

توسعه الگوریتم جستجوی ممنوع (Tabu Search) برای حل مساله زمانبندی پروژه های دارای محدودیت منابع با در نظر گرفتن حالت پیش نیازی عمومی (GRCPS)

مسعود حداد

دانشجوی دکترای مدیریت دانشگاه آزاد اسلامی واحد علوم تحقیقات اصفهان (msdhaddad@yahoo.com)

حمیدرضا حداد

دانشجوی کارشناسی مهندسی صنایع دانشگاه علم و صنعت ایران (hamidrhaddad@gmail.com)

چکیده:

زمانبندی پروژه با در نظر گرفتن محدودیت منابع، تحت روابط پیش نیازی عمومی (GRCPS) دارای ادبیات موضوع بسیار غنی است. این قسمت از مهندسی صنایع به دو دلیل مورد توجه قرار گرفته است: اول این که این مسائل با توجه به شرایط متفاوت کاربردی و صنعتی از نظر تابع هدف، خصوصیت فعالیت ها، منابع و نوع رابطه پیش نیازی بسیار متنوعند و دوم این که با توجه به NP-HARD بودنشان محققین همواره به دنبال ارایه راه حل های کاراتری برای حل آن ها بوده اند. در این مقاله، یک روش فرا ابتکاری بر اساس الگوریتم جستجوی ممنوع (Tabu Search) جهت تولید جواب های مناسب با در نظر گرفتن محدودیت منابع و حالت پیش نیازی و همچنین فعالیت هایی با زمان های اجرای قطعی و احتمالی توسعه داده شده است. این الگوریتم از لیست های ممنوعه مختلف، حافظه تصادفی کوتاه مدت و جواب های اولیه مختلف به عنوان وسیله ای برای ایجاد تنوع در فضای جستجو استفاده می نماید. در طی مثال های مختلف مشخص می شود که روش ارائه شده به عنوان روشی کارا در ایجاد جواب های مناسب برای زمان های قطعی و احتمالی کاربرد دارد.

واژگان کلیدی: زمانبندی پروژه، محدودیت منابع تحت روابط پیش نیازی، الگوریتم جستجوی ممنوع

با توجه به این که امروزه زمانبندی یکی از ارکان روند انجام پروژه های مهندسی است و تقریباً در تمامی پروژه های مهندسی می توان اثری از آن دید، اما با اندکی تامل در سیر تحول این روند از گذشته تا کنون می توان دریافت که وجود برنامه زمانبندی همواره به عنوان یک اصل در پروژه های مهندسی مطرح نبوده است، بلکه با پیشرفت علم و کسب تجربه در زمینه چگونگی انجام فعالیت های پروژه، مجریان پروژه کم کم با این واقعیت آشنا شدند که برای افزایش کیفیت انجام پروژه ها، زمانبندی فعالیت ها به صورتی که قبل از شروع پروژه زمان انجام تمامی آن مشخص باشد، امر لازم و ضروری است. با توجه به سیر تحول طی شده برای آن که برنامه زمانبندی به عنوان یک اصل در روند انجام پروژه های مهندسی درآید و با اهتمام به این نکته که امروزه با رشد روزافزون علوم مهندسی، دیگر موضوع امکان پذیر بودن پروژه ها جای خود را به بهینه بودن آن ها داده است، بدیهی است در آینده ای نه چندان دور موضوع بهینه سازی برنامه زمانبندی نیز کم کم به عنوان یک مرحله حیاتی در روند انجام پروژه های مهندسی در آید. اکنون که با لزوم بهینه سازی زمانبندی پروژه های مهندسی آشنا شدیم، نوبت به انتخاب روشی می رسد که بتوان در کمترین زمان ممکن، بهینه ترین برنامه زمانبندی را تهیه نماید.

"روش مسیر بحرانی (CPM)"^۱ و "روش مرور و ارزیابی برنامه ها (PERT)"^۲ برای حل مساله زمانبندی پروژه ها با فرض عدم محدودیت منابع توسعه داده شده اند. این در حالی است که در بسیاری از موارد در شرایط واقعی محدودیت منابع وجود دارد. زمانبندی پروژه در شرایط محدودیت منابع بسیار پیچیده است و لذا "مساله زمانبندی پروژه دارای محدودیت منابع (RCPSP)"^۳ در دسته مسائل NP-hard قرار می گیرد [۱]. Uzdamar و [۲] Ulsoy روشهای متعدد برای یافتن جواب بهینه^۴ مساله زمانبندی پروژه ها در شرایط محدودیت منابع را بررسی نموده اند. روشهای بهینه یابی بدلیل پیچیدگی عموماً در عمل مورد استفاده قرار نمی گیرند، لذا روشهای ابتکاری^۵ متعددی [۳-۷] برای یافتن جوابهای قابل قبول^۶ با صرف زمان معنی دار ابداع شده اند.

Badiru و [۳] Badiru & Pulat [۴] فاکتور تخصیص مرکب (CAF)^۷ را برای تخصیص منابع به هر فعالیت ایجاد نمودند. بر این اساس فعالیتی که منابع بیشتری مصرف می کند، زمانبرتر است و مدت زمان^۸ بیشتری لازم دارد، دارای فاکتور CAF بزرگتری خواهد بود. فعالیتهایی که CAF بزرگتری دارند، از اولویت بالاتری در تخصیص منابع برخوردارند. Li و Willis [۵] روشی تکراری^۹ برای زمانبندی پیشنهاد دادند. در این روش، زمانبندی روبه جلو^{۱۰} و روبه عقب^{۱۱} آنقدر تکرار می شود که دیگر امکان بهبودی در زمان تکمیل پروژه نباشد. Morse [۶] از ترکیب روشهای ابتکاری برای زمانبندی پروژه بهره جست. نتایج مطالعات وی نشان داد که روشهای ابتکاری ترکیبی نسبت به روشهای ابتکاری مجزا عملکرد بهتری دارند. در هر حال عملکرد روشهای ابتکاری از سطح قوی تا ضعیف متغیر است.

الگوریتم جستجوی ممنوع [۸-۱۰]، الگوریتمی برای حل مسائل بهینه سازی است. این الگوریتم احتمال قرار گرفتن در نقاط بهینه محلی^{۱۲} را کاهش می دهد. این روش از مجموعه ای از حرکتها^{۱۳} برای انتقال از یک جواب به جواب دیگر و از یک تابع ارزیابی^{۱۴} برای تعیین ارزش هر یک از این حرکتها استفاده می نماید. شکل روش جستجوی ممنوع بسیار متنوع است و با توجه به نوع مساله از حرکتها و توابع ارزیابی گوناگونی

¹ Critical Path Method (CPM)

² Program Evaluation and Review Technique (PERT)

³ Resource Constrained Project Scheduling Problem (RCPSP)

⁴ Optimum Solutions

⁵ Heuristic Method

⁶ Acceptable Solutions

⁷ Composite Allocation Factor (CAF)

⁸ Duration

⁹ Iterative

¹⁰ Forward

¹¹ Backward

¹² Local Optimum

¹³ Moves

¹⁴ Evaluation Function

استفاده می‌شود. این روش برای برخی از مسائل از جمله "مساله تخصیص مضاعف (QAP)"^{۱۵} [۱۲ و ۱۱] و "مسائل زمانبندی" [۱۳ و ۱۴] به جواب بهینه یا نزدیک بهینه رسیده است.

Skorin-Kapov [۱۲] در مطالعه خود بر روی مساله تخصیص مضاعف، صرف زمان کمتر CPU را با استفاده از روش جستجوی ممنوع نسبت به روشهای ابتکاری گزارش دادند. Aneja و Punnen [۱۴] الگوریتمی را برای جستجوی ممنوع پیشنهاد دادند که در مسائل NP-complete برای زمانبندی تخصیص، جوابهای با کیفیتی را ارائه می‌داد. آنها [۱۱] همچنین الگوریتمی برای مساله تخصیص با محدودیت منابع پیشنهاد دادند که نسبت به روشهای ابتکاری جوابهای بهتری ارائه می‌نمود.

