



## حل مسئله زمانبندی پروژه با هدف کمینه سازی زمان اتمام پروژه با محدودیت منابع با الگوریتم فراابتکاری قورباغه

علیرضا حاجی آخوندی (نویسنده مسؤل)

دانشجوی دکتری، دانشکده مهندسی صنایع، دانشگاه صنعتی مالک اشتر، تهران، ایران

Email: vahid.kameli.2015@gmail.com

غلامرضا توکلی

استادیار دانشگاه صنعتی مالک اشتر، تهران، ایران

پیمان اخوان

دانشیار دانشگاه صنعتی مالک اشتر، تهران، ایران

منوچهر منطقی

دانشیار دانشگاه صنعتی مالک اشتر، تهران، ایران

تاریخ دریافت: ۹۶/۲/۱۱ \* تاریخ پذیرش: ۹۶/۷/۲۵

### چکیده

الگوریتم جهش ترکیبی قورباغه (SFLA) یک الگوریتم مبتنی بر ممیتیک متاهیوریستیک است. این الگوریتم در سال‌های اخیر توسط Eusuff و Lansey ایجاد شد. الگوریتم SFLA از نحوه‌ی جستجوی غذای گروه‌های قورباغه سرچشمه می‌گیرد. این الگوریتم برای جستجوی محلی میان زیرگروه‌های قورباغه از روش نمو ممیتیک استفاده می‌کند. SFLA از استراتژی ترکیب استفاده می‌کند و امکان مبادله پیام در جستجوی محلی را فراهم می‌سازد. الگوریتم جهش ترکیبی قورباغه مزایای الگوریتم نمو ممیتیک و بهینه‌سازی گروه ذرات (PSO) را ترکیب می‌کند. یکی از مسائل مشهور در زمینه کنترل پروژه، زمانبندی پروژه با محدودیت منابع و سایر محدودیتها می‌باشد که زمانبندی پروژه با در نظر گرفتن محدودیت منابع از جمله مسائل دارای پیشینه تحقیقاتی غنی است. مساله زمانبندی پروژه با منابع محدود در واقع کلی ترین مساله زمانبندی است. مسائل زمانبندی کارگاهی، جریان کارگاهی، زمانبندی و سایر مسائل زمانبندی همگی زیر مجموعه ای از این مسئله به حساب می‌آیند. زمانبندی پروژه یکی از وظایف اصلی و فعالیت‌های اصلی در مدیریت پروژه است. وجود محدودیت منابع و همچنین روابط پیش‌نیازی بین فعالیت‌ها مسئله زمانبندی پروژه را امری دشوار می‌سازد. زمانبندی پروژه با در نظر گرفتن محدودیت منابع از جمله مسائل با ادبیات غنی در حوزه مسائل تحقیق در عملیات است. این مسئله توجه محققان را در سالهای اخیر بشدت بخود جلب کرده است و تاکنون با الگوریتم‌های مختلف حل شده است. در این مقاله به بررسی و عملکرد الگوریتم جهش قورباغه (SFLA) در حل مسائل زمانبندی پروژه با محدودیت منابع پایه پرداخته می‌شود که نتایج حاکی از عملکرد مناسب و قوی این الگوریتم فراابتکاری جدید می‌باشد.

کلمات کلیدی: SFLA, RCPSP, زمانبندی پروژه، الگوریتم فراابتکاری.

## ۱- مقدمه

مسئله زمانبندی پروژه با محدودیت منابع (RCPS) یکی از مسائل بسیار معروف و مطرح در زمینه تحقیق در عملیات و مدیریت پروژه است. این دسته از مسائل به دودلیل مورد توجه قرار گرفته‌اند: اول اینکه با توجه به شرایط متفاوت کاربردی و صنعتی از نظر تابع هدف، خصوصیات فعالیت‌ها، نوع روابط پیش‌نیازی بسیار متنوع‌اند و دوم اینکه با توجه به NP-Hard بودن این مسائل، محققین همواره به دنبال ارائه راه حل‌های کارآتری برای حل این دسته از مسائل بوده‌اند (Demeulemeester et al., 2002).

روش‌های ابتکاری برای حل این مسائل را می‌توان به ۲ دسته تقسیم کرد: دسته اول روش‌هایی هستند که فعالیت‌ها را طبق یک قاعده اولویت دهی<sup>۱</sup> مرتب می‌کنند و سپس در هر مقطع زمانی از بین فعالیت‌های باقیمانده با رعایت محدودیت‌های پیش‌نیازی و ظرفیت منابع، فعالیت‌ها را طبق فهرست مرتب شده برای تخصیص منابع در آن مقطع زمانی انتخاب می‌کنند. عیب اصلی این روش‌ها در این است که نمی‌توان یک قاعده کلی برای مرتب کردن فعالیت‌ها ارائه نمود و مطلوبیت جواب حاصل از قواعد اولویت دهی مختلف به شبکه فعالیت‌های پروژه بستگی دارد و بدین معنی که چنانچه قاعده‌ای که برای یک مسئله خاص جواب بهینه را به دست دهد لزوماً همیشه موفق نخواهد بود. دسته دوم روش‌های فراابتکاری از یک یا چند جواب اولیه شروع می‌کند و بر اساس جواب‌های موجود در هر روش به بهبود آن جواب‌ها و نزدیک شدن به جواب بهینه اقدام می‌کند. از جمله این روش‌ها می‌توان روش‌های جستجوی ممنوع، الگوریتم ژنتیک، الگوریتم مورچگان و... نام برد (Brucker et al., 1999). همچنین با وجود مطالعات گسترده صورت گرفته در زمینه زمان‌بندی پروژه بر اساس نظر محققان خبره در این زمینه هنوز کمبودهایی بسیار زیادی وجود دارد که این کمبودها از دو جنبه مطرح هستند:

- توسعه مدل‌هایی نزدیک‌تر به مسائل واقعی
- بهبود روش‌های حل مساله زمان‌بندی

این پژوهش بر روی جنبه دوم متمرکز خواهد شد

برای زمان‌بندی پروژه‌ها بدون در نظر گرفتن محدودیت منابع از متدهایی همچون روش مسیری‌بحرانی<sup>۲</sup> (برای زمان‌بندی پروژه-هایی که زمان فعالیت‌ها در آن معین است) و تکنیک ارزیابی و بازبینی پروژه (برای زمان‌بندی پروژه‌هایی که زمان فعالیت‌ها در آن قطعی نیست) استفاده می‌شود، خصوصاً روش مسیری‌بحرانی دارای کارایی بسیار خوبی است. لیکن امروزه با توجه به اینکه در تمام سطوح پروژه با محدودیت منابع مواجه هستیم، مسئله‌ای به نام زمان‌بندی پروژه تحت شرایط محدودیت منابع مطرح شده است که به علت اینکه در دو روش فوق‌الذکر محدودیت منابع در نظر گرفته نمی‌شود نمی‌توان برای حل این مسئله از آنها استفاده کرد. لذا لزوم ایجاد و بکارگیری روشهایی که انواع محدودیت‌های منابع را در نظر بگیرند آشکار است (Sprecher A., 1997)

در این پژوهش، مسئله زمانبندی پروژه با محدودیت منابع از طریق ایجاد مدلی با در نظر گرفتن اهداف مهمی از قبیل کمینه کردن زمان اتمام پروژه و کمینه کردن هزینه کل پروژه که از مهمترین اهداف در دنیای واقعی است و همواره مدیران پروژه در جستجوی بهترین نقطه درخصوص زمان-هزینه می‌باشند، در نظر گرفته شده است. همچنین برای جامعتر کردن مدل تمامی روابط پیش‌نیازی ممکن بین فعالیت‌های یک پروژه نیز مورد توجه قرار گرفته است و مسئله به نحوی گسترش یافته تا در دنیای واقعی قابل به کارگیری باشد.

