

استفاده از الگوریتم‌های تکاملی در زمانبندی وظایف در محیط محاسبات ابری

پریا اردستانی

استاد گروه کامپیوتر، واحد تهران شرق، دانشگاه آزاد اسلامی، تهران، ایران

Pariya.ardestani@gmail.com

خلاصه

پردازش ابر را می‌توان حوزه‌ای جدید برای محاسبات توزیع‌شده و پردازش موازی در مقیاس بزرگ تعریف کرد. پردازش ابر، پردازش را به عنوان سرویس اینترنتی و به‌ازای میزان استفاده از آن در نظر می‌گیرد. کارایی و عملکرد سرویس‌های پردازش ابر همیشه وابسته به عملکرد وظایف کاربری است که وارد سیستم ابر شده است. زمانبندی وظایف کاربر نقش مهمی در بهبود عملکرد سرویس‌های ابری بازی می‌کند زمانبندی وظایف یکی از روش‌های زمانبندی رایج است. 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 25 و 30 وظیفه بر روی 4 پردازنده انجام می‌شود و نتایج آن با الگوریتم ژنتیک مقایسه می‌شود. نتایج حاکی از عملکرد بهینه الگوریتم بهینه‌سازی علف‌های هرز با زمان تکمیل وظایف 196 و 245 به ترتیب براساس 25 و 30 وظیفه در برابر الگوریتم ژنتیک است.

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

1- 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 [1]. 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

Corresponding author: استاد دانشگاه، گروه کامپیوتر، نرم افزار

Email: pariya.ardestani@gmail.com

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 [2]. لذا در این مقاله روش جدیدی از زمانبندی وظایف با استفاده از الگوریتم بهینه سازی علف های هرز ارائه می‌شود. ادامه ساختار این مقاله به این صورت است که در بخش دوم مروری بر پیشینه 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.

2- مروری بر پیشینه تحقیق

یو و همکاران در سال 2006 به زمانبندی با محدودیت منابع برنامه‌های بروی 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 [3]. رحمان و همکاران در سال 2007 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 نگاشتی 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 [4]. چن و همکاران در سال 2009 با الگوریتم مورچگان به مسأله زمانبندی وظایف در ابر با توجه به نیازهای افزایش کیفیت سرویس پرداختند. باتوجه 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 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 7 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 [5]. پاندی و همکاران در سال 2010 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 3 برابر نسبت به الگوریتم BRS دارد [6]. مائو و همکاران در سال 2011 به خود مقایسه‌سنجی برای کمینه‌سازی 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 بروی

ماشین‌های مجازی و زمانبندی کارها به‌طور پویا انجام شد. این روش با 4 روش دیگر در زمینه محاسبات ابری مورد مقایسه قرار گرفته که نشان از صرفه‌جویی در هزینه از 9.8٪ تا 40.4٪ در مقایسه با دیگر روش‌ها دارد [7]. مالوسکی و همکاران در سال 2012 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 [8]. ابریشمی و همکاران در سال 2013 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 2 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 [9]. رودریگز و همکاران در سال 2014 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 کمینه‌سازی 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 [10].

3- 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 بهینه‌سازی و کمینه‌سازی 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 [11].

3-1 الگوریتم بهینه‌سازی علف‌های هرز

الگوریتم بهینه‌سازی علف‌های هرز یک الگوریتم بهینه‌سازی عددی الهام گرفته شده از رشد علف‌های هرز است. این الگوریتم در سال 2006 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 [12].

مقداردهی اولیه جمعیت: توزیع یک جمعیت اولیه در فضای d بُعدی به صورت تصادفی یک راه حل اولیه است. **تولیدمثل:** هر یک از اعضای جمعیت علفها مجاز است بسته به توان و شرایط خود تولیدمثل کند. هر علف می تواند به طور خطی حداقل و یا حداکثر تعدادی دانه تولید کند. به عبارت دیگر، دانه های یک علف بسته به شرایط آن به طور خطی مورد تولید حداکثر و حداقل قرار می گیرند [2].

تصادفی بودن و سازگاری در الگوریتم در این بخش ارائه شده است. به طور تصادفی با توزیع نرمال و با میانگین صفر، توزیع دانه های تولیدشده در فضای جستجو d بُعدی شکل می گیرد، اما واریانس متفاوت است. این به این معنی است که دانه های تولیدشده به طور تصادفی نزدیک والد خود هستند. با این حال، انحراف معیار (σ) ، تابع تصادفی خواهد بود و از مقدار اولیه قبلی تعریف شده اش کاهش می یابد و در هر مرحله (نسل) σ_{init} به ارزش نهایی σ_{final} تبدیل می شود. در شبیه سازی، تغییر غیرخطی نشان داده می شود که در رابطه (1) بیان شده است [12].

