

## زمانبندی وظایف در محیط محاسبات ابری با استفاده از الگوریتم‌های تکاملی

### پریا اردستانی

گروه مهندسی کامپیوتر، واحد تهران جنوب، دانشگاه آزاد اسلامی، تهران، ایران  
Pariya.ardestani@gmail.com

### کامبیز بدیع

عضو هیئت علمی و دانشیار، پژوهشگاه ارتباطات و فناوری اطلاعات (مرکز تحقیقات مخابرات ایران)  
k\_badie@itrc.ac.ir

### چکیده

محاسبات ابری الگویی از محاسبات توزیع‌شده، مرکب از تعداد زیادی منابع و درخواست‌ها با هدف به اشتراک‌گذاری منابع به صورت سرویس، بر روی بستر اینترنت است. منابعی مانند حافظه، پردازشگر، سرویس همیشه با ارزش می‌باشند و استفاده بهینه از آنها یک چالش بی‌انتهای محسوب می‌شود، از این رو مسئله زمانبندی وظایف در رایانش ابری، مسئله‌ای بسیار مهم است که سعی دارد یک زمانبندی بهینه برای اجرای وظایف مشخص نماید. در این مقاله با استفاده از الگوریتم بهینه‌سازی علف‌های هرز زمانبندی وظایف در محیط ابر در جهت کمینه‌سازی زمان تکمیل وظایف انجام شده است. در جهت بهبود عملکرد الگوریتم بهینه‌سازی علف‌های هرز از عملگرهای تعویض و وارون‌سازی استفاده شده است تا تنوعی بیشتری در فضای پاسخ ایجاد شود و در صورت افتادن در نقاط بهینه محلی باعث خروج و یافتن پاسخ بهینه سراسری شوند. پس از مدل‌سازی مسئله با استفاده از الگوریتم بهینه‌سازی علف‌های هرز و بهبود آن، زمانبندی وظایف به ازای ۱۵، ۲۰، ۲۵ و ۳۰ وظیفه بر روی ۴ پردازنده انجام می‌شود و نتایج آن با الگوریتم ژنتیک مقایسه می‌شود. نتایج حاکی از عملکرد بهینه الگوریتم بهینه‌سازی علف‌های هرز با زمان تکمیل وظایف ۱۲۶، ۱۵۶، ۱۹۶ و ۲۴۵ به ترتیب براساس ۱۵، ۲۰، ۲۵ و ۳۰ وظیفه در برابر الگوریتم ژنتیک است.

**واژگان کلیدی:** محاسبات ابری، زمانبندی وظایف، الگوریتم بهینه‌سازی علف‌های هرز.

**۱- مقدمه**

محاسبات ابری ساختاری شبیه به یک توده ابر دارد و از سوپر کامپیوترهایی تشکیل شده است که از تکنولوژی‌های مجازی سازی، ذخیره‌سازی منابع و ارائه محاسبات امن استفاده می‌کنند و کاربران می‌توانند با استفاده از اینترنت به برنامه‌های کاربردی از هر جای دنیا دسترسی داشته باشند (Zhangjun, et al. 2010). زمانبندی وظایف نقشی کلیدی در بهبود کارایی و بهره‌وری محاسبات ابری دارد و با هدف افزایش کارایی و کیفیت سیستم، بالابردن قابلیت اطمینان، کاهش زمان پاسخ‌دهی و استفاده بهینه از منابع مطرح می‌شوند. وظیفه زمانبند وظایف این است که بر اساس وضعیت فعلی سیستم و وظایف جدید دریافتی، بهترین تصمیم را اتخاذ نماید. تاکنون الگوریتم‌های ابتدایی زیادی در زمینه زمانبندی در محاسبات ابری پیشنهاد شده و مورد بررسی قرار گرفته است که دارای مشکلات عمده‌ای از قبیل کیفیت سرویس پایین، هزینه اجرا بالا، ضعف در تقسیم بار، ضعف در مدیریت پهنای باند شبکه ابری، عدم پردازش مناسب داده‌ها در مقیاس بزرگ می‌باشند (Kumar and Verma, 2012). لذا در این مقاله روش جدیدی از زمانبندی وظایف با استفاده از الگوریتم بهینه‌سازی علف‌های هرز ارائه می‌شود. ادامه ساختار این مقاله به این صورت است که در بخش دوم مروری بر پیشینه تحقیقات انجام شده می‌شود و در بخش سوم الگوریتم‌های تکاملی بیان می‌شوند. در بخش چهارم مدل پیشنهادی و ارزیابی نتایج استفاده از الگوریتم‌های تکاملی در جهت حل مسأله ارائه می‌شود و در نهایت در بخش آخر نتیجه‌گیری انجام می‌شود.

**۲- مروری بر پیشینه تحقیق**

یو و همکاران در سال ۲۰۰۶ به زمانبندی با محدودیت منابع برنامه‌های بر روی ابر با استفاده از الگوریتم ژنتیک پرداختند. در محاسبات ابری کاربران از خدماتی که در بستر ابر فراهم شده است، استفاده می‌کنند در نتیجه کاربران خدمات را از ابر دریافت کرده و متعاقب آن هزینه پرداخت خواهند کرد. نحوه زمانبندی اجرای برنامه‌ها در جهت افزایش کیفیت سرویس انجام شده است. لذا در این پژوهش روشی برای زمانبندی در جهت محدودیت بودجه ارائه شده و زمان اجرای برنامه‌ها را در حالی که نیازمند یک بودجه معین برای کارهای محول شده است کمینه می‌کند. نسخه جدیدی از الگوریتم ژنتیک ارائه شد. نتایج شبیه‌سازی حاکی از بهبود در زمانبندی بر بستر آزمایش است (Yu and Buyya, 2006). رحمان و همکاران در سال ۲۰۰۷ در مقاله خود تحت عنوان الگوریتم مسیر بحرانی پویا برای زمانبندی جریان کاری وظایف در ابر، یک روش DCP براساس الگوریتم زمانبندی جریان کاری که نگاهی کارا از وظایف را توسط محاسبه کردن مسیر بحرانی در گراف وظایف جریان کاری تعیین می‌کند. عملکرد روش پیشنهادی با دیگر روش‌های ابتکاری و فراابتکاری براساس استراتژی‌های زمانبندی برای انواع مختلف و اندازه‌های مختلف جریان کاری حاکی از آن دارد که روش پیشنهادی در مقایسه با روش‌های دیگر از عملکرد بهتری برخوردار است (Rahman et al, 2007). چن و همکاران در سال ۲۰۰۹ با الگوریتم مورچگان به مسأله زمانبندی وظایف در ابر با توجه به نیازهای افزایش کیفیت سرویس پرداختند. با توجه به اینکه در زمینه ابر الگوریتم‌های زمانبندی زیادی معرفی شده است و هرکدام از آنها تنها بخشی از چالش‌های به وجود آمده در این حوزه را برطرف می‌کنند و یا وظایف را می‌توانند مدیریت کنند. در این تحقیق یک روش زمانبندی براساس الگوریتم مورچگان برای جریان کاری با اندازه بزرگ و با پارامترهای زیاد در جهت افزایش کیفیت سرویس ارائه شده است. الگوریتم پیشنهادی کاربران را قادر می‌سازد تا اولویت‌های مختلف افزایش کیفیت سرویس را برآورده کرده و همین‌طور آستانه کمینه کیفیت سرویس را برای یافتن راه‌حلی که تمامی محدودیت‌های افزایش کیفیت سرویس را رفع کند و پارامترهای آن را که برای کاربران حائز اهمیت است را بهینه کند. در این مقاله ۷ روش ابتکاری برای الگوریتم مورچگان طراحی شده است و یک برنامه تطبیق‌پذیر که به مورچگان اجازه می‌دهد روش‌های هوشمند را براساس مقادیر فرمون انتخاب کنند (Chen et al, 2009). پاندى و همکاران در سال ۲۰۱۰ به زمانبندی جریان کارها براساس الگوریتم بهینه‌سازی ازدحام ذرات در محاسبات ابر پرداختند. در این مقاله روشی برای بهینه‌کردن زمان اجرا و هزینه ناشی از انتقال داده بین منابع و هزینه اجرا ارائه شد. روش پیشنهادی برنامه جریان کاری توسط تغییر محاسبات و هزینه‌های

