

معرفی یک روش فرا ابتکاری جدید به نام BOB برای حل مساله زمانبندی پروژه با منابع محدود

محمد صادق سلیمانی اصطهباناتی ، رضا اکبری ، مصطفی فخر احمد

دانشجوی ارشد ، گروه نرم افزار کامپیوتر ، دانشگاه آزاد اسلامی ، واحد علوم تحقیقات ، فارس

عضو هیئت علمی گروه فناوری اطلاعات دانشگاه صنعتی شیراز

عضو هیئت علمی گروه فناوری اطلاعات، دانشکده مهندسی ، دانشگاه شیراز

S.soleimani29@yahoo.com

چکیده:

در این مقاله یک روش فرا ابتکاری جدید به نام BOB که مخفف (backwarding in behavior) می باشد برای اولین بار ارائه و همزمان برای حل مساله زمانبندی پروژه استفاده شده است . این الگوریتم برگرفته شده از واکنش عقب نشینی و بازگشتی تمامی موجودات زنده و بالاخص انسان در شرایط سخت می باشد. این الگوریتم با استفاده از سه روزرسانی که در ادامه بیان می شود لیستهای اولویت را برای پیدا کردن یک جواب بهینه بروز رسانی می کنند.

واژگان کلیدی: الگوریتم فرا ابتکاری، زمانبندی پروژه، زمانبندی پروژه با منابع محدود، لیست اولویت شدنی.

قانون اولویت

۱- مقدمه

امروزه با جهانی شدن تجارت، تلاش بیش از پیش بنگاه‌ها در جهت استفاده بهینه از امکانات و منابع برای بقا در صحنه جهانی، امری واجب شده است. در این مسیر بنگاه‌های اقتصادی چاره‌ای جز بالا بردن بهره‌وری و انجام کارهای بیشتر و بهتر با صرف منابع و زمان کمتر ندارند. از همین جاست که مفاهیمی همچون پروژه، کنترل پروژه، زمانبندی پروژه و ... مطرح شده‌اند.

با وجود آنکه امروزه زمان بندی یکی از ارکان روند انجام پروژه‌ها است و تقریباً در تمامی پروژه‌ها می‌توان اثری از آن دید، اما با اندکی تأمل در سیر تحول این روند از گذشته تا کنون می‌توان دریافت که وجود برنامه زمانبندی همواره به عنوان یک اصل در اجرای پروژه‌ها مطرح نبوده است، بلکه با پیشرفت علم و کسب تجربه در زمینه چگونگی انجام فعالیت‌های پروژه، مجریان پروژه کم‌کم با این واقعیت آشنا شدند که برای افزایش کیفیت انجام پروژه‌ها، زمان بندی فعالیت‌ها به صورتی که قبل از

شروع پروژه زمان انجام تمامی آنها مشخص باشد، امری لازم و ضروری است. با توجه به سیر تحول طی شده برای آنکه برنامه زمان بندی به عنوان یک اصل در روند انجام پروژه‌ها درآید و با اهتمام به این نکته که امروزه با رشد روز افزون علوم مدیریت، دیگر موضوع امکان پذیر بودن پروژه‌ها جای خود را به بهینه بودن آنها داده است، بدیهی است در آینده نه چندان دور موضوع بهینه سازی برنامه زمان بندی نیز کم‌کم به عنوان یک مرحله حیاتی در روند انجام پروژه‌ها درآید.

با توجه به همین مفاهیم، به دنبال طرح و زمانبندی برای انجام یک پروژه خواهیم بود که مسلماً تاثیر به سزایی در موفقیت پروژه و رسیدن به اهداف آن بازی خواهد کرد. این زمانبندی از طرفی باید با توجه به محدودیت‌های منابع باشد و از طرف دیگر باید معیارهایی را نیز مدنظر داشته باشد. مثلاً ممکن است به دلایل مختلف به دنبال حداقل کردن مدت زمان انجام پروژه باشیم و یا به دلایل اقتصادی به دنبال بیشینه کردن ارزش خالص فعلی پروژه باشیم. از همین رو و نیز به علت اهمیت و تاثیر

برنامه یا زمانبندی فعالیت های پروژه، در مبحث مدیریت زمان پروژه، مجموعه فعالیت هایی تحت عنوان تکوین و کنترل زمانبندی ایجاد شده است.

پروژه یک فعالیت بدون تکرار است که با توجه به چند هدف کاملاً از پیش تعریف شده و با در نظر گرفتن منابعی همچون پول، پرسنل و تجهیزات مورد تعهد برای انجام قرار می گیرد. معمولاً با درخواست سازمان یا فرد، یک پروژه تعریف می شود. چرخه عمر یک پروژه را میتوان به پنج فاز مجزا تقسیم کرد. در هر یک از این فازها، وظایف تعریف شده ای برای مدیران پروژه تعریف می شود. شروع با یک طرح پیشنهادی می باشد و معمولاً در این فاز توجیه پروژه صورت می گیرد.

خروجی این فاز قاعدتاً تأثیر یا رد انجام پروژه است. در فاز تعریف پروژه، اهداف پروژه مشخص می شوند، ساختار سازمانی پروژه مشخص می شود، منابع به پروژه تخصیص داده می شوند و فعالیتهای متنوع تعیین می شوند. در فاز برنامه ریزی پروژه ابتدا با تحلیل ساختاری پروژه روابط تقدم-تاخر ما بین فعالیتهای مشخص می شود. سپس تخمین های مصرف منابع و زمان فعالیتهای صورت می گیرد. پس از آن، زمانبندی موقت فعالیت ها، زودترین و دیرترین زمان ممکن برای آغاز فعالیت ها را تعیین می کند. برنامه ریزی پروژه عبارت است از تعیین توالی زمانی یا برنامه زمانبندی جهت انجام یکسری فعالیت های وابسته که تشکیل دهنده

