



مقایسه عملکرد الگوریتم PSO با الگوریتم GA و Cuckoo در یافتن بازه زمانی پویا در ابر

مرضیه رحمانی^۱، فاطمه سعادت‌جو^۲، ولی درهمی^۳

^۱ دانشکده فنی و مهندسی، دانشگاه علم و هنر، یزد

^۲ دانشکده برق و کامپیوتر، دانشگاه یزد، یزد

چکیده

در ابر برای استفاده از امکانات، بین کاربران و سرویس دهنده قراردادی به نام توافقنامه سطح سرویس بسته می‌شود. در این مقاله برای بررسی تخلف از توافقنامه سطح سرویس، روشی پیشنهاد شده است که هدف آن یافتن بازه‌های زمانی است که در آن بیشترین تخلف از توافقنامه، کشف شده است. برای تشخیص تخلف، روش الگوریتم بهینه‌سازی ازدحام ذرات (PSO)، الگوریتم ژنتیک (GA) و الگوریتم پرند فاخته (Cuckoo) با هدف پیدا کردن بازه زمانی با تعداد بیشتری از تخلف‌های کشف شده با حداقل سر بار زمانی مورد مقایسه قرار گرفته‌اند. نتایج نشان می‌دهد که الگوریتم مبتنی بر بهینه‌سازی ازدحام ذرات توانسته است نسبت به الگوریتم ژنتیک و الگوریتم فاخته تعادل بهتری بین سر بار زمانی و تعداد تخلفی برقرار نماید.

کلمات کلیدی: رایانش ابری، الگوریتم PSO، الگوریتم GA، الگوریتم Cuckoo، بازه زمانی پویا

تاریخچه مقاله:

تاریخ ارسال: ۹۷/۳/۳۰

تاریخ اصلاحات: ۹۷/۴/۳۰

تاریخ پذیرش: ۹۷/۵/۵

تاریخ انتشار: ۹۷/۵/۱۵

Keywords:

cloud computing
PSO algorithm
GA algorithm
Cuckoo Birding algorithm
dynamic time interval

A Comparison between the Performance of PSO Algorithm with GA and Cuckoo Algorithms in Finding Dynamic Time Interval in the Cloud

Marziyeh Rahmani^{1*}, Fatemeh Saadatjoo², Vali Derhami³

^{1,2}Department of Computer Engineering, Science and Art University, Yazd, Iran

²Faculty of Electrical and Computer Engineering, Yazd University, Yazd, Iran

Abstract

In Cloud To use facilities, a contract called the Service Level Agreement is signed between the users and the server. In this paper, with an aim to investigate violations of the service level agreement, a method has been proposed to find the time interval within which the highest violation of the agreement has been discovered. To detect an offense, the Particle Swarm Optimization (PSO) algorithm, the Genetic Algorithm (GA) and the Cuckoo Birding algorithm are compared with the purpose of finding the time interval with more violations detected within a minimum overhead time. The results show that the Particle Swarm Optimization algorithm has been able to maintain a better balance between overhead time and the number of violations compared to Genetic Algorithm or Cuckoo algorithm.

م. رحمانی، ف. سعادت‌جو، و. درهمی، مقایسه عملکرد الگوریتم PSO با الگوریتم GA و Cuckoo در یافتن بازه زمانی پویا در ابر، دوفصلنامه محاسبات و سامانه‌های توزیع شده، سال اول، شماره اول، ص ۵۹-۷۱، سال انتشار ۱۳۹۷.

روش ارجاع به مقاله:



۱ - مقدمه

زمانی ایستا^۵ و بازه زمانی پویا^۶؟ بازه زمانی ایستا بازه‌هایی هستند که مقادیرشان در طی اجرا برنامه ثابت می‌باشد و بازه زمانی پویا بازه‌هایی می‌باشند که مقادیرشان در طی اجرا برنامه متغیر است [8]. محققان بخاطر برخی از تنگناهای سیستم رایانش ابری اغلب نمی‌توانند در یک محیط ابر واقعی تحقیق کنند، پس شبیه‌سازی برای مدل کردن مکانیسم و ارزیابی نتایج ضروری می‌باشد [9]. تابع محاسباتی بر اساس الگو توافقنامه سطح سرویس تعریف شده‌اند این الگو حاوی اطلاعاتی از جمله خصوصیات خدمات، نام و مقادیر آن است [10].

