

# ارائه مدل ریاضی چندهدفه برای مسأله تبادل هزینه-زمان و در نظر گرفتن ارزش زمانی پول با استفاده از الگوریتم MOPSO

محمدعلی شفیعی<sup>۱</sup>، محمدرضا شهیریاری<sup>۲\*</sup>، فرهاد حسین زاده لطفی<sup>۳</sup>، رضا رادفر<sup>۴</sup>

<sup>(۱)</sup> دانشجوی دکتری مدیریت تکنولوژی، واحد امارات متحده عربی، دانشگاه آزاد اسلامی، دبی، امارات.

<sup>(۲)</sup> دانشیار، دانشکده مدیریت و حسابداری، واحد تهران جنوب، دانشگاه آزاد اسلامی، تهران، ایران. (نویسنده مسئول)

<sup>(۳)</sup> استاد، گروه ریاضی کاربردی، دانشکده علوم پایه، واحد علوم و تحقیقات، دانشگاه آزاد اسلامی، تهران، ایران.

<sup>(۴)</sup> استاد، گروه مدیریت صنعتی، دانشکده مدیریت و اقتصاد، واحد علوم و تحقیقات، دانشگاه آزاد اسلامی، تهران، ایران.

تاریخ دریافت مقاله: ۹۵/۰۸/۱۲ تاریخ پذیرش مقاله: ۹۵/۱۰/۱۴

## چکیده:

مسأله تبادل هزینه و زمان یکی از مهم‌ترین مسائل موجود در زمینه زمان‌بندی پروژه می‌باشد و تاکنون تحقیقات بسیار زیادی با رویکردهای گوناگون کمی و کیفی بر روی این موضوع انجام شده است. در این تحقیق قصد داریم تا با ارائه یک مدل دو هدفه ریاضی، شرایط را برای ایجاد توازن میان فشردگی‌سازی، صرفه‌جویی در هزینه و به تأخیر انداختن زمان اجرای فعالیت‌ها مهیا کنیم به طوری که ابزار مناسبی در اختیار تصمیم‌گیرندگان برای تصمیم‌گیری در رابطه با زمان اجرای هر فعالیت با توجه به امکانات در دسترس و نیز زمان در اختیار، برای اتمام پروژه فراهم آید. در مدل ریاضی پیشنهادی تلاش شده است تا با به کارگیری فرضیاتی نظیر تابع هزینه غیرخطی و همچنین در نظر گرفتن ارزش زمانی پول، شرایط مسئله تا حد امکان به محیط واقعی نزدیک‌تر گردد. در پایان مدل ریاضی ارائه شده در این مقاله را با استفاده از الگوریتم Objective Particle Swarm Optimization (MOPSO) بررسی نموده و تاثیر فشردگی‌سازی و به تأخیر انداختن فعالیت‌ها را بر روی مجموعه نامغلوب نهایی ارائه خواهیم داد.

**واژه‌های کلیدی:** تبادل هزینه و زمان، زوش بهینه‌سازی ازدحام ذرات، ارزش زمانی پول، زمان‌بندی پروژه.

## ۱- مقدمه

مسئله تبادل هزینه و زمان

The discrete time-cost tradeoff problem (DTCTP) تاکنون کاربردهای عملی فراوانی را در حوزه اجرای پروژه‌ها یا طراحی نرم‌افزارها داشته و تحقیقات متنوعی بر روی آن انجام گرفته است، به گونه‌ای که نیاز مشتریان برای دریافت خدمات در بازه زمانی کوتاه‌تر از یک سو و الزام برای کاهش هزینه‌های انجام پروژه از سوی دیگر، اهمیت این نوع مسائل را در بین صاحبان کسب و کار و همچنین محققین تا اندازه‌ی چشمگیری بالا برده است.

DTCTP برای اولین بار در سال ۱۹۷۹ توسط Hindelang و Muth مطرح شد [1] و مورد توجه بسیاری از محققین در حوزه مدیریت و برنامه‌ریزی پروژه قرار گرفت. Prubuddha DE و همکارانش [2] و همچنین Vladimir DE و همکارانش [3] نشان دادند که این مسئله از نوع NP-hard می‌باشد و حل آن دشوار است. یک فرض اساسی در این نوع مسائل که در بسیاری از مطالعات در نظر گرفته شده است، حاکی از این است که هزینه فعالیت‌ها تابعی از طول زمان اجرای آنها می‌باشد و مدت زمان اجرای هر فعالیت در این مسائل به عنوان یک متغیر تصمیم لحاظ شده که دارای حد پایین و حد بالا بوده و مدل ریاضی مربوطه مقدار بهینه آن را مشخص می‌کند.

از مدل‌های ریاضی خطی که در این زمینه معرفی شده‌اند می‌توان به مطالعات Kelley و Walker در برنامه‌ریزی و زمان‌بندی مسیر بحرانی [4]، Fulkerson در محاسبات جریان شبکه برای منحنی‌های هزینه پروژه [5]، Kelley در برنامه‌ریزی و زمان‌بندی مسیر بحرانی [6]، Ford و Fulkerson در جریان‌های شبکه [7]، Siemens و همین‌طور Goyal در الگوریتم تبادل هزینه و زمان در CPM ساده [8]، [9] و Elmaghraby و Salem در تقریب خطی بهینه در فشرده‌سازی پروژه [10]، اشاره نمود. همچنین Moder و همکاران در مدیریت پروژه در شبکه‌های CPM و PERT [11]، تابع هزینه فعالیت‌ها را به صورت پیوسته در نظر گرفتند.

DTCTP تاکنون با روش‌های مختلف حل مورد بررسی قرار گرفته است که از آن جمله می‌توان به روش‌های دقیق اشاره نمود که در مطالعات برنامه‌ریزی پویا توسط (enumeration، Muth و Hindelang، قائده‌ی شمارش algorithm) توسط Harvey و Patterson [12] و روش شاخه و حد (branch and bound) توسط Demeulemeester و همکاران [13]، [14] و همچنین Erenguc و همکاران [15] مورد استفاده قرار گرفته است. اما هیچ یک از روش‌های دقیق قادر به حل مسئله DTCTP در ابعاد بزرگ نمی‌باشند. از همین رو گرایش به سمت استفاده از دیگر روش‌ها در بین پژوهشگران قوت گرفت. تحقیق Akkan نمونه‌ای از یک روش ابتکاری براساس آزادسازی لاگرانژ در حل این نوع مسائل درون شبکه AOA می‌باشد [16]. Liu و همکاران از الگوریتم ژنتیک برای حل DTCTP بهره بردند [17]. همچنین Peng و همکاران با استفاده از الگوریتم ژنتیک به حل DTCTP پرداختند [18]. Elmaghraby و Kamburowski بر روی تابع هدف DTCTP جریمه و پاداش در نظر گرفتند [19].

فشرده‌سازی فعالیت‌ها توسط Ann و همکاران معرفی شد [20]. Van Slyke در سال ۱۹۶۳ نخستین کسی بود که از شبیه‌سازی مونت کارلو در زمینه فشرده‌سازی استفاده نمود، او توانست دستاوردهای زیادی بدست آورد که از آن جمله می‌توان به محاسبه دقیق‌تر طول زمان پروژه‌ها، انعطاف‌پذیری در انتخاب تابع توزیع زمان فعالیت‌ها و توانایی محاسبه میزان بحرانی بودن مسیر اشاره نمود. Vanhoucke [21] و همکاران DTCTP را با در نظر گرفتن محدودیت time-switch مورد مطالعه قرار دادند [22].