پروژه هستند. در هر پروژه تعدادی فعالیت وابسته به هم وجود دارد که بین آنها روابط تقدم تاخر برقرار است، یعنی ممکن است انجام یک فعالیت به انجام چند فعالیت دیگر وابسته باشد که در این صورت می گوئیم پروژه دارای محدودیتهای تقدمی است. تعیین این برنامه زمانبندی با در نظر گرفتن هدف یا اهداف خاصی صورت می گیرد. معمولاً در تمام پروژه ها محدودیت های تقدمی بین فعالیت ها وجود دارند اما علاوه بر این محدودیت ها، ممکن است محدودیهایی تحت عنوان محدودیت های منابع نیز در پروژه وجود داشته باشند. هر فعالیت پروژه می تواند در چندین حالت مختلف اجرا شود که اجرای هر حالت به منبع معینی نیاز دارد. بنابراین در برنامه ریزی پروژه علاوه بر اینکه باید به محدودیت های تقدمی توجه داشت، برنامه ریزی باید به گونه ای انجام گیرد که با محدودیت منابع نیز سازگار باشد. در چنین مسائلی برنامه ریزی اجرای فعالیت ها عمدتاً به صورت پیوسته در نظر گرفته می شود. آن دسته از مسائل برنامه ریزی پروژه که محدودیت های منابع در آنها وجود ندارد یا در نظر گرفته نمی شود به مسائل برنامه ریزی پروژه بدون محدودیت منابع و آن دسته که دارای محدودیت منابع می باشند و این محدودیت ها در برنامه ریزی پروژه در نظر گرفته می شوند به مسائل برنامه ریزی پروژه با محدودیت منابع معروفند. در هنگام تهیه ی این زمانبندی موقت، محدودیت منابع در نظر گرفته نمی شود. مرحله

نهایی تخصیص منابع به فعالیت ها در طی
مراحل انجام پروژه است.

طی فاز اجرای پروژه، با نظارت بر انجام فعالیت
ها، اجرای صحیح و به موقع فعالیت ها کنترل
می شود. در این فاز پیشرفت پروژه با برنامه
زمانبندی تهیه شده مقایسه می شود. در مواردی
که انحراف قابل توجهی از برنامه اتفاق افتاده
است تخصیص منابع باید مجددا صورت پذیرد.
فاز نهایی، فاز ختم پروژه است. در این فاز
پروژه ارزیابی و مستند سازی می شود، تا
مدیریت بتواند در پروژه بعدی از آنها استفاده
کند. هر یک از فازهای پروژه نیازمند تکنیک
های ویژه ای هستند. در این رساله در مورد
روشهای کمی برای فاز برنامه ریزی پروژه بحث
می شود. و به طور ویژه به مسائل زمانبندی
پروژه در شرایط وجود محدودیت منابع پرداخته
می شود.

به علت کاربردهای عملی فراوان و همچنین
پیچیدگی های خاص، این مسئله بسیار مورد
توجه محققین بوده است و در سالهای اخیر
تحقیقات بسیاری بر روی آن صورت گرفته
است. در این تحقیقات معیارهایی برای تشخیص
و تعیین مطلوبیت یک زمانبندی برای پروژه های
تحت بررسی بکار گرفته شده است. در این
پایان نامه یک معیار شناخته شده که انحراف
میانگین از جواب بهینه یا مسیر بحرانی پروژه
است، ملاک تعیین مطلوبیت زمانبندی های
پروژه در نظر گرفته شده است. به علت
پیچیدگی بالا، به دست آوردن جواب بهینه این

مسئله با استفاده از روشهای سنتی بهینه سازی،
بسیار دشوار و یا حتی غیر ممکن است. برای
زمانبندی پروژه ها بدون در نظر گرفتن
محدودیت منابع از روشهایی مانند PERT
استفاده می شود.

با توجه به اینکه در تمام سطوح پروژه با
محدودیت منابع مواجه هستیم، مسئله ای با نام
زمانبندی پروژه با منابع محدود مطرح شده است
که به علت اینکه در روش فوق الذکر محدودیت
منابع در نظر گرفته نمی شود، نمی توان برای
حل این مسائل از آن استفاده کرد. لذا، لزوم
ایجاد و بکارگیری روشهایی که انواع محدودیت
های منابع را در نظر بگیرند، مشخص است.

برای حل اینگونه مسائل بایستی از روشهای بهینه
مبتنی بر الگوریتمهای فرا ابتکاری استفاده
کرد. روشهای بسیاری که تا کنون نیز برای حل
اینگونه مسائل بکار گرفته شده اند، روشهایی
هستند که از قانون طبیعت تبعیت می کنند و بر
اساس رفتار موجودات زنده الگو برداری شده
اند. در این پایان نامه از روشی کاملاً جدید برای
اولین بار در این مقاله معرفی شده است برای حل
مسئله زمانبندی پروژه های با منابع محدود
استفاده شده است. این الگوریتم فراابتکاری
جدید بنام الگوریتم عقب نشینی رفتاری (BOB)
نامیده شده است.

اساس کار الگوریتم مورد استفاده عکس العمل
تعدادی از جانداران در مواجهه با خطر می باشد.
الگوریتم معرفی شده با بکارگیری عملگرهایی که
برای مسئله زمانبندی پروژه تحت محدودیت

منابع طراحی شده اند، در چارچوب کلی الگوریتم، به حل مسئله می پردازند. به منظور سنجش اعتبار الگوریتم ها و نیز مقایسه آن با الگوریتم های دیگری که در ادبیات موضوع وجود دارند، الگوریتم پیشنهادی برای حل مسائل نمونه مختلفی بکار گرفته شده است و نتایج به دست آمده با بهترین الگوریتم های موجود در ادبیات موضوع مقایسه شده است. نتایج این آزمایشات حاکی از آنست که الگوریتم مورد استفاده از کارایی مطلوبی در مقایسه با الگوریتم های دیگر برخوردار است و بهتر از اکثر آنها نتیجه می دهد.

۲- روشهای فراابتکاری:

امروزه برای حل مسائل پیچیده و NP-Hard از روشهای الگو گرفته از رفتار موجودات زنده استفاده می شود. الگوریتمهایی که از این روشها برای حل مسائل استفاده می کنند. با استفاده از یک روند تکراری داده های مساله را به نحوی بروز رسانی می کنند که همواره جواب بدست آمده قبلی را بهبود دهد.

امروزه روشهای فراابتکاری بسیاری که از رفتار جمعی موجودات زنده تقلید می کنند برای حل مسائل گوناگون ارائه شده اند. نمونه هایی از آنها عبارتند از الگوریتم کلونی زنبورها [2]، الگوریتم بهینه زنبورهای مصنوعی [7.3.2]، الگوریتم گروهی زنبورها [2]، الگوریتم کلونی مورچگان [6]، الگوریتم خفاش [13,12.11.4]،

الگوریتم پرندگان [14]، الگوریتم جستجوی ممنوع [1] و الگوریتم ژنتیک [8].

مساله ای که در این تحقیق می خواهیم با استفاده از الگوریتمهای فرا ابتکاری حل کنیم، مساله زمانبندی پروژه با منابع محدود می باشد. روشی که برای این مساله استفاده می شود کاملاً جدید بوده و توسط اینجانب ابداع شده است. روش مذکور به نام الگوریتم سگهای جستجو گر (Finder Dog) نامیده شده است.