در این مقاله با استفاده از الگوریتم PSO^۷ بهترین بازه زمانی پویا به عنوان خروجی پیشنهاد داده شده است بهترین بازه زمانی پویا بازه‌ای است که در این بازه تعادل بین تخطی از توافقنامه سطح سرویس و سربار زمانی سیستم برقرار باشد.

چالش اصلی و در واقع هدف در این مقاله که به دنبال پاسخ به آن هستیم یافتن بازه زمانی پویا با الگوریتم تکاملی PSO (بهینه سازی ازدحام ذرات) و سپس مقایسه آن با الگوریتم‌های GA (ژنتیک) و Cuckoo (پرنده فاخته) است که نشان دهیم کدام یک از این سه الگوریتم نتیجه بهتری در یافتن بهترین بازه زمانی پویا به دست می‌آورند.

ساختار این مقاله بصورت زیر است. در بخش دوم کارهای مرتبط با موضوع همراه با مزایا و معایب کارهای انجام شده در گذشته آورده شده است. در بخش

رایانش ابری^۱ فناوری جدیدی است [1] و امکاناتی فراهم می‌سازد تا کاربران به جای خرید چندین سرور، نیازهای محاسباتی خود را بر روی سرورهای ابری اجرا کرده و تنها به میزان استفاده خود هزینه پرداخت کنند و هرگاه با افزایش هزینه مواجه شدند، به میزان نیاز، منابع بیشتری را به خدمت بگیرند؛ به این تکنولوژی در ابر مجازی سازی گویند که توسط آن اشتراک پویای منابع سخت افزاری فراهم می‌گردد [2] با شروع از زیرساخت‌ها و پلت فرم‌های ابری می‌توان سخت افزارها و برنامه‌های نرم افزاری را به عنوان خدمات به کاربران ارائه داد. ارائه خدمات ابری برپایه توافقنامه سطح سرویس صورت می‌گیرد که این توافقنامه بخش بسیار مهمی به عنوان قرارداد بین ارائه دهنده ابر و مصرف کننده است [3,4]. توافقنامه سطح سرویس^۲ توصیفی از کیفیت خدمات سرویس^۳ است [5]. در صورت بروز خسارت و از دست رفتن داده‌ها وجود توافقنامه می‌تواند به کاربران ابر برای اثبات ادعای خود علیه سرویس‌دهنده ابر کمک کند [6]. اخیراً نیاز به امنیت در توافقنامه سطح سرویس پیشنهاد شده است [7].

در رایانش ابری، بارکاری^۴ مجموعه‌ای از فایل‌های متنی می‌باشد که شامل نمونه‌هایی از درصد استفاده از پردازنده در هر بازه زمانی است و داده‌هایش از بیش از ۵۰۰ مرکز داده در سراسر جهان جمع‌آوری شده است. بازه زمانی زمان بر حسب دقیقه می‌باشد که بر اساس این بازه تخطی‌های توافقنامه سطح سرویس کشف می‌شود. بازه‌های زمانی دو نوع می‌باشند بازه

^۴ workload

^۵ Static Interval

^۶ Dynamic Interval

^۷ Particle Swarm Optimization

^۱ Cloud Computing

^۲ Service Level Agreement

^۳ Quality of Service



در [12] برای مدیریت توافقنامه سطح سرویس درباره چارچوب^۹ اضافه شده به شبکه گرید بحث شده است. این توسعه توافقنامه سطح سرویس هر زمان که هدف، مدیریت توافقنامه سطح سرویس در ابر باشد، به شبکه گرید اعمال می‌شود.