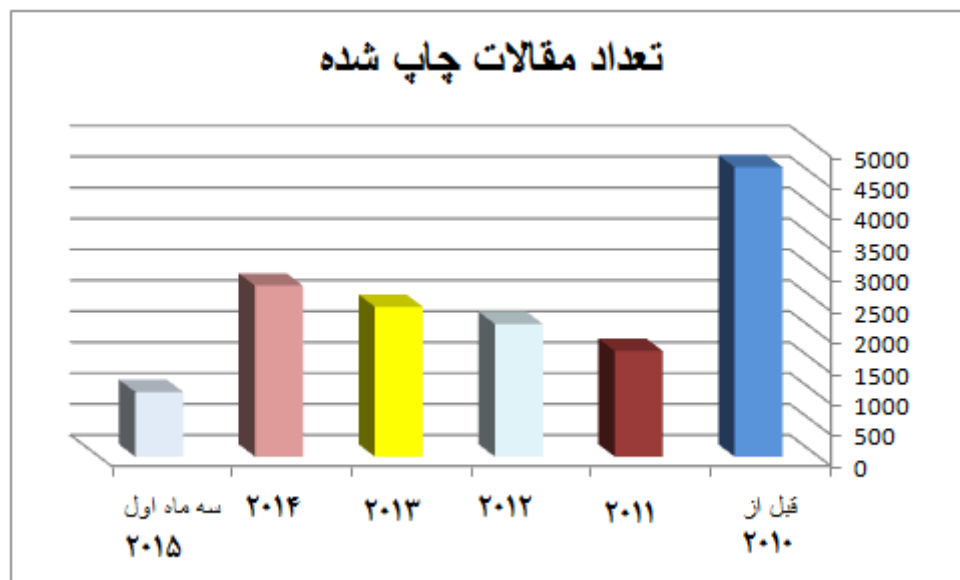
## ۲- مواد و روشها

از سال ۱۹۵۰ تاکنون مقالات بیشماری در این زمینه به چاپ رسیده است و این امر هنوز هم ادامه دارد که این خود گواه بر اهمیت و ضرورت انجام تحقیقات آتی در این زمینه است.

<sup>1</sup> Priority Rule

<sup>2</sup> Critical Path Method (CPM)

تعداد مقالات چاپ شده در این زمینه که در نمودار شماره ۱ زیر تحت جستجویی که با عنوان "Resource constrained project scheduling" در سایت ساینس دایرکت و اسکوپس صورت گرفته است نشان دهنده اهمیت این موضوع و همچنین جذابیت این مسئله از ابتدا تاکنون و ضرورت انجام تحقیقات آتی در این زمینه است.



شکل شماره (۱): نمودار تعداد مقالات چاپ شده در زمینه RCPSP در سال‌های مختلف

روند صعودی تعداد مقالات چاپ شده در سال‌های متوالی خود گواه بر جذابیت و اهمیت این مسئله بوده و گستردگی آن و نیاز پژوهشی آینده آن را نمایان کرده است.

چرخه یک پروژه را می‌توان به پنج مرحله مجزا تقسیم کرد. در هر یک از این مرحله‌ها، وظایف تعریف شده‌ای برای مدیران پروژه تعریف می‌شود. پروژه با یک طرح پیشنهادی شروع می‌شود و انجام مطالعات آغازین همچون امکان‌سنجی، تحلیل اقتصادی یا آنالیز ریسک معمولاً در مرحله توجیه پروژه صورت می‌گیرد. خروجی این مرحله‌ها قاعدتاً تایید و یا رد انجام پروژه است. در مرحله تعریف پروژه، اهداف پروژه مشخص می‌شوند، ساختار سازمانی پروژه مشخص می‌شود. منابع به پروژه تخصیص داده می‌شود و فعالیت‌های متنوع با مقاطع بازنگری مشخص می‌شوند. در مرحله برنامه‌ریزی پروژه ابتدا با تحلیل ساختاری پروژه روابط تقدم-تاخر مابین فعالیت‌های پروژه مشخص می‌شود، سپس زمان اجرای فعالیت‌ها و منابع مورد نیاز هر فعالیت تخمین زده می‌شود. در نتیجه این تحلیل‌های ساختاری، پروژه به وسیله شبکه پروژه نمایش داده می‌شود. پس از آن، زمان‌بندی موقت فعالیت‌ها، زودترین و دیرترین زمان ممکن برای آغاز فعالیت‌ها را مشخص می‌کند. در هنگام تهیه این زمان‌بندی موقت، محدودیت منابع در نظر گرفته نمی‌شود. مرحله نهایی و مشکل‌ترین مرحله‌ای این فاز، تخصیص منابع به فعالیت‌ها در طی مراحل انجام پروژه است. طی فاز اجرای پروژه، نظارت بر انجام فعالیت‌ها، اجرای صحیح و به‌موقع فعالیت‌ها کنترل می‌شود. در این فاز پیشرفت پروژه با برنامه زمان‌بندی تهیه شده که در فاز برنامه‌ریزی پروژه تهیه شده مقایسه می‌شود. در مواردی که انحراف قابل توجهی از برنامه اتفاق افتاده باشد، تخصیص منابع باید دوباره صورت گیرد (Buddhakulsomsiria J., Kim D., 2006).

مرحله نهایی، فاز خاتمه پروژه<sup>۳</sup> است. در این مرحله، پروژه ارزیابی و مستندسازی می‌شود تا مدیریت بتواند در پروژه‌های بعدی از آنها استفاده کند. هر یک از فازها نیازمند تکنیک‌هایی است. این پژوهش در مورد روش‌های کمی برای فاز برنامه‌ریزی پروژه است. (Brucker et al., 1999).

برای زمان‌بندی پروژه‌ها بدون در نظر گرفتن محدودیت منابع از متدهایی همچون روش مسیری بحرانی<sup>۴</sup> (برای زمان‌بندی پروژه-هایی که زمان فعالیت‌ها در آن معین است) و تکنیک ارزیابی و بازبینی پروژه برای زمان‌بندی پروژه‌هایی که زمان فعالیت‌ها در

<sup>3</sup> Project Termination

آن قطعی نیست) استفاده می‌شود، خصوصاً روش مسیر بحرانی دارای کارایی بسیار خوبی است. لیکن امروزه با توجه به اینکه در تمام سطوح پروژه با محدودیت منابع مواجه هستیم، مسئله‌ای به نام زمان‌بندی پروژه تحت شرایط محدودیت منابع<sup>۵</sup> مطرح شده است که به علت اینکه در دو روش فوق‌الذکر محدودیت منابع در نظر گرفته نمی‌شود نمی‌توان برای حل این مسئله از آنها استفاده کرد. لذا لزوم ایجاد و بکارگیری روشهایی که انواع محدودیت‌های منابع را در نظر بگیرند آشکار است. مساله زمان‌بندی پروژه با منابع محدود در واقع کلی‌ترین مساله زمان‌بندی است. مسائل زمان‌بندی کارگاهی<sup>۶</sup>، جریان کارگاهی<sup>۷</sup>، زمان‌بندی و سایر مسائل زمان‌بندی همگی زیر مجموعه‌ای از این مسئله به حساب می‌آید (Buddhakulsomsiria J., Kim D., 2007). زمان‌بندی پروژه یکی از وظایف اصلی و فعالیت‌های اصلی در مدیریت پروژه است. وجود محدودیت منابع و همچنین روابط پیش‌نیازی بین فعالیت‌ها مسئله زمان‌بندی پروژه را امری دشوار می‌سازد. در عمل نرم افزارهای خاصی برای فرآیند زمان‌بندی به کار برده می‌شود. اساس این نرم افزارها یک مدل رسمی است که اجازه می‌دهد که پروژه واقعی تنها با یک مجموعه از محدودیت‌های زمان‌بندی و یک تابع هدف توصیف شود (Schultmann F., Rentz O., 2001). برای زمان‌بندی پروژه‌ها به طور کلی از دو حالت استفاده می‌شود:

در روش اول ابتدا برنامه زمان‌بندی فقط با در نظر گرفتن روابط پیش‌نیازی تعیین شده (با روش CPM/PERT) ایجاد شده، سپس با وارد کردن منابع، میزان مصرف منابع در دوره‌های زمانی و نحوه توزیع آنها در طول برنامه بدست می‌آید. با توجه به این اطلاعات اقدام به تامین منابع یا برون‌سپاری کارها با هدف تحقق برنامه صورت می‌گیرد. در صورت وجود محدودیت در سطوح منابع اقدام به تسطیح منابع می‌گردد. تسطیح منابع ممکن است منجر به طولانی شدن زمان پروژه گردد. در این روش زمان‌بندی بهینه نیست اما قابل قبول است ولی به علت سهولت محاسباتی این روش بیشتر مورد استفاده قرار می‌گیرد و نرم افزارهای تجاری، این روش را پوشش می‌دهند.