برقراری ارتباط آزمایش شد و مقایسه‌ای براساس صرفه‌جویی در هزینه با استفاده از الگوریتم بهینه‌سازی ازدحام ذرات انجام شد. نتایج حاکی از کاهش هزینه با استفاده از الگوریتم بهینه‌سازی ازدحام ذرات تا ۳ برابر نسبت به الگوریتم BRS دارد (Pandey et al, 2010). مائو و همکاران در سال ۲۰۱۱ به خود مقایسه سنجی برای کمیته‌سازی هزینه و برطرف کردن موعد تحویل وظایف در جریان‌های کاری در ابر پرداختند. در این پژوهش روشی براساس عناصر محاسباتی بنیادی در ماشین‌های مجازی با هزینه‌ها و اندازه‌های متفاوت ارائه شد. کاربران با استفاده از این روش می‌توانند نیازمندی‌های عملیاتی خود را توسط تعیین موعد تحویل برای کارها برآورده کنند. هدف در اینجا اطمینان از این است که تمامی کارها در زمان مقرر و با کمترین هزینه مالی انجام شوند. برای رسیدن به این اهداف تخصیص وظایف بر روی ماشین‌های مجازی و زمانبندی کارها به‌طور پویا انجام شد. این روش با ۴ روش دیگر در زمینه محاسبات ابری مورد مقایسه قرار گرفته که نشان از صرفه‌جویی در هزینه از ۹,۸٪ تا ۴۰,۴٪ در مقایسه با دیگر روش‌ها دارد (Mao and Humphrey, 2011). مالوسکی و همکاران در سال ۲۰۱۲ به محدودیت موعد تحویل و هزینه ایجاد شده بر روی زیرساخت به عنوان سرویس پرداختند. در این مقاله الگوریتم‌های موردنظر به صورت پویا و ایستا برای زمانبندی وظایف و تخصیص منابع مورد ارزیابی واقع شدند. ارزیابی‌های انجام شده براساس شبیه‌سازی مجموعه‌ای از جریان‌های کاری با انواع مختلفی از بودجه و موعد تحویل است. عدم قطعیت در زمان اجرا و تأخیر و نقص در انجام کارها را در نظر می‌گیرد. نتایج حاکی از آن است که عملکرد یک الگوریتم توسط توانایی آن برای تصمیم‌گیری در جریان‌های کاری برای اجرا یا رد درخواست برای اجرا شدن دارد. نتایج نشان می‌دهد که رویه‌های پذیرفته شده براساس ساختار جریان کار و تخمین زمان اجرای وظایف می‌تواند به طور موثری کیفیت راه‌حل‌ها را بهبود بخشد (Malawski et al, 2012). ابریشمی و همکاران در سال ۲۰۱۳ به الگوریتم‌های زمانبندی جریان کاری با محدودیت موعد تحویل به عنوان زیرساخت به عنوان سرویس در ابر پرداختند. در این مقاله الگوریتم PCP در محیط ابر مورد آزمایش قرار گرفت. همچنین دو الگوریتم زمانبندی جریان کاری ارائه شد. الگوریتم اول با نام IC-PCP یک مرحله‌ای است و الگوریتم دوم IC-PCPD2 نام دارد. هردوی این الگوریتم‌ها دارای پیچیدگی زمانی درجه ۲ هستند که آنها را گزینه‌ای مناسب برای زمانبندی جریان‌های کاری می‌کند. نتایج نشان می‌دهد هردوی الگوریتم‌ها عملکرد قابل قبولی دارند اما الگوریتم IC-PCP در اکثر موارد عملکرد بهتری از خود نشان می‌دهد (Abrishami et al, 2013). رودریگز و همکاران در سال ۲۰۱۴ به ایجاد منابع براساس موعد تحویل و الگوریتم زمانبندی بر روی جریان‌های کاری در ابر پرداختند. روش پیشنهادی در پی فراهم کردن منابع موردنیاز در بستر ابر و ایجاد استراتژی زمانبندی برای جریان‌های کاری در زیرساخت به عنوان سرویس است. الگوریتم پیشنهادی PSO بود و هدف آن کمیته‌سازی هزینه اجرای کلی جریان کاری با محدودیت زمان تحویل است. شبیه‌سازی‌های انجام شده با جریان‌های کاری با اندازه‌های مختلف نشان از بهبود این روش نسبت به روش‌های دیگر را نشان می‌دهد (Rodriguez et al, 2014).

### ۳- الگوریتم‌های تکاملی

مفهوم بهینه‌سازی بدین صورت است که در بین پارامترهای یک تابع به دنبال مقادیری باشیم که تابع را کمینه یا بیشینه نماید. کلیه مقادیر مناسب جهت این امر را، راه‌حل‌های ممکن و بهترین مقدار از این مقادیر را، راه‌حل بهینه می‌نامند. الگوریتم‌های تکاملی هر دو نوع مسائل بیشینه‌سازی و کمیته‌سازی را پوشش می‌دهند. بهینه‌سازی همواره با مشکلات فراوانی همراه بوده است. شیوه‌های سابق برای حل کردن مشکلات بهینه‌سازی، مستلزم تلاش‌های محاسباتی بی‌شماری است. الگوریتم‌هایی از جمله الگوریتم‌های هوش جمعی تا حدی این مشکل را حل نموده‌اند. توسط این الگوریتم‌ها راه‌حل‌هایی پیدا می‌شوند که تقریباً به جواب نزدیکند (Shilane et al, 2008).

### ۳-۱- الگوریتم بهینه‌سازی علف‌های هرز

الگوریتم بهینه‌سازی علف‌های هرز یک الگوریتم بهینه‌سازی عددی الهام گرفته‌شده از رشد علف‌های هرز است. این الگوریتم در سال ۲۰۰۶ توسط محرابیان و لوکاس در قالب مقاله‌ای پیشنهاد شد. علف‌های هرز گیاهانی هستند که رشد هجوم آورنده و شدید آن‌ها تهدید مهمی برای گیاهان زراعی محسوب می‌شود. علف‌های هرز بسیار پایدار و تطابق‌پذیر در مقابل تغییرات محیط می‌باشند، بنابراین با الهام گرفتن و شبیه‌سازی خصوصیات آن‌ها می‌توان به یک الگوریتم بهینه‌سازی قوی رسید (Mehrabian and Lucas, 2006).

**مقداردهی اولیه جمعیت:** توزیع یک جمعیت اولیه در فضای  $d$  بُعدی به صورت تصادفی یک راه‌حل اولیه است. **تولیدمثل:** هر یک از اعضای جمعیت علف‌ها مجاز است بسته به توان و شرایط خود تولیدمثل کند. هر علف می‌تواند به طور خطی حداقل و یا حداکثر تعدادی دانه تولید کند. به عبارت دیگر، دانه‌های یک علف بسته به شرایط آن به طور خطی مورد تولید حداکثر و حداقل قرار می‌گیرند (Mehrabian and Lucas, 2006).

تصادفی نبودن و سازگاری در الگوریتم در این بخش ارائه شده است. به طور تصادفی با توزیع نرمال و با میانگین صفر، توزیع دانه‌های تولیدشده در فضای جستجو  $d$  بُعدی شکل می‌گیرد، اما واریانس متفاوت است. این به این معنی است که دانه‌های تولیدشده به طور تصادفی نزدیک والد خود هستند. با این حال، انحراف معیار  $(\sigma)$ ، تابع تصادفی خواهد بود و از مقدار اولیه قبلی تعریف شده‌اش کاهش می‌یابد و در هر مرحله (نسل)  $\sigma_{init}$  به ارزش نهایی  $\sigma_{final}$  تبدیل می‌شود. در شبیه‌سازی، تغییر غیرخطی نشان داده می‌شود که در رابطه (۱) بیان شده است (Mehrabian and Lucas, 2006).

$$\sigma_{iter} = \frac{(iter_{max} - iter)^n}{(iter_{max})^n} (\sigma_{init} - \sigma_{final}) + \sigma_{final} \quad (1)$$

که  $iter_{max}$  بیشترین مقدار تکرار و  $\sigma_{iter}$  انحراف معیار در هر مرحله و  $n$  شاخص مدولاسیون غیرخطی است. این تغییر تضمین می‌کند که احتمال از بین رفتن دانه در یک منطقه دور غیرخطی کاهش می‌یابد (Mehrabian and Lucas, 2006). **حذف رقابتی:** اگر یک علف حذف شود دانه‌های آن هم حذف خواهند شد. لازم است از نوعی رقابت بین علف‌ها برای محدود کردن حداکثر تعداد رشد در یک کلونی جلوگیری کرد. بعد از هر تکرار، تعداد علف‌ها به سرعت در کلونی زیاد می‌شود و انتظار می‌رود که علف‌های مشخصه بیش از علف‌های نامطلوب تکثیر شده باشند. با نزدیک شدن به حداکثر تعداد علف‌ها،  $p_{max}$  یک مکانیسم برای از بین بردن علف‌ها با شرایط نامناسب در هر نسل است. مکانیسم از بین بردن علف‌ها به این شرح است: با توجه به مکانیسم هر علف هرز مجاز به تولید دانه است، که در تولیدمثل ذکر شد. دانه‌های تولید شده مجاز به گسترش در ناحیه جستجو می‌باشند، که در پراکندگی محیط گفته شد. زمانی که همه‌ی دانه‌ها موقعیت خود را در منطقه‌ی جستجو پیدا کردند، آن‌ها با والد خود رتبه‌بندی می‌شوند (به عنوان یک کلونی از علف‌های هرز). سپس، برای رسیدن به حداکثر جمعیت مجاز در یک کلونی، علف‌های هرز با شایستگی کمتر حذف می‌شوند. در این روش، علف‌ها و دانه‌ها با هم رتبه‌بندی شده و آن‌هایی که شایستگی بهتری دارند، مجاز به تکثیر می‌شوند. این مکانیسم یک فرصت به علف‌ها با شایستگی بهتر می‌دهد و اگر دانه‌های آن‌ها دارای شایستگی مناسب باشند می‌توانند زنده بمانند. این مکانیسم خروج رقابتی را تحقق می‌دهد (Mehrabian and Lucas, 2006).

