



A task scheduling method based on differential evolutionary algorithm and particle community in order to increase efficiency in cloud computing environment

Golnaz Aghaee Ghazvini^{1*} , Zahra Heydaran Daroogheh Amnyieh² , Babak Nikmard³

^{1,3} Department of Computer Engineering, Dol.C., Islamic Azad University, Isfahan, Iran.

² Department of Electrical Engineering, Dol.C., Islamic Azad University, Isfahan, Iran.

ARTICLE INFO

Article Type:

Original Research

Received: 2025.01.13

Revised: 2025.03.12

Accepted: 2025.12.14

Keyword:

loud computing
task scheduling
increasing efficiency
differential evolutionary
algorithm
particle community algorithm

*Corresponding Author:

Golnaz Aghaee Ghazvini

Email:

g.ghaee@iau.ac.ir

ABSTRACT

Cloud computing is a new trend in Internet computing that provides various resources as a service. One of the basic challenges in the cloud environment is task scheduling in order to optimally use cloud-based services and increase efficiency, which makes the system in a stable condition. In recent years, extensive research has been done to assign tasks to virtual machines in cloud environments. Although these algorithms work effectively, they still do not accurately address some challenges such as communication delay, effective use of resources, reliability, and assignment of work to unusable machines. One of the significant challenges in the cloud computing environment is choosing the best virtual machine for each task with the aim of increasing efficiency and reducing costs. In this article, a solution called PSODE is presented to schedule tasks using evolutionary difference algorithms and particle community algorithm with the aim of increasing efficiency in the cloud computing environment. In order to evaluate, the proposed solution was compared with the difference evolutionary algorithm (DE) and the particle swarm intelligence algorithm (PSO). The evaluation results for 1000 tasks indicate a reduction in response time compared to the mentioned methods.



EXTENDED ABSTRACT

Introduction

Cloud computing technology is capable of sharing a paradigm of computing and storage infrastructures over a scalable network of internet resources; consequently, cloud users can execute dispersed information and services through remote servers worldwide. Virtualization is the most critical technology employed in cloud computing. In this context, virtualization refers to the simultaneous sharing of a software, resource, or physical device among multiple users in the form of virtual instances. Companies providing virtualization services offer users capabilities such as the creation, maintenance, and management of Virtual Machines (VMs) on off-site hardware. Currently, with the rapid advancement of virtualization technology, the creation of virtual machines—as the primary resources in cloud computing—has emerged as a novel solution for addressing the challenges of dynamic resource management. In fact, virtualization enables the execution of multiple virtual machines on a single physical machine by sharing all of its underlying hardware resources.

Methodology

Matlab software was used for simulation. In evaluating the results, the parameters of the PSODE algorithm, as well as the number of tasks and the number of virtual machines in the cloud environment, were considered and evaluated in different scenarios.

By implementing PSODE in task scheduling, the objective function reduction trend by this algorithm has been evaluated as shown in Figure 1.

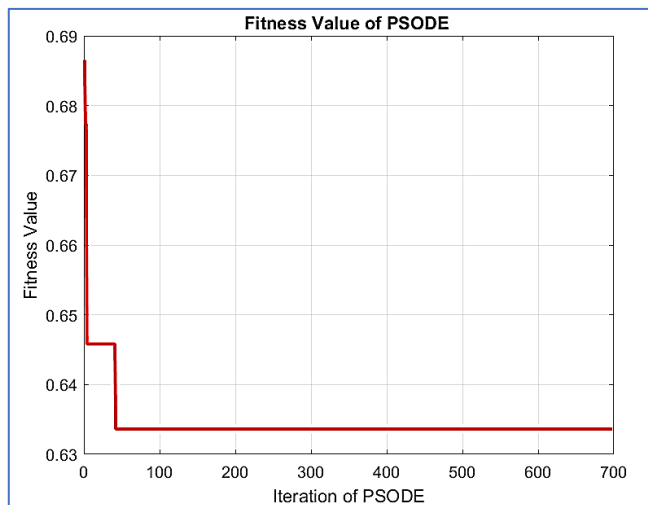


Figure 1. Minimization of the objective function by the PSO algorithm

As shown in the above figure, the objective function has been reduced in the proposed algorithm. Also, the fitness function (makespan) has been evaluated in

accordance with Figure 2.

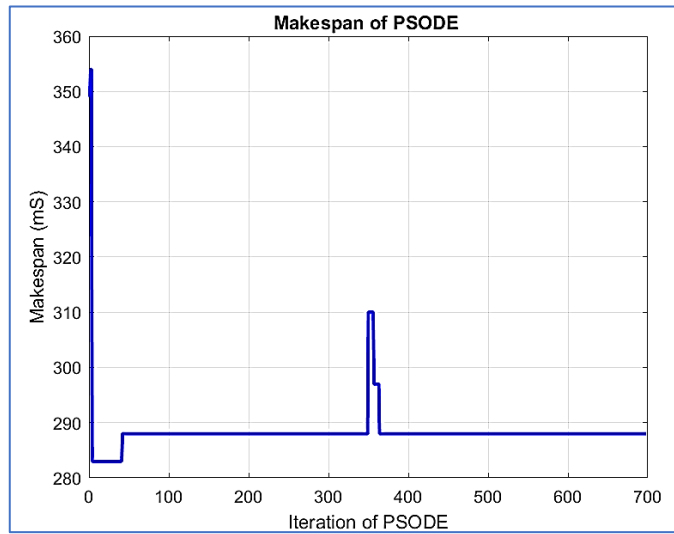


Figure 2. Makespan minimization by PSODE algorithm

In Figure 3, the throughput of the algorithm is analyzed, which reached a constant value after about 40 iterations and had an increasing trend.

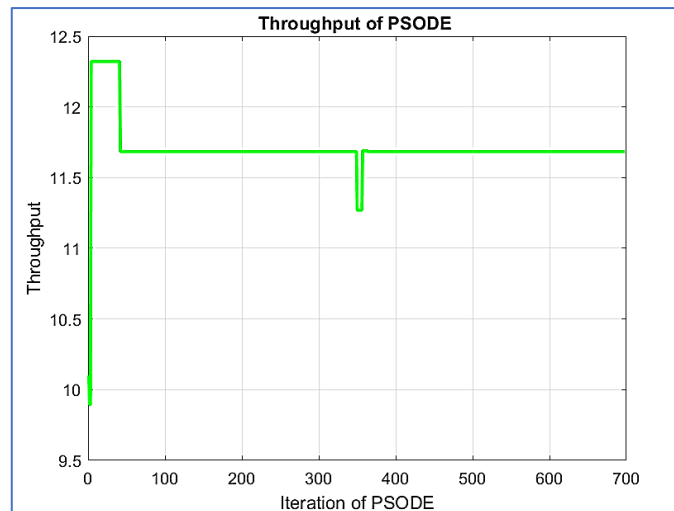


Figure 3. Throughput evaluation by PSODE algorithm

The values of the evaluation parameters in the final iteration of the algorithm have been calculated according to Table 2. Also, the result of the best response obtained in the final iteration of the algorithm, which is the result of scheduling tasks on virtual machines, has been evaluated as Table 1.

Table 1. Throughput and fitness evaluation in the final iteration

Fitness	MP	THr
۰,۷۱۰۷۷	۳۹۱	۹,۴۹۴۵

Also, the result of the best response obtained in the final iteration of the algorithm, which is the result of scheduling tasks on virtual machines, has been calculated as Table 2.

Table 2. The result of the best answer in the final iteration of the proposed algorithm

ماشین مجازی	تعداد کارها	زمان اجرا
۱	۸,۱۰,۲	۳۹۱
۲	۱,۳,۵,۷	۳۷۰
۳	۴,۶,۹	۳۰۰

In the algorithm proposed in this paper, Differential Evolution (DE) and Particle Swarm Optimization (PSO) algorithms are employed for task scheduling. The primary objective of this approach is to reduce response time and increase the throughput of the scheduling system. In the algorithm's objective function, two parameters, MP and Thr, are defined to evaluate response time and system throughput, respectively. The main goal of the PSODE algorithm is to minimize this objective function.

In this algorithm, the length of each solution is equal to the number of tasks submitted for scheduling, and each value in the solution structure represents the identifier of the virtual machine assigned to the corresponding task.

The steps of the Particle Swarm Optimization (PSO) algorithm are defined as follows:

1. Population initialization: An initial population of particles is generated randomly.
2. The velocity vector of each particle is randomly initialized. Then, the objective-function value of each particle is calculated according to Equation (7) presented in the paper. The objective function aims to minimize response time and maximize throughput.
3. The best position of each particle (personal best or pbest) is updated.
4. The best fitness value among the entire population is stored as the global best position (gbest).
5. The velocity and position of each particle are updated.

Results and discussion

Based on the results presented in the article, the proposed PSODE algorithm, which combines Differential Evolution (DE) with Particle Swarm Optimization (PSO), achieves a higher convergence speed than the standalone DE and PSO algorithms. PSODE reaches the optimal objective-function values within approximately 40 iterations, demonstrating its strong capability to rapidly reduce response time and increase throughput in cloud environments. In contrast, the DE algorithm exhibits a slower convergence rate and requires more iterations to approach the optimal values. It also shows relatively weak performance when handling high-volume workloads. Although PSO performs better than DE in some cases, its convergence rate is still slower than that

of the hybrid PSODE algorithm. Overall, the results clearly demonstrate the superiority of PSODE in terms of convergence speed and objective-function minimization. In the evaluation table, which considers 300, 500, 700, and 1,000 tasks allocated to 50 virtual machines, the objective-function values obtained by the algorithms are presented as follows.

Table 3. Comparison of the convergence speed of the algorithms

تعداد وظایف	الگوریتم PSODE	الگوریتم DE	الگوریتم PSO
300	1.2746	1.9044	1.6561
500	1.4819	1.4819	1.8872
700	1.0240	1.8872	1.0240
1000	1.6561	1.9044	1.4819

The PSODE algorithm, owing to its intelligent integration of the advantages of Differential Evolution (DE) and Particle Swarm Optimization (PSO), achieves a higher convergence rate and performs better than the individual DE and PSO algorithms in terms of reducing response time and increasing throughput. This algorithm reaches the optimal solution more rapidly, making it more suitable for cloud computing environments and scenarios involving a large number of tasks. Therefore, by combining the desirable characteristics of DE and PSO, the PSODE algorithm delivers superior overall performance, particularly regarding convergence speed and response-time optimization.

