

## بهینه سازی چند هدفه‌ی مسئله زمان بندی کار در پردازش ابری

ابوذر زندوکیلی<sup>۱</sup>، نجمه منصوری<sup>۲\*</sup>، محمدمسعود جاویدی<sup>۳</sup>

\*نویسنده مسئول، دریافت: ۹۹/۰۷/۰۶، بازنگری: ۹۹/۱۰/۱۰، پذیرش: ۹۹/۱۲/۱۵

<sup>۱</sup> دانشجوی کارشناسی ارشد، علوم کامپیوتر، دانشکده ریاضی و کامپیوتر، دانشگاه شهید باهنر، کرمان، ایران

<sup>۲</sup> استادیار، دانشکده ریاضی و کامپیوتر، دانشگاه شهید باهنر، کرمان، ایران

<sup>۳</sup> دانشیار، دانشکده ریاضی و کامپیوتر، دانشگاه شهید باهنر، کرمان، ایران

### چکیده

در محیط ابر، زمان بندی بهینه کارها، به عوامل زیادی همچون زمان اتمام کارها، تاخیر و تعادل بار روی ماشین ها بستگی دارد. اما توجه اصلی سایر مقالات انجام شده فقط به کمینه کردن زمان اتمام کارهاست. در این مقاله، پارامترهای موثر بر کیفیت خدمات یعنی کمینه‌ی زمان اتمام کارها، کمینه‌ی میزان دیرکرد از موعد تحویل و بیشینه‌ی تعادل بار روی ماشین ها به عنوان توابع هدف، همزمان در نظر گرفته شده است. زمان بندی جزء مسائل پیچیده‌ای است که برای آن، راه حل از مرتبه‌ی چند جمله‌ی وجود ندارد و به همین دلیل از الگوریتم های فرااكتشافی مانند بهینه سازی گروه ذرات، شبیه سازی حرارت، جستجوی ممنوعه و الگوریتم ژنتیک چند هدفه استفاده شده است. نتایج شبیه سازی نشان داده است که بین الگوریتم های تک هدفه، الگوریتم بهینه سازی گروه ذرات در زمان کمتری اجرا شده و حتی برای تعداد ۳۰ کار، از نظر بهینه سازی تابع هدف، قابل رقابت با الگوریتم های چند هدفه می باشد. در تعداد کار زیاد، الگوریتم های چند هدفه، از نظر مدت زمان اجرا و همچنین بهینه سازی توابع هدف، مناسب ترند. زیرا الگوریتم ژنتیک مرتب شده یا رتبه بندی شده نامغلوب، توابع هدف را به صورت مجزا در نظر گرفته و بنابراین امکان انتخاب جواب های مطلوب را میان جواب های غیر برتر در راستای مصالحه بین اهداف مختلف را دارد.

**کلمات کلیدی:** پردازش ابری، زمان بندی کار، بهینه سازی چند هدفه.

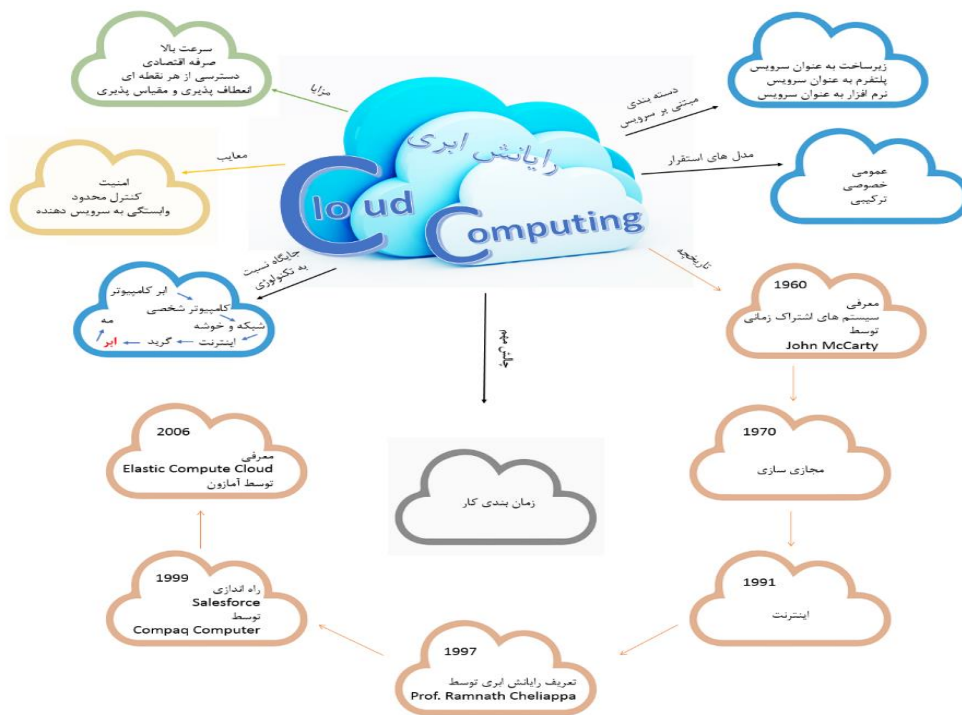
### ۱- مقدمه

مصرف، پرداخت داشته باشند. شکل ۱ به اختصار مزایا، معایب و دیگر پارامترهای مربوط به پردازش ابری را نمایش می دهد. در ابر همه چیز به عنوان سرویس<sup>۲</sup> ارائه می شود. از مزایای این قابلیت می توان به موارد زیر اشاره کرد: کاهش هزینه ها از طریق تبدیل هزینه های سرمایه ای به هزینه های عملیاتی، افزایش بهره وری و صرفه جویی در زمان، مقیاس پذیری، افزایش امنیت به دلیل تمرکز داده ها، انعطاف پذیری و دسترس پذیری از هر نقطه ای.

شکل ۱ به اختصار مزایا، معایب، جایگاه، تاریخچه، مدل های استقرار و چالش های اصلی در پردازش ابری را نشان می دهد. در این شکل معماری سرویس محور، شامل سطوح زیرساخت به شکل زیر است [۲]:  
زیرساخت به عنوان سرویس<sup>۳</sup>: به بخش های پایه ای رایانشی اشاره دارد که به صورت فیزیکی، مجازی، فضای ذخیره سازی و شبکه، قابل اجاره کردن هستند.

#### ۱-۱- پردازش ابری<sup>۱</sup>

پردازش ابری یک روش نوین پردازش است که با تجمیع حجم عظیمی از منابع محاسباتی به صورت مجازی سازی شده و قابل گسترش، به ایجاد یک سیستم یکپارچه پرداخته و از طریق شبکه های ارتباطی مانند اینترنت عرضه می شود. محوریت این مدل، سرویس دهی به کاربر بر اساس تقاضا است، بدون آن که کاربر نیازی به تجهیزات خاصی برای پردازش داشته یا از محل انجام این پردازش، آگاه باشد [۱]. یکی از جذابیت های پردازش ابری آن است که کاربران می توانند از هزینه ها و پیچیدگی رو به افزایش مالکیت و نگهداری زیرساخت های فناوری اجتناب کرده و به جای آن، برای قابلیتی که استفاده می کنند و در زمان



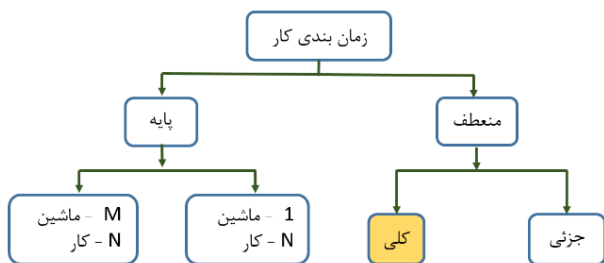
شکل ۱- پردازش ابری در یک نگاه

فضای جستجو به شکل رابطه‌ی ۱ می‌باشد.

$$Space\ Size = \frac{(N \times M)!}{(M!)^N} \quad (1)$$

در رابطه‌ی ۱،  $N$ : تعداد کارها و  $M$ : تعداد ماشین‌ها، می‌باشد.

در نتیجه، مسئله‌ی زمان‌بندی کار جزء مسائلی می‌باشد که برای آن راه‌حل از مرتبه‌ی چندجمله‌ای وجود ندارد و حل آن با روش‌های دقیق، بسیار زمان‌بر و پرهزینه می‌باشد. بنابراین بهترین انتخاب، استفاده از الگوریتم‌های فرااکتشافی<sup>۱۵</sup> است.



شکل ۲- مسئله‌ی زمان‌بندی کار

در عمل می‌توان مسئله‌ی زمان‌بندی کار را در حالت‌های مختلف، مدل کرد (شکل ۲). در این تقسیم‌بندی نوع پایه<sup>۱۶</sup>، به دو زیر شاخه تقسیم می‌شود. در نوع پایه، ماشین‌ها یکسان هستند. در نوع منعطف<sup>۱۷</sup>، کارها می‌توانند به بخش‌های کوچک‌تر (معمولاً با نام وظیفه<sup>۱۸</sup> شناخته می‌شوند) تقسیم و این بخش‌ها می‌توانند مستقلاً یا با ترتیب خاصی، روی ماشین‌ها اجرا شوند. در نوع منعطف در زیرشاخه‌ی کلی<sup>۱۹</sup> همه‌ی ماشین‌ها توانایی انجام همه‌ی کارها را دارند اما در زیرشاخه‌ی جزئی<sup>۲۰</sup>، بعضی ماشین‌ها توانایی انجام بعضی از کارها را ندارند.

در این مقاله ماشین‌ها توانایی انجام همه‌ی کارها را دارند ولی هر کار به شکل منفرد<sup>۲۱</sup>، (یعنی هر کار شامل بخش‌های کوچک‌تر نمی‌باشد) در نظر گرفته شده است.

برای حل مسئله در حالت تک ماشین، می‌توان جایگشت‌های مختلف کارها را در نظر گرفت و بهترین جایگشت را انتخاب کرد. یکی از معروف‌ترین مسائل در

این مورد، برای سازمان‌هایی جذاب است که بدنیا ساخت برنامه‌های کاربردی از پایه بوده و قصد دارند تقریباً هر عنصری را خودشان کنترل کنند. سکو به عنوان سرویس<sup>۴</sup>: این لایه نیز مانند فضای ذخیره‌سازی، شبکه و سرورهای مجازی، شامل ابزارها و نرم‌افزارهایی است که توسعه‌دهندگان نیاز دارند برنامه‌های کاربردی خود را بر فراز آن‌ها بسازند. این ابر، شامل میان‌افزار، مدیریت پایگاه داده، سیستم‌عامل‌ها و ابزارهای توسعه می‌باشد.

نرم‌افزار به عنوان سرویس<sup>۵</sup>: ارائه برنامه‌های کاربردی به عنوان سرویس است. در شکل ۱ مدل‌های استقرار شامل بخش‌های زیر است: ابر عمومی<sup>۶</sup>: یک مدل پردازش ابری رایج است که در آن کاربران می‌توانند به قدرت محاسباتی بالایی از طریق اینترنت دسترسی داشته باشند. یکی از مزایای مهم، توانایی مقیاس بخشیدن سریع به یک سرویس است. ابر خصوصی<sup>۷</sup>: به سازمان‌ها این امکان را می‌دهد تا از برخی از مزایای ابر عمومی، بدون نگرانی پیرامون از دست دادن کنترل خود روی داده‌ها و خدمات، بهره‌مند شوند.

ابر ترکیبی<sup>۸</sup>: در این نوع ابر، برخی از داده‌ها در ابر عمومی و بخشی از پروژه در ابر خصوصی قرار دارد. ارائه‌دهندگان، سرویس‌های پردازش ابری گوناگون و سطوح متفاوتی برای استفاده از پردازش ابری، به کاربر ارائه می‌کنند. یکی از مهم‌ترین چالش‌ها در پردازش ابری، مسئله‌ی زمان‌بندی کار<sup>۹</sup> می‌باشد. چرا که ارائه‌دهندگان خدمات ابری جهت رقابت با یکدیگر ملزم به ارائه‌ی خدمات در کمترین زمان می‌باشند. زمان‌بندی کار، همان تصمیم‌گیری در مورد اینکه کدام کار توسط کدام ماشین انجام شود، می‌باشد.

## ۱-۲- زمان بندی کار

مسئله زمان بندی کار در پردازش ابری کاربرد بسیاری دارد و به بررسی نحوه توزیع کارها بین ماشین‌های انجام‌دهنده می‌پردازد [۳]. به طوری که این کارها در کمترین زمان ممکن انجام شوند. چیدمان درست کارها باعث بهینه‌سازی پارامترهای زیادی از جمله: مصرف انرژی<sup>۱۰</sup> کمتر، بهره‌وری<sup>۱۱</sup> بالاتر، توزیع بار بهتر بین ماشین‌ها، زمان انتظار<sup>۱۲</sup> کمتر برای کارها، کم شدن جریمه تاخیر<sup>۱۳</sup> و در نهایت زمان اتمام<sup>۱۴</sup> کارها در کمترین زمان ممکن، می‌شود [۴].

در [۲۰] از الگوریتم جستجوی فاخته<sup>۳۳</sup> برای زمان‌بندی کارها در محیط ابر استفاده شده است. پارامتری که بهینه شده زمان پاسخ سرتاسری می‌باشد. در این مقاله کارها را بر اساس دو پارامتر (توان پردازشی ماشین و طول کار) زمان‌بندی می‌کنند.

در مرجع [۲۱] مسئله‌ی زمان‌بندی کار با استفاده از الگوریتم اجتماع زنبور<sup>۳۴</sup>، در محیط ابر انجام شده است. در این مرجع به صورت تک‌هدفه و چندهدفه پارامترهایی همچون: زمان اتمام کارها، حجم کار ماشین‌ها و حجم کار همه‌ی تجهیزات، بهینه شده‌اند. رویکرد این مرجع استفاده از دو نوع ماشین بوده است، ماشین‌های همسان و ماشین‌های غیرهمسان.

