

## ارائه مدلی یکپارچه برای انتخاب سبد پروژه و سرمایه‌گذاری در منابع با هدف پیشینه‌سازی ارزش خالص فعلی و حل آن توسط الگوریتم ژنتیک

بهرروز افشارنجفی\*، حمیدرضا شهابی فرد\*\*

تاریخ دریافت: ۹۳/۷/۲۹

تاریخ پذیرش: ۹۵/۲/۱۲

### چکیده:

در این مقاله، یک مدل بهینه‌سازی جهت انتخاب بهترین سبد پروژه از بین پروژه‌های موجود، بهترین سطح استخدام منابع، سپس زمان‌بندی پروژه‌های انتخاب شده جهت پیشینه‌کردن ارزش خالص فعلی با رعایت محدودیت‌ها ارائه شده است. چون مدل توسعه یافته در زمره مسایل سخت از نظر محاسباتی قرار دارد، لذا برای حل این مساله یک الگوریتم فراابتکاری بر مبنای الگوریتم ژنتیک پیشنهاد شده است. در الگوریتم حل پیشنهادی علاوه بر کاربرد عملگرهای معمول ژنتیک مانند تقاطع و جهش از عملگرهایی هوشمند جهت جستجوی محلی در حوزه منابع و جابجایی فعالیت‌های با جریان مالی منفی استفاده شده است. پارامترهای کلیدی الگوریتم در راستای تسریع همگرایی آن با استفاده از روش تاگوجی تنظیم شده است. سپس تعداد ۹۰ مساله آزمایشی، شامل ۳۰ مساله در ابعاد کوچک، ۳۰ مساله در ابعاد متوسط و ۳۰ مساله در ابعاد بزرگ با استفاده از روش پیشنهادی حل شده و کارایی آن گزارش شده است. همچنین در مسائل ساینز کوچک جواب‌های حاصل از الگوریتم ژنتیک با جواب‌های بهینه موضعی مدل ریاضی بدست آمده با نرم‌افزار لینگو مقایسه شده، که میانگین جواب‌های حاصل از الگوریتم ژنتیک بهتر از جواب‌های بهینه موضعی حاصل از لینگو بوده است. در مسائل ساینز متوسط و بزرگ که هیچ جوابی با استفاده از لینگو در زمان محدود شده بدست نیامده بود، نتایج نشان می‌دهد که جواب‌های حاصل از الگوریتم پیشنهادی دارای پایداری مناسب می‌باشند.

**واژه‌های کلیدی:** انتخاب سبد پروژه، زمان‌بندی پروژه، الگوریتم ژنتیک، سرمایه‌گذاری در منابع، ارزش خالص فعلی.

\*دانشیار، دانشکده مهندسی صنایع و مکانیک، دانشگاه آزاد اسلامی، واحد قزوین، (نویسنده مسئول)

Afsharnb@alum.sharif.edu

\*\*کارشناسی ارشد، دانشگاه علوم و تحقیقات، دانشگاه آزاد اسلامی، واحد علوم و تحقیقات قزوین

## مقدمه

زمان‌بندی پروژه بدون در نظر گرفتن هیچ محدودیتی عبارت است از مشخص نمودن زمان آغاز و پایان فعالیت‌های یک پروژه با توجه به روابط پیش‌نیازی فعالیت‌ها. ولی در عمل شرایط و محدودیت‌هایی در اجرای یک پروژه واقعی وجود دارد که نمی‌توان آنها را نادیده گرفت. عواملی چون محدودیت منابع یا فرضیات دیگری که می‌توانند بر روی این محدودیت‌ها تاثیر گذارند؛ مانند منابع تجدیدپذیر یا تجدیدناپذیر، منابع با محدودیت دوگانه و یا منابع چندحالتی. و بسیاری از فرضیات دیگر که در این مقاله مورد بحث نیست. با توجه به این‌ها زمان‌بندی پروژه به معنی تخصیص منابع و زمان به فعالیت‌های یک پروژه می‌باشد.

یکی از مسائل زمان‌بندی پروژه، مساله تسطیح منابع است که در آن مقدار استفاده از منابع با توجه به محدودیت زمانی اجرای پروژه کمینه می‌گردد. هدف مطرح شده در این مسائل کمینه کردن هزینه‌های ناشی از استخدام منابع است که به آنها مسائل سرمایه‌گذاری در منابع<sup>۱</sup> یا در بعضی منابع، مسائل قیمت دسترس‌پذیری منابع<sup>۲</sup> نیز گفته می‌شود.

در زمینه مسائل سرمایه‌گذاری در منابع که شکل خاص تخصیص منابع در کنترل پروژه می‌باشد، کارهای صورت گرفته کمتر از مسائل زمان‌بندی پروژه با محدودیت منابع است. این مساله اولین بار در سال ۱۹۸۴ توسط آقای موهرینگ<sup>۳</sup> ارائه گردید. او نشان داد که مسائل سرمایه‌گذاری در منابع از نوع NP-Hard - سخت می‌باشند [۴]. از تحقیقاتی نیز که اخیراً روی این موضوع انجام شده می‌توان به این موارد اشاره کرد: در سال ۲۰۰۷ شادرخ و کیانفر الگوریتم ژنتیک را در مساله سرمایه‌گذاری در منابع بکار بردند و با حل ۶۹۰ مثال با روش ارائه شده و مقایسه آن با نتایج حاصل از الگوریتم‌های دیگر که قبلاً حاصل شده، نشان دادند که نتایج رضایت‌بخش است [۸]. در سال ۲۰۰۹ نجفی و همکاران الگوریتم ژنتیک را برای حل مساله سرمایه‌گذاری در منابع پروژه با منابع مالی تنزیل شده بکار گرفتند و با حل ۲۴۰ مثال کارایی آنرا نشان دادند [۵] و در سال ۲۰۱۰ نیز، شهسوار و همکاران در مساله

---

1. Resource Investment Problem (RIP)

2. Resource Availability Cost Problem

3. Mohring

سرمایه‌گذاری در منابع با جریان‌های مالی تنزیل شده، فاکتور تورم را در نظر گرفته و سیاست جایزه - جریمه زود کرد یا دیر کرد پروژه را نیز لحاظ کردند و هدف آنها ارائه زمان‌بندی بود که بتواند ارزش خالص فعلی را بیشینه کند [۹]. در سال ۲۰۱۴، افشارنجفی مساله سرمایه‌گذاری در منابع پروژه را با منابع چندگانه مورد بررسی قرار داده که در آن مقدار و تاریخ استخدام منابع حاصل می‌شود. پروژه‌ها شامل فعالیت‌هایی با روابط پیش‌نیازی پایان-شروع و با پس‌افت صفر بوده و منابع نیز تجدیدپذیر هستند. در این تحقیق یک مدل ریاضی عدد صحیح مختلط برای این مساله پیشنهاد شده و سپس یک الگوریتم شبیه‌سازی تبرییدی برای حل آن توسعه داده شده است. او کارایی این الگوریتم را با حل ۳۰۰ مساله مختلف به اثبات رسانده و نتایج را مورد بررسی و بحث قرار می‌دهد [۳].

یکی از موضوعات دیگر در بحث زمان‌بندی پروژه‌ها، انتخاب و تشکیل سبدهای پروژه‌ها (Project Portfolio) از میان پروژه‌های موجود و سپس زمان‌بندی آنها با توجه به محدودیت‌ها و اهداف از پیش تعیین شده است. عموماً سرمایه‌گذاران و مجریان پروژه‌ها از جهات اقتصادی و رقابتی مایلند چند پروژه را به صورت همزمان آغاز کرده و از منابع و سرمایه خود به صورت مشترک استفاده نمایند. اهمیت این مسائل تا جایی است که آقای پاین<sup>۱</sup> اظهار کرده از لحاظ مالی تا ۹۰٪ تمام پروژه‌ها در حالت چند پروژه‌ای اجرا می‌شوند [۶]. ولی انتخاب پروژه‌ها از یک طرف با توجه به محدودیت‌های موجود مانند سرمایه، منابع و ... و از طرفی دیگر برآورده کردن انتظارات کارفرما مانند اتمام پروژه‌ها در زمانی مشخص، و همچنین انتظار سود بیشتر از اجرای پروژه‌ها، چالشی را فراروی آنها قرار می‌دهد که مهارت در مدیریت و زمان‌بندی همزمان پروژه‌ها را می‌طلبد. به این نوع مسائل، زمان‌بندی چندپروژه‌ای<sup>۲</sup> گفته می‌شود.

این مساله اولین بار در سال ۱۹۶۹ در قالب یک برنامه خطی صفر و یک توسط آقای پریسکر<sup>۳</sup> معرفی گردید [۷]. سپس بسیاری از تحقیقات انجام شده در این دسته از مسائل فقط

---

1. Payne

2. Multi-Project Scheduling

3. Pritsker

بر روی انتخاب پروژه‌های در دسترس یا زمان‌بندی پروژه‌های در دست‌یابی به اهداف با بودجه‌های مشخص تمرکز کرده‌اند. تعدادی از مطالعات صرفاً بر روی مسائل انتخاب پروژه‌ها در صنایع فن‌آوری اطلاعات و مهندسی انجام شده است و مدل‌هایی نیز از طریق برنامه‌ریزی ریاضی، روش‌های ابتکاری انعطاف‌پذیری را از قبیل روش‌های غیر قطعی فازی توسعه داده‌اند. ولی تعداد کمی از تحقیقات انجام شده به طور هم‌زمان انتخاب سببی از پروژه‌ها و سپس زمان‌بندی فعالیت‌های آنها را با توجه به محدودیت منابع در بر می‌گیرند. با توجه به این مقدمه، تاکنون کاربرد مساله سرمایه‌گذاری در منابع برای چندپروژه‌ای مورد مطالعه قرار نگرفته است. بنابراین آنچه در این تحقیق مورد واکاوی و بررسی قرار خواهد گرفت، تشکیل سببی از پروژه‌ها از بین پروژه‌های موجود و سپس حل مساله سرمایه‌گذاری در منابع برای مشخص کردن سطح بهینه استخدام آنها و زمان‌بندی پروژه‌های انتخاب شده با منابع مشترک در زمانی مشخص با هدف بیشینه کردن ارزش خالص فعلی می‌باشد.

## بیان مسأله و مدلسازی ریاضی

### فرضیات

تعداد  $M$  پروژه بالقوه که هر کدام شامل  $nm$  فعالیت می‌باشند جهت اجرا وجود دارند. مدل در نظر گرفته شده برای هر پروژه از نوع گرهی بوده و فعالیت‌های  $m1$  و  $nm$  برای هر پروژه موهومی در نظر گرفته می‌شوند. فعالیت‌ها قابل شکسته شدن نبوده و مهلت اجرای کلیه پروژه‌ها زمان محدود  $T$  می‌باشد. روابط پیش‌نیازی فعالیت‌ها در حالت پایان به شروع بوده و زودترین زمانی که یک فعالیت می‌تواند آغاز شود بلافاصله بعد از اتمام فعالیت(های) پیش‌نیازش می‌باشد. کلیه پارامترهای مدل قطعی و غیرتصادفی است. برای اجرای فعالیت‌های پروژه(ها) به  $K$  نوع منبع از نوع تجدیدپذیر نیاز است. مقدار سطح هر منبع نیز در طول دوره ثابت است. هر فعالیت در هر پروژه (غیر از فعالیت‌های موهومی) می‌تواند شامل هزینه‌هایی غیر از هزینه استفاده از منابع باشد، مانند هزینه‌های سربار و ... که این هزینه‌ها در ابتدای هر فعالیت (گره) رخ می‌دهند. هر فعالیت در هر پروژه می‌تواند شامل درآمد باشد، مانند مبالغ

توافق شده با کارفرما که در پایان فازهای مختلف پروژه پرداخت می‌شود و این درآمدها در پایان هر فعالیت (گره) کسب می‌گردند.

### پارامترهای مدل

- $d_{im}$ : مدت زمان انجام فعالیت  $i$  از پروژه  $m$ .  $i_m=1,2,\dots,n_m$
- $P(i_m)$ : مجموعه فعالیت‌های پیش‌نیاز فعالیت  $i$  از پروژه  $m$ .
- $f_{imk}$ : میزان احتیاج فعالیت  $i_m$  به منبع  $k$  در هر واحد زمانی.
- $C_k$ : هزینه استخدام هر واحد منبع تجدیدپذیر  $k$ .  $k=1,2,\dots,K$
- $E_{im}$ : میزان درآمد حاصل از فعالیت  $i$  از پروژه  $m$ .  $i_m=1,2,\dots,n_m$
- $O_{im}$ : هزینه اجرای فعالیت  $i$  از پروژه  $m$ .  $i_m=1,2,\dots,n_m$
- $T$ : حداکثر مدت زمان مجاز اجرای پروژه‌ها.
- $\alpha$ : نرخ بهره.