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

که $iter_{max}$ بیشترین مقدار تکرار و σ_{iter} انحراف معیار در هر مرحله و n شاخص مدولاسیون غیرخطی است. این تغییر تضمین می کند که احتمال از بین رفتن دانه در یک منطقه دور غیرخطی کاهش می یابد [12].

حذف رقابتی: اگر یک علف حذف شود دانه های 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 می شود و انتظار می رود 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. دانه های تولید شده مجاز به گسترش در ناحیه جستجو می باشند، که در پراکندگی محیط گفته شد. زمانی که همه ی دانه ها موقعیت خود را در منطقه ی جستجو پیدا کردند، آن ها با والد خود رتبه بندی می شوند (به عنوان یک کلونی از علفهای هرز). سپس، برای رسیدن به حداکثر جمعیت مجاز در یک کلونی، علفهای هرز با شایستگی کمتر حذف می شوند. در این روش، علفها و دانه ها با هم رتبه بندی شده و آنهایی که شایستگی بهتری دارند، مجاز به تکثیر می شوند. این مکانیسم یک فرصت به علفها با شایستگی بهتر می دهد و اگر دانه های آن ها دارای شایستگی مناسب باشند می توانند زنده بمانند. این مکانیسم خروج رقابتی را تحقق می دهد [12].

الگوریتم IWO فرصت آزمایش با استراتژی های مختلف کشف فضاهای جستجو را فراهم می کند. مراحل زیر الگوریتم را با استفاده از کنوانسیون اصطلاحات سازگار و با الهام از ایده های «طبیعی» بیان می کند.

- 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 روش های پراکنده سازی، 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 [12].

3-2- 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 به صورت امروزی آن نخستین بار توسط جان هالند متخصص علوم کامپیوتر دانشگاه میشیگان در سال ۱۹۷۵ پیشنهاد گردید. یک کاربرد متداول الگوریتم ژنتیک، استفاده از آن به عنوان 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 به وجود می‌آیند. در الگوریتم ژنتیک هر فرد از جمعیت به صورت 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 [13]. به طور 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 [13].

انتخاب والدین: انتخاب والدین می‌تواند به سه صورت انجام شود. روش اول، والدین به صورت 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 به صورت تصادفی انتخاب می‌شوند و سپس عضوی از اعضای انتخاب شده که بهترین مقدار تابع هدف را داشته باشد به عنوان برنده رقابت انتخاب می‌شود [13].

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 تک-نقطه‌ای، 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 [13].

جهش: این عملگر یک یا چند ژن از یک کروموزوم را تغییر می‌دهد [13].

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 قرار گرفتن 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 [13].

4- 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_{i1,i2,j}: زمان آماده‌سازی وظیفه i_2 بعد از اتمام اجرای وظیفه i_1 بر روی پردازنده j ام

4-1- قیدها در فرآیند شبیه‌سازی

همه وظایف بایستی اجرا شوند.

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

(2) هر وظیفه فقط یک‌بار اجرا می‌شود.

(3) زمان شروع وظیفه بعدی توسط یک پردازنده برابر است با زمان اتمام وظیفه قبلی به علاوه زمان آماده‌سازی وظیفه بعدی (طبق رابطه (2))

$$\text{Start Time(Next Task)} = \text{Finish Time(Previous Task)} + \text{Setup Time(Next Task)} \quad (2)$$

(4) زمان اتمام وظیفه فعلی برابر است با زمان شروع وظیفه فعلی به علاوه زمان اجرای وظیفه فعلی (طبق رابطه (3))

$$\text{Finish Time(Current Task)} = \text{Start Time(Current Task)} + \text{Process Time(Current Task)} \quad (3)$$

4-2- تعریف متغیر تصمیم در شبیه‌سازی

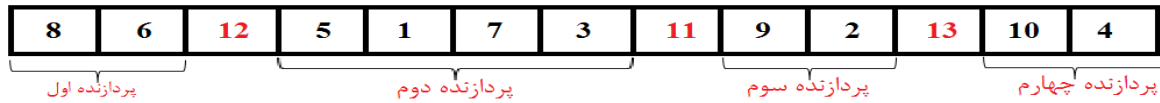
برای شبیه‌سازی متغیرهای تصمیم در مسأله از کدگذاری گسسته استفاده شده است و ساختار متغیر تصمیم شامل I+J-1 بعد است. با فرض اینکه تعداد وظایف برابر 10 و تعداد پردازنده‌ها برابر 4 باشد شکل (1) نحوه کدگذاری متغیر تصمیم را نشان می‌دهد.

