطه اولیه حفظ و جستجو برای همین بعد تکرار می‌گردد.

عملگرهای تقاطعی و جهشی کلاسیک ترکیب نموده است. در روش ارایه شده عملگرهای تقاطعی و جهشی به منظور تولید بردارهای آزمایشی جدید که پایه تولید جواب‌های جدید است، استفاده می‌گردد.

۲- الگوریتم الکترومغناطیس

این روش که نخستین بار توسط بیربیل و فنگ^{۴۶} (۲۰۰۳) پیشنهاد شد، همانند الگوریتم ژنتیک یک روش مبتنی بر جمعیت^{۴۷} است. در این روش هر نقطه یک ذره باردار در فضا فرض شده، مقدار بار آن نیز بر اساس مقدار تابع هدف آن تعیین می‌گردد. پس از تعیین بار هر نقطه از جمعیت، برآیند نیروی وارد بر نقاط و برای حرکت آنها در هر تکرار مشخص می‌گردد. همانند نیروهای الکترومغناطیسی، برآیند نیروی وارد بر هر نقطه، از جمع برداری تمام نیروهای وارد بر آن به دست می‌آید. این روش در برخی موارد تفاوت‌هایی با تئوری الکترومغناطیس دارد که در تشریح مراحل الگوریتم توضیح داده می‌شود. الگوریتم الکترومغناطیس به منظور حل مسایلی با متغیرهای حقیقی و محدود طراحی شده است. شکل کلی این گونه مسایل در زیر آورده شده است.

$$\min f(x)$$

$$\text{s.t. } x \in S$$

(۱)

$$S = \{x \in \mathbb{R}^n \mid -\infty < l_k \leq x_k \leq u_k < \infty, k = 1, \dots, n\}$$

که در آن n بعد مسأله، u_k حد بالا و l_k حد پایین بعد k ام بوده، $f(x)$ تابعی است که باید حداقل شود.

بهترین مقدار تابع هدف را دارد، سایر نقاط را جذب و نقطه ای که بدترین مقدار تابع هدف را دارد، سایر نقاط را دفع می‌کند. مقدار نیروی وارده از طرف نقطه j بر نقطه i به صورت زیر محاسبه می‌گردد:

$$F^{i,j} = \begin{cases} (x^j - x^i) \frac{q^i q^j}{\|x^j - x^i\|^2}, & \text{if } f(x^j) < f(x^i) \\ (x^i - x^j) \frac{q^i q^j}{\|x^j - x^i\|^2}, & \text{if } f(x^i) < f(x^j) \end{cases}, \quad i=1,2,\dots,m, j=1,2,\dots,m \quad (3)$$

نهایتاً بردار نیروی کل وارد بر هر نقطه از جمع برداری نیروهای وارد بر آن نقطه از طرف سایر نقاط حاصل می‌شود.

$$F^i = \sum_{j \neq i} F^{i,j}, \quad i=1,2,\dots,m \quad (4)$$

به منظور جلوگیری از همگرا شدن الگوریتم به یک جواب حداقل محلی و یا اصطلاحاً همگرایی زودرس یا پیش از موعد^۹، بیربیل و همکاران^۵ (۲۰۰۵) فاز محاسبه نیروی کل را کمی تغییر دادند. آنها دورترین نقطه از x^{best} را انتخاب کرده و آن را x^p نامیدند.

$$x^p = \arg \max \{ \|x^{best} - x^i\|, i=1,2,\dots,m \} \quad (5)$$

در این ویرایش جدید از الگوریتم، محاسبه نیروی کل برای همه نقاط به غیر از x^p بدون تغییر مانده، ولی اجزای تشکیل دهنده نیروی کل برای x^p به شکل زیر تغییر می‌کند تا حداقل یکی از نقاط شانس حرکت به سمت نواحی حذف شده از جمعیت را داشته باشد.

فرآیند جستجو حداکثر به تعداد $LSITER$ مرتبه برای هر بعد تکرار شود.

۲-۳- محاسبه بردار نیروی کل

بر اساس تئوری الکترومغناطیس، نیرویی که دو ذره باردار بر هم وارد می‌کنند، با مجذور فاصله آنها نسبت عکس و با مقدار بار هر یک از آنها نسبت مستقیم دارد. در این الگوریتم، در هر تکرار بار نقاط بر اساس رابطه زیر تعیین می‌گردد.

$$q^i = \exp \left(-n \frac{f(x^i) - f(x^{best})}{\sum_{k=1}^m (f(x^k) - f(x^{best}))} \right), \quad i=1,2,\dots,m \quad (2)$$

که در آن q^i میزان بار نقطه i ام، n تعداد ابعاد مسأله، $f(x^i)$ مقدار تابع هدف نقطه i ام و $f(x^{best})$ بهترین مقدار تابع هدف در تکرار مربوطه است. بار هر نقطه قدرت جذب یا دفع آن نقطه را تعیین کرده، نقاطی که دارای مقدار تابع هدف بهتری هستند، بار بیشتری خواهند داشت. بنابراین، در هر تکرار بیشترین بار متعلق به x^{best} خواهد بود. با توجه به فرمول مورد استفاده، بار یک نقطه ممکن است در دو تکرار مختلف متفاوت باشد. نکته قابل توجه اینکه برخلاف بارهای الکتریکی، بار نقاط در این روش هیچ علامتی نداشته، برای نیروی وارد بر هر جفت نقطه پس از مقایسه مقدار تابع هدف آنها تعیین می‌شود. بین هر جفت نقطه، نقطه ای که تابع هدف بهتری دارد، نقطه دیگر را جذب و نقطه ای که تابع هدف بدتری دارد، نقطه دیگر را دفع می‌کند و در صورتی که توابع هدف برابر باشند، نقاط نیرویی به هم وارد نخواهند کرد. لذا در هر تکرار نقطه‌ای که

یکنواخت بین صفر و یک بوده و RNG نیز از رابطه زیر به دست می‌آید:

$$RNG_k = \begin{cases} u_k - x_k^i & \text{if } F_k^i > 0 \\ x_k^i - l_k & \text{if } F_k^i < 0 \end{cases} \quad k = 1, 2, \dots, n \quad (8)$$

هر یک از مؤلفه‌های بردار RNG حداکثر حرکت مجاز به سمت u_k یا l_k را برای بعد مربوطه به منظور جلوگیری از تجاوز نقاط از حدود بالا و پایین تعیین می‌کنند. در صورتی که $F_k^i = 0$ ، حرکتی در بعد مورد نظر نخواهیم داشت. حاصل ضرب α و RNG برداری است تصادفی که حرکت نقطه با آن شدنی بودن نقطه جدید را تضمین می‌کند. میزان جا به جایی نقاط در راستای نیرو به این علت به شکل تصادفی انتخاب می‌شود که تمامی نقاط در راستای نیرو شانس بررسی را داشته باشند. در هر تکرار، x^{best} تنها نقطه‌ای است که جابه‌جا نشده و عیناً به تکرار بعد منتقل می‌شود، بنابراین، می‌توان از محاسبه نیروهای وارد بر این نقطه چشم‌پوشی کرد.

گلعلی‌خانی و همکاران^{۵۱} (۲۰۰۹) روش ترکیبی جدیدی بر اساس روش الکترومغناطیس پیشنهادی بیربیل و همکاران^{۵۲} (۲۰۰۵) و روش جستجوی محلی سولیس و وتس^{۵۳} (۱۹۸۱) برای مسائل بهینه‌سازی پیوسته ارائه نمودند. همچنین، ویرایش‌های مختلفی از روش شبه الکترومغناطیس توسط ناجی عظیمی و همکاران^{۵۴} (۲۰۱۰) برای حل مسأله پوشش مجموعه و جوادیان و همکاران^{۵۵} (۲۰۰۸ و ۲۰۰۹) برای حل مسأله فروشنده دوره‌گرد و زمان‌بندی تک ماشین استفاده شده است. دبلز و همکاران^{۵۶} (۲۰۰۶) نیز ترکیبی از روش

(۶)

$$F^{p,j} = \begin{cases} (x^j - x^i) \frac{\lambda q^p q^j}{\|x^j - x^p\|^2}, & \text{if } f(x^j) < f(x^p) \\ (x^p - x^j) \frac{\lambda q^p q^j}{\|x^j - x^p\|^2}, & \text{if } f(x^p) < f(x^j) \end{cases}, \quad j=1,2,\dots,m$$

در رابطه ۶، λ یک متغیر تصادفی یکنواخت در بازه (۰,۱) است. علاوه بر تغییر اندازه هر یک از نیروهای وارد بر x^p با ضرب λ ، برای نیروها نیز ممکن است تغییر کند. به همین منظور، یک متغیر تصادفی دیگر مانند U انتخاب شده و اگر $\lambda < U$ باشد؛ آنگاه برای آن نیرو عکس می‌گردد. نکته درخور توجه اینکه هر نقطه‌ای غیر از x^{best} می‌تواند به عنوان x^p انتخاب شود. در این روش، دورترین نقطه از بهترین نقطه در تکرار فعلی به عنوان x^p انتخاب شده است، چون نیروی جاذبه بهترین نقطه بر این نقطه کمتر از نیروی جاذبه بر نقاط دیگر است.