هر چند این تکنیک‌ها مزایای خود را دارند، تمام آنها با DTCTP به صورت مسئله‌ای تک‌هدفه برخورد می‌کنند که هدف آنها متمرکز بر کاهش هزینه‌های پروژه یا کاستن زمان اتمام پروژه می‌باشد. در این میان برخی محققان متوجه شده‌اند که برای تعیین زمان اجرای فعالیت‌های پروژه باید ملاحظات دیگری در نظر گرفته شوند، مثلاً ممکن است مدیریت پروژه تمایل به دستیابی به سطح بالاتری از سرعت انجام پروژه با هزینه پایین را داشته باشد.

در این صورت یک راه حل قدیمی می تواند این باشد که زمان اتمام پروژه و هزینه را با دو هدف برای یک تابع هدف اسکالر منحصر به فرد در نظر گرفت Sasaki و [23],[24]. Pasandideh Gen و همکاران [25] این تابع جدید را با استفاده از تکنیک های تک هدفه نظیر روش IP-metric بهینه نمودند. تحقیقات صورت گرفته قدم های مهمی را بسوی یافتن رویکردهای مؤثرتر و کارآمدتر برای مدل های ریاضی تک هدفه برداشته اند، اما به هر حال برای رویکردهای تک هدفه، باید جهت دستیابی به عملکرد بهتر، مکانیزم های پیچیده ترکیبی چند هدفه را طراحی کنیم. از سوی دیگر جمع دو هدف ممکن است احتمال یافتن چند راه حل را از بین ببرد که در عمل انتخاب های کمتری را برای تصمیم گیرنده باقی می گذارد. ما نیز در این تحقیق قصد داریم تا با ارائه ی یک مدل ریاضی دوهدفه، بهینه سازی هزینه و زمان را به صورت توأم و با استفاده از یک الگوریتم MOPSO پیشنهادی انجام دهیم.

فرض فشرده سازی فعالیت ها در مسئله ی تبادله هزینه و زمان به دلیل ارائه ی راه حل هایی مطرح شد که در آنها بتوان با صرف هزینه های مشخص، سرعت انجام پروژه را تا حد امکان بالا برد. نکته ی قابل توجه در این است که ارائه ی راه حل هایی مبتنی بر سرعت پایین انجام پروژه ولی با صرفه جویی بیشتر هزینه ها می تواند صاحب ارزش باشد. در یک نگرش کلی به تاخیر افتادن پروژه ها واقعیتی انکارناپذیر در انجام پروژه های بزرگ می باشد. چنان که سازمان جهانی سدها (WCD) طی تحقیقی که به صورت بسیار کلی روی ۹۹ پروژه انجام داده است مطرح می نماید که تنها نیمی از آنها در زمان مقرر به پایان رسیده اند و حدود ۳۰٪ آنها از یک تا ۲ سال و ۴ پروژه هم بیش از ۱۰ سال تاخیر داشته است. دلایل اصلی تاخیر در این پروژه ها عبارتند از مشکلات مالی، عدم کارایی پیمانکار و مدیریت ساخت، زمان بندی غیرواقعی، نارضایتی نیروی انسانی و اعتراض و چالش های قانونی و حقوقی [26].

لذا بر خلاف تحقیقات صورت گرفته پیشین در این زمینه که حد بالای زمان اجرای فعالیت ها را برابر با

زمان نرمال آنها در نظر می گرفتند، ما در مدل ریاضی ارائه شده در این مقاله بنا داریم تا این حد را تا مقداری برابر حداکثر زمان ممکن ارتقاء دهیم، به نحوی که هر چه از زمان نرمال انجام فعالیت به سمت زمان فشرده آن حرکت می نماییم، می بایست هزینه های فشرده سازی بیشتری را متحمل شویم و هر چه از زمان نرمال انجام فعالیت به سمت حد بالای در نظر گرفته شده برای آن میرویم با صرفه جویی بیشتر هزینه ها مواجه هستیم. در این صورت مرز نهایی پاسخ های بدست آمده از الگوریتم MOPSO طیفی از پاسخ های متنوع را در برابر تصمیم گیرنده قرار می دهد که وی با توجه به شرایط بودجه و زمان مورد نیاز خود جهت اتمام پروژه یکی از این پاسخ ها را برمیگزیند. مسئله مهم دیگر در نظر گرفتن ارزش زمانی پول است چرا که در پروژه های بزرگ مهم است که بدانیم هزینه بیشتر یا صرفه جویی آن در چه مقطعی از زمان صورت می پذیرد و تاثیر نرخ بهره ی پول بر آن چقدر می باشد. در تمام تحقیقات انجام شده که به آنها اشاره شد ارزش زمانی پول در نظر گرفته نشده است در حالی که این عامل فاکتور مهمی برای تصمیم گیری در انجام پروژه های بزرگ و سرمایه بر است. در این مقاله قصد داریم تا ضمن در نظر گرفتن این عامل به بهینه سازی هزینه و زمان بپردازیم.

## ۲- متغیرها و پارامترها:

$t_i$ :	زمان رخداد رویداد $i$
$D_{f(ij)}$ :	حداقل زمان مجاز برای فعالیت $i-j$
$D_{n(ij)}$ :	زمان نرمال فعالیت $i-j$
$D_{m(ij)}$ :	حد اکثر زمان مجاز برای فعالیت $i-j$
$d_{ij}$ :	زمان برنامه‌ریزی شده (عملی) برای اجرای فعالیت $i-j$
$H$ :	هزینه غیرمستقیم (بالاسری) پروژه
$K_n$ :	هزینه مستقیم پروژه
$C_{f(ij)}$ :	هزینه‌ی فشرده‌سازی فعالیت $i-j$
$C_{n(ij)}$ :	هزینه‌ی نرمال فعالیت $i-j$
$C_{m(ij)}$ :	هزینه‌ی انجام با تاخیر فعالیت $i-j$
$C_{ij}$ :	ضریب هزینه فشرده‌سازی فعالیت $i-j$
$C'_{ij}$ :	ضریب صرفه‌جویی به تاخیر انداختن فعالیت $i-j$
$t_{Max}$ :	حد اکثر زمان مورد نظر برای اتمام پروژه
$C_{Max}$ :	حد اکثر بودجه در اختیار
$I_0$ :	نرخ بهره
$y_{ij}$ :	اگر فعالیت مربوطه فشرده شود ۱ در غیر این صورت ۰ است
$y'_{ij}$ :	اگر فعالیت مربوطه با تاخیر انجام شود ۱ در غیر این صورت ۰ است
$y''_{ij}$ :	اگر فعالیت مربوطه در زمان نرمال انجام شود ۱ در غیر این صورت ۰ است

## ۳- بیان مسئله:

زمانی که فعالیت‌های آغازین یک پروژه فشرده می‌شوند سرمایه مورد نیاز برای فشرده‌سازی تا روز پایان پروژه درگیر خواهد بود و چنانچه فعالیت‌های پایانی پروژه فشرده شوند این سرمایه زمان کمتری را در پروژه درگیر خواهد بود. همین اثر در رابطه با صرفه‌جویی حاصل از به تاخیر انداختن انجام فعالیت‌ها نسبت به زمان نرمال در طول زمان انجام پروژه مشهود است. لذا ارزش زمانی پول به عنوان یک عامل تأثیرگذار در این زمینه مطرح خواهد بود. لازم به توضیح است که در این مقاله به دنبال ارائه مدلی که فاکتور ارزش زمانی پول را در مسائل تبادل هزینه و زمان لحاظ نماید، خواهیم بود.