۳- زمانبندی پروژه

برای ارائه ی یک تعریف کلی از زمانبندی پروژه چنین می توانیم بیان کنیم، پروسه زمانبندی پروژه تصمیم گیری است که در مورد زمان انجام فعالیتها، بر اساس یک سری قوانین و مقررات خاص و منظم در کارخانه ها و مراکز خدمات رسانی، اتخاذ می شود. در این مسائل بهینه سازی معمولاً مدت زمان اتمام پروژه کمینه می شود. این تصمیم گیریها در مراکز خدمات رسانی، تولید، حمل و نقل، توزیع، ارتباطات، مخابرات و غیره نقش مهمی را بازی می کنند.

نرم افزارهای زمانبندی در یک کمپانی که در تصمیم گیری هایش از تکنیکهای ریاضی و الگوریتم های فرامکاشفه ای^۱ استفاده می کند، کاربرد فراوانی دارند. بطور مثال در پروسه تخصیص منابع^۲ هر کمپانی برای بهینه سازی اهدافش، از روشی خاص استفاده می کند. در این نوع مسائل منابع دارای شکلهای متنوعی از قبیل، ماشینها در یک کارگاه، باندهای پرواز در

¹ -Meta heuristic

² -Resource allocation

یک فرودگاه و واحدهای پروسه در یک محیط محاسباتی می باشند. در مقابل فعالیت ها نیز اشکال متفاوتی مانند اپراتورهای یک کارگاه، بلند شدن و فرود هواپیماها در یک فرودگاه و برنامه های کامپیوتری در یک واحد محاسباتی دارند. در این دسته مسائل، هر فعالیت یک سری پیش نیازها و ملزومات دارد. بطور مثال هر فعالیت، می تواند محدودیتی به شکل زودترین زمان شروع یا دیرترین زمان اتمام داشته باشد، بدین معنی که لازم است این فعالیت حتما در یک بازه زمانی مشخص انجام پذیرد. در مسائل زمانبندی در نظر گرفتن تمام محدودیتهایی که برای اجرای فعالیت ها وجود دارند الزامی است. از این رو مدل کردن یک مسئله واقعی به طور صحیح از اهمیت فراوانی برخوردار است. نکته دیگری که در مسائل زمانبندی پروژه اهمیت پیدا می کند تعریف تابع هدف مسئله می باشد که با توجه به پروژه دارای اشکال متفاوتی است، مانند کمینه کردن کل زمان پروژه^۳، کمینه کردن تعداد فعالیت هایی که بعد از زمان تعیین شده به پایان می رسند در ادامه مطلب، به بررسی دقیق تر مسائل زمانبندی پروژه ها در حالیکه ظرفیت منابع محدود می باشد می پردازیم. این مسائل ازگسترده ترین مسائل در زمینه زمانبندی می باشند. هدف در چنین مسائلی کمینه کردن زمان اجرای کامل پروژه می باشد.

۳-۱ زمان بندی پروژه با منابع محدود

مسئله زمانبندی پروژه با منابع محدود، یک مسئله ی عمومی زمانبندی می باشد که در مدل کردن تعداد زیادی از مسائل کاربردی در جامعه کاربرد دارد. اجرای هر فعالیت در این پروژه نیاز به منابع متفاوتی دارد. ضمن اینکه هر یک از این منابع دارای ظرفیت مشخص هستند.

اساسا ما بین منابعی که بطور انحصاری در دسترس فعالیتهای در حال اجرا می باشند و پس از اتمام اجرای این فعالیتهای همچنان قابل استفاده هستند (مانند نیروی انسانی و ماشینی) و منابعی که پس از استفاده شدن در طی پروژه دیگر قابل استفاده نمی باشند (مانند بودجه و فضای یک انبار) تفاوت قائل هستیم. ما در این پایان نامه به مسائل می پردازیم که منابع آنها از نوع اول می باشند که به اصطلاح (منابع تجدیدشدنی)^۴ خوانده می شوند) منابعی که پس از استفاده، مجددا قابل استفاده در اجرای دیگر فعالیت ها می باشند. در زمانبندی فعالیت ها باید دقت داشته باشیم تا مجموع منابع مورد نیاز فعالیت های در حال اجرا در هر زمان مشخص، از میزان ظرفیت آن منبع مشخص تجاوز نکند. در سال ۱۹۸۳ بلازویچ^۵ NP-hard بودن مسائل زمانبندی پروژه را ثابت کرد.

در اینجا مدل RCPSP را تعریف می کنیم. مجموعه فعالیتهای پروژه را با $N=\{0.1.2.3.....n\}$ نشان می دهیم. در اینجا فعالیت های ۰ و n فعالیت های زائد پروژه نامیده می شوند. مدت زمان اجرای فعالیت iام در این پروژه را با d_i

⁴ Removable Resource

⁵ Markesan

³ Makespan

نشان می دهیم زوج مرتب (i, j) نشان دهنده ی این خاصیت می باشد که فعالیت i ام پیش نیاز فعالیت j ام است، بدین معنی که لازم است قبل از اجرای فعالیت j ام حتما فعالیت i ام اجرا شده باشد.

مجموعه همه این زوج مرتب ها را با A نمایش می دهیم. فرض کنید که R مجموعه ی همه منابع تجدید شدنی باشد که دارای K منبع متفاوت است. در این مسئله منبع i ام دارای ظرفیت $R(i) \in R \geq 0 \cup \{\infty\}$ واحد می باشد که در طول اجرای پروژه در هر واحد زمانی می تواند مورد استفاده قرار گیرد $R(i) = \infty$ به این معنی می باشد که منبع هیچ محدودیت ظرفیتی ندارد و به هر اندازه که لازم باشد می توان از این منبع در طول پروژه در هر زمانی استفاده کرد.