در روش دوم زمان‌بندی، در واقع تعیین برنامه با توجه همزمان به پیش‌نیازها و سطح منابع صورت می‌گیرد. در واقع محاسبات شبکه بر ارضاء محدودیت‌های منابع اولویت ندارد بلکه به صورت همزمان در نظر گرفته می‌شوند. در این روش هدف در اغلب اوقات حداقل کردن زمان اتمام پروژه است. تک هدفه بودن از کاستی‌های مطالعات انجام گرفته درباره این مدل‌ها می‌باشد. مدل‌هایی که برای تهیه زمان‌بندی به روش دوم توسعه داده شده‌اند در ادبیات به RCPSPP مشهور هستند (Demeulemeester et al., 2007).

زمان‌بندی پروژه با در نظر گرفتن محدودیت منابع (RCPSPP) عبارت است از زمان‌بندی فعالیت‌های پروژه با توجه به روابط پیش‌نیازی و محدودیت منابع. مسئله RCPSPP شامل یک پروژه با  $J$  فعالیت است که بصورت  $J = 1, \dots, J$  نمایش داده می‌شوند. زمان انجام هر فعالیت  $J$  با  $D_j$  نمایش داده می‌شود. هر فعالیت فقط یکبار می‌تواند شروع شود، و فعالیت می‌تواند قابل انقطاع<sup>۸</sup> باشد یا نباشد. به علت نیازهای فنی، یک سری روابط پیش‌نیازی<sup>۹</sup> بین فعالیت‌ها وجود دارد که به این صورت مجموعه‌ای از روابط به صورت  $P_j$  نمایش داده می‌شود که نشان می‌دهد که یک فعالیت  $J$  امکان شروع شدن ندارد مگر در حالتی که تمامی روابط پیش‌نیازی و پیش‌نیازهایش ( $i \in P_j$ ) کامل شده باشد. روابط پیش‌نیازی می‌تواند با استفاده از شبکه‌های فعالیت روی گره<sup>۱۰</sup> (AON) نمایش داده شود که این با فرض غیر مدور بودن شبکه نمایش داده می‌شود. هر فعالیت یک مقدار مشخص از منابع را برای انجام و اجرا نیاز دارد.

منابعی "منابع تجدیدپذیر" نامیده می‌شوند که تمامی ظرفیت آنها در تمامی دوره‌های زمانی موجود باشد. اگر  $K$  منبع تجدیدپذیر در پروژه موجود باشد، این منابع به صورت  $K = 1, \dots, k$  نمایش داده می‌شود. برای هر منبع  $k$  در هر دوره سطح

<sup>4</sup> Critical Path Method (CPM)

<sup>5</sup> Resource Constrained Project Scheduling problem (RCPSPP)

<sup>6</sup> Job shop

<sup>7</sup> Flow shop

<sup>8</sup> Preemptive

<sup>9</sup> Precedence Relations

<sup>10</sup> Activity On Node

دسترسی وجود دارد که در تمامی زمان‌ها ثابت است که این سطح دسترسی منابع تجدیدپذیر با  $R_k$  تعریف می‌شود (Ranjbar M., Kianfar F., 2007) فعالیت  $j$  به  $r_{jk}$  واحد از منبع  $k$  در هر دوره‌ای که آن فعالیت در حال اجراست نیاز دارد. دو فعالیت مجازی  $J = 0$  و  $J = J + 1$  که زمان شروع و پایان پروژه را نشان می‌دهند نیز در نظر گرفته می‌شود که زمان انجام این دو فعالیت صفر و بدون نیاز به منبع می‌باشند.

همه اطلاعات به صورت قطعی می‌باشند و پارامترها هم به صورت عدد صحیح و غیر منفی می‌باشند. هدف این مسئله پیدا کردن برنامه‌ای است که زمان‌های شروع  $S_j$  را برای فعالیت‌های  $j = 0, \dots, j + 1$  به گونه‌ای معین کند که هدف برنامه یعنی حداقل کردن زمان اتمام پروژه تحقق یابد.

بروکر و همکاران<sup>۱۲</sup> (Ranjbar M et al., 2009) یک سری نماد و نشانه برای طبقه بندی این دسته از مسائل ایجاد کرده‌اند. این نماد به سه نماد اصلی  $\alpha | \beta | \gamma$  که توسط گراهام و همکاران<sup>۱۳</sup> (Schulmann F., Rentz O., 2001) بوجود آمده است، منتهی می‌شود. در زمینه زمان بندی پروژه، سمبل  $\alpha$  مخصوص منابع است و سمبل  $\beta$  فعالیت‌ها را توصیف می‌کند و می‌تواند بیش از یک مقدار بگیرد و نهایتاً سمبل  $\gamma$  هدف یا هدف‌های مسئله را نشان می‌دهد.

مدل استاندارد RCPSP به صورت  $ps | prec | C_{max}$  نمایش داده می‌شود که به ترتیب نمایانگر مسئله زمانبندی پروژه با روابط پیش‌نیازی بین فعالیت‌ها و هدف کمینه کردن زمان اتمام پروژه است. با آنکه مدل RCPSP بیان شده در بالا یک مدل بسیار توانا است. اما نمی‌تواند تمامی موقعیت‌ها را در واقعیت و عمل پوشش دهد، بنابراین خیلی از محققین مدل‌های کلی بسیاری را برای مسئله زمان بندی پروژه توسعه داده‌اند که غالباً با یک RCPSP استاندارد به عنوان نقطه شروع کار می‌کند. از دهه اول ۱۹۹۰ تعداد زیادی از مقالات در این زمینه به چاپ رسیده است. محققان بر روش‌ها و انواع مسائل RCPSP توجه کرده‌اند. برای نمونه می‌توان به هارتمن و کولیش<sup>۱۴</sup> و هارتمن و کولیش اشاره کرد. (Buddhakulsomsiria J., Kim D., 2006)

حال به مدل‌های ریاضی که تاکنون توسعه داده شده اند پرداخته می‌شود: در ابتدا مدل‌های ریاضی که از ابتدا تاکنون برای این دسته از مسائل بوجود آمده اند بصورت لیست وار بررسی می‌شوند و سپس به روش‌های حل اشاره می‌شود. در جدول زیر مدل‌های ایجاد شده تاکنون آورده شده است:

نام مدل	سال	منبع
مدل پریسکر <sup>۱۵</sup>	۱۹۶۹	(Sadeghi et al., 2011)
مدل کاپلن <sup>۱۶</sup>	۱۹۸۸	(Sadeghi et al., 2011)
مدل الوارز	۱۹۹۳	(Sadeghi et al., 2011)
مدل کلین <sup>۱۷</sup>	۲۰۰۰	(Sadeghi et al., 2011)

در خصوص روش‌های حل مدل نیز می‌توان بیان نمود که مسائل RCPSP از دسته مسائل NP-Hard هستند، لذا از دو طریق قابل حل هستند:

- ۱- روش‌های دقیق<sup>۱۸</sup>  
 ۲- روش‌های ابتکاری

<sup>11</sup> Dummy

<sup>12</sup> Brucker

<sup>13</sup> Graham

<sup>14</sup> Hartmann and coilsch

<sup>15</sup> Pritsker

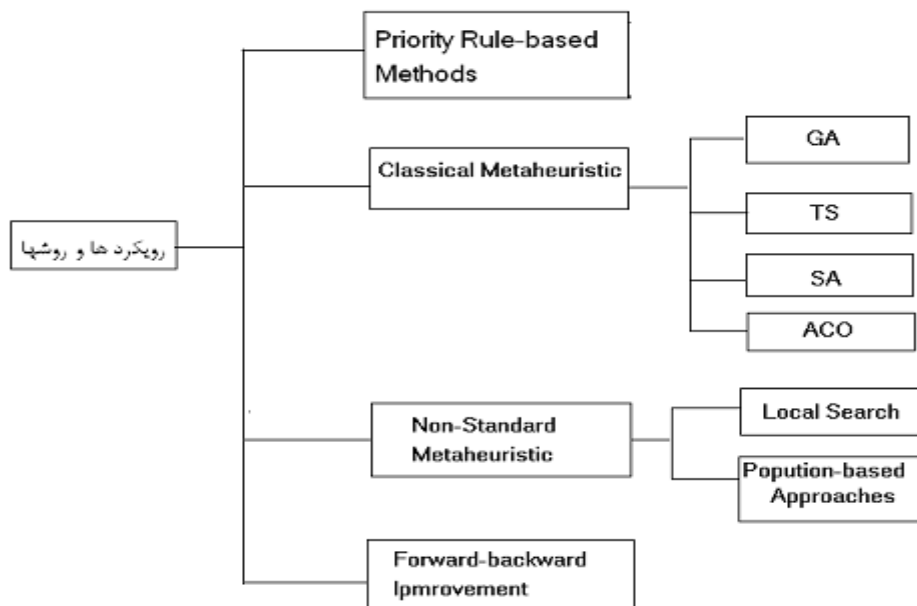
<sup>16</sup> Kaplan

<sup>17</sup> Klein

<sup>18</sup> Exact algorithm

در روش‌های دقیق در انتهای اجرای الگوریتم جواب بهینه مسئله به دست می‌آید. از جمله این روش‌ها، انواع برنامه ریزی خطی، برنامه ریزی صفر و یک، برنامه ریزی عدد صحیح، برنامه ریزی پویا، برنامه ریزی غیر خطی، روش‌های شاخه و کران و... می‌باشد. از آنجا که برای مسائل پیچیده معمولاً این نوع الگوریتم‌ها در زمان معقول جواب بهینه را بدست نمی‌آورند، برای این گونه از مسائل به الگوریتم‌های دیگری متوسل می‌شویم که در زمان کم، جواب قابل قبولی به ما بدهند. این دسته از مسائل را روش‌های نادقیق (ابتکاری) می‌گویند که اخیراً توجه بیشتری را به خود جلب نموده‌اند.