۲-۴- حرکت نقاط در راستای بردار نیروی کل پس از تعیین بردار نیروی کل برای تمام نقاط، هر یک از نقاط در راستای نیروی وارد به اندازه‌ای که به صورت تصادفی انتخاب می‌شود، حرکت می‌کند. مختصات نقاط جدید از رابطه زیر حاصل می‌شود:

$$x^i = x^i + \alpha(RNG) \frac{F^i}{\|F^i\|} \quad i=1,2,\dots,m \quad (7)$$

در رابطه ۷، از تقسیم بردار نیروی کل وارد بر هر نقطه بر اندازه آن، بردار راستای حرکت آن نقطه حاصل می‌شود که اندازه‌ای برابر با یک دارد. اندازه تصادفی بردار حرکت نیز از ضرب α و NRG حاصل می‌شود. α متغیری تصادفی با تابع توزیع

نمایش مراحل الگوریتم پیشنهادی، مسأله شماره j1012_1 از مجموعه مثال‌های استاندارد موجود در کتابخانه مسایل زمانبندی پروژه انتخاب و مراحل بر روی آن اعمال می‌گردد. این مجموعه مسایل در <http://129.187.106.231/psplib> در دسترس است. مسأله انتخابی، ۱۰ فعالیت غیر مجازی داشته که زمان هرکدام بین ۱ تا ۱۰ بوده و به دو منبع تجدید پذیر و دو منبع تجدید ناپذیر نیاز دارد. در جدول ۱ پسنیازهای هر فعالیت، زمان انجام، مصرف منابع آن و سطح در دسترس هر یک از منابع آورده شده است. ضمناً شایان ذکر است، سطح دسترس نخستین منبع تجدید ناپذیر از ۵۴ به ۴۱ و سطح دسترس دومین منبع تجدید ناپذیر از ۴۸ به ۳۵ کاهش داده شده است، زیرا با در نظر گرفتن حدود اصلی، هر فهرست روش تصادفی از نظر مصرف منابع تجدید ناپذیر قابل قبول و شدنی خواهد بود.

الکترومغناطیس و جستجوی پراکنده را برای حل مسأله زمانبندی پروژه با منابع محدود و فعالیت‌های چند حالت به کار گرفتند.

۳- الگوریتم پیشنهادی به منظور حل مدل

به منظور حل MRCPSP این مسأله به دو زیر مسأله تقسیم شده است. تعیین روش اجرایی هر فعالیت و سپس یافتن بهترین زمانبندی فعالیت‌ها به منظور حداقل نمودن زمان پروژه. با استفاده از روش الکترومغناطیس به هر فعالیت یکی از روش‌های ممکن تخصیص پیدا کرده و زمان اجرای هر فعالیت و میزان مورد نیاز از هر منبع با توجه به روش انتخاب شده برای اجرای آن تعیین می‌گردد. پس از آن به صورت تصادفی چند برنامه زمان بندی تولید شده و یک الگوریتم جستجوی محلی هر یک از برنامه‌ها را بهبود می‌دهد. سپس بهترین برنامه زمانبندی انتخاب و به عنوان زمان تکمیل متناظر با فهرست روش مربوطه گزارش می‌گردد. به منظور

جدول ۱- اطلاعات مسأله نمونه

فعالیت (j)	Mod 1						Mod 2				Mod 3				
	p_{j1}	r_{j1}^p	r_{j1}^v	p_{j2}	r_{j2}^p	r_{j2}^v	p_{j3}	r_{j3}^p	r_{j3}^v	p_{j4}	r_{j4}^p	r_{j4}^v	p_{j5}	r_{j5}^p	r_{j5}^v
۱	۴,۳,۲	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰
۲	۸	۱	۰	۹	۷	۰	۷	۷	۰	۵	۰	۹	۰	۹	۴
۳	۶	۳	۷	۰	۰	۴	۷	۵	۰	۶	۰	۷	۶	۰	۳
۴	۶,۵	۲	۰	۵	۱۰	۰	۵	۰	۴	۰	۳	۷	۸	۰	۳
۵	۱۱,۹,۸	۱	۷	۰	۵	۰	۷	۰	۲	۰	۸	۹	۲	۰	۸
۶	۱۱,۸,۷	۳	۰	۷	۷	۰	۸	۷	۰	۰	۴	۹	۴	۰	۶
۷	۹	۲	۳	۰	۱۰	۰	۲	۰	۴	۵	۰	۵	۰	۴	۲
۸	۱۰	۳	۷	۰	۰	۱۰	۴	۷	۰	۰	۹	۵	۰	۷	۸
۹	۱۲	۲	۰	۵	۳	۰	۶	۲	۰	۰	۸	۱۰	۰	۳	۴
۱۰	۱۲	۶	۰	۵	۰	۵	۷	۰	۵	۶	۰	۸	۰	۴	۰
۱۱	۱۲	۱	۹	۰	۰	۴	۵	۰	۷	۰	۳	۸	۰	۶	۳
۱۲	-	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰

$(R_1^p, R_2^p, R_1^v, R_2^v) = (۱۴, ۱۲, ۴۱, ۳۵)$: سطح در دسترس منابع

۳-۱- نحوه نمایش جواب ها

در روش پیشنهادی به منظور حل مدل، هر جواب به صورت دو فهرست n عنصری که n برابر با تعداد فعالیت‌هاست، نشان داده می‌شود. فهرست اول شامل روش‌های اجرایی فعالیت‌ها بوده، آن را فهرست روش^{۵۷} می‌نامیم و با μ نشان می‌دهیم. در این فهرست عنصر J ام نشان دهنده روش اجرایی فعالیت J ام است. فهرست دوم یک جایگشت شدنی از فعالیت‌هاست که در آن هر فعالیت پس از تمام پیشنهادهاش ظاهر می‌شود. این فهرست، فهرست فعالیت^{۵۸} نامیده شده، با λ نمایش داده می‌شود. کولیش و هارتمن^{۵۹} (۱۹۹۹) به تشریح انواع مختلف نمایش زمان‌بندی فعالیت‌ها پرداختند. هارتمن و کولیش^{۶۰} (۲۰۰۰) بر اساس نتایج حاصل شده از آزمایش‌های صورت گرفته نشان دادند نمایش زمان‌بندی فعالیت‌ها به شکل فهرست فعالیت بهتر از سایر روش‌ها بوده، به ایجاد نتایج بهتری منجر می‌گردد.

۳-۲- جمعیت اولیه

روش حل پیشنهادی، با ساخت یک جمعیت اولیه از فهرست‌های روش شروع می‌شود. هر فهرست روش μ با انتخاب تصادفی یکی از روش‌های ممکن برای هر فعالیت حاصل می‌شود. سپس شدنی بودن هر فهرست روش از نظر مصرف منابع تجدید ناپذیر بررسی می‌شود. اگر مقدار مورد نیاز منبع تجدید ناپذیر k از حداکثر حد در دسترس آن منبع تجاوز

کند، میزان اضافه در خواست فهرست روش μ برای منبع k از رابطه زیر حاصل می‌شود:

$$E_k^V(\mu) = \max(0, \sum_{j=1}^J r_{jk}^V - R_k^V) \quad (9)$$

در واقع، مقدار مثبت برای $E_k^V(\mu)$ نشان دهنده نشدنی بودن فهرست روش μ است. جمع کل اضافه درخواست فهرست روش μ برای منابع تجدید ناپذیر از رابطه زیر حاصل می‌شود:

$$E^V(\mu) = \sum_{k=1}^{R^V} E_k^V(\mu) \quad (10)$$

بنابراین، یک فهرست روش شدنی است اگر و فقط اگر $E^V(\mu) = 0$. در صورتی که فهرست روش ایجاد شده نشدنی باشد، یک الگوریتم جستجوی محلی به منظور شدنی ساختن فهرست روش بر آن اعمال می‌گردد. در این الگوریتم یک فعالیت تصادفی با بیش از یک روش اجرایی انتخاب و روش فعلی آن با یک روش تصادفی دیگر تعویض می‌گردد. اگر فهرست روش جدید به خوبی قبلی بود، یعنی میزان اضافه درخواست برای منابع تجدید ناپذیر افزایش پیدا نکرد، روش انتخاب شده برای فعالیت تثبیت شده و فهرست جدید مبنای ادامه الگوریتم قرار می‌گیرد و در غیر این صورت، ادامه الگوریتم بر روی فهرست روش قبلی اجرا می‌گردد. این الگوریتم تا J (تعداد فعالیت‌ها) تلاش ناموفق متوالی به منظور کاهش سطح درخواست منابع و یا تا شدنی شدن فهرست روش ادامه پیدا می‌کند.