اما ممکن است که برای استفاده از این منبع به ازاء هر واحد از آن در هر زمان معین هزینه مشخصی در نظر گرفته شود، لذا در این صورت کمینه کردن استفاده از این منبع خاص نیز باید مد نظر قرار گیرد. هر فعالیت به میزان مشخصی از منابع مجموعه ی R نیاز دارد. میزان مورد نیاز فعالیت i ام به منبع K ام را $r_{ik} \in R \geq 0$ واحد در نظر می گیریم r_{ik} یعنی این که از زمان شروع فعالیت i ام تا زمان اجرای کامل این فعالیت به میزان r_{ik} واحد از ظرفیت منبع K ام درگیر انجام فعالیت i ام می باشد که برای اجرای دیگر فعالیتها در این بازه زمانی قابل استفاده نیست. اگر در زمانبندی پروژه زمان شروع فعالیت i ام S_i در نظر گرفته

شود آنگاه فعالیت i ام در بازه $[S_i, S_i + d_i]$ در حال اجرا می باشد و به میزان $(R_k - r_{ik})$ واحد از منبع K ام برای اجرای دیگر فعالیتها در این بازه باقی می ماند. $r_{ik}=0$ نشان میدهد که فعالیت i ام برای اجرا به منبع K ام هیچ نیازی ندارد. باتوجه به توضیحات بالا برای همه فعالیتها و منابع این پروژه نامساوی زیر برقرار است.

$$r_{ik} \leq R_k \quad k = 1, 2, \dots, K \quad i = 0, 1, \dots, n \quad (1)$$

همانطور که در ابتدای مبحث بیان شد، این مسئله دارای $n+1$ فعالیت می باشد که $n-1$ عدد از آنها فعالیت های اصلی و دو فعالیت دیگر زائد می باشند. فعالیت های زائد عبارتند از فعالیت صفر و n که به ترتیب فعالیت شروع و فعالیت پایان پروژه می باشند. هر دو فعالیت دارای مدت زمان صفر هستند یعنی $d_0=d_n=0$ این دو فعالیت هیچ نیازی به هیچ یک از منابع مجموعه ی R ندارند یعنی برای هر منبع پروژه داریم $r_{0k}=r_{nk}=0$ این دو فعالیت تنها برای روشنتر شدن بیشتر مسئله تعریف می شوند. همانطور که از تعریف این دو فعالیت مشخص است، برای فعالیت اول هیچ پیش نیازی وجود ندارد یعنی این که نیازی نیست قبل از این فعالیت هیچ فعالیت دیگری از مسئله اجرا گردد. ضمن اینکه در مقابل برای فعالیت پایانی لازم است که همه فعالیت های $0, 1, 2, \dots, n-1$ انجام پذیرند و پس از اتمام این فعالیت ها فعالیت n ام شروع به اجرا شود.

فرض کنید $S = (S_0, S_1, \dots, S_n)$ یک زمانبندی شدنی برای مسئله مورد نظر باشد و t یک نقطه

در زمان بندی مشخص S ، $c_i = s_i + d_i$ را زمان
اتمام فعالیت نام می نامیم. طبق این زمانبندی
فعالیت i ام در بازه زمانی $[s_i, c_i]$ اجرا می شود.
در نهایت تابع هدف مسئله، کمینه کردن S_n
می باشد. با تعریف تابع f صورت کلی یک
مسئله ی زمان بندی پروژه بامنابع محدود به
شکل زیر می باشد.

$$\text{Minimize } f(s)$$

Subject to

$$S \in S_T \cap S_R$$

(۷)

با توجه به تعریف بالا، مسئله زمانبندی پروژه با
منابع محدود، بصورت خطی زیر قابل مدل کردن
است.

$$\text{Minimize } S_n$$

Subject to

$$S_i \leq S_j - d_i \quad \forall (i, j) \in A \quad (۸)$$

$$\sum_{i \in p(s,t)} r_{ik} \leq R_k \quad k \in R, t \in [0, s_n]$$

(۹)

۴- الگوریتم پیشنهادی

۴-۱ الگوریتم عقب نشینی رفتاری (backwarding
(in behavior)

این الگوریتم برای اولین بار است که در زمینه
حل مسائل بهینه سازی و توسط اینجانب استفاده
شده است. این الگوریتم از عکس العمل
موجودات زنده در هنگام مواجهه با خطر الگو
برداری شده است.

۴-۲ مقدمه

زمانی خاص و مشخص، مجموعه ی $p(s,t)$ را به
صورت:

$$P(s, t) = \{i \in N \mid S_i \leq t < S_i + d_i\} \quad (۲)$$

ت

عریف می کنیم n مجموعه همه فعالیتهای پروژه
می باشد، مجموعه ی $p(s,t)$ مجموعه همه
فعالیتهای در حال اجرای مسئله در زمان t بر
اساس زمانبند S می باشند.

با توجه به تعریف بالا تابع $r_k(s,.)$ را بدین
صورت تعریف می کنیم

$$R_k(s,t): [0, s_n] \rightarrow \mathbb{R}_{\geq 0} \quad (۳)$$

$$R_k(s,t) = \sum_{i \in p(s,t)} r_{ik} \quad k=1,2,\dots,k \quad (۴)$$

$r_k(s,.)$ میزان استفاده از منبع k ام در زمان t را
مشخص می کند. حال با توجه به تعریف مسئله
و محدود بودن منابع برقراری نامساوی

$$R_k(s,t) \leq R_{\eta} \quad k=1,2,\dots,K \quad 0 \leq t \leq s_n \quad (۵)$$

بدیهی است S_n در حقیقت زمان شروع فعالیت
زائد پایانی می باشد که مشخص کننده زمان
اجرای کامل پروژه است.

اگر تمام زمانبندیهایی که در آن محدودیت منابع
رعایت شده باشند را با SR نمایش دهیم، و در
مقابل تمام زمانبندیهایی که در آن محدودیتهای
زمانی و پیش نیازهای مسئله رعایت شده باشند
را با ST نمایش دهیم، آنگاه می توانیم S را به
صورت زیر تعریف کرده و آن را مجموعه همه
زمانبندیهای شدنی مسئله بنامیم.

$$S = S_R \cap S_T \quad (۶)$$

موجودات زنده مختلف اعم از انسان در یک زندگی دسته جمعی با هموعان خودش همواره از بهترینهای خودشان تقلید کرده و رفتارشان را به رفتار نامبرده نزدیک می کنند. در حالت عادی کلیه موجودات زنده برای رسیدن به هر هدفی، از کل شرایط موجود، بهترین را انتخاب می کنند؛ مثلاً زمانی که گرسنه هستند بدنبال مکانی می گردند که بیشترین منبع غذایی در آنجا وجود دارد. برای ساختن لانه، مکانی را انتخاب می کنند که از همه جهت مناسب باشد مثلاً دسترسی شکارچیان به آنجا محدود باشد، هنگام بارندگی، آب به داخل لانه نفوذ نکند و همچنین نزدیکترین مکان به منابع غذایی باشد. که برای انجام این کارها طبق رفتار گروهی، همواره به سمت هموعی از خودش گرایش پیدا می کند که بیشترین منبع غذایی را و همچنین بهترین مکان برای اسکان را پیدا کرده است و تا جایی این کار را انجام می دهد تا به هدف برسد. پس در اینجا هدف نزدیک شدن به بهترین شرایط از بین شرایط موجود است. ولی در بعضی از مواقع همین موجودات در وضعیتی قرار می گیرند که مجبورند بدترین شرایط را از بین شرایط موجود انتخاب کرده و از آن دوری کنند. یکی از این وضعیتهای، وضعیتی است که موجود احساس خطر می کند.