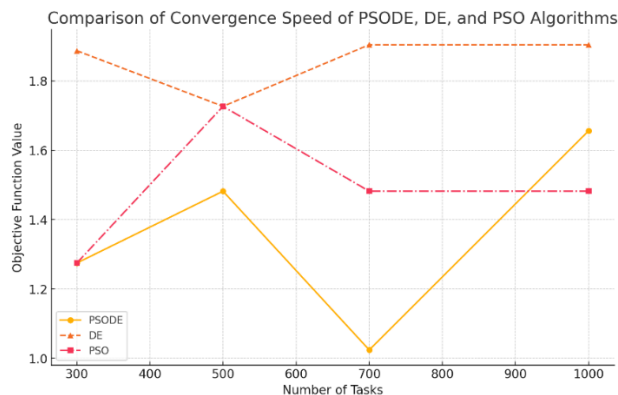


Figure 4. Comparison of the convergence speed of the PSODE, DE, and PSO algorithms

The figure above compares the convergence rate of the proposed algorithm (PSODE) with two other algorithms: Differential Evolution (DE) and Particle Swarm Optimization (PSO). As can be observed, the PSODE algorithm (solid line) demonstrates the best performance by reaching the optimal value more quickly, as indicated by the faster reduction in the objective function value. The DE algorithm (dashed line) has the slowest convergence rate and remains at a higher objective function value. The PSO algorithm (dotted line) shows moderate performance; it performs better than DE but worse than PSODE. Therefore, in terms of convergence speed and reaching the optimal solution, the hybrid PSODE algorithm demonstrates superior performance.

Conclusion

In this paper, an efficient scheduling algorithm was proposed to increase throughput and reduce the response time of virtual machines in a cloud computing environment. To this end, the Differential Evolution (DE) and Particle Swarm Optimization (PSO) metaheuristic algorithms were employed. This algorithm, which mathematically models a socio-political evolutionary process, has been used to solve a wide range of optimization problems. This evolutionary optimization strategy has demonstrated strong performance in terms of convergence rate and the ability to reach the global optimum. In the proposed approach, the Particle Swarm Optimization algorithm selects the best task, while the Differential Evolution algorithm selects the most suitable virtual machine on which the task should be executed. The evaluation results of the PSODE algorithm indicate that the proposed method can improve resource utilization and reduce response time in cloud computing environments. The use of other evolutionary algorithms may also be beneficial, particularly when they are integrated with the proposed method to solve problems in various areas of cloud computing. Furthermore, a fuzzy system can be combined with the proposed approach to address problems characterized by insufficient certainty. In future work, new algorithms will be developed to extend this type of load balancing and scheduling to workflows consisting of dependent tasks. This algorithm considers priority as the main Quality of Service (QoS) parameter.



یک روش زمانبندی وظایف مبتنی بر الگوریتم تکاملی تفاضلی و اجتماع ذرات به منظور افزایش کارایی در محیط محاسبات ابری

گلناز آقایی قزوینی^{۱*}، زهرا حیدران داروقه‌امنیه^۲، بابک نیک‌مرد^۳

۱ و ۳- گروه کامپیوتر، واحد دولت آباد، دانشگاه آزاد اسلامی، اصفهان، ایران.

۲- گروه برق، واحد دولت آباد، دانشگاه آزاد اسلامی، اصفهان، ایران.

چکیده

اطلاعات مقاله

نوع مقاله: مقاله پژوهشی

دریافت مقاله: ۱۴۰۳/۱۰/۲۴

بازنگری مقاله: ۱۴۰۳/۱۲/۲۲

پذیرش مقاله: ۱۴۰۴/۰۹/۲۳

کلید واژگان:

محاسبات ابری
زمانبندی وظایف
افزایش کارایی
الگوریتم تکاملی تفاضلی
الگوریتم اجتماع ذرات

*نویسنده مسئول:

گلناز آقایی قزوینی

پست الکترونیکی:

g.aghaee@iaau.ac.ir

محاسبات ابری یک روند جدید در محاسبات اینترنت بوده که منابع مختلف را به عنوان سرویس ارائه می‌نماید. یکی از چالش‌های اساسی در محیط ابری زمانبندی وظایف به منظور استفاده بهینه از خدمات مبتنی بر ابر و افزایش کارایی است که باعث می‌شود سیستم در شرایط پایدار قرار داشته باشد. در سال‌های اخیر تحقیقات وسیعی به جهت اختصاص وظایف به ماشین‌های مجازی در محیط‌های ابری ارائه شده است. اگر چه این الگوریتم‌ها به طور موثری کار می‌کنند ولی هنوز به برخی چالش‌ها مانند تاخیر ارتباطی، بکارگیری موثر منابع، قابلیت اطمینان و تخصیص کار به ماشین‌های غیرقابل استفاده، به طور دقیقی نپرداخته‌اند. یکی از چالش‌های حائز توجه در محیط محاسبات ابری، انتخاب بهترین ماشین مجازی برای هر وظیفه با هدف افزایش کارایی و کاهش هزینه است. در این مقاله، راهکاری به نام PSODE به منظور زمانبندی وظایف با بکارگیری الگوریتم‌های تفاضل تکاملی و الگوریتم اجتماع ذرات با هدف افزایش کارایی در محیط محاسبات ابری ارائه شده است. به منظور ارزیابی، راهکار پیشنهادی با الگوریتم تفاضلات تکاملی (DE) و الگوریتم هوش ازدحامی ذرات (PSO) مقایسه گردید. نتایج ارزیابی برای ۱۰۰۰ وظیفه، حاکی از کاهش زمان پاسخدهی نسبت به روش‌های مذکور است.



مقدمه

فناوری محاسبات ابری، قادر است یک نمونه از زیرساخت‌های محاسباتی و ذخیره‌سازی را بر روی یک شبکه مقیاس پذیر از منابع اینترنتی به اشتراک بگذارد، لذا کاربران ابری می‌توانند اطلاعات پراکنده و خدمات را از طریق سرورهای راه دور در سراسر جهان اجرا کنند. مجازی‌سازی مهمترین تکنولوژی مورد استفاده در محاسبات ابری است. مجازی‌سازی در محاسبات ابری، به معنای به اشتراک گذاشتن یک نرم افزار، منبع یا دستگاه واقعی به صورت هم‌زمان بین چند کاربر به صورت نسخه‌های مجازی است. شرکت‌های ارائه دهنده خدمات مجازی‌سازی، خدماتی مانند ایجاد، نگهداری و مدیریت یک ماشین مجازی را روی سخت‌افزارهای خارج از یک محل به کاربران ارائه می‌دهند. در حال حاضر، پیشرفت سریع فناوری مجازی‌سازی، ایجاد ماشین‌های مجازی، به عنوان منابع اصلی در محاسبات ابری، به یک راه‌حل جدید برای رفع مشکل مدیریت پویای منابع تبدیل شده است. در واقع مجازی‌سازی امکان اجرای چندین ماشین مجازی بر روی یک ماشین فیزیکی را با اشتراک‌گذاری تمامی منابع سخت‌افزاری آن ارائه می‌نماید.

چالش اصلی به‌منظور مجازی‌سازی در محیط توزیع‌شده، زمانبندی وظایف و در نتیجه آن برقراری توازن بین درخواست‌های ورودی است که باعث می‌شود تا سیستم در شرایط پایدار قرار داشته باشد. بحث زمان‌بندی در محیط محاسبات ابری، از اهمیت ویژه‌ای برخوردار است [1].

زمان‌بندی کار یکی از مهم‌ترین و چالش‌انگیزترین موضوعات در حوزه محاسبات ابری است. مدل‌های زمان‌بندی مختلفی وجود دارد که به توسط محققان در محیط‌های محاسباتی توزیع شده ارائه شده است، اما راه‌حل‌های موجود برای مسأله زمان‌بندی اغلب کامل نیستند، زیرا آن‌ها تنها بر روی یک هدف خاص مانند کمینه‌سازی زمان اجرا یا حجم کار تمرکز دارند و از خصیصه‌های محاسبات ابری برای زمان‌بندی کار استفاده نمی‌کنند. یک زمان‌بند کار در محاسبات ابری باید کاربران ابری را با کیفیت سرویس توافق شده ارضا کند و سود و منفعت فراهم‌کننده ابری را نیز بهبود ببخشد [۲]. زمان‌بندی و اختصاص منابع نه تنها با کیفیت سرویس دهی مرتبط است، بلکه مستقیماً بر روی سود ارائه‌دهندگان سرویس ابری نیز تاثیرگذار است. لذا، در سال‌های اخیر زمان‌بندی منابع به یکی از بروزترین موضوعات محاسبات ابری تبدیل شده و الگوریتم‌های متعددی توسط محققان در این زمینه ارائه شده است [۳]. اگر چه این الگوریتم‌ها به طور موثری کار می‌کنند، ولی هنوز به برخی چالش‌ها به طور دقیق پرداخته نشده است که از آن جمله می‌توان به سرعت پایین همگرایی در الگوریتم‌های فرا اکتشافی، در نظر نگرفتن زمان اجرای وظایف، زمان تکمیل کار بالا، عدم توجه به در دسترس بودن منابع و بار پردازنده‌ها، عدم توازن بار، زمان تکمیل بالا در محیط سازگار، انتساب وظایف به پردازنده‌ای با زمان اجرای نامناسب اشاره نمود. اغلب سرویس گیرنده بر اساس نیاز خود به بررسی محیط ابر می‌پردازد تا بتواند جریان کاری خود را توسط یک یا چند منبع مناسب به انجام رساند. چالش اساسی زمانی رخ می‌دهد که تعداد کارها و منابع زیاد باشد، در این زمان سرویس گیرنده نمی‌تواند به راحتی منابع مناسب را انتخاب کند. علت به وجود آمدن این مشکل، پیچیدگی محاسباتی بالا و سرعت پایین در همگرایی اولیه الگوریتم‌های فرا اکتشافی موجود است که در نهایت منجر به کاهش سرعت اجرای الگوریتم می‌گردد.

برای حل این چالش، الگوریتم PSODE برای زمان‌بندی کارها از ترکیب الگوریتم تفاضل تکاملی و الگوریتم اجتماع ذرات برای نگاشت کارها به ماشین‌های مجازی به منظور کاهش زمان پاسخ و افزایش توان عملیاتی ارائه شده است. این الگوریتم، با استفاده از الگوریتم‌های فرا ابتکاری PSO و DE تخصیص جریان کارها به منابع مناسب، جهت رسیدن به مناسب‌ترین (کمترین) زمان پاسخ و توان عملیاتی مورد بررسی و ارزیابی قرار می‌گیرد. ساختار مقاله به صورت زیر سازمان‌دهی شده است.