در سالهای اخیر نیز روشهای متعددی برای حل مساله زمانبندی پروژه با محدودیت منابع توسعه داده شده است. از جمله Valls و همکاران [۱۵] الگوریتم فراابتکاری جدیدی را برای ارائه نموده اند که از الگوریتم جستجوی ممنوع همراه با حافظه های قرار داده شده در چارچوبی مبتنی بر داده ها استفاده می نماید و از استراتژی جستجوی جغرافیائی فضای جواب بهره می برد. Nonobe-Ibaraki [۱۶] نیز مساله را در حالتی که اجازه بازپس گیری منبع وجود دارد، حل نموده اند. همچنین مثالهایی کاربردی از مساله زمانبندی پروژه با محدودیت منابع، توسط محققانی نظیر Mika و همکاران [۱۷] برای حوزه برنامه ریزی مالی پروژه ها حل شده است.

در این مقاله روش فراابتکاری^{۱۶} جستجوی ممنوع (TS)^{۱۷} برای حل مساله زمانبندی پروژه با محدودیت منابع (RCPS) با هدف دستیابی به بهبود در کیفیت جوابها نسبت به روشهای ابتکاری موجود بکار گرفته شده است. برای سنجش عملکرد الگوریتم ارائه شده، از اطلاعات ۱۱۰ پروژه جمع آوری شده توسط Patterson [۷] در زمینه پروژههای دارای محدودیت منابع استفاده شده است.

در این مقاله الگوریتم فراابتکاری جستجوی ممنوع (TS) برای حل مساله زمانبندی پروژه با محدودیت منابع (RCPS) در هر دو حالت زمان قطعی^{۱۸} و احتمالی^{۱۹} فعالیتها بکار گرفته شده است.

هدف از ارائه این الگوریتم، دستیابی به بهبود کیفیت در جوابها نسبت به روشهای ابتکاری است. در ادامه مقاله و در بخش ۲، به تعریف مساله زمانبندی پروژه دارای محدودیت منابع (RCPS) خواهیم پرداخت. در بخش ۳ الگوریتم جستجوی ممنوع (TS) توسعه داده شده و در بخش ۴ مثالی عددی برای درک نحوه عملکرد الگوریتم جستجوی ممنوع ارائه شده است. بکارگیری الگوریتم جستجوی ممنوع (TS) برای حل مساله در فصل ۵ و نتایج حاصل از آزمایش الگوریتم در بخش ۶ آمده است.

¹⁵ Quadratic Assignment Problem (QAP)

¹⁶ Meta-heuristic

¹⁷ Tabu Search (TS)

¹⁸ Deterministic Activity Duration

¹⁹ Stochastic

۲- تعریف مساله:

۲-۱- مدل پایه محدودیت منابع (RCPSP)

فرض کنید که یک پروژه بر اساس یک شبکه AON (فعالیت ها بر روی گره ها) به صورت $G(V,E)$ تعریف شود که در آن V مجموعه تمام گره ها می باشد که نماینده فعالیت ها می باشند و E مجموعه کمان هایی که روابط پیش نیازی را به صورت FS مشخص می کنند می باشد. N تعداد فعالیت ها در پروژه، $I = \{1, 2, \dots, N\}$ مجموعه تمام فعالیت ها و Π مجموعه تمام جایگشت های^{۲۰} تعریف شده بر روی I باشد. هر جایگشت $\pi \in \Pi$ بصورت $(\pi(1), \pi(2), \dots, \pi(N))$ تعریف می شود. فعالیت های فرضی ۱ و N فعالیت های شروع و پایان پروژه و با مدت زمان صفر هستند. فعالیت ها باید بدون انقطاع صورت گیرند.

میزان طول فعالیت با d_i ($1 \leq i \leq N$) و زمان شروع آن با s_i و زمان پایان آن با f_i مشخص می شود. به تعداد K نوع منبع تجدید پذیر مفروض نیاز است به طوری که r_{ik} ($1 \leq i \leq N, 1 \leq k \leq K$) میزان ثابت نیازمندی فعالیت i به منبع k می باشد و a_k میزان ثابت در دسترس از منبع نوع k می باشد.

مساله RCPSP به صورت زیر مدل می شود [21]:

$$\text{Min } f_n \quad (1)$$

$$\text{st : } f_1 = 0 \quad (2)$$

$$f_j - d_j \geq f_i, \forall (i, j) \in H \quad (3)$$

$$\sum_{i \in S_t} r_{ik} \leq a_k, t = 1, 2, \dots, f_n; k = 1, 2, \dots, K \quad (4)$$

H مجموعه جفت فعالیت هایی است که دارای روابط پیش نیازی هستند و S_t مجموعه ای از فعالیت هاست که در بازه زمانی $[t-1, t]$ قرار دارند به طوری که: $S_t = \{f_i - d_i < t\}$

وقتی ترتیب فعالیتها در یک جایگشت π با ارتباطات پیش نیازی در پروژه سازگار باشد، یک "ترتیب ممکن"^{۲۱} نامیده می شود. این بدان معنی است که فعالیتها می توانند به همان صورت گفته شده در "ترتیب" کامل شوند. هر جایگشت، لزوماً یک ترتیب ممکن نیست.

فرض کنید F مجموعه تمام جایگشت های ممکن بر روی Π باشد. در این حالت می توان پروژه را بر اساس ترتیب ممکن مشخص شده و با رعایت محدودیت منابع زمان بندی نموده و زمان کل انجام پروژه را تعیین نمود. قبل از شروع فرایند زمان بندی، اولویت بندی نسبی^{۲۲} تمامی فعالیتها انجام می شود. این رتبه بندی نسبی معمولاً توسط برخی از روشها یا فرمولهای ابتکاری انجام می گیرد. سپس منابع بر اساس رتبه بندی اولویتی به فعالیتها تخصیص داده می شوند.

یک فعالیت امکان پذیر^{۲۳} فعالیتی است که تمامی فعالیت های الزامی قبلی اش تکمیل شده باشند. در هر لحظه زمان بندی، تنها فعالیت های ممکن برای تخصیص منابع در نظر گرفته می شوند. در لحظه k ، یک فعالیت ممکن با بالاترین اولویت، در صورت کافی بودن منابع موجود، زمان بندی می شود. وقتی در زمان k ، تعداد منابع موجود برای انجام فعالیت بعدی کافی نباشد، زمان $k+1$ در نظر گرفته خواهد شد. باید توجه داشت که تخلیه یا

²⁰ Permutation

²¹ Feasible Solution

²² Relative Priority Ranking

²³ Feasible Activity

بازپس‌گیری پیش‌هنگام^{۲۴}؛ یعنی باز پس‌گیری منبع توسط یک فعالیت، در شرایطی که قبلاً فعالیتی شروع به استفاده از آن منبع نموده است؛ مجاز نبوده و همچنین تخصیص منابع بصورت کسری (منقطع)^{۲۵} مجاز نمی‌باشد.

وقتی میزان منابع محدود باشد، استفاده از روش مسیر بحرانی بسیار مشکل می‌گردد. Bowers [۱۸] برای حل این مشکل استفاده از رابط‌های منابع^{۲۶} را پیشنهاد داده است. این رابط‌ها برای ردیابی استفاده منابع، تعیین وابستگی‌های قوی منابع که تعیین‌کننده برنامه پروژه هستند، و نیز تعیین مسیر بحرانی به کار می‌روند. در اینجا رابط‌های دیگری به نام رابط‌های تقدم^{۲۷} اضافه شده است. برای محاسبه زودترین زمان شروع (EST)^{۲۸}، رابط‌های منابع و رابط‌های تقدم مورد نظر قرار می‌گیرند. بعد از مسیر رفت، رابط‌های اضافی به شبکه افزوده می‌شوند. سپس دیرترین زمان شروع (LST)^{۲۹} بوسیله رابط‌های اصلی و رابط‌های اضافه شده بر اساس روش برگشتی شبیه آنچه که در CPM انجام می‌شود، محاسبه می‌گردد.