مهم‌ترین پارامترها در کیفیت سرویس‌دهی ارائه‌دهندگان ابر، به قرار زیر است:

۱- زمان اتمام کارها: این پارامتر به مفهوم، حداقل‌سازی زمان انجام آخرین کار می‌باشد. در واقع، کمینه بودن این زمان به معنی عملکرد بهتر سرویس‌دهنده می‌باشد. منابع زودتر آزاد شده و ارائه خدمات به تعداد بیشتری از کاربران، صورت می‌پذیرد.

۲- تعادل بار<sup>۳۵</sup> روی ماشین‌ها: این پارامتر برای جلوگیری از انباشت کارها روی یک ماشین می‌باشد. حفظ تعادل و رعایت انصاف در تقسیم کارها بین ماشین‌ها، بر روی زمان اتمام کارها و مصرف انرژی تأثیرگذار است. طبیعتاً اگر ارائه‌دهنده‌ی سرویس از تمام ظرفیت‌هایی که در اختیار دارد استفاده نکند و نتواند به شکل بهینه تقسیم کار کند، زمان اتمام کارها زیاد شده و از طرفی مصرف انرژی به دلیل عدم استفاده از منابع بیکار، زیاد می‌شود، بنابراین این پارامتر، مصرف انرژی را نیز تحت پوشش قرار می‌دهد.

۳- میزان دیرکرد از موعد تحویل<sup>۳۶</sup>: این پارامتر به میزان رضایت‌مندی مشتری ارتباط مستقیم دارد، زیرا مشتری درخواست خود را با کمترین تأخیر دریافت کرده و زمینه را برای درخواست بعدی فراهم می‌بیند، و این اتفاق در بلند مدت موجب رونق ارائه‌دهنده‌ی سرویس می‌شود.

با توجه به اهمیت پارامترهای مذکور، محدود شدن توجه اصلی سایر مقالات به کمینه کردن زمان اتمام کارها و نپرداختن همزمان به این پارامترها، باعث گردید در این مقاله تابع هدف به صورت ترکیب خطی و وزن‌دار کمینه‌ی زمان اتمام کارها، کمینه‌ی میزان دیرکرد از موعد تحویل و بیشینه‌ی تعادل بار روی ماشین‌ها، برای الگوریتم‌های تک‌هدفه و بصورت مستقل برای الگوریتم‌های چندهدفه تعریف شود. همچنین الگوریتم‌ها بر اساس این توابع هدف مقایسه شده و با استفاده از معیار زمان اجرا، سرعت عمل آن‌ها در مواجهه با مسئله‌ی زمان‌بندی مشخص گردید. تفکیک توابع هدف در هنگام شبیه‌سازی، این امکان را می‌دهد که الگوریتم‌ها نسبت به تک‌تک توابع هدف، مقایسه شوند. به عنوان مثال می‌توان الگوریتم‌ها را فقط بر اساس کمینه‌ی زمان اتمام کارها یا تعادل بار روی ماشین‌ها، ارزیابی نمود.

در این مقاله دو رویکرد دنبال شده است، اول اینکه با استفاده از الگوریتم‌های بهینه‌سازی ازدحام ذرات<sup>۳۷</sup>، شبیه‌سازی حرارت<sup>۳۸</sup> و جستجوی ممنوعه که همگی جزء الگوریتم‌های تک‌هدفه هستند و دوم، با استفاده از الگوریتم ژنتیک مرتب شده‌ی نامغلوب<sup>۳۹</sup> و الگوریتم ژنتیک رتبه‌بندی شده‌ی نامغلوب<sup>۴۰</sup> که جزء الگوریتم‌های چندهدفه هستند، به حل مسئله‌ی زمان‌بندی پرداخته و تمامی روش‌ها نسبت به بهینه‌سازی تابع هدف، با یکدیگر مقایسه شده‌اند. در شکل‌های ۳ و ۴ تقسیم‌بندی الگوریتم‌های استفاده شده، قابل مشاهده است.

در ادامه، بخش دوم به مرور کارهای مرتبط می‌پردازد، در بخش سوم تعاریف و الگوریتم‌های بهینه‌سازی ارائه خواهد شد، در بخش چهارم مدل‌سازی مسئله معرفی می‌شود، در بخش پنجم به شبیه‌سازی و در نهایت در بخش ششم جمع‌بندی و نتیجه‌گیری بیان می‌شود.

این زمینه، مسئله‌ی یک ماشین و تأخیر کارهای وزن‌دار است که به شکل  $1 / \sum w_i T_i$  نشان داده می‌شود. بررسی کلیه حالات در ترتیب قرارگیری کارها بسیار پیچیده و زمان‌بر است. مسئله  $1 / \sum w_i T_i$  به این صورت تعریف می‌شود: مجموعه‌ای از  $n$  کار مستقل که در زمان صفر بر روی یک ماشین قابل دسترس، وجود دارد. متناظر با هر کار  $i$ ،  $(i = 1, 2, \dots, n)$  یک زمان پردازش  $p_i$ ، و یک موعد تحویل  $(d_i)$  و وزن مربوط  $w_i$  تعیین شده است. فرض کنید  $\sigma = (1, 2, \dots)$  یک ترتیب از کارها باشد و  $i, i'$  امین کار باشد:

$$C_i = \sum_{j=1}^i p_j, \quad T_i = \max\{c_i - d_i, 0\} \quad (2)$$

$C_i$ : زمان تکمیل و  $T_i$ : تأخیر  $i$  امین کار هستند. هدف، پیدا کردن ترتیب  $\sigma$  است به نحوی که

$$Z(\sigma) = \sum_{i=1}^n w_i T_i \quad (3)$$

کمینه شود. تحقیقات زیادی روی مسئله  $1 / \sum w_i T_i$  انجام گرفته و رهیافت‌های متفاوتی برای حل آن ارائه شده است. در [۱] نشان داده شده است که  $1 / \sum w_i T_i$  یک مسئله سخت<sup>۴۲</sup> است. در [۵] با اولویت دادن به کارها، روشی را برای زمان‌بندی کارها ارائه کرده‌اند. نظریه‌ی آن‌ها شرایطی را تعیین می‌کند که در چه زمانی کار کوچک‌تر بر کار بزرگ‌تر مقدم است و بالعکس. راه‌حل‌های مسئله  $1 / \sum w_i T_i$  در دو نوع خلاصه می‌شوند: (۱) روش‌های حل دقیق (۲) روش‌های اکتشافی. هر چند راه‌حل‌های دقیقی با استفاده از برنامه‌ریزی پویا<sup>۴۳</sup> [۶-۷] و الگوریتم‌های شاخه و حد<sup>۴۴</sup> [۸-۹] وجود دارد، اما اجرای آن‌ها نیاز به حافظه و زمان زیادی دارد. در بیشتر کارهای اخیر، روش‌های مبتنی بر جستجوی ممنوعه<sup>۴۵</sup> با همسایگی‌های خاص ارائه شده که نتایج بهتری نسبت به روش‌های اکتشافی دارند [۱۰]. در الگوریتم ژنتیک<sup>۴۶</sup> متداول، عملگر ترکیب به عنوان عملگر پایه برای بهبود کارایی الگوریتم استفاده می‌شود و پیشرفت الگوریتم بستگی زیادی به این عملگر دارد [۱۱]. در [۱۲] با استفاده از داده‌های واقعی تأثیر ده عملگر ترکیب روی مسئله ذکر شده، بررسی شده است.

اما حالت چندماشینی، پیچیدگی بیشتری دارد. بدون شک حل مسئله زمان‌بندی کار به خاطر ماهیت این نوع مسائل با روش‌های دقیق، بسیار زمان‌بر و دارای بار محاسباتی بالایی است، بنابراین می‌توان با انتخاب مناسب الگوریتم‌های فرااکتشافی به جواب قابل قبول رسید. تاکنون روش‌های بسیاری برای حل این نوع مسائل ارائه شده است، ولی از آنجایی که این روش‌ها نیاز به زمان محاسباتی قابل توجه و یا محاسبات ریاضی پیچیده‌ای دارند، استفاده از GA در طی دو دهه گذشته بصورت موفقیت‌آمیزی آغاز گردیده و تاکنون نیز ادامه دارد [۱۳-۱۷].

در مرجع [۱۸] با استفاده از الگوریتم بهینه‌سازی نهنگ<sup>۴۷</sup> مسئله‌ی زمان‌بندی کار حل شده و علاوه بر آن به بهبود الگوریتم نهنگ هم پرداخته و پارامترهای، زمان اجرا<sup>۴۸</sup> و زمان پاسخ<sup>۴۹</sup> را کاهش و توان عملیاتی<sup>۳۰</sup> محیط پردازش ابری را افزایش داده است.

یکی از جدیدترین الگوریتم‌های فرااکتشافی، که در دسته‌ی الگوریتم‌های زیستی و مبتنی بر جمعیت می‌باشد، الگوریتم بهینه‌سازی ازدحام جوجه<sup>۴۱</sup> است. این الگوریتم با الهام از رفتار سلسله مراتبی جوجه‌ها در یک جمعیت، برای یافتن غذا است. حرکات متنوع جوجه‌ها در یافتن غذا باعث ایجاد تعادل بین جستجوی محلی و عمومی می‌شود. الگوریتم بهینه‌سازی کلاغ<sup>۴۲</sup> در دسته‌ی الگوریتم‌های زیستی و مبتنی بر جمعیت می‌باشد. مزیت این الگوریتم در استفاده از شیوه ادراک فردی در فرایند جستجوی فضای مسئله است. مرجع [۱۹] از ویژگی‌های این دو الگوریتم استفاده و یک الگوریتم ترکیبی ایجاد کرده که از آن برای زمان‌بندی کار بهره برده است. پارامترهایی که بهینه می‌کند به این قرار است: زمان اجرا و زمان پاسخ را کاهش و توان عملیاتی محیط پردازش ابری را افزایش می‌دهد.

تکرار کمتر به جواب برسند، الگوریتم حریصانه<sup>۴۹</sup> را بکار گرفته و GA را بهبود داده‌اند.

با استقرار روز افزون مراکز داده در سرتاسر جهان، پردازش ابری به عنوان یکی از مهم‌ترین قابلیت‌ها، مطرح شده است. با این حال، محیط ابر با چالش‌های زیادی از جمله مصرف انرژی، زمان اجراء، انتشار گرما و انتشار گاز منوکسیدکربن<sup>۵۰</sup> و همچنین هزینه عملیاتی، روبرو است. زمان‌بندی کارآمد در محیط ابر یکی از مهمترین راه‌ها برای مقابله با مشکلات فوق و دستیابی به استفاده بهینه از منابع است. در مرجع [۳۰] یک الگوریتم بهینه‌سازی آگاه از انرژی و زمان<sup>۵۱</sup> بر اساس الگوریتم BA پیشنهاد شده است. هدف این است که بدون از دست دادن کیفیت خدمات<sup>۵۲</sup>، مصرف انرژی و زمان اجرای کار را به حداقل برسانند. الگوریتم پیشنهادی با الگوریتم ACO و الگوریتم جستجوی تصادفی<sup>۵۳</sup> مقایسه شده است.

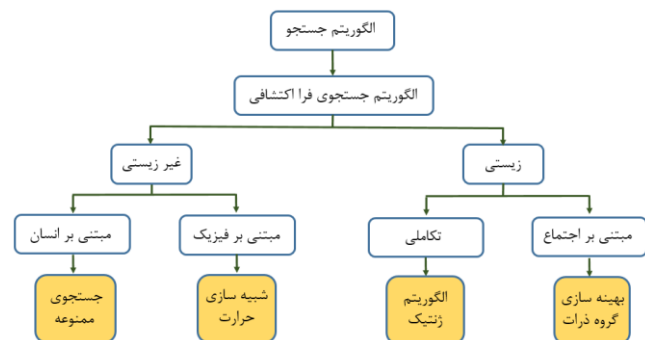
زمان‌بندی کار در پردازش ابری می‌تواند به طور مستقیم بر مصرف منابع و هزینه عملیاتی سیستم تأثیر داشته باشد. در مرجع [۳۱]، WOA برای زمان‌بندی کار و یک مدل بهینه‌سازی چندهدفه، به منظور بهبود عملکرد سیستم ابر استفاده شده و بر این اساس، یک روش پیشرفته به نام الگوریتم بهینه‌سازی نهنگ بهبودیافته<sup>۵۴</sup> ارائه شده است. شبیه‌سازی‌ها نشان می‌دهد که الگوریتم پیشنهادی دارای سرعت همگرایی و دقت بهتری نسبت به الگوریتم‌های فرااکتشافی فعلی می‌باشد. علاوه بر این، برای کارهای کوچک و بزرگ نیز می‌تواند در استفاده از منابع سیستم عملکرد بهتری داشته باشد. الگوریتم بهبودیافته با الگوریتم‌های ACO، PSO و WOA، مقایسه شده است.

پردازش ابری، به عنوان یک مدل سرویس‌دهی فناوری، به طور موثری انواع مختلف درخواست‌ها را زمان‌بندی می‌کند، که از نظر صرفه‌جویی در منابع، بهبود کیفیت خدمات و سرعت پاسخگویی مراکز داده ابری، بسیار مهم است. در مرجع [۳۲] یک استراتژی زمان‌بندی با هدف کاهش ماشین‌های مجازی و کاهش زمان اتمام کارها معرفی شده است. روش موجود در این مقاله از مزایای الگوریتم ACO استفاده می‌کند. با کاهش زمان کار ماشین‌های مجازی، ضمن در نظر گرفتن زمان اجرای کارها، استفاده از منابع مرکز داده، بهبود می‌یابد. شبیه‌سازی در محیط CloudSim انجام شده و اثربخشی الگوریتم، در مقایسه با GA و الگوریتم RS، اثبات شده است. در این مقاله، زمان اتمام کارها و تعداد ماشین‌های مجازی به عنوان پارامترهای بهینه‌سازی انتخاب شده‌اند.