شهریاری و همکاران در سال ۲۰۰۹ مسئله تبادل هزینه و زمان را با در نظر گرفتن ارزش زمانی پول مورد بررسی قرار دادند. دستاورد آنها ارائه یک مدل ریاضی

در زمان‌بندی شبکه‌های پرت بود. در این تحقیق آنها نائل به توسعه بزرگی بر مقوله فشرده‌سازی زمان در شبکه‌های پرت و سی‌پی‌ام گردیدند که ارزش زمانی پول به عنوان یک متغیر تأثیرگذار بر تابع هزینه کل به صورت خطی، غیرخطی و چندهدفه مورد مطالعه قرار گرفت [27]. البته شهریاری در مقاله‌ای دیگر نیز با استفاده از NSGAII و مرتب‌سازی در داده‌های غیرمسلط، راه‌حلی برای DTCTP ارائه نمود که در نهایت منجر به ارائه تابعی برای در نظر گرفتن مطلوبیت تصمیم‌گیرنده در مسئله فشرده‌سازی با در نظر گرفتن معیار ارزش زمانی پول و تبادل آن با میزان فشرده‌سازی گردید [28].

از این‌رو هزینه‌ی فشرده‌سازی به تابع هزینه کل اضافه و صرفه‌جویی حاصل از به تاخیر انداختن فعالیت‌ها از آن کم می‌شود تا تبادل زمان و هزینه در شبکه‌های پرت

در رابطه‌ی بالا  $I_0$  نرخ بهره است و  $C_{ij}$  و  $C'_{ij}$  تحت عنوان ضریب هزینه و صرفه‌جویی عمل می‌کنند. در گذشته تحقیقاتی بر روی چگونگی تاثیرگذاری هزینه‌های پروژه و نیز فشردگی‌سازی هزینه‌ها انجام شده است. از آن جمله می‌توان به مطالعه انجام شده توسط Ameen اشاره نمود که با تعریف شاخص شیب هزینه و ارائه تکنیکی تحت عنوان CAPERTSIM به تصمیم‌گیری تحت شرایط عدم قطعیت برای تبادل هزینه و زمان در فشردگی‌سازی و رابطه بین آنها پرداخت. در این تحقیق شاخص شیب هزینه، نسبت هزینه اعمال شده به میزان فشردگی‌سازی تعریف گردید و با در نظر گرفتن طول زمان‌های هر فعالیت به صورت احتمالی مدلی نسبتاً ابتدایی بر مبنای یک شبیه‌سازی ساده برای فشردگی‌سازی ارائه گردید [29]. در این تحقیق بنا داریم رابطه‌ای غیرخطی برای  $C_{ij}$  و  $C'_{ij}$  نسبت به طول زمان اجرا فعالیت‌ها ارائه دهیم که برای این منظور فرض شده رابطه هزینه فشردگی‌سازی فعالیت‌ها با طول زمان اجرا آنها به شکل ۱ باشد:

به صورت واقعی‌تر و با در نظر گرفتن ارزش زمانی پول به انجام تبادل هزینه و زمان بپردازد.

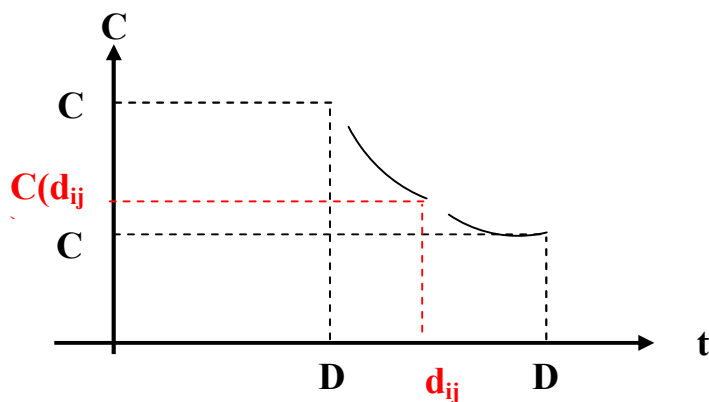
بنابراین تابع هزینه این مدل متشکل از هزینه‌های مستقیم، غیرمستقیم و هزینه فشردگی‌سازی و صرفه‌جویی حاصل از به تاخیر افتادن انجام فعالیت‌ها می‌باشد.

$$z_1 = H(t_n - t_1) + K_n + \sum_i \sum_j C_{ij} \cdot y_{ij} (D_{n(ij)} - dij - ij \cdot C'_{ij} \cdot y_{ij} D_{n(ij)} - dij)$$

برای درج روابط که قادر به لحاظ نمودن فاکتور ارزش زمانی پول برای فشردگی‌سازی باشد، به ترتیب زیر عمل می‌کنیم.

اگر فرض کنیم تعدادی از فعالیت‌ها فشردگی شده است و برای هر واحد زمانی فشردگی‌سازی فعالیت  $i-j$  به میزان  $C_{ij}$  واحد پول هزینه صرف شده باشد، واضح است که  $\sum_i \sum_j C_{ij} (t_n - t_i)$  بیان‌گر مقدار پولی است که از روز  $t_1$  تا روز  $t_n$  (روز اتمام پروژه) در پروژه درگیر بوده است. به همین ترتیب میزان صرفه‌جویی هزینه‌ها نیز می‌بایست متأثر از ارزش زمانی پول باشد بنابراین داریم:

$$z_1 = H(t_n - t_1) +$$



شکل ۱- نمودار رابطه هزینه و زمان

$$(7) \quad \sum_i \sum_j y_{ij} \cdot \alpha_{ij} \cdot e^{-\beta_{ij}(d_{ij})} (1 + (t_n + t_i)) I_0 - \sum_i \sum_j y_{ij} \cdot \alpha_{ij} \cdot e^{-\beta_{ij}(d_{ij})} (1 + (t_n + t_i)) I_0 + H(t_n - t_1) \leq c_{max}$$

$$(8) \quad t_i \geq 0 \quad y_{ij} \in \{0,1\}$$

### ۱-۴- تابع هدف اول:

در تابع هدف اول میزان کل هزینه‌های مستقیم و غیرمستقیم پروژه، هزینه‌های حاصل از فشردگی فعالیت‌ها و صرفه‌جویی ناشی از به تاخیر انداختن انجام فعالیت‌ها قرار دارد. نکته‌ی قابل توجه در تابع هدف اول در نظر گرفتن ارزش زمانی پول در محاسبه هزینه‌های فشردگی و صرفه‌جویی در تاخیر فعالیت‌ها است. به طور کلی می‌توانیم این تابع را به صورت زیر تفکیک نماییم.