مد فهرست ها	$\sum_{j=1}^{12} r_{j\mu_1}^V$	$\sum_{j=1}^{12} r_{j\mu_2}^V$	$E_1^V(\mu)$	$E_2^V(\mu)$	$E^V(\mu)$	feasibility
	۴۵	۱۷	۴	۰	۴	۷
	۴۵	۱۸	۴	۰	۴	۷
	۳۵	۲۱	۰	۰	۰	۳

شکل ۱- مراحل فرآیند تبدیل یک فهرست روش نشدنی به یک مد فهرست شدنی

۳-۳- محاسبه تابع هدف

پس از تشکیل جمعیت باید مقدار تابع هدف برای نقاط محاسبه گردد، چرا که مبنای محاسبه بار برای آنها خواهد بود. تابع هدف محاسبه شده برای هر فهرست روش، C_{max} یا همان طول زمان پروژه خواهد بود. مسلماً محاسبه طول پروژه مستقیماً با استفاده از فهرست روش میسر نیست. لذا ابتدا مدت زمان هر فعالیت و مصرف منابع تجدید پذیر برای هر فعالیت بر اساس فهرست روش تعیین گردیده و N برنامه زمان بندی (فهرست فعالیت) به صورت تصادفی بر اساس روش تولید برنامه زمان بندی سری تولید می گردد. هر کدام از این N برنامه زمان بندی توسط یک الگوریتم جستجوی محلی بهبود داده شده و کمترین C_{max} به عنوان تابع هدف فهرست روش گزارش می گردد. همان گونه که توضیح داده شد، الگوریتم جستجوی محلی ارایه شده برای شدنی نمودن فهرست های روش بعد از l تلاش ناموفق متوقف می شود، لذا امکان دارد برخی از فهرست های روش همچنان نشدنی باقی بمانند. چند مدل مختلف می توان با فهرست های روش نشدنی رفتار نمود. می توان آنها را از جمعیت حذف نمود و یا با فهرست روش نشدنی دیگری جایگزین نمود و یا آنها را جریمه کرد. هارتمن^{۱۱} (۲۰۰۱) تابع هدفی برای نقاط تعریف کرد که در آن برای نقاط نشدنی جریمه در

نظر گرفته شده است. این تابع هدف توسط جوزفوسکا و همکاران^{۱۲} (۲۰۰۱) نیز به کار گرفته شد. الکاراز و همکاران^{۱۳} (۲۰۰۳) به این نکته اشاره کردند که در تعریف مذکور دو جواب نشدنی مختلف با اضافه درخواست منبع یکسان و طول پروژه مختلف یک مقدار تابع هدف برابر خواهند داشت. به همین جهت، آنها تابع هدف دیگری پیشنهاد کردند. لووا و همکاران^{۱۴} (۲۰۰۹) به این موضوع اشاره کردند که در تابع هدف پیشنهادی الکاراز و همکاران^{۱۵} (۲۰۰۳) واحدهای زمان مربوط به طول اجرای پروژه و واحدهای منبع مربوط به اضافه درخواست با یکدیگر جمع شده اند و این در حالی است که این دو از یک جنس نیستند. به منظور حل این مشکل آنها تابع هدفی را پیشنهاد دادند که در آن متغیرها نرمالیزه شده اند. این تابع هدف از توابع پیشنهادی قبلی منطقی تر به نظر می رسد، اما اولاً مقدار طول پروژه مستقیماً از آن قابل دریافت نبوده؛ ثانیاً امکان دارد یک جواب در دو تکرار مختلف مقادیر تابع هدف متفاوتی به خود بگیرد. ضمناً هرچند مقادیر زمان و منبع پس از نرمال شدن با هم جمع می شوند، اما همچنان از دو جنس متفاوت هستند.

روش سری استفاده می‌گردد که با روش نمایش برنامه زمان‌بندی به شکل فهرست فعالیت سازگارتر است. تعداد مراحل این الگوریتم برابر با تعداد فعالیت‌های پروژه است چرا که در هر مرحله دقیقاً یک فعالیت انتخاب و زمان‌بندی می‌شود. در هر مرحله از این روش کل فعالیت‌ها به سه دسته تقسیم می‌شوند: فعالیت‌هایی که تا این مرحله زمان‌بندی شده‌اند، فعالیت‌هایی که برای انتخاب و زمان‌بندی واجد شرایط هستند و فعالیت‌هایی که تا این مرحله زمان‌بندی نشده‌اند و امکان زمان‌بندی آنها در این مرحله نیست، چون تمام پیشنیازهای آنها تا قبل از این مرحله زمان‌بندی نشده‌اند. در هر مرحله، یک فعالیت به صورت تصادفی از بین فعالیت‌های واجد شرایط انتخاب و در زودترین زمان شدنی ممکن از نظر پیشنیازی فعالیت‌ها و سطح در دسترس منابع تجدید پذیر زمان‌بندی می‌شود. در نخستین مرحله، تنها فعالیتی که می‌تواند انتخاب شود، فعالیت مجازی شروع و در آخرین مرحله هم فعالیت مجازی پایان خواهد بود. پس از انتخاب و زمان‌بندی کلیه فعالیت‌ها، زمان فعالیت مجازی پایان به عنوان زمان پایان پروژه گزارش می‌گردد. ضمناً خلاصه مراحل زمان‌بندی شامل ترتیب انتخاب فعالیت‌ها برای زمان‌بندی، به صورت فهرست فعالیت $\lambda = \{j_1, j_2, \dots, j_n\}$ ذخیره می‌گردد. این فهرست نشان می‌دهد فعالیت j_g در مرحله g انتخاب و زمان‌بندی شده است.

۳-۳-۲- بهبود برنامه زمان‌بندی

مطابق آنچه توضیح داده شد، به منظور محاسبه C_{max} برای یک فهرست روش، پس از تولید N برنامه زمان‌بندی تصادفی با استفاده از روش تولید برنامه زمان‌بندی سری برای آن فهرست روش، تلاش می‌گردد هر یک از این برنامه‌های زمان‌بندی با استفاده از یک الگوریتم جستجوی محلی بهبود یابد. در پایان، بهترین برنامه زمان‌بندی با کمترین طول پروژه

در این مقاله تابع هدف جدیدی پیشنهاد می‌گردد. که در آن تا حد امکان نواقص توابع هدف قبلی بر طرف گردیده است.

(۱۱)

$$f = \begin{cases} C_{\max}(\lambda, \mu) & \text{if } E^v(\mu) = 0 \\ C_{\max}(\lambda, \mu) + (T - CP) + T(\max\left\{\frac{E_k^v(\mu)}{R_k^v}, k=1,2,\dots,R\right\}) & \text{otherwise} \end{cases}$$

در این تعریف، برای جواب‌های شدنی، طول پروژه به عنوان تابع هدف در نظر گرفته شده است و برای نقاط نشدنی دو عبارت به عنوان جریمه به این مقدار اضافه می‌گردد. مقدار اول، اختلاف حد بالای زمان پروژه و حد پایین آن؛ یعنی طول مسیر بحرانی بر اساس حداقل زمان فعالیت‌هاست و مقدار دوم، با این فرض حاصل می‌شود که اگر نیاز به منابع تجدید ناپذیر بیش از ظرفیت مجاز باشد، اضافه درخواست می‌تواند پس از حد بالای زمان اتمام پروژه تأمین گردد. لذا مقدار زمان لازم برای تأمین مقدار منبع اضافه از ضرب حداکثر نسبت درخواست منابع در T حاصل می‌شود. در واقع، این فرض بدین معنی است که اگر برای مثال، حداکثر نیاز به منابع تجدید ناپذیر، ۵٪ بیش از مقدار مجاز باشد، این مقدار می‌تواند با افزایش حد بالای زمان اتمام پروژه به میزان ۵٪ تأمین گردد. بدیهی است اضافه درخواست سایر منابع که کمتر از این مقدار است، در این زمان قابل تأمین خواهد بود

۳-۳-۱- تولید برنامه زمان‌بندی

دو روش اصلی برای تولید برنامه زمان‌بندی در مقالات وجود دارد. تولید برنامه زمان‌بندی به روش سری^{۶۶} (کلی^{۶۷}، ۱۹۶۳) و موازی^{۶۸} (بدورث و بیلی^{۶۹}، ۱۹۸۲). در روش حل پیشنهادی در این مقاله از

جایگاه قدیم حرکت داده می‌شوند. اگر فعالیت انتخاب شده مجاور آخرین پیشنهاد و نخستین پسینازش باشد، امکان حرکت نداشته و فعالیت دیگری به منظور حرکت انتخاب می‌گردد.