### متغیرهای تصمیم

- $S_{im}$ : زمان شروع فعالیت  $i$  از پروژه  $m$ .  $i_m=1,2,\dots,n_m$
- $R_k$ : سطح استخدام یا تهیه منبع  $k$ .  $k=1,2,\dots,K$
- $X_m$ : مقدار ۱ را می‌گیرد اگر پروژه  $m$  جهت اجرا در دوره  $T$  انتخاب شود و در غیر این صورت مقدار صفر را می‌گیرد.  $m=1,2,\dots,M$
- $Y_{imt}$ : مقدار ۱ را می‌گیرد اگر فعالیت  $i$  از پروژه  $m$  در زمان  $t$  شروع شود و در غیر این صورت مقدار صفر را می‌گیرد  $t \in \{0, \dots, T\}$

### مدل ریاضی

مدل توسعه داده شده یک مدل برنامه‌ریزی عدد صحیح مختلط بوده و با توجه به تابع هدف آن که بیشینه کردن ارزش خالص فعلی است، مقدار بهینه سه متغیر  $S_{im}$ ،  $R_k$  و  $X_m$  را مشخص خواهد نمود. در حقیقت مشخص خواهد نمود که از سبد پروژه‌های موجود، چه

پروژه‌هایی برای اجرا انتخاب شده، برای هر منبع تجدیدپذیر چه سطحی بکار گرفته شود و شروع فعالیت‌های پروژه‌های منتخب در چه روزی خواهد بود.

$$\begin{aligned} \text{Max } Z = & \sum_{m=1}^M \sum_{i_m=2}^{n_m-1} X_m E_{im} e^{-\alpha(S_{im}+d_{im})} \\ & - \sum_{m=1}^M \sum_{i_m=2}^{n_m-1} X_m O_{im} e^{-\alpha S_{im}} - \sum_{k=1}^K C_k R_k \end{aligned} \quad (1)$$

$$S_{1m} = 0 \quad , m = 1, \dots, M \quad (2)$$

$$S_{nm} \leq T \quad , m = 1, \dots, M \quad (3)$$

$$\begin{aligned} S_{im} - S_{jm} \geq X_m d_{jm} \quad \forall j_m \in P(i_m) \quad , i_m \\ = 2, 3, \dots, n_m \quad , m = 1, \dots, M \end{aligned} \quad (4)$$

$$\sum_{t=0}^T Y_{imt} = X_m \quad , i_m = 1, 2, \dots, n_m \quad , m = 1, \dots, M \quad (5)$$

$$S_{im} = \sum_{t=0}^T t Y_{imt} \quad , i_m = 1, 2, \dots, n_m \quad , m = 1, \dots, M \quad (6)$$

$$\begin{aligned} \sum_{m=1}^M \sum_{i_m=2}^{n_m-1} \sum_{u=t-d_{im}+1}^t r_{imk} Y_{imu} \leq R_k \quad , t \\ = 0, 1, \dots, T-1 \quad , k = 1, 2, \dots, K \end{aligned} \quad (7)$$

$$\begin{aligned} R_k \geq 0 \quad \forall k \quad , S_{im} \geq 0 \text{ \& Integer } \quad , Y_{imt} \\ = \{0, 1\} \quad , X_m = \{0, 1\} \end{aligned} \quad (8)$$

رابطه یک، تابع هدف مساله بوده که ارزش خالص فعلی حاصل از اجرای پروژه‌های انتخاب شده با سطوح منابع مشخص را بیشینه می‌کند. این رابطه از سه بخش تشکیل شده است. بخش اول آن مربوط به درآمدهای کسب شده در هر فعالیت (گره) می‌باشد و همانطور که قبلاً ذکر شد در آخر هر فعالیت محقق می‌شود؛ بخش دوم مربوط به هزینه‌های سربار و هزینه‌هایی غیر از منابع تجدیدپذیر می‌باشد که در ابتدای هر فعالیت (گره) پرداخت می‌شوند؛ و قسمت آخر این رابطه مربوط به هزینه منابع استخدام شده جهت اجرای پروژه‌هاست که در اول دوره رخ می‌دهد.

در رابطه دو مشخص می‌شود که شروع تمام فعالیت‌های نخستین مجازی در هر پروژه، زمان

صفر می‌باشد. رابطه سه بر مهلت زمانی اتمام تمامی پروژه‌ها تأکید دارد. رابطه چهار روابط پیش‌نیازی بین فعالیت‌ها در هر پروژه را مشخص می‌کند. رابطه پنج و شش نشان می‌دهند که در صورت انتخاب پروژه  $m$ ، فعالیت  $im$  می‌بایست در یک زمان واحد شروع شود. رابطه هفت محدودیت سقف مجاز استفاده از منابع نوع  $k$  توسط تمامی فعالیت‌های پروژه‌ها در هر واحد زمانی را مشخص می‌کند.

### توسعه یک الگوریتم ژنتیک جهت حل مساله

مساله تخصیص منابع در کنترل پروژه به عنوان تعمیمی از مساله کلاسیک زمان‌بندی و توالی فعالیت‌ها از نوع  $in$ -پی‌سخت است. بنابراین برای مسائل متوسط تا بزرگ، الگوریتم‌های بهینه‌ساز در زمان معقول قادر به حل مساله نمی‌باشند و حل این مساله به دلیل پیچیدگی، گسترده‌گی و دشواری با رویکردهای بهینه‌نظیر برنامه‌ریزی ریاضی، برنامه‌ریزی پویا و یا شاخه و کران غیر عملی است و حتی روش‌هایی که فقط یک جواب قابل قبول را می‌دهند مفید محسوب می‌شوند. به همین دلیل در این تحقیق، الگوریتم فراابتکاری ژنتیک<sup>۱</sup> برای توسعه مدل در نظر گرفته شده است.

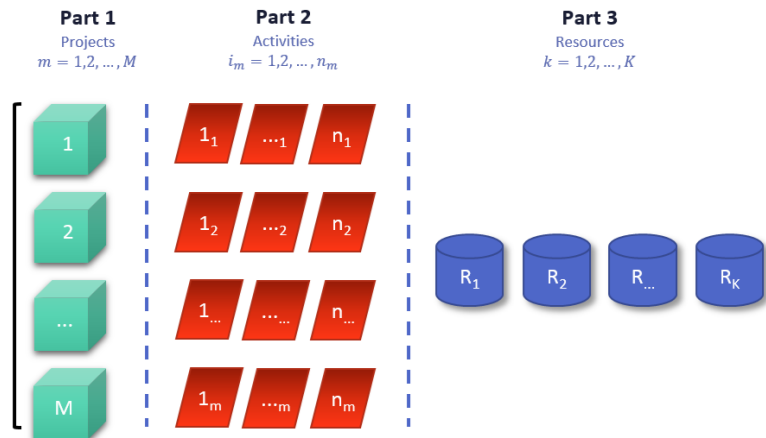
از آنجایی که مقدار سطح منابع جز متغیرهای مساله بوده و همچنین در بعضی از فعالیت‌ها جریان مالی منفی وجود دارد، معیار تابع هدف مساله از نوع نامنظم می‌باشد. به این معنی که با افزایش سطح یک منبع و زمان‌بندی فعالیت‌ها در زمانی زودتر و پیرو آن با اتمام زودتر یک پروژه مقدار تابع هدف که ارزش خالص فعلی است لزوماً افزایش نخواهد داشت و حتی ممکن است موجب زیان بیشتری شود. بنابراین در الگوریتم ژنتیک طراحی شده علاوه بر عملگرهای جهش و تقاطع از دو الگوریتم دیگر یکی برای جستجوی محلی در حوزه منابع و یکی نیز برای شناسایی فعالیت‌های با جریان مالی منفی و جابجا کردن آنها روی محور زمان به سمت راست (زمان شروع دیرتر) استفاده شده است. برای الگوریتم جستجوی محلی در حوزه منابع، پارامتری در نظر گرفته شده تا این الگوریتم بر روی درصد دلخواهی از جمعیت ایجاد شده اعمال شود. ولی الگوریتم شناسایی و جابجایی فعالیت‌های با جریان مالی منفی بر روی

کل جمعیت ایجاد شده به کار گرفته می‌شود. هر یک از اعضای جمعیت یک کروموزوم (Chromosome) است که بعد از رمزگشایی (Decode)، پروژه‌های فعال، زمان‌بندی فعالیت‌های هر پروژه و سطح منابع استخدام شده مشخص می‌شوند. سپس بلافاصله مقدار تابع هدف مساله یا همان ارزش خالص فعلی برای تک‌تک اعضای جمعیت محاسبه شده و در واقع عمل برازش کروموزوم‌ها صورت می‌گیرد. در ابتدا جمعیت به صورت تصادفی ایجاد شده، سپس در نسل‌های بعدی عملگرهای اشاره شده بکار گرفته می‌شوند. برای انتخاب والد در عملگرهای جهش و تقاطع و همچنین انتخاب کروموزوم‌ها برای عملگر جستجوی محلی در حوزه منابع از روشی به نام چرخه رولت (Roulette wheel) استفاده شده است. توسط این روش اعضای از جمعیت که مقدار برازش بهتری دارند شانس بیشتری نیز برای انتخاب خواهند داشت. پارامتری که در چرخه رولت برای تنظیم احتمال انتخاب جواب‌های بهتر در نظر گرفته شده پارامتر فشار نامیده می‌شود. پارامتر فشار اگر صفر باشد احتمال انتخاب تمامی اعضای جمعیت یکسان است و هرچه بزرگتر شود احتمال انتخاب اعضای بهتر جمعیت بیشتر خواهد شد. به طور تجربی وقتی این پارامتر روی عدد پنج تنظیم شود مجموع احتمال‌های نصف بهتر جمعیت حدود ۸۰٪ خواهد بود، به عبارت بهتر نصف جمعیت با احتمال ۸۰٪ انتخاب خواهند شد و مابقی با احتمال ۲۰٪.

### طرح کروموزوم

جهت داشتن یک الگوریتم ژنتیک کارا، طرح یک کروموزوم مناسب اهمیت زیادی دارد (Genotype). سپس می‌توان با رمزگشایی این کروموزوم، جواب مورد نظر را استخراج کرد (Phenotype). همانطور که از بخش مدل ریاضی مشخص شد، این مدل دارای سه نوع متغیر تصمیم می‌باشد. بنابراین کروموزوم طراحی شده نیز از نوع رشته‌ای بوده و دارای سه رشته (بخش) می‌باشد. بخش اول پروژه‌ها، بخش دوم توالی فعالیت‌ها و بخش سوم نیز منابع می‌باشد. که در ادامه توضیحات هر کدام ذکر شده است.





شکل ۱: تصویری نمادین از کروموزوم طراحی شده در الگوریتم ژنتیک

### رشته اول کروموزوم: انتخاب پروژه‌ها از سبد موجود

بخش اول مربوط به انتخاب پروژه‌ها از کل سبد موجود است. متغیرهای این دسته جز متغیرهای صفر و یک (Binary) بوده و در نتیجه دارای دو حالت خواهند بود؛ انتخاب یا عدم انتخاب یک پروژه. ممکن است زمان اتمام بعضی از پروژه‌ها بر اساس زودترین زمان اجرای فعالیت‌ها<sup>۱</sup> و بدون در نظر گرفتن محدودیت منابع، بیش از مدت زمان مجاز اجرای پروژه‌ها (T) باشد؛ که انتخاب آنها در کروموزوم باعث خواهد شد شرط اتمام پروژه‌ها زودتر از زمان مشخص T رعایت نشده و به اصطلاح جواب غیر شدنی حاصل شود. برای جلوگیری از انتخاب چنین پروژه‌هایی، هنگام اجرای برنامه و قبل از ایجاد جمعیت اولیه، مدت زمان اتمام تمامی پروژه‌های موجود در سبد، با روش زودترین زمان اجرا و بدون در نظر گرفتن محدودیت منابع مورد آزمایش قرار گرفته و پروژه‌هایی که مدت زمان اتمام آنها در این حالت از مدت زمان مجاز T فراتر رود تا انتهای اجرای الگوریتم از سبد موجود کنار گذاشته می‌شوند.

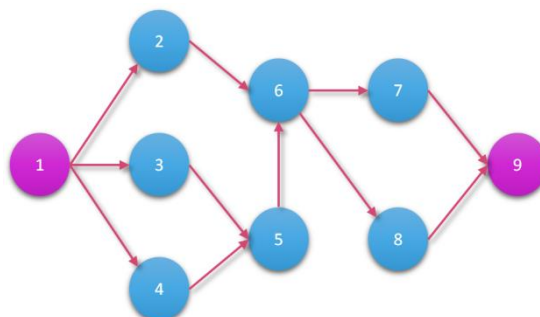
هنگام ایجاد جمعیت اولیه، برای ایجاد این بخش در کروموزوم یعنی انتخاب پروژه‌های فعال از بین m عدد پروژه در دسترس، به ایجاد تصادفی m عدد صفر و یک بسنده خواهد شد.