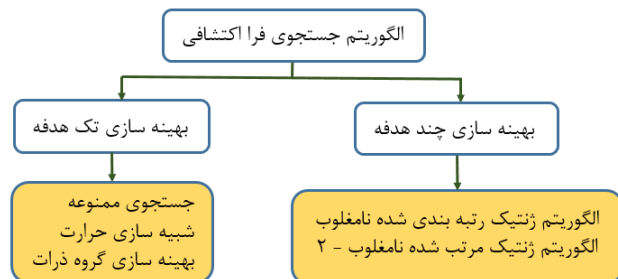
در [۳۳] GA با الگوریتم ACO ترکیب شده است تا پارامترهایی همچون: زمان اتمام کارها و مصرف انرژی را بهینه کنند. نویسندگان از محیط شبیه‌سازی CloudSim استفاده کرده و الگوریتم بهبودیافته را با GA و الگوریتم ACO، مقایسه کرده‌اند.

در [۳۴] برای بهینه‌سازی عملکرد ارائه‌دهنده خدمات ابر، روش بهینه‌سازی فرااکتشافی<sup>۵۵</sup> برای تعادل بار پیشنهاد شده است. تکنیک پیشنهادی مناسب شرایط ایستا و پویا بوده و شامل دو مرحله بهینه‌سازی فرااکتشافی-ایستا<sup>۵۶</sup> و بهینه‌سازی فرااکتشافی-پویا<sup>۵۷</sup> برای مقابله با خصوصیات ایستا و پویای کار ارسالی، می‌باشد. تجزیه و تحلیل نتایج در مقایسه با الگوریتم بهینه‌سازی جایگذاری ماشین مجازی آگاه از محتوی<sup>۵۸</sup> و الگوریتم بهینه شده هزینه<sup>۵۹</sup>، ثابت می‌کند که روش پیشنهادی عملکرد بهتری در محیط‌های پیچیده دارد.

پردازش ابری فناوری است که دنیای محاسبات را از طریق قدرت خود تغییر می‌دهد، تا هر کاربری که از طریق اینترنت به قدرت محاسبات بالاتری نیاز دارد را تأمین کند. در [۳۵] الگوریتم بهینه‌سازی ازدحام ذرات صحیح<sup>۶۰</sup>، برای زمان‌بندی کار در محیط ابر پیشنهاد شده که می‌تواند برای بهینه‌سازی یک تابع هدف واحد و یا چندین تابع هدف استفاده شود. مطالعات تجربی در مورد انواع مختلف مجموعه وظایف توصیف‌کننده ترافیک عادی و ترافیک ناگهانی در محیط پردازش ابری نشان می‌دهد که این روش از همگرایی و تعادل بار مناسبی، برخوردار است.



شکل ۳- تقسیم‌بندی الگوریتم‌های استفاده شده از لحاظ ماهیت الگوریتم.



شکل ۴- تقسیم‌بندی الگوریتم‌های استفاده شده از لحاظ نوع بهینه‌سازی.

## ۲- کارهای مرتبط

استفاده از الگوریتم PSO را می‌توان در مرجع [۲۲] مشاهده کرد. در این مرجع از الگوریتم PSO با باینری<sup>۴۱</sup> (تفاوت آن با الگوریتم PSO معمولی این است که موقعیت هر ذره با تابع سیگموئید<sup>۴۲</sup> به‌روزرسانی می‌شود و فقط سرعت جاری ذره در به‌روزرسانی موقعیت دخالت دارد) برای کمینه‌کردن زمان اتمام کارها و همچنین تعادل بار روی ماشین‌ها استفاده شده است.

در مرجع [۲۳] از الگوریتم PSO و GA به صورت ترکیبی استفاده شده است تا پارامترهای زمان اتمام کار، مقیاس‌پذیری و در دسترس بودن را بهینه کنند. برای این کار از دو عدد صف با عناوین صف اولویت<sup>۴۳</sup> و صف بر اساس تقاضا<sup>۴۴</sup> استفاده شده است. مراجع زیادی به ترکیب الگوریتم‌های بهینه‌سازی پرداخته و از مزایای آن‌ها استفاده کرده‌اند. [۲۴] از الگوریتم بهینه‌سازی اجتماع مورچه<sup>۴۵</sup> برای تنظیم پارامترهای الگوریتم PSO استفاده کرده است. این الگوریتم یکپارچه قادر به نگه داشتن ذرات در سطح مناسبی از سازگاری است و تنوع جمعیت را تضمین می‌کند. علاوه بر این، بهترین راه‌حل سراسری با همگرایی بالا را می‌توان با تنظیم فاکتور یادگیری، بدست آورد.

در مرجع [۲۵] از یک الگوریتم ترکیبی برای زمان‌بندی جریان‌های کارها در محیط ابر استفاده شده است. پارامترهایی که بهینه شده‌اند عبارتند از: کمترین زمان برای اتمام تمام کارها و کاهش هزینه‌ی اجراء. در این مقاله الگوریتم گریه‌ماهی<sup>۴۶</sup> با الگوریتم PSO ترکیب شده‌اند. محیط شبیه‌سازی، شبیه‌ساز WorkflowSim [۲۶] بوده که گسترش‌یافته‌ی شبیه‌ساز معروف CloudSim [۲۷] می‌باشد.

در مرجع [۲۸] برای بهبود کیفیت خدمات از بهینه‌سازی ازدحام ذرات چندهدفه<sup>۴۷</sup> استفاده شده است تا پارامترهایی همچون: زمان، هزینه، توان پردازشی و نرخ پذیرش کارها را بهینه کنند. نویسندگان از محیط شبیه‌ساز CloudSim استفاده کرده و با الگوریتم‌های PSO، ABC، الگوریتم خفاش<sup>۴۸</sup> و PSO بهبودیافته مقایسه کرده‌اند. در مرجع [۲۹] برای بهبود زمان اتمام کارها، کیفیت انجام کارها و میانگین زمان پاسخ، از GA استفاده شده است. نویسندگان برای اینکه در تعداد

جدول ۱- بررسی کارهای انجام شده در زمینه زمان‌بندی کار

| الگوریتم استفاده شده  | تابع هدف  |            |              |                                |           |                                | زمان اتمام کارها |
|---|-----------|------------|--------------|--------------------------------|-----------|--------------------------------|------------------|
|   | زمان پاسخ | مصرف انرژی | هزینه‌ی اجرا | میزان دیرکرد از موعد تحویل کار | تعادل بار | میزان دیرکرد از موعد تحویل کار |                  |
| Binay PSO (BPSO)  | -         | -          | -            | -                              | +         | +                              | [۲۲]<br>۲۰۱۹     |
| ترکیب الگوریتم PSO با GA  | -         | -          | -            | -                              | -         | +                              | [۲۳]<br>۲۰۱۸     |
| ترکیب الگوریتم PSO با الگوریتم ACO  | -         | -          | +            | -                              | -         | +                              | [۲۴]<br>۲۰۱۷     |
| ترکیب الگوریتم PSO با الگوریتم Cat-fish                                   | -         | -          | +            | -                              | -         | +                              | [۲۵]<br>۲۰۱۶     |
| الگوریتم PSO و الگوریتم ACO   | +         | -          | +            | -                              | -         | +                              | [۲۸]<br>۲۰۱۹     |
| بهبود GA  | +         | -          | -            | -                              | -         | +                              | [۲۹]<br>۲۰۲۰     |
| بهبود الگوریتم BA و تبدیل آن به الگوریتم بهینه‌سازی آگاه از انرژی و زمان  | -         | +          | -            | -                              | -         | +                              | [۳۰]<br>۲۰۲۰     |
| بهبود WOA   | -         | +          | +            | -                              | +         | -                              | [۳۱]<br>۲۰۲۰     |
| الگوریتم ACO  | -         | -          | -            | -                              | -         | +                              | [۳۲]<br>۲۰۲۰     |
| ترکیب GA با الگوریتم ACO  | -         | +          | -            | -                              | -         | +                              | [۳۳]<br>۲۰۲۰     |
| الگوریتم بهینه‌سازی جدید بر پایه‌ی الگوریتم‌های فرااکتشافی                | -         | -          | -            | -                              | +         | -                              | [۳۴]<br>۲۰۲۰     |
| بهبود الگوریتم PSO و ایجاد الگوریتم جدید با نام بهینه‌سازی گروه ذرات صحیح | -         | -          | +            | -                              | -         | +                              | [۳۵]<br>۲۰۱۹     |

در رابطه‌ی ۴،  $st_i$ : اندازه‌ی کار  $i$ ام و  $pr_j$ : نرخ پردازش ماشین  $i$ ام می‌باشد.

### ۳- تعاریف و الگوریتم‌های بهینه‌سازی

#### ۳-۱- تعاریف

**تعاریف ۱:** (ماشین مجازی  $vm$ )<sup>۶۱</sup>. یک ماشین مجازی می‌تواند پارامترهایی همچون: شماره مشخصه  $(id)$ <sup>۶۲</sup> و توان پردازشی  $(MIPS)$ <sup>۶۳</sup> را داشته باشد. در این مقاله ماشین‌ها توان پردازشی متفاوتی دارند و به طور موازی کار می‌کنند.

**تعاریف ۲:** (کار  $64$ ). کار می‌تواند پارامترهایی همچون: شماره مشخصه، تعداد وظیفه، وزن و اندازه داشته باشد. در این مقاله هر کار شامل شماره‌ی مشخصه و اندازه می‌باشد و هر کار به صورت واحد در نظر گرفته می‌شود (به عبارت دیگر به بخش‌های کوچک‌تر تقسیم نشده است).

**تعاریف ۳:** (زمان آماده‌سازی اولیه برای هر کار). که با نماد  $S_0$  مشخص شده است.

**تعاریف ۴:** (زمان آماده‌سازی بین کارها). که با نماد  $S$  مشخص شده است. در واقع  $S$  یک ماتریس  $N \times N$  می‌باشد. با این تحلیل که مدت زمان آماده‌سازی، همان مدت زمان مورد نیاز برای رسیدن کار بعدی است.

**تعاریف ۵:** (نسبت اندازه‌ی کار به نرخ پردازشی ماشین  $(et_{i,j})$ ). نسبت اندازه-ی کار  $i$ ام به توان پردازشی ماشین  $i$ ام، روی کارهایی است که توسط ماشین  $i$ ام انجام می‌شوند. هر چه اندازه این عدد بزرگتر باشد، یعنی این کار برای این ماشین نامناسب‌تر است و بهتر است که این کار به ماشین دیگری سپرده شود. از این نماد برای جریمی‌های ماشین استفاده می‌شود. بنابراین می‌توان این نماد را در فرمول محاسبه‌ی زمان اتمام کار برای هر ماشین لحاظ کرده و در بهینه‌سازی دخالت داد [۳۶].

$$et_{i,j} = st_i/pr_j \quad (۴)$$

**تعاریف ۶:** (زمان اتمام کارها در هر ماشین  $(CTM)$ ). زمان اجرای کارها روی هر ماشین را با نماد  $ST_{i,j}$  نشان می‌دهیم. برای محاسبه‌ی  $ST_{i,j}$  می‌بایست برای اولین کاری که روی این ماشین اجرا می‌شود  $S_{01}$  را به اضافه‌ی زمان مورد نیاز برای اجرای این کار در نظر بگیریم  $(S_{01}+ptt_1)$ ، ولی برای کارهای بعدی (به عنوان مثال کار  $i$ ام) زمان آماده‌سازی بین کار قبلی (کار  $a$ ام) و این کار به اضافه‌ی زمان مورد نیاز برای اجرای این کار در نظر گرفته می‌شود  $(ptt_a+S_{ab+})$  و در نتیجه زمان اتمام کار یک ماشین بدست می‌آید. شکل ۶ به توصیف این فرایند می‌پردازد. سیستمی مفروض است که تعدادی کار و تعدادی ماشین دارد. ماشین شماره‌ی ۲ موظف است کارهای ۱ و ۳ را به ترتیب اجرا کند، پس  $vm_2 = \{1,3\}$  از طرفی زمان آماده‌سازی اولیه برای این دو کار بصورت  $S_{03} = 2$  و  $S_{01} = 3$  و زمان پردازش کار اول و سوم برابر  $ptt_1 = 12$  و  $ptt_3 = 15$  می‌باشد. رابطه‌ی ۵ بیانگر فاصله‌ی زمانی بین این کارها می‌باشد.

$$S = \begin{bmatrix} S_{1,1} = \infty & S_{1,3} = 5 \\ S_{3,1} = 8 & S_{3,3} = \infty \end{bmatrix} \quad (۵)$$

با توجه به توضیحاتی که داده شد چون کار شماره‌ی ۱ اولین کاری است که روی ماشین ۲ اجرا می‌شود پس فعلاً زمان اتمام کار ماشین شماره ۲ به صورت مجموع دو عنصر  $S_{01}$  و  $ptt_1$  است (شکل ۵).

فرایند بالا را می‌توان به طور خلاصه بدین شکل بیان کرد که باید کارهایی که روی هر ماشین اجرا می‌شوند را جدا کرده و به دو مورد توجه کنیم اول اینکه زمان آماده‌سازی اولیه برای هر کار (در صورتی که اولین کاری باشد که روی ماشین اجرا می‌شود  $S_0$ ) و دوم زمان اتمام کار قبلی که می‌بایست به زمان اتمام کار فعلی اضافه شود  $(CT_{i-1})$ . بنابراین برای هر ماشین داریم:

$$CTM_j = \sum_{i=1}^k et_{i,j} + ST_{i,j} \quad (۶)$$

$k$ : تعداد کارهایی که روی ماشین  $i$ ام اجرا می‌شوند.