(۲) دو فعالیت مجاور به صورت تصادفی انتخاب شده و در صورتی که فعالیت اول پیشنهاد فعالیت دوم نباشد، جا به جا می‌شوند.

پس از تولید برنامه زمانبندی همسایه، به منظور مقایسه با جواب قبلی باید زمان پایان آن محاسبه گردد. این کار توسط الگوریتم تولید برنامه زمانبندی به روش سری که مراحل آن توضیح داده شد، انجام می‌گیرد. البته، با این تفاوت که در این مرحله انتخاب فعالیت‌ها به روش گفته شده انجام نشده و فعالیت‌ها به ترتیب فهرست فعالیت‌ها انتخاب و زمانبندی می‌شوند. در شکل ۲ مراحل الگوریتم ایجاد برنامه زمانبندی همسایه نشان داده شده است.

به عنوان برنامه زمانبندی متناظر آن فهرست روش گزارش می‌گردد. اساس کار الگوریتم جستجوی محلی به این شکل است که با شروع از یکی از آن N برنامه زمانبندی تولید شده سعی می‌گردد، با ایجاد تغییر کوچکی در برنامه، یک برنامه زمانبندی همسایه ایجاد گردد. سپس زمان پایان این برنامه جدید محاسبه و در صورت بهبود جایگزین برنامه قبلی شده و الگوریتم بر روی این برنامه تکرار می‌گردد و در غیر این صورت، برنامه قبلی مبنای ادامه الگوریتم قرار می‌گیرد. دو روش مختلف به منظور تولید برنامه زمانبندی همسایه به کار گرفته می‌شود.

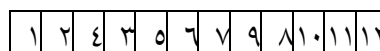
(۱) یک فعالیت به صورت تصادفی انتخاب شده و به محلی تصادفی بین آخرین پیشنهاد و نخستین پسینازش در فهرست منتقل می‌شود. سپس مجموعه فعالیت‌هایی که بین مکان قدیم و مکان جدید آن فعالیت قرار گرفته‌اند، به اندازه یک جایگاه به سمت

(الف)

: مد فهرست



: یک برنامه زمان بندی نمونه



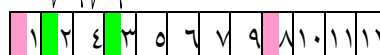
$$C_{max} = 32$$

(ب)

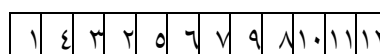
آخرین پیشنهاد

نخستین پسیناز

: اعمال روش اول تولید همسایه بر روی



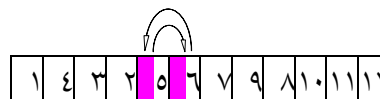
: برنامه زمانبندی همسایه تولید شده به روش



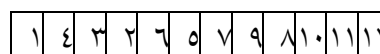
$$C_{max} = 26$$

(ج)

: اعمال روش دوم تولید همسایه بر روی برنامه



: برنامه زمانبندی همسایه تولید شده به روش دوم



$$C_{max} = 25$$

شکل ۲- مراحل بهبود فهرست فعالیت‌ها

	فهرست روش	فهرست فعالیت	C_{max}
جواب اولیه	۱ ۲ ۲ ۳ ۱ ۳ ۱ ۱ ۱ ۱ ۱ ۱	۱ ۴ ۳ ۲ ۶ ۵ ۷ ۹ ۸ ۱۰ ۱۱ ۱۲	۲۵
جواب بهبود یافته	۱ ۲ ۲ ۳ ۱ ۱ ۱ ۱ ۱ ۱ ۱ ۱	۱ ۳ ۴ ۵ ۶ ۲ ۸ ۷ ۱۰ ۹ ۱۱ ۱۲	۲۳

شکل ۳- الگوریتم بهبود فهرست روش

صورتی مورد قبول قرار می‌گیرد که از نظر مقدار درخواست منابع تجدید ناپذیر شدنی باشد. برای این فهرست روش همسایه، بهترین برنامه زمانبندی با حداقل زمان اتمام پروژه به روش گفته شده تعیین می‌گردد. اگر فهرست روش حاصل شده بهتر از فهرست روش اولیه باشد جایگزین آن شده و الگوریتم بر روی این فهرست روش تکرار می‌شود و در غیر این صورت، فهرست روش قبلی مبنای ادامه الگوریتم قرار می‌گیرد.

۳-۵- محاسبه بردار نیروی کل

به منظور نمایش مراحل باقی مانده از الگوریتم، جدول ۲ یک جمعیت نمونه با پنج فهرست روش را ارائه می‌کند. برای هر فهرست روش بهترین برنامه زمانبندی به شیوه توضیح داده شده ایجاد و زمان پایان پروژه برای آن گزارش می‌گردد.

در شکل شماره ۲- الف یک برنامه زمانبندی تصادفی برای فهرست روش به دست آمده در شکل ۱ تولید شده است. در شکل شماره ۲- ب یک برنامه زمانبندی همسایه به روش اول برای آن تولید شده و چون برنامه تولید شده بهتر از قبلی است، جایگزین آن گردیده است. در شکل شماره ۲- ج به کمک الگوریتم دوم برای این برنامه جدید یک برنامه همسایه تولید شده و چون موجب بهبود گردیده، پذیرفته شده است.

۳-۴- جستجوی محلی

این مرحله از الگوریتم پیشنهادی می‌تواند در هر تکرار بر روی یک یا چند یا حتی همه فهرست‌های روش به منظور پیدا کردن جواب‌هایی بهتر از جواب فعلی با جستجو در همسایگی آنها اعمال گردد. با شروع از یک فهرست روش، یک جواب همسایه برای آن پیدا می‌شود. فهرست روش همسایه تنها در

جدول ۲- یک جمعیت نمونه با پنج فهرست روش

i	فهرست روش	$\sum_{j=1}^{12} r_{ij}^v$	$\sum_{j=1}^{12} r_{ij}^v$	فهرست فعالیت	C_m
۱	۱ ۱ ۲ ۱ ۲ ۱ ۲ ۱ ۲ ۱ ۲ ۱	۳۵	۳۴	۱ ۳ ۴ ۲ ۶ ۵ ۱۱ ۷ ۸ ۹ ۱۰ ۱۲	۱۹
۲	۱ ۲ ۲ ۳ ۱ ۱ ۱ ۱ ۱ ۱ ۱ ۱	۳۶	۲۱	۱ ۳ ۴ ۵ ۲ ۶ ۷ ۸ ۱۰ ۹ ۱۱ ۱۲	۲۳
۳	۱ ۲ ۱ ۱ ۱ ۱ ۲ ۱ ۱ ۱ ۲ ۱	۳۵	۲۲	۱ ۳ ۲ ۴ ۶ ۷ ۵ ۸ ۱۱ ۱۰ ۹ ۱۲	۱۶
۴	۱ ۱ ۱ ۱ ۱ ۱ ۳ ۱ ۱ ۱ ۳ ۱	۳۲	۲۴	۱ ۲ ۴ ۵ ۳ ۶ ۷ ۱۱ ۸ ۱۰ ۹ ۱۲	۱۷
۵	۱ ۳ ۱ ۱ ۱ ۱ ۲ ۲ ۱ ۱ ۲ ۱	۳۴	۲۱	۱ ۴ ۲ ۳ ۵ ۶ ۸ ۱۱ ۷ ۱۰ ۹ ۱۲	۲۴

دوم است، به عنوان x^p در محاسبات بعدی لحاظ می‌گردد. البته، نقاط دیگری نیز می‌تواند به عنوان x^p انتخاب گردد، اما یکی از بهترین انتخاب‌ها همان دورترین نقطه از بهترین نقطه است. جدول ۳ محاسبه بار هر نقطه و تعیین x^{best} و x^p در جمعیت فعلی را نشان می‌دهد.