۳-۴ بررسی رفتار شطرنج باز موقع عقب نشینی و فرار از مات شدن

بازی شطرنج یک بازی فکری است که هر دو طرف، تمام سعی خود را برای مات کردن طرف

مقابل خود انجام می دهند. برای پیروزی در بازی شطرنج، شطرنج باز با سنجیدن شرایط تک تک مهره های خودش و همچنین بررسی شرایط مهره های حریف بهترین حرکت را انجام می دهد. ولی گاهی ممکن است مهره های یک شطرنج باز در موقعیت خطرناکی و حتی در خطر مات شدن قرار بگیرد، اینجا است که شطرنج باز بایستی با یک تکنیک عقب نشینی، مهره های خودش را (مخصوصاً مهره شاه) را از تهدید مهره های مقابل دور کند.

در این موقعیت شطرنج باز همواره مهره ای از مهره های خودش را جابجا می کند که در موقعیت بدتری نسبت به دیگر مهره ها قرار داشته باشد و خطر، بیشتر از همه آن مهره را تهدید می کند.

۴-۴ شرح الگوریتم:

در الگوریتم مذکور که به اختصار الگوریتم BOB (Backwarding of behavior) نامیده شده از تکنیک عقب نشینی برای فرار از مات شدن در بازی شطرنج الگو گرفته است که به شرح زیر می باشد:

۱- ابتدا به مهره ها در همان موقعیتی که قرار دارند یک سری مقادیر تصادفی بین [0,1] طبق رابطه ۱۰ به آنها اختصاص داده می شود.

$$x_{ij} = rand \quad (10)$$

۲- در یک روند تکراری عملیات زیر انجام می شود:

الف) موقعیت تک تک مهره ها بررسی شده و مهره ای که در موقعیت بدتری نسبت به بقیه

مهره ها قرار دارد شناسایی می شود. این مهره را با x_w مشخص می کنیم. سپس شطرنج باز سعی می کند که از بین دیگر مهره هایش مهره ای را که هم موقعیت بهتری نسبت به مهره x_w دارد و هم این که مهره مورد نظر بتواند مهره x_w را پوشش دهد، شناسایی کرده و مهره x_w را به سمت آن مهره و به نزدیکی آن حرکت دهد.

برای شناسایی بهترین مهره که بتواند مهره x_w را پوشش دهد. شطرنج باز بایستی طبق رابطه ۱۱ موقعیت تمامی مهره ها را جهت نزدیک کردن x_w به آن بسنجد. مکان های بدست آمده مکانهای موقت نامیده می شوند و دلیل آن این است که مهره x_w به صورت فیزیکی به این مکانها منتقل نشده اند و فقط شطرنج باز در ذهنش مهره ها را به این مکانها می برد و میزان امن بودن x_w را در کنار این مهره ها محاسبه می کند.

$$temp_{ij} = x_w + (x_{ij} - x_w) \times r \quad (11)$$

در رابطه ۱۱ r یک عدد تصادفی در بازه $[0,1]$ می باشد.

سپس تک تک موقعیتهای موقت بدست آمده $temp_{ij}$ با موقعیت دیگر مهره ها مقایسه می شود، اگر موقعیت بدست آمده x_w در همسایگی دیگر مهره ها (x_{ij}) که همان $temp_{ij}$ می باشد از موقعیت کنونی خود مهره ها x_{ij} بهتر باشد، آنگاه مکان مهره $temp_{ij}$ به عنوان مهره x_{ij} در نظر گرفته می شود و مهره x_{ij} نادیده گرفته می شود. و در غیر اینصورت این مکان که در ذهن شطرنج باز برای پوشش x_w در نظر گرفته شده

از بین گزینه های انتخابی شطرنج باز حذف می شود و شطرنج باز مکان مهره دیگری را برای پوشش دادن x_w بررسی می کند و این روند به تعداد مهره های موجود در صفحه شطرنج انجام می شود. تک تک موقعیتهای جدید بدست آمده برای بدترین مهره (x_w) در همسایگی مهره ها (x_{ij}) که همان $(temp_{ij})$ می باشد با موقعیت بهترین مهره تا این مرحله از بازی (x_*) مقایسه شده، در صورتیکه موقعیت آنها بهتر از موقعیت (x_*) باشد جایگزین (x_*) می شوند.

علاوه بر آن موقعیت تک تک مهره ها نیز با موقعیت بدترین مهره (x_w) مقایسه می شود در صورتیکه بدتر از x_w باشند جایگزین آن می شوند.

ب) در این مرحله از بروز رسانی مکان مهره های شطرنج، شطرنج باز علاوه بر نزدیک کردن مهره x_w به مهره های دیگر برای محافظت از مهره x_w ، طبق رابطه ۱۲ مهره های دیگر را نیز که پشت سر مهره x_w قرار دارند را به سمت بهترین وضعیت x_* حرکت داده تا هم وضعیت مهره های دیگر را بهبود داده و هم اینکه مکان را برای منتقل شدن مهره x_w به یک مکان امن خالی کند

$$x_{i,j}^t = x_{i,j}^{t-1} + (x_* - x_{i,j}^{t-1}) \times \beta \quad (12)$$

در رابطه ۱۲ r یک عدد تصادفی در بازه $[0,1]$ می باشد.

تک تک موقعیتهای جدید بدست آمده برای بدترین مهره (x_w) در همسایگی مهره ها (x_{ij}) که همان $(temp_{ij})$ می باشد با موقعیت بهترین

مهره تا این مرحله از بازی (x_*) مقایسه شده ،
در صورتیکه موقعیت آنها بهتر از موقعیت (x_*)
باشد جایگزین (x_*) می شوند.

علاوه بر آن موقعیت تک تک مهره ها نیز با
موقعیت بدترین مهره (x_w) مقایسه می شود در
صورتیکه بدتر از x_w باشند جایگزین آن می
شوند.