در [13]، نظارت ابری بر روی برنامه با استفاده از روش mOSAIC^{۱۰} ارائه شده است. توسعه و فن‌های نظارتی بر برنامه‌های کاربردی با استفاده از API^{۱۱} mOSAIC توصیف شده و روی محیط ابری با هدف جمع‌آوری اطلاعات گسترش یافته است. در آن مقاله، کشف تخطی از توافقنامه سطح سرویس به منظور اجتناب از هزینه در نظر گرفته نمی‌شود و به‌علاوه عمومی نیست. چون تنها برنامه‌هایی را کنترل می‌کند که با استفاده از mOSAIC API، توسعه یافته‌اند.

در [14] درباره تأمین سرویس پویای توافقنامه سطح سرویس با استفاده از GRIA^{۱۲} بحث شده است. نظارت بر اساس موافقت توافقنامه سطح سرویس از مزایای این روش و عدم نداشت معیارهای سطح پایین به معیارهای سطح بالا و محدود بودن به شبکه گرید از معایب کار آن مقاله به شمار می‌رود.

در [15] تعریف فرآیندی برای اجرای توافقنامه سطح سرویس بدون چارچوب SOI^{۱۳} و توافقنامه سطح سرویس پروژه اتحادیه امریکا تعریف شده است. در دسترس بودن داده‌های تاریخی برای ارائه توافقنامه سطح سرویس و قابلیت ارزیابی بر شرایط

سوم توضیحی در رابطه روش پیشنهادی آورده شده است و در بخش چهارم به بررسی و ارزیابی نتایج حاصل از پژوهش پرداخته شده است و در نهایت در بخش پنجم نتیجه‌گیری و پیشنهادهای آینده آورده شده است.

۲ - کارهای مرتبط

روش‌های مختلفی برای کنترل و نظارت بر توافقنامه سطح سرویس در محیط ابر انجام شده است. هر کدام از این روش‌ها دارای مزایا و معایبی است. این روش‌ها به دو دسته کلی تقسیم می‌شوند. دسته اول روش‌هایی هستند که در طی اجرای برنامه با تغییر در محاسبات برنامه مقادیرشان ثابت و تغییر نمی‌کند این روش‌ها ایستا نام دارند. دسته دوم در طی اجرای برنامه مقادیرشان مدام در حال عوض شدن است و حالت ثابتی ندارند به این دسته پویا گفته می‌شود. در ادامه به بررسی برخی از این روش‌ها در قالب کارهای انجام شده در سایر مقالات و مزایا و معایب هر روش پرداخته شده است.

در [11] توسعه یک سیستم با فرآیند خودکار نظارت به نام Sandpiper در محیط ایستا ارائه شده است. امکان فرآیند خودکار نظارت و تشخیص نقاط مهم و تنظیم مجدد ماشین مجازی^۸ در زمان لازم و جلوگیری از تخطی از توافقنامه سطح سرویس از مزایای و عدم نداشت معیارهای سطح پایین از قبیل پردازنده و حافظه به پارامترهای توافقنامه سطح سرویس از معایب آن مقاله به شمار می‌رود.

¹¹ Application Programming Interface

¹² GRIA Is a Service Oriented Architecture framework

¹³ Silicon On Insulator

⁸ Virtual Machine

⁹ Framework

¹⁰ mOSAIC Is a Open Source API and Platform Multiple Cloud



انرژی که باعث پیدا کردن تخطی بیشتر توافقنامه سطح سرویس نسبت به روش های دیگر شده؛ از مزایای آن مقاله و عدم کاهش سربار زمانی سیستم و همچنین عدم پویا سازی از معایب آن مقاله به شمار می رود.

در [18] با استفاده از الگوریتم رقابت استعماری، تضاد منافع میان سربار سیستم و تخطی در محاسبات ابری جهت یافتن بازه های پویا در معماری کشف خودکار تخطی^{۱۴} را ارائه شده است. در آن مقاله یک روش دقیق با یک روش نا دقیق مورد مقایسه قرار گرفته که ذات این مقایسه صحیح نیست در ضمن عدم پیاده سازی در محیط کلودسیم^{۱۵} از معایب آن مقاله به شمار می رود.

