

## زمان بندی همزمان پروژه و برنامه ریزی تجهیزات با الگوریتم های تکاملی

عباس شفیعی خانی\* - امیر عباس نجفی\*\* - سید تقی اخوان نیکی\*\*\*

(تاریخ دریافت: ۹۰/۶/۲۰ - تاریخ پذیرش: ۹۱/۶/۲۲)

### چکیده

برنامه ریزی و زمان بندی پروژه یکی از مباحث بسیار مهمی است که مدیران پروژه با آن مواجه اند و از عوامل کلیدی در موفقیت یا عدم موفقیت پروژه محسوب می شود. تحقیقات انجام گرفته در زمان بندی پروژه نشان می دهد که به طور مرسوم در زمان بندی پروژه برنامه ریزی تجهیزات را نادیده گرفته و یا ابتدا فعالیت های پروژه زمان بندی و سپس تجهیزات بر اساس آن برنامه ریزی می شود. این کار سبب خارج شدن از بهینه زمان بندی می شود. در این مقاله با مد نظر قراردادن همزمان زمان بندی پروژه و برنامه ریزی تجهیزات، دو الگوریتم ژنتیک و شبیه سازی تبرید برای کمینه کردن هزینه ها، ارائه می شود. برای بررسی کارایی الگوریتم های پیشنهادی، مسائلی با تعداد فعالیت ها و تجهیزات مختلف مورد بررسی قرار گرفته و نتایج با هم مقایسه می شوند. نتایج نشان می دهد که الگوریتم های پیشنهادی قادرند تا مدل برنامه ریزی خطی عدد صحیح مختلط موجود را با تعداد گره های بالا و ضریب پیچیدگی های مختلف در زمانی کوتاه حل کنند.

کلمات کلیدی: زمان بندی پروژه، الگوریتم ژنتیک، شبیه سازی تبرید، برنامه ریزی

### تجهیزات

---

\* کارشناس ارشد مهندسی صنایع، دانشگاه آزاد اسلامی قزوین، دانشکده مهندسی صنایع و مکانیک، قزوین، ایران  
(نویسنده مسئول)  
abbas\_shafikhani22@yahoo.com

\*\* استادیار و عضو هیئت علمی گروه مهندسی صنایع، دانشکده مهندسی صنایع، دانشگاه صنعتی خواجه نصیرالدین

طوسی، تهران، ایران

\*\*\* استاد و عضو هیئت علمی گروه مهندسی صنایع، دانشکده مهندسی صنایع، دانشگاه صنعتی شریف، تهران، ایران

## مقدمه

برنامه‌ریزی و زمان‌بندی پروژه یکی از مباحث بسیار مهمی است که مدیران پروژه با آن مواجه‌اند و از عوامل کلیدی در موفقیت یا عدم موفقیت پروژه محسوب می‌شود. مسئله زمان‌بندی پروژه با محدودیت منابع کلی‌ترین مسئله زمان‌بندی بوده و مسائل زمان‌بندی کار کارگاهی<sup>۱</sup>، جریان کارگاهی<sup>۲</sup> و سایر مسائل زمان‌بندی زیرمجموعه‌ای از آن محسوب می‌شوند. (Yang, 2006؛ Lova, 2009؛ Tavakoli, 2007).

به‌طور کلی مسئله زمان‌بندی پروژه با محدودیت منابع به‌دست‌آوردن توالی مناسبی از فعالیت‌هاست طوری که محدودیت‌های تقدم و تاخر شبکه پروژه و سایر محدودیت‌های منابع موجود به‌طور هم‌زمان در نظر گرفته شده تا معیارهای سنجش معینی از جمله زمان و هزینه انجام پروژه یا سایر معیارها بهینه شوند. در واقع مسائل زمان‌بندی پروژه با محدودیت منابع یک مسئله بهینه‌یابی ترکیبی است که به لحاظ پیچیدگی در رده مسائل NP\_Hard دسته‌بندی می‌شوند (Peteghem and Vanhoucke, 2010؛ Soltani, 2010).

مسئله برنامه‌ریزی پروژه با منابع محدود برای نخستین بار در سال ۱۹۶۳ مطرح شد (Wiest, 1963؛ Jarboui, 2008). طی سال‌های اخیر پیشرفت‌های قابل توجهی در تدوین روش‌های کارآمد حل این مسئله صورت گرفته‌است. به‌طور کلی دو رویکرد برای حل این مسئله وجود دارد: روش‌های حل بهینه و دیگری روش‌های حل ابتکاری. روش‌های حل مبتنی بر رویکرد اول مورد توجه این مقاله نیستند. برای بررسی و مرور بر روش‌های برنامه ریاضی، به مراجع (Deckro, 1991؛ Carruthers and Battersby, 1996؛ Icmeli and Rom, 1996؛ Petrovic, 1968) و در خصوص روش‌های شاخه و کران می‌توان به (Herroelen, 2007) و (Demeulemeester؛ Brucker, 2004) مراجعه کرد. شایان ذکر است که در پروژه‌های بزرگ روش‌های حل بهینه کارایی لازم را ندارند (Herroelen, 2004). از این رو اکثر مسائل واقعی برنامه‌ریزی پروژه با منابع محدود به کمک روش‌های ابتکاری حل می‌شوند.

تحقیقات انجام گرفته در زمان‌بندی پروژه نشان می‌دهد که ابتدا فعالیت‌های پروژه زمان‌بندی شده و سپس متناسب با زمان‌بندی به‌دست‌آمده تجهیزات برنامه‌ریزی می‌شوند. این در حالی است که در دنیای واقعی به‌خاطر استفاده از تجهیزات، فعالیت‌ها می‌توانند فشرده شوند و این فشردگی لزوماً

1. Job Shop
2. Flow Shop

مربوط به فعالیت‌های بحرانی نخواهد بود و هم فعالیت‌های بحرانی و هم غیربحرانی را شامل خواهد شد (Dodin and 2008, Elimam). با توجه به اینکه هر روز استفاده بهتر از منابع سازمانی اعم از تجهیزات و ماشین‌آلات و نیروی انسانی و ... مورد توجه بیشتری قرار می‌گیرد محدودیت منابع برای برنامه‌ریزی برای دستیابی به اهداف قراردادها در پروژه‌ها و در رأس آن تعهدات زمانی و هزینه‌ای برای اتمام پروژه نقش بیشتری پیدا می‌کند. بنابراین همه هزینه‌های مربوط به زمان‌بندی پروژه و برنامه‌ریزی تجهیزات باید به صورت یکپارچه در مدل در نظر گرفته شوند تا زمان‌بندی به دست آمده بهینه باشد. سروکار داشتن با متغیرهای متعدد در این مسئله هم در بحث تئوری و هم در بحث عملی بسیار پیچیده و مشکل است. در مقالات و کتب اخیر شامل دمیولمستر و هرولن (Herroelen, 2002) و Demeulemeester and هرولن (1997؛ 1998)، Herroelen، المغربی (Elmaghraby, 2003)، ازمار و السوی (Ozdamar and Ulusoy, 1995) اسملی (Icmeli, 2003) این مطلب را می‌توان ملاحظه کرد. پس از تحقیق در کارهای انجام شده مشخص شد که فقط دودین و الیمام مدل یکپارچه زمان‌بندی پروژه و برنامه‌ریزی تجهیزات را در نظر گرفته‌اند. مدل ارائه شده توسط آنان یک مدل برنامه‌ریزی خطی عدد صحیح مختلط بوده و از لحاظ رده پیچیدگی NP-hard تلقی می‌شود و پیچیدگی آن با افزایش تعداد فعالیت‌ها به شکل معناداری افزایش می‌یابد (Dodin and Elimam, 2008). ولی روش ابتکاری ارائه شده توسط آن‌ها برای شبکه‌های بزرگ کارا نبوده و به مدت زمان زیادی برای حل نیاز دارد.