- هزینه‌های حاصل از فشردگی فعالیت‌ها:

$$\sum_i \sum_j y_{ij} \cdot \alpha_{ij} \cdot e^{-\beta_{ij}(d_{ij})} (1 + (t_n + t_i)) I_0$$

- صرفه‌جویی ناشی از به تاخیر انداختن فعالیت‌ها:

$$\sum_i \sum_j y'_{ij} \cdot \alpha'_{ij} \cdot e^{-\beta'_{ij}(d_{ij})} (1 + (t_n + t_i)) I_0$$

- هزینه‌های مستقیم و غیرمستقیم پروژه:

$$H(t_n - t_1) + k_n$$

### ۲-۴- تابع هدف دوم:

کاملاً مشخص است که در تابع هدف دوم زمان اتمام پروژه را  $\min$  می‌کنیم. بهینه‌سازی این هدف به همراه هدف اول، یک ساختار همگون از پاسخ‌ها را ایجاد می‌نماید. به طوری که بدون دخالت دادن هدف دوم در روند بهینه‌سازی مسئله، جواب‌ها به سمت به تاخیر انداختن پروژه تشویق می‌شوند، چرا که برای بهینه نمودن هدف اول تمایل به صرفه‌جویی بیشتر هزینه‌ها دارند. اما واضح است که صرفه‌جویی بیشتر هزینه‌ها زمان طولانی‌تر اجرای پروژه را به دنبال دارد و بدین ترتیب تابع هدف اول غیربهینه می‌شود. از طرفی اجرای پروژه در بازه‌ی زمانی کوتاه‌تر با فشردگی بیشتر فعالیت‌ها و پرداخت هزینه‌های بیشتر روبرو است.

با توجه به شکل مذکور چنانچه رابطه میزان فشردگی زمان با هزینه فشردگی از توزیع نمایی منفی با ضابطه‌ای به شکل زیر پیروی کند، آنگاه داریم:

$$C(d_{ij}) = a e^{-\beta(d_{ij})}$$

با مشخص بودن مختصات نقاط به طول  $D_f$  و  $D_n$  مقادیر  $\alpha$  و  $\beta$  قابل محاسبه است:

$$a = e^{(L_n(C_n))}$$

$$\beta = \frac{L_n \left( \frac{C_n}{C_f} \right)}{D_f - D_n}$$

به همین ترتیب برای ضریب صرفه‌جویی هزینه‌ها داریم:

$$C(d_{ij}) = \alpha e^{-\beta(d_{ij})}$$

$$\alpha = e^{(L_n(C_m) + \beta \cdot D_m)}$$

$$\beta = \frac{L_n \left( \frac{C_m}{C_n} \right)}{D_n - D_{nm}}$$

از سوی دیگر هدف دوم پروژه را برابر زمان اتمام پروژه قرار می‌دهیم:  $Z_2 = t_n$

### ۴-۵- مدل ریاضی:

با توجه به توضیحات بالا مدل ریاضی ارائه شده به صورت زیر می‌باشد:

$$\begin{aligned} \min Z_1 &= \sum_i \sum_j y_{ij} \cdot \alpha_{ij} \cdot e^{-\beta_{ij}(d_{ij})} (1 + (t_n + t_i)) I_0 \\ &- \sum_i \sum_j y'_{ij} \cdot \alpha'_{ij} \cdot e^{-\beta'_{ij}(d_{ij})} (1 + (t_n + t_i)) I_0 \\ &+ H(t_n - t_1) + k_n \end{aligned}$$

$$\min Z_2 = t_n$$

s. t:

- (1)  $t_j - t_i \geq d_{ij}$
- (2)  $y_{ij} \cdot Df_{ij} \leq y_{ij} \cdot d_{ij} < y_{ij} \cdot Dn_{ij} \quad \forall i, j$
- (3)  $y_{ij} \cdot Dn_{ij} < y_{ij} \cdot d_{ij} \leq y_{ij} \cdot Dm_{ij} \quad \forall i, j$
- (4)  $y''_{ij} \cdot d_{ij} = Dn_{ij} \cdot y''_{ij} \quad \forall i, j$
- (5)  $y_{ij} + y_{ij} + y''_{ij} = 1 \quad \forall i, j$
- (6)  $t_n \leq t_{max}$

**محدودیت ششم:** محدودیت زمان اجرای پروژه حداکثر تا زمان  $t_{max}$  است.

**محدودیت هفتم:** محدودیت بودجه در اختیار حداکثر تا مقدار  $C_{max}$  است.

### ۵- روش حل:

تاکنون تحقیقات زیادی در رابطه با استفاده از الگوریتم GA در حل مدل‌های ریاضی تک‌هدفه و چندهدفه انجام شده است. برای مثال Sou-Sen Leu, با ارائه مقاله‌ای در مجله بین‌المللی کنترل پروژه و با استفاده از الگوریتم ژنتیک بر مبنای نظریه فازی، اثرات عدم قطعیت و داده‌های نادقیق در زمینه تبادل هزینه و زمان را بررسی نموده‌اند [30]. در تحقیقی دیگر Heng.Li و J.N.Cao از دانشگاه پلی‌تکنیک هنگ‌کنگ با ترکیبی از الگوریتم ژنتیک و متدهای learning machine دست به ایجاد تکنیکی به نام MLGAS زده‌اند و ادعا کرده‌اند در حالتی که رابطه بین هزینه‌ها و فعالیت‌ها غیرخطی باشد، این تکنیک جواب‌های بهتری بدست خواهد داد [31]. در میان پژوهش‌های انجام شده، مقاله Chung – Wei Feng و Scott. A.Burns از دانشگاه ایلی‌نیوز، ضمن ارائه الگوریتمی بر پایه الگوریتم ژنتیک، یک برنامه محاسباتی نیز در زمینه محاسبه کارایی الگوریتم تهیه کرده است. این پژوهش کامل‌ترین پژوهشی است که با استفاده از الگوریتم ژنتیک، در زمینه تبادل هزینه و زمان انجام شده است [32]. Chua و همکاران در سال ۱۹۹۷ با استناد به آن که برای حل مسایل فشرده‌سازی زمان، مدل‌های ریاضی مانند برنامه‌ریزی خطی و برنامه‌ریزی عدد صحیح از لحاظ زمان محاسبات به صرفه نمی‌باشند و برای مسائل با حجم بزرگ کارایی چندانی ندارند، اقدام به توسعه الگوریتم ژنتیک برای آن نمود. در این مدل محدودیت منابع برای هر یک از فعالیت‌ها نیز مدنظر قرار گرفت [33].

به طور کلی الگوریتم‌های بهینه‌سازی تک‌هدفه، حل بهینه را با توجه به یک هدف می‌یابند و این در حالی است که در مسائل چندهدفه یک حل بهینه مجزا را نمی‌توان یافت. پس طبیعی است که با یک مجموعه‌ای

**محدودیت اول:** این محدودیت برای رعایت پیش‌نیازی فعالیت‌ها قرار داده شده است به طوری که اگر فعالیت B پیش‌نیاز فعالیت A باشد برای زمان اتمام فعالیت A داریم:

$$t_A = d_{BA} + t_B$$

اما علامت بزرگتر مساوی به این علت استفاده شده است که به عنوان مثال فرض کنید فعالیت‌های B و C پیش‌نیاز فعالیت A باشد آنگاه برای زمان اتمام فعالیت A داریم:

$$t_A = \max(d_{BA} + t_B, d_{CA} + t_C)$$

**محدودیت دوم:** محدودیت کنترلی برای فشرده‌سازی فعالیت‌ها است به طوری که اگر فعالیتی فشرده شود ( $y_{ij} = 1$ ) حد پایین این فشرده‌سازی برابر  $Df_{ij}$  می‌باشد از طرفی اگر فعالیتی فشرده شود با توجه به محدودیت ۲ و ۵، حتما ( $y_{ij} = 1$ ) خواهد شد.