در [1] الگوریتم جهش قورباغه برای یافتن بازه زمانی پویا ارائه شده است. برقراری تعادل نسبی بین کشف تخطی و سربار زمانی سیستم از مزایای آن مقاله به شمار می رود. اما مصرف زیاد حافظه از معایب آن الگوریتم به شمار می رود.

الگوریتم های تکاملی می توانند جهت انتخاب بازه های زمانی پویا به کار گرفته شوند. علت استفاده از الگوریتم های تکاملی آن است که تعداد بازه های زمانی که برای کشف تخطی می خواهیم بررسی شود به علت حجم بالای بار کاری پلنت لب^{۱۶} استفاده شده زیاد است و هر بار که بخواهیم بازه زمانی پویا به دست آوریم حجم کار عملاً بالا می رود و با استفاده از روش دستی قادر به محاسبه سربار زمانی و کشف تخطی نیستیم.

توافقنامه سطح سرویس ارائه شده از مزایا و عدم نظارت بر معیارهای سطح پایین و عدم نگاشت آن به مفاد سطح بالا از معایب آن مقاله به شمار می رود.

در [16] روش مبادله ابری ارائه شده است که عملکرد بهتر، کاهش هزینه در محیط ابری از مزایا و عدم پشتیبانی از زمان بندی پویا از معایب آن مقاله به شمار می رود.

در [10] یک رویکرد برای سازگاری قالب توافقنامه سطح سرویس کاربران با توافقنامه های عمومی ارائه شده است؛ از مزایای آن روش این است که محدودیت قالب توافقنامه ایستا با این قالب رفع می گردد و تعریف نگاشت بین توافقنامه عمومی و خصوصی برای کاربران انجام می شود؛ از طرفی تحمیل هزینه به کاربر، عدم مدیریت تخطی از توافقنامه سطح سرویس از معایب آن روش به شمار می رود.

در [6] معماری زیرساخت کشف خودکار تخطی از توافقنامه سطح سرویس ارائه شده است؛ این معماری به طور خودکار تخطی از توافقنامه سطح سرویس در بازه های ایستا را پیدا می کرد. به حداقل رساندن تعامل با کاربران و پیشگیری از تخطی از توافقنامه سطح سرویس از مزایای آن مقاله و افزایش سربار زمانی در بازه های زمانی کوتاه از معایب آن مقاله به شمار می رود.

در [17] روش کمترین زمان مهاجرت برای کاهش تخطی از توافقنامه سطح سرویس در بازه های ایستا به عنوان یک نیاز جهت بالا بردن کیفیت سرویس و راضی کردن مشتریان ارائه شده است. کاهش مصرف

¹⁶ Planet Lab

¹⁴ Desvi

¹⁵ Cloud sim



داده‌های پلنت لب تنها در بازه‌های زمانی که ضریب صحیحی از ۵ دقیقه می‌باشد؛ تعداد تخطی‌های کشف شده را گزارش می‌نمایند لذا برای روش کار ما مناسب نیستند. به همین دلیل نیاز به تولید یک بارکاری داریم که بتواند در هر بازه زمانی تعداد تخطی‌های کشف شده را مشخص نماید.