بنابراین در این مقاله سعی بر توسعه دو الگوریتم ژنتیک و شبیه‌سازی تبرید برای حل مسائل با ابعاد بزرگ و درجه پیچیدگی بالا در مدت زمان بسیار کم را داریم. هزینه‌هایی که در این مدل وجود دارد شامل هزینه‌های فشرده‌سازی فعالیت‌ها، نگهداری فعالیت‌ها، پاداش برای زودکرد و جریمه برای دیرکرد، حمل و نقل تجهیزات از محل یک فعالیت به محل فعالیت دیگر، آماده‌سازی تجهیزات، بیکاری تجهیزات و اضافه‌کاری اپراتورهاست. همچنین تجهیزات موردنظر در این تحقیق از نوع منابع تجدیدپذیر (Dodin and Elimam, 2008؛ Patterson and Roth, 1976؛ Dodin and Elimam, 1998) هستند که در دوره‌های متناوب قابل دسترس‌اند، یعنی، مقدار این منابع از دوره‌ای به دوره بعد تجدید می‌شود و کل استفاده از این منابع تنها در هر لحظه زمانی محدود است. نیروی انسانی، ماشین‌آلات، ابزارآلات، تجهیزات و ... نمونه‌هایی از منابع تجدیدپذیرند. این منابع در هر دوره به میزان محدود و معینی در دسترس‌اند و در دوره بعدی نیز این میزان دسترس‌پذیری تجدید می‌شود (Ozdamar and ulusoy, 1995؛ Icmeli, 1993).

ساختار این مقاله بدین صورت است که در بخش ۲ تعریف و مدل‌سازی مسئله به همراه فرض‌های در نظر گرفته شده ارائه می‌شود. در بخش ۳ به الگوریتم‌های حل پیشنهادی پرداخته شده و سپس مراحل لازم برای استفاده از الگوریتم‌های پیشنهادی در حل مسئله زمان‌بندی همزمان پروژه و برنامه‌ریزی تجهیزات تشریح می‌شود. بخش ۴ به نتایج محاسباتی و نحوه تولید مسائل آزمایشی اختصاص دارد. در بخش ۵ نیز نتیجه‌گیری و پیشنهادهای برای مطالعات آتی مطرح می‌شود.

## تعریف و مدل‌سازی مسئله

### فرضیات

شبکه پروژه به صورت گرهی  $G(N,A)$  تعریف شده است، به طوری که مجموعه گره‌های شبکه ( $N$ ) بیانگر فعالیت‌ها و مجموعه بردارهای شبکه ( $A$ ) بیانگر روابط پیش‌نیازی از نوع خاتمه به شروع با حداقل تأخیر زمانی برابر با صفرند. فعالیت‌های شماره ۱ و  $N$  مجازی بوده و به ترتیب شروع و اتمام پروژه را نشان می‌دهند.

بریدگی فعالیت‌ها مجاز نیست.

تجهیزات در همه زمان‌ها در دسترس‌اند.

امکان اضافه کاری علاوه بر ساعات عادی کار وجود دارد.

یک تجهیز در یک زمان نمی‌تواند بیش از یک فعالیت را خدمت‌دهی کند.

چنانچه یک فعالیت به چند تجهیز نیاز داشته باشد باید همه آن تجهیزات را در زمان اجرا در اختیار داشته باشد.

امکان فشرده‌سازی فعالیت‌ها وجود دارد.

با توجه به اینکه ممکن است تعدادی از فعالیت‌ها به تجهیز  $i$ ام احتیاج نداشته باشند، این تجهیز می‌تواند بازگردانده شود، یا بی‌کار نگاه داشته شود تا زمانی که مجدداً به آن احتیاج پیدا شود یا اینکه با صرف هزینه، فعالیت‌هایی که به آن تجهیز احتیاج ندارند را فشرده نموده تا هزینه بیکاری تجهیز کم شود.

## گزینه‌های حل مسئله زمان‌بندی همزمان پروژه و برنامه‌ریزی تجهیزات

پروژه نیازمند به اجاره تجهیزات خاصی است که روی تعدادی از فعالیت‌های در پروژه کار می‌کنند. این تجهیز روی یک یا تعدادی از فعالیت‌ها کار می‌کند و سپس بی‌کار می‌ماند

(درحالی‌که برای آن پول پرداخت می‌شود) تا زمانی‌که سایر فعالیت‌ها آماده برای پردازش در پروده‌های دیگر شوند. در نتیجه این تجهیز می‌تواند یا بازگردانده شود یا بیکار نگاه داشته شود تا زمانی‌که مجدداً به آن احتیاج پیدا شود.

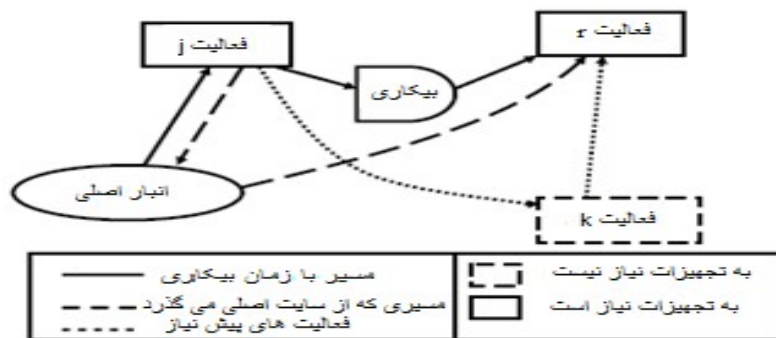
برای زمان‌بندی تجهیزات جهت انجام فعالیت گزینه‌های متعددی وجود دارد. شکل ۱ بیانگر این گزینه‌هاست.

گزینه ۱: ابتدا تجهیز مسیر از انبار اصلی تا سایت را می‌پیماید و فعالیت موردنظر را انجام می‌دهد. سپس بیکار مانده تا فعالیت K که به تجهیز نیازی ندارد انجام شود.

سپس مسیر از سایت J به R را می‌پیماید. در نتیجه آماده‌سازی<sup>۱</sup> یک‌بار اتفاق می‌افتد. ولی استفاده از تجهیزات کم بوده و ممکن است هزینه بیکاری قابل ملاحظه باشد. بنابراین هزینه‌ها شامل هزینه بیکاری و هزینه آماده‌سازی و هزینه حمل از سایت J به R است.

گزینه ۲: ابتدا تجهیز مسیر از انبار اصلی تا سایت را می‌پیماید و فعالیت مورد نظر را انجام می‌دهد. سپس مجدداً به سایت اصلی بازگردانده می‌شود و تا زمانی‌که فعالیت K به اتمام برسد در آنجا باقی مانده و سپس مجدداً از سایت اصلی به سراغ فعالیت R می‌رود. در نتیجه آماده سازی دوبار اتفاق می‌افتد، بی‌کاری نداریم و سیاست شبیه به سیاست Lot For Lot داریم.

گزینه ۳: پیرو حالت A1 درحالی‌که فعالیت K را فشرده<sup>۲</sup> می‌سازیم، زمان بیکاری ماشین‌آلات را کم می‌کنیم تا زمان پروژه کاهش پیدا کند. بنابراین فرقی نمی‌کند که فعالیت بحرانی یا غیر بحرانی باشد. در نتیجه هزینه بیکاری آماده‌سازی کاهش می‌یابد ولی هزینه فشرده‌سازی افزایش می‌یابد. بنابراین باید بالانسی را بین هزینه‌های فشرده‌سازی و بیکاری و آماده‌سازی برقرار کنیم.



شکل ۱ - گزینه‌های حل مسئله زمان‌بندی همزمان پروژه و برنامه‌ریزی تجهیزات

1. Setup
2. Crash

معرفی هزینه‌های موجود در مسئله زمان‌بندی همزمان پروژه و برنامه‌ریزی تجهیزات

هزینه فشردگی فعالیت‌ها<sup>۱</sup> :

این هزینه پیوسته و خطی در طول مدت زمان انجام فعالیت است (هم برای فعالیت‌های بحرانی و هم برای غیربحرانی)

هزینه نگهداری<sup>۲</sup> :

به‌عنوان درصدی از پروژه و در انتهای هر پریود محاسبه می‌شود.

پاداش و جریمه<sup>۳</sup> :

این واقعیت که یک تجهیز نمی‌تواند بیشتر از یک فعالیت را در یک زمان انجام دهد منجر به اتمام پروژه در آن‌سوی موعد تحویل<sup>۴</sup> می‌شود. از این‌رو جریمه یا پاداش می‌بندیم که سبب فشردگی و کمتر کردن بیکاری و کاهش تعداد دفعات آماده‌سازی شویم.