الگوریتم IWO فرصت آزمایش با استراتژی‌های مختلف کشف فضاهای جستجو را فراهم می‌کند. مراحل زیر الگوریتم را با استفاده از کنوانسیون اصطلاحات سازگار و با الهام از ایده‌های «طبیعی» بیان می‌کند.

- تولید و ایجاد اولین جمعیت متشکل از  $n$  عضو به طور تصادفی.
- برای هر عضو، محاسبه برای به حداقل رساندن مقدار تابع شایستگی به صورت عددی متقابل

- تا زمانی که تابع معیار شرایط مناسبی ندارد برای هر عضو از کلونی، محاسبه تعداد دانه بسته به مقدار تابع شایستگی برای هر دانه، تعیین محل قرارگرفتن دانه با احتمال ثابت توسط یکی از روش‌های پراکنده‌سازی، گسترش یا غلطش دانه.
- ایجاد یک عضو جدید به طور تصادفی.
- محاسبه مقدار تابع شایستگی برای عضو جدید.

ایجاد یک جامعه جدید متشکل از  $n$  عضو با بهترین شایستگی و با توجه به اعضای جمعیت سابق و همچنین اعضای جمعیت جدید (Mehrabian and Lucas, 2006).

### ۳-۲- الگوریتم ژنتیک

الگوریتم ژنتیک، الهامی از علم ژنتیک و نظریه تکامل داروین است و براساس بقای برترین‌ها یا انتخاب طبیعی استوار است. این الگوریتم برای کاربردهای مهندسی و به‌صورت امروزی آن نخستین‌بار توسط جان هالند متخصص علوم کامپیوتر دانشگاه میشیگان در سال ۱۹۷۵ پیشنهاد گردید. یک کاربرد متداول الگوریتم ژنتیک، استفاده از آن به‌عنوان تابع بهینه‌کننده است. اگرچه این الگوریتم پس از الگوریتم استراتژی تکاملی پیشنهاد گردید ولی مشهورترین روش از بین الگوریتم‌های تکاملی است. در یک الگوریتم ژنتیک یک جمعیت از افراد طبق مطلوبیت آنها در محیط بقا می‌یابند. افرادی با قابلیت‌های برتر، شانس ازدواج و تولیدمثل بیش‌تری را خواهند یافت. بنابراین بعد از چند نسل فرزندان با کارایی بهتر به‌وجود می‌آیند. در الگوریتم ژنتیک هر فرد از جمعیت به‌صورت یک کروموزوم معرفی می‌شود. کروموزوم‌ها در طول چندین نسل کامل‌تر می‌شوند. در هر نسل کروموزوم‌ها ارزیابی می‌شوند و متناسب با ارزش خود امکان بقا و تکثیر می‌یابند (Goldberg, 2012). به‌طور کلی، الگوریتم‌های ژنتیک از اجزاء زیر تشکیل می‌شوند:

- **کروموزوم:** یک نقطه در فضای جستجو و یک راه‌حل ممکن برای مسأله موردنظر است. کروموزوم‌ها از تعداد ثابتی ژن (متغیر) تشکیل شده‌اند. برای نمایش کروموزوم‌ها معمولاً از کدگذاری‌های دودویی (رشته‌های بیتی) استفاده می‌شود.
- **جمعیت:** به مجموعه‌ای از کروموزوم‌ها گفته می‌شود که با تأثیر عملگرهای ژنتیکی بر روی هر جمعیت، جمعیت جدیدی با همان تعداد کروموزوم تشکیل می‌شود.
- **تابع برازندگی:** برای هر کروموزوم، این تابع عددی غیرمنفی را برمی‌گرداند که نشان‌دهنده شایستگی یا توانایی فردی آن کروموزوم است.

در الگوریتم‌های ژنتیک، در طی مرحله تولیدمثل از عملگرهای ژنتیکی استفاده می‌شود. با تأثیر این عملگرها بر روی یک جمعیت، نسل بعدی آن جمعیت تولید می‌شود. عملگرهای انتخاب، تقاطع و جهش معمولاً بیش‌ترین کاربرد را در الگوریتم‌های ژنتیکی دارند (Goldberg, 2012).

**انتخاب والدین:** انتخاب والدین می‌تواند به سه صورت انجام شود. روش اول، والدین به‌صورت تصادفی انتخاب می‌شوند در این روش هر یک از اعضای جمعیت با شانس یکسان می‌توانند انتخاب شوند. در روش دوم، والدین براساس چرخ رولت انتخاب می‌شوند در این روش هر یک از اعضای جمعیت که مقدار هدف بهتری داشته باشد، شانس بیش‌تری برای انتخاب شدن دارد. در روش سوم، والدین براساس روش انتخاب رقابتی انتخاب می‌شوند. در این روش ابتدا بخشی از اعضای جمعیت به‌صورت تصادفی انتخاب می‌شوند و سپس عضوی از اعضای انتخاب شده که بهترین مقدار تابع هدف را داشته باشد به‌عنوان برنده رقابت انتخاب می‌شود (Goldberg, 2012).

**تولیدمثل (تقاطع):** عملگر اصلی در ایجاد نسل جدید است. هر یک از فرزندان حاصل از تقاطع بخشی از خصوصیات پدر و بخشی از خصوصیات مادر را باهم به‌ارث می‌برند. انواع تقاطع به صورت تک‌نقطه‌ای و دونقطه‌ای وجود دارد. در حالت تک‌نقطه‌ای، دو کروموزوم والد از یک نقطه شکسته شده و کروموزوم‌های فرزندان از جابه‌جایی قسمت اول دو والد تشکیل می‌شوند. در

تقاطع دو نقطه‌ای، دو کروموزوم والد از دو نقطه شکسته شده و کروموزوم‌های فرزندان از جابه‌جایی قسمت دوم دو والد ایجاد می‌شوند (Goldberg, 2012).

**جهش:** این عملگر یک یا چند ژن از یک کروموزوم را تغییر می‌دهد (Goldberg, 2012).

روند کلی الگوریتم به این صورت است، قبل از این که یک الگوریتم ژنتیک بتواند اجرا شود، ابتدا باید کدگذاری (نمایش) مناسبی برای مسأله مورد نظر پیدا شود. معمولی‌ترین شیوه نمایش کروموزوم‌ها در الگوریتم ژنتیک به شکل رشته‌های دودویی است. هر متغیر تصمیم‌گیری به صورت دودویی درآمده و سپس با کنار هم قرارگرفتن این متغیرها کروموزوم ایجاد می‌شود. هم‌چنین یک تابع برازندگی نیز باید ابداع شود تا به هر راه‌حل کدگذاری شده ارزشی را نسبت دهد. در طی اجرا، والدین برای تولیدمثل انتخاب می‌شوند و با استفاده از عملگرهای آمیزش و جهش با هم ترکیب می‌شوند تا فرزندان جدیدی تولید کنند. این فرآیند چندین بار تکرار می‌شود تا نسل بعدی جمعیت تولید شود. سپس این جمعیت بررسی می‌شود و در صورتی که ضوابط همگرایی برآورده شوند، فرآیند فوق خاتمه می‌یابد (Goldberg, 2012).