1. Earliest Start Time (EST)

## رشته دوم کروموزوم: توالی فعالیت‌ها [۲]

ترتیب یا توالی فعالیت‌های هر پروژه در بخش دوم کروموزوم مشخص می‌شود. هنگام ایجاد جمعیت اولیه، این توالی با در نظر گرفتن روابط پیش‌نیازی به صورت تصادفی ایجاد می‌گردد. به این معنی که شماره فعالیت‌های پیش‌نیاز، سمت چپ فعالیت‌های پس‌نیاز آنها قرار می‌گیرند. به عنوان مثال شکل دو شبکه گره‌ای پروژه‌ای را نشان می‌دهد که شامل نه فعالیت بوده (فعالیت‌های یک و نه موهومی هستند) و یک ترتیب تصادفی از آن می‌تواند به صورت

$$EAS = \{1,3,2,4,5,6,8,7,9\} \quad \text{مجموعه روبه‌رو باشد:}$$



شکل ۲: شبکه گره‌ای شامل نه فعالیت

این بخش یک آرایه  $i * m$  بوده،  $m$  تعداد سطرها یا همان پروژه‌های در دسترس بوده و  $i$  تعداد ستون‌ها یا همان فعالیت‌های هر پروژه است. البته  $i$  بر اساس بیشترین تعداد فعالیت‌های یک پروژه از بین پروژه‌های موجود مشخص می‌شود تا بتواند دربرگیرنده تمام فعالیت‌های بالقوه (در تمام پروژه‌ها) شود<sup>۱</sup>. در هنگام اجرای برنامه در ابتدا آرایه مورد نظر خالی است. سپس جهت ایجاد جمعیت اولیه، بدون توجه به اینکه در رشته اول کروموزوم چه پروژه‌هایی فعال یا غیرفعال است (دارای مقادیر یک یا صفر هستند)، ترتیب اجرای فعالیت‌های کلیه پروژه‌ها در سطرها مرتب آرایه<sup>۲</sup> مشخص خواهد شد. زیرا در تکرارهای بعدی جهت ایجاد نسل‌های جدید، هنگام اجرای اپراتورهای جهش و تقاطع ممکن است به این توالی‌ها نیاز

۱. تعداد فعالیت‌های پروژه‌ها ممکن است با هم متفاوت باشند، لذا تعداد ستون این آرایه به تعداد فعالیت‌های بزرگترین پروژه موجود در سبک در نظر گرفته می‌شود.

۲. سطر اول مربوط به پروژه اول، سطر دوم پروژه دوم و سطر  $m$  مربوط به پروژه  $m$  می‌باشد.

شود. ولی در هنگام رمزگشایی کروموزوم و برازش آن، صرفاً برنامه زمان‌بندی پروژه‌های فعال استخراج خواهند شد. جهت اجرای الگوریتم ایجاد ترتیب تصادفی (بخش دوم کروموزوم) تعدادی مجموعه به شرح زیر تعریف می‌گردند: مجموعه یا آرایه Activity List که دربرگیرنده لیست فعالیت‌های پروژه‌های در دسترس می‌باشد.

$$Activity List = \begin{bmatrix} i_{11} & \dots & i_{n1} \\ \vdots & \ddots & \vdots \\ i_{1m} & \dots & i_{nm} \end{bmatrix} ($$

باید شماره فعالیت‌ها به نحوی باشد که اگر  $\forall i_m \in P(j_m)$  آنگاه  $i_m < j_m$  باشد. یعنی شماره فعالیت‌های پیش‌نیاز می‌بایست کوچکتر از شماره فعالیت‌های پس‌نیازشان باشد. این لیست در ابتدا شامل تمامی فعالیت‌ها بوده و در هر مرحله فعالیت‌ها از آن برداشته شده و با شرایطی در کروموزوم منتقل خواهند شد.

مجموعه یا آرایه Serial که همان بخش دوم کروموزوم است و لیست ترتیبی فعالیت‌های هر پروژه که به صورت تصادفی ایجاد می‌گردد، در هر سطر آرایه قرار می‌گیرد. مجموعه EAS<sup>1</sup> یا مجموعه فعالیت‌های مجاز برای برنامه‌ریزی است. در هر مرحله مجموعه فعالیت‌هایی که پیش‌نیازهای آنها در مراحل قبل در مجموعه Serial قرار گرفته‌اند، به این لیست منتقل می‌شوند.

حال مراحل اجرای الگوریتم به صورت زیر می‌باشد:  
آرایه Serial را به صورت زیر تشکیل دهید.

$$serial = \begin{bmatrix} h_{11} & \dots & h_{n1} \\ \vdots & \ddots & \vdots \\ h_{1m} & \dots & h_{nm} \end{bmatrix} = \begin{bmatrix} null & \dots & null \\ \vdots & \ddots & \vdots \\ null & \dots & null \end{bmatrix}$$

در نظر بگیرید:  $m=1$ .

در نظر بگیرید:  $h=1, q=1$ .

فعالیت یک از سطر  $m$  (پروژه  $m$ ) در Activity List را خارج کنید و به لیست EAS اضافه کنید.

اگر EAS خالی بود به قدم ۱۱ بروید، اگر نه به قدم شش بروید.

یک عدد صحیح با توزیع یکنواخت گسسته بین یک و  $q$  ایجاد کنید و آنرا  $a$  بنامید.

$a$  امین فعالیت داخل EAS را از آن خارج کرده و آنرا در مکان  $h$  ام سطر  $m$  ام آرایه Serial قرار دهید و نام آنرا job بنامید.

در نظر بگیرید:  $h=h+1$  ,  $q=q-1$ .

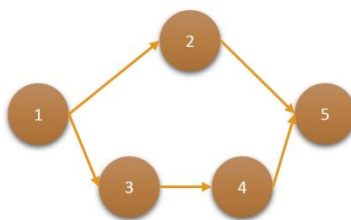
کلیه فعالیت‌های پس‌نیاز فعالیت job را که در Activity List قرار دارد در نظر بگیرید.

اگر  $f$  عدد از آنها فعالیت‌هایی باشند که کلیه پیش‌نیازهای آنها در Serial وجود دارد، آنها را از Activity List خارج کرده و در EAS قرار دهید.

در نظر بگیرید:  $q=q+f$  و به قدم پنج باز گردید.

اگر  $m$  برابر بود با بزرگترین شماره سطر آرایه Activity List (آخرین پروژه) الگوریتم پایان یافته است و از آن خارج شوید، غیر از این  $m$  را برابر با  $m=m+1$  قرار داده و به قدم سه باز گردید.

توسط این الگوریتم  $m$  ترتیب تصادفی در هر کروموزوم ایجاد می‌شود و بدین ترتیب با تکرار این الگوریتم به تعداد دلخواه می‌توان جمعیت اولیه را تشکیل داد. باید توجه کرد که با در نظر گرفتن توزیع یکنواخت گسسته برای انتخاب تصادفی فعالیت‌ها از مجموعه EAS، ترتیب‌های تصادفی ایجاد شده با توزیع یکنواخت ایجاد نمی‌شوند و در واقع شانس حضور فعالیت‌ها در این ترتیب متفاوت می‌باشد. برای مثال پروژه  $m$  که در شکل سه نشان داده شده است را در نظر بگیرید:



شکل ۳: شبکه گره‌ای پروژه  $m$  شامل پنج فعالیت

ابتدا فعالیت یک در مکان اول سطر  $m$  ام آرایه Serial قرار می‌گیرد. سپس فعالیت‌های دو و سه که پس‌نیازهای آن هستند در EAS قرار می‌گیرند. اگر  $P_i^h$  احتمال قرار گرفتن فعالیت  $i$  در مکان  $h$  باشد، خواهیم داشت:

$$P_2^2 = \frac{1}{2}, \quad P_2^3 = \frac{1}{2} * \frac{1}{2} = \frac{1}{4}, \quad P_2^4 = \frac{1}{2} * \frac{1}{2} = \frac{1}{4}$$

همانطور که مشاهده می‌شود فعالیت یک تمایل دارد که زودتر برنامه‌ریزی شده و با احتمال بیشتری در مکان دوم قرار می‌گیرد. برای اینکه توزیع قرارگیری فعالیت‌ها در هر کدام از مکان‌های موجود یکنواخت‌تر شود به هر کدام از فعالیت‌های داخل مجموعه EAS وزن تخصیص می‌دهیم:

$w_i$ : وزن فعالیت  $i$ . اگر وزن فعالیت  $j$ ،  $w_j$  باشد و  $i \in P_j$  آنگاه داریم:

$$w_i = \max_{\forall j | i \in P_j} \{w_j + 1\} \quad (9)$$

اولین وزن را به  $n$  امین فعالیت تخصیص می‌دهیم و آنرا صفر می‌گذاریم. برای شبکه مثال بالا خواهیم داشت:

$$w_1 = 3, \quad w_2 = w_4 = 1, \quad w_3 = 2, \quad w_5 = 0$$

در این صورت اگر در EAS تعداد  $u$  فعالیت وجود داشته باشد و وزن‌های آنها را به ترتیب  $w(1)$  تا  $w(u)$  بنامیم، احتمال انتخاب فعالیت  $i$  مطابق قدم شش الگوریتم به صورت زیر است:

$$P_i = w(i) / \sum_{j=1}^u w(j) \quad (10)$$

بنابراین احتمال قرارگیری فعالیت دو در رشته ترتیبی کروموزوم بعد از ایجاد وزن‌ها به صورت زیر خواهد بود:

$$P_2^2 = \frac{1}{1+2} = \frac{1}{3}, \quad P_2^3 = \frac{2}{3} * \frac{1}{2} = \frac{1}{3}, \quad P_2^4 = 1 - (P_2^2 + P_2^3) = \frac{1}{3}$$

همانطور که مشاهده می‌شود بعد از ایجاد وزن‌ها احتمال قرار گرفتن فعالیت شماره دو در تمام

مکان‌های ممکن یکسان می‌شود. البته ممکن است بعد از وزن دادن فعالیت‌ها با این روش، باز هم توزیع انتخاب فعالیت‌ها در داخل کروموزوم یکنواخت نشود، ولی این روش باعث خواهد شد تا به توزیع یکنواخت نزدیک‌تر شویم.

### رشته سوم کروموزوم: مشخص نمودن سطح منابع [۲]

مساله مورد بررسی در این تحقیق از دسته مسائل سرمایه‌گذاری در منابع بوده و سطح منابع مورد استفاده جزء متغیرهای مساله محسوب می‌شود. بنابراین سطح منابع در یک بازه مشخص و به صورت تصادفی در تمامی نسل‌های الگوریتم ژنتیک ایجاد خواهند شد<sup>۱</sup>. جهت مشخص کردن این بازه نیاز است که مقادیر حد کمینه و بیشینه جهت استفاده هر منبع مشخص شود. اگر حد پایین را با  $R_k$  نشان دهیم، یک حد پایین می‌تواند به صورت زیر بدست آید:

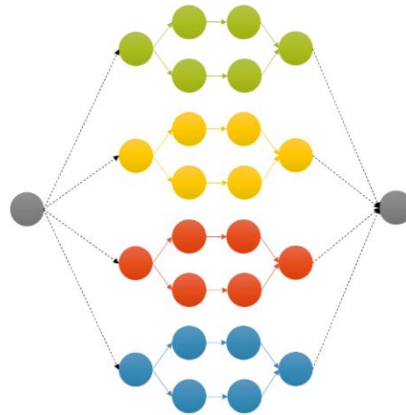
$$\underline{R}_k = \sum_{\forall j} r_{jk} \times D_j \quad (11)$$

ولی می‌توان آن را به این صورت بهبود داد:

$$\underline{R}_k = \text{Max} \left\{ \frac{\sum_{\forall j} r_{jk} \times D_j}{\min\{T, \sum_{\forall j} D_j\}}, \max_{\forall j} \{r_{jk}\} \right\} \quad (12)$$

در این صورت می‌توان حد پایین مناسبی را بدست آورد. ولی نکته‌ای که وجود دارد این است که رابطه ۱۲ برای یک پروژه می‌تواند استفاده شود. در صورتی که مساله مورد نظر یک مساله چندپروژه‌ای است. برای رفع این مشکل باید به این صورت تصور کرد که هر پروژه فعال یک زیرپروژه در پروژه اصلی (که مجموع پروژه‌های انتخاب شده است) است که از منابع به صورت مشترک بهره می‌برند.

۱. عملگرهای جهش و تقاطع برای این بخش از کروموزوم استفاده نخواهند شد. در عوض الگوریتمی جهت بهینه سازی منابع بکار خواهد رفت.