در بخش دوم این مقاله، مروری بر مهم‌ترین تحقیقات مرتبط با حوزه مربوطه جمع‌آوری شده است. در بخش سوم، الگوریتم PSODE جهت زمانبندی وظایف در محیط ابر معرفی شده است. در بخش چهارم، نتایج ارزیابی و مقایسه و در بخش پنجم، نتیجه‌گیری و پیشنهادات آتی بررسی شده است.

۲-پیشینه پژوهش

یکی از مزایای مهم سیستم توزیع شده سرعت بالای اجرای برنامه است. زیرا یک برنامه همزمان می‌تواند از چندین سیستم برای اجرا شدن استفاده کند که از آن حمله می‌توان به شبکه‌های حسگر بیسیم، شبکه‌های اقتضایی خودرویی، اینترنت اشیا و رایانش ابری اشاره نمود. اصطلاح «رایانش توزیع شده» به مدلی گفته می‌شود که در آن اجرای یک سیستم نرم‌افزاری بین چندین کامپیوتر، به منظور بهبود کارایی و عملکرد، به اشتراک گذاشته می‌شود. برخی از مسایل اصلی در چنین محیطی، مقیاس پذیری، دسترسی بیشتر، تعادل بار و هزینه و زمانبندی اجرای وظایف است. فرآیند زمانبندی جریان‌های کار در رایانش توزیع شده، به نگاشت وظایف به منابع محاسباتی موجود و زمانبندی اجرای آنها به طوری که وابستگی میان آن‌ها حفظ شود اطلاق می‌شود. ساختار جریان‌های کاری اغلب به صورت یک گراف بدون دور و همانند یک ساختار درختی تعریف می‌شود. زیرساخت‌های مختلفی به منظور برنامه‌ریزی جریان کار طراحی شده که می‌توان این زیرساخت‌ها را در سه گروه متمرکز، سلسله مراتبی و غیرمتمرکز طبقه‌بندی کرد. در زمانبندی متمرکز، یک زمان‌بند مرکزی منحصربه‌فرد مسئول تصمیم‌گیری درباره نگاشت وظایف موجود در جریان کار به منابع است [۴]. در رویکرد سلسله مراتبی یک زمان‌بند اصلی مسئولیت مدیریت مجموعه‌ای از زمان‌بندهای سطح پایین که آن‌ها نیز به نوبه خود مسئول زیرمجموعه‌ای از وظایف جریان‌های کار هستند را برعهده دارد. در نهایت، در معماری غیرمتمرکز، مجموعه‌ای مستقل از زمان‌بندهای غیرمتمرکز وجود دارند و هرکدام از آن‌ها مسئول زمانبندی یک جریان کاری هستند [۵،۶].

محاسبات ابری در راه توسعه خود با مشکلات مختلفی از جمله مصرف انرژی مواجه است. بالابودن مصرف انرژی، هزینه‌های عملیاتی و زمان انجام کارها را به واسطه عدم مدیریت بهینه منابع در محیط ابری رخ می‌دهد. مطمئناً در نظر نگرفتن چنین مسائلی می‌تواند منجر به بوجود آمدن نتایج فاجعه باری برای فراهم‌کنندگان منابع و سرویس‌های ابری شود. اهمیت بالای کاهش مصرف انرژی و تأثیرات آن بر هزینه‌های تمام شده سرویس‌ها، باعث شده که در سال‌های اخیر تحقیقات گسترده‌ای در زمینه بهینه‌سازی مصرف انرژی در محیط ابر صورت بگیرد [۷،۸]. اخیراً روش‌های زیادی برای مدیریت انرژی در سیستم‌های توزیع‌شده با مقیاس بزرگ ارائه شده است که در ادامه بررسی می‌شوند.

در [۹] یک مدل تخصیص منابع آگاه از مکان برای محیط‌های ابری استفاده شده است که دارای دو عملکرد مهم شامل تعیین جایگذاری ماشین‌های مجازی و دیگری تصمیم برای مهاجرت به ماشین‌های مجازی است. بدین صورت که اگر ارائه‌کننده درخواستی از کاربر دریافت کند، در آن صورت ارائه‌کننده نزدیکترین میزبان فیزیکی در این ناحیه را می‌یابد. ارائه‌کننده سرویس، یک ماشین مجازی مناسب یافته و سطح بهره‌وری آن را با ماشین فیزیکی خاص بررسی می‌کند، اگر سطح بهره‌وری ماشین فیزیکی برای تخصیص یک ماشین مجازی مناسب باشد، فراهم‌کننده ماشین مجازی را به آن اختصاص می‌دهد. مزیت این مدل این است که از کاهش عملکرد مرکز داده جلوگیری می‌کند، ولیکن این مدل در یک محیط واقعی ابری قابل اجرا نیست.

در [۱۰] الگوریتم راهکاری جهت تخصیص ماشین‌های مجازی روی منابع در دسترس فیزیکی ارائه شده است. با استفاده از آن، مصرف انرژی ماشین‌های مجازی را روی تمام ماشین‌های فیزیکی مجاسبه نموده و در انتها آن را روی میزبانی قرار می‌دهد که کمترین مصرف انرژی را داشته باشد.

در [۱۱] یک راهکار مبتنی بر الگوریتم‌های فرا اکتشافی به منظور زمانبندی ارائه شده است. در این روش، ابتدا همه ماشین‌های مجازی بر اساس فاکتور مرتب‌سازی خود مرتب می‌گردند، سپس تمام تخصیص‌های ممکن که در آن شرایط خواسته شده نقض نشود محاسبه می‌گردد. پیچیدگی زمانی این الگوریتم نسبتاً بالا است و در زمره الگوریتم‌های بهینه قرار نمی‌گیرد.

در [۱۲] راهکاری به منظور شناسایی تعادل بین مصرف انرژی و کارایی ارائه‌دهندگان ابر ارائه شده است. در این راهکار میزان افت کارایی در شرایطی بررسی می‌گردد که منابع فیزیکی باید با سایر ماشین‌های مجازی به اشتراک گذاشته شود. پایین‌ترین سطح کارایی قابل قبول، میزانی است که در توافقنامه بین کاربر و ارائه‌دهنده قرارداد شده است. سه پارامتر شامل تعداد درخواست‌های رقیب برای یک منبع، بار کلی روی یک منبع و مقدار منبع مورد نیاز خود برنامه، در کارایی یک برنامه کاربردی موثر در نظر گرفته شده است. ولی به نظر می‌رسد که باید پارامترهایی نظیر تأخیر شبکه، یا تأثیر ناهمگنی مراکز داده که از جمله مشخصه‌های محیط‌های ابری محسوب می‌شود، نیز بررسی شوند.

در [۱۳] مسئله جایگذاری پویای برنامه‌ها در سیستم مجازی، بررسی شده است، با این هدف که ضمن حفظ قراردادهای مابین کاربر و ارائه‌دهندگان، مصرف توان کمینه شود. برای این منظور، کتابخانه‌ای برای جایگذاری برنامه به نام pmaper معرفی گردید. این کتابخانه شامل سه بخش مدیریتی و یک بخش میانجی است که اعمال و عملکردهای خود را با یکدیگر هماهنگی و برای تخصیص تصمیم‌گیری می‌کنند. مدیریت کارایی به بررسی رفتار برنامه‌ها و تغییر اندازه ماشین‌های مجازی مطابق با نیازمندی‌ها منابع جاری و قرارداد بین کاربر و ارائه‌دهنده می‌پردازد. مدیریت توان، وظیفه تنظیم حالات توان سخت افزار و اعمال مقیاس‌پذیری پویای فرکانس و ولتاژ را بر عهده دارد. مدیریت مهاجرت، وظیفه اجرای دستورالعمل‌های لازم برای انجام مهاجرت به منظور مدیریت حجم کاری موجود را بر عهده دارد.

در [۱۴]، راهکاری برای زمانبندی برنامه‌های تحت وب چند لایه در سیستم‌های ناهمگن مجازی شده به منظور کمینه کردن مصرف انرژی ارائه شده است. در این بررسی، افزایش بهره‌وری منجر به کاهش کارایی شده است، لذا این نتیجه حاصل می‌شود که مصرف انرژی در هر تراکنش به شکل یک کمان U شکل است که می‌توان نقطه بهینه بهره‌وری را در آن مشخص کرد. در [۱۲] راهکاری به نام سیاست آگاه از حجم کاری و مشتری ۱ ارائه شده است که از روش منحصر به فردی برای تعیین گره‌ها و تخصیص درخواست به هر گره استفاده شده است.

افزایش نیاز به استفاده از اینترنت سبب گردیده نیاز به قابلیت اعتماد و اطمینان و کیفیت سرویس برای برنامه‌های کاربردی وب افزایش یابد [۱۵] وجود وب سرور در گستره جغرافیایی باعث شده که برای بالا بردن میزان بهره‌وری و پایین آوردن زمان پاسخ درخواست کنندگان، تعادل بار در محاسبات ابری به صورت چالشی درآید که باید همواره مورد بررسی قرار گیرد.

در [۱۶] راهکاری برای تعادل بار مبتنی بر پایه کلونی مورچگان و تئوری شبکه پیچیده در فدراسیون محاسبات ابری مطرح شد. در این جا از خصوصیات یک دنیای کوچک و مقیاس آزاد در شبکه پیچیده برای دستیابی به تعادل بار بهتر استفاده می‌کند. مورچه‌ها دارای حافظه بسیار محدود هستند. با این حال مورچه‌ها قدرت مدیریت و انجام انواع کارهای پیچیده با قابلیت اطمینان بالا را دارند و مورچه‌ها دارای قدرت بالایی برای ساختن لانه و جستجو برای یافتن غذا دارند. برخی از تکنیک‌های محاسبات با الهام از زندگی اجتماعی حشراتی مانند مورچه طراحی گردیده است. در این راهکار، برای جلوگیری از ایجاد سربار در گره و ایجاد توازن بار در گره

^۱ Workload and Client Aware Policy (wcap)

ها بدین ترتیب عمل شده است که وقتی گرهی با حجم کاری بیش از حد آستانه پیدا شود، مورچه (تقاضا) فرستاده شده، توسط آن نود خارج گردیده و گرهی که حجم کاریش کم است، مورچه پس فرستاده شده را می پذیرد به این ترتیب تعادل بار بین گره‌ها بهبود می‌یابد و کارایی این مکانیزم بالا می‌رود. همچنین، از جدول برای ذخیره اطلاعات گره‌های همسایه استفاده می‌گردد. مورچه‌ای که سر بار ایجاد می‌کند با توجه به این جدول به نزدیکترین همسایه فرستاده می‌شود. این تکنیک، برای محیط‌های ناهمگن مناسب است که قابل تطبیق با محیط‌های پویا است و تحمل پذیری گره‌ها در مقابل خطا را نیز تامین می‌کند و خاصیت مقیاس پذیری خوبی دارد در نتیجه کارایی را بطور قابل ملاحظه‌ای بالا می‌برد.