الگوریتم‌های مختلف حل مسائل RCPSP در شکل زیر خلاصه شده است:



شکل شماره (۲): انواع الگوریتم‌های حل مسئله (Marco et al., 2000)

در این پژوهش از الگوریتم‌های فراابتکاری برای حل کمک گرفته خواهد شد، لذا فقط این الگوریتم‌ها مورد بررسی قرار می‌گیرد. در خصوص الگوریتم‌های حل نیز به الگوریتم جهش ترکیبی قورباغه پرداخته می‌شود که الگوریتم جهش ترکیبی قورباغه (SFLA) یک الگوریتم مبتنی بر ممتیک متاهپوریستیک است. این الگوریتم در سال‌های اخیر توسط Eusuff و Lansey ایجاد شد. الگوریتم SFLA از نحوه‌ی جستجوی غذای گروه‌های قورباغه سرچشمه می‌گیرد. این الگوریتم برای جستجوی محلی میان زیرگروه‌های قورباغه از روش نموممتیک استفاده می‌کند. SFLA از استراتژی ترکیب استفاده می‌کند و امکان مبادله پیام در جستجوی محلی را فراهم می‌سازد. این الگوریتم مزایای الگوریتم نمو ممتیک و بهینه‌سازی گروه ذرات (PSO) را ترکیب می‌کند. در SFLA نه تنها در جستجوی محلی بلکه در جستجوی سراسری نیز پیام‌ها مبادله می‌شوند. بدین ترتیب جستجوی محلی و سراسری به خوبی در این الگوریتم ترکیب می‌شوند. جستجوی محلی امکان انتقال مم را میان افراد، ممکن می‌سازد و استراتژی ترکیب امکان انتقال مم را میان کال جمعیت ممکن می‌سازد. الگوریتم جهش ترکیبی قورباغه یک الگوریتم بهینه‌سازی مبتنی بر کلونی است. SFLA قابلیت بالایی برای جستجوی سراسری دارد و پیاده‌سازی آن آسان است. الگوریتم SFLA می‌تواند بسیاری از مسائل غیر خطی قابل تشخیص ۲۰ و چندحالتی ۲۱ را حل کند.

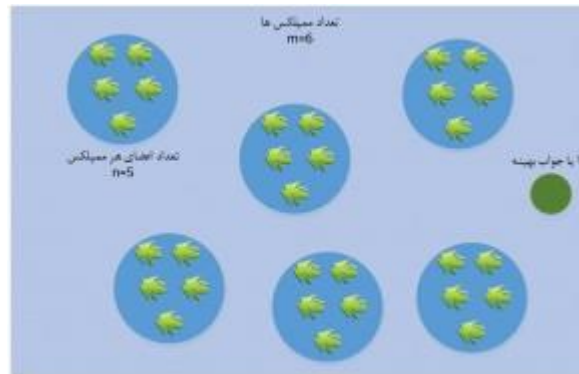
SFLA از یک جمعیت مجازی از قورباغه‌های تشکیل شده است که در یک آبگیر به دنبال غذا هستند. بطور کلی همواره قورباغه‌های این آبگیر به سمت نقاطی حرکت میکنند که غذای بیشتری در آن‌ها وجود داشته باشد. بنابراین اگر آبگیر را به

<sup>19</sup> particle swarm optimization

<sup>20</sup> non-differentiable

<sup>21</sup> multimode

عنوان فضای جواب در نظر بگیریم، نقاطی که در آنها غذا وجود دارد نقاط بهینه محلی و نقطه ای که بیشترین غذا در آن وجود دارد، بهینه سراسری مساله میباشد.



شکل شماره (۳): جمعیت مجازی قورباغه ها

نحوه کار SFLA بدین صورت است که قورباغه ها در ابتدا بر حسب برازندگی و با رویه ای معین به گروه هایی تقسیم میشوند که هر کدام از این گروه ها را یک ممپلکس (Memplex) مینامند. سپس در چندین تکرار، مکان بدترین قورباغه هر ممپلکس (که نشان دهنده یک جواب شدنی برای مساله است) با توجه به مکان بهترین قورباغه درون آن ممپلکس، و همینطور مکان بهترین قورباغه موجود در آنگیر بهبود پیدا میکند. بعد از اینکه این بهبود N بار انجام شد، بار دیگر قورباغه های آنگیر بر اساس برازندگی مرتب شده و مجدداً ممپلکس های جدیدی شکل میگیرد. این کار تا رسیدن به شرط توقف ادامه پیدا میکند. از اصلی ترین نقاط قوت الگوریتم SFLA میتوان به سرعت بسیار زیاد آن اشاره کرد (Wang et al., 2013).

در خصوص الگوریتم ترکیبی جهش قورباغه که مورد توجه در این پژوهش است می توان گفت که الگوریتم SFLA ترکیب روش قطعی و روش تصادفی است. روش قطعی به الگوریتم امکان می دهد تا پیام ها را به صورت کارایی مبادله کند. روش تصادفی انعطاف پذیری و مقاومت الگوریتم را تضمین می کند. الگوریتم با انتخاب تصادفی گروه های قورباغه شروع می شود. گروه های قورباغه به چندین زیرگروه تقسیم می شوند. هر یک از این زیرگروه ها می توانند جستجوی محلی را به صورت مستقل و با روش متفاوتی انجام دهند. قورباغه های موجود در یک زیرگروه می توانند بر روی سایر قورباغه های موجود در همان زیرگروه اثر بگذارند. بدین ترتیب قورباغه های موجود در یک زیرگروه تکامل می یابند. تکامل ممپلکس کیفیت ممپلکس قورباغه های منفرد را بهبود و قابلیت دستیابی به هدف را افزایش می دهد. برای رسیدن به یک هدف خوب می توان وزن قورباغه های خوب را افزایش داد و وزن قورباغه های بد را کاهش داد. بعد از تکامل برخی از ممپلکس ها، زیرگروه ها با هم ترکیب می شوند. به واسطه ترکیب ممپلکس ها در حوزه سراسری بهینه می شوند و به وسیله مکانیزم ترکیب زیرگروه های قورباغه ای جدیدی ایجاد می شود. ترکیب، کیفیت ممپلکس هایی که تحت تأثیر زیرگروه های مختلف قرار می گیرند را افزایش می دهد. جستجوی محلی و جستجوی سراسری تا برآورده شدن شرط همگرایی ترکیب می شوند. توازن بین مبادله پیام سراسری و جستجوی محلی به الگوریتم امکان می دهد تا به راحتی از مینیمم محلی پرش کند و تا دستیابی به بهینه سازی توسعه یابد. یکی از خصیصه های الگوریتم SFLA همگرایی سریع آن است.