۲-۲- مدل RCPSP تحت روابط پیش‌نیازی عمومی (GRCPSP)

تحقیقات زیادی برای زمانبندی در حالت روابط کلی پیش‌نیازی (GPR) انجام شده است. مدل موجود در ادبیات به صورت زیر توسعه داده شده است [22]:

$$\text{Min } f_n \quad (10)$$

$$\text{st : } f_1 = 0 \quad (11)$$

$$f_j - d_j \geq f_i - d_i + SS_{ij}, \forall (i, j) \in H_1 \quad (12)$$

$$f_j \geq f_i - d_i + SF_{ij}, \forall (i, j) \in H_2 \quad (13)$$

$$f_j - d_j \geq f_i + FS_{ij}, \forall (i, j) \in H_3 \quad (14)$$

$$f_j \geq f_i + FF_{ij}, \forall (i, j) \in H_4 \quad (15)$$

$$f_i - d_i \geq g_i, i = 1, 2, \dots, N \quad (16)$$

$$f_i \leq h_i, i = 1, 2, \dots, N \quad (17)$$

$$\sum_{i \in S_t} r_{ik} \leq a_k, t = 1, 2, \dots, f_n; k = 1, 2, \dots, K \quad (18)$$

²⁴ Pre-emption

²⁵ Partial Resource Allocation

²⁶ Resource Link

²⁷ Precedence Links

²⁸ Earliest Start Time (EST)

²⁹ Latest Start Time (LST)

H_1, H_2, H_3, H_4 مجموعه جفت فعالیت هایی هستند که انواع روابط پیش نیازی را دارا هستند. g_i زمان آماده به کار فعالیت i ، h_i زمان i را $deae$ line فعالیت i می باشد.

۳- الگوریتم جستجوی ممنوع

۳-۱- نمادسازی، متغیرها و پارامترهای جستجوی ممنوع

هدف از الگوریتم جستجوی ممنوع، یافتن یک "ترتیب ممکن" در فضای جواب F است، به نحوی که برنامه بهینه یا شبه-بهینه z^* را ارائه نماید. در ادامه به برخی از تعاریف پرداخته می شود.

یک "جواب اولیه"^{۳۱}، یک ترتیب ممکن است که از یکی از روشهای ابتکاری حاصل شده است و بوسیله الگوریتم جستجوی ممنوع، بهبود خواهد یافت. در اینجا دو روش ابتکاری برای ایجاد جواب اولیه برای جستجوی ممنوع مورد استفاده قرار گرفته‌اند. روش کمترین شناوری^{۳۲} (MINSLK) بسیار ساده است. این روش، برنامه زمانی پروژه را بر اساس روابط پیش‌نیازی روش CPM و با توجه به منابع پروژه، به روشی ابتکاری ارائه می‌نماید و برای زمانبندی پروژه با در نظر گرفتن منابع به کار می‌رود. در ابتدا زمان شناوری برای CPM/PERT بدون محدودیت منابع محاسبه می‌گردد. پس از تخصیص منابع، بیشترین اولویت به فعالیت داده می‌شود که کمترین زمان شناوری محاسبه شده به روش CPM/PERT داشته باشد. عبارتی در این روش منابع به فعالیت‌هایی که زودترین تاریخ انجام آنها فرا رسیده و پیش‌نیازهای آنها تکمیل شده است، تخصیص داده می‌شود. با توجه به امکان وجود فعالیت‌های متعدد با این وضعیت، اولویت تخصیص منابع به فعالیت‌هایی داده می‌شود که بر اساس روشهای CPM/PERT دارای زمان شناوری (Slack) کمتری باشند. در صورتی که فعالیتها در زمان شناوری نیز مشابه باشند، اولویت بعدی به فعالیت‌هایی با زمان اجرای کمتر تخصیص می‌یابد. در صورت شرایط مشابه در این فاکتور نیز اولویت بعدی بر حسب شماره کوچکتر فعالیت خواهد بود. یکی دیگر از روشها جهت تولید جواب اولیه، روش ابتکاری "فاکتور تخصیص مرکب (CAF)" است. در روش CAF، بیشترین اولویت به فعالیت داده می‌شود که ارزش فاکتور CAF بیشتری داشته باشد. برای هر فعالیت i ، فاکتور CAF بعنوان جمع موزون و مقیاس‌دار دو جزء فاکتور تخصیص هزینه^{۳۳} و فاکتور زمان تصادفی فعالیت^{۳۴} محاسبه می‌گردد. در مساله حاضر، از آنجا که کیفیت جواب بدست آمده توسط الگوریتم جستجوی ممنوع (TS)، به جواب اولیه وابسته است، از روشهای ابتکاری MINSLK و CAF برای ایجاد جواب اولیه مناسب استفاده شده است.

یک "حرکت"^{۳۵}، بمعنی انتقال از یک ترتیب ممکن به ترتیب ممکن دیگر، بوسیله جابجا کردن جایگاه دو فعالیت i و j می‌باشد. "تابع هدف"^{۳۶} برابر زمان انجام پروژه برای یک ترتیب ممکن می باشد و "ارزش حرکت"^{۳۷} برابر تفاوت مقدار تابع هدف ایجاد شده توسط جابجائی فعالیتها و تابع هدف ترتیب ممکن فعلی می‌باشد. اگر ارزش حرکت منفی باشد، حرکت به نام "حرکت بهبوددهنده"^{۳۸} نامیده خواهد شد.

در الگوریتم جستجوی ممنوع، می‌توان تنها یک "لیست ممنوعه"^{۳۹} برای ثبت حرکت‌های اخیر ایجاد نمود که این لیست از پدیدار شدن مجدد یک جواب در تعداد تکرارهای معین شده، جلوگیری می‌نماید. در این مقاله ما به دلیل تفاوت خصوصیات فعالیتها در پروژه، از دو لیست ممنوعه

³⁰ Optimal or near-optimal

³¹ Starting Solution

³² Minimum Slack

³³ Resource Allocation Factor(RAF)

³⁴ Stochastic Activity Duration Factor (SAF)

³⁵ Move

³⁶ Objective Function

³⁷ Value of Move

³⁸ Improvement Move

³⁹ Tabu List

استفاده خواهیم نمود؛ چرا که فعالیتهای پروژه قابل تفکیک به دو دسته هستند: فعالیتهای بحرانی^{۴۰} و فعالیتهای غیربحرانی^{۴۱}. اگر فعالیتی در برنامه پروژه به روش CPM/PERT بر روی مسیر بحرانی قرار گرفته باشد، فعالیتی بحرانی خواهد بود و در غیر اینصورت، غیر بحرانی است. عبارتی فعالیتی که میزان لقی یا شناوری زمانی آن صفر باشد، فعالیت بحرانی خواهد بود.

در اینجا لیستی به نام TabuListC شامل فعالیتهای بحرانی و لیست ممنوعه دیگری به نام TabuListNC در برگیرنده فعالیتهای غیربحرانی خواهد بود. در حالتی که محدودیت منابع برقرار باشد، تاخیر در هر یک از فعالیتهای بحرانی، موجب ایجاد تاخیر در فعالیتهای پیامد^{۴۲} آن شده و در نتیجه به احتمال قوی زمانی کلی اتمام پروژه با تاخیر مواجه خواهد شد.

در حالتی که یک فعالیت بحرانی، در یک حرکت بسمت جلو (ابتدای پروژه) حرکت می کند (یعنی زودتر آغاز می شود)، می تواند زودتر از زمان مقرر شده تمام شود. در این حالت این فعالیت در TabuListC قرار گرفته و در وضعیت ممنوعه^{۴۳} خواهد بود و به تعداد TabuTenureC تکرار در آنجا خواهد ماند که TabuTenureC طول لیست ممنوعه برای فعالیتهای بحرانی است که بصورت تعداد تکرار و توسط طراح الگوریتم مشخص می شود. TabuListNC و TabuTenureNC نیز بصورتی مشابه استفاده می شوند، با این تفاوت که فعالیتهای غیربحرانی ممنوعه تا تعداد TabuTenureNC تکرار بسمت جلو حرکت نخواهند کرد.

عدد صحیح NumOfMove نشانگر تعداد حرکتهای داوطلبی^{۴۴} است که در هر تکرار، تولید و ارزیابی می شوند. این حرکتهای داوطلب در لیست داوطلب^{۴۵} ثبت می شوند. هر حرکت داوطلب در لیست داوطلب بوسیله یک تابع ارزیابی^{۴۶}، ارزیابی می شود. تابع ارزیابی ترکیبی از "ارزش حرکت" و یک تابع جریمه^{۴۷} است. تابع جریمه وقتی به کار می رود که برخی از محدودیتهای از پیش تعیین شده نقض شده باشند. از آنجا که در این برنامه تنها از ترتیبهای ممکن استفاده می شود، عملاً نیازی به تابع جریمه وجود ندارد.