### ۳-۲-۱- الگوریتم PSO

هوش جمعی<sup>۷۰</sup> نوعی روش هوش مصنوعی است که مبتنی بر رفتارهای جمعی در سامانه‌های نامتمرکز و خودسامانده بنیان شده است. این سامانه‌ها معمولاً جمعیتی از کنش‌گران ساده‌اند که بطور محلی با یکدیگر و با محیط خود در تعامل هستند. با وجود اینکه معمولاً هیچ کنترل تمرکز یافته‌ای، چگونگی رفتار کنش‌گران را به آن‌ها تحمیل نمی‌کند، تعاملات محلی آن‌ها به پیدایش رفتاری عمومی می‌انجامد. از جمله این روش‌ها می‌توان به الگوریتم PSO، که یکی از مهم-ترین الگوریتم‌هایی است که در حوزه هوش جمعی جای می‌گیرد، اشاره کرد. این الگوریتم، با الهام از رفتار اجتماعی حیواناتی چون ماهی‌ها و پرندگان که در گروه‌هایی کنار هم زندگی می‌کنند، توسط جیمز کندی و راسل سی برهات در سال ۱۹۹۵ معرفی گردید [۳۹]. در الگوریتم PSO، اعضای جمعیت جواب‌ها، به صورت مستقیم با هم ارتباط دارند و از طریق تبادل اطلاعات، به حل مسأله می‌رسند. از مزایای الگوریتم PSO می‌توان به سرعت همگرایی بالا، بهره‌مندی از حافظه، همکاری و اشتراک‌گذاری اطلاعات بین ذرات و راحتی و سادگی پیاده‌سازی و اجرا، اشاره کرد. از معایب الگوریتم PSO می‌توان به همگرایی زودرس، گرفتار شدن در بهینه‌ی محلی و کاهش تنوع جمعیت، اشاره کرد.

تفاوت PSO با باینری PSO معمولی این است که موقعیت هر ذره با تابع احتمال بروزسانی می‌شود و در مورد خاص فقط سرعت جاری ذره در بروزسانی موقعیت دخالت دارد. در واقع موقعیت هر ذره با یک احتمال به‌روز می‌شود. توابع احتمال متفاوتی، از جمله تابع سیگموئید را می‌توان استفاده کرد. رابطه‌ی ۱۰ بیانگر تابع سیگموئید است.

$$\rho = \frac{1}{1+e^{-z}} \quad (10)$$

$z$ : سرعت ذره در لحظه‌ی  $t$  است. در این مقاله از همین تفاوت استفاده شده و برای اینکه موقعیت ذره به‌روز شود از این تابع به این صورت استفاده می‌شود که  $\rho$  برای هر بعد از بردار سرعت ذره در لحظه‌ی  $t$  محاسبه می‌گردد، اگر این احتمال از ثابت  $d$  کمتر بود موقعیت در همین بعد به‌روز شده و اگر بیشتر بود موقعیت ذره در این بعد بدون تغییر باقی می‌ماند. در واقع با زیاد شدن سرعت ذره در یک بعد، احتمال  $\rho$  هم زیاد شده، بنابراین اجازه‌ی تغییر موقعیت در این بعد، به ذره داده نمی‌شود.

الگوریتم PSO به صورت زیر می‌باشد [۳۹]:

۱. در ابتدا جمعیت ذرات و پارامترهای الگوریتم را تنظیم کنید.
۲. مرحله‌ی ۳ را به ازای تعداد تکرار در نظر گرفته شده، اجرا کنید.
۳. برای تمامی اعضای جمعیت مراحل زیر را انجام دهید:

- سرعت و موقعیت هر ذره را بر اساس معادلات زیر به‌روزسانی کنید:

$$v_i^{t+1} = w \times v_i^t + c_1 \times r_1 \times (pbest_i^t - x_i^t) + c_2 \times r_2 \times (gbest_i^t - x_i^t) \quad (11)$$

$$x_i^{t+1} = x_i^t + v_i^{t+1} \quad (12)$$

- تابع برازش<sup>۷۱</sup> را برای این ذره محاسبه کنید.
- بهترین محلی ( $pbest_i^t$ ) و بهترین سراسری ( $gbest_i^t$ ) را بروز کنید.

در رابطه‌ی (۱۱)،  $w$ : وزن اینرسی،  $c_1$  و  $c_2$  ثابت،  $x_i^t$ : موقعیت ذره  $i$ ام در تکرار  $t$ ام،  $v_i^t$ : سرعت ذره  $i$ ام در تکرار  $t$ ام،  $pbest$ : بهترین موقعیت ملاقات شده توسط هر ذره و  $gbest$ : بهترین موقعیت ملاقات شده در کل جمعیت می‌باشد.

### ۳-۲-۲- الگوریتم SA

الگوریتم SA، یک الگوریتم فرااكتشافی می‌باشد که از یک پدیده‌ی طبیعی الهام گرفته شده است. فرایند سخت‌سازی فلز را در نظر بگیرید. برای این کار

لذا باید بررسی شود که این کار، اولین کاری هست که روی ماشین اجرا می‌شود یا نه؟ اگر به عنوان اولین کار باشد که فقط زمان آماده‌سازی اولیه ( $S_0$ ) به زمان پردازش آن کار اضافه می‌شود. ولی اگر به عنوان اولین کار نباشد، بنابراین زمان اتمام کارهای قبلی هم مهم و تاثیرگذار است و باید هر تعداد کار که قبل از این کار روی این ماشین اجرا شده را در نظر گرفته و به زمان کامل شدن این کار اضافه شود.

**تعریف ۷:** (زمان اتمام آخرین کار<sup>۶۵</sup>). این پارامتر بیانگر این است که هدف مسأله، یافتن بهترین ترتیب انجام کارها به منظور انجام تمامی کارها در سریع‌ترین زمان ممکن است. این پارامتر بهره‌وری را بطور قابل ملاحظه‌ای افزایش می‌دهد. کاربرد این تابع هدف در مواقعی است که درخواست‌ها بصورت گروهی تحویل سرویس‌گیرنده می‌گردند.

**تعریف ۸:** (زمان کامل شدن هر کار<sup>۶۶</sup>)). از آنجا که هر کار تنها می‌تواند روی یک ماشین اجرا شود و تا زمانی که به اتمام نرسد ماشین را رها نمی‌کند، زمان کامل شدن مجموع زمان‌های اجرا، فاصله‌ی زمانی بین این کار و کار قبلی و زمان کامل شدن کارهای قبلی، می‌باشد. به عنوان مثال زمان کامل شدن اولین کار در شکل ۵ حاصل جمع دو پارامتر اول و زمان کامل شدن کار شماره‌ی ۳ برابر مجموع تمامی زمان‌هایی است که در شکل ۶ نمایش داده شده (البته به غیر از  $et_{1,j}$  و  $et_{3,j}$ ).

**تعریف ۹:** (موعد تحویل<sup>۶۶</sup>  $(d_i)$ ). مدت زمانی است که کاربر برای گرفتن سرویس، تعیین می‌کند. یا به بیان دیگر، زمانی است که سرویس‌دهنده متعهد به تحویل کار به مشتری نهایی است. تحویل کار به مشتری پس از موعد تحویل امکان‌پذیر است ولی سرویس‌دهنده بایستی جریمه‌ای را بابت دیرکرد متحمل شود. برای محاسبه‌ی موعد تحویل هر کار می‌توان به دو طریق عمل کرد. اول اینکه با استفاده از زمان پردازش و زمان آماده‌سازی هر کار به زمان اتمام کارها به صورت فرضی دست پیدا کرد و با استفاده از آن موعد تحویل هر کار را تخمین زد. دوم اینکه، با استفاده از ماتریس  $et_{i,j}$  و برای هر سطر یعنی برای هر کار، کمترین و بیشترین مقدار را محاسبه کرده و یک بازه را ایجاد نمود و موعد تحویل هر کار را از این بازه تخمین زد.

$$CTM_2: \quad S_{01} \quad ptt_1 \quad et_{1,j}$$

شکل ۵- محاسبه‌ی زمان اتمام اولین کار

بعد از ورود کار شماره‌ی ۳ به ماشین شماره‌ی ۲، زمان اتمام کار این ماشین مانند شکل ۶ تغییر می‌کند:

$$CTM_2: \quad S_{01} \quad ptt_1 \quad et_{1,j} \quad S_{1,3} \quad ptt_3 \quad et_{3,j}$$

شکل ۶- محاسبه‌ی زمان اتمام کارها روی ماشین شماره‌ی ۲

**تعریف ۱۰:** (مدت زمان تاخیر کار<sup>۶۷</sup>). که با رابطه‌ی ۷ مشخص می‌شود [۳۷].

$$lateness_i = CTT_i - d_i \quad (7)$$

در رابطه‌ی ۷،  $CTT_i$ : زمان اتمام کار  $i$ ام و  $d_i$ : موعد تحویل کار  $i$ ام می‌باشد.

**تعریف ۱۱:** (واحد جریمه<sup>۶۸</sup>). برای هر کار بصورت رابطه‌ی ۸ تعریف می‌شود [۳۷].

$$U_i = \begin{cases} 1 & \text{if } CTT_i > d_i \\ 0 & \text{otherwise} \end{cases} \quad (8)$$

**تعریف ۱۲:** (میزان دیرکرد از موعد تحویل کار<sup>۶۹</sup>  $(T_i)$ ). که با رابطه‌ی ۹ مشخص می‌شود [۳۸].

$$T_i = \max(lateness_i, 0) \quad (9)$$

### ۳-۲- الگوریتم‌های بهینه‌سازی

۲. الگوریتم بهترین جواب همسایه را از میان همسایه‌های جواب فعلی انتخاب می‌کند. در صورتی که این جواب در فهرست ممنوعه قرار نداشته باشد، الگوریتم به جواب همسایه حرکت می‌کند، در غیراین‌صورت الگوریتم معیاری به نام معیار تنفس را چک خواهد کرد. بر اساس معیار تنفس اگر جواب همسایه از بهترین جواب یافت شده تاکنون بهتر باشد، الگوریتم به آن حرکت خواهد کرد، حتی اگر آن جواب در فهرست ممنوعه باشد.

۳. پس از حرکت الگوریتم به جواب همسایه، فهرست ممنوعه به‌روزرسانی می‌شود، به این معنا که حرکت قبل، که بوسیله‌ی آن به سمت جواب همسایه حرکت کرده بود، در فهرست ممنوعه قرار داده می‌شود تا از بازگشت مجدد الگوریتم به آن جواب و ایجاد چرخه جلوگیری شود.

۴. پس از قرار دادن حرکت قبلی در فهرست ممنوعه، تعدادی از حرکت‌هایی که قبلاً در فهرست ممنوعه قرار گرفته بودند از فهرست خارج می‌شوند. مدت زمانی که حرکت‌ها در فهرست ممنوعه قرار می‌گیرند توسط یک پارامتر که زمان ممنوعه<sup>۷۶</sup> نام دارد تعیین می‌شود.

۵. حرکت از جواب فعلی به جواب همسایه (مرحله‌ی ۲) تا جایی ادامه می‌یابد که شرط خاتمه دیده شود. شرط‌های خاتمه‌ی متفاوتی می‌توان برای الگوریتم در نظر گرفت. به‌طور مثال محدودیت تعداد حرکت به جواب همسایه می‌تواند یک شرط خاتمه باشد.

### ۳-۲-۴- NSGA-II

الگوریتم NSGA-II یکی از پرکاربردترین و قدرتمندترین الگوریتم‌های موجود برای حل مسائل بهینه‌سازی چندهدفه است که کارایی آن در حل مسائل مختلف، به اثبات رسیده است. اسرینیاس و دپ در سال ۱۹۹۵ روش بهینه‌سازی NSGA را برای حل مسائل بهینه‌سازی چندهدفه معرفی نمودند، جوابی که هیچ جواب دیگری، به طور قطع بهتر از آن نباشد، دارای امتیاز بیشتری است. جواب‌ها بر اساس این که چند جواب بهتر از آن‌ها وجود داشته باشند، رتبه‌بندی و مرتب می‌شوند. شایستگی (برازندگی) برای جواب‌ها، بر حسب رتبه‌ی آن‌ها و عدم غلبه‌ی سایر جواب‌ها، اختصاص می‌یابد. از شیوه اشتراک برازندگی برای جواب‌های نزدیک استفاده می‌شود تا به این ترتیب پراکندگی جواب‌ها به نحو مطلوبی تنظیم شود و جواب‌ها به طور یکنواخت در فضای جستجو پخش شوند.

با توجه به حساسیت نسبتاً زیادی که نحوه عملکرد و کیفیت جواب‌های الگوریتم NSGA به پارامترهای اشتراک برازندگی و سایر پارامترها دارند، نسخه دوم الگوریتم NSGA با نام الگوریتم فرااکتشافی NSGA-II معرفی گردید [۴۳]. در کنار تمام کارایی‌هایی که الگوریتم فرااکتشافی NSGA-II دارد، می‌توان آن را الگوی شکل‌گیری بسیاری از الگوریتم‌های بهینه‌سازی چندهدفه دانست. این الگوریتم و شیوه منحصر به فرد آن در برخورد با مسائل بهینه‌سازی چندهدفه، بارها توسط افراد مختلف برای ایجاد الگوریتم‌های بهینه‌سازی چندهدفه جدیدتر، مورد استفاده قرار گرفته است.

### ۳-۲-۵- NPGA

در الگوریتم NSGA، روش انتخاب در عملگرهای جهش و ترکیب، مسابقه دودویی<sup>۷۷</sup> ولی در NPGA، روش انتخاب، چرخ رولت<sup>۷۸</sup> می‌باشد.