در [۱۷]، با استفاده از الگوریتم RR_1 به توازن بار در مراکز داده پرداخته شده است. الگوریتم RR یکی از روش‌های برقراری توازن بار با رویکرد بهینه‌بودن مصرف انرژی است. این الگوریتم، کارها را به تمام ماشین مجازی توزیع می‌کند. تمامی کارهای به ماشین‌های مجازی بر اساس ترتیب RR تخصیص داده می‌شوند. به این معنی که اولین کار به اولین ماشین مجازی که امکان اجرای کار را داشته باشد، تخصیص داده می‌شود. انتخاب ماشین‌های مجازی به صورت محلی و مستقل از تخصیص دیگر ماشین‌های مجازی، روی هر مرکز داده صورت می‌پذیرد.

با مرور تحقیقات قبل این نتیجه حاصل می‌شود که اگرچه راهکارهای بسیاری توسط محققین برای زمانبندی وظایف در محیط ابر ارائه شده است، اما راه‌حل‌های موجود برای مساله زمانبندی برای محاسبات ابری همچنان با چالش‌هایی مواجه هستند. زیرا آن‌ها تنها بر روی هدف خاصی تمرکز دارند مثل کمینه سازی زمان اجرا یا حجم کار و از خصیصه‌های محاسبات ابری برای زمانبندی کار استفاده نمی‌کنند [۱۸، ۱۹]. یک زمانبند کار در محاسبات ابری باید رضایت کاربران ابری را با کیفیت سرویس توافق شده تضمین کند و سود و منفعت فراهم‌کننده ابری را نیز بهبود بخشد [۲۰]. در این مقاله برای بهینه سازی زمان پاسخ و توان عملیاتی سیستم از ترکیب الگوریتم ازدحام ذرات و تکامل تفاضلی استفاده خواهد شد که با استفاده از ترکیب این دو الگوریتم می‌توان بهترین منابع را برای انجام کارها در محیط ابر انتخاب کرد.

۳- روش شناسی

در الگوریتم ارائه شده در این مقاله، برای زمانبندی کارها از الگوریتم‌های تفاضل تکاملی و الگوریتم اجتماع ذرات استفاده شده است. هدف اصلی در این راهکار، کاهش زمان پاسخ و افزایش توان عملیاتی سیستم زمانبندی است. در تابع هدف الگوریتم، دو پارامتر MP و Thr برای ارزیابی زمان پاسخ و توان عملیاتی تعریف شده است که هدف اصلی کمینه سازی این تابع توسط الگوریتم PSODE است.

در محیط محاسبات ابری، درخواست‌های ارسال شده به ابر بسته به نوع کار، یا در ابر ذخیره می‌شوند و یا به سمت کاربر دیگر ارسال خواهند شد و یا بعد از عمل پردازش به سمت کاربر برگشت داده می‌شوند. از آنجایی که ممکن است یک درخواست کاربر توسط دستگاه (نرم افزار نصب شده در دستگاه کاربر) به چند بخش تجزیه شود، لذا به آن‌ها واژه کار نسبت داده می‌شود. به عبارت دیگر، یک درخواست اغلب به چند کار تبدیل می‌شود و هر کار توسط منابع مختلف در ابر مورد پردازش قرار می‌گیرد. لذا، اطلاعات پردازشی مورد نیاز هر کار همراه با آن به سمت ابر ارسال می‌شوند. منابع موجود در محیط محاسبات ابری در قالب ماشین‌های مجازی در نظر گرفته می‌شوند که توانایی پردازش کارهای ارسال شده به سمت ابر را دارند. به عبارت دیگر، هر ماشین فیزیکی موجود در ابر به دلیل قابلیت سخت‌افزاری زیاد، به چندین ماشین مجازی تقسیم می‌شود. از این‌رو، اطلاعات

ماشین‌های مجازی موجود در ابر نیز باید در محلی ذخیره شوند. این محل با نام «مولفه زمانبند متمرکز» در محیط ابری تعریف می‌شود. مولفه زمانبند متمرکز، علاوه بر ذخیره‌سازی اطلاعات ماشین‌های مجازی موجود، اطلاعات کارهای ارسال‌شده به ابر را نیز در خود نگه می‌دارد. در این مولفه زمانبند متمرکز، الگوریتم تکاملی ترکیبی پیشنهادی نیز وجود دارد و این الگوریتم در این مولفه اجرا می‌شود. هدف اصلی در این مقاله، بهینه‌سازی تابع هدف الگوریتم ارائه شده است که میزان زمان پاسخ و توان عملیاتی را بهبود دهد. از آنجایی که الگوریتم‌های بکارگرفته شده در این راهکار، از نوع الگوریتم‌های تکاملی هستند، در چند تکرار به مقدار بهینه تابع هدف دست پیدا می‌کنند. هر پاسخ در تکرارهای مختلف الگوریتم، نشان‌دهنده یک عمل زمان‌بندی K کار ارسال به M ماشین مجازی است. هر ماشین مجازی دارای ظرفیت پردازشی است که بر اساس رابطه ۱ محاسبه می‌شود [۲۱]:

$$C_{vmi} = Mips_{vmi} \times Pes_{vmi} \quad (۱)$$

در رابطه فوق، C_{vmi} نشان‌دهنده ظرفیت پردازشی ماشین مجازی i ، Pes_{vmi} تعداد پردازنده‌ها در ماشین مجازی و $Mips_{vmi}$ تعداد میلیون دستورالعمل در ثانیه همه پردازنده‌های ماشین مجازی i است. برای رسیدن به هدف نهایی در روش پیشنهادی، باید زمان اجرای کارها در هر ماشین مجازی به حداقل برسد تا زمان پاسخ کاهش یابد. زمان اجرا و طول زمان اجرای هر کار با روابط زیر تعریف می‌شوند:

$$TL_j = C_{vmi} / ET_j \quad (۲)$$

$$RT_j = FT_j - AT_j \quad (۳)$$

در رابطه فوق، TL_j زمان انجام کار j ، AT_j زمان ورود کار، RT_j زمان پاسخ کار و FT_j زمان اتمام هر کار را نشان می‌دهد. از آنجایی که مجموعه کارها بر روی چند ماشین مجازی زمانبندی می‌شوند، ممکن است بعد از عمل تخصیص، زیرمجموعه‌ای از کارها به هر ماشین مجازی تخصیص یابند و لذا ممکن است تعداد کارهای تخصیص‌یافته به هر ماشین مجازی با سایر ماشین‌های مجازی متفاوت باشد. بنابراین زمان اتمام کارها بر روی هر ماشین مجازی متفاوت خواهد بود. زیرا برخی از ماشین‌های مجازی ممکن است تعداد کارهای کمتر یا بیشتری از سایر ماشین‌ها دریافت کنند و در نتیجه زودتر و یا دیرتر از سایر ماشین‌های مجازی پردازش را به اتمام برسانند. بنابراین، زمان پاسخ کارهای تخصیص‌یافته به ماشین مجازی در مقایسه با سایر ماشین‌ها متفاوت است. در محاسبه زمان پاسخ در الگوریتم ارائه شده، زمان ورود مجموعه کارها به محیط ابری برابر با صفر است و زمان پاسخ برابر با زمان پایان زیرمجموعه کارهای تخصیص‌یافته به هر ماشین مجازی است. پارامتر MP در الگوریتم نشان‌دهنده زمان پاسخ زیرمجموعه کارهای تخصیص‌یافته به ماشین‌های مجازی است. بیشترین زمان لازم برای به اتمام رسیدن پردازش کارهای تخصیص‌یافته در بین ماشین‌های مجازی باید به عنوان زمان پاسخ کلی الگوریتم در نظر گرفت که در ادامه بیان شده است. از آنجایی که زمان پردازش هر کار بر روی هر ماشین مجازی ممکن است بر اساس توانایی پردازنده متفاوت باشد، بنابراین زمان اجرای کارها بر روی هر ماشین مجازی در روش پیشنهادی به صورت یک ماتریس P با ابعاد $m \times n$ نشان داده می‌شود که زمان پردازش کار j بر روی

ماشین مجازی i -ام در ماتریس P به صورت $P_{i,j}$ مشخص می شود. در رابطه زیر زمان پایان یافتن کارها توسط m ماشین های مجازی بر اساس زمان بندی اختصاص یافته محاسبه شده است [۲۱]:

$$MP = \max_{i=1}^m \{ \sum_{j=1}^n x_{i,j} \times p_{i,j} \} \quad (۴)$$

در رابطه فوق، ماتریس X با ابعاد $m \times n$ حاوی مقادیر صفر و یک است که سطرهای آن تعداد ماشین های مجازی را نشان می دهند و ستون های آن نشان دهنده تعداد کارهایی هستند که به ماشین های مجازی (مطابق با شماره سطر) اختصاص یافته اند. مقدار صفر نشان دهنده اختصاص نیافتن آن کار به ماشین مجازی است و مقدار یک نشان دهنده اختصاص یافتن آن کار به ماشین مجازی است. اگر کار j ، توسط ماشین i انجام شده باشد، عنصر $X_{i,j}$ در ماتریس X مقدار یک و در غیر این صورت آن عنصر برابر با صفر خواهد شد. لازم به ذکر است که یک کار فقط می تواند به یک ماشین مجازی تخصیص یابد و بر روی آن اجرا شود و تا اتمام آن کار، قابل تحویل دادن به ماشین مجازی دیگر نخواهد بود. ساختار ماتریس X همان ساختار پاسخها در الگوریتم PSODE است. پارامتر دیگر، افزایش توان عملیاتی است که برای این منظور باید تعداد کارهای انجام شده در واحد زمان به حداکثر برسد که این پارامتر در رابطه زیر آمده است:

$$Thr_{vmi} = NoJ_{vmi} / RT_{vmi} \quad (۵)$$

در رابطه فوق، Thr_{vmi} بیانگر توان عملیاتی ماشین مجازی i ، NoJ_{vmi} نشان دهنده تعداد کارهای انجام شده توسط پردازنده های ماشین مجازی i و RT_{vmi} نشان دهنده زمان اجرای کارهای تخصیص داده شده به همان ماشین مجازی است. محاسبه توان عملیاتی باید برای تمامی ماشین های مجازی موجود در ابر که در عمل زمان بندی توسط الگوریتم انتخاب شدند، انجام شود. توان عملیاتی نهایی برای این مجموعه ماشین مجازی، برابر با میانگین توان عملیاتی ماشین های مجازی است که در رابطه ۶، نشان داده شده است [۲۱].