شکل ۴: شبکه گره‌ای شامل چهار زیر پروژه

با این توضیح می‌توان رابطه ۱۲ را برای تمام پروژه‌های انتخاب شده تعمیم داد. در این صورت رابطه ۱۳ حاصل خواهد شد:

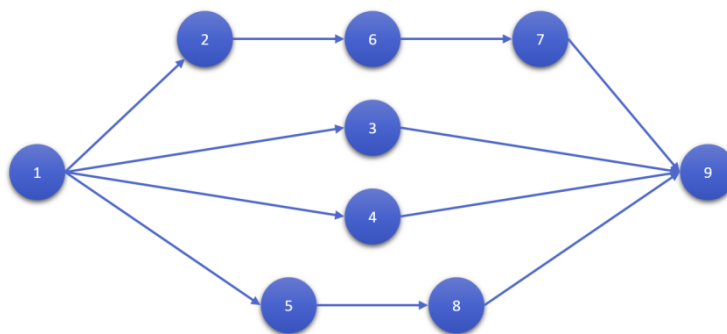
$$\frac{R_k}{=} = \text{Max} \left\{ \frac{\sum_{\forall jm} r_{jmk} \times D_{jm}}{\min\{T, \sum_{\forall jm} D_{jm}\}}, \max_{\forall jm} \{r_{jmk}\} \right\} \quad \text{for active project} \quad (13)$$

در بکارگیری رابطه ۱۳ باید دقت کرد که پروژه‌های غیرفعال در رشته اول کروموزوم نقشی در تعیین حد پایین منبع نوع  $k$  ندارند.

حد بالا را نیز با  $\overline{R}_k$  نشان می‌دهیم و مقدار حداکثر استفاده منبع  $k$  ام در حالتی که برای یک پروژه فعالیت‌ها در زودترین زمان اجرا زمان‌بندی شوند، در نظر گرفته می‌شود. حال برای مساله چندپروژه‌ای این مقدار برای تک تک پروژه‌های فعال حساب شده و سپس مجموع این مقادیر به عنوان حد بالا انتخاب خواهد شد. سپس برای هر منبع  $k$  یک عدد تصادفی در بازه  $[\underline{R}_k, \overline{R}_k]$  انتخاب شده و به عنوان سطح منبع  $k$  ام در نظر گرفته می‌شود. باید توجه کرد که به دلیل وجود پروژه‌های فعال و غیرفعال متفاوت در هر عضو از اعضای جمعیت نسبت به دیگر اعضا، مقدار حد پایین و بالای بازه مورد بحث به صورت جداگانه برای هر کروموزوم محاسبه می‌شود و نمی‌توان از یک بازه برای تمامی کروموزوم‌ها در مساله چندپروژه‌ای استفاده نمود.

## رمزگشایی و برازش کروموزوم‌ها

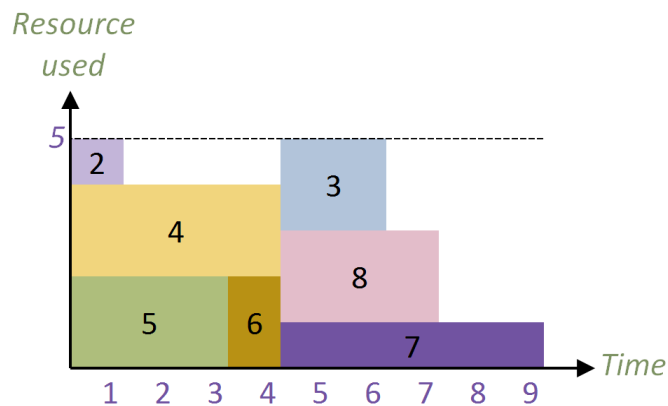
پس از ایجاد هر یک از اعضای جمعیت یا همان کروموزوم با کمک سه رشته توضیح داده شده، هر کدام از آنها یک جواب شدنی برای مساله محسوب می‌شود. اما قبل از آنکه بتوان آنرا برازش نمود می‌بایست به یک برنامه زمان‌بندی تبدیل شده و به اصطلاح رمزگشایی شود. برای این کار از روش تولید برنامه‌های موازی یا به اصطلاح PSGS<sup>۱</sup> استفاده شده است. طبق این روش، فعالیت‌ها در نقاطی تحت عنوان نقاط تصمیم (نقاطی که در آنها اجرای فعالیت‌های زمان‌بندی شده به پایان می‌رسد) طبق توالی مشخص شده در بخش دوم کروموزوم به برنامه زمان‌بندی اضافه می‌شوند. همچنین فعالیت‌های پیش‌نیاز، قبل از این نقاط زمان‌بندی شده‌اند. به عنوان مثال فرض کنید یک ترتیب تصادفی به صورت  $\{1, 2, 6, 5, 7, 4, 8, 3, 9\}$  از شبکه گره‌ای نشان داده شده در شکل پنج موجود است. همچنین سطح منبع در نظر گرفته شده برای زمان‌بندی این شبکه پنج واحد در نظر گرفته شده است.



شکل ۵: شبکه گره‌ای شامل نه فعالیت

پس از اجرای روش PSGS، زمان‌بندی حاصل مطابق با شکل شش خواهد بود:



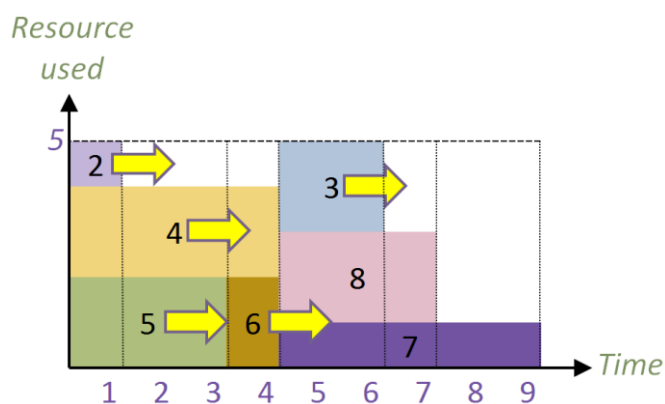


شکل ۶: زمان‌بندی حاصل از روش تولید برنامه‌های موازی

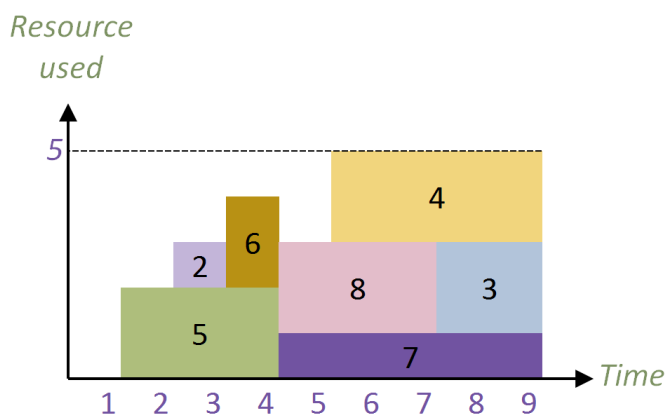
در نهایت بعد از پایان زمان‌بندی با روش تولید برنامه‌های موازی از هر کروموزوم یک جواب شدنی برای مساله مورد نظر حاصل می‌شود. باید توجه کرد که جواب حاصل از نوع برنامه‌های فعال<sup>۱</sup> بوده که برای معیارهای منظم کارایی مانند زودترین زمان اتمام پروژه (Makespan)، بهترین جواب بین جواب‌های فعال قرار دارد [۱]. ولی از آنجایی که معیار تابع هدف این مساله نامنظم است لازم است که فعالیت‌های با جریان مالی منفی و خنثی را شناسایی و تا حد امکان اجرای آنها را به تاخیر انداخت. برای این کار از همان مفهوم نقطه تصمیم استفاده شده است. الگوریتم طراحی شده، از انتهای پروژه یعنی زمان اتمام آن شروع کرده و به سمت چپ محور زمان حرکت می‌کند و فعالیت‌هایی که زمان اتمام آنها در آن نقطه قرار دارد را شناسایی و در صورت داشتن جریان مالی منفی یا خنثی با رعایت شرایط پس‌نیازی و محدودیت منبع اقدام به جابجایی آنها می‌کند<sup>۲</sup>. به این دلیل از آخر شروع می‌کند زیرا فعالیت‌ها قبلاً تا جای ممکن از سمت چپ محور زمان به یکدیگر مماس شده‌اند و باید

۱. برنامه‌ای را فعال می‌گویند که هیچ فعالیتی در آن را نمی‌توان روی محور زمان به سمت چپ حرکت داد.  
 ۲. این الگوریتم طوری طراحی شده که هیچگاه فعالیت موهومی آخر را جابجا نکند. در نتیجه زمان اتمام پروژه دست‌نخورده می‌ماند. به این دلیل که فرض شده درآمد عمده پروژه‌ها در انتهای هر پروژه حاصل می‌شود. حال اگر با تاخیر انداختن آخرین فعالیت، ارزش خالص فعلی پروژه بهبود یابد، می‌توان اینگونه نتیجه گرفت که انتخاب نکردن آن پروژه و عدم اجرای آن در کل بهتر است که این موضوع با بهبود مستمری که در الگوریتم ژنتیک انتظار می‌رود اتفاق خواهد افتاد.

کار جابجا کردن از سمت راست آغاز شده تا به ترتیب فضای خالی برای جابجایی فعالیت‌های ابتدایی‌تر نیز به وجود آید. این الگوریتم بلافاصله بعد از رمزگشایی کروموزم‌ها، برای کلیه اعضای جمعیت و در تمام نسل‌ها به کار می‌رود. به عنوان مثال اگر در برنامه زمان‌بندی حاصل در این بخش فرض شود که فعالیت‌های دو تا پنج هزینه‌زا بوده، فعالیت شش خنثی و فعالیت هفت و هشت نیز دارای جریان مالی مثبت باشند، پس از اجرای این الگوریتم، زمان‌بندی نشان داده شده در شکل هشت بدست خواهد آمد.



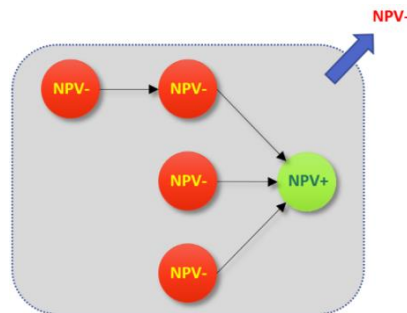
شکل ۷: فعالیت‌های مجاز به جابجایی



شکل ۸: زمان‌بندی ایجاد شده توسط الگوریتم جابجایی فعالیت‌ها

گاهی جابجا کردن یک فعالیت با جریان مالی مثبت به سمت راست یعنی عقب انداختن اجرای آن در کل می‌تواند مفید واقع شود. این به این دلیل است که برآیند هزینه‌های

فعالیت‌های پیش‌نیاز آن با درآمد حاصل از این فعالیت عددی منفی می‌شود. به عبارت بهتر با جابجا کردن آن فعالیت اجازه داده می‌شود تا فعالیت‌های شامل هزینه قبل از آن نیز جابجا شده و در نهایت باعث بهبود ارزش خالص فعلی پروژه خواهد شد. از این رو قبل از اجرای الگوریتم توضیح داده شده در بالا و جابجا کردن فعالیت‌های با جریان مالی منفی، الگوریتمی طراحی شده که فعالیت‌های با جریان مالی مثبت که می‌بایست جابجا شوند را در کلیه پروژه‌های موجود شناسایی می‌کند. شناسایی این نوع فعالیت‌ها با کمک مفاهیمی چون محاسبات زودترین زمان‌های شروع هر پروژه بدون در نظر گرفتن محدودیت منابع، تشکیل درخت اولیه<sup>۱</sup> و محاسبات برآیند ارزش خالص فعلی روی تک تک فعالیت‌های هر پروژه انجام می‌گیرد [۷]. الگوریتم طراحی شده یک بار و در ابتدای اجرای الگوریتم ژنتیک اجرا شده و با کمک آن فعالیت‌های با جریان مالی مثبت حائز شرایط برای جابجایی در هر پروژه، شناسایی شده و علامت‌گذاری می‌شوند تا هنگام اجرای الگوریتم جابجایی فعالیت‌های با جریان مالی منفی به راست، آنها نیز جابجا شوند.