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

شکل 1- شکل متغیر تصمیم

در شکل (1) 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 جدادکننده وظایفی که مشاهده شوند به ترتیب به پردازنده‌ها تخصیص می‌یابند.

شکل (2) نحوه تخصیص وظایف به پردازنده‌ها را نشان می‌دهد.



شکل 2- نحوه تخصیص وظایف به پردازنده‌ها

3-4- 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 (4) 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.

4-4- 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 4 و تعداد وظایف 15، 20، 25 و 30 در نظر گرفته شده است.

1-4-4- 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.

4-1-1-1- روش تعویض

در این روش دو بعد یا دو بلوک از پاسخ به طور تصادفی انتخاب شده و موقعیت آن‌ها تعویض می‌شود. شکل (3) مثالی از روش تعویض را نمایش می‌دهد.



شکل 3- روش تعویض

4-1-2- روش وارون‌سازی

دو بعد از پاسخ به طور تصادفی انتخاب شده و ترتیب آن‌ها در بین این دو بعد معکوس می‌شود. شکل (4) مثالی از روش وارون‌سازی را نمایش می‌دهد.



شکل 4- روش وارون‌سازی

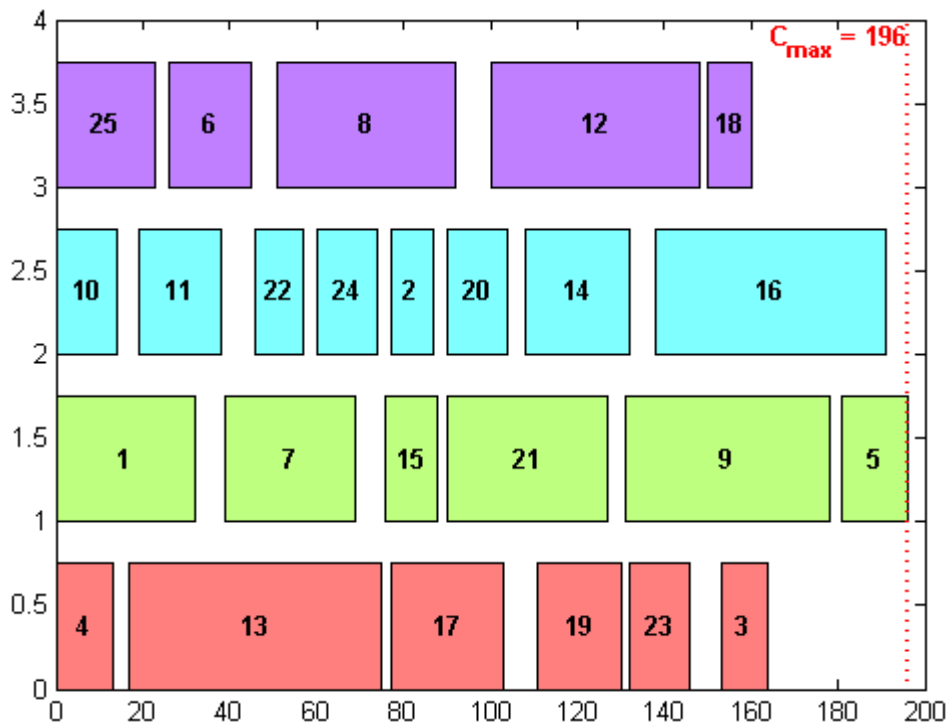
4-2-4- نتایج استفاده از الگوریتم بهینه‌سازی علف‌های هرز

در الگوریتم بهینه‌سازی علف‌های هرز اندازه جمعیت برابر 100، حداکثر تعداد نسل‌ها برابر 200، مقدار اولیه شعاع پراکندگی برابر 0.1، مقدار نهایی شعاع پراکندگی برابر 0.001، حداقل تعداد دانه‌ها برابر 1 و حداکثر تعداد دانه‌ها برابر 10 در نظر گرفته شده است.