Pseudo کد این الگوریتم در ذیل آورده شده است:

Begin;

Generate random population of P solutions (frogs) feasible zone;

For each individual i P: calculate fitness (i);

Sort the population P in descending order of their fitness;

Divide P into m memplexes;

For i=1 to number of generations

For each memplex;

Determine the best and worst frogs;

Improve the worst frog position using Eqs. (1), (2)

End;  
 Combine the evolved memplexes;  
 Sort the population P in descending order of their fitness;  
 Check if termination=true;  
 End;  
 End;

#### جستجوی سراسری SFLA

- (۱) مقاردهی اولیه:  $m$  و  $n$  را انتخاب کن،  $m$  تعداد memplex ها و  $n$  تعداد قورباغه‌های موجود در هر memplex را نشان می‌دهد. بنابراین اندازه کل جمعیت موجود در آنگیر از طریق رابطه‌ی  $F = m \cdot n$  به دست می‌آید.
- (۲) تولید جمعیت مجازی: از فضای شدنی،  $F$  قورباغه مجازی  $(U(1), U(2), \dots, U(F))$  را نمونه‌برداری کن. مقدار شایستگی  $f(t)$  هر قورباغه  $U(i)$  را به ازاء هر  $U(i) = (U_i^1, U_i^2, \dots, U_i^d)$ ، محاسبه کن.  $d$  تعداد متغیرهای تصمیم است.
- (۳) درجه‌بندی قورباغه‌ها: قورباغه‌ها را بر اساس شایستگی‌شان به صورت نزولی مرتب و در آرایه‌ی  $X = \{U(i), f(i), i = 1, \dots, F\}$  ذخیره کن. موقعیت بهترین قورباغه  $P_x$  در کل جمعیت را ثبت کن (که  $P_x = U(1)$ ).
- (۴) تقسیم قورباغه‌ها در memplex ها: آرایه‌ی  $X$  را در  $m$  memplex  $(Y^1, Y^2, \dots, Y^m)$  که هر کدام شامل  $n$  قورباغه هستند، تقسیم کن.
- (۵) تکامل ممپلکس در هر memplex: هر memplex  $(Y^k, k = 1, \dots, m)$  به وسیله‌ی جستجوی محلی (الگوریتم جهش قورباغه) که در ادامه توضیح داده شده است تکامل می‌یابد.
- (۶) ترکیب memplex ها: بعد از اینکه در هر memplex تعداد معینی تکامل ممپلکس انجام شد، memplex ها را  $(Y^1, \dots, Y^m)$  در  $X$  قرار بده، به طوری که رابطه‌ی  $X = Y^k, k = 1, \dots, m$  برقرار باشد. موقعیت قورباغه‌ی موجود در جمعیت  $(P_x)$  را بهنگام کن.
- (۷) بررسی همگرایی: اگر شرایط همگرایی برآورده شده است، موقوف شو. در غیر این صورت به مرحله‌ی چهارم از جستجوی سراسری برو.

#### جستجوی محلی SFLA

در مرحله‌ی پنجم جستجوی سراسری، تکامل هر memplex به صورت مستقل از  $N$  بار انجام می‌شود. بعد از اینکه memplex ها تکامل یافتند، الگوریتم جهت انجام ترکیب به جستجوی سراسری بازمی‌گردد. در ادامه جزئیات جستجوی محلی در هر memplex تشریح می‌شود:

(۱) مقاردهی اولیه:  $im$  و  $in$  را برابر صفر قرار بده،  $im$  تعداد memplex ها و  $in$  تعداد مراحل تکامل را می‌شمارد.

$$im = im + 1 \quad (2)$$

$$in = in + 1 \quad (3)$$

(۴) ایجاد یک submemplex: هدف قورباغه‌های این است که با بهبود مم هایشان به سمت موقعیت‌های بهینه حرکت کنند. روش انتخاب submemplex تخصیص وزن‌های بیشتر به قورباغه‌هایی که کارایی بالاتر دارند، و وزن‌های کمتر به قورباغه‌هایی با مقادیر کارایی کمتر است. وزن‌ها با توزیع احتمال مثلثی تخصیص داده می‌شوند، یعنی برای ساخت آرایه‌ی submemplex  $(Z)$  از هر  $n$  قورباغه‌ی موجود در هر memplex تعداد  $q$  قورباغه به طور تصادفی انتخاب می‌شوند. قورباغه‌های موجود در submemplex برحسب میزان شایستگی‌شان به صورت نزولی مرتب می‌شوند. موقعیت بهترین قورباغه و بدترین قورباغه‌ی موجود در submemplex به ترتیب با  $P_B$  و  $P_W$  مشخص می‌شوند.



۵) تصحیح موقعیت بدترین قورباغه: موقعیت جدید بدترین قورباغه‌ی موجود در submemplex (قورباغه‌ای که بدترین مقدار کارایی را دارد) از طریق رابطه‌ی  $U(q) = P_W + S$  محاسبه می‌شود.  $S$  اندازه‌ی گام (میزان جهش) قورباغه است و از طریق رابطه‌ی زیر به دست می‌آید:

$$S = \min\{\text{int}[\text{rand}(P_B - P_W)], S_{\max}\} \text{ for a positive step,} \\ = \max\{\text{int}[\text{rand}(P_B - P_W)], S_{\min}\} \text{ for a negative step.}$$

۶) محاسبه‌ی اندازه‌ی گام به‌وسیله‌ی  $P_x$ : اگر در مرحله‌ی ۵ نتیجه‌ی بهتری تولید نشد، آنگاه اندازه‌ی گام قورباغه از طریق رابطه‌ی زیر محاسبه می‌شود:

$$p_1 = 2(n + 1 - f)/n(n + 1), j = 1, \dots, n$$

و موقعیت جدید  $U(q)$  به‌وسیله‌ی رابطه‌ی  $U(q) = P_W + S$  محاسبه می‌شود. اگر  $U(q)$  در بین فضای شدنی باشد، مقدار کارایی جدید  $t(q)$  محاسبه می‌شود. چنانچه  $f(q)$  جدید بهتر از قبلی است، آنگاه  $U(q)$  جدید را جایگزین  $U(q)$  قبلی کن و به مرحله‌ی هشتم جستجوی محلی برو. در غیر این صورت به مرحله‌ی هفتم جستجوی محلی برو.

۷) سانسور: اگر موقعیت جدید در ناحیه‌ی شدنی نیست، به‌صورت تصادفی یک قورباغه جدید ( $r$ ) در یک مکان شدنی تولید و جایگزین قورباغه‌ای می‌شود که موقعیت جدیدش برای پیشروی مناسب نیست.  $F(r)$  را محاسبه کن و  $U(q)$  را برابر  $r$  و  $f(q)$  را برابر  $f(r)$  قرار بده.

۸) بهنگام کردن memplex: بعد از تغییر ممپلک بدترین قورباغه‌ی موجود در submemplex، قورباغه‌های موجود در  $Z$  را در موقعیت اصلی‌شان  $Y^{im}$  قرار بده.  $Y^{tm}$  را بر اساس مقدار کارایی به‌صورت نزولی مرتب کن.

۹) اگر  $iN < N$  است، به مرحله‌ی سوم جستجوی محلی برو.

۱۰) اگر  $im < m$  است، به مرحله‌ی اول جستجوی محلی برو. در غیر این صورت جهت ترکیب memplex ها به جستجوی سراسری بازگرد.

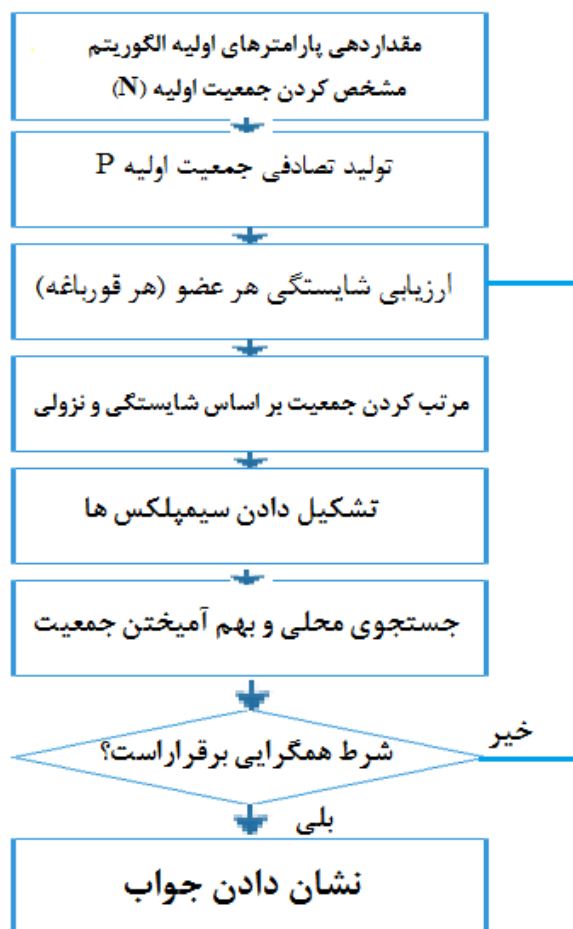
پارامترهای بهینه سازی SFLA

مانند سایر الگوریتم های تکاملی پارامترها نقش مهمی در بازدهی نهایی دارند:

- ✓ تعداد قورباغه های هر زیردسته:  $n$
  - ✓ تعداد دسته های زیر جمعیت (Memplex):  $m$
  - ✓ تعداد نسل های هر زیر جمعیت قبل از جابه جایی (Shuffling):  $q$
  - ✓ تعداد دوره های جابه جایی:  $N$
  - ✓ بیش ترین اندازه گام:  $S_{\max}$
- مهم ترین عامل  $m*n$  است که به پیچیدگی مسئله بستگی دارد.
- ✓ هرچه  $F$  بیشتر: بازدهی بهتر، بار محاسباتی بالاتر
  - ✓ کاهش تعداد  $n$ : کاهش تاثیر استراتژی تکاملی
  - ✓  $q$  بالاتر: زمان همگرایی رابطه غیر مستقیم دارد.
  - ✓  $N$  کمتر: جابجایی کمتر اطلاعات محلی و افزایش تاثیر جستجوی محلی
  - ✓ اندازه گام کوچک: توانایی کمتر جستجوی سراسری،
  - ✓ اندازه گام بلند: احتمال واگرا شدن مسئله

انواع جستجوهای SFLA

- ✓ جستجوی محلی
- ✓ جستجوی جهانی (جهش قورباغه ها)



شکل شماره (۴): فلوچارت الگوریتم

پارامترهای بهینه سازی SFLA

مانند سایر الگوریتم های تکاملی پارامترها نقش مهمی در بازدهی نهایی دارند:

- ✓ تعداد قورباغه ها
- ✓ تعداد دسته های زیر جمعیت (Memplex)
- ✓ تعداد نسل های هر زیر جمعیت قبل از جابه جایی (Shuffling)
- ✓ تعداد دورهای جابه جایی

در خصوص تحقیقات داخلی و خارجی که تاکنون صورت گرفته می توان به موارد زیر اشاره نمود: از جمله تحقیقات داخلی صورت گرفته در زمینه الگوریتم قورباغه می توان به موارد زیر اشاره کرد:

طالبی و همکاران در سال ۱۳۸۹ برای جایابی بهینه خازن که در شبکه توزیع شعاعی یکی از مسائل مهم بهره برداری از سیستمهای قدرت به شمار میرود و با هدف کاهش تلفات در شبکه توزیع، اصلاح پروفیل ولتاژ، اصلاح ضریب توان و نیز کاهش مجموع اغتشاشات هارمونیک THD دنبال می شود و به دلیل دارا بودن مدل ریاضیاتی غیرخطی و غیرتحلیلی از جمله مسایل پیچیده و دشوار در طراحی شبکههای توزیع محسوب میگردد از الگوریتم قورباغه به عنوان یکی از الگوریتمهای بهینه سازی استفاده نمودند (Anas Alghazi et al., 2012).

فارسانی و همکاران نیز در سال ۱۳۸۹ در شبکه های توزیع آرایش شبکه به منظور کاهش تلفات، افزایش قابلیت اطمینان شبکه و بهبود کیفیت توان در شبکه و ... روش حل جدیدی بر مبنای الگوریتم قورباغه بهبود یافته برای حل مسئله ی تجدید آرایش استفاده نمودند. الگوریتم قورباغه علیرغم سرعت بالایش در بعضی از موارد به جواب صحیح همگرا نمی شود و در نتیجه نیاز به

بهبود دارد (Local search). نتایج بدست آمده بیانگر قدرت بالای روش مذکور است (Muzaffar Eusuffa et al., 2006).

جهانی و همکاران نیز در سال ۱۳۹۰ با الگوریتم الگوریتم قورباغه بهبود یافته جهت یافتن تعداد و محل بهینه کنترل کننده های یکپارچه توان UPFC در سیستمهای پرداختند که کنترل کننده های یکپارچه توان به جهت آنکه می توانند توان اکتیو راکتیو خط و ولتاژ شین ها را همزمان کنترل کنند از اهمیت خاصی برخوردار هستند و انعطاف پذیری بالایی در شبکه دارند در این تحقیق از مدل مجزای کنترل کننده یکپارچه توان جهت حداکثر کردن ظرفیت بارگذاری خطوط انتقال استفاده می شود در ضمن محدودیت ظرفیت خطوط انتقال و سطح ولتاژ معین به عنوان قیود در نظر گرفته می شود جایابی بهینه UPFC توسط الگوریتم قورباغه بهبود یافته منجر به پروفیل ولتاژی صاف و افزایش ظرفیت و پایداری خطوط انتقال توان می گردد. درانتها نیز به شبیه سازی و بررسی نتایج حاصل از آن خواهیم پرداخت (TAVAKOLAN, 2011).

عروجی و همکاران در سال ۱۳۹۲ نیز مدل ماسکینگام که از جمله روش های روندیابی هیدرولوژیک است که دقت در تخمین پارامترهای آن بر هیدروگراف روندیابی شده، به خصوص بر حداکثر مقدار سیلاب، تاثیرگذار است با الگوریتم بهینه سازی جهش قورباغه در دو مورد مطالعاتی آزمایشی و واقعی را کار گرفتند و مجموع مربعات و قدر مطلق انحرافات دبی روندیابی شده مشاهداتی و محاسباتی به عنوان تابع هدف و میزان انحراف دبی اوج مشاهداتی و محاسباتی به عنوان پارامتر مهم دخیل در هیدروگراف روندیابی شده سیلاب در نظر گرفتند. نتایج حاکی از آن است که استفاده از الگوریتم بهینه سازی جهش قورباغه به ترتیب سبب کاهش ۰/۰۳ و ۰/۴۰ درصدی در مجموع مربعات و قدر مطلق انحرافات دبی روندیابی شده مشاهداتی و محاسباتی نسبت به بهترین مقادیر موجود در مورد مطالعاتی آزمایشی و ۳/۶۷ و ۰/۲۷ درصدی در مورد مطالعاتی واقعی شده است. همچنین الگوریتم بهینه سازی جهش قورباغه مقدار میزان انحراف دبی اوج مشاهداتی و محاسباتی محاسبه شده را به ترتیب در مورد مطالعاتی آزمایشی و واقعی به ترتیب ۳/۶۷ و ۱۷/۶۵ درصد نسبت به بهترین جواب گزارش شده بهبود داده است (Afshar, 2009).

تحقیقات خارجی صورت گرفته در زمینه زمانبندی پروژه با محدودیت منابع و الگوریتم قورباغه می توان به موارد زیر اشاره نمود: مندز و همکاران در سال ۲۰۰۹ از الگوریتم ژنتیک برای حل مسئله زمانبندی پروژه تک حالت استفاده نمودند. چن و همکاران در سال ۲۰۱۰ نیز از یک الگوریتم هیبریدی برای این دسته از مسائل کمک گرفتند. راهیویی در سال ۲۰۱۲ نیز از یک الگوریتم هیبریدی توزیع برای حل مسائل RCPS و MRCPS استفاده نموده است. ژانگ و ژائو در سال ۲۰۱۲، از الگوریتم قورباغه بر روی سیستمهای رادیویی شناختی استفاده کردند. وانگ و شن در سال ۲۰۱۳، از الگوریتم جهش قورباغه توسعه یافته با بهینه سازی حدی در بهینه سازی مستمر استفاده کردند (Duan, 2012).

جاکینگ و همکاران در سال ۲۰۱۲ به حل مساله Job shop scheduling انعطاف پذیر با الگوریتم قورباغه پرداختند. نتایج نشان دهنده بهره وری و اثربخشی بیشتر این الگوریتم نسبت به سایر الگوریتمهای که تاکنون به کار گرفته شده اند بود. در الگوریتم پیشنهادی، روش های متعددی برای ساخت جمعیت اولیه با سطح بالایی از کیفیت ارائه شده است که هر یک از قورباغه ها در جمعیت به یک memplex مربوطه با توجه به تعداد افرادی که به آن تسلط و پس از آن تعداد قورباغه که آن را تحت سلطه قرار می گیرند. در فرایند تکامل memetic، دو اپراتور متقاطع ارائه شده برای به اشتراک گذاشتن اطلاعات در میان بهترین قورباغه ها و بدترین قورباغه استفاده شده است (Alghazi et al., 2012).