فهرست‌های روش متغیر و زمان‌های اتمام پروژه به عنوان مقادیر تابع هدف در نظر گرفته می‌شوند. فهرست فعالیت‌ها تنها به عنوان واسطه‌ای برای تعیین زمان اتمام پروژه متناظر با هر فهرست روش عمل می‌کند. نقطه سوم بهترین مقدار تابع هدف را داشته، بنابراین در x^{best} ذخیره می‌گردد. بر اساس محاسبه فاصله نقاط از x^{best} دورترین نقطه از x^{best} که نقطه

جدول ۳- محاسبه بار هر نقطه و تعیین x^{best} و x^p در جمعیت فعلی

i	x^i	$f(x^i)$	x^{best}	q^i	$\ x^{best}-x^i\ $	x^p
۱	۱ ۱ ۲ ۱ ۲ ۱ ۲ ۱ ۲ ۱ ۲ ۱	۱۹		۰,۱۵۰۴	۲	
۲	۱ ۲ ۲ ۳ ۱ ۱ ۱ ۱ ۱ ۱ ۱ ۱	۲۳		۰,۰۱۲۰	۲,۶۵	←
۳	۱ ۲ ۱ ۱ ۱ ۱ ۲ ۱ ۱ ۱ ۲ ۱	۱۶	←	۱	۰	
۴	۱ ۱ ۱ ۱ ۱ ۱ ۳ ۱ ۱ ۱ ۳ ۱	۱۷		۰,۵۳۱۸	۱,۷۳	
۵	۱ ۳ ۱ ۱ ۱ ۱ ۲ ۲ ۱ ۱ ۲ ۱	۲۴		۰,۰۰۶۴	۱,۴۱	

هدف بدتری دارد، دفع می‌کند. برای محاسبه نیروهای وارد بر نقطه x^p متغیرهای تصادفی λ و u به ترتیب برابر با $۰/۹۳$ و $۰/۸۳$ در نظر گرفته شده است. جدول ۴ برآیند نیروی وارد بر هر نقطه را نشان می‌دهد. ضمناً همان گونه که گفته شد x^{best} حرکت داده نشده و بدون تغییر به تکرار بعد منتقل می‌شود. بنابراین، می‌توان از محاسبه نیروی وارد بر این نقطه صرف نظر کرد.

نیروی کل وارد بر هر نقطه، F^i ، از جمع برداری نیروهای وارد بر آن از طرف سایر نقاط حاصل می‌گردد. شایان ذکر است بارها علامت نداشته و برای نیروی بین هر دو نقطه بعد از مقایسه مقدار تابع هدف آنها تعیین می‌گردد؛ یعنی همان گونه که قبلاً توضیح داده شد، بین هر دو نقطه، نقطه‌ای که مقدار تابع هدف بهتری دارد، جذب می‌کند، نقطه‌ای که تابع

جدول ۴- نیروی کل وارد بر هر نقطه

i	F^i											
	F_1^i	F_2^i	F_3^i	F_4^i	F_5^i	F_6^i	F_7^i	F_8^i	F_9^i	F_{10}^i	F_{11}^i	F_{12}^i
۱	۰	۰,۰۳۷۱	-۰,۰۵۳۳	-۰,۰۰۰۴	-۰,۰۵۳۳	۰	۰,۰۱۶۲	-۰,۰۰۰۱	-۰,۰۵۳۳	۰	۰,۰۱۶۲	۰
۲	۰	-۰,۰۰۰۶	-۰,۰۰۲۰	-۰,۰۰۴۴	۰,۰۰۰۲	۰	۰,۰۰۲۶	-۰,۰۰۰۰	۰,۰۰۰۲	۰	۰,۰۰۲۶	۰
۳	۰	۰,۲۱۱۶	-۰,۰۳۹۳	-۰,۰۰۳۴	-۰,۰۳۷۶	۰	-۰,۱۷۵۵	-۰,۰۰۳۲	-۰,۰۳۷۶	۰	-۰,۱۷۵۵	۰
۴	۰	۰,۱۷۵۸	-۰,۰۱۶۴	-۰,۰۰۰۹	-۰,۰۱۶۰	۰	-۰,۱۵۹۹	-۰,۰۰۰۵	-۰,۰۱۶۰	۰	-۰,۱۵۹۹	۰
۵	۰	-۰,۰۰۴۴	۰,۰۰۰۱	۰,۰۰۰۰	۰,۰۰۰۱	۰	۰,۰۰۰۵	-۰,۰۰۳۸	۰,۰۰۰۱	۰	۰,۰۰۰۵	۰

۳-۶- حرکت نقاط در راستای بردار نیرو

پس از تعیین نیروی کل وارد بر هر نقطه، آن نقطه در راستای نیروی محاسبه شده به اندازه‌ای که به صورت تصادفی تعیین می‌شود، حرکت می‌کند. جدول ۵ بردار حرکت هر یک از نقاط را نشان می‌دهد. به منظور سهولت در محاسبات مقدار متغیر تصادفی α برای همه نقاط برابر با ۰/۹ در نظر گرفته شده است. دقت کنید که برای محاسبه بردار RNG ، حد پایین و

بالا برای بُعد اول و آخر که مربوط به نخستین و آخرین فعالیت مجازی هستند، ۱ و برای سایر ابعاد به ترتیب برابر ۱ و ۳ در نظر گرفته شده است. پس از محاسبه بردارهای حرکت، هر نقطه با استفاده از بردار متناظرش به نقطه‌ای جدید منتقل می‌شود. مختصات جدید هر نقطه بعد از حرکت با بردار حرکت متناظرش در جدول ۶ آورده شده است.

جدول ۵- بردار حرکت هر یک از نقاط

i	m^i											
	m_1^i	m_2^i	m_3^i	m_4^i	m_5^i	m_6^i	m_7^i	m_8^i	m_9^i	m_{10}^i	m_{11}^i	m_{12}^i
۱	۰	۰,۶۵۴۶	-۰,۴۷۱۰	۰	-۰,۴۶۹۲	۰	۰,۱۴۲۶	۰	-۰,۴۶۹۲	۰	۰,۱۴۲۶	۰
۲	۰	-۰,۰۹۰۸	-۰,۲۹۵۳	-۱,۲۹۰۹	۰,۰۵۴۸	۰	۰,۷۷۰۰	۰	۰,۰۵۴۸	۰	۰,۷۷۰۰	۰
۳	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰
۴	۰	۱,۰۹۹۸	۰	۰	۰	۰	-۱,۰۰۰۰	۰	۰	۰	-۱,۰۰۰۰	۰
۵	۰	-۱,۳۵۲۹	۰,۰۳۹۴	۰,۰۰۵۲	۰,۰۳۶۸	۰	۰,۰۷۳۱	-۰,۵۸۳۷	۰,۰۳۶۸	۰	۰,۰۷۳۱	۰

جدول ۶- مختصات جدید نقاط بعد از حرکت

i	x^i											
	x_1^i	x_2^i	x_3^i	x_4^i	x_5^i	x_6^i	x_7^i	x_8^i	x_9^i	x_{10}^i	x_{11}^i	x_{12}^i
۱	۱	۱,۶۵۴۶	۱,۵۲۹۰	۱	۱,۵۳۰۸	۱	۲,۱۴۲۶	۱	۱,۵۳۰۸	۱	۲,۱۴۲۶	۱
۲	۱	۱,۹۰۹۲	۱,۷۰۴۷	۱,۷۰۹۱	۱,۰۵۴۸	۱	۱,۷۷۰۰	۱	۱,۰۵۴۸	۱	۱,۷۷۰۰	۱
۳	۱	۲	۱	۱	۱	۱	۲	۱	۱	۱	۲	۱
۴	۱	۲,۰۹۹۸	۱	۱	۱	۱	۲,۰۰۰۰	۱	۱	۱	۲,۰۰۰۰	۱
۵	۱	۱,۶۴۷۱	۱,۰۳۹۴	۱,۰۰۵۲	۱,۰۳۶۸	۱	۲,۰۷۳۱	۱,۴۱۶۳	۱,۰۳۶۸	۱	۲,۰۷۳۱	۱

معرفی شده تبدیل به یک نقطه شدنی می‌گردد. سپس بهترین برنامه زمانبندی ممکن به آن نسبت داده شده و C_{max} آن نیز به عنوان تابع هدف فهرست روش مذکور گزارش می‌گردد. در صورتی که بعد اعمال الگوریتم فهرست روش همچنان نشدنی باقی بماند، جریمه ای مطابق با رابطه پیشنهاد شده محاسبه و به مقدار تابع هدف آن اضافه می‌گردد. جدول ۷ نقاط نهایی حاصل شده از الگوریتم را نشان می‌دهد.

پایین (۱) و بالا (۳) برای هر بعد جواب به سه زیر بازه مساوی تقسیم، مقادیر کوچکتر از $1/6665$ تبدیل به ۱، مقادیر بین $1/6665$ و $2/3335$ تبدیل به ۲ و مقادیر بزرگتر از $2/3335$ تبدیل به ۳ و بدین ترتیب مقادیر غیر صحیح به مقادیر صحیح ۱، ۲ و ۳ تبدیل می‌شود.