**محدودیت سوم:** محدودیت کنترلی برای به تاخیر انداختن فعالیت‌ها است. به طوری که اگر فعالیتی با تاخیر مواجه شود ( $y''_{ij} = 1$ ) حد بالای این تاخیر برابر  $Dm_{ij}$  می‌باشد، از طرفی اگر فعالیتی با تاخیر مواجه شود با توجه به محدودیت ۳ و ۵ حتما ( $y'_{ij} = 1$ ) خواهد شد.

**محدودیت چهارم:** محدودیت کنترلی برای انجام فعالیت در زمان نرمال است به طوری که اگر فعالیتی در زمان نرمال انجام شود ( $y''_{ij} = 1$ )، این زمان برابر  $Dn_{ij}$  می‌باشد. از طرفی اجرای فعالیت در زمان  $Dn_{ij}$  با توجه به محدودیت ۳ و ۵ حتما ( $y''_{ij} = 1$ ) خواهد شد.

**محدودیت پنجم:** به این معنی است که برای طول زمان اجرای هر فعالیت به طور کلی سه حالت وجود دارد و برای هر فعالیت می‌بایست یکی از این سه حالت را برگزینیم.

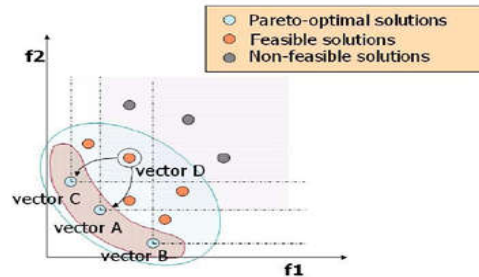
- فعالیت فشرده می‌شود.

- فعالیت با تاخیر مواجه می‌شود.

- فعالیت در زمان نرمال انجام می‌شود.

مناسب جواب‌هایی خواهد بود که عملکرد قابل قبولی را نسبت  
 علت این است که معمولاً جواب بهینه خاصی برای این  
 روش‌ها بدست نمی‌آید [36]. شکل ۲ مجموعه  
 جواب‌های پارتو را در فضای حل مسائل چندهدفه نشان  
 می‌دهد.

از حل‌ها به نام حل‌های مغلوب نشده موثر سر و کار  
 داشته باشیم. از بین این مجموعه حل‌های متناهی، حل  
 به همه اهداف داشته باشد. حل مسائل چندهدفه با  
 رویکرد پارتو از دسته مسائل پیچیده‌تر در حل مسائل  
 چندهدفه می‌باشند [34].



شکل ۲- مجموعه جواب‌های پارتو در حل مسائل چندهدفه (۳۴)

## ۱- الگوریتم بهینه‌سازی اجتماع ذرات چندهدفه<sup>۱</sup>

(MOPSO):

چندین رویکرد مختلف از الگوریتم PSO برای مسائل  
 چندهدفه توسعه داده شده است. الگوریتم Moore و  
 Chapman در سال ۱۹۹۹ از اولین الگوریتم‌ها در این  
 زمینه است [37]. این الگوریتم در یک سند غیر چاپی  
 ارایه شد و مبتنی بر پارتو است. همان‌طور که می‌دانیم  
 الگوریتم‌های مبتنی بر پارتو از مفهوم غلبه جهت  
 رتبه‌بندی اعضای جمعیت استفاده می‌کنند. مشابه  
 الگوریتم NSGA-II، در این الگوریتم نیز از دو مفهوم  
 غلبه و فاصله ازدحامی جهت امتیازدهی به تک تک  
 افراد جمعیت استفاده شده است. از بهترین الگوریتم‌های  
 بهینه‌سازی اجتماع ذرات چند هدفه، الگوریتم ارایه شده  
 توسط Coello Coello و همکاران در سال ۲۰۰۴ است  
 که در این تحقیق از آن استفاده می‌شود [38]. رویکرد  
 آنها بر مفهوم جواب‌های پارتو برای تخمین مسیر پرواز  
 هر ذره و نگهداری جواب‌های نامغلوب قبلی یافت شده  
 در یک مخزن است، به طوری که از این جواب‌ها، ذرات  
 دیگر برای مسیر پروازشان استفاده کنند.

### ۱-۶- تعریف پارامترها و اجزای بکار رفته در

#### الگوریتم:

از آنجایی که در چارچوب بهینه‌سازی چندهدفه global  
 best salutation یا بهترین جواب کلی نمی‌تواند  
 تعریف گردد، لذا از یک سطح بایگانی جواب‌های  
 غیرمغلوب به نام مخزن استفاده می‌شود و دسته ذرات  
 به صورت تصادفی از مخزن انتخاب می‌گردند. اعضای  
 مخزن آنهایی هستند که در فضای جستجو با ناحیه  
 ازدحامی کمتر، شانس بیشتری برای انتخاب شدن به  
 صورت دسته را دارند. یک ساختار مبتنی بر شبکه  
 مطابق MOPSO، یک معیار تراکم برای کاوش در  
 فضای جستجو را می‌دهد.

### ۲-۶- مقداردهی اولیه مکان و سرعت و ایجاد

#### جمعیت اولیه:

ساختار جواب در نظر گرفته شده برای مسئله ارائه شده  
 مطابق با شکل زیر می‌باشد که شامل ماتریسی سطری

<sup>1</sup> Multi-Objective Particle Swarm Optimization (MOPSO)



وابسته به پارامترهای اصلی آن شامل  $W$ ،  $c_1$  و  $c_2$  می‌باشد، تصمیم بر آن شد تا جهت افزایش کارایی الگوریتم، پارامترها در بهترین مقدار خود تنظیم شوند. یکی از کاراترین روش‌های ارایه شده در بکارگیری پارامترهای الگوریتم، طرح تطبیقی<sup>۱</sup> است که توسط کلرک و کندی<sup>۲</sup> (۲۰۰۲) ارایه شد [39]. روند محاسبه پارامترهای اصلی در روابط زیر نشان داده شده است.

$$\gamma = \frac{2}{\phi_1 + \phi_2 - 2 + \sqrt{(\phi_1 + \phi_2)^2 - 4 \times (\phi_1 + \phi_2)}}$$

$$c_1 = \gamma \times \phi_1$$

$$c_2 = \gamma \times \phi_2$$

$$w = \gamma$$

$$w = w \times w_{damp}$$

که  $\phi_1 + \phi_2 > 4$  و بهترین مقدار در  $\phi_1 = \phi_2$  و  $\phi_1 + \phi_2 = 4.1$  بدست می‌آید.

ب) موقعیت هر ذره با اضافه کردن سرعت بدست آمده از مرحله قبل با رابطه زیر بروزرسانی می‌شود.

$$POP[i] = POP[i] + VEL[i]$$

ج) ذراتی در فضای جستجو که از حدودشان تخطی نمی‌کنند، نگهداری می‌شوند.

د) هر ذره در POP ارزیابی می‌شود.

ه) محتوای REP را با توجه به مختصات ذره در مکعب‌های چند بعدی بروزرسانی می‌شود. این بروزرسانی شامل اضافه کردن همه جواب‌های نامغلوب جاری به مخزن است. در این فرآیند هریک از جواب‌های مغلوب شده از مخزن حذف می‌شوند. از آنجایی که ظرفیت مخزن محدود است، هر زمانی که ظرفیت آن تکمیل شد معیار دومی را برای بقاء بکار می‌بریم. ذراتی که در مکان‌هایی از فضای جواب با ازدحام کمتر جمعیت هستند اولویت بالاتری برای بقاء دارند.