#### ۴- ارائه مدل پیشنهادی

برای شبیه‌سازی مسأله زمانبندی وظایف در محیط ابر مفروضات زیر در فرآیند شبیه‌سازی در نظر گرفته شده است:

I: تعداد وظایف در محیط ابر

J: تعداد پردازنده‌ها برای پردازش وظایف

$P_{ij}$ : زمان پردازش وظیفه  $i$ ام بر روی پردازنده  $j$ ام

$S_{i_1, i_2, j}$ : زمان آماده‌سازی وظیفه  $i_2$  بعد از اتمام اجرای وظیفه  $i_1$  بر روی پردازنده  $j$ ام

#### ۴-۱- قیدها در فرآیند شبیه‌سازی

(۱) همه وظایف بایستی اجرا شوند.

(۲) هر وظیفه فقط و فقط توسط یک پردازنده اجرا می‌شود.

(۳) هر وظیفه فقط یک‌بار اجرا می‌شود.

(۴) زمان شروع وظیفه بعدی توسط یک پردازنده برابر است با زمان اتمام وظیفه قبلی به علاوه زمان آماده‌سازی وظیفه بعدی (طبق رابطه (۲))

$$\text{Start Time(Next Task)} = \text{Finish Time(Previous Task)} + \text{Setup Time(Next Task)} \quad (۲)$$

(۵) زمان اتمام وظیفه فعلی برابر است با زمان شروع وظیفه فعلی به علاوه زمان اجرای وظیفه فعلی (طبق رابطه (۳))

$$\text{Finish Time(Current Task)} = \text{Start Time(Current Task)} + \text{Process Time(Current Task)} \quad (۳)$$

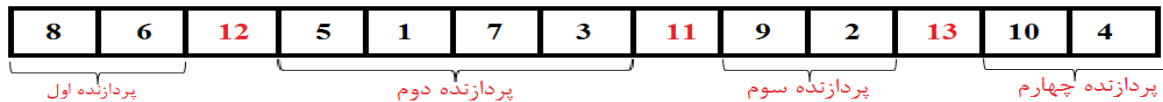
#### ۴-۲- تعریف متغیر تصمیم در شبیه‌سازی

برای شبیه‌سازی متغیرهای تصمیم در مسأله از کدگذاری گسسته استفاده شده است و ساختار متغیر تصمیم شامل  $I+J-1$  بعد است. با فرض اینکه تعداد وظایف برابر ۱۰ و تعداد پردازنده‌ها برابر ۴ باشد شکل (۱) نحوه کدگذاری متغیر تصمیم را نشان می‌دهد.

8	6	12	5	1	7	3	11	9	2	13	10	4
---	---	----	---	---	---	---	----	---	---	----	----	---

شکل ۱- شکل متغیر تصمیم

در شکل (۱) اعداد سیاه رنگ نشان‌دهنده شماره وظایف و اعداد قرمز رنگ نشان‌دهنده جدادکننده‌ها است یعنی وظایفی که بایستی توسط هر پردازنده اجرا شوند را نشان می‌دهد. در این روش از ابتدای لیست شروع کرده و تا رسیدن به هر جداکننده وظایفی که مشاهده شوند به ترتیب به پردازنده‌ها تخصیص می‌یابند. شکل (۲) نحوه تخصیص وظایف به پردازنده‌ها را نشان می‌دهد.



شکل ۲- نحوه تخصیص وظایف به پردازنده‌ها

#### ۴-۳- تعریف تابع هدف

در مسأله زمانبندی وظایف در محیط ابر هدف زمانبندی و تخصیص مناسب وظایف به پردازنده‌ها در جهت کمینه‌سازی زمان تکمیل اجرای وظایف است. بنابراین تابع هدف به صورت رابطه (۴) و به صورت کمینه‌سازی زمان تکمیل وظایف تعریف می‌شود.

$$f(x) = \text{Min} (\text{CpuCompletionTime}_i) \quad \forall i \quad (4)$$

به عبارت دیگر در این شبیه‌سازی برای هر راه‌حل، زمان تکمیل وظایف هر پردازنده مشخص می‌شود سپس پردازنده‌ای که دارای بیشترین زمان تکمیل وظایف باشد به عنوان مقدار تابع هدف در نظر گرفته می‌شود.

#### ۴-۴- نتایج استفاده از الگوریتم‌های تکاملی در زمانبندی وظایف

در این بخش قصد داریم به بررسی نتایج شبیه‌سازی زمانبندی وظایف در محیط ابر با استفاده از الگوریتم‌های تکاملی بپردازیم. الگوریتم‌های بهینه‌سازی علف‌های هرز و ژنتیک از جمله الگوریتم‌های هستند که در جهت زمانبندی وظایف در محیط ابر استفاده شده‌اند. در کلیه مراحل شبیه‌سازی تعداد پردازنده‌ها برابر ۴ و تعداد وظایف ۱۵، ۲۰، ۲۵ و ۳۰ در نظر گرفته شده است.

#### ۴-۴-۱- بهبود الگوریتم‌های تکاملی با استفاده از عملگرها

به دلیل اینکه مسأله زمانبندی وظایف به صورت یک مسأله جایگشتی تعریف شده است بنابراین در این مقاله قصد داریم با اعمال عملگرهای تعویض و وارون‌سازی باعث تنوع در فضای پاسخ الگوریتم‌ها شویم. اعمال عملگرها باعث اجتناب از همگرایی به بهینه محلی و ایجاد تنوع و گوناگونی در فضای پاسخ می‌شود.

#### ۴-۴-۱-۱- روش تعویض

در این روش دو بعد یا دو بلوک از پاسخ به‌طور تصادفی انتخاب شده و موقعیت آن‌ها تعویض می‌شود. شکل (۳) مثالی از روش تعویض را نمایش می‌دهد.



شکل ۳- روش تعویض

**۴-۱-۲- روش وارون سازی**

دو بعد از پاسخ به طور تصادفی انتخاب شده و ترتیب آنها در بین این دو بعد معکوس می‌شود. شکل (۴) مثالی از روش وارون سازی را نمایش می‌دهد.



شکل ۴- روش وارون سازی

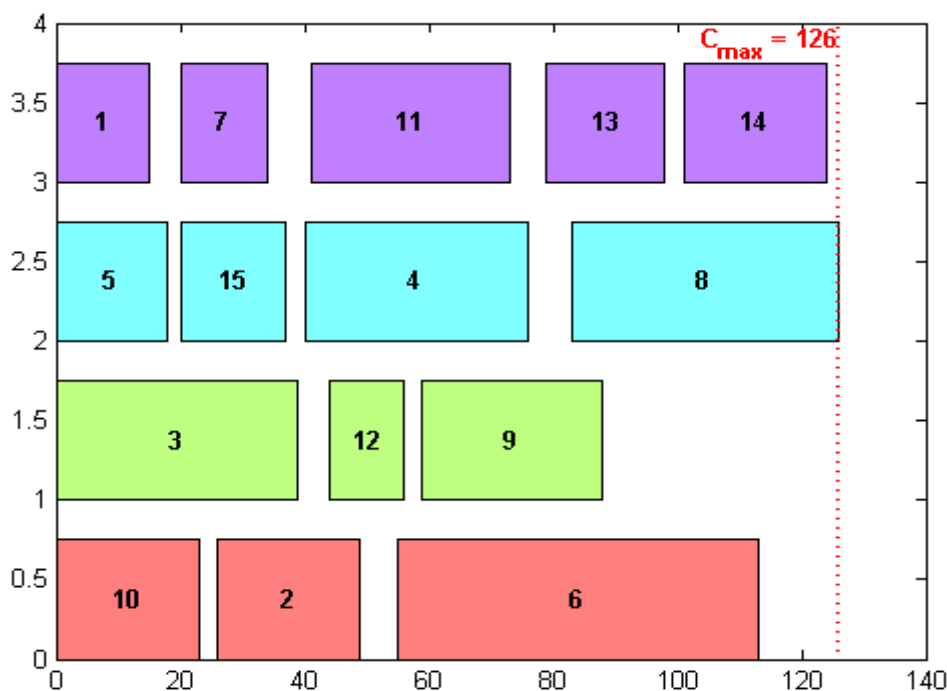
**۴-۴-۲- نتایج استفاده از الگوریتم بهینه سازی علف‌های هرز**

در الگوریتم بهینه‌سازی علف‌های هرز اندازه جمعیت برابر ۱۰۰، حداکثر تعداد نسل‌ها برابر ۲۰۰، مقدار اولیه شعاع پراکندگی برابر ۰٫۱، مقدار نهایی شعاع پراکندگی برابر ۰٫۰۰۱، حداقل تعداد دانه‌ها برابر ۱ و حداکثر تعداد دانه‌ها برابر ۱۰ در نظر گرفته شده است. جدول (۱) موارد مربوط به تعداد وظایف، متوسط زمان پاسخ، متوسط زمان انتظار، متوسط زمان تکمیل، اتلاف وقت و درصد کارایی را در هر پردازنده در حالت ۱۵ وظیفه را نشان می‌دهد.