ج) در این مرحله از بروزرسانی، پس از بروز
رسانی مکان مهره ها بر اساس موقعیت بهترین
مهره ،(طبق رابطه ۱۲) مهره ای که در بدترین
وضعیت قرار داشت و طبق رابطه ۱۱ در موقعیت
موقت (طبق تصور ذهن شطرنج باز) قرار گرفت
مکان خودش را طبق رابطه ۱۳ به مکان دیگر
مهره ها که الان مکانشان به مکان بهترین مهره
نزدیک شده است نزدیک می کند.

$$x_{i,j}^t = temp_{ij} + (x_{i,j}^{t-1} - temp_{ij}) \times \beta \quad (13)$$

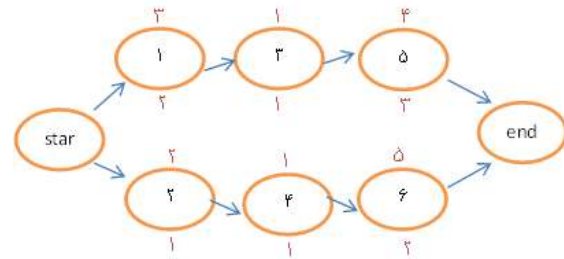
که $x_{i,j}^{t-1}$ موقعیت قبلی عضو مورد نظر و $x_{i,j}^t$
موقعیت جدید عضو مورد نظر و β یک عدد
تصادفی بین $[0,1]$ و $temp_{ij}$ موقعیت موقتی
بدترین مهره (x_w) در همسایگی مهره های دیگر
می باشد.

۵- حل مسأله زمانبندی پروژه با منابع محدود به وسیله الگوریتم عقب نشینی رفتاری (BOB)

به منظور استفاده از الگوریتم BOB باید طرح
مناسبی پیدا کنیم تا اعضاء را در مسأله RCPSB
پیاده سازی کنیم .اولویت بندی اولیه در الگوریتم
BOB، از جمعیتی از ذرات به طور تصادفی و
طبق رابطه ۱۰ شروع می شود و طبق مکانیسم

مخصوص هر یک از روش های الگوریتم BOB
که در بند ۴-۴ توضیح داده شد به روزرسانی می
شود . در فضای N بعدی N بعد از مکان هر
عضو گروه نشان دهنده تعداد فعالیت های پروژه
است .بعد از اینکه ابعاد هر ذره را به ترتیبی قرار
دادیم که شدنی باشد فعالیتها به یک ترتیب شدنی
پشت سر هم قرار می گیرد حال نوبت به تشخیص
فعالیت های موازی می رسد.

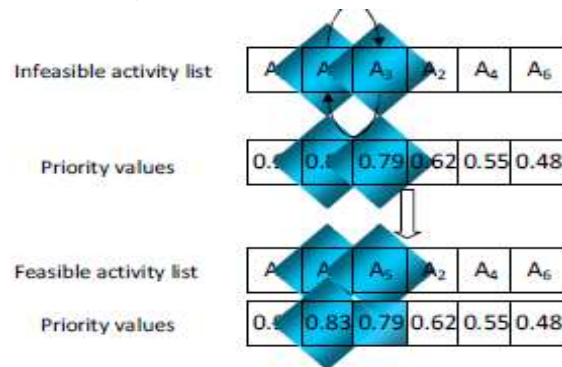
فعالیت های موازی آن دسته از فعالیت هایی است
که هر دو شرایط اولویت بندی و محدودیت منابع
را شامل شوند و بتوانند در برنامه زمانی یکسان
انجام شوند. الگوریتم BOB نیز مانند دیگر
الگوریتم های جمعیتی درگام اول به دنبال ساخت
یک سری جواب به عنوان جمعیت اولیه می باشد
.اعضای گروهی از موجوداتی که قرار است
عقب نشینی کنند در مرحله اول یک دسته جواب
شدنی تصادفی را به عنوان موقعیت اولیه ی خود
اختیار می کنند . به طور مثال یک عضو از گروه
برای ساخت یک جواب شدنی لازم است که یک
لیست اولویت ایجاد کند .در ابتدا برای بدست
آوردن لیست اولیت، به تعداد فعالیتها (N) اعداد
بین 0 و 1 را تولید می کنیم سپس اعداد را به
ترتیب نزولی مرتب می کنیم . از آنجایی که هر
عدد متناظر با یک فعالیت است پس فعالیت ها هم
به همان ترتیب اولویت بندی می شوند . بنابراین
متناظر با هر عضو دو بردار داریم، که یکی شامل
اعداد بین ۰ و ۱ و دیگری نشان دهنده شماره
فعالیت ها می باشد .برای مثال پروژه زیر را در
نظر بگیرید:



شکل ۱ گراف فعالیتها

۵-۱ تبدیل جواب های نشدنی به جواب های شدنی

برای این منظور از انتهای لیست اولویت یکی یکی فعالیتها را چک می کنیم و هر فعالیت که شرط پیش نیازی را نقض کرده باشد با اولین پیش نیازش در لیست اولویت جابجا می کنیم



شکل ۲ تبدیل جواب های نشدنی به جواب های شدنی در لیست اولویت

به تعداد اعضای گروه روند فوق برای رسیدن به یک لیست اولویت شدنی طی می شود. در الگوریتم ارائه شده بردار متناظر با مقدار اولویت بروزرسانی می شود و پس از مرتب کردن بردار بصورت نزولی، لیست اولویت جدید بدست می آید.

پس از بدست آوردن لیست اولویت برای هر عضو گروه، makespan متناظر با هر لیست اولویت را با استفاده از روش زمانبندی سری پسرو و پیشرو حساب می کنیم. بعد از مقدار

دهی اولیه وارد پروسه تکرار شونده الگوریتم می شویم. در هر تکرار، الگوریتم طبق ۴-۴ عمل نموده و لیست اولویت فعالیتهای پروژه را بر اساس شبیه سازی رفتار یک شطرنج باز در حالتی که خطر مهره هایش را تهدید می کند و سعی دارد که موقعیت مهره هایش را از مکانی که بیشترین خطر آنها را تهدید می کند به مکان امنی منتقل کند طی ۳ مرحله گفته شده در بند ۴-۴ بروزرسانی می کنند.

در مرحله اول بروز رسانی لیستهای اولویت ، بدترین لیست اولویت طبق رابطه ۱۱ عمل نموده $temp_{ij} = x_w + (x_{ij} - x_w) \times r$ و مقدار اولویت تک تک فعالیتهایش را به سمت مقدار اولویت فعالیتهای دیگر لیستهای اولویت نزدیک می کند. در واقع یک حرکت عقب گرد عقب گرد انجام شده است. سپس بوسیله تابع تخمین زمان اجرای پروژه که به صورت سری انجام میشود، makespan مربوط به هر لیست اولویت بدست می آید. در صورتی که زمان بدست آمده برای هر لیست اولویت از زمان قبلی همان لیست بهتر باشد آنگاه ترتیب فعالیتهای لیست جاری جایگزین ترتیب فعالیتهای همان لیست که در تکرارهای قبلی بدست آمده است می شود. و همچنین در صورتیکه ترتیب فعالیتهای هر لیست اولویت زمان کمتری را نسبت به زمان بدست آمده برای بهترین لیست اولویت باشد ، لیست جاری جایگزین بهترین لیست می شود.