وقتی یک ترتیب ممکن بوسیله حرکتی که حاوی یک فعالیت ممنوعه است، ایجاد شده باشد، "حرکت محدود شده ممنوعه"^{۴۸} نامیده شده و تست تنفس^{۴۹} بر روی آن انجام خواهد گرفت. تست تنفس یک فاکتور مهم برای ایجاد انطاف در الگوریتم جستجوی ممنوع است. اگر این تست ارضاء شود، وضعیت ممنوعه فعالیت در نظر گرفته نخواهد شد، در غیر اینصورت، حرکت مجاز نمی باشد. تست تنفس مورد نظر در الگوریتم حاضر بدین شکل است که اگر زمان تکمیل پروژه^{۵۰} در ترتیب ممکن جدید، کوتاهتر از بهترین زمان یافته شده تا کنون باشد، می توان وضعیت ممنوع را در نظر نگرفت. با این توضیحات، می توان چنین گفت که "حرکت مجاز"^{۵۱}، حرکتی است که محدود شده ممنوعه نبوده و یا اگر هست، تست تنفس را اضاء نماید. "بهترین حرکت"^{۵۲}، حرکتی مجاز است که بیشترین ارزش را در ارزیابی بوسیله تابع ارزیابی دارا باشد. ترتیب ایجاد کننده بهترین حرکت، بعنوان جواب اولیه برای تکرار بعدی در نظر گرفته می شود. باید توجه داشت که بهترین ترتیب، لزوماً بهبود در جواب فعلی نخواهد بود. در واقع الگوریتم جستجوی ممنوع به این طریق از افتادن در نقطه کمینه محلی^{۵۳} جلوگیری می نماید.

عملکرد الگوریتم جستجوی ممنوع به پارامترهای متعددی بستگی دارد. یکی از مهمترین پارامترها طول لیست ممنوع^{۵۴} است. اندازه مرجح برای طول لیست ممنوعه بر پایه تست عملی^{۵۵} قرار دارد. طول کوتاه لیست، به احتمال زیاد باعث افتادن جستجو در دور (حلقه) خواهد شد و بر عکس

⁴⁰ Critical Activity

⁴¹ Non-critical Activity

⁴² Successors

⁴³ Tabu Status

⁴⁴ Candidate moves

⁴⁵ Candidate list

⁴⁶ Evaluation Function

⁴⁷ Penalty Function

⁴⁸ Tabu Restricted Move

⁴⁹ Aspiration Test

⁵⁰ Duration

⁵¹ Admissible Solution

⁵² Best Move

⁵³ Local Minimum

⁵⁴ Tabu Tenure

⁵⁵ Empirical Test

در صورت بزرگ بودن طول لیست، جستجو تنها به ناحیه‌ای کوچک محدود خواهد شد. طول لیست ممنوع برای یک الگوریتم زمانبندی عموماً می‌تواند بصورت تابعی ساده از تعداد فعالیتها، نظیر جذر تعداد آنها، تعیین شود. پارامتر مهم دیگر، تعداد حرکت‌های داوطلبی⁵⁶ است که در هر تکرار در نظر گرفته می‌شود. هر دو فاکتور سرعت و کیفیت جواب، می‌تواند از طریق اتخاذ یک استراتژی مناسب برای لیست داوطلب تاثیر بپذیرد. برای محدود کردن تعداد دفعات جستجوی الگوریتم از دو متغیر "تعداد تکرار برای یافتن جواب مجاز"⁵⁷ و "تعداد تکرار برای یافتن جواب"⁵⁸ استفاده شده است.

۳-۲- الگوریتم جستجوی ممنوع پیشنهادی جهت مساله RCPS

- الگوریتم جستجوی ممنوع، برای یک مساله زمانبندی پروژه با محدودیت منابع و زمانهای مشخص فعالیتها بصورت زیر پیشنهاد می‌گردد:
- گام ۱:** یک ترتیب ابتدائی ممکن را انتخاب کرده و آنرا S_i بنامید.
- جواب اولیه را با استفاده از روش ابتکاری حداقل شناوری (MINSLK) تولید نمائید.
 - S_b (بهترین جواب یافته شده تا کنون) را برابر ترتیب ممکن S_i قرار دهید.
- **گام ۲:** فعالیتها را به دو دسته بحرانی و غیربحرانی تقسیم نمائید.
 - پروژه را بدون در نظر گرفتن محدودیت‌های منابع، با روش CPM زمانبندی کنید.
 - فعالیت‌های بحرانی و غیر بحرانی را شناسائی نمائید. (فعالیت بحرانی، فعالیتی است که زمان شناوری آن صفر می‌باشد).
 - **گام ۳:** مقداردهی اولیه به متغیرهای مورد استفاده در الگوریتم جستجوی ممنوع را انجام دهید.
 - ماتریس تقدم (پیش‌نیازی) را برای پروژه در حالت دارای محدودیت منابع بسازید.
 - لیست‌های ممنوع TabuListC و TabuListNC را خالی نمائید.
 - مقادیر TabuTenureC و TabuTenureNC را مشخص نمائید.
 - تعداد حرکت‌های داوطلب (Candidate Move) در لیست داوطلب (Candidate List) را توسط متغیر NumOfMove مشخص نمائید.
 - مقادیر لازم را به شروط توقف، یعنی MaxTryOnAdmissible و MaxTryOnBetter تخصیص دهید.
 - مقادیر اولیه متغیرهای NotFindAdmissible (تعداد تکرارهایی که در آنها هیچ حرکت مجازی پیدا نشده است) و NotFindBetter (تعداد تکرارهایی که در آنها هیچ ترتیب ممکن بهتری از S_b یافت نشده است) را برابر صفر قرار دهید. این متغیرها در واقع شمارنده‌های تعداد تکرارهای ناموفق هستند.
 - **گام ۴:** لیست داوطلب را که حاوی تعداد NumOfMove حرکت داوطلب است، ایجاد نمائید.
 - دو فعالیت از ترتیب فعلی بصورت تصادفی انتخاب می‌شوند و مکان این دو فعالیت با هم تعویض می‌گردد. اگر ترتیب ایجاد شده ممکن نیست (پیش‌نیازی‌ها را رعایت نمی‌کند)، زوج دیگری را انتخاب نمائید.
 - این کار را آنقدر ادامه دهید که به تعداد NumOfMove جابجائی پیدا شود.
 - **گام ۵:** بهترین حرکت داوطلب مجاز را انتخاب کنید.
 - متغیرهای بولین FindAdmissible و FindBetter را برابر False قرار دهید.
 - برای هر حرکت داوطلب در لیست داوطلب، کارهای زیر را انجام دهید:

⁵⁶ Candidate Move

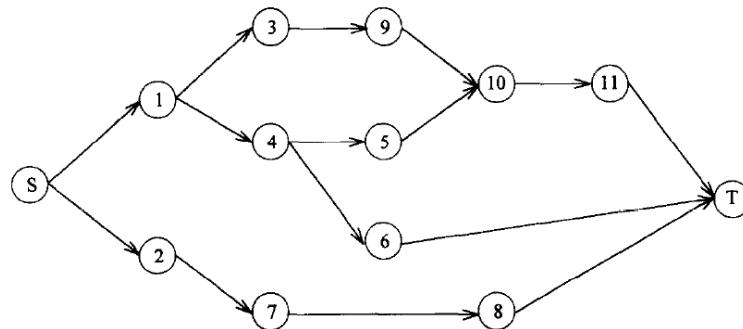
⁵⁷ MaxTryOnAdmissible

⁵⁸ MaxTryOnBetter

- فرض کنید S_j ترتیب ممکن بعد از ایجاد جابجائی در S_i باشد.
 - ارزش حرکت $(f(S_j) - f(S_i))$ را محاسبه نمائید.
 - اگر ارزش حرکت، از ارزش همه حرکت‌های مجاز پیدا شده تا کنون در لیست داوطلب، بیشتر است، آنگاه:
 - وضعیت ممنوعیت حرکت را بررسی کنید. اگر فعالیتی که بسمت انتهای شبکه حرکت می‌کند، فعالیت بحرانی است و در لیست TabuListC قرار دارد یا فعالیتی که بسمت ابتدای شبکه جابجا می‌شود، فعالیتی غیربحرانی است و در لیست TabuListNC قرار دارد، در اینصورت این جابجائی، جابجائی محدود شده ممنوع است.
 - اگر جابجائی، محدود شده ممنوع نباشد، آنگاه:
 - این جابجائی را بعنوان بهترین حرکت مجاز داوطلب پذیرفته و با M_b نمایش دهید.
 - ارزش متغیر FindAdmissible را True نمائید.
 - مقدار متغیر NotFindadmissible را مجدداً برابر صفر قرار دهید.
 - در غیر اینصورت:
 - این جابجائی، تست Aspiration را خواهد گذراند، در صورتیکه $f(S_j) < f(S_b)$.
 - اگر حرکت از تست Aspiration عبور کرد، آنگاه:
 - این جابجائی را بعنوان بهترین حرکت مجاز داوطلب پذیرفته و آنرا با M_b نمایش دهید.
 - ارزش متغیر FindAdmissible را True نمائید.
 - مقدار متغیر NotFindadmissible را مجدداً برابر صفر قرار دهید.
 - **گام ۶:** بهترین حرکت مجاز داوطلب را انجام دهید.
 - فرض کنید S_k ترتیب ممکن پس از اعمال حرکت M_b بر روی S_i باشد.
 - اگر $f(S_j) < f(S_b)$ باشد، آنگاه S_j جایگزین S_b شده و ارزش متغیر FindBetter برابر True می‌گردد.
 - اگر ارزش متغیر FindAdmissible برابر True نیست، مقدار NotFindAdmissible+1 را در NotFindAdmissible قرار دهید.
 - اگر ارزش متغیر FindBetter برابر True نیست، مقدار NotFindBetter+1 را در NotFindBetter قرار دهید.
 - **گام ۷:** شروط توقف را بررسی کنید.
 - اگر مقدر $\text{NotFindAdmissible} \geq \text{MaxTryOnAdmissible}$ است یا $\text{NotFindBetter} \geq \text{MaxTryOnBetter}$ است، در اینصورت S_b جواب خواهد بود. در غیر اینصورت به گام ۸ بروید.
 - **گام ۸:** محدودیتهای ممنوعه و معیار تنفس را بروز نموده و سپس به گام ۴ بروید.
 - TabuListC و TabuListNC را بروز نمائید.
- نکته قابل ذکر این است که از آنجا که ممکن است ترکیب دو یا چند جابجائی، جواب بهتری را تولید نماید، در **گام ۵**، M_b ممکن است تنها یک جابجائی نبوده، بلکه ترکیبی از چند جابجائی باشد. بنابراین مقایسه بین جوابهای ترتیبهای ممکن تولید شده، ترتیبهای حاصل از چند جابجائی را نیز در بر خواهد گرفت. عبارتی الگوریتم ابتدا جابجائی‌های تکی را اعمال نموده و ارزش آنها را محاسبه می‌کند. سپس جابجائی‌های دوتائی، سه‌تائی و ... و NumOfMove تائی را نیز اعمال نموده و ارزش این جابجائی‌ها را نیز محاسبه می‌نماید و نهایتاً ترتیب بهینه جدید را تعیین خواهد نمود.

۴- مثال عددی

در اینجا برای تشریح بهتر الگوریتم، از یکی از نمونه پروژه های ارائه شده توسط پترسون [۷] استفاده می کنیم. شبکه پروژه مورد نظر (بصورت فعالیتها بر روی گرهها^{۵۹}) و منابع مورد نیاز در شکل ۱ و جدول ۱ آورده شده اند. این پروژه نمونه دارای سه نوع منبع مختلف (از نوع غیرمصرفی) است که آنها را بترتیب با A، B و C نشان می دهیم و موجودی آنها بترتیب برابر ۶، ۷ و ۶ عدد است.



شکل ۱- شبکه AON برای یک پروژه نمونه [۷]

جدول ۱- مدت اجرا، نیازمندیهای منابع و زمانبندی بهینه یک پروژه نمونه

شماره فعالیت (گره)	مدت اجرای فعالیت (روز)	نیازمندیهای منابع	زمان شروع اجرا در برنامه بهینه
۱	۳	۱،۲،۳	۰
۲	۵	۲،۴،۲	۰
۳	۶	۲،۱،۳	۵
۴	۲	۱،۳،۴	۳
۵	۳	۳،۰،۲	۹
۶	۳	۱،۱،۱	۱۲
۷	۴	۱،۱،۳	۵
۸	۵	۲،۲،۲	۱۲
۹	۴	۳،۲،۳	۱۱
۱۰	۲	۰،۱،۴	۱۵
۱۱	۳	۲،۴،۵	۱۷

نیازمندی ها به ترتیب از راست به چپ مربوط به منابع A، B و C می باشند.

روش ابتکاری MINSLK برای ایجاد یک ترتیب اولیه برای این پروژه به کار گرفته می شود. زمان تکمیل پروژه برای این ترتیب، یعنی ۲۲ روز، بوسیله بکارگیری روش ذکر شده در ابتدای این بخش بدست می آید. نتایج حاصل از بکارگیری الگوریتم جستجوی ممنوع بر روی این پروژه نمونه در جدول ۲ آورده شده است.

⁵⁹ Activity On Node (AON)

جدول ۲- برنامه زمانبندی استخراج شده بوسیله الگوریتم جستجوی ممنوع

شماره فعالیت (گره)	مدت اجرا	زودترین زمان شروع (EST)	زودترین زمان تکمیل (ECT)	دیرترین زمان شروع فعالیت (LST)	دیرترین زمان تکمیل (LCT)	زمان شناوری	بحرانی بودن
۱	۳	۰	۳	۰	۳	۰	۰
۲	۵	۰	۵	۰	۵	۰	۱
۳	۶	۵	۱۱	۵	۱۱	۰	۱
۴	۲	۳	۵	۳	۵	۰	۱
۵	۳	۹	۱۲	۹	۱۲	۰	۱
۶	۳	۱۲	۱۵	۱۲	۱۵	۰	۱
۷	۴	۵	۹	۵	۹	۰	۱
۸	۵	۱۲	۱۷	۱۲	۱۷	۰	۱
۹	۴	۱۱	۱۵	۱۱	۱۵	۰	۱
۱۰	۲	۱۵	۱۷	۱۵	۱۷	۰	۱
۱۱	۳	۱۷	۲۰	۱۷	۲۰	۰	۱

ترتیب اجرای فعالیتها: ۱-۲-۳-۴-۵-۶-۷-۸-۹-۱۰-۱۱
 زمان کل اجرای پروژه: ۲۰ روز

در این جدول، EST برابر زودترین زمان شروع فعالیت، ECT برابر زودترین زمان تکمیل فعالیت، LST دیرترین زمان شروع فعالیت، LCT دیرترین زمان تکمیل فعالیت، Slack برابر زمان شناوری یا لقی هر فعالیت و Critical نشانگر بحرانی بودن فعالیت می باشد. جواب حاصله از الگوریتم جستجوی ممنوع، زمان تکمیلی برابر ۲۰ روز را ارائه می دهد که دقیقاً برابر با جواب بهینه بدست آمده از روشهای قطعی مطابق جدول ۱ می باشد.

۵- بکارگیری الگوریتم جستجوی ممنوع در حالت احتمالی

روشهای مورد استفاده فعلی برای یافتن نقطه بهینه، نظیر روشهای برنامه ریزی عدد صحیح^{۶۰}، تنها برای مسائلی با زمانبندی مشخص قابل کاربرد هستند، اما در شرایط عملی، زمان واقعی مورد نیاز برای اجرای فعالیتها، قبل از انجام آنها مشخص نیست. در این شرایط استفاده صرف از زمان انتظاری^{۶۱} اجرای فعالیت و بکارگیری روشهای زمانبندی مسائل قطعی، اثرات محتمل ناشی از احتمالی بودن زمان فعالیتها را در نظر نمی گیرد. در اینجا با ایجاد تغییر کوچک در تابع ارزیابی، الگوریتم جستجوی ممنوع برای حالت قطعی را توسعه داده و در حالت احتمالی نیز از آن استفاده خواهیم نمود.