شکل ۹: برآیند جریان‌های مالی مثبت و منفی

### عملگر تقاطع (Crossover Operator)

عملگر تقاطع توسط انتخاب دو کروموزوم (والد) روی بخش دوم کروموزوم یعنی توالی فعالیت‌ها صورت می‌گیرد که نتیجه آن دو کروموزوم جدید (فرزند) است. در این عمل انتظار می‌رود ویژگی‌های مطلوب والدین با یکدیگر ترکیب شده و فرزندان بهتری حاصل شود. از

آنجایی که مساله مورد نظر چند پروژه‌ای است، عمل تقاطع بین دو والد به صورت مجزا بین توالی فعالیت‌های پروژه‌های یکسان انجام می‌گیرد. یعنی پروژه اول از والد اول با پروژه اول از والد دوم تا پروژه  $m$ ام از والد اول و پروژه  $m$ ام از والد دوم. همچنین برای فرزند اول عمل تقاطع روی توالی پروژه‌های فعال والد یک و برای فرزند دوم این عمل بر روی توالی پروژه‌های فعال والد دو انجام خواهد شد. در نتیجه فرزند اول، پروژه‌های فعال و سطح منابع تعیین شده را از والد اول به ارث برده و فرزند دوم نیز آنها را از والد دوم به ارث می‌برد. در ادامه مثالی با استفاده از روش تقاطع یک نقطه‌ای (Single Crossover) ارائه شده است. فرض کنید دو والد که شامل چهار پروژه و هر پروژه شامل ۱۰ فعالیت است وجود دارد. در والد اول پروژه‌های یک و سه فعال و در والد دوم نیز پروژه‌های دو و چهار فعال هستند.

$$P1: \begin{pmatrix} 1 & 4 & 2 & 3 & 7 & 6 & 5 & 8 & 9 & 10 \\ 1 & 3 & 2 & 5 & 4 & 7 & 8 & 6 & 9 & 10 \\ 1 & 2 & 5 & 3 & 4 & 6 & 8 & 9 & 7 & 10 \\ 1 & 2 & 4 & 5 & 3 & 7 & 6 & 9 & 8 & 10 \end{pmatrix} \quad \text{projects 1 and 3 is activated}$$

$$P2: \begin{pmatrix} 1 & 5 & 3 & 4 & 2 & 6 & 7 & 8 & 9 & 10 \\ 1 & 2 & 3 & 6 & 7 & 4 & 8 & 5 & 9 & 10 \\ 1 & 4 & 3 & 6 & 2 & 5 & 7 & 8 & 9 & 10 \\ 1 & 5 & 3 & 4 & 2 & 6 & 8 & 7 & 9 & 10 \end{pmatrix} \quad \text{projects 2 and 4 is activated}$$

شکل ۱۰: رشته دوم کروموزوم (توالی فعالیت‌ها) والد اول و دوم

برای ایجاد فرزند اول، عمل تقاطع روی توالی پروژه‌های اول و سوم هر دو والد انجام خواهد گرفت. اگر فرض شود طول توالی پروژه  $m$  مقدار  $n$  باشد برای هر پروژه فعال یک عدد تصادفی در بازه  $[1, n-1]$  انتخاب می‌شود. اگر عدد تصادفی ایجاد شده برای پروژه  $m$  را  $cpm^1$  بنامیم، اعضای یک تا  $cpm$  از توالی فعالیت‌های پروژه  $m$  عیناً از والد اول در فرزند اول قرار گرفته و اعضای  $cpm+1$  تا  $n$  نیز از این توالی با ترتیبی که در توالی آن پروژه در والد دوم می‌باشد، اضافه خواهند شد. به طور مثال اگر نقطه برش برای پروژه اول عدد چهار باشد، توالی ایجاد شده برای پروژه اول در فرزند اول به صورت زیر خواهد شد.

## 1. Cut Point

$$Y1: \begin{pmatrix} 1 & 4 & 2 & 3 & 5 & 6 & 7 & 8 & 9 & 10 \end{pmatrix}$$

شکل ۱۱: رشته دوم کروموزوم فرزند اول و پروژه اول

عمل بالا با ایجاد یک عدد تصادفی در بازه  $[1, 9]$  برای پروژه سوم نیز انجام خواهد شد و بدین ترتیب توالی پروژه سوم فرزند اول نیز مشخص می‌شود. توالی فعالیت‌های پروژه‌های دو و چهار که در والد اول غیرفعال است نیز بدون هیچ تغییری در فرزند اول قرار می‌گیرد. به طور مشابه برای ایجاد فرزند دوم نیز از آنجا که پروژه‌های دو و چهار در والد دوم فعال است، عمل تقاطع روی آن پروژه‌ها انجام خواهد گرفت و توالی فعالیت‌های پروژه‌های یک و سه که در والد دوم غیرفعال است نیز عیناً در فرزند دوم قرار می‌گیرد.

جهت اجرای تقاطع دونقطه‌ای (Double Crossover) نیز تمامی توضیحات قبل صادق بوده با این تفاوت که به جای یک نقطه برش در بازه  $[1, n-1]$ ، برای هر پروژه فعال در این بازه دو نقطه برش متفاوت به صورت تصادفی ایجاد می‌شود. اگر نقاط برش را  $cp1_m$  و  $cp2_m$  بنامیم ( $cp1_m < cp2_m$ )، اعضای فاصله  $cp1_m+1$  تا  $cp2_m$  از پروژه فعال  $m$  در والد اول عیناً در توالی فرزند اول قرار گرفته و اعضای فواصل یک تا  $cp1_m$  و  $cp2_m+1$  تا  $n$  از والد اول با ترتیبی که در والد دوم قرار دارد در فرزند اول قرار می‌گیرد.

حال این سوال مطرح است که کدام روش بهتر عمل می‌کند. موضوع این است که برای مسائل مختلف ممکن است هر روش نسبت به دیگری برتری داشته باشد. از این رو برای مساله مورد بررسی در این تحقیق نیز باید یکی از دو روش تقاطع را انتخاب نمود. بدین منظور آزمایشی ترتیب داده شد. بدین صورت که ۳۰ مساله متفاوت در اندازه‌های مختلف ایجاد گردید<sup>۱</sup>. هر مساله با شرایط کاملاً یکسان یک بار با کاربرد تقاطع یک‌نقطه‌ای و یک بار نیز با کاربرد تقاطع دونقطه‌ای در الگوریتم طراحی شده اجرا و حل گردید. سپس مشاهده شد که میانگین حاصل از حل مسائل با روش تقاطع دونقطه‌ای کمی بیش از حل آنها با روش تقاطع

۱. تولید پروژه‌های اتفاقی با محدودیت منابع توسط نرم‌افزار Rangen انجام شده است. ولی هزینه‌ها و درآمدهای

تعریف شده برای فعالیت‌ها توسط نگارنده این تحقیق مشخص گردیده‌اند.

یک نقطه‌ای است. سپس با تشکیل یک آزمون فرض و با استفاده از آزمون تی زوجی (Paired t-Test) فرضیه‌ها سنجیده شدند و مشخص گردید که تفاوت معناداری بین میانگین‌های هر دو روش وجود ندارد. از این رو این نتیجه حاصل شد که انتخاب روش یک نقطه‌ای یا دو نقطه‌ای برای عمل تقاطع، اولویتی نسبت به یکدیگر نخواهند داشت<sup>۱</sup>.

### عملگر جهش (Mutation Operator)

جهت دستیابی به نقاط خوب احتمالی در فضای جواب از عملگر جهش استفاده می‌شود. ماهیت عمل جهش به نوعی دگرگون کردن جواب‌های کنونی است و در بیشتر مواقع منجر به جواب‌های خوبی نخواهد شد. ولی اگر با موفقیت همراه باشد معمولاً می‌تواند تاثیر بسزایی روی تابع هدف گذاشته و فضای جدیدی را در فضای جواب‌ها بگشاید. بنابراین جهت اینکه عملگر جهش بتواند تغییرات عمده‌ای در کروموزوم‌های موجود ایجاد کند، این عملگر روی رشته اول کروموزوم یعنی انتخاب پروژه‌ها به کار گرفته شده است.

در اینجا تنها به یک والد نیاز است و حاصل این عمل نیز یک فرزند جهش یافته خواهد بود. در عمل جهش تعداد مشخصی از پروژه‌های فعال در کروموزوم والد غیر فعال شده و همان تعداد نیز از پروژه‌های غیر فعال به فعال تبدیل می‌شوند. انتخاب پروژه‌های فعال و غیر فعال کاملاً تصادفی و با احتمال‌های یکسان است. اگر تعداد این پروژه‌ها در کروموزوم والد از تعداد مشخص شده کمتر باشد همان تعدادی که وجود دارد برای تبدیل در نظر گرفته می‌شود.

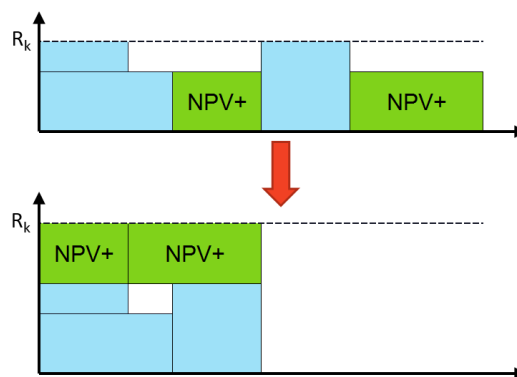
بعد از انجام عمل جهش، پروژه‌های فعال و غیر فعال متفاوتی در رشته اول کروموزوم نسبت به کروموزوم والد به وجود خواهد آمد. همچنین رشته دوم کروموزوم والد یعنی توالی فعالیت‌ها بدون تغییر به فرزند منتقل خواهد شد. ولی در مورد رشته سوم یعنی سطح منابع، به دلیل اینکه در کروموزوم فرزند، پروژه‌های فعال و غیرفعال جدیدی ایجاد شده، بنابراین سطح منابع تصادفی ایجاد شده در والد برای فرزند قابل استفاده نبوده و الگوریتم تعیین تصادفی مشخص نمودن سطح منابع، مجدداً اجرا شده و سطح منابع برای هر فرزند به صورت تصادفی

۱. در این تحقیق از روش تقاطع دو نقطه‌ای استفاده شده است.

ایجاد می‌گردد. جهت انجام عمل جهش در نسل‌های مختلف، دو پارامتر قابل تنظیم در نظر گرفته شده است. یکی جهت درصد انجام این عمل نسبت به اندازه جمعیت بوده و دیگری درصد تعداد پروژه‌های انتخاب شده نسبت به تعداد کل پروژه‌ها جهت دگرگون سازی را کنترل می‌کند.

### عملگر جستجوی محلی در حوزه منابع

طبق توضیحات قبل، سطوح منابع تجدیدپذیر در بخش سوم کروموزوم در بازه‌هایی که حدود پایین و بالای آن مشخص شده‌اند به صورت تصادفی ایجاد می‌شوند. بنابراین ممکن است یک کروموزوم صفات خوبی را در رشته اول و دوم خود دارا بوده ولی به دلیل انتخاب نامناسب سطح منابع در نهایت منجر به جواب خوبی نشود. به عنوان مثال یک کروموزوم که در آن سطح منبع انتخاب شده بیش از حد پایین بوده و باعث طولانی شدن زمان اجرای پروژه و از دست دادن سود زیادی می‌شود. یا برعکس سطح منابع خیلی بالا در نظر گرفته شده است و باعث هدر رفتن سرمایه زیادی جهت استخدام منابع می‌شود.



شکل ۱۲: زمان‌بندی یک پروژه، قبل و بعد از اصلاح مقدار منبع  $R_k$

جهت یافتن یک سطح منبع بهتر در بازه مورد نظر، ابتدا از یک الگوریتم کاهش استفاده می‌شود. بدین صورت که سطح منبع  $k$  از کروموزوم والد را یک واحد کاهش داده سپس کروموزوم موجود با سطح منبع کاهش یافته رمزگشایی و برازش می‌شود. اگر در همین مرحله مقدار برازش شده تابع هدف از مقدار اولیه آن (کروموزوم والد) کمتر بود، این نتیجه حاصل

می‌شود که کاهش سطح منبع تاثیر مخربی در درآمد حاصل از اجرای پروژه‌ها داشته و بنابراین الگوریتم افزایشی وارد عمل می‌شود. ولی اگر مقدار برازش شده از مقدار اولیه آن بهتر شد، الگوریتم مجدداً به کاهش سطح منبع ادامه می‌دهد و این عمل تکرار شده تا جایی که این کاهش منجر به بهبود درآمد نشده و مجدداً مقدار برازش شده کاهش یابد. در اینجا الگوریتم متوقف شده و سطح منبع جدید در کروموزوم فرزند، جایگزین سطح منبع اولیه خواهد شد.

در الگوریتم افزایشی نیز سطح منبع یک واحد یک واحد افزایش یافته و روند کار مشابه با الگوریتم کاهشی است. در نهایت الگوریتم کاهشی یا افزایشی برای همه منابع موجود در کروموزوم‌های والد اجرا شده تا بهترین سطح منابع برای آنها مشخص شوند و این صفت به فرزندان منتقل می‌شود. جهت انجام عمل جستجوی محلی در حوزه منابع در نسل‌های مختلف، پارامتری قابل تنظیم در نظر گرفته شده است که درصد انجام این عمل نسبت به اندازه جمعیت را مشخص می‌کند.