هزینه حمل<sup>۵</sup> :

هزینه حرکت فعالیت از محل یک سایت فعالیت به فعالیت دیگر (شبه هزینه حرکت از یک شهر به شهر دیگرست). (فروشنده دوره گرد)<sup>۶</sup>

هزینه آماده‌سازی<sup>۷</sup> :

هزینه آماده‌سازی و حمل تجهیزات برای فعالیت‌ها و همچنین بازگشت به محل شروع یا همان انبار اصلی است)

هزینه زمان بیکاری<sup>۸</sup> :

هزینه‌ای است که به‌واسطه بیکار نگاه داشتن تجهیز به صورت بلااستفاده در سایت است.

هزینه اضافه‌کاری<sup>۹</sup> :

هزینه تجهیزات به صورت روزانه شارژ می‌شود.

1. Crashing Cost
2. Holding Cost
3. Project Penalty Or Reward
4. Due Date
5. Transition Cost
6. Tsp
7. Set Up Cost
8. Idle Time Cost
9. Over Time Cost

هزینه اپراتورها در هر شیفت اتفاق می‌افتد.

هزینه اضافه‌کاری اپراتورها ساعتی اتفاق می‌افتد.

همان‌گونه که در شکل ۱ توضیح داده شد، یک تجهیز دو نوع حرکت دارد. ابتدا از محل اصلی تجهیزات به سوی فعالیت‌ها حرکت کرده و خدمت‌دهی می‌کند و در نهایت به محل خود باز خواهد گشت. حرکت دوم، حرکت از محل یک فعالیت به فعالیت دیگرست که ممکن است مطابق گزینه دوم که در بخش ۲,۲ ارائه شد ابتدا تجهیز مسیر از انبار اصلی تا سایت را می‌پیماید و فعالیت موردنظر را انجام می‌دهد. سپس مجدداً به سایت اصلی باز گردانده می‌شود و تا زمانی که فعالیت K به اتمام برسد در آنجا باقی مانده و سپس مجدداً از سایت اصلی به سراغ فعالیت R می‌رود. بنابراین سایت اصلی نقش Hub را خواهد داشت. لذا دو فعالیت مجازی  $N_1$  و  $N_2$  به فعالیت‌های پروژه اضافه خواهد شد، که  $N_1$  معرف سایت اصلی تجهیزات و  $N_2$  معرف Hub در حرکت تجهیز از یک فعالیت به فعالیت دیگرست. بنابراین فعالیت مجازی  $N_2$  تنها فعالیتی خواهد بود که تجهیز می‌تواند از آن بیش از یک مرتبه عبور کند.

### مدل ریاضی

هدف حداقل کردن هزینه‌های مسئله زمان‌بندی همزمان پروژه و برنامه‌ریزی تجهیزات به وسیله تعیین مقدار بهینه مدت زمان فعالیت‌ها، زمان اتمام فعالیت‌ها و زمان‌بندی تجهیزات با در نظر گرفتن محدودیت‌هاست.

### اندیس‌ها

$j = 1, \dots, N$	شاخص فعالیت‌های پروژه
$i = 1, \dots, M$	شاخص تجهیزات مورد استفاده در پروژه
$t = 0, \dots, H$	شاخص زمان

### پارامترهای مربوط به فعالیت

مجموعه فعالیت‌هایی که پیش‌نیاز فوری فعالیت  $j$  هستند، که در آن  $B_j$

$B_o$  مجموعه‌ای است که شامل فعالیت‌های بدون پیش‌نیاز است

$b_j$	هزینه فعالیت $j$ در حالت فشرده (واحد پولی)
$c_j$	هزینه هر واحد فشرده‌سازی فعالیت $j$ (در یکک پرپود)
$u_j$	حد بالای زمانی فعالیت $j$ (زمان نرمال فعالیت)
$v_j$	حد پایین زمانی فعالیت $j$ (زمان فشرده فعالیت)
$e_j$	زودترین زمان تکمیل فعالیت $j$ (با فرض استفاده از زمان فشرده و شروع در زمان صفر)
$l_j$	دیرترین زمان تکمیل فعالیت $j$ (با فرض استفاده از زمان فشرده و اینکه پروژه در $H$ تکمیل شود)
$A_{ij}$	مجموعه فعالیت‌هایی که نیازمند تجهیز $i$ هستند و می‌توانند بعد از فعالیت $j$ شروع شوند.
$EA_{ij}$	مجموعه فعالیت‌هایی که می‌توانند پس از فعالیت $j$ و یا به موازات فعالیت $j$ شروع شوند و نیازمند استفاده از تجهیز $i$ هستند.
$EB_{ij}$	مجموعه فعالیت‌هایی که می‌توانند پیش از فعالیت $j$ و یا در طول فعالیت $j$ شروع شوند و نیازمند استفاده از تجهیز $i$ هستند.
$J_t$	مجموعه فعالیت‌هایی که می‌توانند در پرپود زمانی $t$ تکمیل شوند.

### پارامترهای مربوط به تجهیزات

$AE_i$	مجموعه فعالیت‌هایی که نیازمند استفاده از تجهیز $i$ هستند
$C_i$	هزینه بیکاری تجهیز $i$ (واحد پولی بر دوره زمانی) که فرض می‌شود دوره زمانی روزانه است.
$CO_i$	هزینه اضافه‌کاری کارکنانی که روی تجهیز بیش از شیفت استاندارد کار می‌کنند (واحد پولی بر دوره زمانی)
$EE_i$	مجموعه فعالیت‌هایی که توسط تجهیز $i$ خدمت‌دهی شده‌اند.
$G_i$	کل هزینه ستاپ رفت و برگشت تجهیز $i$ از انبار اصلی به سایت پروژه
$ru_i$	نرخ (آهن‌گ) استفاده از تجهیز $i$ (ساعت بر شیفت)
$SE_i$	مجموعه فعالیت‌هایی که توسط تجهیز $i$ می‌توانند خدمت‌دهی شوند



$S_{ijk}$  هزینه حمل‌ونقل تجهیز نام که بدون عبور از hub از فعالیت  $j$  به فعالیت  $k$  می‌رود (واحد پولی)

$a_{ij}$  میزان زمان اضافه‌کاری که برای عملیات تجهیز  $i$  روی فعالیت  $j$  نیازست طوری که فعالیت در مدت زمان فشرده به اتمام برسد (فرض می‌شود که محتوای کار فعالیت ثابت بماند).

### پارامترهای مربوط به پروژه

$d$  موعد تحویل پروژه

$H$  ماکزیمم طول افق برنامه‌ریزی پروژه ( $H > d$ )

$p$  هزینه جریمه به ازای هر پریودی که از موعد تحویل پروژه بیشتر شود (واحد پولی بر دوره زمانی)

$r$  پاداش پرداخت‌شده برای تکمیل پروژه قبل از موعد تحویل (واحد پولی بر دوره زمانی)

$s$  بهای فعالیت تکمیل‌شده که با هزینه نگهداری نمایش داده می‌شود (درصدی از دوره زمانی).

### متغیرهای تصمیم

$\delta_{jt} \begin{cases} 1 \\ 0 \end{cases}$  اگر فعالیت  $j$  در دوره  $t$  تکمیل شده باشد  
در غیر این صورت

$Y_{ijk} \begin{cases} 1 \\ 0 \end{cases}$  اگر تجهیز نام فعالیت  $k$  را پس از فعالیت  $j$  خدمت دهد  
در غیر این صورت

$IM_{ijk}$  زمان بیکاری تجهیز نام اگر به‌خاطر انجام فعالیت  $k$  منتظر اتمام فعالیت  $j$  شود.

$TO_i$  کل زمان اضافه‌کاری مورد نیاز برای تجهیز نوع  $i$

$ot_{ij}$  نیروی انسانی مورد نیاز برای اضافه‌کاری تجهیز نوع  $i$  روی فعالیت  $j$

$W_t$  بهای پروژه در انتهای پریود  $t$

$W_{jt}$  بهای فعالیت  $j$  که در پریود  $t$  تکمیل شده‌است (واحد پولی). هزینه واقعی فعالیت  $j$  شامل منابع تجدیدپذیر و تجدیدنپذیرست.