جدول (1) 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 25 وظیفه را نشان می‌دهد.

جدول 1- نتایج زمانبندی وظایف در الگوریتم بهینه‌سازی علف‌های هرز در حالت 25 وظیفه				
	پردازنده اول	پردازنده دوم	پردازنده سوم	پردازنده چهارم
تعداد وظایف در هر پردازنده	6	6	8	5
متوسط زمان پاسخ در هر پردازنده	105.16	115	87.125	93.6
متوسط زمان انتظار در هر پردازنده	81.66	86.16	67.25	65.4
زمان تکمیل وظایف در هر پردازنده	164	196	191	160
اتلاف وقت در هر پردازنده	32	0	5	36
درصد کارایی هر پردازنده	83.67	100	97.44	81.63

شکل (5) زمانبندی و تخصیص وظایف به پردازنده‌ها در حالت 25 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.

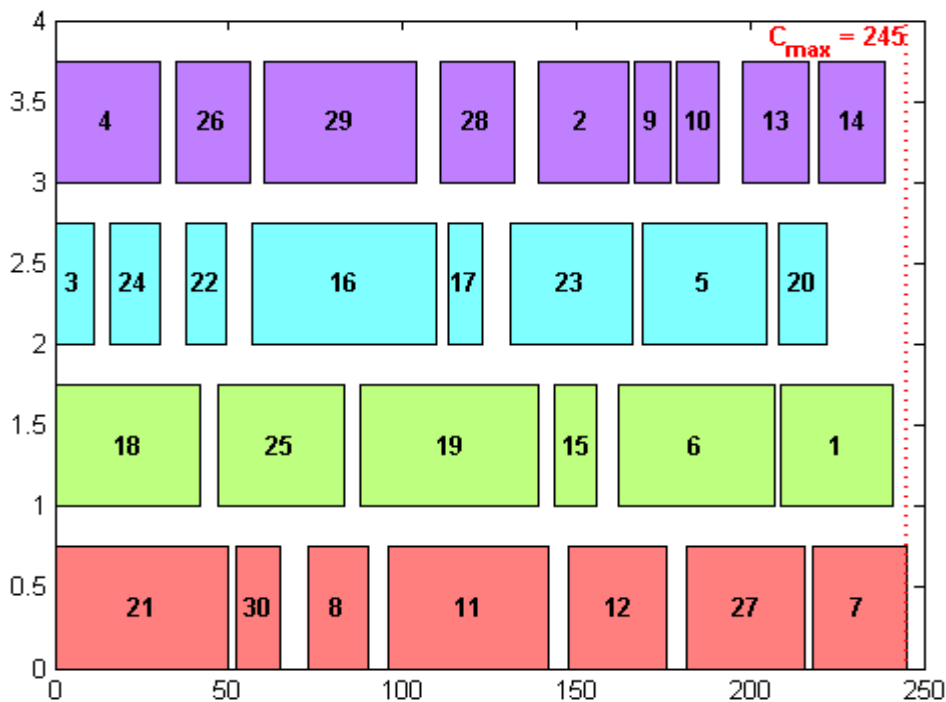


شکل 5- زمانبندی وظایف با استفاده از الگوریتم بهینه‌سازی علف‌های هرز در حالت 25 وظیفه

جدول (2) 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 30 وظیفه را نشان می‌دهد.

جدول 2- نتایج زمانبندی وظایف در الگوریتم بهینه‌سازی علف‌های هرز در حالت 30 وظیفه				
	پردازنده اول	پردازنده دوم	پردازنده سوم	پردازنده چهارم
تعداد وظایف در هر پردازنده	7	6	8	9
متوسط زمان پاسخ در هر پردازنده	140.57	144.66	114.5	145.66
متوسط زمان انتظار در هر پردازنده	109.85	108.33	91.5	123.22
زمان تکمیل وظایف در هر پردازنده	245	241	222	239
اتلاف وقت در هر پردازنده	0	4	23	6
درصد کارایی هر پردازنده	100	98.36	90.61	97.55

شکل (6) زمانبندی و تخصیص وظایف به پردازنده‌ها در حالت 30 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 6- زمانبندی وظایف با استفاده از الگوریتم بهینه‌سازی علف‌های هرز در حالت 30 وظیفه

جدول (3) 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 25 و 30 وظیفه برای 4 پردازنده در الگوریتم بهینه‌سازی علف‌های هرز نشان می‌دهد.