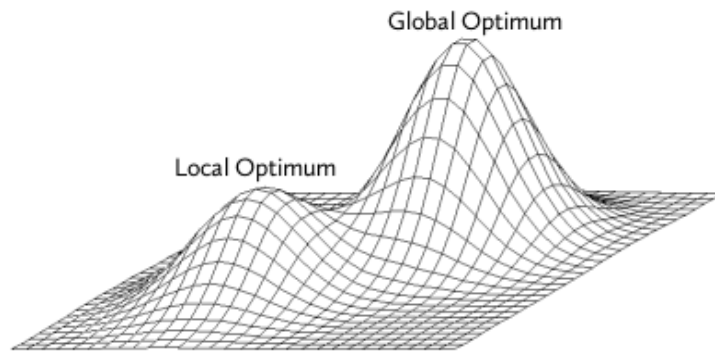
### ترکیب جمعیت

تاکنون عملگرهای استفاده شده در الگوریتم ژنتیک طراحی شده برای این تحقیق توضیح داده شد. در هر نسل هر کدام از این عملگرها با توجه به پارامترهای تنظیم شده برای آنها، تعدادی فرزند با مقادیر برازش شده مختلف پدید می‌آورند. یعنی علاوه بر جمعیت قبلی که از نسل قبل وارد شده‌اند، افراد دیگری نیز به آنها افزوده می‌شود. ولی با توجه به پارامتر تنظیم شده برای اندازه جمعیت، می‌بایست جهت رفتن به نسل‌های بعدی اندازه جمعیت موجود حفظ شود. در غیر این صورت در گذار به هر نسل تعداد افراد جمعیت به صورت کنترل نشده‌ای افزایش خواهد یافت.

برای این کار در هر تکرار از الگوریتم، افراد ایجاد شده از طریق تمامی عملگرها و جمعیت وارد شده از نسل قبل روی هم تجمیع شده، بر اساس مقدار برازش شده مرتب می‌شوند و سپس اگر اندازه جمعیت npop تعریف شده باشد، تعداد npop تا از بهترین جواب‌های موجود نگه داشته شده و مابقی حذف می‌شوند. به این عمل نخبه‌گرایی (Intensification)



گفته می‌شود. ولی استفاده از رویکرد نخبه‌گرایی به تنهایی صحیح نیست. زیرا الگوریتم به سرعت در یک فضای بهینه محلی (Local Optimum) گیر افتاده و ایجاد افراد جدید در جمعیت حاضر در اطراف همین محل رخ می‌دهد که در این صورت شانس دستیابی به دیگر نقاط از فضای جواب بسیار کم می‌شود.



شکل ۱۳: نقاط بهینه سراسری و محلی

همچنین با شروع از یک نسل اولیه و ایجاد نسل‌های بعدی توسط تقاطع و جهش ممکن است بعضی از ویژگی‌های مطلوب بالقوه دیده نشده باشد [۱]. بنابراین نیاز است که در هر نسل تعداد مشخصی از افراد جمعیت همانند جمعیت اولیه به صورت تصادفی وارد جمعیت جدید گردند. این کار باعث بهبود تنوع‌گرایی (Diversification) در حل مساله خواهد شد و اثر نخبه‌گرایی را تعدیل خواهد کرد. البته در رویکرد تنوع‌گرایی هم می‌بایست مراقب بود تا جواب‌ها بیش از حد متفرق نشوند. بنابراین باید سعی شود بین این دو دیدگاه تعادل برقرار باشد.

جهت ایجاد کروموزوم‌ها به صورت تصادفی در نسل‌های مختلف، پارامتری قابل تنظیم در نظر گرفته شده است که درصد انجام این عمل نسبت به اندازه جمعیت را مشخص می‌کند. پس از ایجاد کروموزوم‌های تصادفی، به تعداد این کروموزوم‌ها، افراد بدتر جمعیت بدست آمده از طریق نخبه‌گرایی، حذف شده و جای خود را به کروموزوم‌های جدید خواهند داد. با این کار فرصت بهبودهای بالقوه در نسل‌های بعد به وجود خواهد آمد. زیرا ممکن است کروموزوم‌های تصادفی جدید، دارای صفاتی باشند که تا تکرار کنونی در هیچ یک از اعضای جمعیت دیده نشده است.

## بررسی نتایج محاسباتی

جهت اطمینان از عملکرد صحیح الگوریتم ژنتیک طراحی شده در بخش قبل، مثال‌هایی ایجاد و با این الگوریتم حل شده است. سپس جواب‌ها تحلیل و بررسی شده‌اند. ولی قبل از آن، نیاز است که مقدار مناسب پارامترهای مختلف این الگوریتم مانند نرخ عمل تقاطع، جهش و دیگر عملگرها را مشخص نمود. زیرا این پارامترها تاثیر بسزایی روی عملکرد الگوریتم ایجاد شده دارند و تنظیم صحیح آنها می‌تواند به جواب‌های بهتر بیانجامد. جهت انجام این کار از طراحی آزمایش‌ها به روش تاگوچی (Taguchi) بهره برده شد. در روش تاگوچی توسط تعدادی آزمایش مشخص می‌توان تاثیر عوامل مختلف در آزمایش را سنجید و سطوح بهینه آنها را پیش‌بینی کرد. در این روش می‌بایست پارامترهای تاثیرگذار در مساله را شناسایی نمود تا بعداً آنها را به صورت گروهی آزمایش کرد. همچنین چند سطح برای هر پارامتر تعریف می‌شود. به همین منظور پارامترهای تاثیرگذار در الگوریتم ژنتیک همراه با سطوح مختلف مطابق جدول ۱ مشخص است. در اینجا شش پارامتر و سه سطح تعریف شده است.

جدول ۱- پارامترهای تاثیرگذار در الگوریتم ژنتیک و سطوح انتخاب شده جهت آزمایش تاگوچی

سطح	اندازه جمعیت و تعداد نسل	درصد تقاطع	درصد جهش	درصد جستجوی محلی در حوزه منابع	درصد جمعیت تصادفی	فشار انتخاب
۱	۲۴-۴	۰/۵	۰/۱	۰/۱	۰/۲	۲
۲	۱۲-۸	۰/۷	۰/۱۵	۰/۲	۰/۳	۴
۳	۶-۱۶	۰/۹	۰/۲	۰/۳	۰/۴	۶

دو پارامتر تاثیرگذار در الگوریتم، اندازه جمعیت و تعداد نسل‌ها می‌باشد. بدیهی است که هرچه مقدار آنها بزرگتر باشد، احتمال پیدایش جواب‌های خوب بیشتر خواهد شد و در آزمایش با روش تاگوچی نیز اگر آنها را جداگانه بررسی کنیم، به همین نتیجه خواهیم رسید. ولی از جهتی بیش از حد بزرگ گرفتن این دو مقدار، سرعت و انعطاف‌پذیری الگوریتم ژنتیک را پایین می‌آورد. همینطور هرچه جواب‌های بهتری پیدا شوند، مقدار بهبود آنها نسبت

به جواب قبلی کمتر و کمتر می‌شود. بنابراین برای یک طراح مهم است که در زمان معقول و صرف هزینه کمتر به جوابی خوب دست یابد و دیگر بهبودی اندک در جواب با صرف زمانی زیاد، منطقی به نظر نمی‌رسد. بنابراین بجای آزمایش جداگانه این دو پارامتر، آنها را در تقابل با یکدیگر قرار داده تا مشخص شود کدام یک بیشتر از دیگری موثر است. به این صورت که مطابق جدول بالا در هر سطح، مقدار یکی افزایش می‌یابد و دیگری کاهش. همچنین با توجه به تجاربی که از حل مثال‌های مختلف با الگوریتم ژنتیک طراحی شده توسط نگارنده این تحقیق حاصل شده، مقدار هر دو پارامتر به حد لازم بزرگ در نظر گرفته نشده است تا جواب‌ها در آزمایش‌های طراحی شده به حد بلوغ نرسند و بتوان تاثیر آنها را بهتر مشاهده کرد.

برای شروع طراحی آزمایش‌ها با روش تاگوچی، توسط نرم‌افزار مینی‌تب (Minitab)، با وارد کردن تعداد پارامترها و سطوح در آن، مشخص می‌شود که می‌بایست طبق جدول اوردگونال (Orthogonal) تعداد ۲۷ آزمایش مختلف با شرایطی که این نرم‌افزار مشخص کرده انجام گردد. جدول ۲ آزمایش‌ها با مقادیر مختلفی که برای هر پارامتر در نظر گرفته شده را نشان می‌دهد.

سپس جهت انجام آزمایش‌ها، یک مساله با تعداد پنج پروژه که هر کدام شامل ۱۲ فعالیت است، در نظر گرفته شده. هر آزمایش ۳۰ بار انجام گردید تا میانگین دقیقی از هر کدام بدست آید. یعنی در مجموع ۸۱۰ آزمایش جداگانه صورت گرفت. میانگین حاصل از تمامی آزمایش‌ها در جدول ۳ نشان داده شده است.

جدول ۲- آزمایش‌های پیشنهاد شده برای اجرای روش تاگوچی در نرم‌افزار Minitab

test	Population size & Iteration	Crossover	Mutation	Optimum resource	Random population	Selection pressure
1	4 - 18	0.5	0.1	0.1	0.2	2
2	4 - 18	0.5	0.1	0.1	0.3	4
3	4 - 18	0.5	0.1	0.1	0.4	6
4	4 - 18	0.7	0.15	0.2	0.2	2
5	4 - 18	0.7	0.15	0.2	0.3	4
6	4 - 18	0.7	0.15	0.2	0.4	6
7	4 - 18	0.9	0.2	0.3	0.2	2
8	4 - 18	0.9	0.2	0.3	0.3	4

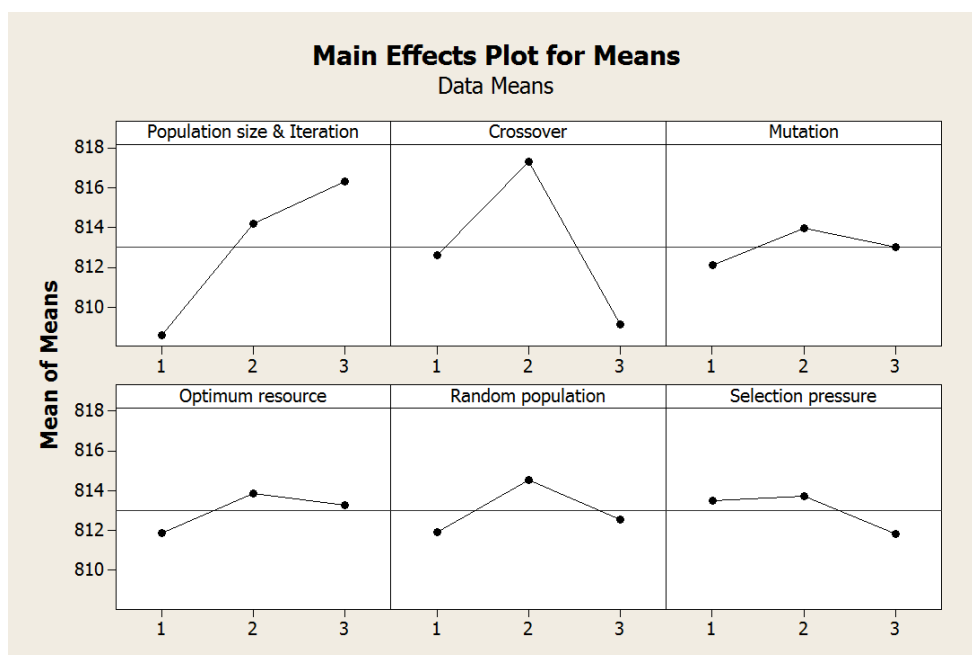
9	4 - 18	0.9	0.2	0.3	0.4	6
10	8 - 12	0.5	0.15	0.3	0.2	4
11	8 - 12	0.5	0.15	0.3	0.3	6
12	8 - 12	0.5	0.15	0.3	0.4	2
13	8 - 12	0.7	0.2	0.1	0.2	4
14	8 - 12	0.7	0.2	0.1	0.3	6
15	8 - 12	0.7	0.2	0.1	0.4	2
16	8 - 12	0.9	0.1	0.2	0.2	4
17	8 - 12	0.9	0.1	0.2	0.3	6
18	8 - 12	0.9	0.1	0.2	0.4	2
19	16 - 6	0.5	0.2	0.2	0.2	6
20	16 - 6	0.5	0.2	0.2	0.3	2
21	16 - 6	0.5	0.2	0.2	0.4	4
22	16 - 6	0.7	0.1	0.3	0.2	6
23	16 - 6	0.7	0.1	0.3	0.3	2
24	16 - 6	0.7	0.1	0.3	0.4	4
25	16 - 6	0.9	0.15	0.1	0.2	6
26	16 - 6	0.9	0.15	0.1	0.3	2
27	16 - 6	0.9	0.15	0.1	0.4	4

جدول ۳- نتایج حاصل از آزمایش‌های انجام شده در روش ناگوچی با الگوریتم ژنتیک

test	Average of results
1	805.4765
2	812.8192
3	800.1379
4	814.6585
5	821.3124
6	807.8179
7	802.5340
8	807.5674
9	804.6662
10	810.7553
11	814.8251
12	819.4187
13	815.9449
14	817.7931
15	818.2103
16	806.6279
17	809.7987

18	814.2769
19	815.9258
20	818.1811
21	816.0246
22	818.8654
23	820.8626
24	820.1862
25	816.5653
26	807.9241
27	812.1822

سپس جواب هر ۲۷ آزمایش به مینی‌تب منتقل شده تا نتایج تحلیل پارامترها مشخص گردند. یکی از مهم‌ترین نمودارهایی که این نرم‌افزار ارائه می‌دهد، نسبت‌های سیگنال به نویز نام دارد که به اختصار به آن اس‌ان‌آر (Signal to Noise Ratios) گفته می‌شود و می‌توان از آن جهت پیدا کردن مقادیر بهتر پارامترها استفاده نمود. شکل ۱۴ این نمودار را نشان می‌دهد که به صورت جداگانه برای هر شش پارامتر رسم شده است. محورهای افقی تعداد سطوح آزمایش را نشان می‌دهند و هرچه مقدار اس‌ان‌آر بیشتر باشد بهتر است.