و) زمانی که موقعیت فعلی ذره بهتر از موقعیت شامل حافظه‌اش است، موقعیت ذره با استفاده از رابطه زیر بروز

$$Pbest[i] = POP[i] \text{ می‌شود؛}$$

معیار برای تصمیم‌گیری این که موقعیت کدام ذره از حافظه بایستی باقی بماند، به آسانی با پارتوی غالب بدست می‌آید (برای مثال اگر موقعیت جاری بوسیله موقعیت در حافظه

با تعداد ستون‌هایی برابر با تعداد فعالیت‌های موجود در پروژه است. هر آرایه‌ی این ماتریس معرف *duration* یک فعالیت به خصوص می‌باشد.



### ۳-۶- مراحل تکرار:

پس از مقداردهی اولیه مکان و سرعت هر ذره، نوبت به تکرار الگوریتم می‌رسد تا شرط توقف مسئله برآورده گردد. مراحل به قرار زیر می‌باشند:

۱- ارزیابی هریک از ذرات

۲- ذخیره مکانی ذراتی که بردارهای نامغلوب ارایه می‌دهند (در مخزن REP).

۳- تولید مکعب‌های چند بعدی و قرار دادن ذرات در این مکعب‌ها (برای مثال اگر دو تابع هدف وجود داشته باشد هر محور شامل یک تابع هدف است).

۴- ارزش‌دهی دوباره حافظه هر ذره (این حافظه برای هر ذره به عنوان یک راهنما برای حرکت می‌باشد و همچنین این حافظه در مخزن REP ذخیره می‌شود).

الف) برای ذره  $i=0$  تا  $i=MAX$

ب)  $Pbest[i] = POP[i]$

۵- تا زمانی که حداکثر تکرارها بدست نیامده قدم‌های زیر را انجام می‌دهیم:

الف) سرعت هر ذره با رابطه زیر محاسبه می‌شود:

$$VEL[i] = w \times V$$

$Pbest[i]$  بهترین مکانی است که ذره  $i$  تا این لحظه پیدا کرده است.  $REP[h]$  مقداری است که از مخزن گرفته می‌شود، از آنجایی که همگرایی الگوریتم به شدت

<sup>1</sup> Adaptive version

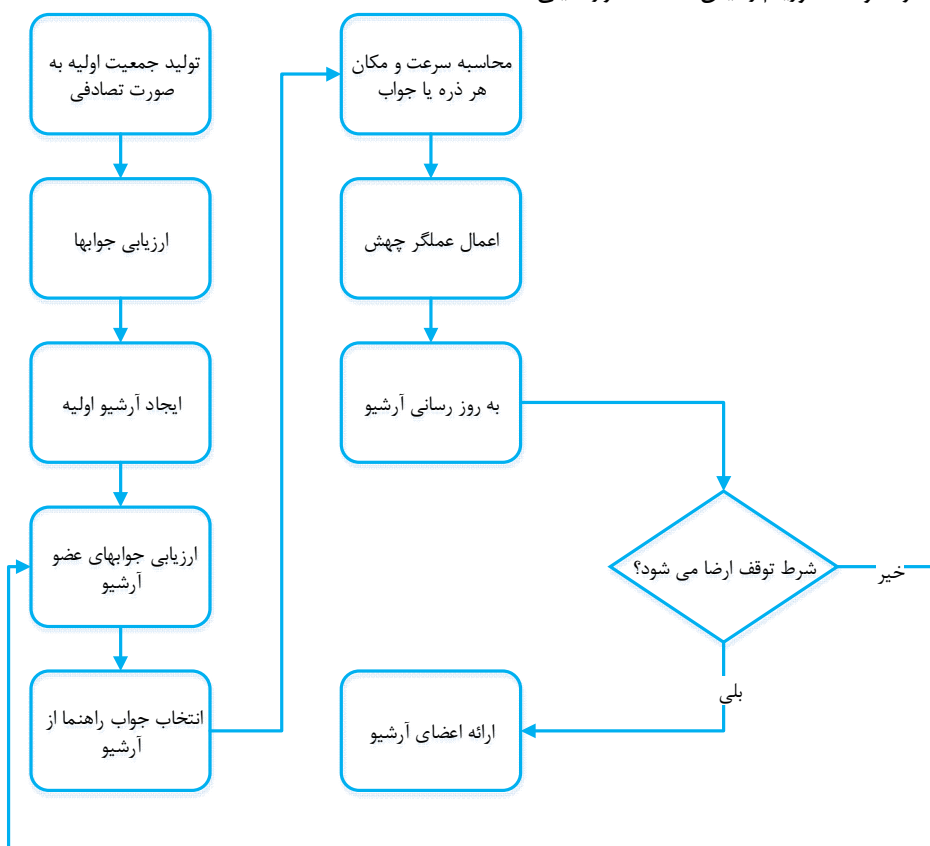
<sup>2</sup> Clerc and Kennedy

شکل ۶ فلوجارت کلی الگوریتم MOSPO را نشان می‌دهد.

مغلوب شود، موقعیت در حافظه حفظ می‌شود، در غیر این صورت موقعیت جاری با موقعیت حافظه جایگزین می‌شود و اگر هیچ‌کدام از آنها توسط دیگری مغلوب نشوند یکی از آنها را به تصادف انتخاب می‌کنیم).

(ز) افزایش شمارنده حلقه

۶- شرط توقف الگوریتم رسیدن تا تعداد تکرار معینی است.



شکل ۶ - فلوجارت الگوریتم MOSPO

## ۷- مثال عددی:

فرض این نمونه به این ترتیب است که فعالیت‌ها یا در زمان نرمال انجام می‌شوند یا با فشردگی مواجه خواهند شد با توجه به فرضیات موجود در مدل پیشنهادی این نمونه را برای حالتی که امکان انجام فعالیت‌ها با تاخیر میسر باشد، تعمیم می‌دهیم در جدول ۲ اطلاعات فعالیت‌ها آمده است.

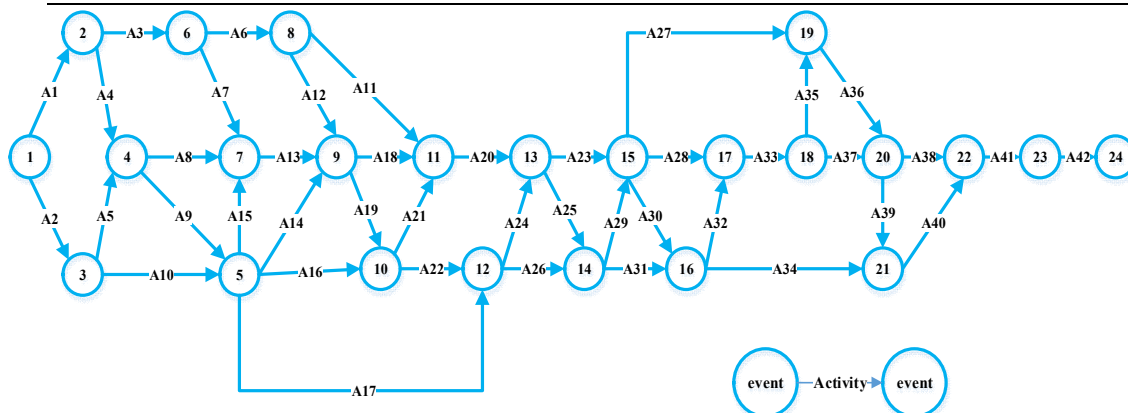
نمونه موردی از مقاله Crashing PERT network using mathematical programming شماره ۱۹ مجله International Journal of Project Management در سال ۲۰۰۱ به چاپ رسیده است، انتخاب شده است. شبکه مورد استفاده در این مقاله بارها و بارها در مقالات و تحقیقات معتبر دیگری به کار گرفته شده است و در زمینه فشردگی زمان، منبعی معتبر به شمار می‌آید.