نهایتاً فهرست روش به دست آمده از نظر مقدار درخواست منابع تجدید ناپذیر بررسی شده، در صورت اضافه درخواست با استفاده از الگوریتم

جدول ۷- نقاط نهایی حاصل شده از الگوریتم

i	مد فهرست												$\frac{d_2}{d_1}$		فهرست فعالیت												C_{max}
	۱	۱	۱	۱	۱	۱	۲	۱	۱	۱	۲	۱	۳۷	۲۲	۱	۴	۵	۳	۲	۶	۷	۸	۱۰	۹	۱۱	۱۲	
۱	۱	۱	۱	۱	۱	۱	۲	۱	۱	۱	۲	۱	۳۷	۲۲	۱	۴	۵	۳	۲	۶	۷	۸	۱۰	۹	۱۱	۱۲	۱۵
۲	۱	۲	۲	۲	۱	۱	۲	۱	۱	۱	۲	۱	۳۱	۲۱	۱	۴	۳	۲	۶	۵	۸	۷	۱۰	۹	۱۱	۱۲	۱۹
۳	۱	۲	۱	۱	۱	۱	۲	۱	۱	۱	۲	۱	۳۵	۲۲	۱	۳	۲	۴	۶	۷	۵	۸	۱۱	۱۰	۹	۱۲	۱۶
۴	۱	۲	۱	۱	۱	۱	۲	۱	۱	۱	۲	۱	۳۵	۲۲	۱	۴	۲	۳	۶	۵	۷	۸	۱۱	۹	۱۰	۱۲	۱۶
۵	۱	۱	۱	۱	۱	۱	۲	۱	۱	۱	۲	۱	۳۷	۲۲	۱	۴	۲	۵	۳	۶	۱۱	۸	۷	۱۰	۹	۱۰	۱۵

۴- بحث

۱۰، ۱۲، ۱۴، ۱۶، ۱۸، ۲۰ و ۳۰ فعالیت غیر مجازی است که در آنها برای هر فعالیت ۳ روش اجرایی در نظر گرفته شده است. انجام هر فعالیت به هر یک از روش‌های اجرایی به مقادیری از ۲ منبع تجدید پذیر و ۲ منبع تجدید ناپذیر نیاز دارد. زمان هر فعالیت در هر روش اجرایی عددی بین ۱ تا ۱۰ است. جواب بهینه مسایل با اندازه ۱۰ تا ۲۰ در بیشتر موارد با استفاده از روش‌های دقیق تعیین گردیده و در کتابخانه معرفی شده موجود است، ولی برای مسایل با ۳۰ فعالیت تنها بهترین جواب‌های به دست آمده با روش‌های ابتکاری در دسترس است، لذا این دسته مسایل استفاده نشده است.

در این قسمت، عملکرد الگوریتم پیشنهادی در ارتباط با حل مسأله MRCPSP با بهترین روش‌های فرا ابتکاری ارایه شده برای حل این مسأله تاکنون با استفاده از مثال‌های موجود در کتابخانه مسایل برنامه‌ریزی پروژه (PSPLIB) (کولیش و اسپرچر^{۷۰}، ۱۹۹۶) مقایسه می‌گردد. مجموعه مثال‌های PSPLIB توسط نرم افزار ProGen (کولیش و همکاران^{۷۱}، ۱۹۹۵) تولید گردیده و در آدرس <http://129.187.106.231/psplib/> در دسترس است. تعداد مسائل هر مجموعه مسأله در PSPLIB در جدول ۸ آمده است. این مجموعه شامل مسایلی با

جدول ۸- تعداد مسائل هر مجموعه مسأله در PSPLIB.

مجموعه مسائل	J10	J12	J14	J16	J18	J20	J30
تعداد مسائل	۵۳۶	۵۴۷	۵۵۱	۵۵۰	۵۵۲	۵۵۴	۵۵۲

حل به ترتیب در جداول ۹ و ۱۰ آورده شده است. در این جداول میانگین انحراف از جواب بهینه به درصد (ADO) و درصد جواب‌های بهینه پیدا شده (POF) در هر مجموعه مثال برای هر روش گزارش گردیده است. در جدول ۹ عملکرد الگوریتم پیشنهادی با الگوریتم‌های ژنتیکی ون پتقم و ونهوک^{۷۲} (۲۰۱۰)، لووا و همکاران^{۷۳} (۲۰۰۹) و الکازار و همکاران^{۷۴} (۲۰۰۳)، جستجوی پراکنده رنجبرو همکاران^{۷۵} (۲۰۰۹) و شبیه‌سازی تبرید جوزفوسکا و همکاران^{۷۶} (۲۰۰۱) بر اساس ۵۰۰۰ جواب تولید شده مقایسه گردیده است؛ یعنی برای هر دسته مسأله عملیات پس از تولید ۵۰۰۰ جواب متوقف گردیده و بهترین جواب به دست آمده تا آن لحظه به عنوان جواب مسأله گزارش می‌گردد. همان گونه که نتایج نشان می‌دهد، در مسائل با ۱۰، ۱۲ و ۱۶ فعالیت، الگوریتم پیشنهادی با اختلافی ناچیز در

با مرور مقالات مرتبط، به این نکته می‌رسیم که الگوریتم‌های ارایه شده برای حل MRCPSP بر اساس دو شرط توقف اصلی تعداد جواب‌های تولید شده و زمان حل با یکدیگر مقایسه می‌گردند. به منظور مقایسه الگوریتم پیشنهاد شده با سایر روش‌ها، ابتدا تعداد جواب‌های تولید شده مورد نظر قرار گرفت؛ لذا برنامه نویسی با نرم افزار MATLAB انجام شد. پس از آن با مقایسه زمان حل برنامه نوشته شده با زمان‌های حل ارایه شده در برخی مقالات برای حل مسأله با نرم افزار ++C به این نتیجه رسیدیم که نرم افزار MATLAB با معیار زمان، قابلیت رقابتی خود را از دست می‌دهد، لذا به منظور مقایسه زمانی، کد برنامه به زبان ++C برگردانده شد. مقایسه عملکرد الگوریتم پیشنهادی با سایر الگوریتم‌ها، برای مسایل PSPLIB با جواب‌های بهینه معلوم، بر مبنای تعداد جواب‌های تولید شده و زمان

برای هر فعالیت راه‌اندازی گردیده، ولی بولیمین و لکوک^{۸۲} (۲۰۰۳) الگوریتم شبیه‌سازی تبرید خود را بر روی یک رایانه با پردازنده ۱۰۰ مگا هرتز با ۵ ثانیه به ازای هر فعالیت راه‌اندازی نمودند. همان گونه که نتایج نشان می‌دهد، در مسائل با ۱۰ فعالیت، الگوریتم پیشنهادی از نظر میانگین انحراف نسبت به جواب بهینه در رتبه دوم و از نظر درصد جواب‌های بهینه به دست آمده رتبه سوم را کسب می‌کند و برای مسائل با ۱۲ فعالیت، الگوریتم پیشنهادی از نظر میانگین انحراف از جواب بهینه رتبه اول و از نظر تعداد جواب‌های بهینه حاصل شده رتبه دوم را به دست می‌آورد. الگوریتم پیشنهادی برای دسته مسائل دیگر بهترین عملکرد را داراست

رتبه دوم قرار گرفته، ولی برای مسائل با ۱۴، ۱۸ و ۲۰ فعالیت، بهترین عملکرد را داراست.