$$Thr = \sum_{i=1}^m Thr_{vmi} / m \quad (۶)$$

در رابطه ۶، m تعداد ماشین های مجازی موجود در ابر برای عمل زمان بندی است. تابع هدف پیشنهادی با ترکیب دو پارامتر MP و Thr به صورت زیر تعریف شده است با این هدف که مقدار MP کاهش و میزان Thr افزایش یابد.

$$Fit = \alpha \frac{MP}{Avg(RT)} + \beta \frac{m}{Thr} \quad (۷)$$

در این رابطه، پارامتر MP بر میانگین زمان اجرا ($Avg(RT)$) تقسیم شده است، در بخش دوم، مقدار Thr برای تبدیل بیشینه سازی به کمینه سازی، مقدار آن در مخرج کسر قرار گرفته است که مقدار آن برابر صفر نخواهد شد. با توجه به اینکه در تابع هدف باید عمل کمینه سازی صورت گیرد، بر اساس میزان تاثیر و اهمیت دو پارامتر MP و Thr از دو ضریب α و β استفاده شده است که $\beta = 1 - \alpha$ که بر اساس ساله مورد نظر در شرایط مختلف، با تغییر این دو ضریب می توان میزان اثرگذاری دو پارامتر مورد نظر را تغییر داد. اگر هر دو

پارامتر موجود در تابع هدف از اثر یکسانی برخوردار هستند، می‌توان مقدار هر دو ضریب را برابر با ۰/۵ در نظر گرفت، در غیر این صورت، می‌توان مقادیر این دو ضریب را متفاوت در نظر گرفت.

۳-۱- ساختار پاسخ در الگوریتم PSODE

در این الگوریتم، طول پاسخ‌ها برابر با تعداد کارهای ارسال شده جهت زمانبندی است و مقادیری که در ساختار پاسخ‌ها قرار می‌گیرند شامل شماره ماشین‌های مجازی خواهد بود. به عنوان مثال، اگر تعداد کارهای ارسال شده جهت زمانبندی برابر با ۸ کار و تعداد ماشین‌های مجازی موجود برابر با ۳ ماشین مجازی باشد، در این صورت، ساختار یکی از پاسخ‌های تولیدشده در الگوریتم پیشنهادی به صورت جدول زیر خواهد بود.

جدول ۱. ساختار پاسخ در الگوریتم PSODE							
اندیس کار	۸	۷	۶	۵	۴	۳	۲
اندیس ماشین مجازی	۲	۳	۳	۲	۲	۳	۱

هر عدد در ساختار فوق، نشان‌دهنده شماره کار و شماره ماشین مجازی تخصیص یافته به کار است. در الگوریتم PSODE، ترکیب دو الگوریتم تفاضلات تکاملی و هوش ازدحامی ذرات برای یافتن میزان بهینه تابع هدف استفاده می‌شود که با بهینه‌سازی این تابع، زمانبندی به صورت بهینه صورت خواهد گرفت الگوریتم با تکرار الگوریتم‌های تکاملی به نتیجه می‌رسد، بخشی از تکرار الگوریتم ترکیبی PSODE توسط الگوریتم تفاضلات تکاملی انجام می‌شود و بخشی دیگر از تکرار آن توسط الگوریتم هوش ازدحامی ذرات انجام می‌شود. این عمل چند بار به صورت متناوب صورت می‌گیرد تا در نهایت پاسخ بهینه زمانبندی به دست آید. در ابتدا، الگوریتم تکاملات تفاضلی به ازای تعدادی از جمعیت اجرا می‌شود و بعد از چند تکرار، نتیجه آن به الگوریتم هوش ازدحامی ذرات داده می‌شود. الگوریتم هوش ازدحامی ذرات نیز در طی چند تکرار پاسخ‌های دریافت کرده از الگوریتم تفاضلات تکاملی را بهینه می‌کند. در ادامه، الگوریتم تفاضلات تکاملی نتایج الگوریتم هوش ازدحامی ذرات را دریافت کرده و روند بهینه‌سازی را ادامه می‌دهد. بعد از این عمل، نتایج به الگوریتم هوش ازدحامی ذرات سپرده می‌شوند تا توسط این الگوریتم بهینه شوند. این عمل چندین بار انجام می‌شود تا الگوریتم PSODE به تکرار نهایی خود برسد.

در الگوریتم ازدحام ذرات، ابتدا یک جمعیت اولیه از ذرات به صورت تصادفی تولید می‌شود. هر ذره شامل یک موقعیت (پاسخ اولیه زمانبندی وظایف روی ماشین‌های مجازی) و سرعت اولیه است.

هر ذره در PSO به صورت یک بردار نشان داده می‌شود که شماره ماشین مجازی اختصاص داده شده به هر وظیفه را مشخص می‌کند مانند آنچه در ساختار پاسخ الگوریتم PSODE بیان شده است.

گام‌های الگوریتم PSO به صورت زیر تعریف می‌شود:

- (۱) شامل مقداردهی اولیه جمعیت (تولید یک جمعیت اولیه از ذرات به طور تصادفی)
- (۲) مقداردهی تصادفی اولیه به بردار سرعت هر ذره، محاسبه مقدار تابع هدف برای هر ذره با توجه به رابطه (۷) که در مقاله ذکر شده است (کمینه‌سازی زمان پاسخ و بیشینه‌سازی توان عملیاتی).
- (۳) به‌روزرسانی بهترین موقعیت‌ها

۴) ذخیره بهترین مقدار برازندگی در کل جمعیت به عنوان بهترین موقعیت سراسری

۵) به روزرسانی سرعت و موقعیت ذرات

۴- نتایج تحلیلی و آزمایشات

برای شبیه سازی از نرم افزار متلب استفاده شده است. در ارزیابی نتایج، پارامترهای الگوریتم PSODE و همچنین تعداد کارها و تعداد ماشین های مجازی موجود در محیط ابری متفاوت در نظر گرفته شده است و به صورت سناریوهای مختلف ارزیابی انجام می شود. در شبیه سازی ها، تعدادی کار و تعدادی ماشین مجازی در محیط ابری تعریف شده است که هر کار بر روی هر ماشین مجازی دارای زمان اجرایی متفاوتی خواهد بود. زمان ورود کارها جهت انجام زمانبندی، برابر با صفر خواهد بود و زمان اجرای هر کار نیز بر اساس سوابقی که در سیستم زمانبند از قبل موجود بوده است و بر اساس توانایی ماشین های مجازی، در ابتدای عملیات زمانبندی تعیین شده است. برای انجام عمل زمانبندی و تعیین میزان برازندگی هر پاسخ در الگوریتم PSODE، از این زمان اجرا استفاده خواهد شد.

جدول ۲. پارامترهای الگوریتم PSODE در ارزیابی

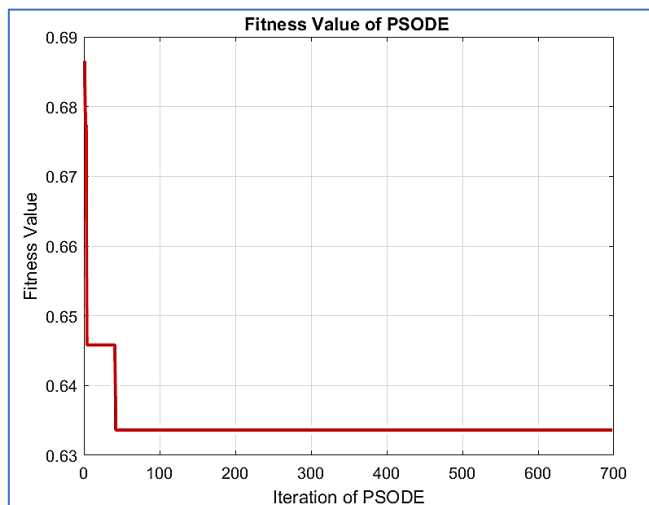
پارامتر	مقدار تنظیم شده برای DE	مقدار تنظیم شده برای PSO
تعداد تکرار الگوریتم	۵۰	۳۰۰
تعداد جمعیت	۲۰	۲۰
نرخ عملگر جهش	۰/۴	-
نرخ عملگر تقاطع	۰/۷	-
نرخ ضریب تغییر سرعت	-	۰/۰۱
ضریب α	۰/۵	۰/۵
ضریب β	۰/۵	۰/۵

تعداد ماشین های مجازی در بازه [۳ - ۵۰] در نظر گرفته شده است، لذا در شبیه سازی های مختلف، تعداد ماشین های مجازی در این بازه انتخاب می شوند. شوند. تعداد کارها در این شبیه سازی برابر با ۱۰ کار و تعداد ماشین های مجازی برابر با ۳ ماشین مجازی است و زمان اجرای اولیه هر کار بر روی هر ماشین مجازی قبل از انجام عمل زمانبندی به صورت زیر در نظر گرفته می شود.