$X_j$  زمان تکمیل فعالیت  $j$   
 $Z_j$  مدت زمان فعالیت  $j$

با توجه به متغیرها و پارامترهای تعریف شده مدل ریاضی مسئله به صورت زیر است.

$$\text{Minimize } \left[ \sum_{j=1}^N [b_j - c_j(z_j - v_j)] + \sum_{i=1}^M G_i \left( \sum_{j \in SE_i} Y_{iN1j} + \sum_{j \in AE_i} Y_{iN2j} \right) \right] + \sum_{i=1}^M \sum_{j \in AE_i} \sum_{k \in A_{ij}} S_{ijk} Y_{ijk}$$

$$+ \sum_{i=1}^M \sum_{j \in AE_i} \sum_{k \in A_{ij}} CI_i IM_{ijk} + \sum_{i=1}^M co_i To_i + \sum_{t=1}^{H-1} sW_t - \sum_{t=e_N}^{d-1} r(d-t)\delta_{Nt} + \sum_{t=d+1}^{l_n} p(t-d)\delta_{Nt}$$

Subject to: (۱)

$$X_j - X_k \geq Z_j \quad \text{for each } k \in B_j \quad (۲)$$

$$X_j \geq Z_j \quad \text{for each } j \in B_0$$

$$Z_j \leq u_j \quad \text{for } j=1, \dots, N \quad (۳)$$

$$Z_j \geq v_j \quad \text{for } j=1, \dots, N$$

$$X_j \leq L_j \quad \text{for } j=1, \dots, N \quad (۴)$$

$$X_j \geq e_j \quad \text{for } j=1, \dots, N$$

$$\sum_{t=e_j}^{l_j} \delta_{jt} = 1 \quad \text{for } j=1, \dots, N \quad (۵)$$

$$\sum_{t=e_j}^{l_j} t \cdot \delta_{jt} \geq X_j \quad \text{for } j=1, \dots, N$$

$$\sum_{j \in SE_i} Y_{iN1j} = 1 \quad \text{for } j=1, \dots, N$$

$$\sum_{j \in EE_i} Y_{ijN1} = 1 \quad \text{for } j=1, \dots, N$$

$$\sum_{k \in EB_j} Y_{ijk} - \sum_{k \in EA_j} Y_{ijk} = 0 \quad j \in AE_i \cup N_2 \quad (۶)$$

$$\sum_{k \in EA_j} Y_{ijk} = 1 \quad \text{for } i=1, \dots, M \text{ and } j \in AE_i$$

$$\sum_{k \in EB_k} Y_{ijk} = 1 \quad \text{for } i=1, \dots, M \text{ and } k \in AE_i$$

$$IM_{ijk} \geq X_k - X_j - Z_k - H(1 - Y_{ijk}) \quad \& \quad (v)$$

$$IM_{ijk} \geq 0 \quad \text{for } i=1, 2, \dots, M, j \in AE_i, k \in A_{ij}$$

$$ot_{ij} \geq a_{ij} - ru_i(Z_j - v_j) \quad \text{for } i=1, 2, \dots, M, j \in AE_i \quad (\lambda)$$

$$ot_{ij} \geq 0$$

$$O\Gamma_i = \sum_{j \in AE_i} ot_{ij}$$

$$W_{jt} \geq (b_j - c_j(Z_j - v_j)) - \left( \sum_{j=1}^N b_j \right) (1 - \delta_{jt}) \quad (9)$$

$$\text{for } j=1, 2, \dots, N \quad \& \quad t \in [e_j, l_j]$$

$$W_t \geq W_{t-1} + \sum_{j \in I_t} W_{jt} \quad \text{for } t=1, 2, \dots, e_N$$

$$W_t \geq W_{t-1} + \sum_{j \in I_t} W_{jt} - \left( \sum_{j=1}^N b_j \right) \delta_{Nt} \quad \text{for } t=e_N+1, \dots, H$$

$$W_t \geq 0 \quad \text{for } t=1, 2, \dots, H$$

$$W_0 = 0$$

تابع هزینه کل (۱) شامل هشت هزینه است. این هزینه‌ها به ترتیب شامل هزینه فشرده‌سازی فعالیت، هزینه آماده‌سازی برای حمل تجهیزات، هزینه جابجایی تجهیزات از محل یک فعالیت به فعالیت دیگر، هزینه بیکاری تجهیزات، هزینه اضافه‌کاری کارکنانی که روی تجهیزات کار می‌کنند، هزینه نگهداری، هزینه پاداش برای زودکرد و جریمه برای دیرکرد است.

محدودیت (۲) روابط پیش‌نیازی است. محدودیت (۳) مربوط به مدت زمان فعالیت‌هاست که باید بین زمان‌های نرمال و فشرده باشد. محدودیت (۴) محدودیت بازه زمانی اتمام فعالیت‌هاست. محدودیت (۵) مربوط به زمان اتمام فعالیت بوده و تضمین می‌کند که هر فعالیت باید

در بازه زمانی [e,z], [j,z] به اتمام برسد. محدودیت (۶) مرتبط با حمل تجهیزات بوده که به ترتیب شامل این موضوع بوده که اولاً هر تجهیز کار خود را روی پروژه آغاز خواهد کرد. ثانیاً هر تجهیز باید پروژه را ترک کند و ثالثاً هر تجهیز روی یک فعالیت کار نموده و سپس آنرا به منظور خدمت‌دهی به فعالیت دیگری که بدان نیاز دارد ترک نموده و یا به محل اصلی خود بازمی‌گردد. در ضمن برای همه فعالیت‌هایی که نیازمند تجهیز i هستند، تجهیز فقط یک بار به آن فعالیت وارد شده و از آن خارج می‌شود. محدودیت‌های (۷) مرتبط با بیکاری تجهیزات است و نشان‌دهنده بیکاری تجهیز i بوده که منتظر خدمت‌دهی به فعالیت k پس از تکمیل فعالیت j است. محدودیت (۸) مربوط به اضافه کاری اپراتورهاست که مبین اضافه کاری مورد نیاز برای تجهیزات در ساعات اضافه کاری است. محدودیت (۹) مرتبط با هزینه نگهداری است که در آن  $J_i$  مجموعه فعالیت‌هایی است که می‌توانند در زمان t تکمیل شوند.

### الگوریتم‌های حل پیشنهادی

#### الگوریتم ژنتیک

الگوریتم ژنتیک روش جستجوی احتمالی است که بر پایه عملکرد انتخاب ژنتیک طبیعی بنا شده است. الگوریتم ژنتیک برخلاف سایر روش‌های جستجو، با یک مجموعه ابتدایی از جواب‌های تصادفی که جمعیت اولیه<sup>۱</sup> نامیده می‌شود، آغاز می‌شود. هر عضو در جمعیت، یک کروموزوم<sup>۲</sup> نامیده می‌شود که نمایانگر یک حل برای مسئله موجود است. یک کروموزوم رشته‌ای از علائم است که در بردارنده پارامترهای مسئله است. طی هر تکرار الگوریتم ژنتیک، مجموعه جدیدی از کروموزوم‌ها تولید می‌شود. جمعیت در زمان معلوم را نسل<sup>۳</sup> می‌نامند. در هر نسل قابلیت کروموزوم‌ها توسط تابع تطابق<sup>۴</sup> که یک کروموزوم را با توجه توجه به تابع هدف مسئله برآورد می‌کند، تعیین می‌شود. انتخاب کروموزوم‌ها برای تولید کروموزوم‌های نسل بعد براساس میزان تطابق آن‌ها که به وسیله تابع تطابق نرمالیزه شده<sup>۵</sup>، تعیین می‌شود، صورت می‌پذیرد. پس از انتخاب کروموزوم‌ها، طی فرایند بازتولید<sup>۶</sup> عملگرهای

1. Initial population
2. chromosome
3. Generation
4. Fitness function
5. Scaling fitness function
6. reproduction

الگوریتم ژنتیک روی آن‌ها اعمال می‌شوند. این عملگرها عبارتند از: عملگر تقاطعی<sup>۱</sup> و عملگر جهشی<sup>۲</sup>. به کروموزوم‌هایی که از این طریق تولید می‌شوند فرزند<sup>۳</sup> اطلاق می‌شود. سپس تطابق فرزندان ارزیابی شده و توسط یکی از رویه‌های انتخاب کروموزوم‌های بهتر انتخاب و به نسل بعد منتقل می‌شوند.