جدول ۱- نتایج زمانبندی وظایف در الگوریتم بهینه‌سازی علف‌های هرز در حالت ۱۵ وظیفه				
	پردازنده اول	پردازنده دوم	پردازنده سوم	پردازنده چهارم
تعداد وظایف در هر پردازنده	۳	۳	۴	۵
متوسط زمان پاسخ در هر پردازنده	۶۱٫۶۶	۶۱	۶۴٫۲۵	۶۸٫۸
متوسط زمان انتظار در هر پردازنده	۲۷	۳۴٫۳۳	۳۵٫۷۵	۴۸٫۲
زمان تکمیل وظایف در هر پردازنده	۱۱۳	۸۸	۱۲۶	۱۲۴
اتلاف وقت در هر پردازنده	۱۳	۳۸	۰	۲
درصد کارایی هر پردازنده	۸۹٫۶۸	۶۹٫۸۴	۱۰۰	۹۸٫۴۱

شکل (۵) زمانبندی و تخصیص وظایف به پردازنده‌ها در حالت ۱۵ وظیفه با استفاده از الگوریتم بهینه‌سازی علف‌های هرز را نشان می‌دهد. محور افقی زمان و محور عمودی تعداد وظایف تخصیص یافته به هر پردازنده را نشان می‌دهد.



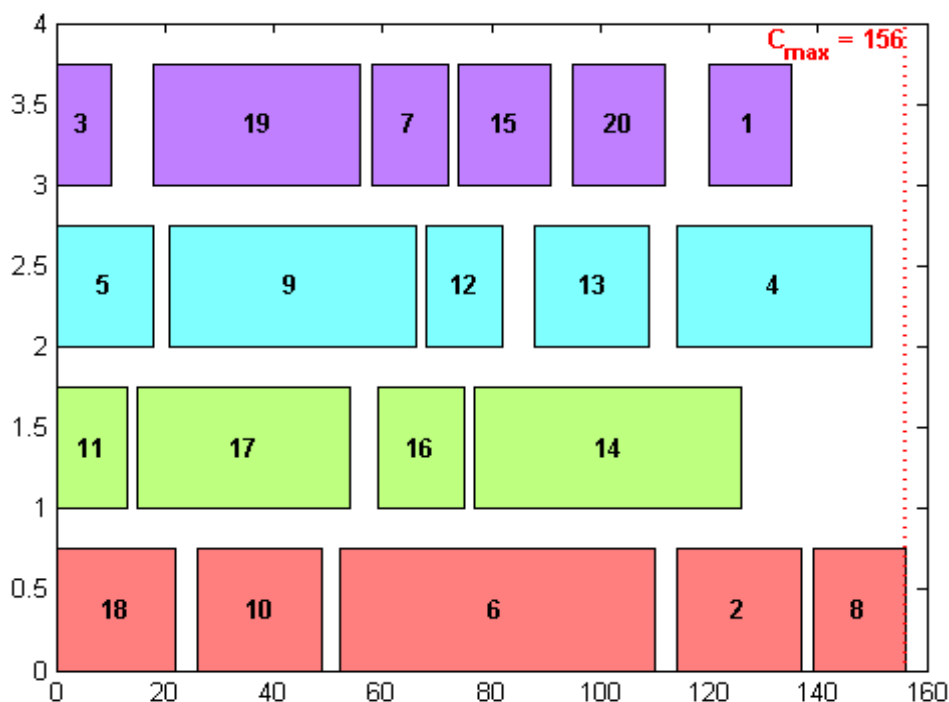


شکل ۵- زمانبندی وظایف با استفاده از الگوریتم بهینه‌سازی علف‌های هرز در حالت ۱۵ وظیفه

جدول (۲) موارد مربوط به تعداد وظایف، متوسط زمان پاسخ، متوسط زمان انتظار، متوسط زمان تکمیل، اتلاف وقت و درصد کارایی را در هر پردازنده در حالت ۲۰ وظیفه را نشان می‌دهد.

جدول ۲- نتایج زمانبندی وظایف در الگوریتم بهینه‌سازی علف‌های هرز در حالت ۲۰ وظیفه				
	پردازنده اول	پردازنده دوم	پردازنده سوم	پردازنده چهارم
تعداد وظایف در هر پردازنده	۵	۴	۵	۶
متوسط زمان پاسخ در هر پردازنده	۹۴,۸	۶۷	۸۵	۷۹,۳۳
متوسط زمان انتظار در هر پردازنده	۶۶,۲	۳۷,۷۵	۵۸,۲	۶۰,۸۳
زمان تکمیل وظایف در هر پردازنده	۱۵۶	۱۲۶	۱۵۰	۱۳۵
اتلاف وقت در هر پردازنده	۰	۳۰	۶	۲۱
درصد کارایی هر پردازنده	۱۰۰	۸۰,۷۶	۹۶,۱۵	۸۶,۵۳

شکل (۶) زمانبندی و تخصیص وظایف به پردازنده‌ها در حالت ۲۰ وظیفه با استفاده از الگوریتم بهینه‌سازی علف‌های هرز را نشان می‌دهد. محور افقی زمان و محور عمودی تعداد وظایف تخصیص‌یافته به هر پردازنده را نشان می‌دهد.

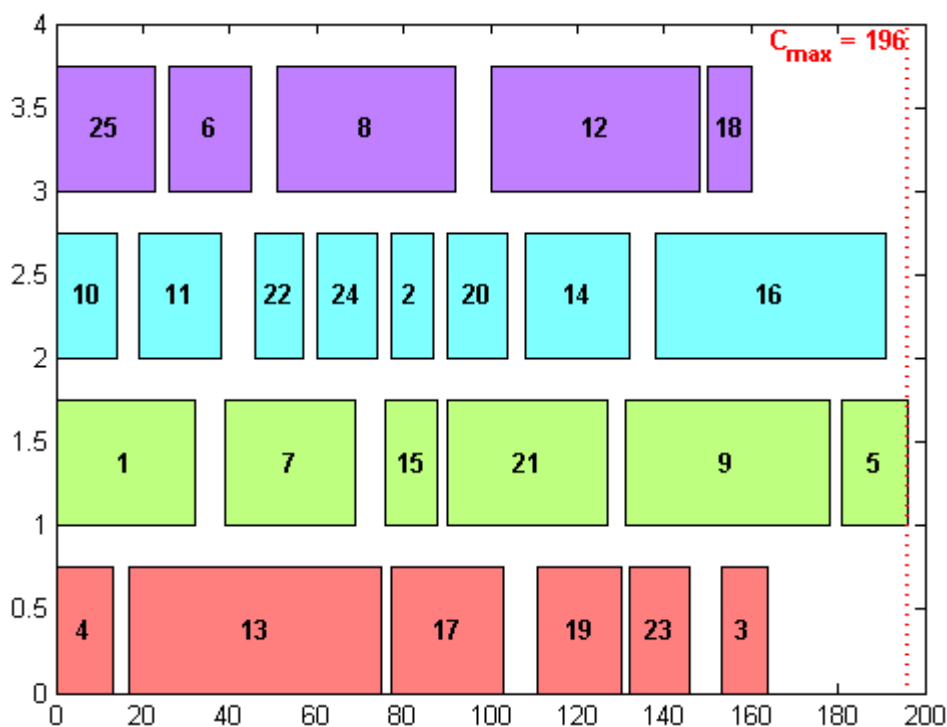


شکل ۶- زمانبندی وظایف با استفاده از الگوریتم بهینه‌سازی علف‌های هرز در حالت ۲۰ وظیفه

جدول (۳) موارد مربوط به تعداد وظایف، متوسط زمان پاسخ، متوسط زمان انتظار، متوسط زمان تکمیل، اتلاف وقت و درصد کارایی را در هر پردازنده در حالت ۲۵ وظیفه را نشان می‌دهد.