شکل ۱۴- نمودار SNR حاصل از اجرای آزمایش‌های تاگوچی در Minitab

در نتیجه مقادیر بهینه برای سطوح در نظر گرفته شده سلول‌های رنگی جدول ۴ خواهد بود.

جدول ۴- مقادیر بهینه پارامترهای تاثیرگذار در الگوریتم ژنتیک

سطح	اندازه جمعیت و تعداد نسل	درصد تقاطع	درصد جهش	درصد جستجوی محلی در حوزه منابع	درصد جمعیت تصادفی	فشار انتخاب
۱	۲۴-۴	۰/۵	۰/۱	۰/۱	۰/۲	۲
۲	۱۲-۸	۰/۷	۰/۱۵	۰/۲	۰/۳	۴
۳	۶-۱۶	۰/۹	۰/۲	۰/۳	۰/۴	۶

در ارتباط با پارامتر اندازه جمعیت و تعداد نسل، این نتیجه حاصل می‌شود که اندازه جمعیت نسبت به تعداد نسل‌ها بیشتر تاثیرگذار می‌باشد. از آنجایی که اندازه این دو پارامتر به اندازه مساله نیز بستگی داشته و مساله آزمایش شده در اینجا از نوع کوچک محسوب می‌شود، لذا برای حل مسائل متوسط و بزرگ، نسبت گرفته خواهد شد. جدول ۵ مقدار در نظر گرفته شده

برای حل انواع مسائل با این دو پارامتر را نشان می‌دهد. در انتخاب این مقادیر سعی شده تا حد امکان بزرگ در نظر گرفته شوند.

جدول ۵- مقادیر در نظر گرفته شده برای اندازه جمعیت و تعداد نسل‌ها جهت انجام آزمایش‌ها

اندازه نسل‌ها	اندازه جمعیت	اندازه مساله
40	20	کوچک
60	40	متوسط
80	80	بزرگ

بعد از مشخص شدن مقادیر پارامترهای الگوریتم، نوبت به حل مسائل خواهد رسید. برای این منظور تعداد ۹۰ آزمایش شامل ۳۰ آزمایش کوچک، ۳۰ آزمایش متوسط و ۳۰ آزمایش بزرگ با کمک نرم‌افزار Rangen ایجاد گردید تا کارایی این الگوریتم در طیف مختلفی از مسائل مورد آزمایش قرار گیرد. سپس برای فعالیت‌هایی در هر پروژه، جریان‌های مالی منفی و مثبت در نظر گرفته شد تا شرایط هر پروژه مطابق با مفروضات در نظر گرفته شده باشد. تعداد منابع تجدیدپذیر بکار رفته نیز یک عدد است که هزینه استخدام هر واحد آن ۳۰ واحد پولی در نظر گرفته شده و پارامتر نرخ بهره هم ۰/۰۲ لحاظ شده است. مدت زمان مجاز اجرای پروژه‌ها: مسائل کوچک ۴۰ روز، مسائل متوسط ۱۲۵ روز و مسائل بزرگ نیز ۱۵۰ روز می‌باشد. سایر پارامترها نیز مطابق نتایج حاصل از روش آزمایش‌های تاگوجی تنظیم شده‌اند. در مسائل کوچک هر مساله شامل پنج پروژه و هر پروژه نیز شامل ۱۲ فعالیت می‌باشد. همچنین این ۳۰ مساله یک بار دیگر نیز توسط نرم‌افزار لینگو (Lingo) با مدل ریاضی ارائه شده در بخش ۲-۴، حل گردیدند تا بتوان مقایسه‌ای با جواب‌های حاصل از الگوریتم ژنتیک داشت. پس از حل این مسائل توسط الگوریتم ژنتیک، در اغلب موارد جواب‌های حاصل به طور محسوسی بهتر از جواب‌های بدست آمده در این نرم‌افزار بود<sup>۱</sup> که در کل متوسط رشد

۱. در ارتباط با اختلاف بین میانگین جواب‌های حاصل از الگوریتم ژنتیک و حل با نرم‌افزار لینگو، با استفاده از آزمون t زوجی نیز تحقیق گردید که نشان از اختلاف معناداری در سطح ۹۵٪ بین هر دو میانگین داشت و نشان می‌داد میانگین حاصل از حل در الگوریتم ژنتیک به طور معناداری از حل در نرم‌افزار لینگو بهتر است.

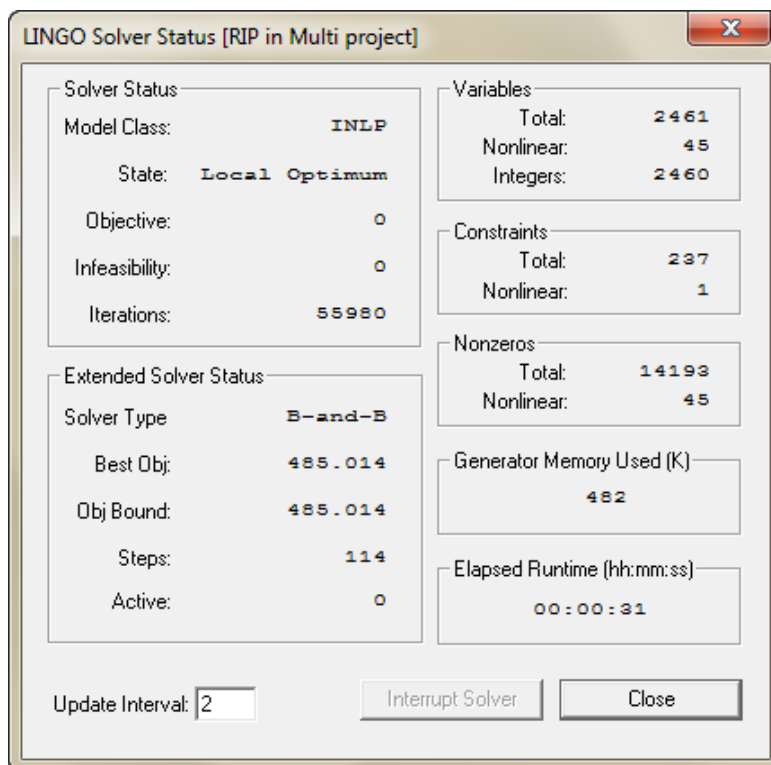
۲۷/۱٪ بهبود را در جواب‌ها نشان می‌دهد. باید به این نکته اشاره کرد که لینگو روش شاخه و کران را برای حل این نوع مساله بر می‌گزیند؛ در نتیجه اغلب جواب‌هایی که از آن حاصل می‌شوند از نوع بهینه محلی (Local Optimum) بوده و نه بهینه سراسری (Global Optimum). ولی توسط الگوریتم ژنتیک می‌توان به جواب‌های بهتری نیز دست یافت.

جدول ۶- نتایج حاصل از حل ۳۰ مساله با اندازه کوچک توسط الگوریتم ژنتیک و نرم‌افزار Lingo

مساله	جواب نهایی توسط الگوریتم ژنتیک	جواب نهایی توسط نرم افزار لینگو
1	624.9258	485.0142
2	150.4217	162.8490
3	213.7156	182.3320
4	100.6816	115.7400
5	543.4797	544.6970
6	216.4399	216.4400
7	365.4235	376.4820
8	80.8506	80.8506
9	399.8346	131.7690
10	275.6600	269.8350
11	135.0298	135.0300
12	476.9912	280.6460
13	86.7855	132.1090
14	594.7716	470.7740
15	163.8117	118.4600
16	368.0705	368.0710
17	380.8388	380.8390
18	274.1039	67.4388
19	346.4955	241.6190
20	51.0605	51.0605
21	328.5948	283.1480
22	366.3351	250.8660
23	38.3537	38.3537
24	515.9634	106.4200
25	643.8240	518.2330
26	544.0112	517.3940
27	615.6532	414.5740
28	490.3852	383.1770
29	350.8512	228.4570
30	228.5441	233.7450
<b>میانگین</b>	<b>332.3969</b>	<b>259.5475</b>

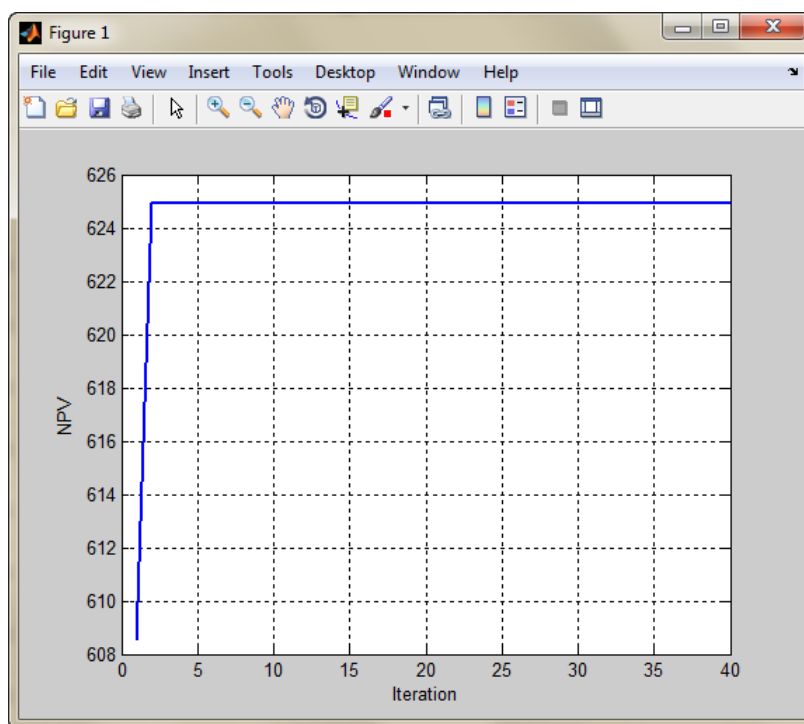


به عنوان نمونه شکل ۱۵، تصویر جواب نهایی حاصل از حل مدل ریاضی مساله شماره ۱ در نرم‌افزار لینگو است:



شکل ۱۵- تصویر جواب مساله شماره ۱ در لینگو

همچنین تصویر زیر جواب نهایی حاصل از حل همین مساله توسط الگوریتم ژنتیک را نشان می‌دهد.