جدول ۳. زمان اجرای اولیه کارها روی ماشین های مجازی

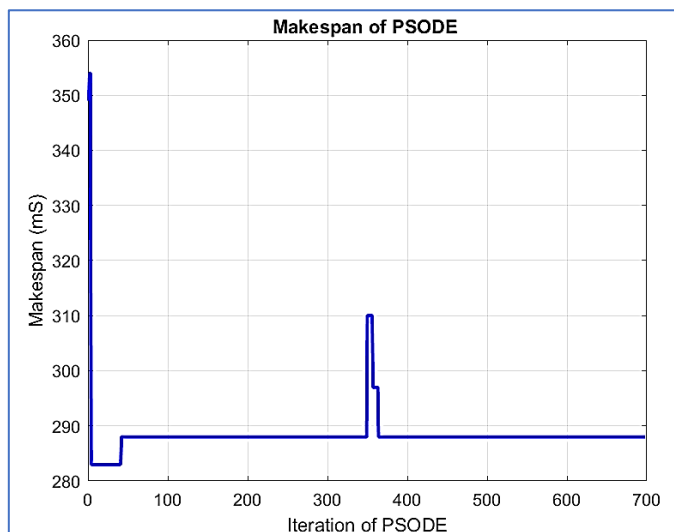
	۱	۲	۳	۴	۵	۶	۷	۸	۹	۱۰
۱	۹	۱۷۰	۱۴۴	۱۹۸	۸۴	۱۴۶	۵۱	۱۵۳	۶۲	۶۸
۲	۴۰	۱۳	۵۷	۱۲۱	۱۵۷	۶۳	۱۱۶	۸۵	۱۵۱	۱۵۰
۳	۱۵۴	۲	۱۰۵	۱۹۹	۷۹	۱۶	۱۰۹	۱۸۲	۸۵	۱۵۸

با اجرای PSODE در زمانبندی وظایف، روند کاهش تابع هدف توسط این الگوریتم به صورت شکل (۱) ارزیابی شده است.



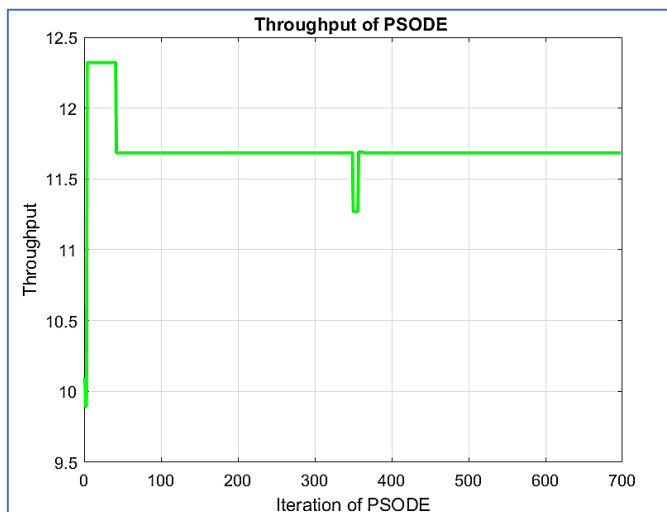
شکل ۱. کمینه‌سازی تابع هدف توسط الگوریتم PSODE

همانطور که در شکل فوق مشخص است، تابع هدف در الگوریتم پیشنهادی کاهش داشته است. همچنین، میزان تابع برازندگی (makespan) در این مطابق با شکل ۲ ارزیابی شده است.



شکل ۲. کمینه‌سازی makespan توسط الگوریتم PSODE

در شکل (۳)، میزان توان عملیاتی الگوریتم تحلیل شده است که بعد از حدوداً ۴۰ تکرار به مقدار ثابتی رسیده و روند افزایشی داشته است.



شکل ۳. ارزیابی توان عملیاتی توسط الگوریتم PSODE

مقادیر پارامترهای ارزیابی در تکرار نهایی الگوریتم مطابق با جدول ۲، محاسبه شده است. همچنین، نتیجه بهترین پاسخ به دست آمده در تکرار نهایی الگوریتم که همان نتیجه زمانبندی کارها بر روی ماشین‌های مجازی است به صورت جدول (۴) ارزیابی شده است.

جدول ۴. ارزیابی توان عملیاتی و برازندگی در تکرار نهایی

Fitness	MP	THr
۰.۷۱۰۷۷	۳۹۱	۹,۴۹۴۵

همچنین، نتیجه بهترین پاسخ به دست آمده در تکرار نهایی الگوریتم که همان نتیجه زمانبندی کارها بر روی ماشین‌های مجازی است به صورت جدول (۵) محاسبه شده است.

جدول ۵. نتیجه بهترین پاسخ در تکرار نهایی الگوریتم روش پیشنهادی

زمان اجرا	تعداد کارها	ماشین مجازی
۳۹۱	۸,۱۰۰,۲	۱
۳۷۰	۱,۳۰۵,۷	۲
۳۰۰	۴,۶,۹	۳

همانطور که در جدول فوق مشخص است، کارهای شماره ۲ و ۸ و ۱۰ به ماشین مجازی ۱ زمانبندی شدند که زمان اتمام این کارها برابر با ۳۹۱ میلی‌ثانیه شده است و کارهای شماره ۱ و ۳ و ۵ و ۷ به ماشین مجازی شماره ۲ تخصیص یافته و با زمان ۳۷۰ میلی‌ثانیه به اتمام می‌رسند و بقیه کارها به ماشین مجازی شماره ۳ زمانبندی شدند که زمان اتمام آنها برابر با ۳۰۰ میلی‌ثانیه شده است. در بررسی دیگر، تعداد ماشین‌های مجازی برابر ۳ و ۶ و همچنین تعداد کارها نیز به ترتیب برابر با ۳۰، ۵۰ و ۷۰ کار در نظر گرفته شده است که باید هر

یک از این مجموعه کارها بر روی آن ماشین‌های مجازی زمانبندی شوند. نتیجه این تحلیل در نمودار زیر نشان داده شده است.



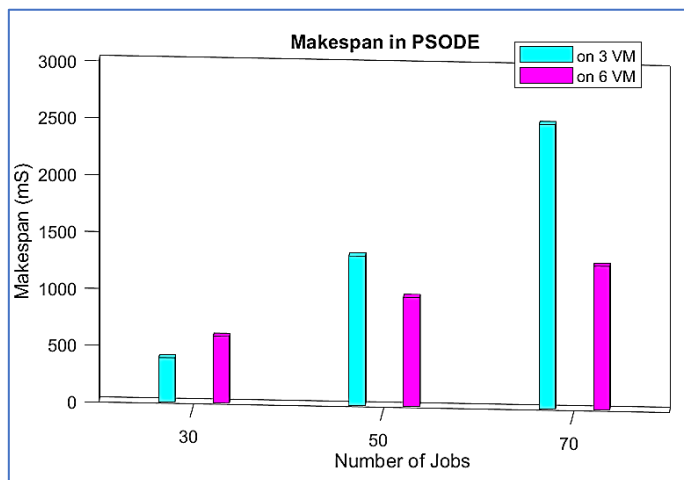
شکل ۴. میزان تابع هدف در عمل زمانبندی توسط روش پیشنهادی
بر روی تعداد کارها و ماشین‌های مجازی مختلف

در جدول (۶)، میزان تابع هدف در عمل زمانبندی توسط الگوریتم PSODE بر روی تعداد کارها و ماشین‌های مجازی مختلف نشان داده شده است.

جدول ۶. تحلیل تابع هدف در عمل زمانبندی الگوریتم PSODE

تعداد کارها تعداد ماشین مجازی	۷۰	۵۰	۳۰
۳ ماشین مجازی	۰.۶۹۱۶	۰.۶۲۲۳	۰.۷۱۰۸
۶ ماشین مجازی	۰.۸۸۳۶	۰.۸۸۵۸	۰.۹۲۷۴

همچنین در ادامه، نتایج تحلیل پارامتر makespan نیز در این شبیه‌سازی در شکل (۵) و نتایج تحلیل عددی این پارامتر در جدول (۷) نشان داده شده است.



شکل ۵. ارزیابی پارامتر makespan در عمل زمانبندی توسط روش پیشنهادی بر روی تعداد کارها و ماشین‌های مجازی مختلف

جدول ۷. تحلیل پارامتر makespan در عمل زمانبندی الگوریتم PSODE

تعداد کارها	
۷۰ ۵۰ ۳۰	
تعداد ماشین مجازی	
۳ ماشین مجازی	۲۵۰۱ ۱۳۱۵ ۳۹۱
۶ ماشین مجازی	۱۲۶۴ ۹۵۹ ۵۸۷

همچنین در ادامه، پارامتر توان عملیاتی مورد تحلیل و ارزیابی قرار گرفت که نتایج آن به ترتیب در شکل (۶) و جدول (۸) نشان داده شده است.

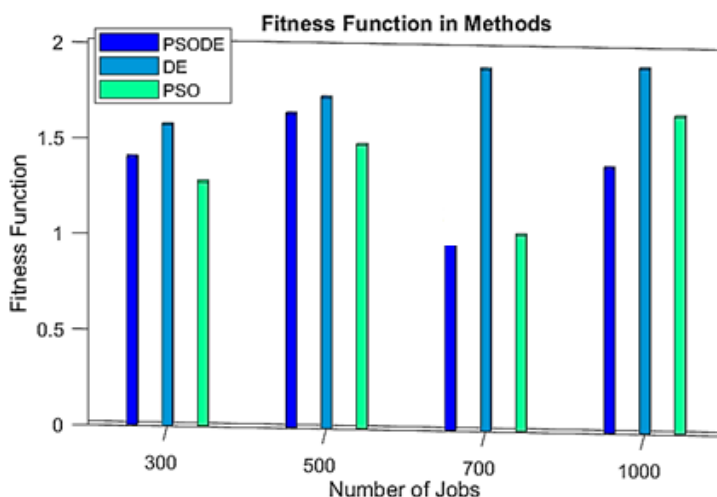


شکل ۶. ارزیابی توان عملیاتی در عمل زمانبندی توسط روش پیشنهادی
بر روی تعداد کارها و ماشین‌های مجازی مختلف

جدول ۸. تحلیل پارامتر توان عملیاتی در عمل زمانبندی الگوریتم PSODE

تعداد کارها	تعداد ماشین مجازی	۷۰ ۵۰ ۳۰
۳ ماشین مجازی	۱۰,۱۲۷۰	۱۲,۸۴۳۰ ۹,۴۹۴۵
۶ ماشین مجازی	۱۱,۷۰۳۰	۱۰,۶۰۸۰ ۱۴,۷۱۴۰

در بخش دوم، الگوریتم ارائه شده با الگوریتم‌های تکامل تفاضلی و الگوریتم هوش ازدحامی ذرات مورد مقایسه قرار گرفته است. به منظور این ارزیابی، تعداد ماشین‌های مجازی برابر با ۵۰ در نظر گرفته شده و تعداد کارها نیز به ترتیب برابر با ۳۰۰، ۵۰۰، ۷۰۰ و ۱۰۰۰ کار در نظر گرفته شده است که باید هر یک از این مجموعه کارها بر روی آن ماشین مجازی توسط الگوریتم PSODE و دو الگوریتم دیگر به صورت مجزا زمانبندی شوند. نتیجه این ارزیابی در شکل (۷) و به صورت عددی در جدول (۹) نشان داده شده است.