	پردازنده اول	پردازنده دوم	پردازنده سوم	پردازنده چهارم
تعداد وظایف در هر پردازنده	۶	۶	۸	۵
متوسط زمان پاسخ در هر پردازنده	۱۰۵,۱۶	۱۱۵	۸۷,۱۲۵	۹۳,۶
متوسط زمان انتظار در هر پردازنده	۸۱,۶۶	۸۶,۱۶	۶۷,۲۵	۶۵,۴
زمان تکمیل وظایف در هر پردازنده	۱۶۴	۱۹۶	۱۹۱	۱۶۰
اتلاف وقت در هر پردازنده	۳۲	۰	۵	۳۶
درصد کارایی هر پردازنده	۸۳,۶۷	۱۰۰	۹۷,۴۴	۸۱,۶۳

شکل (۷) زمانبندی و تخصیص وظایف به پردازنده‌ها در حالت ۲۵ وظیفه با استفاده از الگوریتم بهینه‌سازی علف‌های هرز را نشان می‌دهد. محور افقی زمان و محور عمودی تعداد وظایف تخصیص‌یافته به هر پردازنده را نشان می‌دهد.

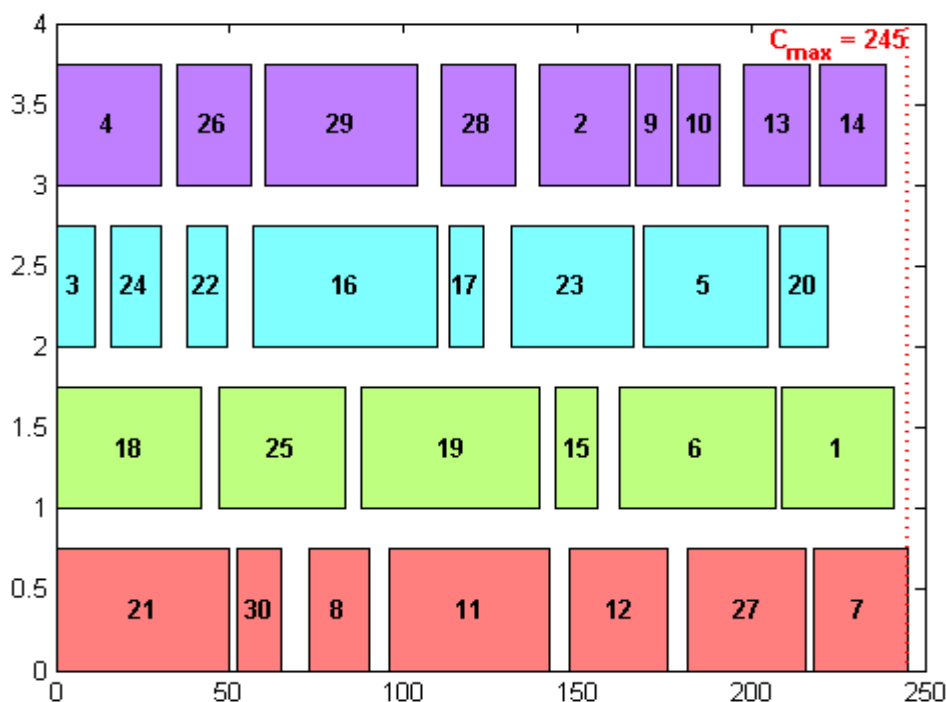


شکل ۷- زمانبندی وظایف با استفاده از الگوریتم بهینه‌سازی علف‌های هرز در حالت ۲۵ وظیفه

جدول (۴) موارد مربوط به تعداد وظایف، متوسط زمان پاسخ، متوسط زمان انتظار، متوسط زمان تکمیل، اتلاف وقت و درصد کارایی را در هر پردازنده در حالت ۳۰ وظیفه را نشان می‌دهد.

جدول ۴- نتایج زمانبندی وظایف در الگوریتم بهینه‌سازی علف‌های هرز در حالت ۳۰ وظیفه				
	پردازنده اول	پردازنده دوم	پردازنده سوم	پردازنده چهارم
تعداد وظایف در هر پردازنده	۷	۶	۸	۹
متوسط زمان پاسخ در هر پردازنده	۱۴۰٫۵۷	۱۴۴٫۶۶	۱۱۴٫۵	۱۴۵٫۶۶
متوسط زمان انتظار در هر پردازنده	۱۰۹٫۸۵	۱۰۸٫۳۳	۹۱٫۵	۱۲۳٫۲۲
زمان تکمیل وظایف در هر پردازنده	۲۴۵	۲۴۱	۲۲۲	۲۳۹
اتلاف وقت در هر پردازنده	۰	۴	۲۳	۶
درصد کارایی هر پردازنده	۱۰۰	۹۸٫۳۶	۹۰٫۶۱	۹۷٫۵۵

شکل (۸) زمانبندی و تخصیص وظایف به پردازنده‌ها در حالت ۳۰ وظیفه با استفاده از الگوریتم بهینه‌سازی علف‌های هرز را نشان می‌دهد. محور افقی زمان و محور عمودی تعداد وظایف تخصیص‌یافته به هر پردازنده را نشان می‌دهد.



شکل ۸- زمانبندی وظایف با استفاده از الگوریتم بهینه‌سازی علف‌های هرز در حالت ۳۰ وظیفه

جدول (۵) زمان تکمیل وظایف، تعداد وظایف، متوسط تعداد وظایف، متوسط زمان پاسخ، متوسط زمان انتظار، متوسط زمان تکمیل وظایف، متوسط زمان اتلاف وقت و متوسط کارایی را به‌ازای ۱۵، ۲۰، ۲۵ و ۳۰ وظیفه برای ۴ پردازنده در الگوریتم بهینه‌سازی علف‌های هرز نشان می‌دهد.

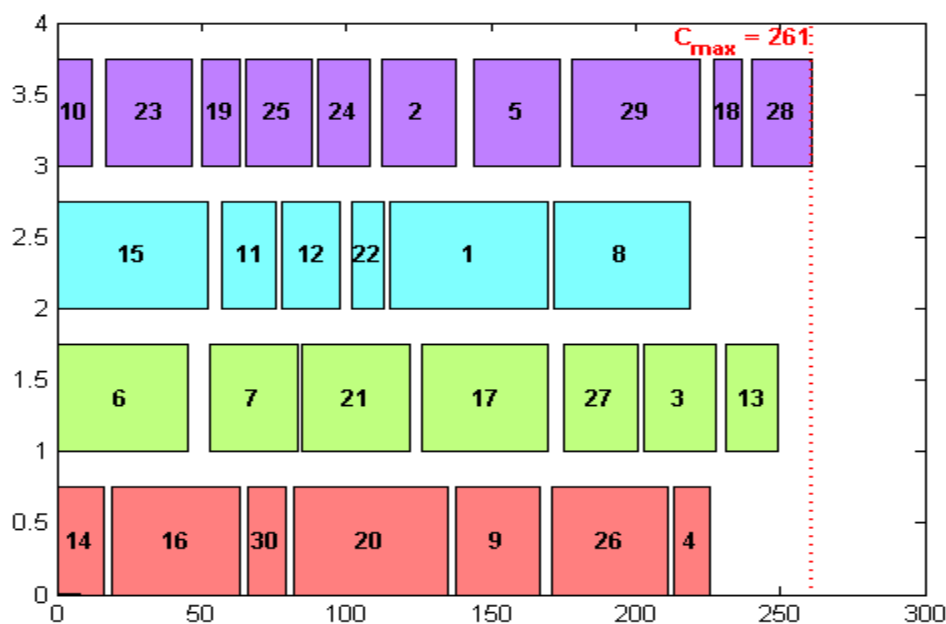
معیار	۱۵	۲۰	۲۵	۳۰
زمان تکمیل وظایف	۱۲۶	۱۵۶	۱۹۶	۲۴۵
تعداد وظایف	۱۵	۲۰	۲۵	۳۰
متوسط تعداد وظایف	۳,۷۵	۵	۶,۲۵	۷,۵
متوسط زمان پاسخ	۶۳,۲۲	۸۱,۵۳	۱۰۰,۲۲	۱۳۶,۳۵
متوسط زمان انتظار	۳۶,۳۲	۵۵,۷۴	۷۵,۱۲	۱۰۸,۲۲
متوسط زمان تکمیل وظایف	۱۱۲,۷۵	۱۴۱,۷۵	۱۷۷,۷۵	۲۳۶,۷۵
متوسط زمان اتلاف	۱۳,۲۵	۱۴,۲۵	۱۸,۲۵	۸,۲۵
متوسط کارایی	۸۹,۴۱	۹۰,۸۶	۹۰,۶۸	۹۶,۶۳

۴-۳- نتایج استفاده از الگوریتم ژنتیک

در الگوریتم ژنتیک اندازه جمعیت برابر ۵۰، حداکثر تعداد نسل‌ها برابر ۵۰، احتمال آمیزش برابر ۰,۷، احتمال جهش برابر ۰,۶ و نرخ جهش برابر ۰,۷ در نظر گرفته شده است. جدول (۶) موارد مربوط به تعداد وظایف، متوسط زمان پاسخ، متوسط زمان انتظار، متوسط زمان تکمیل، اتلاف وقت و درصد کارایی را در هر پردازنده در حالت ۱۵ وظیفه را نشان می‌دهد.