شکل ۱۶- تصویر جواب مساله شماره ۱ توسط الگوریتم ژنتیک

همانطور که در تصویر بالا مشخص است، الگوریتم ژنتیک در ارتباط با این مساله با سرعتی بالا (در تکرار دوم) بهترین جواب را یافته است که تا نسل ۴۰ نیز تغییری نداشته و جوابی به مراتب بهتر از حل مدل ریاضی در لینگو است. جهت نمونه کد مدل ریاضی این مساله در لینگو در پایین درج شده است:

```

model:
sets:
  Projects/m1,m2,m3,m4,m5/:x;
  Resources/k1/:Rk,Ck;
  Tasks/n1,n2,n3,n4,n5,n6,n7,n8,n9,n10,n11,n12/;
  Time/1..40/:t;
  Project_Task(Projects,Tasks):S,d,E,O;
  Needs_to_resources(Project_Task,Resources):r;
  Y_index(Project_Task,Time):y;
  Precedences(Project_Task,Project_Task)/
    
```

						m1,n1,m1 ,n5	m1,n1,m1 ,n4		m1,n1,m1 ,n2
								m1,n2,m1 ,n3	
			m1,n3,m1 ,n8		m1,n3,m1 ,n6				
			m1,n4,m1 ,n8	m1,n4,m1 ,n7					





30	0	80	0	50	300	0	50	0	120	100	0
50	0	40	30	0	90	80	70	100	0	50	0
90	70	20	10	10	0	80	70	40	50	40	0
0	80	80	0	150	0	120	100	0	70	60	0
0	0	300	0	120	0	100	0	10	10	40	0

r= ;

0	6	6	5	4	3	5	2	4	4	1	0
0	7	2	7	2	2	1	5	5	8	1	0
0	7	3	4	4	3	4	2	8	4	1	0
0	3	6	5	2	7	2	4	5	5	1	0
0	6	3	6	6	4	4	3	3	4	1	0

;

enddata

```
max=@sum(Projects(i):@sum(Tasks(j):x(i)*E(i,j)*@exp(-
a*(S(i,j)+d(i,j))))
-@sum(Projects(i):@sum(Tasks(j):x(i)*O(i,j)*@exp(-a*S(i,j))))
-@sum(Resources(i):Ck(i)*Rk(i));
```

```
@for(Project_Task(i,j)|j#EQ#1:S(i,j)=0);
@for(Project_Task(i,j)|j#EQ#10:S(i,j)=>(size)time));
@for(Precedences(g,h,i,j):S(i,j)-S(g,h)>=x(i)*d(g,h));
@for(Projects(i@:(for)tasks(j@:(sum)Time(l):y(i,j,l))=x(i)));
@for(Projects(i@:(for)tasks(j@:(sum)Time(l):t(l)*y(i,j,l))=s(i,j
)));
@for(Resources(z):@for(Time(l):@sum)Projects(i@:(sum)Tasks(j@:(su
m)Time(u)|u#GE#(l-
d(i,j)+1))#and#(u#LE#l):r(i,j,z)*y(i,j,u)))<=Rk(z));
@for(Projects(i@:(BIN)x(i);((
@for(Y_index(i,j,l@:(BIN)y(i,j,l)));
@for(Project_Task) i,j@:(GIN)s(i,j)));
```

end

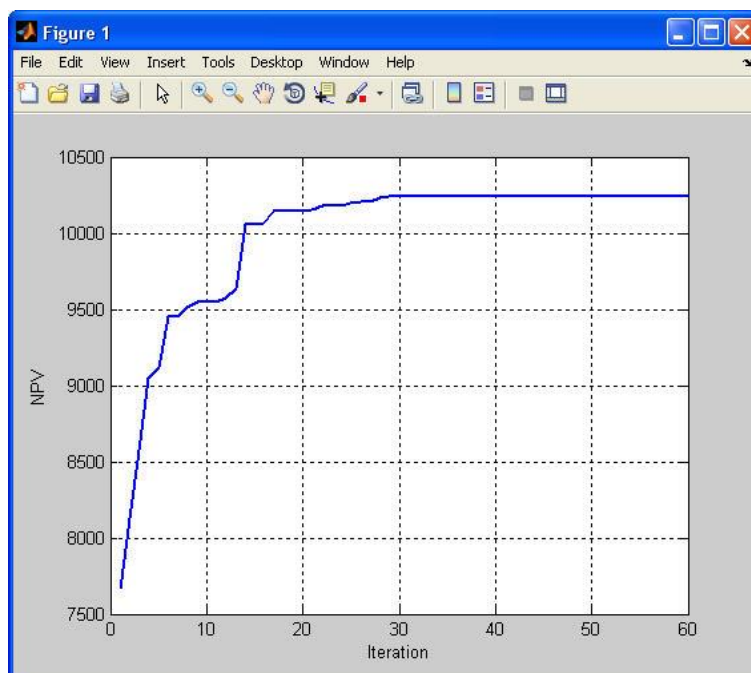
مسائل متوسط هر کدام شامل ۱۰ پروژه و هر پروژه شامل ۳۲ فعالیت می‌باشد. مسائل بزرگ نیز هر کدام شامل ۲۰ پروژه و هر پروژه شامل ۵۲ فعالیت می‌باشد. به دلیل اینکه حل مسائل متوسط و بزرگ در نرم‌افزار لینگو در زمان معمول و با پردازشگرهای فعلی امکان‌پذیر نبود، لذا برای اینکه مشخص شود الگوریتم ژنتیک در مورد این مسائل چقدر کارا می‌باشد، بهترین جواب تصادفی موجود در جمعیت اولیه ایجاد شده در هر مساله، استخراج گردیده تا بتوان مقدار بهبود را در مساله مورد نظر نسبت به آن مقایسه کرد. در نهایت پس از حل مسائل مذکور و مقایسه جواب‌های اولیه و نهایی، به طور متوسط مقدار  $\frac{27}{3}$ ٪ بهبود را در جواب‌های مسائل متوسط و  $\frac{33}{15}$ ٪ بهبود را در جواب‌های مسائل بزرگ نشان می‌دهد.

جدول سه، میانگین جواب‌های مربوط به حل ۳۰ مساله کوچک، ۳۰ مساله متوسط و ۳۰ مساله بزرگ را نشان می‌دهد.

جدول ۷: نتایج حاصل از حل مسائل طرح شده توسط الگوریتم ژنتیک

اندازه مساله	میانگین جواب نهایی حاصل از حل ۳۰ مساله طرح شده توسط الگوریتم ژنتیک	میانگین بهترین جواب تصادفی بدست آمده در جمعیت اولیه توسط الگوریتم ژنتیک	میانگین جواب نهایی حاصل از حل ۳۰ مساله طرح شده توسط مدل ریاضی در نرم افزار لینگو
کوچک	۳۳۲/۳۹	---	۲۵۹/۵۴
متوسط	۹۸۶۵/۷۲	۷۸۹۲/۰۵	---
بزرگ	۲۳۷۹۸/۶۴	۱۸۰۶۹/۱۹	---

به عنوان نمونه نمودار نشان داده شده در شکل ۱۷، بهبود جواب در مساله شماره هفت از دسته مسائل متوسط را در هر نسل نشان می دهد. در این مساله بهترین جواب تصادفی در جمعیت اولیه مقدار ۷۶۶۶/۱۶ و بهترین جواب در نسل ۶۰ ام مقدار ۱۰۲۴۰/۳۲ بدست آمده است که گویای رشد ۳۳/۵۸٪ می باشد.



شکل ۱۷: نمودار بهبود جواب توسط الگوریتم ژنتیک در ۶۰ نسل ایجاد شده

نکته دیگری نیز که از نمودار شکل ۱۷ استنباط می‌شود این است که در الگوریتم ژنتیک مقدار بهبود جواب در نسل‌های نخست با سرعت زیادی انجام شده و رفته رفته از این روند کاسته می‌شود و در نهایت ممکن است در چندین نسل پیاپی دیگر بهبودی مشاهده نشود.

### نتیجه‌گیری

مساله انتخاب سبد پروژه و مساله سرمایه‌گذاری در منابع پروژه بطور مجزا در ادبیات موجود زمان بندی پروژه یافت می‌شود. در این مقاله، این دو مساله چالشی بطور یک مساله واحد مدل‌سازی شده است. هدف مساله یکپارچه انتخاب سبد پروژه‌ها و سرمایه‌گذاری در منابع بیشینه کردن ارزش خالص فعلی کسب شده از اجرای پروژه‌های منتخب در نظر گرفته شده است. انتخاب سبد بهینه، تعیین سطوح بهینه منابع و زمان بندی پروژه‌های انتخاب شده متغیرهای تصمیم در مدلی ریاضی غیرخطی مساله مورد بررسی هستند. به دلیل اینکه مساله مورد نظر از نوع سخت محاسباتی می‌باشد، بنابراین در ادامه یک روش فراابتکاری بر پایه الگوریتم ژنتیک برای حل آن توسعه داده شد. کروموزوم بکار رفته از نوع رشته‌ای بوده و سه بخش برای کد کردن پروژه‌های فعال و غیرفعال، توالی فعالیت‌ها و سطح منابع مورد نیاز در آن استفاده شده است. برای رمزگشایی کروموزومها نیز از روش تولید برنامه‌های موازی استفاده شده است. عملگر جهش بر روی رشته اول کروموزوم بکار گرفته می‌شود و در آن برخی پروژه‌های فعال یا غیرفعال می‌شوند. عملگر تقاطع بر روی رشته دوم کروموزوم یعنی توالی فعالیت‌ها استفاده می‌شود. نوع روش تقاطع بر اساس آزمایشهای آماری انجام انتخاب شده است. در الگوریتم توسعه یافته در این مقاله، علاوه بر عملگرهای تقاطع و جهش از عملگرهای هوشمند دیگری برای جستجوی محلی در حوزه منابع و جابجایی فعالیت‌های با جریان مالی منفی نیز استفاده شده است. در نهایت برای ایجاد تعادل بین نخبه‌گرایی و تنوع‌گرایی در ترکیب جمعیت ایجاد شده در هر نسل، علاوه بر جمعیت ایجاد شده توسط عملگرهای مذکور، تعدادی از افراد جمعیت نیز همانند جمعیت اولیه به صورت تصادفی ایجاد شده‌اند. جهت ارزیابی الگوریتم ژنتیک توسعه داده شده ابتدا پارامترهای الگوریتم توسط روش تاگوشی و با استفاده از تعدادی آزمایش تنظیم شد. سپس تعداد ۹۰ مساله در ابعاد کوچک، متوسط و بزرگ جهت حل با الگوریتم ژنتیک بطور تصادفی تولید شدند. مسائل ساینز کوچک با استفاده از دو روش الگوریتم ژنتیک و مدل ریاضی کد شده

در نرم افزار لینگو حل شدند که در اکثر موارد جواب های حاصل از الگوریتم ژنتیک بهتر از جواب بهینه موضعی حاصل از لینگو بود. در مسائل متوسط و بزرگ نیز جواب های حاصل از الگوریتم ژنتیک با بهترین جواب تصادفی ایجاد شده در جمعیت اولیه مقایسه شدند که در مسائل متوسط شاهد ۲۷٪ بهبود و در مسائل بزرگ نیز شاهد ۳۳٪ بهبود بصورت متوسط نسبت به جواب های اولیه می باشیم. همچنین اثبات شد که ضمن کارایی و پایداری مناسب الگوریتم ژنتیک توسعه داده شده، این الگوریتم در زمان هایی کوتاه، قادر به حل مسائلی با ابعاد بزرگ است و می تواند جواب هایی قابل اتکا ایجاد کن، بویژه زمانی که توسط روش های بهینه یابی قادر به یافتن حتی یک جواب موجه نبودیم. برای ادامه تحقیقات آتی در ارتباط با موضوع این مقاله می توان محدودیت هایی مانند سقف میزان سرمایه گذاری برای استخدام منابع یا استفاده از منابع محدود چندگانه را مد نظر قرار داد. همچنین می توان مدل توسعه یافته در این تحقیق را در فضای چند هدفه تعمیم داد.



## منابع

کلامی هریس، سیدمصطفی. (۱۳۹۰). فیلم آموزشی الگوریتم ژنتیک، متلب سایت (مرجع آموزش هوش مصنوعی).

شادرخ، شهرام. (۱۳۷۸). مسئله سرمایه‌گذاری در کنترل پروژه. دومین کنفرانس ملی مهندسی صنایع، دانشگاه یزد.

Afshar-nadjafi, Behrouz. (2014). *Multi-mode resource availability cost problem with recruitment and release dates for resources. Applied Mathematical Modelling* 38, Pages 5347–5355.

Mohring, R.H. (1984). *Minimizing Costs of Resource Requirments in Project Networks Subject to A Fix Completion Time. Operational Research*. 32, 89-120.

Najafi, Amir Abbas. AkhavanNiaki, SeyedTaghi. Shahsavar, Moslem. (2009). *A parameter-tuned genetic algorithm for the resource investment problem with discounted cash flows and generalized precedence relations. Computers & Operations Research* 36. 2009. 2994 – 3001.

Payne, J. H. (1995). Management of multiple simultaneous projects: a state-of-the-art review. *International Journal of Project Management* 13. 163-168.

Pritsker, L. J., Walters, A. A. B., and Wolfe, P. M.. (1969). *Multi-project scheduling with limited resources: a zero one programming approach. Management Science* 16. 93-108.

Shadrokh, Shahram. Kianfar, Fereydoon. (2007). A genetic algorithm for resource investment project scheduling problem, tardiness permitted with penalty. *European Journal of Operational Research* 181. 2007. 86–101.

Shahsavar, Moslem. Akhavan Niaki, Seyed Taghi. Najafi, Amir Abbas. (2010). *An efficient genetic algorithm to maximize net present value of project payments under inflation and bonus–penalty policy in resource investment problem. Advances in Engineering Software* 41. 2010. 1023–1030.