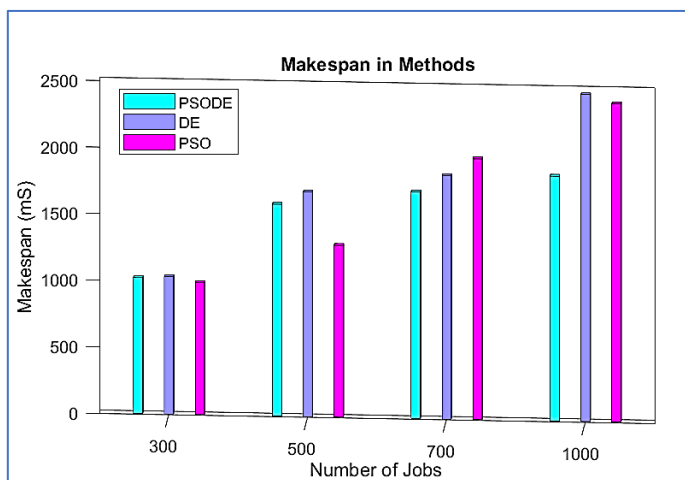
شکل ۷. ارزیابی تابع هدف در عمل زمانبندی توسط الگوریتم PSODE در مقایسه با الگوریتم‌های تکامل تفاضلی و هوش ازدحامی ذرات

جدول ۹. تحلیل تابع هدف در عمل زمانبندی الگوریتم‌های PSODE, DE, PSO

تعداد کارها	تعداد ماشین مجازی	۱۰۰۰ ۷۰۰ ۵۰۰ ۳۰۰
الگوریتم PSODE		۱,۳۸۳۶۱,۰۱۲۰ ۱,۶۳۷۹ ۱,۴۰۰۲
الگوریتم تفاضلات تکاملی (DE)		۱,۹۰۴۴ ۱,۸۸۷۲ ۱,۷۲۶۷ ۱,۵۷۱۴
الگوریتم هوش ازدحامی ذرات (PSO)		۱,۶۵۶۱ ۱,۰۲۴۰ ۱,۴۸۱۹ ۱,۲۷۴۶

نتایج مقایسه‌ای مقدار تابع هدف الگوریتم پیشنهادی با الگوریتم‌های تفاضل تکاملی و هوش ازدحامی ذرات را برای تعداد وظایف مختلف (۳۰۰، ۵۰۰، ۷۰۰ و ۱۰۰۰ وظیفه) و تعداد ثابت ۵۰ ماشین مجازی نشان می‌دهد. طبق این نتایج، الگوریتم PSODE در تمامی سناریوهای بررسی شده، مقادیر تابع هدف کمتری را در مقایسه با الگوریتم‌های DE و PSO ارائه کرده است. اگرچه الگوریتم PSO در برخی سناریوها عملکرد بهتری نسبت به DE داشته است، اما همچنان از نظر بهینه‌سازی و رسیدن به کمترین مقدار تابع هدف، نسبت به الگوریتم پیشنهادی PSODE عملکرد ضعیف‌تری دارد. بنابراین، می‌توان نتیجه گرفت که الگوریتم پیشنهادی به‌طور واضح از لحاظ بهبود مقدار تابع هدف و سرعت همگرایی، بر الگوریتم‌های مقایسه‌شده برتری دارد.

در ادامه، پارامتر makespan نیز در این شبیه‌سازی با الگوریتم‌های مختلف مورد ارزیابی قرار گرفته است که نتایج در شکل (۸) و به صورت عددی در جدول (۱۰) جمع‌آوری شده است.



شکل ۸. ارزیابی پارامتر makespan در عمل زمانبندی توسط الگوریتم PSODE, DE, PSO بر روی تعداد کارها و ماشین‌های مجازی مختلف

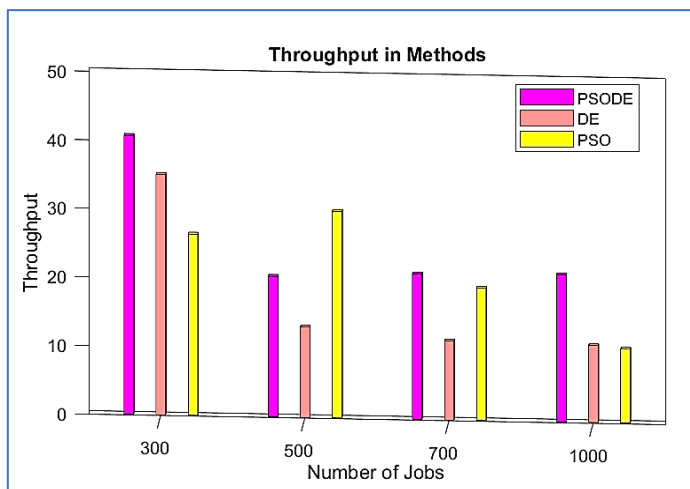
جدول ۱۰. تحلیل پارامتر makespan در عمل زمانبندی الگوریتم PSODE, DE, PSO

تعداد کارها	تعداد ماشین مجازی			
	۱۰۰۰	۷۰۰	۵۰۰	۳۰۰
روش پیشنهادی (PSODE)	۱۸۳۹	۱۷۰۲	۱۵۹۱	۱۰۲۰
الگوریتم تفاضلات تکاملی (DE)	۲۴۵۷	۱۸۲۹	۱۶۸۶	۱۰۳۱
الگوریتم هوش ازدحامی ذرات (PSO)	۲۳۹۱	۱۹۵۹	۱۲۹۰	۹۹۰

نتایج تحلیل پارامتر Makespan در الگوریتم پیشنهادی PSODE را برای تعداد وظایف متفاوت و ماشین‌های مجازی ارائه می‌دهد. بر اساس مقادیر این جدول، مشخص است که با افزایش تعداد ماشین‌های مجازی، مقدار Makespan به‌طور قابل توجهی کاهش یافته است. این امر نشان‌دهنده تأثیر مستقیم افزایش تعداد ماشین‌های مجازی در کاهش زمان پاسخ‌دهی کلی سیستم است و بیانگر کارایی الگوریتم پیشنهادی

PSODE در تخصیص بهینه وظایف به ماشین‌ها و کاهش زمان کلی اجرای وظایف در محیط محاسبات ابری است.

میزان توان عملیاتی در شبیه‌سازی توسط الگوریتم‌های PSO، PSPDE و DE مورد ارزیابی و تحلیل قرار گرفته است که نتایج در شکل (۹) و تحلیل عددی در جدول (۱۱) جمع‌آوری شده است.



شکل ۹. ارزیابی توان عملیاتی در عمل زمانبندی توسط الگوریتم‌های PSO، PSODE و DE بر روی تعداد کارها و ماشین‌های مجازی مختلف

جدول ۱۱. تحلیل پارامتر توان عملیاتی در عمل زمانبندی الگوریتم PSODE

تعداد کارها				تعداد ماشین مجازی
۱۰۰۰	۷۰۰	۵۰۰	۳۰۰	
۲۱,۴۴۳۰	۲۱,۱۷۸۰	۲۰,۴۷۵۰	۴۰,۶۴۶۰	الگوریتم PSODE
۱۱,۲۱۹۰	۱۱,۵۲۲۰	۱۳,۱۵۱۰	۳۵,۰۱۵۰	الگوریتم تفاضلات تکاملی (DE)
۱۰,۷۷۳۰	۱۹,۳۰۰۰	۳۰,۱۱۲۰	۲۶,۴۲۵۰	الگوریتم هوش ازدحامی ذرات (PSO)

همان‌طور که در این جدول مشاهده می‌شود، افزایش تعداد ماشین‌های مجازی باعث بهبود محسوس در توان عملیاتی سیستم شده است. این افزایش توان عملیاتی نشان‌دهنده کارایی الگوریتم PSODE در تخصیص بهینه وظایف و استفاده بهتر از ظرفیت منابع موجود در محیط ابری است.

۴-۱- سرعت همگرایی الگوریتم PSODE در مقایسه با الگوریتم‌های PSO و DE

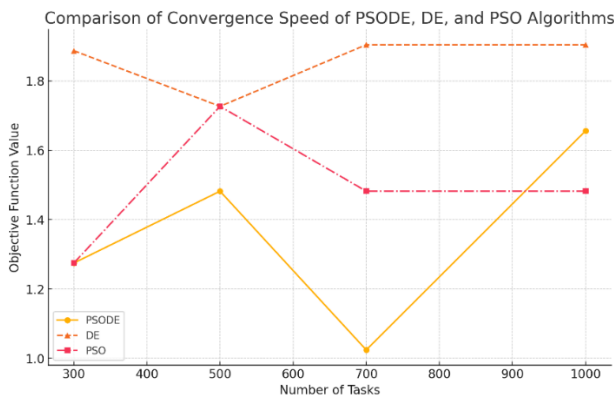
بر اساس نتایج ارائه شده در مقاله، الگوریتم پیشنهادی PSODE که از ترکیب الگوریتم تکاملی تفاضلی و الگوریتم هوش ازدحامی ذرات بهره می‌گیرد، در مقایسه با الگوریتم‌های DE و PSO به صورت جداگانه دارای سرعت همگرایی بهتری است. الگوریتم PSODE در طی حدود ۴۰ تکرار به مقادیر بهینه تابع هدف دست می‌یابد که نشان‌دهنده توانایی بالای این الگوریتم در کاهش سریع زمان پاسخ و افزایش توان عملیاتی در محیط ابری است. در مقابل، الگوریتم DE با سرعت همگرایی پایین‌تر، در تکرارهای بیشتری به مقادیر بهینه نزدیک

می شود و در وظایف با حجم بالا، عملکرد نسبتاً ضعیفی دارد. الگوریتم PSO نیز اگرچه در برخی موارد عملکرد بهتری نسبت به DE دارد، ولی همچنان در مقایسه با الگوریتم ترکیبی PSODE از نرخ همگرایی کندتری برخوردار است و به طور کلی نتایج نشان دهنده برتری واضح الگوریتم PSODE از لحاظ سرعت همگرایی و کاهش مقدار تابع هدف است. در جدول ارزیابی ارائه شده برای ۳۰۰، ۵۰۰، ۷۰۰ و ۱۰۰۰ کار با ۵۰ ماشین مجازی، مقادیر تابع هدف برای الگوریتم‌ها به شرح زیر است.