هر تکرار این روند یک نسل را ایجاد می‌کند. تعداد نسل‌ها به دلخواه تعیین می‌شود. در این فرایند الگوریتم به بهترین کروموزوم همگرا می‌شود که نمایانگر جواب بهینه مسئله است. الگوریتم زمانی متوقف می‌شود که یکی از معیارهای توقف برآورده شود.

### مشخصات کروموزوم

اندازه کروموزوم: کروموزوم یک ماتریس دو سطری با اندازه ستون به تعداد فعالیت‌هاست. به طوری که سطر اول نشان‌دهنده زمان انجام فعالیت و سطر دوم زمان پایان فعالیت است. در نسل اولیه به صورت تصادفی زمان‌های انجام فعالیت از بازه انتخاب می‌شوند. حد پایین کروموزوم: یک ماتریس دو سطری است که سطر اول حاوی حد پایین برای زمان انجام فعالیت‌ها (زمان فشرده) و سطر دوم حاوی حد پایین برای زمان خاتمه فعالیت‌ها<sup>۴</sup> است. زمان‌بندی ect براساس استفاده از زمان‌های فشرده (vj) بوده و زمان‌بندی از لحظه صفر شروع شده است.

حد بالای کروموزوم: یک ماتریس دو سطری است که سطر اول حاوی حد بالا برای زمان انجام فعالیت‌ها (زمان نرمال) و سطر دوم حاوی حد بالا برای زمان خاتمه فعالیت‌ها<sup>۵</sup> است. زمان‌بندی lct براساس استفاده از زمان‌های فشرده vj و اتمام زمان پروژه در زمان H محاسبه می‌شود.

استراتژی تقاطع از نوع تک نقطه‌ای است

استراتژی جهش از نوع چند نقطه‌ای تصادفی است. حداکثر تعداد نقاط جهش جزء صحیح تقسیم تعداد فعالیت‌ها بر عدد ۵ به دست می‌آید. مثلاً اگر ۱۰ فعالیت داشته باشیم حداکثر تعداد نقاط جهش برابر دو است.

1. Crossover operator
2. Mutation operator
3. child
4. Early completion times
5. late completion times

در نسل اول به صورت تصادفی کروموزم تولید می‌کنیم و برای نسل‌های بعدی با استفاده از پارامترهای ژنتیک نسل جدید ایجاد خواهیم کرد. به هر کروموزم یک رتبه rank و یک درجه شایستگی fitness اختصاص داده می‌شود.

### الگوریتم برنامه

۱- پارامترهای ورودی ساختار مسئله و تنظیمات ژنتیک را بخوان  
 ۲- زمان‌بندی فعلی را براساس زمان‌بندی ect قرار بده  
 ۳- از نسل یک تا آخر کارهای زیر را انجام بده  
 زمان آغاز حل این نسل را ذخیره کن  
 اگر نسل اول بود (اولین اجرای حلقه) نسل اولیه تصادفی ایجاد کن و اگر نسل‌های بعدی بود با استفاده از عملگرهای ژنتیک نسل ایجاد کن.  
 برای تمام کروموزم‌هایی که در این نسل شایستگی به اندازه کامل n ندارند کارهای زیر را انجام بده

کروموزم را در تابع ارزیابی، مورد ارزیابی قرار بده  
 شایستگی کروموزم را از نظر زمان ضرب‌العجل بررسی کن.  
 اگر زمان خاتمه این زمان‌بندی بیشتر از ضرب‌العجل (H) بود شایستگی کروموزم را برابر صفر قرار بده در غیر این صورت شایستگی کروموزم را برابر n قرار بده.  
 برگرد به اول حلقه برای کروموزم بعدی بدون شایستگی در این نسل. در صورت اتمام حلقه خارج شو.

رتبه این نسل از کروموزم‌ها را به این ترتیب محاسبه کن:  
 برای همه کروموزم‌ها در این نسل مقدار تابع هدف را بر شایستگی تقسیم کن و سپس نتیجه را به صورت افزایشی مرتب کن. با توجه به حلقه بالا شایستگی یا ۰ یا n است در این صورت کروموزم‌هایی که شایستگی ندارند بدون توجه به مقدار تابع هدف در رتبه آخر قرار می‌گیرند (تقسیم بر صفر یعنی بینهایت). کروموزم‌هایی که شایستگی دارند با توجه به مقدار تابع هدف به صورت افزایشی مرتب می‌شوند.

بهترین کروموزم و زمان حل این نسل را ذخیره کن.  
 برگرد به اول حلقه برای محاسبه نسل بعدی. در صورت اتمام حلقه خارج شو.

با استفاده از بهترین کروموزم بهترین زمان‌بندی را محاسبه کن و به‌عنوان خروجی تابع قرار بده

### پایان الگوریتم

#### الگوریتم شبیه‌سازی تبرید

در این روش کل مسئله به صورت یک صفحه شبیه‌سازی می‌شود که هر یک از جواب‌ها یک مولکول در آن صفحه هستند. ابتدا یک نقطه تصادفی در صفحه به‌عنوان جواب اولیه انتخاب می‌شود. سپس یک همسایه با روشی مخصوص مسئله که بهترین جواب را حاصل کند تولید می‌شود. این همسایه اگر جواب بهتری از جواب اولیه حاصل کند به‌عنوان جواب جدید انتخاب می‌شود. در غیر این صورت هنوز هم با احتمالی امکان انتخاب شدن دارد. حسن این روش همین احتمال است که باعث می‌شود جواب‌های بد نیز قبول شوند تا از رسیدن به نقطه مینیمم محلی جلوگیری شود.

ابتدا صفحه تا درجه حرارت حداکثر حرارت داده می‌شود و مولکول‌ها (جواب‌ها) حرکت زیادتری دارند (بالاترین احتمال انتخاب). سپس به تدریج در قدم‌هایی صفحه سرد شده (احتمال انتخاب کم می‌شود) تا به کمترین حرکت مولکول‌ها برسیم. این فرایند مدام در زنجیره‌ای که به نام مارکوف معروف است تکرار می‌شود تا بهترین جواب حاصل شود. به این زنجیره‌های گرم و سرد کردن زنجیره مارکوف گفته می‌شود. جواب اولیه در اینجا همانند GA تولید می‌شوند.

### الگوریتم

ورودی‌های برنامه را بگیر:

جواب اولیه را به‌عنوان بهترین جواب و جواب فعلی را ذخیره کن

به تعداد زنجیره‌های مارکوف کارهای زیر را انجام بده

درجه حرارت فعلی را مساوی با درجه حرارت اولیه حداکثر قرار بده.  $T=T_0\max$

تعداد همسایه‌ها را برابر با تعداد همسایه‌ها در قدم ۰ قرار بده.  $N_s=N_0$

به تعداد حداکثر قدم‌ها کارهای زیر را انجام بده

به تعداد  $N_s$  کارهای زیر را انجام بده

یک همسایه با روش ابتکاری زیر درست کن  
 به طور تصادفی تعدادی از فعالیت‌ها را انتخاب کن طوری که با افزایش همسایه‌ها تعداد  
 آن‌ها کاهش یابد.

یک بازه از این فعالیت‌ها در نظر بگیر  
 به صورت تصادفی (رندوم) زمان‌های اجرای فعالیت‌ها و زمان‌های پایان فعالیت‌ها را تغییر بده  
 جواب به دست آمده را در تابع ارزیابی، مورد ارزیابی قرار بده  
 شایستگی جواب شکست را از نظر زمان ضرب العجل بررسی کن. اگر زمان خاتمه این  
 زمان بندی بیشتر از ضرب العجل بود مقدار تابع هدف جواب شکست را برابر با بی نهایت قرار  
 بده تا در احتمال انتخاب همسایه انتخاب نشود.