جدول ۲ - مشخصات هزینه و زمان مربوط به فعالیت‌ها

Act	(D <sub>f</sub> )	(D <sub>n</sub> )	(D <sub>m</sub> )	(C <sub>f</sub> )	(C <sub>n</sub> )	(C <sub>m</sub> )	Act	(D <sub>f</sub> )	(D <sub>n</sub> )	(D <sub>m</sub> )	(C <sub>f</sub> )	(C <sub>n</sub> )	(C <sub>m</sub> )
A1	3	8	10	2000	500	375	A22	4	9	11	4050	1900	1425
A2	3	6	8	2050	850	637	A23	6	9	11	6020	500	375
A3	3	5	6	2480	480	360	A24	6	8	10	1900	360	270
A4	10	15	19	1830	380	285	A25	2	5	6	2070	870	652
A5	4	8	10	2100	300	225	A26	5	7	9	4000	500	375
A6	11	12	15	1890	690	517	A27	2	6	8	15020	1180	885
A7	5	9	11	320	200	150	A28	6	15	19	64600	7000	5250
A8	4	11	14	25000	1200	900	A29	5	12	15	3710	700	525
A9	5	6	8	770	230	172	A30	2	4	5	470	370	277
A10	6	8	10	615	175	131	A31	4	8	10	7720	920	690
A11	3	7	9	6420	2300	1725	A32	14	19	24	16800	5000	3750
A12	12	19	24	19980	5700	4275	A33	3	5	6	1110	940	705
A13	3	5	6	3150	550	412	A34	7	12	15	27550	5300	3975
A14	2	4	5	525	125	94	A35	5	7	9	5600	3400	2550
A15	11	13	16	3520	660	495	A36	9	11	14	3580	300	225
A16	6	9	11	2535	885	664	A37	4	8	10	8690	770	577
A17	4	7	9	12660	3000	2250	A38	6	11	14	2190	690	517
A18	6	9	11	610	250	187	A39	1	5	6	6520	3000	2250
A19	10	13	16	9980	980	735	A40	6	8	10	2940	2200	1650
A20	5	7	9	3440	1440	1080	A41	4	9	11	12600	1100	825
A21	8	11	14	3970	280	210	A42	3	8	10	3850	600	450

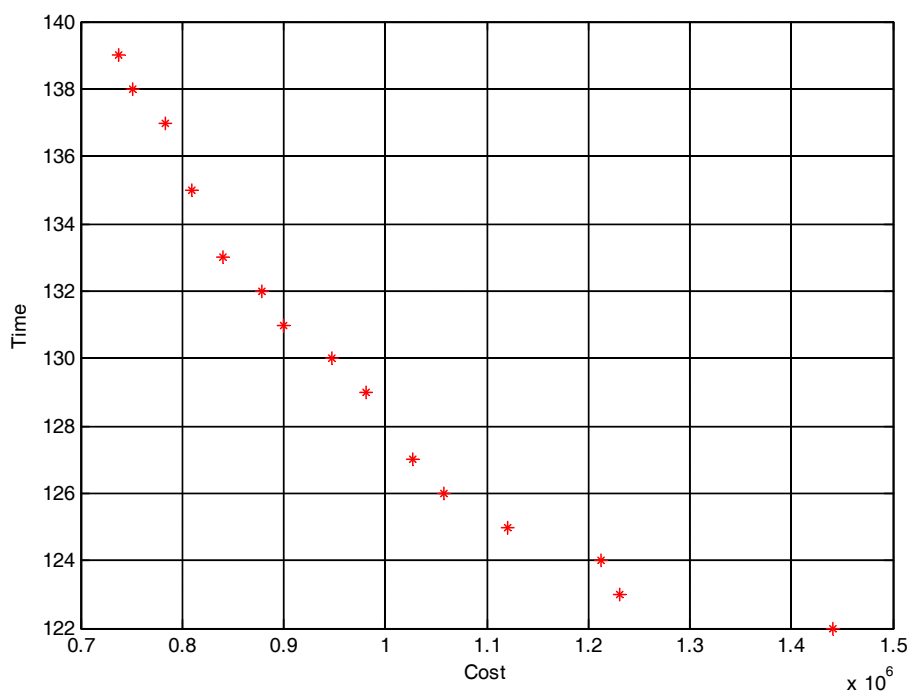
حداکثر زمان مجاز برای اتمام پروژه  $t_{max}=140$  لحاظ می‌کنیم.  
در تصویر شکل ۷، شبکه AOA مربوطه نمایش داده شده است.

در این مثال مقدار  $H=2000$ ،  $Kn=100,000$  و نرخ بهره را برابر  $I_0=0.1$  در نظر می‌گیریم. همچنین حداکثر بودجه‌ی در اختیار را  $C_{max}=1000,000$  و



شکل ۷ - دیاگرام AOA نمونه موردی مورد مطالعه

حل مثال فوق با استفاده از الگوریتم MOPSO نتایج زیر را در بر دارد که در شکل ۸ آمده است.



شکل ۸ - نمودار مجموعه جواب نا مغلوب در الگوریتم MOPSO

از صرفه‌جویی بیشتر هزینه‌ها برخوردار خواهیم بود و در این حالت میزان به تاخیر افتادن فعالیت‌ها افزایش خواهد یافت. در جدول ۳ مقدار تابع هدف اول و دوم و نیز میزان فشردگی و به تاخیر انداختن فعالیت‌ها در ۱۷ پاسخ الگوریتم MOPSO آمده است.

همان‌طور که در شکل ۸ مشخص است، هر چه به سمت اجرای سریع‌تر پروژه می‌رویم مجبور به پرداخت هزینه‌های بیشتر هستیم. در چنین شرایطی ناگزیر به فشردگی بیشتر فعالیت‌ها می‌باشیم. از سوی دیگر اگر تمایل به استفاده حداکثری از زمان مجاز اتمام پروژه داشته باشیم،

جدول ۳ - مشخصات مجموعه جواب نامغلوب نهایی در الگوریتم NSGA-II

NO.	time	Cost	Crash	Delay
1	137	783340.736	49	18
2	139	737992.270	48	18
3	131	899801.061	58	16
4	125	1120550.54	65	16
5	138	750907.495	48	18
6	126	1058317.68	60	17
7	135	810118.795	53	17
8	132	878142.461	56	17
9	133	840096.238	58	18
10	127	1026633.00	62	14
11	130	947583.241	59	16
12	129	981333.564	59	16
13	129	981333.564	59	16
14	129	981333.564	59	16
15	124	1213074.61	64	18
16	122	1441155.61	69	15
17	123	1231290.69	66	19

حضور در مزایده‌ها و رقابت‌های اقتصادی جهت جذب پروژه‌های بزرگ فراهم می‌آورند.