در مرحله دوم بروزرسانی، لیستهای اولویت طبق رابطه ۱۲ و بر اساس ترتیب فعالیتها در بهترین لیست اولویت بروزرسانی را انجام می دهد.

سپس بوسیله تابع تخمین زمان اجرای پروژه که به صورت سری انجام می شود، makespan مربوط به هر لیست اولویت بدست می آید. در صورتی که زمان بدست آمده برای هر لیست اولویت از زمان قبلی همان لیست بهتر باشد آنگاه ترتیب فعالیتهای لیست جاری جایگزین ترتیب فعالیتهای همان لیست که در تکرارهای قبلی بدست آمده است می شود. و همچنین در صورتیکه ترتیب فعالیتهای هر لیست اولویت زمان کمتری را نسبت به زمان بدست آمده برای بهترین لیست اولویت باشد، لیست جاری جایگزین بهترین لیست می شود.

علاوه بر بهترین لیست اولویت بدترین لیست اولویت نیز در بین همه لیست اولویت مشخص می شود.

در مرحله سوم بروزرسانی مقدار اولویتهای بدترین لیست اولویت که به مقادیر لیستهای اولویت مجاور نزدیک شده بود ($temp_{ij}$) طبق رابطه ۱۳ مقادیر اولویت خود را به مقادیر اولویت دیگر لیستهای اولویت که آنها نیز مقادیر

اولویتشان را به مقادیر اولویت بهترین لیست اولویت نزدیک کرده بودند نزدیک می کنند.

$$x_{i,j}^t = temp_{ij} + (x_{i,j}^{t-1} - temp_{ij}) \times \beta$$

سپس طبق معمول بعد از بروز رسانی و دادن مقادیر جدید به لیستهای اولویت، زمان اتمام پروژه (makespan) برای ترتیب فعالیتهای هر لیست اولویت محاسبه شده، سپس از بین زمانهای بدست آمده آن لیست اولویتی که زمان کمتری را برای اتمام پروژه بدست آورده باشد به عنوان بهترین لیست اولویت شناخته می شود. و علاوه بر آن لیست اولویتی که بیشترین زمان را برای اتمام پروژه (ضعیفترین نتیجه) را بدست آورده به عنوان بدترین لیست اولویت معرفی می گردد.

۶- نتایج محاسباتی الگوریتم BOB

۶-۱ تاثیر اندازه جمعیت و تعداد تکرار بر

کارایی الگوریتم BOB

در اولین آزمایش ما تاثیر تعداد جمعیت و تعداد تکرار بر کارایی الگوریتم BOB را بررسی کرده ایم. اندازه جمعیت را از ۱۰ تا ۱۰۰ با طول گام های ۱۰ و تعداد تکرارها را از ۲۵۰ تا ۲۵ تغییر داده ایم، به گونه ای که تعداد زمانبندی ها در هر بار انجام آزمایش مقدار ثابت ۲۵۰۰ شود. نتایج بدست آمده را می توان در جدول ۱ مشاهده کرد.

جدول ۱ تاثیر اندازه جمعیت و تعداد تکرار بر کارایی الگوریتم BOB

Population size	40	50	60	70	80	90	100
#iterations	63	50	42	36	31	28	25
Success rate	j30	82.51%	82.35%	82.51%	82.74%	82.13%	82.63%
	j60	68.25%	68.62%	68.25%	68.62%	68.58%	68.45%
	j90	66.83%	66.21%	67.09%	66.21%	66.83%	67.00%
	j120	19.44%	19.76%	19.34%	19.44%	19.76%	19.61%

جدول ۳- نتایج مربوط به نرخ همگرایی مسائل J60

#schedules			Method
50000	5000	1000	
72.18%	69.53%	67.48%	BOB

جدول ۴- نتایج مربوط به نرخ همگرایی مسائل J90

#schedules			Method
50000	5000	1000	
70.59%	67.47%	65.38%	BOB

جدول ۵- نتایج مربوط به نرخ همگرایی مسائل J120

#schedules			Method
50000	5000	1000	
25.17%	21.29%	18.31%	BOB

۴-۶ فاصله درصدی از جواب بهینه و فاصله

درصدی از طول مسیر بحرانی :

فاصله درصدی به عنوان پارامتری برای ارزیابی

کارایی الگوریتم BOB که طبق رابطه ۱۴ بدست

می آید برای همه پروژه های J30، J90 و

J120 در جداول ۶ و ۷ و ۸ آمده است.

$$dev = \frac{1}{instances} \sum_{instances} \left(\frac{E-CPM}{CPM} \right) \times 100 \quad (14)$$

۲-۶ نرخ همگرایی به عنوان ملاکی برای ارزیابی

کارایی الگوریتم BOB

نرخ همگرایی:

برای هر آزمایش، نسبت تعداد پروژه های حل

شده به تعداد کل پروژه را مشخص کرده و

درصد آن را در جدول نتایج درج می کنیم مثلاً

اگر ۱۸۰ پروژه داشته باشیم و از بین همه آنها

۱۱۰ پروژه حل شود و نتایج بهتری نسبت به

نتیجه پیشین آنها بدست بیاید آنگاه داریم

$$\frac{110}{180} \times 100 = 61.11$$

با توجه به این فاکتور نرخ همگرایی را برای همه

پروژههای J30، J60، J90 و J120 بدست می

آوریم . نتایج بدست آمده در جداول ۲ ، ۳ ، ۴

و ۵ آمده است.

جدول ۲ نتایج مربوط به نرخ همگرایی مسائل J30

#schedules			Method
50000	5000	1000	
93.14%	86.23%	79.42%	BOB

جدول ۶ فاصله درصدی از طول مسیر بحرانی برای مسائل j60

j60	SGS	1000	5000	50000
Sampling—LFT, FBI	Both	11.88	11.62	11.36
SA—activity list	Serial	12.75	11.90	-
TS—activity list	Serial	12.97	12.18	11.58
Sampling— adaptative	Both	12.94	12.58	-
TS—schedule scheme	Related	13.80	13.48	-
GA—random key	Serial	14.68	13.32	12.25
GA—priority rule	Serial	13.30	12.74	12.26
Sampling—LFT	Parallel	13.59	13.23	12.85
Sampling—LFT	Serial	13.96	13.53	12.97
Sampling—random	Parallel	14.89	14.30	13.66
Sampling—WCS	Parallel	13.66	13.21	-
GA—problem space	Mod. Par.	14.33	13.49	-
Neurogenetic	Serial	11.51	11.29	-
ACOSS	ser./par.	11.75	10.98	10.67
GAPS	Param. active	11.72	11.04	10.67
ANGEL	Serial	11.94	11.27	-
ABC	Serial	14.57	13.12	12.53
BSO	Serial	13.67	12.70	12.45
BA	Serial	13.35	12.83	12.41
BABC	Serial	11.91	11.08	10.67
FD	Serial	13.46	13.32	13.21
BOB	Serial	13.53	13.45	13.32