احتمال انتخاب همسایه را بررسی کن  
 $۳,۳,۱,۷,۱$  ابتدا اختلاف مقدار تابع هدف جواب اصلی و همسایه جدید را محاسبه کن (Delta)

$۳,۳,۱,۷,۲$  اگر  $\Delta < 0$  (یعنی همسایه بهتر بود)  
 $۳,۳,۱,۷,۲,۱$  همسایه را به عنوان جواب فعلی قرار بده  
 $۳,۳,۱,۷,۲,۲$  اگر همسایه از بهترین جواب بهتر بود همسایه را به عنوان بهترین جواب قرار  
 بده

$۳,۳,۱,۷,۳$  اگر  $\Delta > 0$  (اگر همسایه بدتر بود)  
 $۳,۳,۱,۷,۳,۱$  احتمال انتخاب همسایه  $P$  را از رابطه  $P = e^{(-\Delta/T)}$  محاسبه کن  
 $۳,۳,۱,۷,۳,۱,۱$  اگر  $P < Y$  (Y یک عدد تصادفی بین ۰ و ۱ است) همسایه بدتر را به  
 عنوان جواب فعلی قرار بده

اگر همسایه انتخاب شد آن را به عنوان جواب فعلی قرار بده  
 اگر همسایه بهترین جواب بود آن را به عنوان بهترین جواب قرار بده  
 خاتمه همسایه‌ها  $N_s$   
 درجه حرارت را طبق فرمول  $T = \alpha * T$  کاهش بده  
 خاتمه قدم‌ها Step  
 زمان حل این زنجیر مارکوف را ذخیره کن  
 بهترین نقطه در این زنجیر را پیدا کن.



در صورتی که زمان اجرا از حد زمان اجرای نهایی عبور کرد (شرط توقف) خارج شو و گرنه به زنجیره بعدی برو. (بازگشت به ابتدای حلقه)  
خاتمه زنجیره مارکوف Chain  
بهترین نقطه را به عنوان جواب برگشت بده  
پایان الگوریتم

### مشترکات دو الگوریتم

#### تابع زمان‌بندی

این تابع دو سطر یک برنامه زمان‌بندی را ارزیابی کرده و فعالیتی که کمترین زمان شروع را دارد انتخاب می‌کند و با شرایط مسئله تطبیق می‌دهد. در صورتی که در هر مرحله برنامه زمان‌بندی مطابق شرایط نباشد آن را با توجه به شرایط و محدودیت‌ها اصلاح کرده و زمان‌بندی انجام‌شدنی را به عنوان خروجی ارائه می‌دهد.  
ورودی:

برنامه زمان‌بندی ارزیابی نشده (مثلاً در اینجا کروموزم به دست آمده از ژنتیک را مورد بررسی قرار خواهیم داد).  
خروجی:

برنامه زمان‌بندی ارزیابی شده و انجام‌شدنی

#### الگوریتم برنامه

بردار اولویت‌های زمان شروع را با کم کردن سطر دوم کروموزم از سطر اول کروموزم به ترتیب مقابل محاسبه کن. ( $\text{prior\_start} = \text{finish time} - \text{process time}$ )  
برنامه زمان‌بندی انجام‌شدنی تهی ایجاد کن (چون فعالیت اول همیشه ابتدا وارد می‌شود می‌توان از آن شروع کرد)  
تا اتمام ساخته شدن زمان‌بندی انجام‌شدنی کارهای زیر را انجام بده: (با توجه به اینکه فعالیت  $n$  آخرین است لذا می‌توان گفت تا اختصاص فعالیت  $n$ )  
با توجه به روابط پیش‌نیازی فعالیت‌هایی را که امکان انجام‌شدن دارند پیدا کن  
از فهرست فعالیت‌های انجام‌شدنی فعالیتی را که در بردار اولویت دارای کمترین زمان شروع است انتخاب کن.

زمان شروع و عملگر processor را برای فعالیت انتخابی محاسبه کن. زمان عملگر processor را برابر با زمان خاتمه فعالیت قرار بده. تجهیزات مورد نیاز فعالیت انتخابی را بخوان.

با توجه به این موضوع که یک تجهیز در یک زمان نمی‌تواند بیش از یک فعالیت را خدمت‌دهی کند و چنانچه یک فعالیت به بیش از یک تجهیز نیاز داشته باشد می‌بایست که همه آن تجهیزات را در یک زمان در اختیار داشته باشد، برای همه فعالیت‌هایی که در ادامه به این تجهیزات نیاز دارند و هنوز انجام نشده‌اند زمان امکان انجام شدن release time را برابر با زمان خاتمه این فعالیت قرار بده. (در این صورت فعالیت‌هایی که به این تجهیزات نیاز دارند باید صبر کنند تا کار روی این فعالیت تمام شود.)

برگرد به ابتدای حلقه

برنامه زمان‌بندی انجام‌شدنی را به‌عنوان خروجی قرار بده

پایان

به‌عنوان مثال کروموزوم زیر را که شامل ده فعالیت است را در نظر بگیرد (جدول ۱). همانطوری که قبلاً نیز اشاره شد سطر اول معرف مدت زمان انجام فعالیت و سطر دوم معرف زمان خاتمه فعالیت است.

chromosome =

1 1 3 2 3 2 2 2 3 2

8 4 11 13 8 4 9 13 11 9

فرض کنید که در این مثال سه تجهیز وجود دارد که ماتریس فعالیت‌هایی که به تجهیزات نیاز دارند به صورت زیر است

psep.equipment = [1 0 0 0 1 0 1 0 1 0 ...

[0 0 0 0 1 1 1 0 1 1 ...

[1 0 0 0 0 0 1 1 1 0]

همانگونه که مشاهده می‌شود در ماتریس psep.equipment هر سطر مربوط به یک تجهیز است. به‌عنوان مثال فعالیت‌های اول و پنجم و هفتم و نهم برای اجرا به تجهیز اول نیازمندند. طبق الگوریتم مراحل را اجرا می‌کنیم:

prior\_start = 7 3 8 11 5 2 7 11 8 7

از فهرست فعالیت‌هایی که می‌توانند شروع شوند فعالیتی را که دارای کمترین زمان شروع است انتخاب کن. لذا فعالیت ۱ انتخاب می‌شود. حال بردار شروع زمان فعالیت را تشکیل می‌دهیم. بنابراین فعالیت اول در زمان صفر شروع می‌شود.

start = 0 0 0 0 0 0 0 0 0 0

تجهیزات موردنیاز فعالیت انتخابی را بخوان.

فعالیت یک، به تجهیز اول و سوم احتیاج دارد لذا فعالیت‌های ۵ و ۷ و ۹ به‌خاطر تجهیز اول و فعالیت ۸ هم به‌خاطر تجهیز سوم باید تا زمان اتمام فعالیت ۱ در زمان یک منتظر بمانند.

release = 0 0 0 0 1 0 1 1 1 0

تکرار دوم: با توجه به روابط پیش‌نیازی فعالیت‌هایی را که امکان انجام‌شدن دارند پیدا می‌کنیم

لذا از بین فعالیت‌های ۲ و ۳ که امکان انجام‌شدن دارند فعالیتی را که در بردار prior\_start

دارای کمترین زمان شروع است انتخاب می‌کنیم.