## ۴- مدل‌سازی مسئله

محیطی با  $N$  کار و  $M$  ماشین را در نظر بگیرید که کارها مستقل هستند و ماشین‌ها قابلیت اجرای همه‌ی کارها را داشته و محدودیتی وجود ندارد. هر یک از کارها دارای اندازه و متناسب با آن دارای یک زمان مورد نیاز برای اجرا هستند. هر یک از ماشین‌ها هم دارای یک توان پردازشی مشخص می‌باشند.

می‌بایست فلز را حرارت دهند و سپس به تدریج سرد کنند. در طی این فرایند بر اثر حرارت، انرژی جنبشی ذرات تشکیل‌دهنده‌ی فلز زیاد شده و ذرات جابجا می‌شوند. در این حالت ذرات در حالت پایداری قرار می‌گیرند و پس از سرد شدن، مجدداً بین آن‌ها پیوند برقرار می‌شود ولی در این حالت، پیوند به مراتب محکم‌تر از قبل است، چرا که انرژی پتانسیلی که در ذرات ذخیره شده بود، به انرژی جنبشی تبدیل شده و اصطلاحاً آزاد می‌شود. تصور کنید فردی که در سرازیری ایستاده است را به راحتی می‌توان حرکت داد ولی همان فرد را وقتی که در سطح صاف قرار می‌گیرد، نمی‌توان به راحتی حرکت داد.

همین فرایند را برای پیدا کردن راه‌حل یک مسئله در نظر بگیرید، وقتی الگوریتم جواب بهتری به نسبت قبل پیدا کرد، بدون هیچ شرطی آنرا قبول می‌کند ولی خیلی زود در بهینه‌ی محلی گیر می‌افتد برای فرار از بهینه‌ی محلی می‌بایست با یک احتمال مشخص (فاکتور احتمالی بولتزمن<sup>۷۲</sup> که با رابطه‌ی ۱۴ مشخص است) از بین همسایگان بد، یکی را انتخاب کند. در این لحظه دما کمتر می‌شود. در دماهای بالا (یعنی وقتی الگوریتم در شروع کار است و هنوز راه‌حل‌های مختلفی را جستجو نکرده است) احتمال انتخاب حالت بدتر، بالاتر می‌رود. از طرفی هر چه، اختلاف نقطه‌ی جدید پیدا شده با نقطه‌ی فعلی کمتر باشد، باز هم احتمال قبول کردن این حالت بدتر، بالاتر می‌رود. پس هر چه در مراحل الگوریتم پیش برود، احتمال این که الگوریتم، بدتر شدن شرایط را قبول کند، کمتر می‌شود [۴۰].

$$\Delta E = E_2 - E_1 \quad (13)$$

$$BPF = e^{\frac{-\Delta E}{T}} \quad (14)$$

$E_1$ : ارزش فعلی.  $E_2$ : ارزش همسایه.  $T$ : زمان.

الگوریتم SA به صورت زیر می‌باشد [۴۱]:

۱. در ابتدا یک دمای اولیه و یک راه حل اولیه تولید می‌شود.

۲. مراحل ۳ تا ۵ را تا هنگامی که پارامتر دما تا حد کافی کم شود (سرد شود) تکرار می‌کند:

۳. یک تغییر کم در راه‌حل فعلی می‌دهد تا به یک حالت دیگر که در همسایگی این حالت است، برسد.

۴. اگر حالت جدید بهتر بود که قبول می‌کند و اگر بهتر نبود با توجه به فاکتور احتمالی بولتزمن تصمیم می‌گیرد که این حالت جدید را قبول کند یا خیر.

۵. در نهایت کمی پارامتر دما را کاهش می‌دهد و دوباره از مرحله‌ی ۳ شروع می‌کند.

### ۳-۲-۳- الگوریتم TS

الگوریتم TS، یک الگوریتم بهینه‌سازی فرااکتشافی است که برای اولین بار در سال ۱۹۸۶ توسط گلوور<sup>۷۳</sup> معرفی شد [۴۲]. این الگوریتم یک لیست از حرکات یا نقاط ممنوعه درست می‌کند تا در جستجوهای بعدی، دیگر آن حرکات را انجام ندهد. با این کار امید دارد که از بهینه‌ی محلی خارج شده و بتواند به سمت بهینه‌ی سراسری حرکت کند. جابه‌جایی از جواب جاری به جواب همسایه‌ی امکان‌پذیر، زمانی انجام می‌شود که در جدول قرار نداشته باشد. در غیر اینصورت، جواب همسایه دیگری که در ارزیابی جواب‌های همسایه در رده بعدی قرار گرفته است، انتخاب شده و جابه‌جایی به آن صورت می‌گیرد. جستجوی ممنوعه، دو عنصر اصلی دارد. لیست ممنوعه<sup>۷۴</sup> و شرایط تنفس<sup>۷۵</sup> (یعنی شرایطی که به خاطر خوب بودن زیاد، شما قانون لیست ممنوعه را نقض می‌کنید) که تحقیقات نشان داده است با استفاده از این دو قاعده‌ی ساده، می‌تواند به نتایج خوبی دست پیدا کند.

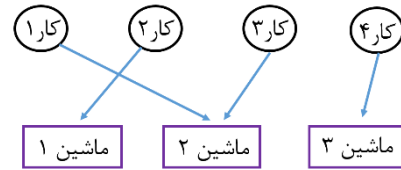
الگوریتم TS به صورت زیر می‌باشد [۴۲]:

۱. ابتدا از یک جواب اولیه شروع به حرکت می‌کند.

جدول ۲- نمادها و تعاریف

|              |  |
|--------------|--|
| $N$          | تعداد کارها                                      |
| $st_i$       | اندازه‌ی کار نام                                 |
| $ptt_i$      | زمان پردازش کار نام                              |
| $T_i$        | میزان دیرکرد از موعد تحویل کار نام               |
| $d_i$        | موعد تحویل کار نام                               |
| $CTT_i$      | زمان اتمام کار نام                               |
| $lateness_i$ | تاخیر کار نام                                    |
| $S_0$        | زمان آماده‌سازی اولیه هر یک از کارها             |
| $S_{ab}$     | زمان آماده‌سازی بین کارها (ماتریس $N \times N$ ) |
| $M$          | تعداد ماشین‌ها                                   |
| $pr_j$       | نرخ پردازش ماشین نام                             |
| $CTM_j$      | زمان اتمام کار روی ماشین نام                     |
| $std_j$      | انحراف از میانگین اتمام کار روی ماشین نام        |
| $et_{i,j}$   | اندازه‌ی کار نام به توان پردازشی ماشین نام       |

با فرض اینکه تعداد کارها ۴ و تعداد ماشین‌ها ۳ عدد باشد، می‌توان یک نگاشت از کارها و ماشین‌ها بدین صورت در نظر گرفت:  $x = \{2,1,2,3\}$  بدین معنی که کار اول روی ماشین شماره‌ی ۲، کار دوم روی ماشین شماره‌ی ۱، کار سوم روی ماشین شماره‌ی ۲ و کار چهارم روی ماشین شماره‌ی ۳، اجرا می‌شوند (شکل ۷).



شکل ۷- نمونه‌ای از تخصیص کارها به ماشین‌ها

در ابتدا تعداد کارها و تعداد ماشین‌ها را معین کرده و پس از آن اندازه و زمان اجرای کارها و همچنین قدرت پردازشی ماشین‌ها مشخص می‌شود. برای حل مسئله‌ی زمان‌بندی کار با استفاده از الگوریتم‌های فرااکتشافی، نیاز به ایجاد راه‌حل‌های تصادفی است بنابراین مانند بردار  $x$  که قبلاً توصیف شد، نمونه‌هایی ایجاد کرده و به عنوان راه‌حل مقدماتی به الگوریتم‌های فرااکتشافی داده و سعی می‌شود بر اساس تابع هدف، پارامترهای مورد نظر بهینه گردند.

حداقل‌سازی زمان انجام آخرین کار و در نتیجه آزاد شدن سریعتر منابع، جلوگیری از انباشت کارها روی یک ماشین و دریافت خدمت توسط مشتری با کمترین تاخیر، ویژگی‌هایی است که یک ارائه‌دهنده‌ی خدمات ابری را متمایز می‌کند. با توجه به اینکه در نظر گرفتن توامان این پارامترها به عنوان یک تابع هدف، می‌تواند به رضایت‌مندی مشتری و ارتقاء سطح خدمات سرویس‌دهنده کمک کند، لذا در این مقاله، زمان اتمام کارها، تعادل بار روی ماشین‌ها و میزان دیرکرد از موعد تحویل، به عنوان تابع هدف، در نظر گرفته می‌شوند.

طبق رابطه ۱۵، در این مقاله تابع هدف به صورت ترکیب خطی و وزن‌دار کمینه‌ی زمان اتمام کارها، بیشینه‌ی تعادل بار روی ماشین‌ها و کمینه‌ی میزان دیرکرد از موعد تحویل، تعریف شده است.

$$Min F(x) = w_1 Z1 + w_2 Z2 + w_3 Z3 \quad (15)$$

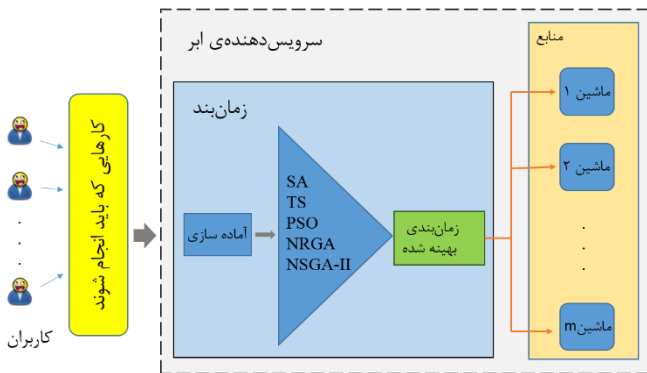
$$Z1 : Min f_1(x) = Max \{CTM_j\} \quad (16)$$

$$Z2 : Min f_2(x) = Max \{std_j\} \quad (17)$$

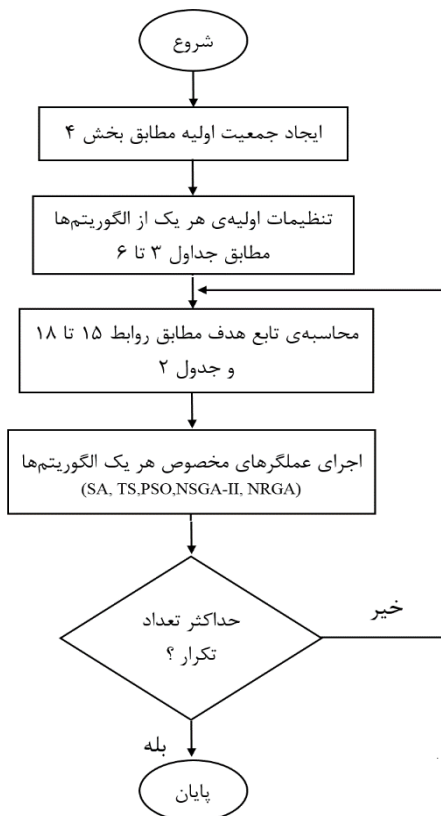
$$Z3 : Min f_3(x) = Max \{T_i\} \quad (18)$$

پارامترهایی که در روابط ۴ تا ۱۸ وجود دارند، در جدول ۲، توصیف شده‌اند. الگوریتم‌های بهینه‌سازی تک‌هدفه متنوعی (مبتنی بر انسان مانند الگوریتم TS، مبتنی بر فیزیک مانند الگوریتم SA، مبتنی بر جمعیت مانند الگوریتم PSO) را در نظر گرفته و در شرایط مساوی با یکدیگر مقایسه شده‌اند. همچنین برای تعیین قدرت و توانایی این الگوریتم‌ها، از الگوریتم‌های چندهدفه هم استفاده گردید. تنوع استفاده از الگوریتم‌ها باعث می‌شود که تصمیم‌گیری برای انتخاب الگوریتم مناسب، راحت‌تر شده و با توجه به اینکه کدام تابع هدف (روابط ۱۶ تا ۱۸) برای کاربر مهم‌تر است، می‌توان الگوریتم مناسب را انتخاب کرد.

در شکل ۸ معماری سیستم زمان‌بند نمایش داده شده است. در این معماری کاربران تقاضای خود را برای محیط ابر ارسال کرده و منتظر اعلام نتیجه می‌مانند. نحوه‌ی کار زمان‌بند در شکل ۹ قابل مشاهده است. در محیط ابر، زمان‌بند موظف است بر اساس شاخص‌هایی که در تابع هدف در نظر گرفته شده، ماشین مناسب را انتخاب کرده و کارها را برای ماشین‌ها ارسال کند. در پایان این فرایند، با استفاده از الگوریتم‌های بهینه‌سازی موجود، زمان‌بندی بهینه استخراج شده و مشخص می‌شود که هر کار می‌بایست روی کدام ماشین و با چه ترتیبی اجرا شود. بنابراین بخش منابع در سرویس‌دهنده‌ی ابر بر اساس این ترتیب، منابع را در اختیار کارها قرار می‌دهد.



شکل ۸- معماری سیستم زمان‌بند استفاده شده



شکل ۹- روند مربوط به کار زمان‌بند

## ۵- شبیه‌سازی و ارائه‌ی نتایج

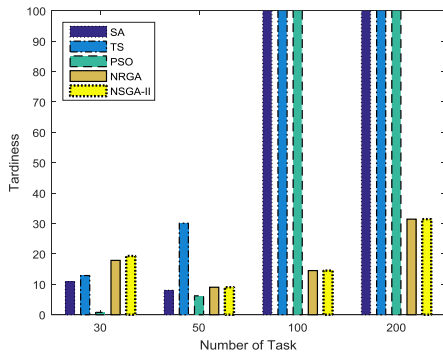
برای شبیه‌سازی از نرم‌افزار متلب استفاده شده است. هر یک از الگوریتم‌ها ۱۰ مرتبه اجرا و از نتایج میانگین گرفته شده و در این ۱۰ مرتبه اجرا، بدترین نتیجه و بهترین نتیجه نیز در جداول آورده شده است. در ابتدا چندین مدل ایجاد می‌گردد که در همه‌ی مدل‌ها تعداد ماشین‌ها ۲۰ ولی تعداد کارها ۵۰، ۱۰۰ و ۲۰۰ می‌باشند. در شکل‌های ۱۰ تا ۱۲، میانگین نتایج الگوریتم‌ها، در نظر گرفته شده است.