در جدول ۱۰، عملکرد روش حل پیشنهادی با الگوریتم‌های پیشنهادی جاربوی و همکاران^{۷۷} (۲۰۰۸)، داماک و همکاران^{۷۸} (۲۰۰۹) و بولیمین و لکوک^{۷۹} (۲۰۰۳) بر اساس زمان حل ۰٫۱۵ ثانیه به ازای هر فعالیت مقایسه گردیده است، یعنی برای هر دسته مسأله با اندازه n ، عملیات پس از $n \times 0.15$ ثانیه متوقف گردیده و بهترین جواب به دست آمده تا آن لحظه به عنوان جواب مسأله گزارش می‌گردد. نکته قابل ذکر اینکه دو الگوریتم اول (جاربوی و همکاران^{۸۰}، ۲۰۰۸ و داماک و همکاران^{۸۱}، ۲۰۰۹) بر روی رایانه‌هایی با مشخصاتی مشابه آنچه در این مقاله استفاده شده: (Pentium 4, 3.2 GHz) processor, 1GB RAM و با قید زمانی ۰٫۱۵ ثانیه

جدول ۹- مقایسه نتایج الگوریتم پیشنهادی با سایر روش‌ها بر مبنای ۵۰۰۰ جواب تولید شده

	J10	J12	J14	J16	J18	J20
	ADO POF	ADO POF	ADO POF	ADO POF	ADO POF	ADO POF
این مقاله	۰٫۰۵ ۹۸٫۸۶	۰٫۱۰ ۹۷٫۹۴	۰٫۱۹ ۹۵٫۰۶	۰٫۳۵ ۹۱٫۷۲	۰٫۳۹ ۸۹٫۳۳	۰٫۴۹ ۸۷٫۲۱
نویسنده مقاله	۰٫۰۱ ۹۹٫۶۳	۰٫۰۹ ۹۸٫۱۷	۰٫۲۲ ۹۴٫۵۶	۰٫۳۲ ۹۲٫۰۰	۰٫۴۲ ۸۸٫۹۵	۰٫۵۷ ۸۵٫۷۴
لووا و همکاران (۲۰۰۹)	۰٫۰۶ ۹۸٫۵۱	۰٫۱۷ ۹۶٫۵۳	۰٫۳۲ ۹۲٫۹۲	۰٫۴۴ ۹۰٫۰۰	۰٫۶۳ ۸۴٫۹۶	۰٫۸۷ ۸۰٫۳۲
رنجبرو همکاران (۲۰۰۹)	۰٫۱۸ -	۰٫۶۵ -	۰٫۸۹ -	۰٫۹۵ -	۱٫۲۱ -	۱٫۶۴ -
الکاراز و همکاران (۲۰۰۳)	۰٫۲۴ -	۰٫۷۳ -	۱٫۰۰ -	۱٫۱۲ -	۱٫۴۳ -	۱٫۹۱ -
جوزفوسکا و همکاران (۲۰۰۱)	۱٫۱۶ ۸۵٫۶۰	۱٫۷۳ ۸۰٫۳۰	۲٫۶۰ ۶۶٫۴	۴٫۰۷ ۵۴٫۷۰	۵٫۵۲ ۴۳٫۵۰	۶٫۷۴ ۳۵٫۷۰

جدول ۱۰- مقایسه نتایج الگوریتم پیشنهادی با سایر روش‌ها بر مبنای ۰٫۱۵ ثانیه به ازای هر فعالیت

نویسنده مقاله	J10	J12	J14	J16	J18	J20
	ADO POF	ADO POF	ADO POF	ADO POF	ADO POF	ADO POF
این مقاله	۰٫۰۴ ۹۹٫۱۲	۰٫۰۶ ۹۹٫۱۶	۰٫۱۷ ۹۸٫۰۷	۰٫۲۳ ۹۷٫۲۵	۰٫۳۲ ۹۲٫۰۲	۰٫۴۲ ۹۰٫۵۲
داماک و همکاران (۲۰۰۹)	۰٫۰۹ ۹۹٫۳۰	۰٫۱۱ ۹۹٫۳۰	۰٫۳۴ ۹۷٫۶۰	۰٫۴۲ ۹۶٫۳۸	۰٫۵۹ ۹۴٫۴۳	۰٫۷۰ ۹۱٫۷۵
جاربوی و همکاران (۲۰۰۸)	۰٫۰۳ ۹۹٫۲۵	۰٫۰۹ ۹۸٫۴۷	۰٫۳۶ ۹۱٫۱۱	۰٫۴۴ ۸۵٫۹۱	۰٫۸۹ ۷۹٫۸۹	۱٫۱۰ ۷۴٫۱۹
بولیمین و لکوک (۲۰۰۳)	۰٫۲۱ ۹۶٫۳۰	۰٫۱۹ ۹۱٫۲۰	۰٫۹۲ ۸۲٫۶۰	۱٫۴۳ ۷۲٫۸۰	۱٫۸۵ ۶۹٫۴۰	۲٫۱۰ ۶۶٫۹۰

۵- نتیجه‌گیری

تاکنون روش‌های دقیق مختلفی به منظور حل مسأله زمان‌بندی پروژه با منابع محدود و فعالیت‌های چند حالت (یعنی امکان انتخاب روش‌های اجرایی مختلف برای فعالیت‌ها) ارایه شده است که بیشتر آنها از توسعه روش‌های ارایه شده برای حالت تک روش اجرایی حاصل شده است. به هر حال، هیچ کدام از این روش‌ها نمی‌تواند برای حل مسایل بزرگ استفاده شود، چرا که قادر به یافتن جواب بهینه در یک مدت زمان منطقی نیستند. در این مقاله، یک روش تلفیقی برای حل مدل ارایه شده و به همین منظور، مسأله به دو زیر مسأله تقسیم شده است: تعیین روش اجرایی هر فعالیت و سپس یافتن بهترین زمان‌بندی فعالیت‌ها به منظور حداقل نمودن زمان پروژه. با استفاده از روش الکترومغناطیس به هر فعالیت یکی از روش‌های ممکن تخصیص پیدا کرده و زمان اجرای هر فعالیت و میزان مورد نیاز از هر منبع با توجه به روش انتخاب شده برای اجرای آن تعیین می‌گردد. پس از آن، به صورت تصادفی چند برنامه زمان‌بندی تولید شده و یک الگوریتم جستجوی محلی هر یک از برنامه‌ها را بهبود می‌بخشند. سپس بهترین برنامه زمان‌بندی انتخاب و به عنوان زمان تکمیل متناظر با فهرست روش مربوطه گزارش می‌گردد. در این روش حل تابع هدف جدیدی برای نقاط پیشنهاد گردید که در آن تا حد امکان نواقص توابع هدف ارایه شده تا کنون بر طرف گردیده است. پس از آن، نتایج عملکرد الگوریتم پیشنهادی با سایر الگوریتم‌های فرا ابتکاری ارایه شده تاکنون، با استفاده از مثال‌های موجود در کتابخانه مسایل مرتبط بر اساس ۵۰۰۰ جواب تولید شده و زمان حل ۰/۱۵ ثانیه به ازای هر فعالیت مقایسه شد. همان گونه که نتایج نشان می‌دهد، عملکرد الگوریتم در حل مدل عالی است.

منابع:

- Alcaraz, J., Maroto, C., & Ruiz, R. (2003). "Solving the multi-mode resource-constrained project scheduling problem with genetic algorithms". *Journal of the Operational Research Society*, 54, 614-626.
- Bedworth, D., & Bailey, J. (1982). *Integrated production control systems-management, analysis, design*. Wiley, New York.
- Birbil, Ş. İ., & Fang, S. -C. (2003). "An electromagnetism-like mechanism for global optimization", *Journal of Global Optimization*, 25, 263-282.
- Birbil, Ş. İ., Fang, S. -C., & Sheu, R. -L. (2005). "On the convergence of a population based global optimization algorithm", *Journal of Global Optimization*, 30, 301-318.
- Boctor, F. F. (1993). "Heuristics for scheduling projects with resource restrictions and several resource-duration modes", *International Journal of Production Research*, 31, 2547-2558.
- Boctor, F. F. (1996a). "A new and efficient heuristic for scheduling projects with resource restrictions and multiple execution modes", *European Journal of Operational Research*, 90, 349-361.
- Boctor, F. F. (1996b). "Resource-constrained project scheduling by simulated annealing", *International Journal of Production Research*, 34, 2335-2351.
- Bouleimen, K., & Lecocq, H. (2003). "A new efficient simulated annealing algorithm for the resource constrained project scheduling problem and its multiple mode version", *European Journal of Operational Research*, 149, 268-281.
- Brucker, P., Drexl, A., Möhring, R., Neumann, K., & Pesch, E. (1999). "Resource-constrained project scheduling: Notation, classification, models and methods", *European Journal of Operational Research*. 112, 3-41.
- Damak, N., Jarboui, B., Siarry, P., & Loukil, T. (2009). "Differential evolution for solving multi-mode resource-constrained project scheduling problems", *Computers & Operations Research*, 36, 2653-2659.
- Debels, D., Reyck, B. D., Leus, R., & Vanhoucke, M. (2006). "A hybrid scatter search/electromagnetism meta-heuristic for