لکویی و همکاران نیز در سال ۲۰۱۲ برای مساله Test Scheduling از این الگوریتم استفاده نمودند که این مسئله ( برنامه ریزی آزمون) یک مسئله مهم برای آزمایش SOC است. این کار از یک الگوریتم اصلاح قورباغه برای زمان بندی آزمون در جهت کاهش زمان برنامه آزمون تحت محدودیت اوج قدرت پیشنهاد شده است. نتایج تجربی در مدارهای معیار نشان می دهد که آن را یکی از الگوریتم های موثر در حل مشکل است استفاده شود (Elbeltagi et al., 2016).

سال چاپ مقاله	مسئله	نام الگوریتم
۲۰۱۴	pick-and-place sequencing optimization problem	Improved Shuffled rog-leaping Algorithm
۲۰۱۴	Optimization for the MDVRP and the MDVRPTW مسئله مسیریابی وسیله نقلیه باقید و بدون قید پنجره زمانی	Shuffled Frog Leaping Algorithm
2014	Multi-depot vehicle routing problem مسئله مسیریابی وسیله نقلیه	Improved Shuffled Frog Leaping Algorithm
۲۰۱۴	Knapsack problem مسئله کوله پشتی	Shuffled Frog Leaping Algorithm
۲۰۱۲	SLLS Problem مسئله اندازه دسته تولید	Shuffled Frog Leaping Algorithm
۲۰۱۱	Multi-objective optimal power flow مسئله چند هدفه جریان بهینه قدرت	Shuffled Frog Leaping Algorithm
۲۰۱۲	Flexible job shop scheduling problems منعطف Job Shop مسئله	Effective shuffled frog-leaping algorithm
۲۰۱۴	In-core fuel management optimization مسئله بهینه سازی سوخت	A multi-objective shuffled frog leaping
۲۰۱۴	Mixed-model assembly line sequencing problem مسئله خط مونتاژ	Hybrid multi-objective shuffled frog-leaping algorithm
۲۰۱۲	Multiobjective Optimization for Optimal Placement and Size of DG مسئله چند هدفه توزیع پراکنده	Shuffled Frog Leaping Algorithm
۲۰۱۳	Unbalanced distribution systems with technical constraints سیستم های توزیع نامتعادل با محدودیت های فنی	Shuffled Frog Leaping Algorithm

### ۳- نتایج و بحث

در ابتدا نتایج الگوریتم جهش قورباغه برای مساله پایه RCPSP آورده شده است. نتایج برای مسائل استاندارد کتابخانه اینترنتی PSPLIB از مجموعه های  $j30$  و  $j60$  و  $j90$  و  $j120$  است. نتایج بر حسب ۵۰۰۰۰ تکرار مرتب شده اند و با جدیدترین الگوریتمهایی که تاکنون برای این مسئله مورد استفاده قرار گرفته اند مقایسه شده است. لازم به ذکر است که برای مسائل  $j30$  جوابهای بهینه در ادبیات موضوع این مسئله بدست آمده است لذا حد پایین برای مقایسه الگوریتمها در این دسته از مسائل همان جواب بهینه (اپتیمال) در نظر گرفته شده است. در مسائل دسته  $j60$  و  $j90$  و  $j120$  هنوز روشی موفق به یافتن تمامی جوابهای بهینه نشده است لذا در این دسته از مسائل، حد پایین جوابهای یافته شده با استفاده از مسیر بحرانی<sup>۲۲</sup> در نظر گرفته شده است که برای دست یافتن به جوابهای این مسائل با استفاده از روش مسیر بحرانی کافی است محدودیت منابع در نظر

<sup>22</sup> Critical Path (CP)

گرفته و مسائل بدون محدودیت منابع اجرا شده اند و نتایج مسیر بحرانی برای هر مسئله بدست آمده است سپس نهایتاً درصد انحراف از میانگین نسبی از این جوابها بدست آمده و در جدول ۲ و ۳ و ۴ گماشته شده است.

نویسندگان مقاله و سال چاپ	نام الگوریتم استفاده شده	منبع	تعداد تکرار-Iteration		
			1000	5000	50000
GA-TS-path relinking-2003	Genetic algorithm & Tabu search	(Zamani,2013)	0/10	0/04	0/00
Sadeghi et al – 2011	Bees algorithm	(sadeghi, et al,2011)	0/15	0/09	0/00
Kuolinas et al-2014	PSO-HHPSO based Hyper Heuristic	(Wang, et al,2014)	0/26	0/04	0/01
پژوهش حاضر	Shuffled Frog-leaping Algorithm	پژوهش حاضر	0/26	0/07	0/01
Vallas et al -2008	Hybrid Genetic Algorithm	(Wang, et al,2014)	0/27	0/06	0/02
Vallas et al-2011	GA-FBI	(Wang, et al,2014)	0/34	0/20	0/02
Wang et al-2012	SFL	(Zamani,2013)	0/38	0/14	#
Ziarati et al-2011	BA-FBI	(sadeghi, et al,2011)	0/42	0/19	0/04
Proon et al -2011	Gans	(Jarbou, et al,2012)	1/83	1/27	0/71

جدول شماره (۲): نتایج عملکرد الگوریتم و مقایسه آن در حل مسائل سری j30

با توجه به جدول ۹ مشخص است که الگوریتم بر اساس ۱۰۰۰ بار تکرار در حل مسائل دسته j30 تنها 0/26 درصد اختلاف از حد پایین سری مسائل J30 داشته است و توانسته رتبه ۴ ام را در بین جدیدترین الگوریتم های حل این دسته از مسائل را بخود اختصاص دهد که بیانگر نتایج خوبی است اما در مورد مسائل بزرگتر مانند j60 و j120 الگوریتم بسیار بهتر عمل نموده است و توانسته جایگاه دوم را در بین جدیدترین الگوریتم های که تاکنون در سطح جهانی برای حل این دسته از مسائل مورد استفاده قرار گرفته اند را به خود اختصاص دهد که این امر نمایانگر قدرت این الگوریتم در حل مسائل با ابعاد کوچک و بزرگ در این سری از مسائل می باشد.

نویسندگان مقاله و سال چاپ	نام الگوریتم استفاده شده	منبع	تعداد تکرار-Iteration		
			1000	5000	50000
Debels and Anhoucke, 2007	Genetic algorithm	(Zamani, 2013)	11/31	10/95	10/68
پژوهش حاضر	Shuffled Frog-leaping Algorithm	پژوهش حاضر	11/35	11/01	10/54
Vallas et al., 2008	Hybrid Genetic Algorithm	(Wang, et al., 2014)	11/56	11/10	10/74
Debels et al., 2006	Scatter Search	(Jozefowska, et al., 2006)	11/73	11/10	10/71
Kuolinas et al.,2014	PSO-HH PSO based Hyper Heuristic	(Wang, et al, 2014)	11/74	11/13	10/68
Chen et al., 2010	ACOSS	(Kolisch et al., 2010)	11/75	10/98	10/67
Sadeghi et al., 2011	Bees algorithm	(sadeghi et al., 2011)	11/93	11/48	10/74
Chen., 2011	JPSO-PSO	(Zamani, 2013)	12/03	11/43	11/00
Ziarari et al., 2011	BA-FBI	(Elloumi et al., 2012)	12/55	12/04	11/16

جدول ۳- نتایج عملکرد الگوریتم و مقایسه آن در حل مسائل سری j60

همانطور که از جدول بالا نمایان است الگوریتم در حل این سری از مسائل بسیار خوب عمل نموده است و مقدار انحراف از حد پایین که در این سری مسائل همان جواب بر اساس مسیر بحرانی یعنی بدون محدودیت منابع است، تنها به میزان 11/35

درصد در ۱۰۰۰ تکرار بوده است که اگر مبنای چیدن الگوریتم ها را در جدول بالا در تکرار ۵۰۰۰۰ بار قرار دهیم این الگوریتم قدرتمند جایگاه اول در سری مسائل j60 می باشد که در خیلی از مقالات موجود مبنای چیدمان این دسته از جداول ۵۰۰۰۰ تکرار است و در این پژوهش و تعداد ی مقالات مبنای سرعت رسیدن به بهترین جواب و تعداد تکرار کم انتخاب می گردد لذا در این پژوهش مبنای ۱۰۰۰ تکرار برای چیدن جدول فوق استفاده شده است . در ادامه نتایج حل الگوریتم برای مسائل سری j120 نیز آورده شده است.