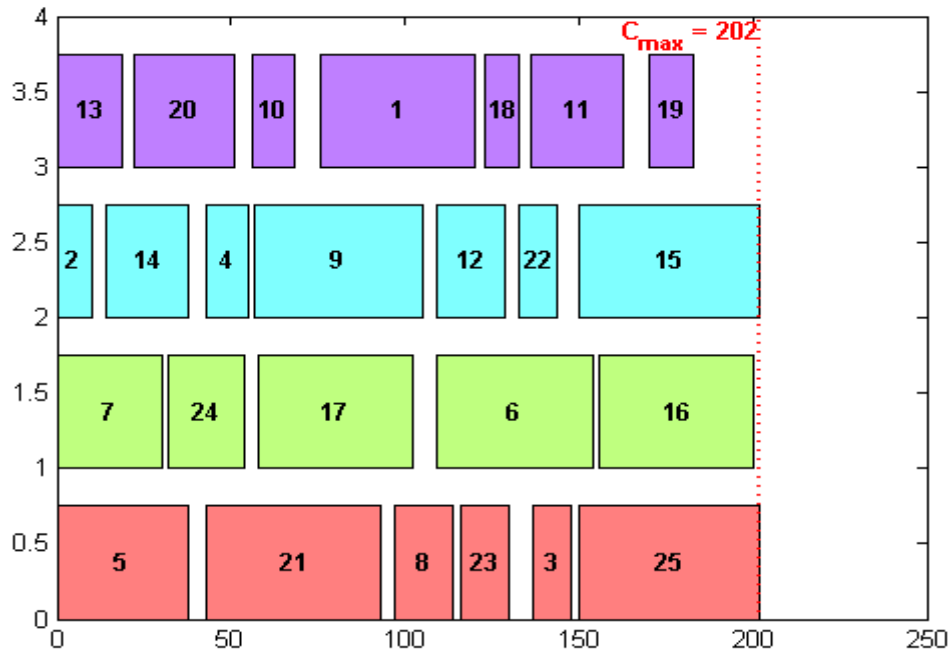
جدول 3- نتایج استفاده از الگوریتم بهینه‌سازی علف‌های هرز با تعداد 4 پردازنده		
زمان تکمیل وظایف	196	245
تعداد وظایف	25	30
متوسط تعداد وظایف	6.25	7.5
متوسط زمان پاسخ	100.22	136.35
متوسط زمان انتظار	75.12	108.22
متوسط زمان تکمیل وظایف	177.75	236.75
متوسط زمان اتلاف	18.25	8.25
متوسط کارایی	90.68	96.63

4-4-3- نتایج استفاده از الگوریتم ژنتیک

در الگوریتم ژنتیک اندازه جمعیت برابر 50، حداکثر تعداد نسل‌ها برابر 50، احتمال آمیزش برابر 0.7، احتمال جهش برابر 0.6 و نرخ جهش برابر 0.7 در نظر گرفته شده است. جدول (4) 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 25 وظیفه را نشان می‌دهد.

جدول 4- نتایج زمانبندی وظایف در الگوریتم ژنتیک در حالت 25 وظیفه				
	پردازنده اول	پردازنده دوم	پردازنده سوم	پردازنده چهارم
تعداد وظایف در هر پردازنده	6	5	7	7
متوسط زمان پاسخ در هر پردازنده	120.83	108	97.57	105.88
متوسط زمان انتظار در هر پردازنده	90.5	71	72.28	83.88
زمان تکمیل وظایف در هر پردازنده	202	200	202	123
اتلاف وقت در هر پردازنده	0	2	0	19
درصد کارایی هر پردازنده	100	99	100	90.59

شکل (7) زمانبندی و تخصیص وظایف به پردازنده‌ها در حالت 25 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.