- In: Muth, J.F., Thompson, G.L. (Eds.), *Industrial Scheduling*", Prentice-Hall, New Jersey, 347-365.
- Kolisch, R., & Drexl, A. (1997). "Local search for nonpreemptive multi-mode resource-constrained project scheduling", *IIE Transactions*, 29, 987-999.
- Kolisch, R., & Hartmann, S. (1999). "Heuristic algorithms for solving the resource-constrained project scheduling problem: classification and computational analysis. In: Węglarz, J. (Ed.), *Project scheduling: Recent models, algorithms and applications*", Kluwer Academic Publishers, 147-178.
- Kolisch, R., & Hartmann, S. (2006). "Experimental investigation of heuristics for resource-constrained project scheduling: An update", *European Journal of Operational Research*, 174, 23-37.
- Kolisch, R., & Sprecher, A. (1996). "PSPLIB – a project scheduling problem library", *European Journal of Operational Research*, 96, 205-216.
- Kolisch, R., Sprecher, A., & Drexl, A. (1995). "Characterization and generation of a general class of resource-constrained project scheduling problems", *Management Science*, 41, 1693-1703.
- Lova, A., Tormos, P., & Barber, F. (2006). "Multi-mode resource constrained project scheduling: Scheduling schemes, priority rules and mode selection rules", *Inteligencia Artificial*, 30, 69-86
- Lova, A., Tormos, P., Cervantes, M., & Barber, F. (2009). "An efficient hybrid genetic algorithm for scheduling projects with resource constraints and multiple execution modes", *Int. J. of Production Economics*, 117, 302-316.
- Naji-Azimi, Z., Toth, P., & Gall, L. (2010). "An electromagnetism metaheuristic for the unicost set covering problem", *European Journal of Operational Research*, 205(2), 290-300.
- Özdamar, L. (1999). "A genetic algorithm approach to a general category project scheduling problem", *IEEE Transactions on Systems, Man and Cybernetics*, Part C 29, 44-59.
- Özdamar, L., & Ulusoy, G. (1994). "A local constraint based analysis approach to project scheduling", *European Journal of Operational Research*, 169(2), 638-653.
- Elmaghraby, S. E. (1977). *Activity networks: Project planning and control by network models*. Wiley, New York
- Gol-Alikhani, M., Javadian, N., & Tavakkoli-Moghaddam, R. (2009). "A novel hybrid approach combining electromagnetism-like method with Solis and Wets local search for continuous optimization problems", *J. of Global Optimization*, 44 (2), 227-234.
- Hartmann, S. (2001). "Project scheduling with multiple modes: a genetic algorithm", *Annals of Operations Research*, 102, 111-135.
- Hartmann, S., & Drexl, A. (1998). "Project scheduling with multiple modes: A comparison of exact algorithms", *Networks*, 32, 283-297.
- Hartmann S., & Kolisch R. (2000). "Experimental evaluation of state-of-the-art heuristics for the resource-constrained project scheduling problem", *European Journal of Operational Research*, 127, 394-407.
- Jarboui, B., Damak, N., Siarry, P., & Rebai, A. (2008). "A combinatorial particle swarm optimization for solving multi-mode resource-constrained project scheduling problems", *Applied Mathematics and Computation*, 195, 299-308.
- Javadian, N., Golalikhani, M., & Tavakkoli-Moghaddam, R. (2008). "A discrete binary version of electromagnetism-like heuristic for solving traveling salesman problem. *Advanced Intelligent Computing Theories and Applications*", Springer-Verlag, 123-130.
- Javadian, N., Golalikhani, M., & Tavakkoli-Moghaddam, R. (2009). "Solving a single machine scheduling problem by a discrete version of the electromagnetism-like method", *Journal of Circuits Systems and Computers*, 18(8), 1597-1608.
- Józefowska, J., Mika, M., Rozycki, R., Waligora, G., & Węglarz, J. (2001). "Simulated annealing for multi-mode resource-constrained project scheduling", *Annals of Operations Research*, 102, 137-155.
- Kelley J. E. Jr. (1963). "The critical path method: Resource planning and scheduling.

- scheduling by a simple, general and powerful sequencing algorithm", *European Journal of Operational Research*, 107, 431-450.
- Sprecher, A., Hartmann, S., & Drexl, A. (1997). "An exact algorithm for project scheduling with multiple modes", *OR Spektrum*, 19, 195-203.
- Talbot, F. B. (1982). "Resource-Constrained Project Scheduling with Time-Resource Tradeoffs: The Nonpreemptive Case", *Management Science*, 38, 1498-1509.
- Van Peteghem, V., & Vanhoucke, M. (2010). "A genetic algorithm for the preemptive and non-preemptive multi-mode resource-constrained project scheduling problem", *European Journal of Operational Research*, 201, 409-418.
- Zhang, H., Tam, C. M., & Li, H. (2006). "Multimode project scheduling based on particle swarm optimization", *Computer-Aided Civil and Infrastructure Engineering*, 21, 93-103.
- Zhu, G., Bard, J., & Tu, G. (2006). "A branch-and-cut procedure for the multimode resource-constrained project scheduling problem", *Journal on Computing*, 18 (3), 377-390.
- project scheduling under general resource constraints",. *European Journal of Operational Research*, 79, 287-298.
- Patterson, J. H., Sowinski, R., Talbot, F. B., & Weglarz, J. (1989). "An algorithm for a general class of precedence and resource constrained scheduling problems", In: Sowinski, R., Weglarz, J., (Eds.), *Advances in Project Scheduling*, Elsevier, Amsterdam, 3-28.
- Ranjbar, M., De Reyck, B., & Kianfar, F. (2009). "A hybrid scatter search for the discrete time/resource trade-off problem in project scheduling", *European Journal of Operational Research*, 193, 35-48.
- Slowiński, R. (1980). "Two approaches to problems of resource allocation among project activities – a comparative study", *Journal of the Operational Research Society*, 31, 711-723.
- Slowiński, R., Soniewicki, B., & Węglarz, J. (1994). "DSS for multiobjective project scheduling", *European Journal of Operational Research*, 79, 220-229.
- Solis, F. J., & Wets, J.-B. (1981). "Minimization by random search techniques", *Mathematical & Operations Research*, 6, 19-30.
- Sprecher, A., & Drexl, A. (1998). "Solving multi-mode resource-constrained project

پی‌نوشت:

- 1 - Electromagnetism-like Algorithm
- 2 - Resource Constrained Project Scheduling Problem
- 3 - Multi mode Resource Constrained Project Scheduling Problem
- 4 - Elmaghraby
- 5 - Slowiński
- 6 - Talbot
- 7 - Patterson et al.
- 8 - Sprecher et al.
- 9 - Hartmann and Drexl
- 10 - Sprecher and Drexl
- 11 - Zhu et al.
- 12 - Talbot
- 13 - Sprecher and Drexl
- 14 - Boctor
- 15 - Özdamar and Ulusoy
- 16 - Boctor
- 17 - Critical Path Method
- 18 - Kolisch and Drexl
- 19 - Lova et al.
- 20 - Slowiński et al.
- 21 - Simulated Annealing Algorithm
- 22 - Boctor
- 23 - Bouleimen and Lecocq

- 24 - Józefowska et al.
- 25 - Genetic Algorithm
- 26 - Özdamar
- 27 - Hartmann
- 28 - Alcaraz et al.
- 29 - Lova et al.
- 30 - Van Peteghem and Vanhoucke
- 31 - Preemption
- 32 - Zhang et al.
- 33 - Particle Swarm
- 34 - Jarboui et al.
- 35 - Combinatorial
- 36 - Zhang et al.
- 37 - Ranjbar et al.
- 38 - Hybrid Scatter Search
- 39 - Discrete Time/Resource Trade-off Problem
- 40 - DTRTP with Multiple resource types
- 41 - Alcaraz et al.
- 42 - Crossover
- 43 - Mutation
- 44 - Damak et al.
- 45 - Differential Evolution
- 46 - Birbil and Fang
- 47 - Population-based
- 48 - Birbil and Fang
- 49 - Premature convergence
- 50 - Birbil et al.
- 51 - Gol-Alikhani et al.
- 52 - Birbil et al.
- 53 - Solis and Wets
- 54 - Naji-Azimi et al.
- 55 - Javadian et al.
- 56 - Debels et al.
- 57 - Mod List
- 58 - Activity list (AL)
- 59 - Kolisch and Hartmann
- 60 - Hartmann and Kolisch
- 61 - Hartmann
- 62 - Józefowska et al.
- 63 - Alcaraz et al.
- 64 - Lova et al.
- 65 - Alcaraz et al.
- 66 - Serial SGS
- 67 - Kelley et al.
- 68 - Parallel SGS
- 69 - Bedworth and Bailey
- 70 - Kolisch and Sprecher
- 71 - Kolisch et al.
- 72 - Van Peteghem and Vanhoucke
- 73 - Lova et al.
- 74 - Alcaraz et al.
- 75 - Ranjbar et al.
- 76 - Józefowska et al.
- 77 - Jarboui et al.
- 78 - Damak et al.
- 79 - Bouleimen and Lecocq
- 80 - Jarboui et al.
- 81 - Damak et al.
- 82 - Bouleimen and Lecocq