در این حالت بجای استفاده از زمان انتظاری انجام فعالیت در محاسبه زمان کلی اجرای پروژه، این زمان را بر اساس داده های قبلی یا توزیعهای احتمالی تعیین می گردد. در این مقاله برای مدلسازی زمان انجام فعالیتها از توزیع β استفاده شده است، اما در حالت کلی می توان از هر یک از توزیعهای احتمالی برای این کار استفاده نمود. سه تخمین زمانی مورد استفاده در روش PERT، یعنی تخمین زمانی خوشبینانه a ، تخمین

⁶⁰ Integer Programming

⁶¹ Expected Duration

زمانی محتمل ترین زمان m و تخمین زمانی بدبینانه b برای محاسبه پارامترهای توزیع بتا بکار گرفته می‌شوند. فرمولهای ارائه شده توسط Badiru [۳] و Badiru & Pulat [۴] برای محاسبه پارامترهای α و β توزیع بتا بصورت روابط زیر خواهند بود:

$$\phi = \frac{5a - 4m - b}{a + 4m - 5b}, \quad (1)$$

$$\alpha = \phi\beta, \quad (2)$$

$$\beta = \frac{-(\phi^2 - 34\phi + 1)}{(\phi + 1)^3}. \quad (3)$$

از آنجا که زمانهای انجام فعالیتهای بصورت تصادفی توزیع شده‌اند، زمان کلی انجام پروژه نیز یک متغیر تصادفی احتمالی خواهد بود. با وارد کردن یک ترتیب ممکن (که بوسیله استفاده از زمان انتظاری هر فعالیت و روش MINSLK یا بوسیله الگوریتم جستجوی ممنوع ایجاد شده‌است)، زمان انتظاری برای تکمیل پروژه بصورت زیر محاسبه می‌شود:

۱. برای هر فعالیت، یک زمان انجام بصورت تصادفی بوسیله توزیع بتا با پارامترهای محاسبه شده از طریق سه تخمین زمانی، مشخص می‌شود.

۲. با داشتن ترتیب ممکن و زمان تصادفی تولید شده برای هر فعالیت، زمان انجام کل پروژه محاسبه می‌گردد.

۳. محاسبه زمان انجام پروژه به تعداد دفعات مشخص (مثلا ۱۰۰ بار) با مجموعه‌های مختلفی از زمانهای انجام فعالیتهای صورت گرفته و میانگین نتایج حاصله بعنوان زمان انتظاری انجام پروژه برای هر ترتیب ممکن در نظر گرفته می‌شود.

بعبارت دیگر، ترتیب ممکن ابتدا بکمک زمانهای انتظاری مشخص می‌گردد و سپس زمان متوسط ترتیب مشخص شده، تعیین می‌شود. بعنوان یک نمونه، الگوریتم جستجوی ممنوعه برای حالت احتمالی شبکه پروژه نشان داده شده در شکل ۱ بکار گرفته شده است. تخمین‌زننده زمانی خوشبینانه a برابر 0.8 زمان قطعی فعالیت، تخمین‌زننده زمانی محتمل ترین زمان m برابر زمان قطعی انجام فعالیت و نهایتاً تخمین‌زننده بدبینانه b ، 1.5 برابر زمان قطعی انجام فعالیت در نظر گرفته شده است. با استفاده از این مقادیر برای a ، m و b زمانهای انتظاری محاسبه شده برای هر فعالیت بیشتر از زمان بهینه و 1.05 برابر آن بدست آمده است.

با استفاده از این زمانهای انتظاری و بکمک روش MINSLK، جوابهای اولیه برای الگوریتم جستجوی ممنوع محاسبه شده‌اند. زمان انتظاری برای اجرای کل پروژه برای ترتیب ممکن اولیه، به روش فوق محاسبه شده است که با استفاده از اندازه نمونه ۱۰۰ تائی، متوسط زمان اجرا برابر 24.346 روز می‌باشد.

جدول ۳- میانگین زمان اجرای پروژه‌های نمونه در حالت زمانهای احتمالی بوسیله الگوریتم جستجوی ممنوع [۱۹]

شماره فعالیت (گره)	مدت اجرا	زودترین زمان شروع (EST)	زودترین زمان تکمیل (ECT)	دیرترین زمان شروع فعالیت (LST)	دیرترین زمان تکمیل (LCT)	زمان شناوری	بحرانی بودن
۱	۳,۱۷۰	۰,۰۰۰	۳,۱۷۰	۰,۱۵۴	۳,۳۲۴	۰,۱۵۴	۰,۶۳
۲	۵,۱۹۹	۰,۰۰۰	۵,۱۹۹	۰,۳۵۵	۵,۵۵۴	۰,۳۵۵	۰,۳۷
۳	۶,۲۷۳	۵,۲۷۰	۱۱,۵۴۴	۵,۸۵۴	۱۲,۱۲۷	۰,۵۸۴	۰,۳۶
۴	۲,۱۰۰	۳,۱۷۰	۵,۲۷۰	۳,۳۲۴	۵,۴۲۴	۰,۱۵۴	۰,۶۳
۵	۳,۱۸۰	۵,۵۲۹	۸,۷۰۸	۵,۷۰۵	۸,۸۸۴	۰,۱۷۶	۰,۶۴
۶	۳,۱۵۷	۱۷,۶۲۱	۲۰,۷۷۸	۱۸,۵۷۸	۲۱,۷۳۶	۰,۹۵۷	۰,۱۰
۷	۴,۲۸۳	۸,۷۰۸	۱۲,۹۹۱	۸,۸۸۴	۱۳,۱۶۷	۰,۱۷۶	۰,۶۴

۰,۶۶	۰,۱۹۷	۱۸,۵۱۷	۱۳,۲۲۹	۱۸,۳۲۰	۱۳,۰۳۲	۵,۲۸۸	۸
۰,۳۴	۰,۵۸۸	۱۶,۳۲۵	۱۲,۱۳۲	۱۵,۷۳۶	۱۱,۵۴۴	۴,۱۹۳	۹
۰,۳۴	۰,۵۸۸	۱۸,۴۴۶	۱۶,۳۲۵	۱۷,۸۵۷	۱۵,۷۳۶	۲,۱۲۱	۱۰
۰,۹۰	۰,۰۲۹	۲۱,۷۳۶	۱۸,۵۸۶	۲۱,۷۰۶	۱۸,۵۵۷	۳,۱۵۰	۱۱

ترتیب اجرای فعالیتها: ۱-۲-۳-۴-۵-۶-۷-۸-۹-۱۰-۱۱-۱۲

زمان کل اجرای پروژه: ۲۱,۷۰۶ روز

نتایج حاصل از بکارگیری الگوریتم جستجوی ممنوع برای حالت احتمالی در جدول ۳ آورده شده است [۱۹]. این جدول میانگین نمونه‌ای متغیرهای مربوط به برنامه مشخص شده به روش جستجوی ممنوع، و نیز جواب مساله را که میانگینی برابر 21.706 دارد را نشان می‌دهد. این مقدار بهبودی به اندازه 10.8% نسبت به جواب ارائه شده با بکارگیری روش MINSLK را می‌دهد.

۶- نتایج محاسباتی الگوریتمهای SA و TS؛ بعنوان پایه ای برای مقایسه الگوریتم جدید

همانطور که قبلا اشاره شد، پترسون [۷] تعداد ۱۱۰ پروژه نمونه را در زمینه پروژه‌های دارای محدودیت منابع ارائه داده است. در ادامه برای اثبات برتری الگوریتم TS جدید، از اطلاعات این ۱۱۰ پروژه و حل آنها بوسیله الگوریتمهای قدیمی TS و نیز الگوریتم آنیلینگ شبیه‌سازی شده (SA)^{۶۲} (یکی دیگر از روشهای فراابتکاری قابل استفاده برای حل مساله زمانبندی) استفاده می‌کنیم. بعلاوه مقایسه‌ای بین نتایج حاصل از بکارگیری الگوریتم جستجوی ممنوع و نتایج ایجاد شده توسط الگوریتم آنیلینگ شبیه‌سازی شده در اینجا اشاره شده است.

لازم به ذکر است که تعداد ۱۰۸ پروژه از ۱۱۰ پروژه ارائه شده توسط پترسون، توسط Morse [۶] و بوسیله روش ابتکاری MINSLK حل شده‌اند و جوابهای آنها در دسترس است. درصد افزایش زمان تکمیل پروژه‌ها حاصله از حل بوسیله روش MINSLK، برابر 10.9% و تعداد پروژه هائی که زمان بهینه اجرای آنها یافت شده است، برابر ۲۰ پروژه می‌باشد [۱۹].