نویسندگان مقاله و سال چاپ	نام الگوریتم استفاده شده	منبع	تعداد تکرار-Iteration		
			1000	5000	50000
Kuolinas et al., 2014	PSO-HH PSO based Hyper Heuristic	(Wang, et al., 2014)	35/20	32/59	31/23
Debels et al., 2006	Scatter Search	(Jozefowska, et al., 2006)	35/22	33/10	31/57
پژوهش حاضر	Shuffled Frog-leaping Algorithm	پژوهش حاضر	35/25	33/00	31/58
Wang et al., 2012	HEDA	(Zamani, 2013)	35/44	33/61	#
Sadeghi et al., 2011	Bees Algorithm	(sadeghi, et al., 2011)	35/80	33/33	31/55
Chen- 2011	JPSO-PSO	(Wang, et al., 2014)	35/71	33/88	32/89
Ziarari et al., 2011	BA-FBI	(Wang, et al., 2014)	37/72	36/76	34/55
Hartmann, 2002	GA-Activity List	(Kolisch, et al., 2002)	39/37	36/74	34/03

جدول شماره (۴): نتایج عملکرد الگوریتم و مقایسه آن در حل مسائل سری j120

الگوریتم پیشنهادی در حل این سری از مسائل نیز جایگاه سوم را به خود اختصاص داده است که در واقع میزان انحراف از حد پایین در ۱۰۰۰ بار تکرار ۳۵/۲۵٪ بوده است که عملکرد بسیار خوب این الگوریتم را در حل این دسته مسائل بزرگ RCPSP پایه را نمایان می سازد . در نهایت می توان بیان نمود که در این مقاله به بررسی عملکرد الگوریتم فراابتکاری قورباغه جهنده در حل مسائل RCPSP پایه پرداخته شده که نتایج حاکی از عملکرد قوی این الگوریتم در حل این دسته از مسائل بوده است در تحقیقات آتی می توان به حل مسائل دیگری از قبیل MRCPSP و MORCPSP با استفاده از این الگوریتم پرداخت تا بررسی های جامع تری در مورد عملکرد این الگوریتم در حل مسائل پیچیده تر RCPSP مشاهده نمود.

#### ۴- منابع

1. Demeulemeester W S., Herrolen W S.(2002). *Project Scheduling*. Kluwer Academic Publishers.
2. Brucker P. Drexl A. Mohring R. Neumann K. Pesch E. (1999). Resource-constrained project scheduling: Notation, classification, models, and methods. *European Journal of Operational Research*, 112, 3-41.
3. Sprecher A. (1997). Exact algorithm for RCPSP in multi-mode case. *OR Spectrum*, 19(3), 95-203.
4. Buddhakulsomsiria J., Kim D.S. (2006). Properties of multi-mode resource constrained project scheduling problems with resource vacations and activity splitting, *European Journal of Operational Research*, 175, 279-295.
5. Buddhakulsomsiria J. Kim D.S. (2007). Priority rule-based heuristic for multi-mode resource-constrained project scheduling problems with resource vacations and activity splitting. *European Journal of Operational Research*, 178,374-390.

6. Schultmann F. Rentz O. (2001). Environment-oriented project scheduling for the dismantling of buildings. *OR Spectrum*, 23(1), 51–78.
7. Demeulemeester E.L. de Reyck B. Herroelen W.S.(2000). The discrete time/resource trade-off problem in project networks – A branch-and-bound approach. *IIE Transactions*, 32,1059–1069.
8. Ranjbar M. Kianfar F. (2007). Solving the discrete time/resource trade-off problem in project scheduling with genetic algorithms. *Applied Mathematics and Computation*, 191, 451–456.
9. 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.
10. Akkan C., Drexel A., Kimms A., (2005). Network decomposition-based benchmark results for the discrete time-cost tradeoff problem, *European Journal of Operational Research*, 165, 339–358.
11. Maniezzo V. Mingozzi A.(1999). The project scheduling problem with irregular starting time costs. *Operations Research Letters*, 25(4), 175–182.
12. Mohring R.H. Schulz A.S. Stork F. Uetz M. (2001). On project scheduling with irregular starting time costs. *Operations Research Letters*, 28 (4): 149–154.
13. Wang Qiusheng, Autom. Sci. & Electr. Eng., Beihang Univ., Beijing. (2013). China Browse Conference Publications .Instrumentation, Measurement. A Modified Shuffled Frog Leaping Algorithm with Convergence of Update Process in Local Search,IEEE,1016 - 1019
14. Alghazi A.. Shokri Z. Selim. And Elazouni A. .(2012).Performance of Shuffled Frog-Leaping Algorithm in Finance-Based Scheduling, *Journal of Computing in Civil Engineering*, 396-408.
15. Eusuffa M. Lanseyb K. And Pashab F. (2006).Shuffled frog-leaping algorithm: a mimetic meta-heuristic for discrete optimization. *Engineering Optimization*, 38(2), 129-154.
16. Tavakolan M.(2011).Applying the Shuffled Frog-Leaping Algorithm to Improve Scheduling Of Construction Projects With Activity Splitting Allowed. Management and Innovation for a Sustainable Built Environment.
17. Afshar A. Kasaeian Ziaraty A. Kaveh A. Sharifi F. Nondominated .(2009).Archiving Multicolony Ant Algorithm in Time-Cost Trade-Off Optimization. *J. Constr. Eng. Mgmt.*, 135(7), 668-674.
18. Duan Q. Sorooshian S. and Gupta V. (1992).Effective and Efficient Global Optimization for Conceptual Rainfall Runoff models. *Journal of Water Resources Res*, 28,1015-1031.
19. Elbeltagi E. Hegazy T. and Grierson D. (2005).Comparison Among Five Evolutionary-based Optimization Algorithms, *Adv. Eng. Inf.*, 19 (1), 43-53.
20. Elbeltagi, E. (2016). A Modified Shuffled Frog Leaping Algorithm for Optimizing Bridge-Deck Repairs, *Int. Conference on Bridge Mgmt. Systems. Egypt*, Cairo, 1-10.
21. Eusuff M. M., Lansey K.E. (2003). Optimizing of Water Distribution Network Design Using the Shuffled Frog Leaping Algorithm, *J. Water Resource Plan Mgmt.*, 129(3), 210-225.

22. Zamani, R. (2013). A competitive magnet-based genetic algorithm for solving the resource-constrained project scheduling problem. *European Journal of Operational Research*, 229, 552-559.
23. Sadeghi A. Kalanaki A. Noktehdan A, Safi Samghabadi A. Barzinpour F.(2011).Using Bees Algorithm to Solve the Resource Constrained Project Scheduling Problem in PSPLIB, *Communications in Computer and Information Science*, 164:486-494.
24. Brucker P. Drexl A.Mohring R. Neumann K. Pesch E. (1999).Resource-constrained project scheduling: Notation, classification, models, and methods, *European Journal of Operational Research*: 3-41.
25. Wang A. Wang Xu A. GeXian-long B. Lei D. (2014).Multi-objective optimization model for multi-project scheduling on critical chain. *Advances in Engineering Software*, 31-48.
26. Afshar B., Nadjafi J., Rahimi A., Karimi H. (2013). *A genetic algorithm for mode identity and the resource constrained project scheduling problem*.
27. Alcaraz J.Maroto C. Ruiz R.(2003).Solving the multi-mode resource-constrained project scheduling problem with genetic algorithms. *Journal of Operation Research Society*, 54, 614-626.
28. Elloumi S. Fortemps P.(2012).A hybrid rank-based evolutionary algorithm applied to multi-mode resource-constrained project scheduling problem. *European Journal of Operational Research*, 205: 31-41.
29. Jarboui B. Damak N. Siarry P.Rebai A. (2012). A combinatorial particle swarm optimization for solving multi-mode resource-constrained project scheduling problems. *Applied Mathematics and Computation*, 195, 299-308.
30. Jozefowska J., Mika M., Rozycki R., Waligora G. (2006). Simulated annealing for multi-mode resource-constrained project scheduling. *Annals of Operations Research*, 102, 137-155.
31. Kolisch R. Drexl A. (2010).Local search for non-preemptive multi-mode resource-constrained project scheduling. *IIE Transactions* 29, 987-999.
32. Kolisch R. Hartmann S. (1999). *Heuristic algorithms for solving the resource-constrained project scheduling problem: classification. Computational, analysis*. Project Scheduling: Recent Models, Algorithms and Applications. Kluwer Press, Berlin, 147-178.
33. Kolisch R.(2002). Serial and parallel resource-constrained project scheduling methods revisited: theory and computation. *European Journal of Operational Research*, 90:320-333.