میزان تاخیر از موعد تحویل، پارامتری است که در این مقاله به تابع هدف اضافه شد. طبیعتاً هر چه این مقدار کمتر باشد، زمان پاسخ‌گویی کمتر بوده و رضایت‌مندی کاربران ابر، بیشتر می‌شود. در بین الگوریتم‌های استفاده شده و برای تعداد کار کم (به عنوان مثال ۵۰ کار) الگوریتم PSO با کمترین میزان تاخیر توانسته نتایج بهتری نسبت به بقیه‌ی الگوریتم‌ها داشته باشد (شکل ۱۰). از فاکتورهای مهم الگوریتم PSO، سرعت و سادگی آن می‌باشد که این نتیجه را در برداشته است. اما در تعداد کار زیاد، الگوریتم‌هایی که مختص توابع چندهدفه هستند (NSGA-II و NPGA) توانسته‌اند با تاخیر کمتر زمان‌بندی را انجام دهند.

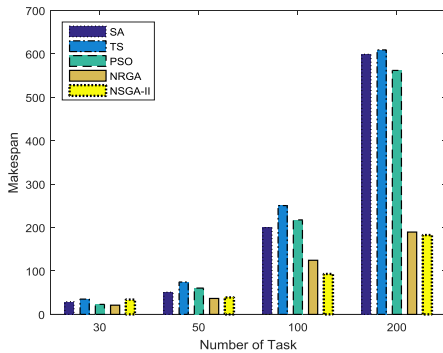
در شکل ۱۱ به مقایسه‌ی الگوریتم‌ها از نظر زمان اتمام کارها، پرداخته شده است. در تعداد کار کم، تقریباً الگوریتم‌ها عملکرد مشابهی داشته‌اند. با افزایش تعداد کارها، الگوریتم‌های NPGA و پس از آن NSGA-II توانسته‌اند کارها را در زمان کمتری به اتمام برسانند. ماهیت الگوریتم‌های چندهدفه ایجاب می‌کند که در فضای جستجو نقاط بهینه‌تری را برای توابع هدف و بصورت تک‌تک، بدست آورند. در شکل ۱۲ و تقریباً در تمام نتایج، الگوریتم جستجوی ممنوعه عملکرد ضعیفی داشته و نتوانسته نقاط مناسب را در فضای جستجو پیدا کند. ماهیت این الگوریتم شبیه روش‌های دقیق بوده و به همین دلیل مقداری زمان‌بر و در تعداد تکرار کم (۱۰۰ تکرار) به جواب مطلوب نرسیده است.

جدول ۶- پارامترهای قابل تنظیم NSGA-II

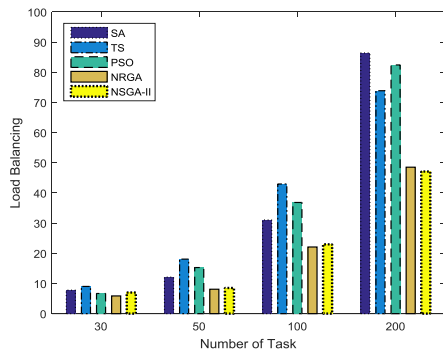
|             |     |
|-------------|-----|
| تعداد جمعیت | ۵۰  |
| تعداد تکرار | ۱۰۰ |
| نرخ تقاطع   | ۰/۸ |
| نرخ جهش     | ۰/۳ |



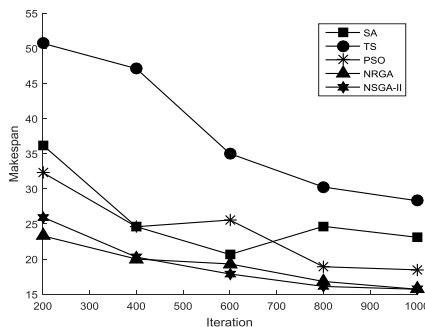
شکل ۱۰- مقایسه‌ی الگوریتم‌ها بر اساس میزان تاخیر از موعد تحویل



شکل ۱۱- مقایسه‌ی الگوریتم‌ها بر اساس زمان اتمام کارها



شکل ۱۲- مقایسه‌ی الگوریتم‌ها بر اساس تعادل بار روی ماشین‌ها



شکل ۱۳- مقایسه‌ی الگوریتم‌ها بر اساس تعداد تکرار و زمان اتمام کارها (برای ۳۰ کار و ۲۰ ماشین)

کار و ۲۰ ماشین)

جدول ۳- پارامترهای قابل تنظیم الگوریتم SA

|                       |      |
|-----------------------|------|
| تعداد جمعیت           | ۵    |
| تعداد تکرار           | ۱۰۰  |
| تعداد تکرار در هر دما | ۱۰   |
| $T_0$                 | ۱۰   |
| TF                    | ۰/۰۱ |

جدول ۴- پارامترهای قابل تنظیم الگوریتم PSO

|                           |                         |
|---------------------------|-------------------------|
| تعداد جمعیت               | ۵۰                      |
| تعداد تکرار               | ۱۰۰                     |
| $P1$                      | ۲/۰۵                    |
| $P2$                      | ۲/۰۵                    |
| $P$                       | $P1+P2$                 |
| $C$                       | $2/(P-2+\sqrt{P^2-4P})$ |
| وزن اینرسی                | ۰/۳                     |
| نرخ تعدیل وزن اینرسی      | $C$                     |
| ضریب یادگیری اجتماعی $c1$ | ۰/۱                     |
| ضریب یادگیری شخصی $c2$    | $C \times P1$           |
| سرعت ذرات                 | $C \times P2$           |
|                           | Range of VMs            |

جدول ۵- پارامترهای قابل تنظیم الگوریتم TS

|              |                               |
|--------------|-------------------------------|
| تعداد مبادله | $N \times (N-1)/2$            |
| تعداد برگشت  | $N \times (N-1)/2$            |
| تعداد درج    | $N \times (N-1)$              |
| تعداد عملیات | nSwap+ nReversion+ nInsertion |
| تعداد تکرار  | ۱۰۰                           |

الگوریتم TS به این شکل قابل توجه است که با پیچیده‌تر شدن مسئله، این الگوریتم به زمان بیشتری برای رسیدن به جواب بهینه نیاز دارد. مطابق شکل ۱۳ در تکرار بیشتر این الگوریتم، نتایج بهتری حاصل می‌شود.

جدول ۹ و ۱۰ از نظر عملکرد الگوریتم‌ها مشابه بوده و تنها نکته قابل ذکر، عملکرد ضعیف الگوریتم PSO در تعداد ۲۰۰ کار به نسبت ۱۰۰ کار می‌باشد. به دلیل پیچیده‌تر شدن مسئله این الگوریتم نیاز به تعداد جمعیت اولیه‌ی بیشتری برای رسیدن به جواب مطلوب، دارد. الگوریتم‌های چندهدفه، چه از نظر زمان اجرا، چه از نظر بهینگی تابع هدف، همچنان به نسبت بقیه‌ی الگوریتم‌ها عملکرد بهتری دارند.

به طور خلاصه، برای مسئله‌ی زمان‌بندی کار در محیط ابر، در ابتدا تعداد ۲۰ ماشین با توان پردازشی متفاوت و تعدادی کار با اندازه‌های متفاوت در نظر گرفته شد. هدف کمینه‌کردن زمان اتمام کارها، تعادل بار بین ماشین‌ها و در نهایت کمینه کردن تاخیر انجام کارها بود. جزئیات این توابع هدف در جدول ۲ قابل مشاهده است. در این راستا سه الگوریتم SA، TS و PSO که جزء الگوریتم‌های تک‌هدفه می‌باشند به کار گرفته شد. تابع هدفی که برای این الگوریتم‌ها تعریف شد به صورت جمع وزن دار سه مورد مذکور می‌باشد. همچنین برای مقایسه‌ی بهتر، از دو الگوریتم NSGA-II و NPGA که تفاوت اندکی با هم دارند، استفاده شد. این الگوریتم‌ها خاص مسائل چندهدفه می‌باشند. پارمترهای قابل تنظیم برای الگوریتم‌های مذکور در جدول ۳ تا ۶ و همچنین نتایج شبیه‌سازی در جدول ۷ تا ۱۰ برای ۲۰ ماشین و به ترتیب ۳۰، ۵۰، ۱۰۰ و ۲۰۰ کار قابل مشاهده است. ستون میانگین زمان اجرا در همه‌ی جداول، مربوط به میانگین زمان اجرای هر الگوریتم به ازای ۱۰ اجرا، می‌باشد. نتایج این ستون نشان می‌دهد که بین الگوریتم‌های تک‌هدفه الگوریتم PSO در زمان بسیار کمتری اجرا شده و حتی برای تعداد ۳۰ کار (جدول ۷) از نظر بهینه‌سازی تابع هدف، قابل رقابت با الگوریتم‌های چندهدفه می‌باشد. اما همانطور که انتظار می‌رفت الگوریتم‌های NSGA-II و NPGA نتایج بهتری به نسبت الگوریتم‌های تک‌هدفه داشته و بهتر توانسته‌اند اهداف مسئله را بهینه کنند. از آنجایی که الگوریتم TS شبیه روش‌های دقیق عمل می‌کند، بنابراین برای رسیدن به جواب بهینه نیاز به تعداد تکرار زیاد دارد. با توجه به اینکه زمان سرویس‌دهی در محیط ابر بسیار مهم است، این الگوریتم برای حل مسئله‌ی زمان‌بندی کار، گزینه‌ی مناسبی نیست. همچنین می‌توان پارمترهای قابل تنظیم الگوریتم مورد استفاده را با کمک ابزارهایی مانند سیستم فازی، جهت بالا بردن کارایی، بهینه کرد.

با توجه به شکل ۱۳، تعداد تکرار مراحل الگوریتم برای همه‌ی الگوریتم‌ها پارامتر موثری بوده و در نتیجه‌ی کار تأثیرگذار است. در شکل ۱۳ عملکرد مناسب الگوریتم TS مشهود است. این الگوریتم در تکرار ۲۰۰ با زمان اتمام کارها برابر ۵۲ مواجه است، اما در تکرار ۱۰۰۰ این زمان را به ۲۷ کاهش داده و از این نظر بیشترین کاهش را به نسبت بقیه‌ی الگوریتم‌ها داشته است.

جدول ۷ نتایج الگوریتم‌ها را برای زمان‌بندی ۳۰ کار نمایش می‌دهد. در این تعداد کار، الگوریتم PSO برای تابع هدف Z1 با میانگین ۲۳/۱۲، برای تابع هدف Z2 با میانگین ۶/۶۸ و برای تابع هدف Z3 با میانگین ۰/۷۷ توانسته در زمان ۱۰ ثانیه به جواب برسد. با این تفاسیر الگوریتم PSO به نسبت بقیه نتایج بهتری داشته و می‌تواند برای درخواست‌های آبی و دارای اولویت، گزینه‌ی مناسبی باشد. در این تعداد کار، الگوریتم SA با زمان ۱۷۰ ثانیه بدترین زمان و الگوریتم TS از نظر بهینه‌سازی توابع هدف بدترین عملکرد را داشته اند. بین الگوریتم‌های چندهدفه، الگوریتم NPGA از نظر بهینه‌سازی توابع هدف بهترین عملکرد را داشته و با از دست دادن مقداری زمان به نسبت الگوریتم NSGA-II کارایی بالاتری دارد. با توجه به نتایج الگوریتم‌ها برای زمان‌بندی تعداد کار کم، الگوریتم‌های تک‌هدفه (به‌خصوص PSO) کارایی خوبی داشته و می‌توانند جایگزین مناسبی برای الگوریتم‌های چندهدفه باشند. جدول ۸ نتایج الگوریتم‌ها را برای زمان‌بندی ۵۰ کار نمایش می‌دهد. در این تعداد کار، NPGA برای تابع هدف Z1 با میانگین ۳۶/۷۴، برای تابع هدف Z2 با میانگین ۸/۰۹ و برای تابع هدف Z3 با میانگین ۹/۰۴ توانسته در زمان ۱۳ ثانیه به جواب برسد. با این تفاسیر NPGA در مجموع سه تابع هدف، به نسبت الگوریتم‌های SA، TS، ۶۸ درصد بهبود نتایج داشته و می‌تواند برای این تعداد کار گزینه‌ی مناسبی باشد. با پیچیده‌تر شدن مسئله، نیاز به الگوریتم‌های چندهدفه، بیشتر احساس می‌شود، چرا که این الگوریتم‌ها در بهینه‌سازی توابع هدف، توانسته‌اند با عملکردی مشابه با یکدیگر، به نتایج خوبی دست پیدا کنند. اما اگر مدت زمان مورد نیاز برای زمان‌بندی نسبت به بهینگی توابع هدف، اهمیت بیشتری داشته باشد، الگوریتم PSO گزینه‌ی مناسب‌تری می‌باشد.