7 3 8 11 5 2 7 11 8 7 prior\_start =

پس فعالیت ۲ انتخاب می‌شود

start = 0 1 0 0 0 0 0 0 0 0

release = 0 0 0 0 1 0 1 1 1 0

تکرار سوم: مجدداً با توجه به روابط پیش‌نیازی از بین فعالیت‌های ۳ و ۴ و ۵ که امکان

انجام‌شدن دارند فعالیت ۵ انتخاب خواهد شد.

prior\_start = 7 3 8 11 5 2 7 11 8 7

start = 0 1 0 0 2 0 0 0 0 0

release = 0 0 0 0 1 5 5 1 5 5

چنانچه الگوریتم را به همین صورت ادامه دهیم برنامه زمان‌بندی انجام‌شدنی به شرح زیر

به‌دست خواهد آمد

برنامه زمان‌بندی انجام‌شدنی را به‌عنوان خروجی قرار بده

Finish = 1 2 4 4 5 7 9 14 12 16

## نتایج محاسباتی

### تولید مسائل آزمایشی

گوناگونی حالات مختلف مسائل و الگوریتم‌های حل متعدد سبب شده‌است تا محققان مسائل استاندارد را به وجود آورند. دیویس برای اولین بار ۸۳ مسئله استاندارد RCPSP را به وجود آورد (Davis and Patterson, 1969) و کولیش نیز برای اولین بار بسته نرم‌افزاری ProGen (Kolish, 1998) را ایجاد کرد. در ادامه دیویس، پترسون و هابر (Patterson and Huber, 1974)، تالبوت و پترسون (Talbot and Patterson, 1980) و پترسون (Patterson, 1998) مسائل دیگری را اضافه و مجموعه‌ای شامل ۱۱۰ مثال استاندارد به وجود آوردند.

با توجه به اینکه مسئله PSEP با فرض شکست‌های تصادفی برای تجهیزات اولین بار در این تحقیق مورد بررسی قرار گرفته‌است هیچ‌گونه مسائل تولید شده قبلی برای آن یافت نشد. در این تحقیق برای انجام آزمایش‌ها از RanGen2 (Demeulemeester, 2003؛ 2004, Vanhoucke)، استفاده شد. اطلاعات ورودی مثال‌های مورد استفاده در جدول ۱ نشان داده شده‌اند. پارامترهای ورودی را به شرح جدول ۲ در برنامه تنظیم کردیم.

جدول ۱- اطلاعات ورودی برای مثال ارائه شده

فعالیت J	فعالیت های پیش‌نیازی	زمان فشرده	زمان نرمال
1	-	1	3
2	1	1	3
3	1	2	4
4	2	1	3
5	2	2	3
6	3	1	3
7	3	2	3
8	4,6	1	3
9	5,7	2	3
10	8,9	1	2
تجهیز i ام		فعالیت هایی که نیازمند تجهیز هستند	
1		1, 5, 7, 9	
2		5, 6, 7, 9, 10	
3		1, 7, 8, 9	

جدول ۲- سطوح پارامترهای مسائل PSEP

پارامترها	مقدار			
	7	10	20	30
Number of activities	7	10	20	30
Number of equipment types	0;1;3	0;1;3	0;1;3	0;1;3
Number of activities requiring equipment	0;1;7	0;1;7	0;1;12	0;1;12
activity crash time	[1,10]	[1,10]	[1,10]	[1,10]
activity crash time	$v_j + [1,3]$	$v_j + [1,3]$	$v_j + [1,3]$	$v_j + [1,3]$
reward paid	[0,50]	[0,50]	[0,50]	[0,50]
penalty cost	[0,50]	[0,50]	[0,50]	[0,50]
worth of completed activity	[0.01,0.05]	[0.01,0.05]	[0.01,0.05]	[0.01,0.05]
set-up cost	[1500,3000]	[1500,3000]	[1500,3000]	[1500,3000]
Idle time cost	[300,600]	[300,600]	[300,600]	[300,600]
Operator overtime	[60,100]	[60,100]	[60,100]	[60,100]
Reduction cost (\$/pd.)	[200,600]	[200,600]	[200,600]	[200,600]
Crash cost (\$)	[2000,5000]	[2000,5000]	[2000,5000]	[2000,5000]
$S_{ijk}$	[1000,2500]	[1000,2500]	[1000,2500]	[1000,2500]
d	Randomly selected with factor 1.5, 1.75, 2.00, 2.25, 2.5			
H	Randomly selected with factor 1.5, 1.75, 2.00, 2.25, 2.5			

### مقایسه نتایج آزمایش‌ها

به منظور ارزیابی اجرای الگوریتم‌های پیشنهادی برنامه کامپیوتری با متلب کد شد و سپس بروی ۶۰ مسئله آزمایشی به کار گرفته شد. نتایج برخی از مسائل در جدول ۳ آورده شده است. هر مسئله ۵ بار اجرا شد و بهترین جواب و میانگین زمان اجرا مورد آزمون فرض قرار گرفت (جدول ۴). شکل ۲ نشان‌دهنده نمودار همگرایی مسئله A30410 است. یعنی مسئله‌ای با ۳۰ فعالیت و ۴ تجهیز که از بین ۳۰ فعالیت ۱۰ فعالیت به این ۴ تجهیز نیازمندند.

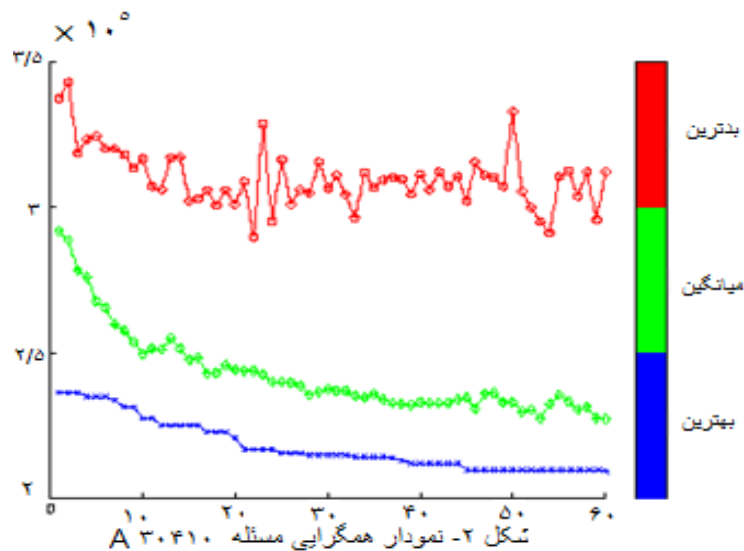
جدول ۳- تست فرضیه اختلاف نتایج دو الگوریتم SA و GA

شماره	اندازه مسئله	نام مسئله	نتایج الگوریتم نتیک		نتایج الگوریتم شبیه سازی تبرید	
			بهترین حل	میانگین زمان اجرا	بهترین حل	میانگین زمان اجرا
1	30 ACTIVITIES	A30312.psep	168665	1.3857309	178130	1.441
2		A30309.psep	145000	0.359	157610	0.766
3		A30207.psep	131290	0.891	141240	0.859
4	20 ACTIVITIES	A20312.psep	118050	1.188	122550	1.131
5		A20309.psep	90894	0.172	97140	0.453
6		A20305.psep	93245	0.516	95367	0.328
7	10 ACTIVITIES	A10307.psep	62866	0.141	62981	0.094
8		A10304.psep	53729	0.531	54673	0.516
9		A10205.psep	36897	0.734	36897	0.5
10	7 ACTIVITIES	A07307.psep	49670	0.4844	50126	0.484
11		A07304.psep	41710	0.0625	41710	0.891
12		A07205.psep	37452	0.4063	37593	0.938

جدول ۴- تست فرضیه اختلاف نتایج دو الگوریتم SA و GA

نتیجه	تصمیم در مورد فرض صفر	آماره t	انحراف معیار نمونه‌ها	اختلاف میانگین‌ها مربوط به:	فرض صفر	اختلاف مقادیر تابع هدف
اختلاف معنی داری وجود ندارد	رد نمی شود	-1.98	2221.29	7activities	$\mu_{1t}=\mu_{2t}$	
اختلاف معنی داری وجود ندارد	رد نمی شود	-2.65	1124.229	10activities	$\mu_{1s}=\mu_{2s}$	
اختلاف معنی داری وجود دارد	رد می شود	-2.85	4218.04	20activities	$\mu_{1M}=\mu_{2M}$	
اختلاف معنی داری وجود دارد	رد می شود	-3.27	6148.19	30activities	$\mu_{1L}=\mu_{2L}$	
اختلاف معنی داری وجود دارد	رد می شود	-8.12	2824.81	60activities	$\mu_{1h}=\mu_{2h}$	
اختلاف معنی داری وجود دارد	رد می شود	-6.16	4319.15	کل مسائل	$\mu_{1T}=\mu_{2T}$	