جدول ۷ فاصله درصدی از جواب بهینه برای مسائل j30

j30	1000	5000	50000
SA - activity list	0.38	0.23	-
GA -activity list	0.54	0.25	0.08
Sampling - adaptive	0.65	0.44	-

TS - schedule scheme	0.86	0.44	-
Sampling -adaptive	0.74	0.52	-
Single pass/sampling - LFT	0.83	0.53	0.27
GA - random key	1.03	0.56	0.23
Sampling - random	1.44	1.00	0.51
GA -priority rule	1.38	1.12	0.88
Single pass/sampling-WCS	1.40	1.28	-
Single pass/sampling-LFT	1.40	1.29	1.13
Sampling-random	1.77	1.48	1.22
GA-problem space	2.08	1.59	-
Sampling -adaptive	0.71	0.59	-
PSO	0.92	0.61	-
PSO	0.69	0.42	-
Neurogenetic	0.13	0.10	-
ACOSS	0.14	0.06	0.01
GAPS	0.06	0.02	0.01
ANGEL	0.22	0.09	-
ABC	0.98	0.57	0.20
BSO	0.65	0.36	0.17
BA	0.63	0.33	0.16
ABC-FBI	0.47	0.28	0.09
BSO-FBI	0.45	0.22	0.07
BA-FBI	0.42	0.19	0.04
BABC	0.16	0.07	0.01
FD	0.31	0.22	0.14
BOB	0.61	0.32	0.15

جدول ۸ فاصله درصدی از طول مسیر بحرانی برای مسائل j120

j120	1000	5000	50000
GA, TS—path relinking	34.74	33.36	32.06
GA—self-adapting	37.19	35.39	33.21
Sampling—LFT, FBI	35.01	34.41	33.71
GA—activity list	39.37	36.74	34.03
SA—activity list	42.81	37.68	-

TS—activity list	40.86	37.88	35.85
GA—priority rule	39.93	38.49	36.51
Sampling—adaptative	39.85	38.70	-
Sampling—LFT	39.60	38.75	37.74
Sampling—WCS	39.65	38.77	-
Sampling—adaptative	41.37	40.45	-
GA—problem space	42.91	40.69	-
GA—random key	45.82	42.25	38.83
Sampling—LFT	42.84	41.84	40.63
Sampling—random	44.46	43.05	41.44
Sampling—random	49.25	47.61	45.60
Neurogenetic	34.65	34.15	-
ACCOS	35.19	32.48	30.56
GAPS	35.87	33.03	31.44
ANGEL	36.39	34.49	-
ABC	43.24	39.87	37.36
BSO	41.18	37.86	35.70
BA	40.38	38.12	36.12
ABC-FBI	37.85	36.82	35.02
BSO-FBI	37.84	36.51	34.86
BA-FBI	37.72	36.76	34.55
BABC	35.37	32.54	31.69
FD	37.77	36.69	34.81
BOB	39.55	37.74	35.92

دو رابطه بروزرسانی ۱۱ و ۱۲ و ۱۳ بهبود می

بخشد.

۷- نتیجه گیری

در این رساله یک روش ابداعی و جدید برای حل مسأله زمانبندی پروژه ها با منابع محدود با عنوان الگوریتم عقب نشینی رفتاری برای اولین بار ارائه شد. این روش براساس گریز افراد، جانداران و همچنین تاکتیک بکار رفته در بعضی از بازیهای ذهنی مثل بازی شطرنج برای فرار از مات شدن الگو گرفته شده است. این الگوریتم با استفاده از سه روش تصمیم گیری پویا، ترتیب اولیه فعالیتها را بروزرسانی می کند. الگوریتم بکار رفته در این مقاله رفتار واکنشی کلیه موجودات در حین مواجهه با خطر را شبیه سازی می کند. و همواره ترتیب فعالیت در لیستهای اولویت را با توجه به

مراجع:

- [1] حداد، مسعود، و حداد، حمید رضا (۱۳۸۸) توسعه الگوریتم جستجوی ممنوع (tabu search) برای حل مساله زمانبندی پروژه های دارای محدودیت منابع با در نظر گرفتن حالت پیش نیازی عمومی (GRCPSP).
- [2] ضیغمی، وحید (1389) مسائل زمان بندی پروژه با منابع محدود. پایان نامه کارشناسی ارشد، بخش ریاضی، دانشکده علوم، دانشگاه شیراز، شیراز، ایران.

engineering optimization. Eng Comput
29(5):464–483

[14] Zhang, C., Sun, J., Zhu, X., & Yang, Q. An improved particle swarm optimization algorithm for flowshop scheduling problem. Information Processing Letters, 108(4), 204–209, (2008)

[3] Akbari R., Zeighami V., and Ziarati K., "Artificial Bee colony for resource constrained project scheduling problem", International Journal of Industrial Engineering omputations, 2(1), pp. 45-60,(2011).

[4] Altringham, J. D.: Bats: Biology and Behaviour, Oxford Univesity Press, (1996).

[5] Dorndorf U, Pesch E, Phan-Huy T. A branch-and-bound algorithm for the resource-constrained project scheduling problem.Mathematical Methods of Operations Research;52:413–39,(2000).

[6] D. Merkle, M. Middendorf, H. Schmeck, Ant colony optimization for resource-constrained project scheduling, IEEE Transactions on Evolutionary Computation 6 333–346, (2002).

[7] Karaboga D, Basturk B. On the Performance of Artificial Bee Colony.Journal of Applied Soft Computing, (2008).

[8] S. Hartmann, A competitive genetic algorithm for resource-constrained project scheduling, Naval Research Logistics 45 (1998) 733–750.

[9] S. Hartmann, A self-adapting genetic algorithm for project scheduling under resource constraints, Naval Research Logistics 49 ,433–448, (2002).

[10] Thomas, P., R., & Salhi S. A tabu search approach for the resource constrained project scheduling problem. *Journal of Heuristics*, 4, 123–139, (1998).

[11] Richardson, P.: Bats. Natural History Museum, London, (2008).

[12] Richardson, P.: The secrete life of bats. <http://www.nhm.ac.uk>.

[13] Yang XS, Gandomi AH (2012) Bat algorithm: a novel approach for global