جدول ۶- نتایج زمانبندی وظایف در الگوریتم ژنتیک در حالت ۱۵ وظیفه				
	پردازنده اول	پردازنده دوم	پردازنده سوم	پردازنده چهارم
تعداد وظایف در هر پردازنده	۴	۳	۴	۴
متوسط زمان پاسخ در هر پردازنده	۶۷,۵	۸۵	۶۳,۷۵	۵۶,۵
متوسط زمان انتظار در هر پردازنده	۳۷,۷۵	۴۸,۳۳	۳۹,۲۵	۳۱,۲۵
زمان تکمیل وظایف در هر پردازنده	۱۲۸	۱۲۳	۱۱۲	۱۱۸
اتلاف وقت در هر پردازنده	۰	۵	۱۶	۱۰
درصد کارایی هر پردازنده	۱۰۰	۹۶,۰۹	۸۷,۵	۹۲,۱۸

شکل (۹) زمانبندی و تخصیص وظایف به پردازنده‌ها در حالت ۱۵ وظیفه با استفاده از الگوریتم ژنتیک را نشان می‌دهد. محور افقی زمان و محور عمودی تعداد وظایف تخصیص یافته به هر پردازنده را نشان می‌دهد.

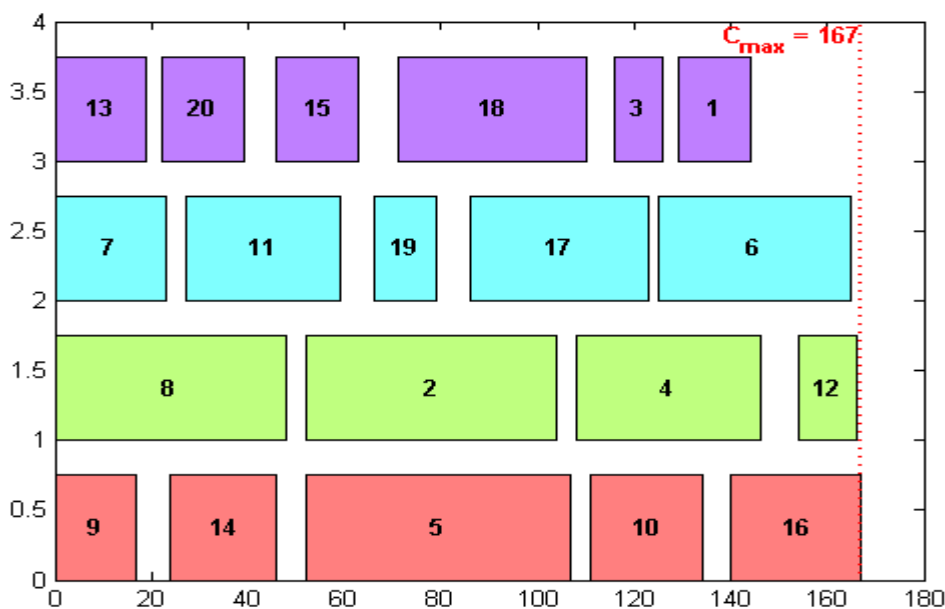


شکل ۹- زمانبندی وظایف با استفاده از الگوریتم ژنتیک در حالت ۱۵ وظیفه

جدول (۷) موارد مربوط به تعداد وظایف، متوسط زمان پاسخ، متوسط زمان انتظار، متوسط زمان تکمیل، اتلاف وقت و درصد کارایی را در هر پردازنده در حالت ۲۰ وظیفه را نشان می‌دهد.

جدول ۷- نتایج زمانبندی وظایف در الگوریتم ژنتیک در حالت ۲۰ وظیفه				
	پردازنده اول	پردازنده دوم	پردازنده سوم	پردازنده چهارم
تعداد وظایف در هر پردازنده	۵	۴	۵	۶
متوسط زمان پاسخ در هر پردازنده	۹۴,۲	۱۱۶	۸۹,۸	۸۳,۵
متوسط زمان انتظار در هر پردازنده	۶۵,۴	۷۸,۵	۶۰,۸	۶۴
زمان تکمیل وظایف در هر پردازنده	۱۶۷	۱۶۶	۱۶۵	۱۴۴
اتلاف وقت در هر پردازنده	۰	۱	۲	۲۳
درصد کارایی هر پردازنده	۱۰۰	۹۹,۴	۹۸,۸	۸۶,۲۲

شکل (۱۰) زمانبندی و تخصیص وظایف به پردازنده‌ها در حالت ۲۰ وظیفه با استفاده از الگوریتم ژنتیک را نشان می‌دهد. محور افقی زمان و محور عمودی تعداد وظایف تخصیص یافته به هر پردازنده را نشان می‌دهد.

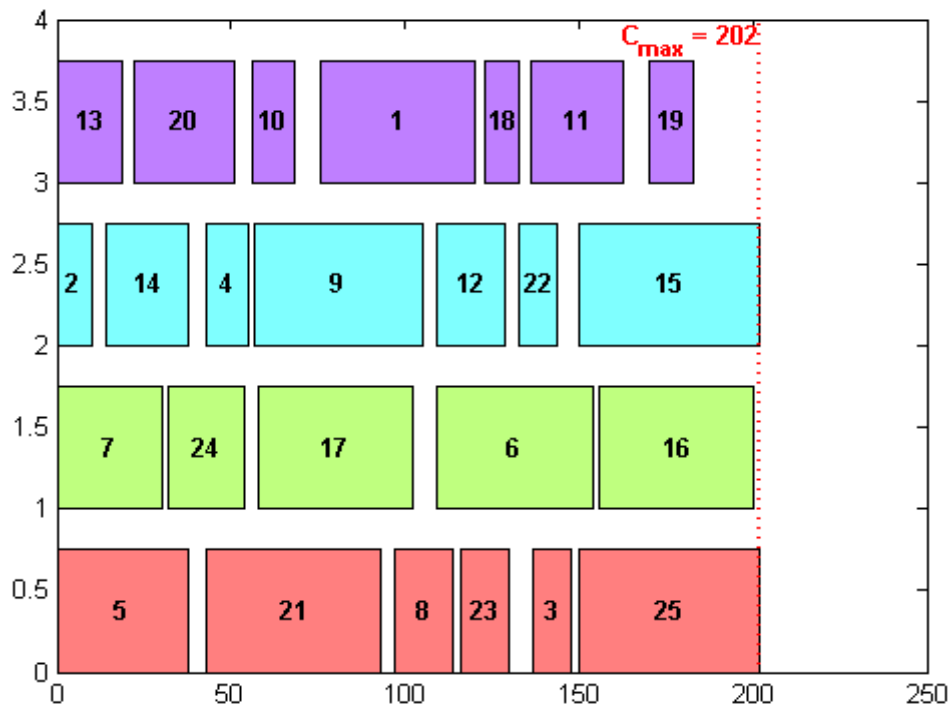


شکل ۱۰- زمانبندی وظایف با استفاده از الگوریتم ژنتیک در حالت ۲۰ وظیفه

جدول (۸) موارد مربوط به تعداد وظایف، متوسط زمان پاسخ، متوسط زمان انتظار، متوسط زمان تکمیل، اتلاف وقت و درصد کارایی را در هر پردازنده در حالت ۲۵ وظیفه را نشان می‌دهد.

جدول ۸- نتایج زمانبندی وظایف در الگوریتم ژنتیک در حالت ۲۵ وظیفه				
	پردازنده اول	پردازنده دوم	پردازنده سوم	پردازنده چهارم
تعداد وظایف در هر پردازنده	۶	۵	۷	۷
متوسط زمان پاسخ در هر پردازنده	۱۲۰,۸۳	۱۰۸	۹۷,۵۷	۱۰۵,۸۸
متوسط زمان انتظار در هر پردازنده	۹۰,۵	۷۱	۷۲,۲۸	۸۳,۸۸
زمان تکمیل وظایف در هر پردازنده	۲۰۲	۲۰۰	۲۰۲	۱۲۳
اتلاف وقت در هر پردازنده	۰	۲	۰	۱۹
درصد کارایی هر پردازنده	۱۰۰	۹۹	۱۰۰	۹۰,۵۹

شکل (۱۱) زمانبندی و تخصیص وظایف به پردازنده‌ها در حالت ۲۵ وظیفه با استفاده از الگوریتم ژنتیک را نشان می‌دهد. محور افقی زمان و محور عمودی تعداد وظایف تخصیص یافته به هر پردازنده را نشان می‌دهد.