اختلاف معنی‌داری وجود دارد	رد می‌شود	4.31	0.287296	7activities	$\mu_{1t}=\mu_{2t}$	اختلاف زمان اجرا
اختلاف معنی‌داری وجود دارد	رد می‌شود	6.67	0.239296	10activities	$\mu_{1s}=\mu_{2s}$	
اختلاف معنی‌داری وجود دارد	رد می‌شود	5.54	0.300438	20activities	$\mu_{1M}=\mu_{2M}$	
اختلاف معنی‌داری وجود دارد	رد می‌شود	5.32	0.324492	30activities	$\mu_{1L}=\mu_{2L}$	
اختلاف معنی‌داری وجود دارد	رد می‌شود	7.80	2.49165	60activities	$\mu_{1h}=\mu_{2h}$	
اختلاف معنی‌داری وجود دارد	رد می‌شود	4.87	2.35690	کل مسائل	$\mu_{1T}=\mu_{2T}$	



### نتیجه‌گیری و تحقیقات آتی

در این مقاله با مدنظر قرار دادن همزمان زمان‌بندی پروژه و برنامه‌ریزی تجهیزات، به‌منظور کمینه کردن هزینه‌های فشرده‌سازی فعالیت‌ها، نگهداری، پاداش برای زودکرد و جریمه برای دیرکرد، حمل‌ونقل، آماده‌سازی تجهیزات، هزینه بیکاری تجهیزات، و اضافه‌کاری اپراتورها برای نخستین بار دو الگوریتم ژنتیک و شبیه‌سازی تبرید ارائه شد.

برای بررسی کارایی الگوریتم‌های پیشنهادی، مسائلی با تعداد فعالیت‌ها و تجهیزات مختلف مورد حل قرار گرفت. نتایج نشان دادند که الگوریتم‌های پیشنهادی قادرند تا مدل برنامه‌ریزی خطی عدد صحیح مختلط موجود را با تعداد گره‌های بالا و ضریب پیچیدگی‌های مختلف در زمانی کوتاه حل کنند.

نتایج آزمون آماری انجام شده برای مسائل با هفت و ده و بیست و سی و شصت فعالیت و در سطح اطمینان ۹۵٪ بهتر بودن جواب‌های الگوریتم ژنتیک خصوصاً در ابعاد بزرگ را نسبت به شبیه‌سازی تبرید نشان می‌دهد.

تحقیقات آتی در این زمینه به شرح زیر است.

استفاده از شبکه‌های GPR به جای AON و در نظر گرفتن پارامترهای مسئله به صورت فازی.

استفاده از الگوریتم‌های رقیب، به عنوان مثال استفاده از الگوریتم‌های فراابتکاری دیگر به عنوان رقیب (همچون امپریالیست و جستجوی ممنوع و کلنی مورچه‌ها و ...)  
الگوریتم‌های هیبریدی شامل: حل انواع مسائل دیگر، راه‌حل بهبودی، روش‌های دیگر هیبریدی و ...



## منابع

- Brucker, P., Schoo, A. and Thiele, O., A branch and bound algorithm for the resource constrained project scheduling problem. *European Journal of Operational Research* 17 (2004): 143-158.
- Carruthers, J.A. and Battersby, A., Advances in critical path methods. *Operational Research Quarterly* 17 (1966): 359-380.
- Davis, E.V. and Patterson, J.H., "An Exact Algorithm for the Multiple Constrained Project Scheduling Problem". Ph.D. Thesis. Yale University. (1969).
- Deckro, R.F., Winkofsky, E.P., Hebert, J.E. and Gagnon, R., A decomposition approach to multi-project scheduling. *European Journal of Operational Research* 51 (1991): 110-118.
- Demeulemeester, E., Herroelen, W. *Project Scheduling: A Research Handbook*". Kluwer Academic Publishers, (2002).
- Demeulemeester, E., Vanhoucke, M. and Herroelen, W., A random network generator for activity-on-the-node networks. *Journal of Scheduling* 6 (2003): 13-34.
- Demeulemeester, E. and Herroelen, W., New benchmark results for the resourceconstrained project scheduling problem. *Management Science* 43 (2007): 1485-1492.
- Dodin, B., Elimam, A.A., Rolland, E. Tabu search in audit scheduling. *European Journal of Operational Research* 106 (1998): 373-392.
- Dodin, B. Elimam, A.A. Integration of equipment planning and project scheduling. *European Journal of Operational Research* 184 (2008): 962-980.
- Elmaghraby, S.E. Activity nets: A guided tour through some recent envelopments. *European Journal of Operational Research* 82 (2002): 383-408.
- Elmaghraby, S.E. *Activity networks: project planning and control by network models*. Wiley and Sons, New York, (2003).
- Herroelen, W., Van Dommelen, P., Demeulemeester, E.L. Project network models with discounted cash flows – A guided tour through recent developments. *European Journal of Operational Research* 100 (1997): 97-121.
- Herroelen, W., De Reyck, B. and Demeulemeester, E., Resource constrained project scheduling: A survey of recent developments. *Computers and Operations Research* 25 (2001): 279-302.

Icmeli, O. and Rom, W.O., Solving the resource-constrained project scheduling problem with optimization subroutine library. *Computers and Operations Research* 23 (1996): 801-817.

Icmeli, O., Erenguc, S.S., Zappe, C.J. Project scheduling problems: A survey. *International Journal of Operations and Production Management* 13 (2003): 80–91.

Jarboui, B., Damak, N., Siarry, P., Rebai, A., A combinatorial particle swarm optimization for solving multi-mode resource-constrained project scheduling problems. *Applied Mathematics and Computation* 195 (2008): 299–308.

Kolish, R., Schwindt, C., and Sprecher, A. Benchmark instances for project scheduling problems. in: J. Weglarz (Ed.): *Handbook on recent advances in project scheduling*, Kluwer Academic Publishers, Dordrecht. (1998).

Lova, A., Tormos, P., Cervantes, M., Barber, F., “An Efficient Hybrid Genetic Algorithm for Scheduling Projects with Resource Constraints and Multiple Execution Modes”, *International Journal of Production Economics*, 117, (2009), 302– 316.

Ozdamar, L., Ulusoy, G. “A survey on the resource constrained project scheduling problem”. *IIE Transactions* 27, 574–586, (1995).

Patterson, J.H., Roth, G.W. “Scheduling a project under multiple resource constraints: A 0-1 programming approach”. *AIIE Transactions* , 8, 449–455, (1976).

Patterson, J.H. and Huber, W.D. “A Horizon Varying, Zero One Approach to Project Scheduling”, *Management Science*, 20:990-998, (1974).

Patterson, J.H, “A Comparison of Exact Approach for Solving the Multiple Constrained Resource, Project Scheduling Problem”, *Management Science*, Vol. 30, No.7. PP.854-867, (1984).

Peteghem, V.V., Vanhoucke, M., “A Genetic Algorithm for the Preemptive and Non-Preemptive Multi-Mode Resource-Constrained Project Scheduling Problem”, *European Journal of Operational Research*, 201, (2010), 409–418.

Petrovic, R., *Optimisation of resource allocation in project planning. Operations Research*, 16 (1968) 559-586.

Soltani, R., Jolai, F., Zandieh, M., “Two Robust Meta-Heuristics for Scheduling Multiple Job Classes on a Single Machine with Multiple Criteria”, *Expert Systems with Applications*, 37, (2010), 5951-5959.

Talbot, F.B, and Patterson, J.H. “An Efficient Integer Programming Algorithm With Network Cuts for Solving Resource Constrained Scheduling”, *Management Science*, Vol.24, No. 11, PP.1163-1174.

Tavakoli M.R, Shahalizadeh M, Khalili K, Editors. *International project management conference*. (2007) February 10; Iran.

Vanhoucke, M., Coelho, J., Tavares, L. and Debels, D., “An evaluation of the adequacy of network generators with systematically sampled networks”. *Working paper 04/272*, Ghent University, (2004).

Wiest, J.D., *The scheduling of large projects with limited resources*. Ph.D. dissertation, Carnegie Institute of Technology, (1963).

Yang, B., Geunes, J. and O’Brien, W. *Resource constrained project scheduling; past work and new directions*. Research Report, Department of Industrial and Systems Engineering, University of Florida, Gainesville, (2006).