جدول ۹ نتایج الگوریتم‌ها را برای زمان‌بندی ۱۰۰ کار نمایش می‌دهد. در این تعداد کار، NPGA برای تابع هدف Z1 با میانگین ۸۸/۹۹، برای تابع هدف Z2 با میانگین ۲۲/۰۹ و برای تابع هدف Z3 با میانگین ۱۴/۵۲ توانسته در زمان ۱۳ ثانیه به جواب برسد. نتایج NSGA-II هم شبیه NPGA بوده و تفاوت چندانی ندارند. اما اگر مدت زمان مورد نیاز برای زمان‌بندی نسبت به بهینگی تابع هدف، اهمیت بیشتری داشته باشد، باز هم الگوریتم PSO گزینه‌ی مناسب‌تری می‌باشد. عملکرد ضعیف

جدول ۷- مقایسه‌ی الگوریتم‌ها برای زمان‌بندی ۳۰ کار روی ۲۰ ماشین بر اساس سه تابع هدف.

|         | Z1     |        |         | Z2     |        |         | Z3     |        |         | میانگین زمان اجرا |
|---------|--------|--------|---------|--------|--------|---------|--------|--------|---------|-------------------|
|         | بدترین | بهترین | میانگین | بدترین | بهترین | میانگین | بدترین | بهترین | میانگین |                   |
| SA      | ۳۶/۲۲  | ۲۳/۱۵  | ۲۹/۴۰   | ۹/۹۵   | ۶/۱۷   | ۷/۷۸    | ۱۸/۱۶  | ۳/۷۵   | ۱۰/۹۰   | ۱۷۰               |
| TS      | ۵۰/۷   | ۲۸/۰۵  | ۳۵/۲۴   | ۱۲/۹۹  | ۷/۲۶   | ۹/۰۲    | ۳۲/۵   | ۲/۴    | ۱۲/۸۹   | ۱۵۰               |
| PSO     | ۳۲/۳   | ۱۲/۳۵  | ۲۳/۱۲   | ۹/۲۱   | ۴/۰۸   | ۶/۶۸    | ۴/۳۵   | ۰      | ۰/۷۷    | ۱۰                |
| NRGA    | ۳۷/۳   | ۱۶/۲۵  | ۲۱/۲۲   | ۷/۹۴   | ۳/۳۸   | ۵/۸۷    | ۵۶/۶۵  | ۵/۰۵   | ۱۷/۸۸   | ۷۰                |
| NSGA-II | ۶۴/۹۶  | ۱۹/۰۷  | ۳۴/۰۵   | ۱۲/۱۵  | ۴/۲۱   | ۷/۰۰    | ۶۱/۱۵  | ۵/۵۵   | ۱۹/۳۲   | ۵۹                |

جدول ۸- مقایسه‌ی الگوریتم‌ها برای زمان‌بندی ۵۰ کار روی ۲۰ ماشین بر اساس سه تابع هدف.

|         | Z1     |        |         | Z2     |        |         | Z3     |        |         | میانگین زمان اجرا |
|---------|--------|--------|---------|--------|--------|---------|--------|--------|---------|-------------------|
|         | بدترین | بهترین | میانگین | بدترین | بهترین | میانگین | بدترین | بهترین | میانگین |                   |
| SA      | ۵۵/۹۰  | ۴۵/۰۵  | ۵۱/۲۱   | ۱۳/۳۴  | ۱۰/۴۴  | ۱۲/۱۱   | ۱۵/۷۵  | ۰/۶    | ۸/۰۴    | ۴۸۲               |
| TS      | ۱۲۲/۷۴ | ۵۵/۴   | ۷۴/۵۷   | ۲۷/۱۹  | ۱۳/۸۰  | ۱۸/۰۷   | ۵۷/۵۵  | ۲/۷۵   | ۳۰/۱۲   | ۱۰۳               |
| PSO     | ۹۱/۳۵  | ۴۰/۷   | ۶۰/۶۲   | ۲۰/۵۱  | ۱۰/۴۵  | ۱۵/۲۸   | ۱۶/۰۵  | ۰      | ۶/۲۲    | ۱۱                |
| NRGA    | ۴۲/۸۵  | ۳۳/۱   | ۳۶/۷۴   | ۸/۸۴   | ۷/۳۹   | ۸/۰۹    | ۹/۰۴   | ۹/۰۴   | ۹/۰۴    | ۱۳                |
| NSGA-II | ۴۳/۲   | ۳۵/۰۵  | ۳۹/۰۵   | ۹/۴۶   | ۷/۵۸   | ۸/۵۱    | ۹/۰۴   | ۹/۰۴   | ۹/۰۴    | ۱۳                |

جدول ۹- مقایسه‌ی الگوریتم‌ها برای زمان‌بندی ۱۰۰ کار روی ۲۰ ماشین بر اساس سه تابع هدف.

|         | Z1     |        |         | Z2     |        |         | Z3      |        |         | میانگین زمان اجرا |
|---------|--------|--------|---------|--------|--------|---------|---------|--------|---------|-------------------|
|         | بدترین | بهترین | میانگین | بدترین | بهترین | میانگین | بدترین  | بهترین | میانگین |                   |
| SA      | ۲۰۹/۳۳ | ۹۷/۱۳  | ۱۹۹/۹۸  | ۳۵/۴۳  | ۲۲/۲۸  | ۳۱/۰۵   | ۲۲۶۴/۶۵ | ۲۰۵۵/۱ | ۲۱۴۵/۶  | ۱۷۱               |
| TS      | ۳۳۵/۹۱ | ۱۵۲/۶  | ۲۵۰/۶۶  | ۵۱/۶۵  | ۲۹/۳۷  | ۴۲/۹۷   | ۲۶۸۰/۸۵ | ۲۳۴۱/۲ | ۲۴۳۶/۲۱ | ۶۳۲               |
| PSO     | ۳۴۹/۸۶ | ۱۵۳/۰۵ | ۲۱۷/۶۸  | ۴۶/۷۱  | ۳۲/۱۶  | ۳۶/۸۵   | ۹۷۴/۷   | ۵۲۴/۲  | ۸۰۷/۷۸  | ۱۴                |
| NRGA    | ۱۰۳/۴۱ | ۸۱/۴   | ۸۸/۹۹   | ۲۴/۵۱  | ۲۰/۲۳  | ۲۲/۰۹   | ۱۴/۵۳   | ۱۴/۵۱  | ۱۴/۵۲   | ۱۳                |
| NSGA-II | ۱۰۲/۶۲ | ۸۲/۹۵  | ۹۳/۱۵   | ۲۴/۷۱  | ۲۱/۱۵  | ۲۲/۹۷   | ۱۴/۵۳   | ۱۴/۵۱  | ۱۴/۵۲   | ۱۴                |

جدول ۱۰- مقایسه‌ی الگوریتم‌ها برای زمان‌بندی ۲۰۰ کار روی ۲۰ ماشین بر اساس سه تابع هدف.

|         | Z1     |        |         | Z2     |        |         | Z3      |         |         | میانگین زمان اجرا |
|---------|--------|--------|---------|--------|--------|---------|---------|---------|---------|-------------------|
|         | بدترین | بهترین | میانگین | بدترین | بهترین | میانگین | بدترین  | بهترین  | میانگین |                   |
| SA      | ۸۰۹/۴۵ | ۳۴۸/۸۱ | ۵۹۹/۱۵  | ۱۲۷/۰۲ | ۵۱/۸۰  | ۸۶/۴۲   | ۶۲۵۸/۳  | ۵۷۳۳/۲  | ۵۸۸۱/۵۸ | ۲۵۶               |
| TS      | ۸۲۶/۹۵ | ۴۶۷/۳۲ | ۶۰۹/۲۱  | ۱۲۵/۸۹ | ۴۸/۵۹  | ۷۳/۸۹   | ۶۸۶۰/۷  | ۵۹۸۲/۱  | ۶۰۰۷/۳۶ | ۳۸۸۳              |
| PSO     | ۸۷۹/۸۹ | ۴۰۲/۵۱ | ۵۶۱/۹۵  | ۹۶/۸۱  | ۷۰/۶۴  | ۸۲/۴۳   | ۳۴۱۰/۰۵ | ۲۴۸۲/۷۵ | ۳۰۷۱/۴۴ | ۲۰                |
| NRGA    | ۲۳۳/۰۵ | ۱۴۶/۷۵ | ۱۸۹/۶۸  | ۵۵/۸۲  | ۴۰/۲۲  | ۴۸/۵۴   | ۳۱/۴۴   | ۳۱/۴۴   | ۳۱/۴۴   | ۱۴                |
| NSGA-II | ۲۱۳/۴۶ | ۱۵۲/۵۳ | ۱۸۲/۷۷  | ۵۱/۷۳  | ۴۱/۱۱  | ۴۷/۱۱   | ۳۱/۴۴   | ۳۱/۴۴   | ۳۱/۴۴   | ۱۴                |

- [2] D. Mrozek, "A Review of cloud computing technologies for comprehensive microRNA analyses," *Computational Biology and Chemistry*, 2020.
- [3] M. Lavanya, B. Shanthi and S. Saravanan, "Multi objective task scheduling algorithm based on SLA and processing time suitable for cloud environment," *Computer Communications*, Vol. 67, 2020.
- [4] N. Mansouri and M. M. Javidi, "Cost-based job scheduling strategy in cloud computing environments," *Distributed and Parallel Databases*, Vol. 18, pp. 365-400, 2020.
- [5] H. Emmons, "One-machine sequencing to minimize certain functions of job tardiness," *Operations Research*, Vol. 17, pp. 701-715, 1969.
- [6] E.L. Lawler, "Efficient implementation of dynamic programming algorithms for sequencing problems," *Report BW 106, Mathematisch Centrum, Amsterdam*, 1979,
- [7] M.L. Fisher, "A dual algorithm for the one machine scheduling problem," *Mathematical Programming*, Vol. 11, pp. 229-252, 1976.
- [8] C.N. Potts and L.N. Van Wassenhove, "A branch and bound algorithm for the total weighted tardiness problem," *Operations Research*, Vol. 33, pp. 177-181, 1985.
- [9] M.S. Akturk and M.B. Yildirim, "A new lower bounding scheme for the total weighted tardiness problem," *Computers and Operations Research*, Vol. 25, pp. 265-278, 1998.
- [10] W. Bozejko, J. Grabowski and M. Wodecki, "Block approach-tabu search algorithm for single machine total weighted tardiness problem," *Computers & Industrial Engineering*, Vol. 50, pp. 1-14, 2006.
- [11] M. Srinivas and L.M. Patnaik, "Genetic algorithms: a survey," *Computer*, Vol. 27, pp. 17-26, 1994.
- [12] T. Kellegoz, B. Toklu and J. Wilson, "Comparing efficiencies of genetic crossover operators for one machine total weighted tardiness problem," *Elsesier Applied Mathematics and Computation*, Vol. 199, pp. 590-598, 2008.
- [13] Y. Xing, Zh. Chen and J. Sun, "An improved adaptive genetic algorithm for job-shop scheduling problem," *Third International Conference on Natural Computation (IEEE ICNC 2007)*, 2007.
- [14] Y. Xing and Zh.Wang, "An improved genetic algorithm with recurrent Search for the job-shop scheduling problem," *The 6th World Congress on Intelligent Control and Automation IEEE*, 2006.
- [15] S.M. Kamrul Hasan, R. Sarker and D. Cornforth, "Hybrid genetic algorithm for solving job-shop scheduling problem," *6th IEEE/ACIS International Conference on Computer and Information Science (IEEE ICIS 2007)*, 2007.
- [16] Y. Tsujimura, Y. Mafune and M. Gen, "Effects of symbiotic evolution in genetic algorithms for job-shop scheduling," *Proceedings of the 34th Hawaii International Conference on System Sciences IEEE*, 2001.
- [17] A. Tariq, I. Hussain and A. Ghafoor, "A hybrid genetic algorithm for job shop scheduling," *Proceedings of the 37th International*

## ۶- نتیجه‌گیری و بحث

یکی از چالش‌های مهم در فناوری ابر، تخصیص بهینه‌ی منابع به کارهاست. زمان‌بندی بهینه تاثیر مستقیم بر رضایت‌مندی کاربر دارد چرا که به دریافت خدمت در زمان مناسب و با سرعت بالا، منجر می‌شود. اما مسئله‌ی زمان‌بندی کار جزء مسائل ان‌پی-سخت است. به دلیل ماهیت ابر و تعداد کاربران، فضای جستجو برای پیدا کردن راه‌حل بهینه، بسیار بزرگ و تخصیص منابع به کارها بسیار زمان‌بر می‌شود. در ارزیابی پارامترهایی مانند زمان اتمام کارها، موعد تحویل کارها، تاخیر در تحویل کارها، هزینه‌ی انجام کارها و تعادل بار روی ماشین‌ها، این نتیجه حاصل شد که پارامترهای زمان اتمام، تاخیر در تحویل و تعادل بار روی ماشین‌ها به شکل توأمان، از اهمیت بالاتری برخوردار بوده و در نتیجه تابع هدفی که نوشته شده بر اساس این سه پارامتر، است. برای بهینه‌سازی تابع هدف از الگوریتم‌های بهینه‌سازی متنوعی مانند الگوریتم PSO (مبتنی بر جمعیت)، GA (مبتنی بر تکامل)، الگوریتم SA (مبتنی بر قوانین فیزیک) و الگوریتم TS (مبتنی بر انسان)، استفاده شد و شبیه‌سازی در محیط نرم‌افزار متلب صورت گرفت. در تعداد کار کم و با از دست دادن مقداری کارایی، الگوریتم PSO از نظر سرعت و سادگی بهترین انتخاب است، زیرا با سرعت بسیار بالاتری (به طور میانگین ۱۶ برابر) به نسبت الگوریتم‌های SA و TS، اجرا شده است. اما در تعداد زیاد کار (۲۰۰ کار) و زمانی که پیچیدگی مسئله بیشتر می‌شود، الگوریتم‌های چندهدفه (NSGA-II و NRGA)، هم از نظر زمان اجرا (به عنوان مثال NSGA-II، ۱۸ بار سریع‌تر از الگوریتم SA و ۲۷۷ بار سریع‌تر از الگوریتم TS اجرا شده است) و هم بهینه‌کردن اهداف مسئله (به عنوان مثال NSGA-II، تابع هدف را ۹۶/۰۴ درصد کمتر از الگوریتم SA و ۹۶/۱۱ درصد کمتر از الگوریتم TS تخمین زده است)، نتایج بهتری به نسبت الگوریتم‌های تک‌هدفه داشته‌اند.