جدول ۱۲. مقایسه سرعت همگرایی الگوریتم‌ها

تعداد وظایف	الگوریتم PSODE	الگوریتم DE	الگوریتم PSO
۳۰۰	۱,۲۷۴۶	۱,۹۰۴۴	۱,۶۵۶۱
۵۰۰	۱,۴۸۱۹	۱,۴۸۱۹	۱,۸۸۷۲
۷۰۰	۱,۰۲۴۰	۱,۸۸۷۲	۱,۰۲۴۰
۱۰۰۰	۱,۶۵۶۱	۱,۹۰۴۴	۱,۴۸۱۹

الگوریتم PSODE به دلیل ترکیب هوشمندانده مزایای الگوریتم DE و PSO، سرعت همگرایی بالاتر و عملکرد بهتری در کاهش زمان پاسخ و افزایش توان عملیاتی نسبت به الگوریتم‌های DE و PSO دارد. این الگوریتم، سریع تر به نقطه بهینه دست می‌یابد، لذا برای استفاده در محیط‌های محاسبات ابری و وظایف با تعداد زیاد مناسب‌تر است. بنابراین، الگوریتم PSODE با ترکیب خصوصیات مطلوب الگوریتم‌های DE و PSO، عملکرد کلی بهتری از لحاظ سرعت همگرایی و بهینه‌سازی زمان پاسخ دارد.



شکل ۱۰. مقایسه سرعت همگرایی الگوریتم‌های PSO.DE.PSODE

شکل (۱۰) سرعت همگرایی الگوریتم پیشنهادی (PSODE) را با دو الگوریتم تفاضل تکاملی (DE) و هوش ازدحامی ذرات (PSO) مقایسه می‌کند. همانطور که مشاهده می‌شود، الگوریتم PSODE (خط پیوسته) بهترین عملکرد را در رسیدن سریع تر به مقدار بهینه (کاهش مقدار تابع هدف) نشان می‌دهد. الگوریتم DE (خط چین) کمترین سرعت همگرایی را داشته و در مقدار تابع هدف بالاتری قرار گرفته است. الگوریتم PSO (خط نقطه‌چین) عملکرد متوسطی دارد، ولی نسبت به DE بهتر و نسبت به PSODE ضعیف‌تر است.

بنابراین از نظر سرعت همگرایی و رسیدن به نقطه بهینه، الگوریتم ترکیبی PSODE عملکرد بهتری را داشته است.

۵- نتیجه‌گیری و پیشنهادات آتی

در این مقاله، یک الگوریتم زمان‌بندی کارآمد به منظور افزایش توان عملیاتی و کاهش زمان پاسخ ماشین‌های مجازی در محیط محاسبات ابری ارائه شد. بدین‌منظور، از الگوریتم‌های فرا ابتکاری تفاضل تکاملی و اجتماع ذرات استفاده شده است. این الگوریتم با مدل‌سازی ریاضی فرایند تکاملی اجتماعی سیاسی، برای حل گسترده وسیعی از مسئله‌های بهینه‌سازی استفاده شده است. این استراتژی تکاملی بهینه‌سازی، عملکرد بالایی در نرخ همگرایی و رسیدن به بهینه سراسری از خود نشان داده است. با استفاده از الگوریتم اجتماع ذرات بهترین کار انتخاب می‌شود و الگوریتم تفاضل تکاملی بهترین ماشین را انتخاب می‌کند تا کار توسط آن اجرا شود. نتایج ارزیابی الگوریتم PSODE نشان می‌دهد، روش پیشنهادی توانایی افزایش میزان بهره‌وری و کاهش زمان پاسخ را در محیط ابری به همراه دارد. استفاده از الگوریتم‌های تکاملی دیگر می‌تواند مفید باشد به‌گونه‌ای که با روش پیشنهادی ترکیب شوند و برای حل مسائل در حوزه‌های مختلف محاسبات ابری مورد استفاده قرار گیرند. همچنین استفاده سیستم فازی نیز می‌تواند در حل مسائلی که قطعیت کافی وجود ندارد در ترکیب با روش پیشنهادی مورد استفاده قرار گیرد. در کارهای آتی، برای گسترش این نوع تعادل بار و زمان‌بندی برای جریان کار با وظایف وابسته الگوریتم‌های جدید ارائه خواهد شد. این الگوریتم اولویت را به‌عنوان پارامتر اصلی کیفیت سرویس در نظر می‌گیرد.

فهرست منابع

- [1] Jin, Y. (2025). An effective method for prospective scheduling of tasks in cloud-fog computing with an energy consumption management approach based on Q-learning. *Engineering Applications of Artificial Intelligence*, 151, 110705. <https://doi.org/10.1016/j.engappai.2025.110705>.
- [2] Prity, F. S., Gazi, M. H., & Uddin, K. A. (2023). A review of task scheduling in cloud computing based on nature-inspired optimization algorithm. *Cluster computing*, 26(5), 3037-3067. <https://doi.org/10.1007/s10586-023-04091-x>.
- [3] Hu, Y., Pan, L., & Pan, X. (2024). Dynamic scheduling of workshop resource in cloud manufacturing environment. *Engineering Applications of Artificial Intelligence*, 138, 109405. <https://doi.org/10.1016/j.engappai.2024.109405>.
- [4] Long, S., Yang, J., Rao, C., Deng, Q., & Cao, K. (2025). D3QN-based secure scheduling of microservice workflows in cloud environments. *Computer Networks*, 263, 111227. <https://doi.org/10.1016/j.comnet.2025.111227>.
- [5] Dehghan, Motahera, & Khosrovian, Irfan. (1402). Privacy-preserving federated learning for advanced threat detection in drone internet systems. *Karafan Scientific Quarterly*, 20(3), 465-484. <https://doi.org/10.48301/kssa.2023.409787.2649>. (in persian)
- [6] De Peer, Mahmoud, Maarif, Armin, & Zoghi, Tektem. (1403). Presenting a multivariate hybrid routing method to improve the performance of vehicular networks. *Karafan Scientific Quarterly*, 21(1), 39-62. <https://doi.org/10.48301/kssa.2024.423334.2752>. (in persian)

- [7] Kamble, N., Saraf, V., Pachghare, V., & Adhao, R. (2025). A Trust Computation Model for Collaborative Cloud Environment. *Procedia Computer Science*, 258, 4169-4178. <https://doi.org/10.1016/j.procs.2025.04.667>.
- [8] Kchaou, H., Kechaou, Z., & Alimi, A. M. (2022). A PSO task scheduling and IT2FCM fuzzy data placement strategy for scientific cloud workflows. *Journal of Computational Science*, 64, 101840. <https://doi.org/10.1016/j.jocs.2022.101840>.
- [9] Pradhan, A., Bisoy, S. K., & Das, A. (2022). A survey on PSO based meta-heuristic scheduling mechanism in cloud computing environment. *Journal of King Saud University-Computer and Information Sciences*, 34(8), 4888-4901. <https://doi.org/10.1016/j.jksuci.2021.01.003>.
- [10] Ding, X., & Wu, J. (2019). Study on energy consumption optimization scheduling for internet of things. *IEEE Access*, 7, 70574-70583. <https://doi.org/10.1109/ACCESS.2019.2919769>.
- [11] Beloglazov, A., Abawajy, J., & Buyya, R. (2012). Energy-aware resource allocation heuristics for efficient management of data centers for cloud computing. *Future generation computer systems*, 28(5), 755-768. <https://doi.org/10.1016/j.future.2011.04.017>.
- [12] Lovász, G., Niedermeier, F., & De Meer, H. (2013). Performance tradeoffs of energy-aware virtual machine consolidation. *Cluster Computing*, 16, 481-496. <https://doi.org/10.1007/s10586-012-0214-y>.
- [13] Verma, A., Ahuja, P., & Neogi, A. (2008). pMapper: power and migration cost aware application placement in virtualized systems. In *ACM/IFIP/USENIX international conference on distributed systems platforms and open distributed processing* (243-264). Berlin, Heidelberg: Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-540-89856-6_13.
- [14] Gholipour, N., Arianyan, E., & Buyya, R. (2020). A novel energy-aware resource management technique using joint VM and container consolidation approach for green computing in cloud data centers. *Simulation Modelling Practice and Theory*, 104, 102127. <https://doi.org/10.1016/j.simpat.2020.102127>.
- [15] Gudivaka, R. K., Basani, D. K. R., Grandhi, S. H., Gudivaka, B. R., Gudivaka, R. L., & Kamruzzaman, M. M. (2025). A multilevel decentralized trust management-aware OS-GRU and S-fuzzy-based dynamic task offloading in block-chain enabled Edge-Cloud. *Sustainable Computing: Informatics and Systems*, 46, 101111. <https://doi.org/10.1016/j.suscom.2025.101111>.
- [16] Tang, C., Zhao, S., Chen, B., Lu, X., & Zhang, Q. (2024). A two-dimensional time-aware cloud service recommendation approach with enhanced similarity and trust. *Journal of Parallel and Distributed Computing*, 190, 104889. <https://doi.org/10.1016/j.jpdc.2024.104889>.
- [17] Shan, L., Sun, L., & Rezaeipanah, A. (2024). Towards a novel service broker policy for choosing the appropriate data center in cloud environments. *Computer Communications*, 228, 107939. <https://doi.org/10.1016/j.comcom.2024.107939>.
- [18] Wang, Z., Zhang, H., Chen, J., & Chen, H. (2024). An effective framework for measuring the novelty of scientific articles through integrated topic modeling and cloud model. *Journal of Informetrics*, 18(4), 101587. <https://doi.org/10.1016/j.joi.2024.101587>.

- [19] Shin, D., Kim, J., Pawana, I. W. A. J., & You, I. (2025). Enhancing cloud-native DevSecOps: A Zero Trust approach for the financial sector. *Computer Standards & Interfaces*, 103975. <https://doi.org/10.1016/j.csi.2025.103975>.
- [20] Jiang, Y., Perng, C. S., Li, T., & Chang, R. N. (2013). Cloud analytics for capacity planning and instant VM provisioning. *IEEE Transactions on Network and Service Management*, 10(3), 312-325. <https://doi.org/10.1109/TNSM.2013.013013.110181>.
- [21] Ramesh, D., Kolla, S. S., Naik, D., & Narvaneni, R. (2025). HGWO-MultiQoS: A hybrid grey wolf optimization approach for QoS-constrained workflow scheduling in IaaS clouds. *Simulation Modelling Practice and Theory*, 103127. <https://doi.org/10.1016/j.simpat.2025.103127>.