هر بارکاری برای یک شبانه روز یعنی ۱۴۴۰ دقیقه ارزیابی می‌شود. با توجه به بازه زمانی انتخابی m تعداد دفعاتی که تخطی سیستم ثبت می‌شود از معادله (۱) به دست می‌آید.

$$P = \frac{1440}{m}, 1 \leq m \leq 1440 \quad (1)$$

برای هر بازه زمانی m دقیقه‌ای، لازم است تا هزینه اندازه‌گیری که رابطه مستقیمی با طول بازه دارد و تعداد تخطی‌های کشف شده و بدنبال آن تعداد تخطی‌های کشف نشده محاسبه شود. جهت ایجاد تعادل بین هزینه اندازه‌گیری و تعداد تخطی کشف شده از معادله (۲) استفاده می‌شود [6].

$$C = \mu \times C_m + \frac{\sum_{\varphi \in \{CPU, Memory, Storage\}} \alpha(\varphi)}{\times C_p} \quad (2)$$

که در آن μ تعداد اندازه‌گیری‌ها، C_m هزینه اندازه‌گیری، $\alpha(\varphi)$ تعداد تخطی‌های کشف نشده توافقنامه سطح سرویس و C_p هزینه از دست رفتن تخطی توافقنامه است.

در رابطه (۲) تعداد اندازه‌گیری رابطه مستقیمی با زمان دارد. برای مثال اگر بازه‌های اندازه‌گیری ۵ دقیقه‌ای یک‌بار باشد و بارکاری ما برای یک شبانه روز (۲۴ ساعت) اجرا شود آنگاه تعداد

در این مقاله از الگوریتم بهینه‌سازی ازدحام ذرات بدلیل داشتن سرعت همگرایی بالا در رسیدن به نقطه بهینه محلی^{۱۷} استفاده شده است.

۳- الگوریتم پیشنهادی

چالش موجود در کشف تخطی از توافقنامه سطح سرویس، این است که هر اندازه بازه زمانی کوچک‌تر در نظر گرفته شود، تعداد تخطی کشف شده و سربار زمانی بیشتر و هر اندازه بازه زمانی بزرگ‌تر انتخاب گردد، تعداد تخطی کشف شده و سربار زمانی وارد شده به سیستم کمتر خواهد شد. هدف اصلی برقراری تعادل بین سربار زمانی سیستم و تعداد تخطی‌های کشف شده می‌باشد. سربار زمانی در بازه‌های ایستا به علت کشف بیشتر تخطی افزایش می‌یابد [6].

اگر رابطه مستقیمی بین سربار زمانی و تعداد تخطی‌ها وجود داشته باشد باعث افت کارایی سیستم می‌شود. برای رفع این مشکل به دنبال یافتن بازه‌های پویا هستیم. برای انتخاب بهترین بازه زمانی که بتواند تعادل فوق را برقرار نماید روشی در این مقاله ارائه می‌شود. این بازه‌های زمانی باید این خصوصیت را داشته باشند که تعادلی بین سربار زمانی سیستم و تعداد تخطی کشف شده برقرار نمایند.

برای انجام این کار ابتدا می‌بایست برای پارامتر حافظه توافقنامه سطح سرویس، یک بارکاری بر مبنای بارکاری پیش فرض کتابخانه کلودسیم تولید شود. این بارکاری، باید میزان تخطی‌ها را در هر بازه زمانی بر اساس دقیقه ارزیابی نماید. از آنجا که

¹⁷ Personal Best



روش بهینه‌سازی ازدحام ذرات (PSO)، از رفتار اجتماعی دسته پرندگان یا گروه ماهی‌ها در حین جستجوی غذا، برای هدایت جمعیت به منطقه امیدبخش در فضای جستجو الهام گرفته شده است [19].

هر پرنده به عنوان یکی از اعضا گروه از تجربه قبلی و یافته‌های سایر اعضا برای یافتن غذا بهره می‌برد. پایه اصلی الگوریتم بهینه‌سازی ازدحام ذرات همین تسهیم اطلاعات بین اعضا گروه است. در الگوریتم بهینه‌سازی ازدحام ذرات هر جواب مسئله، موقعیت یک پرنده در فضای جستجو است که آن را ذره می‌نامند. تمام ذره‌ها دارای یک مقدار شایستگی هستند که توسط تابع برازش^{۱۸} که هدف بهینه‌سازی می‌باشد به دست می‌آیند. همچنین هر ذره دارای مؤلفه‌ای به نام سرعت می‌باشد که مسیر حرکت آن را در فضای جستجو تعیین می‌کند.