با توجه به نتایج بدست آمده از پاسخ ۱ تا ۱۷ به طور کلی شاهد روند صعودی زمان اتمام پروژه، روند نزولی هزینه‌ها، روند نزولی میزان فشردگی و روند صعودی میزان به تاخیر انداختن فعالیت‌ها می‌باشیم.

### نتیجه‌گیری:

در این مقاله نشان دادیم که می‌توان با در نظر گرفتن فرضیاتی، شرایط مسئله را به دنیای واقعی نزدیک‌تر گردانیم. چنان‌که لحاظ نمودن ارزش زمانی پول در مسئله تبادل هزینه و زمان یکی از مهم‌ترین عوامل تصمیم‌گیری برای مدیران و تصمیم‌گیرندگان باشد. زیرا همواره در برنامه‌ریزی اکثر قریب به اتفاق پروژه‌های بزرگ که اجرای آنها شامل مدت زمان طولانی می‌شود، تغییر ارزش پول ایجاد اخلاص می‌نماید. همچنین در نظر گرفتن تابع غیرخطی هزینه از دیگر نکات قابل توجه در این راستا می‌باشد. از سویی دیگر بدست آوردن مجموعه نامغلوب برای مدل‌های ریاضی مربوط به مسائل تبادل هزینه و زمان می‌تواند برای کارفرمایان و پیمانکاران حائز اهمیت باشد. چرا که این پاسخ‌ها انعطاف بیشتری نسبت به زمان اتمام پروژه و بودجه در اختیار دارند و به این ترتیب شرایط بهتری را برای

time/cost tradeoff problem in project network analysis. *Found Control Eng.*

[13] Demeulemeester. E, Herroelen. W, Elmaghraby. S.E (1996). Optimal procedures for the discrete time/cost trade-off problem in project networks.

[14] Demeulemeester. E, De Reyck. B, Foubert. B, et. al (1998). New computational results on the discrete time/cost trade-off problem in project networks.

[15] Erenguc. SS, Ahn. T. Conway. DG (2001). The resource constrained project scheduling problem with multiple crashable modes: an exact solution method. *Naval Res Log.*

[16] Akkan. C (1998). A Lagrangian heuristic for the discrete time-cost tradeoff problem for activity-on-arc project networks.

[17] Liu. SX, Wang. MG, Tang. LX, et. al (2000). Genetic algorithm for the discrete time/cost trade-off problem in project network. *J Northeastern Univ [China].*

[18] Peng. W, Wang. C (2009). A multi-mode resource-constrained discrete time-cost tradeoff problem and its genetic algorithm based solution.

[19] Elmaghraby. SE, Kamburowski. J (1992). The analysis of activity network under generalized precedence relations.

[20] Ann. T, Erenguc. SS (1998). The resource constrained project scheduling problem with multiple crashable modes: a heuristic procedure.

[21] Van Slyke, R.M (1963). Monte carlo methods and the PERT problem.

[22] Vanhoucke. M, Demeulemeester E, Herroelen W (2002). Discrete time/cost trade-offs in project scheduling with time-switch constraints.

[23] M. Sasaki and M. Gen (2003). A method of fuzzy multi-objective nonlinear programming with GUB structure by hybrid

## فهرست منابع:

[1] Hindelang. T.J, Muth. J.F (1979). A dynamic programming algorithm for decision CPM networks. *Operat Res.*

[2] Prabuddha. D.E, Dunne. E.J, Ghosh. J.B, et. al (1997). Complexity of the discrete time-cost tradeoff problem for project networks. *Operat Res.*

[3] Vladimir. G Deineko, Woeginger Gerhard. J (2001). Hardness of approximation of the discrete time-cost tradeoff problem. *Operat Res Lett.*

[4] Kelley. J.E, Walker. M.R (1959). Critical path planning and scheduling: An introduction. Mauchly Associates, Ambler, PA.

[5] Fulkerson. D.R. (1961). A network flow computation for project cost curves.

[6] Kelley. J.E (1961). Critical path planning and scheduling: Mathematical basis.

[7] Ford. L.R, Fulkerson. D.R (1962). *Flows in Networks.* Princeton University Press, Princeton, NJ.

[8] Siemens. N (1971). A simple CPM time/cost trade-off algorithm. *Management Science.*

[9] Goyal, S.K (1975). A note on the paper: A simple CPM time/ cost trade-off algorithm. *Management Science.*

[10] Elmaghraby. S.E, Salem. A (1981). Optimal linear approximation in project compression. *OR Technical Report 171,* North Carolina State University at Raleigh.

[11] Moder. J.J, Phillips. C.R, Davis. E.W (1983). *Project Management with CPM, PERT and Precedence Diagramming,* third ed. Van Nostrand Reinhold Company, New York.

[12] Patterson. JH, Harvey. RT (1979). An implicit enumeration algorithm for the

- dominated sorting genetic algorithm for multiobjective optimization: NSGA-II, In: proceedings of the parallel problem solving from nature VI (PPSN-VI)conference.
- [35] C.A. Coello Coello (2006). Evolutionary multi-objective optimisation: an historical view of the field, Computational Intelligence Magazine.
- [36] Deb. K (2001). Multi-objective optimization using evolutionary algorithms, Chichester.
- [37] Moore. J, Chapman. R (1999). Application of particle swarm to multi-objective optimization”, Department of Computer Science and Software Engineering, Auburn University.
- [38] Coello Coello. C.A, Pulido. G.T, Lechuga. M.S (2004). Handling multiple objectives with particle swarm optimization, IEEE Transactions on Evolutionary Computation.
- [39] Clerc. M, Kennedy. J (2002). The particle swarm-explosion, stability, and convergence in a multi-dimensional complex space, IEEE Transaction on Evolutionary Computation
- genetic algorithm, International Journal of Smart Engineering Design
- [24] M. Sasaki and M. Gen (2003). Fuzzy multiple objective optimal system design by hybrid genetic algorithm, Applications of Software Computation.
- [25] Pasandideh. S.H.R, Niaki. S.T.A, Hajipour. V (2011). A Multi-Objective Facility Location Model with Batch Arrivals: Two Parameter-Tuned Meta-heuristic Algorithms. Journal of Intelligent Manufacturing.
- [26] The report of the world commission on Dams(WCD), Dams and Development, Earthscan publications L+d, London and sterling, VA.November 2000.
- [27] Shahriari. M et. al (2010).A new mathematical model for tome cost trade-off problem with budget limitation based on time value of money.
- [28] Shahriari. Mohammad Reza (2016). Multi-objective optimization of discrete time-cost tradeoff problem in project networks using non-dominated sorting genetic algorithm. Journal of industrial engineering international.
- [29] Ameen (1987). A computer assisted pert simulation. Journal of systems management.
- [30] Sou – sen. leu (2000). A GA – based fuzzy optimal model for construction time- cost trade off International journal of project management.
- [31] Heng. Li ,J-N. Cao (1999). Using Machine Learning and GA to Solve Time-Cost Trade-Off Problems, journal of project management.
- [32] Scott. A.Burns (1994). The LP/IP hybrid method for constru ction time – cost trade off analysis, Construction Management and Economics journal.
- [33] Chau D.K.H, Chan W.T,K. Govindan (1997). A time-cost trade-off model with resource consideration using genetic algorithm, Civil Engineering Systems.
- [34] Deb. K., Agrawal. S, Pratap. A, Meyarivan. T (2000). A fast elitist non-