شکل ۱۱- زمانبندی وظایف با استفاده از الگوریتم ژنتیک در حالت ۲۵ وظیفه

جدول (۹) موارد مربوط به تعداد وظایف، متوسط زمان پاسخ، متوسط زمان انتظار، متوسط زمان تکمیل، اتلاف وقت و درصد کارایی را در هر پردازنده در حالت ۳۰ وظیفه را نشان می‌دهد.

جدول ۹- نتایج زمانبندی وظایف در الگوریتم ژنتیک در حالت ۳۰ وظیفه				
	پردازنده اول	پردازنده دوم	پردازنده سوم	پردازنده چهارم
تعداد وظایف در هر پردازنده	۷	۷	۷	۹
متوسط زمان پاسخ در هر پردازنده	۱۴۱,۲۸	۱۲۵,۸۵	۱۱۸,۸۵	۱۳۱,۳۳
متوسط زمان انتظار در هر پردازنده	۱۱۱,۵۷	۹۳,۱۴	۹۲,۵۷	۱۰۶,۶۶
زمان تکمیل وظایف در هر پردازنده	۲۴۰	۲۵۳	۲۱۴	۲۵۸
اتلاف وقت در هر پردازنده	۱۸	۵	۴۴	۰
درصد کارایی هر پردازنده	۹۳,۰۲	۹۸,۰۶	۸۲,۹۴۵	۱۰۰

شکل (۱۲) زمانبندی و تخصیص وظایف به پردازنده‌ها در حالت ۳۰ وظیفه با استفاده از الگوریتم ژنتیک را نشان می‌دهد. محور افقی زمان و محور عمودی تعداد وظایف تخصیص یافته به هر پردازنده را نشان می‌دهد.



شکل ۱۲- زمانبندی وظایف با استفاده از الگوریتم ژنتیک در حالت ۳۰ وظیفه

جدول (۱۰) زمان تکمیل وظایف، تعداد وظایف، متوسط تعداد وظایف، متوسط زمان پاسخ، متوسط زمان انتظار، متوسط زمان تکمیل وظایف، متوسط زمان اتلاف وقت و متوسط کارایی را به‌ازای ۱۵، ۲۰، ۲۵ و ۳۰ وظیفه برای ۴ پردازنده در الگوریتم ژنتیک نشان می‌دهد.



جدول ۱۰- نتایج استفاده از الگوریتم ژنتیک با تعداد ۴ پردازنده

زمان تکمیل وظایف	۱۲۸	۱۶۷	۲۰۲	۲۵۸
تعداد وظایف	۱۵	۲۰	۲۵	۳۰
متوسط تعداد وظایف	۳,۷۵	۵	۶,۲۵	۷,۵
متوسط زمان پاسخ	۶۸,۱۸	۹۵,۸۷	۱۰۷,۹۲	۱۲۹,۳۳
متوسط زمان انتظار	۳۹,۱۴	۶۷,۱۷	۷۹,۲۶	۱۰۰,۹۸
متوسط زمان تکمیل وظایف	۱۲۰,۲۵	۱۶۰,۵	۱۹۶,۷۵	۲۴۱,۲۵
متوسط زمان اتلاف	۷,۷۵	۶,۵	۵,۲۵	۱۶,۷۵
متوسط کارایی	۹۳,۹۴	۹۶,۱	۹۷,۴	۹۳,۵

### بحث و نتیجه گیری

این مقاله مبتنی بر زمانبندی وظایف در محیط ابر با استفاده از الگوریتم‌های تکاملی است. در بخش دوم ادبیات تحقیق مرور شد و در بخش سوم الگوریتم‌های بهینه‌سازی علف‌های هرز و ژنتیک بررسی شدند. در بخش چهارم به شبیه‌سازی و پیاده‌سازی مسأله پرداخته شد و نتایج کمی و کیفی استفاده از الگوریتم‌های بهینه‌سازی علف‌های هرز و ژنتیک در زمانبندی وظایف در محیط رایانش ابری نمایش داده شد. با توجه به جداول (۱)، (۲)، (۳)، (۴) و (۵) الگوریتم بهینه‌سازی علف‌های هرز با زمان تکمیل وظایف ۱۲۶، ۱۵۶، ۱۹۶ و ۲۴۵، متوسط زمان پاسخ ۶۳،۹۲، ۸۱،۵۳، ۱۰۰،۲۲ و ۱۰۸،۲۲، متوسط زمان اتلاف ۱۳،۲۵، ۱۴،۲۵، ۱۸،۲۵ و ۸،۲۵ و متوسط درصد کارایی ۸۹،۴۱، ۹۰،۸۶، ۹۰،۶۸ و ۹۶،۶۳ به ترتیب به‌ازای ۱۵، ۲۰، ۲۵ و ۳۰ وظیفه زمانبندی وظایف در محیط ابر را انجام می‌دهد در حالی که با توجه به جداول (۶)، (۷)، (۸)، (۹) و (۱۰) الگوریتم ژنتیک با زمان تکمیل وظایف ۱۲۸، ۱۶۷، ۲۰۲ و ۲۵۸، متوسط زمان پاسخ ۶۸،۱۸، ۹۵،۸۷، ۱۰۷،۹۲ و ۱۲۹،۳۳، متوسط زمان اتلاف ۷،۷۵، ۶،۵، ۵،۲۵ و ۱۶،۷۵ و متوسط درصد کارایی ۹۳،۹۴، ۹۶،۱، ۹۷،۴ و ۹۳،۵ به ترتیب به‌ازای ۱۵، ۲۰، ۲۵ و ۳۰ وظیفه زمانبندی وظایف در محیط ابر را انجام می‌دهد. بنابراین می‌توان نتیجه گرفت الگوریتم بهینه‌سازی علف‌های هرز عملکرد بهتری در مقایسه با الگوریتم ژنتیک دارد و به عنوان مدل برتر انتخاب می‌شود.

### منابع

- Wu, Zhangjun, et al. (2010), "A revised discrete particle swarm optimization for cloud workflow scheduling." Computational Intelligence and Security (CIS), 2010 International Conference on. IEEE.
- Kumar, P. and A. Verma, (2012), "Independent task scheduling in cloud computing by improved genetic algorithm". International Journal of Advanced Research in Computer Science and Software Engineering.
- Yu, Jia, and Rajkumar Buyya. (2006), "A budget constrained scheduling of workflow applications on utility grids using genetic algorithms." Workflows in Support of Large-Scale Science, 2006. WORKS'06. Workshop on. IEEE.
- Rahman, Mustafizur, Srikumar Venugopal, and Rajkumar Buyya. (2007), "A dynamic critical path algorithm for scheduling scientific workflow applications on global grids." e-Science and Grid Computing, IEEE International Conference on. IEEE.
- Chen, Wei-Neng, and Jun Zhang. (2009), "An ant colony optimization approach to a grid workflow scheduling problem with various QoS requirements.", Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on 39.1: 29-43.



- Pandey, Suraj, et al. (2010), "A particle swarm optimization-based heuristic for scheduling workflow applications in cloud computing environments." Advanced information networking and applications (AINA), 2010 24th IEEE international conference on. IEEE.
- Mao, Ming, and Marty Humphrey. (2011), "Auto-scaling to minimize cost and meet application deadlines in cloud workflows." Proceedings of 2011 International Conference for High Performance Computing, Networking, Storage and Analysis. ACM.
- Malawski, Maciej, et al. (2012), "Cost-and deadline-constrained provisioning for scientific workflow ensembles in iaas clouds." Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis. IEEE Computer Society Press.
- Abrishami, Saeid, Mahmoud Naghibzadeh, and Dick HJ Epema. (2013), "Deadline-constrained workflow scheduling algorithms for Infrastructure as a Service Clouds." Future Generation Computer Systems 29.1 (2013): 158-169.
- Rodriguez, Maria Alejandra, and Rajkumar Buyya. (2014), "Deadline based resource provisioning and scheduling algorithm for scientific workflows on clouds." Cloud Computing, IEEE Transactions on 2.2 : 222-235.
- Shilane, D. Martikainen, J. S and Dudoit, S. (2008), "A general frame-work for statistical performance comparison of evolutionary computation algorithms", Information Sciences, Journal, 178, 2870-2879.
- Mehrabian, Ali Reza, and Caro Lucas. (2006), "A novel numerical optimization algorithm inspired from weed colonization." Ecological informatics 1.4: 355-366.
- Goldberg, D.E. (2012). Genetic Algorithms in Search, Optimization and Machine Learning, Addison-Wesley.