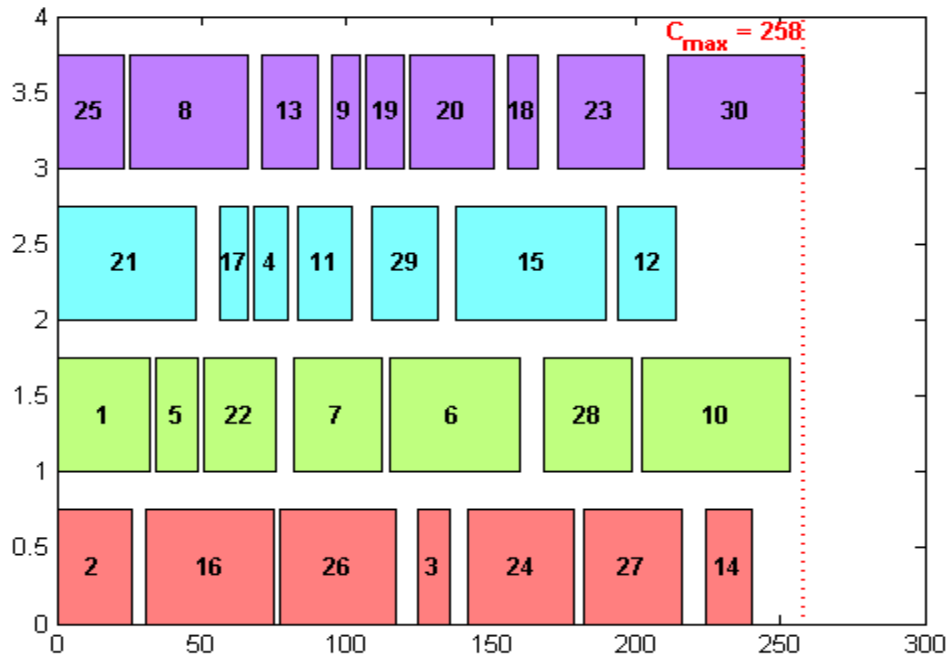


شکل 7- زمانبندی وظایف با استفاده از الگوریتم ژنتیک در حالت 25 وظیفه

جدول (5) 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 30 وظیفه را نشان می‌دهد.

جدول 5- نتایج زمانبندی وظایف در الگوریتم ژنتیک در حالت 30 وظیفه				
	پردازنده اول	پردازنده دوم	پردازنده سوم	پردازنده چهارم
تعداد وظایف در هر پردازنده	7	7	7	9
متوسط زمان پاسخ در هر پردازنده	141.28	125.85	118.85	131.33
متوسط زمان انتظار در هر پردازنده	111.57	93.14	92.57	106.66
زمان تکمیل وظایف در هر پردازنده	240	253	214	258
اتلاف وقت در هر پردازنده	18	5	44	0
درصد کارایی هر پردازنده	93.02	98.06	82.945	100

شکل (8) زمانبندی و تخصیص وظایف به پردازنده‌ها در حالت 30 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 8- زمانبندی وظایف با استفاده از الگوریتم ژنتیک در حالت 30 وظیفه

جدول (6) 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 25 و 30 وظیفه برای 4 پردازنده در الگوریتم ژنتیک نشان می‌دهد.

معیار	نتیجه (25 وظیفه)	نتیجه (30 وظیفه)
زمان تکمیل وظایف	202	258
تعداد وظایف	25	30
متوسط تعداد وظایف	6.25	7.5
متوسط زمان پاسخ	107.92	129.33
متوسط زمان انتظار	79.26	100.98
متوسط زمان تکمیل وظایف	196.75	241.25
متوسط زمان اتلاف	5.25	16.75
متوسط کارایی	97.4	93.5

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 (1)، (2) و (3) الگوریتم بهینه‌سازی علف‌های هرز با زمان تکمیل وظایف 196 و 245، متوسط زمان پاسخ 100.22 و 108.22، متوسط زمان اتلاف 18.25 و 8.25 و متوسط درصد کارایی 90.68 و 96.63 به ترتیب به‌ازای 25 و 30 وظیفه زمانبندی وظایف در محیط ابر را انجام می‌دهد در حالی که با توجه به جداول (4)، (5) و (6) الگوریتم ژنتیک با زمان تکمیل وظایف 202 و 258، متوسط زمان پاسخ 107.92 و 129.33، متوسط زمان اتلاف 5.25 و 16.75 و متوسط درصد کارایی 97.4 و 93.5 به ترتیب به‌ازای 25 و 30 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

1. 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.
2. 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.
3. 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.
4. 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.
5. 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.
6. 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.
7. 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.
8. 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.
9. 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.
10. 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.
11. 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.
12. Mehrabian, Ali Reza, and Caro Lucas. (2006), "A novel numerical optimization algorithm inspired from weed colonization." Ecological informatics 1.4: 355-366.



13. Goldberg, D.E. (2012). Genetic Algorithms in Search, Optimization and Machine Learning, Addison-Wesley.