جدول ۴ نتایج حاصل از ۱۰ بار آزمایش بر روی ۱۱۰ پروژه را با استفاده از الگوریتم SA نشان می‌دهد [۲۰]. N تعداد فعالیتها در هر پروژه را نشان می‌دهد و $N(T)$ نشاندهنده تعداد برنامه‌های تولید شده در هر درجه حرارت در فرایند آنیلینگ است. "Above Optimum" نشاندهنده میانگین افزایش در زمان انجام پروژه نسبت به زمانهای بهینه ارائه شده توسط پترسون است. همچنین "Optimum Obtained" بیانگر درصد جوابهای بهینه یافته شده و "Optimum found in all trials" بیانگر تعداد پروژه‌هائی است که جواب بهینه آنها در تمامی آزمونها بدست آمده است. نهایتا "Execution Time" نشان دهنده میانگین زمان انجام زمانبندی برای تمامی پروژه‌ها در تمامی آزمونهاست. در صورت بکارگیری روش MINSLK (نشان داده شده در جدول ۵)، بهترین نتایج به ازای $N(T)=10N$ بدست آمده است که در آن میانگین افزایش زمان انجام کل پروژه نسبت به حالت بهینه، 0.229% بوده و متوسط زمان اجرا نیز 2.9731 ثانیه است. با استفاده از فاکتور CAF برای تولید جواب ابتدائی نیز همین نتایج حاصل می‌گردد. در این حالت نیز بهترین نتیجه به ازای $N(T)=10N$ حاصل می‌شود که میانگین افزایش زمان آن نسبت به حالت بهینه برابر 0.197% و متوسط زمان اجرای آن 2.9978 ثانیه بوده است.

جدول ۴- نتایج حاصل از بکارگیری الگوریتم SA برای مساله زمان قطعی؛ با استفاده از روش MINSLK برای تولید جواب اولیه [۲۰]

10N	5N	3N	2N	N	N(T)
% ۰,۲۳	% ۰,۳۳	% ۰,۴۳	% ۰,۵۵	% ۰,۸۷	تعداد برنامه‌های تولید شده بالاتر از بهینه
% ۹۱,۹۱	% ۸۹,۱۸	% ۸۶,۱۸	% ۸۳,۶۴	% ۷۷,۰۰	میزان جوابهای بهینه بدست آمده

⁶² Simulated Annealing (SA)

۸۴	۷۳	۶۲	۵۶	۴۷	تعداد جوابهای بهینه پیدا شده در همه تکرارها
۲,۹۷۳	۱,۵۱۳	۰,۹۱۸	۰,۶۲۸	۰,۳۱۷	زمانهای اجرا (ثانیه)
۳,۶۵۴	۱,۸۵۵	۱,۱۳۸	۰,۷۷۹	۰,۳۸۱	انحراف معیار زمانهای اجرا (ثانیه)

الگوریتم جستجوی ممنوع نیز برای همه ۱۱۰ پروژه ارائه شده پترسون بکار گرفته شده است [۱۹]. نتایج عملی حاصل از بکارگیری الگوریتم جستجوی ممنوع در دو بخش ارائه شده است که عبارتند از الگوریتم جستجوی ممنوع برای پروژه‌هایی با زمان قطعی انجام فعالیتها و الگوریتم جستجوی ممنوع برای پروژه‌هایی با زمان تصادفی انجام فعالیتها. مقادیر متغیرهای بکار رفته در الگوریتم بر پایه تستهای عملی قرار دارد. با فرض N بعنوان تعداد کل فعالیتها، مقدار $TabuTenureC$ و $TabuTenureNC$ برابر $\frac{1}{2}\sqrt{N}$ در نظر گرفته شده است. همچنین متغیر $NumOfMove$ نشانگر تعداد حرکتهای در لیست داوطلب برابر \sqrt{N} لحاظ شده است. مقادیر مختلف شرط توقف^{۶۳}، $MaxTryOnAdmissible$ و $MaxTryOnBetter$ برای ارزیابی عملکرد الگوریتم جستجوی ممنوع بکار گرفته شده اند. از آنجا که برنامه همسایگی بصورت تصادفی تولید می‌شود، برای تنظیم مقدار هر پارامتر، برنامه ۱۰ بار اجرا شده است. نتایج حاصله از ۱۰ بار اجرای هر یک از ۱۱۰ پروژه نمونه، که جواب اولیه آنها به روش $MINSLK$ محاسبه شده، در جدول ۵ ذکر گردیده است. میزان بهبود در زمانهای انجام پروژه، برابر میانگین درصد بهبود زمان انجام پروژه نسبت به جواب اولیه ایجاد شده توسط روش $MINSLK$ می‌باشد. همانطور که براحتی قابل مشاهده است، نتایج، از روش بکار گرفته شده توسط $Morse$ [۶] بهتر است.

جدول ۵- نتایج حاصل از بکارگیری الگوریتم جستجوی ممنوع برای مساله زمان قطعی؛ با استفاده از روش $MINSLK$ برای تولید جواب اولیه [۱۹]

۲۰۰۰۰	۱۰۰۰۰	۶۰۰۰	۳۰۰۰	۱۰۰۰	بیشتر تکرار برای یافتن جواب مجاز ($MaxTryOnAdmissible$)
۲۰۰۰	۱۰۰۰	۶۰۰	۳۰۰	۱۰۰	بیشتر تکرار برای یافتن جواب بهتر ($MaxTryOnBetter$)
% ۰,۱۹	% ۰,۲۹	% ۰,۴۰	% ۰,۶۹	% ۱,۴۰	تعداد جوابهای بالاتر از بهینه
% ۱۰,۳۲	% ۱۰,۲۳	% ۱۰,۱۳	% ۹,۸۹	% ۹,۲۹	میزان بهبود در زمانهای انجام پروژه
% ۹۳,۴۶	% ۹۰,۶۴	% ۸۷,۴۶	% ۸۱,۳۶	% ۶۶,۶۴	میزان جوابهای بهینه بدست آمده
۹۵	۸۷	۷۶	۶۲	۴۳	تعداد جوابهای بهینه پیدا شده در همه تکرارها
۳,۳۹۶	۱,۸۵۳	۱,۱۷۳	۰,۶۸۷	۰,۳۸۳	زمانهای اجرا (ثانیه)
۳,۶۹۴	۲,۰۷۴	۱,۳۰۳	۰,۸۱۳	۰,۳۵۳	انحراف معیار زمانهای اجرا (ثانیه)

با مقایسه نتایج ذکر شده در جداول ۴ و ۵، می‌توان نتیجه گرفت که در شرایطی که زمان محاسباتی خیلی کوچک (متوسط 0.3 ثانیه) است، الگوریتم SA بهتر از الگوریتم TS عمل می‌نماید. وقتی که متوسط زمان محاسباتی به بیش از 0.3 ثانیه می‌رسد، به نظر می‌رسد که الگوریتم جستجوی ممنوع در یافتن جوابهای بهینه زمانهای اجرای پروژه توانمندتر است. در این آزمایشها بیش از 93% جوابهای بهینه و در کل جوابهای بهینه برای ۹۵ پروژه با میانگین زمان اجرای الگوریتم برابر 3.4 ثانیه، توسط الگوریتم جستجوی ممنوع بدست آمده است. نتایج مشابه برای الگوریتم SA ارقامی برابر دستیابی به بیش از 91% از جوابهای بهینه و یافتن جواب بهینه ۸۴ پروژه در همه تکرارها با میانگین زمانی اجرای الگوریتم برابر 3.0 ثانیه را نشان می‌دهد. با استفاده از فاکتور CAF برای ایجاد جواب اولیه نیز نتایج مشابهی حاصل خواهد شد. در استفاده از الگوریتم جستجوی ممنوع برای پروژه‌هایی با فعالیتهای دارای زمانی احتمالی، برنامه‌ای (ترتیبی) که کمترین زمان اجرای کلی پروژه را می‌دهد، بعنوان بهترین برنامه یافت شده بوسیله جستجوی ممنوع معرفی می‌گردد. در این حالت بجای استفاده از زمان قطعی انجام فعالیت، سه تخمین زمانی به روشی که قبلاً ذکر شد، یعنی تخمین a بعنوان زمان خوشبینانه، تخمین m بعنوان محتملترین زمان و تخمین b بعنوان زمان بدبینانه مورد استفاده قرار می‌گیرد. در این حالت زمان انتظاری انجام فعالیت 1.05 برابر زمان قطعی انجام فعالیت خواهد بود.