از آنجایی که در الگوریتم‌های بهینه‌سازی، نقطه‌ی شروع و جستجوی همه‌جانبه‌ی فضای جستجو، در نتیجه‌ی نهایی بسیار موثر است، پیشنهاد می‌شود از تئوری آشوب<sup>۷۹</sup> برای ایجاد تنوع در جمعیت استفاده شود.

## ۷- مراجع

- [1] P. Helo, Y. Hao, R. Toshev and V. Boldosova, "Cloud manufacturing ecosystem analysis and design," *Robotics and Computer Integrated Manufacturing*, 2021.

- [39] J. Kennedy and R. C. Eberhart, "Particle Swarm Optimization," IEEE International Conference on Neural Networks (Perth, Australia), IEEE Service Center, Piscataway, NJ, 1995, Vol. IV, 1942-1948.
- [40] S. Kirkpatrick, C. D. Gelatt and M. P. Vecchi, "Optimization by Simulated Annealing," Science, New Series, Vol. 220, 671-680, 1983.
- [41] V. Cerny, "Thermodynamical approach to the traveling salesman problem: An efficient simulation algorithm," journal of optimization theory and applications, Vol. 45, pp. 41-51, 1953.
- [42] F. Glover, "Future paths for integer programming and links to artificial intelligence," Computers and Operations Research, Vol. 13, pp. 533-549, 1986.
- [43] K. Deb, "A fast and elitist multiobjective genetic algorithm: NSGA-II," IEEE transactions on evolutionary computation, Vol. 6, pp. 182-197, 2002.

### ابوذر زندوکیلی تحصیلات خود را در مقطع کارشناسی،

رشته‌ی علوم کامپیوتر از دانشگاه شهید باهنر کرمان به تمام رسانده و در حال حاضر در مقطع کارشناسی ارشد، رشته‌ی علوم کامپیوتر و در دانشگاه شهید باهنر کرمان، مشغول به تحصیل می‌باشد. زمینه‌های مورد علاقه‌ی ایشان عبارتند از: رایانش ابری، یادگیری ماشین و سیستم‌های فازی. آدرس پست الکترونیکی ایشان عبارت است از:



zandvakili.a@math.uk.ac.ir

### نجمه منصوری تحصیلات خود را در مقطع کارشناسی

رشته‌ی کامپیوتر از دانشگاه شهید باهنر کرمان، کارشناسی ارشد رشته‌ی مهندسی کامپیوتر از دانشگاه شیراز و مقطع دکترا رشته علوم کامپیوتر از دانشگاه شهید باهنر کرمان در سال ۱۳۹۸ به پایان رسانده و هم اکنون عضو هیات علمی دانشگاه شهید باهنر کرمان است. زمینه‌های مورد علاقه‌ی ایشان عبارتند از: رایانش ابری، سیستم‌های توزیع شده و الگوریتم‌های بهینه‌سازی. آدرس پست الکترونیکی ایشان عبارت است از:



n.mansouri@uk.ac.ir

### محمدمسعود جاویدی هم اکنون دانشیار بخش علوم

کامپیوتر در دانشگاه شهید باهنر کرمان است. زمینه‌های مورد علاقه‌ی ایشان عبارتند از: رایانش ابری، الگوریتم‌های بهینه‌سازی و کاهش ابعاد. آدرس پست الکترونیکی ایشان عبارت است از:



javidi@uk.ac.ir

- Conference on Computers and Industrial Engineering October 20-23, 2007.*
- [18] F. Hemasian-Etefagh and F. Safi-Esfahani, "Dynamic scheduling applying new population grouping of whales meta-heuristic in cloud computing," *The Journal of Supercomputing*, 2019.
- [19] Sh. Torabi and F. Safi-Esfahani, "A dynamic task scheduling framework based on chicken swarm and improved raven roosting optimization methods in cloud computing," *The Journal of Supercomputing*, 2018.
- [20] M. Agarwal and G. M. Saran Srivastava, "A cuckoo search algorithm-based task scheduling in cloud computing," *advances in intelligent systems and computing*, 2018.
- [21] J. Li and Y. Han, "A hybrid multi-objective artificial bee colony algorithm for flexible task scheduling problems in cloud computing system," *Cluster Computing*, 2018.
- [22] J. P. Buanga Mapetu, Zh. Chen and L. Kong, "Low-time complexity and low-cost binary particle swarm optimization algorithm for task scheduling and load balancing in cloud computing," *Applied Intelligence*, 2019.
- [23] A. M. Senthil Kumar and M. Venkatesan, "Task scheduling in a cloud computing environment using HGPSO algorithm," *Cluster Computing*, 2018.
- [24] X. Chen and D. Long, "Task scheduling of cloud computing using integrated particle swarm algorithm and ant colony algorithm," *Cluster Comput*, 2017.
- [25] S. J. Nirmala and S. M. S. Bhanu, "Catfish-PSO based scheduling of scientific workflows in IaaS cloud," *Computing*, 2016.
- [26] W. Chen, E. Deelman, "WorkflowSim: a toolkit for simulating scientific workflows in distributed environments," In: Proceedings of IEEE 8th international conference on e-Science (e-Science), 8-12 October 2012, Hyatt Regency Chicago, Chicago, Illinois, USA, pp 1-8, 2012.
- [27] R.N. Calheiros, R. Ranjan, A. Beloglazov, "CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms," *Software: Practice and Experience*, Vol. 41(1):pp. 23-50, 2011.
- [28] M. Kumar and S. C. Sharma, "PSO-based novel resource scheduling technique to improve QoS parameters in cloud computing," *Neural Computing and Applications*, Vo. 32, pp. 12103-12126, 2020.
- [29] Z. Zhou, F. Li, H. Zhu, H. Xie, J. H. Abawajy and M. U. Chowdhury, "An improved genetic algorithm using greedy strategy toward task scheduling optimization in cloud environments," *Neural Computing and Applications*, Vol. 32, pp. 1531-1541, 2020.
- [30] Y. Gu, Ch. Budati, "Energy-aware workflow scheduling and optimization in clouds using bat algorithm," *Future Generation Computer Systems*, Vol. 113, pp.106-112, 2020.
- [31] X. Chen, L. Cheng, C. Liu, Q. Liu, J. Liu, Y. Mao, J. Murphy, "A WOA-based optimization approach for task scheduling in cloud computing systems," *IEEE Systems Journal*, Vol. 14(3), 2020.
- [32] Zh. He, J. Dong, Zh. li, W. Guo, "Research on task scheduling strategy optimization based on ACO in cloud computing environment," *IEEE 5th Information Technology and Mechatronics Engineering Conference (ITOEC)*, 2020.
- [33] Z. Zong, "An improvement of task scheduling algorithms for green cloud computing," *The 15th International Conference on Computer Science & Education (ICCSE)*, 2020 August 18-20, 2020.
- [34] S. P. M. Ziyath and S. Senthilkumar, "MHO: Meta heuristic optimization applied task scheduling with load balancing technique for cloud infrastructure services," *Journal of Ambient Intelligence and Humanized Computing*, 2020.
- [35] A. S. Ajeena Beegom and M. S. Rajasree, "Integer-PSO: a discrete PSO algorithm for task scheduling in cloud computing systems," *Evolutionary Intelligence*, 2019.
- [36] Y. Dai, Y. Lou and X. Lu, "A task scheduling algorithm based on genetic algorithm and ant colony optimization algorithm with multi-QoS constraints in cloud computing," *7th International Conference on Intelligent Human-Machine Systems and Cybernetics*, 2015.
- [37] Sh. Su and H. Yu, "Authors Info & Affiliations Minimizing tardiness in data aggregation scheduling with due date consideration for single-hop wireless sensor networks," *Wireless Networks*, Vol. 21, pp. 1259-1273, 2015.
- [38] D. E. Akyol and G. M. Bayhan, "Multi-machine earliness and tardiness scheduling problem: an interconnected neural network approach," *The International Journal of Advanced Manufacturing Technology*, Vol. 37, pp. 576-588, 2008.

<sup>1</sup> Cloud computing

<sup>2</sup> Everything as a Service (XaaS)

<sup>3</sup> Infrastructure as a Service (IaaS)

<sup>4</sup> Platform as a Service (PaaS)

<sup>5</sup> Software as a Service (SaaS)

<sup>6</sup> Public cloud

<sup>7</sup> Private cloud

<sup>8</sup> Hybrid cloud

<sup>9</sup> Job scheduling

<sup>10</sup> Energy consumption

<sup>11</sup> Efficiency

<sup>12</sup> Waiting time

- 
- <sup>13</sup> Delay penalty
  - <sup>14</sup> Completion time
  - <sup>15</sup> Meta heuristic
  - <sup>16</sup> Classic
  - <sup>17</sup> Flexible
  - <sup>18</sup> Task
  - <sup>19</sup> Total
  - <sup>20</sup> Partial
  - <sup>21</sup> Single
  - <sup>22</sup> NP-hard
  - <sup>23</sup> Dynamic programming
  - <sup>24</sup> Branch and bound
  - <sup>25</sup> Tabu Search (TS)
  - <sup>26</sup> Genetic Algorithm (GA)
  - <sup>27</sup> Whale Optimization Algorithm (WOA)
  - <sup>28</sup> Execution time
  - <sup>29</sup> Response time
  - <sup>30</sup> Throughput
  - <sup>31</sup> Chicken Swarm Optimization (CSO)
  - <sup>32</sup> Raven Roosting Optimization (RRO)
  - <sup>33</sup> Cuckoo Search (CS)
  - <sup>34</sup> Artificial Bee Colony (ABC)
  - <sup>35</sup> Load balancing
  - <sup>36</sup> Tardiness
  - <sup>37</sup> Particle Swarm Optimization (PSO)
  - <sup>38</sup> Simulated Annealing (SA)
  - <sup>39</sup> Non-dominated sorting genetic algorithm (NSGA-II)
  - <sup>40</sup> Non-dominated ranking genetic algorithm (NRGA)
  - <sup>41</sup> Binary PSO (BPSO)
  - <sup>42</sup> Sigmoid
  - <sup>43</sup> Priority
  - <sup>44</sup> On-demand
  - <sup>45</sup> Ant Colony Optimization (ACO)
  - <sup>46</sup> Cat-fish algorithm
  - <sup>47</sup> Multi Objective Particle Swarm Optimization (MOPSO)
  - <sup>48</sup> Bat Algorithm (BA)
  - <sup>49</sup> Greedy algorithm
  - <sup>50</sup> CO<sub>2</sub>
  - <sup>51</sup> Energy Aware, Time, and Throughput Optimization heuristic (EATTO)
  - <sup>52</sup> Quality of Service (QoS)
  - <sup>53</sup> Random search
  - <sup>54</sup> Improved WOA for Cloud task scheduling (IWC)
  - <sup>55</sup> Meta Heuristic Optimization (MHO)
  - <sup>56</sup> MHO-S
  - <sup>57</sup> MHO-D
  - <sup>58</sup> Context-Aware VM Placement Optimization
  - <sup>59</sup> Cost optimized
  - <sup>60</sup> Integer-PSO
  - <sup>61</sup> Virtual Machine (VM)
  - <sup>62</sup> Identifier (id)
  - <sup>63</sup> Million Instructions Per Second (MIPS)
  - <sup>64</sup> Job
  - <sup>65</sup> Makespan
  - <sup>66</sup> Due date
  - <sup>67</sup> Lateness
  - <sup>68</sup> Unit Penalty
  - <sup>69</sup> Tardiness
  - <sup>70</sup> Swarm Intelligence
  - <sup>71</sup> Fitness function
  - <sup>72</sup> Boltzman probability factor
  - <sup>73</sup> Glover
  - <sup>74</sup> Tabu List
  - <sup>75</sup> Aspiration Criterion
  - <sup>76</sup> Tabu tenure
  - <sup>77</sup> Tournament Binary
  - <sup>78</sup> Roulette-Wheel Selection
  - <sup>79</sup> Chaos

# Multi-objective optimization of job scheduling problem in cloud computing

**Aboozar Zandvakili, Najme Mansouri & Mohammad Masoud Javidi**

Faculty of Mathematics and Computer, Shahid Bahonar University, Kerman, Iran

---

## Abstract

In the cloud environment, optimal task scheduling depends on different factors such as completion time, latency, and load balancing on the machines. But the main focus of previous works is to minimize completion time. In this paper, several parameters affecting the Quality of Service (QoS), such as minimum completion time, the minimum delay from delivery time, and maximum load balancing on machines simultaneously, are considered as objective functions. Scheduling is one of the problems for which there is no solution of polynomial order, and so metaheuristic algorithms such as Particle Swarm Optimization (PSO), Simulated Annealing (SA), Tabu Search (TS), and multi-objective Genetic Algorithm (GA) have used. Among single-objective algorithms, the PSO algorithm is implemented in much less time, and even for 30 tasks, it is competitive with multi-objective algorithms in terms of optimizing the objective function. In a large number of tasks, multi-objective algorithms are more appropriate in terms of execution time as well as optimization of objective functions. Because the Non-dominated Sorting Genetic Algorithm (NSGA-II) or Non-dominated Ranking Genetic Algorithm (NRGA) consider the objective functions separately and so they can choose the desired solution from the non-superior solutions to obtain a good balancing between different objectives.

**Keywords:** Cloud computing, Task scheduling, Multi-objective optimization.