⁶³ Stopping Criteria

برای ارزیابی نتایج حاصل از بکارگیری الگوریتم جستجوی ممنوع (TS) برای حالت زمانهای احتمالی، زمان اجرای کلی پروژه در دسترس نیست. لذا در این حالت زمان بهینه شناخته شده در حالت قطعی را در 1.05 ضرب نموده و آنرا بعنوان یک حد پائین تخمینی برای ارزیابی عملکرد الگوریتم بکار می‌بریم. جدول ۶ نتایج حاصل از بکارگیری الگوریتمهای SA و TS را با استفاده از جوابهای اولیه مختلف نمایش می‌دهد. در این جدول سطر Above approximate lower bound نشانگر متوسط درصد افزایش در زمان اجرای پروژه نسبت به حد پائین تخمینی، Project duration improved نشاندهنده درصد بهبود در زمان اجرای پروژه، بهبود بیش از ۱۰٪ بیانگر تعداد پروژه‌هایی که در آنها مقدار Project duration improved بیشتر از 10٪ است، Adm برابر ارزش MaxTryOnAdmissible و Better برابر ارزش MaxTryOnBetter می‌باشد.

جدول ۶- نتایج حاصل از بکارگیری الگوریتمهای SA و TS در مساله زمان احتمالی؛ با استفاده از روش MINSLK برای تولید جواب اولیه [۱۹]

جستجوی ممنوع (TS)		آنیلینگ شبیه سازی شده (SA)		
Adm=50 Better=500	Adm=25 Better=250	N(T)=2N	N(T)=N	
٪ ۲,۵۴	٪ ۳,۷۱	٪ ۲,۲۷	٪ ۳,۴۰	بالتر از مرز تخمینی پائین (Above approximate lower bound)
٪ ۹,۵۳	٪ ۸,۵۲	٪ ۹,۶۲	٪ ۸,۶۴	میزان بهبود در زمانهای انجام پروژه (Project duration improved)
۴۷	۴۳	۴۸	۴۵	بهبود بیش از ۱۰٪
۲۷	۲۱	۲۷	۲۲	بهبود بیش از ۱۵٪
۸	۶	۱۱	۷	بهبود بیش از ۲۰٪
۱۱,۲۹۰	۵,۸۳۴	۲۱,۴۱۴	۱۰,۸۰۴	زمانهای اجرا (ثانیه)
۸,۸۲۳	۴,۴۵۳	۱۹,۷۰۸	۹,۸۶۸	انحراف معیار زمانهای اجرا (ثانیه)

بر اساس نتایج بدست آمده، قابل مشاهده است که زمان محاسباتی مورد نیاز برای دستیابی به نتایج مشابه در الگوریتم جستجوی ممنوع تقریباً برابر نصف الگوریتم SA است. زمان اجرای پروژه در تقریباً نیمی از ۱۱۰ پروژه بیش از 10٪ کاهش و در برخی از پروژه‌ها نیز حدود 20٪ کاهش نسبت به جواب اولیه بدست آمده توسط روش ابتکاری MINSLK را نشان می‌دهد. بطور متوسط نیز زمان اجرای پروژه‌ها نسبت به جواب اولیه حدوداً 9٪ کاهش یافته است. در اینجا نیز در صورت استفاده از فاکتور CAF برای تولید جوابهای اولیه، همین نتایج حاصل می‌گردد.

۷- نتیجه گیری

هدف ما در این تحقیق، ارائه الگوریتمی ابتکاری بود که جوابهای بدست آمده برای زمانبندی فعالیتهای پروژه‌های دارای محدودیت منابع را بهبود دهد. بر پایه نتایج عملی که بدست آوردیم، می‌توان الگوریتم جستجوی ممنوع پیشنهادی را بعنوان روشی کارا برای این هدف در هر دو مساله زمان قطعی و زمان احتمالی مطرح نمود.

مراجع:

- [1] J. Blazewicz, J.K. Lenstra, A.H.G. Rinnooy Kan, Scheduling Subject to Resource Constraints: classification and complexity, *Discrete Applied Mathematics* 5 (٢٠٠٥) 1-24.
- [2] L. Ozdamar, G. Ulusoy, A Survey on the Resource Constrained Project Scheduling Problem, *IIE Transactions* 27 (199٩) 574-586.
- [3] A.B. Badiru. A Simulation Approach to PERT Network Analysis, *Simulation* 57 (٢٠٠٠) 245-255.
- [4] A.B. Badiru, P.S. Pulat, *Comprehensive Project Management*, Prentice-Hall, Englewood Cliffs, NJ, 1995.
- [5] K.Y. Li, R.J. Willis, An Iterative Scheduling Technique for Resource Constrained Project Scheduling, *European Journal of Operational Research* 56 (٢٠٠٢) 370-379.
- [6] L.C. Morse, J.O. McIntosh, G.E. Whitehouse, Using Combinations of Heuristics to Schedule Activities of Constrained Multiple Resource Projects, *Project Management Journal* (٢٠٠٤) 34-40.
- [7] J. Patterson, A comparison of exact procedures for solving the multiple constrained resource project scheduling problems, *Management Science* 30 (٢٠٠٤) 854-867.
- [8] F. Glover, Tabu Search, Part I, *ORSA Journal of Computing* 1 (٢٠٠٥) 190-206.
- [9] F. Glover, Tabu Search, Part II, *ORSA Journal of Computing* 2 (٢٠٠٠) 4-32.
- [10] F. Glover. Tabu Search: A Tutorial, *Interfaces* 20 (٢٠٠٢) 74-94.
- [11] A.P. Punnen, Y.P. Aneja, A Tabu Search Algorithm for the Resource Constrained Assignment Problem, *Journal of the Operational Research Society* 46 (٢٠٠٣) 214-220.
- [12] F. Skorin-Kapov, Tabu Search Applied to Quadratic Assignment Problem, *ORSA Journal of Computing* 2 (٢٠٠٦) 33-45.
- [13] F. Glover, C. McMillan, The General Employee Scheduling Problem: An Integration of Management Science and Artificial Intelligence, *Computer and Operations Research* 13 (٢٠٠٤) 563-3593.
- [14] A.P. Punnen, Y.P. Aneja, Categorized Assignment Scheduling: A Tabu Search Approach, *Journal of the Operational Research Society* 44 (٢٠٠٠) 673-679.
- [15] Vicente Valls, Sacramento Quintanilla, Francisco Ballestin; Resource-Constrained Project Scheduling: A critical Activity Reordering Heuristic, *European Journal of Operational Research* 149 (2003) 282–301.
- [16] Koji Nonobe, Toshihide Ibaraki, A Tabu Search Algorithm for a Generalized Resource Constrained Project Scheduling Problem, *The Fifth Metaheuristics International Conference (MIC)*, 2003.
- [17] Marek Mika, Grzegorz Waligora, Jan Wezglarz, Simulated Annealing and Tabu Search for Multi-mode Resource-Constrained Project Scheduling with Positive Discounted Cash Flows and Different Payment Models, *European Journal of Operational Research* 164 (2005) 639–668.
- [18] J.A. Bowers, Criticality in Resource Constrained Networks, *Journal of the Operational Research Society* 46 (٢٠٠٠) 80-91.

- [19] Y. Tsai, D.D. Gemmill, Identifying the Critical Path in Resource Constrained Projects, Working Paper No. 96131, Department of Industrial and Manufacturing Systems Engineering, Iowa State University, IA, ۲۰۰۶.
- [20] Y. Tsai, D.D. Gemmill, Using A Simulated Annealing Algorithm to Schedule Activities of Resource-Constrained Projects, Working Paper No. 96-124, Department of Industrial and Manufacturing Systems Engineering, Iowa State University, IA, 1996, Project Management Journal, to appear.
- [21] Willy Herroelen, Bert De Reyck, and Erick Demeulemeester “Resource-constrained project scheduling: A survey of recent developments” s0305—0548(97)0005s-

[22] برزین پور، فرناز- شاکری، آرنوش- فرزاد، احسان، حل مساله زمانبندی پروژه های دارای محدودیت منابع با استفاده از الگوریتم جستجوی ممنوع، مجموعه مقالات سومین کنفرانس بین المللی مدیریت پروژه، ۱۳۸۵