الگوریتم بهینه‌سازی ازدحام ذرات با یک گروه از جواب‌های تصادفی شروع به کار می‌کند. سپس برای یافتن جواب بهینه در فضای مسئله با به‌روزر کردن موقعیت و سرعت هر ذره به جستجو می‌پردازد. هر ذره به صورت چندبعدی (بسته به طبیعت مسئله) با دو مقدار X_{id} و V_{id} که به ترتیب معرف مکان و سرعت مربوط به بعد d ام از i امین ذره هستند تعریف می‌شود. در هر مرحله از حرکت جمعیت، هر ذره با توجه به دو مقدار بهترین به‌روز می‌شود.

اولین مقدار بهترین جواب از لحاظ شایستگی است که تاکنون برای هر ذره به‌طور جداگانه به دست آمده است. این مقدار بهترین فردی است و $pbest$ نامیده می‌شود.

بازه‌های اندازه‌گیری ۱۴۴۰ دقیقه خواهد بود به همین ترتیب برای بازه‌های ۱۰ دقیقه یکبار تعداد اندازه‌گیری ۵۲۲ خواهد بود. طبیعی است این تعداد برای بازه‌هایی با طول بزرگتر مقدار کمتر و برای بازه‌ها با طول کوچکتر مقدار بزرگتر را ایجاد می‌کند. مجموع این پارامترها باید تعادل را بین هزینه اندازه‌گیری و هزینه تخطی کشف نشده برقرار سازد.

از آنجا که تعداد بازه‌های انتخابی m زیاد می‌باشد و هر بار باید کل بارکاری در هر بازه انتخابی در یک دوره ۲۴ ساعته ارزیابی شود لذا به دنبال یک انتخاب بازه زمانی هستیم که تعادل بین هزینه اندازه‌گیری و تعداد تخطی‌های کشف شده برقرار نماید. برای این منظور یک بازه زمانی m از بین بازه‌های زمانی انتخاب می‌شود که مقدار معادله (۲) را مینیمم نماید.

با توجه به مسائل بیان شده در بالا برای یافتن یک بازه زمانی از بین بازه‌های زمانی موجود نیاز به استفاده از یک الگوریتم تکاملی می‌باشد. در همین راستا از الگوریتم بهینه‌سازی ازدحام ذرات در این مقاله استفاده شده است. عدم گیر افتادن در بهینه محلی دلیل انتخاب این الگوریتم می‌باشد.

همانطور که توضیح داده شد نوآوری این مقاله در یافتن بهترین بازه زمانی است که بتواند بین هزینه اندازه‌گیری و تعداد تخطی‌های کشف نشده تعادل برقرار نماید؛ به عبارت دیگر یافتن بازه پویایی که با توجه به شرایط مسئله تعیین می‌گردد.

۴- الگوریتم بهینه‌سازی PSO



بزرگ باشد تا امکان یافتن جوابهای خوب در همان مراحل اولیه فراهم شود و در مراحل پایانی کوچک بودن w همگرایی بهتری را سبب می شود. الگوریتم بهینه سازی ازدحام ذرات که با استفاده از روابط (۳) و (۴) جمعیت خود را به روز کند، الگوریتم بهینه سازی ازدحام ذرات پایه یا استاندارد نامیده می شود [20].

۴-۱- نحوه تولید ذرات در PSO

برای بکارگیری الگوریتم بهینه سازی ازدحام ذرات در مسئله انتخاب بهترین بازه پویا در کشف تخطی نیاز به تعریف ذره می باشد. با توجه به تعداد پارامترهای توافقنامه سطح سرویس، هر ذره از پنج بخش سرعت، بهترین مقدار و سه پارامتر کشف تخطی تشکیل شده است. این پارامترها شامل حافظه، پردازنده و پهنای باند می باشد. از آنجایی که بارکاری مربوط به پلنت لب در بازه هایی که مضرب پنج باشند اجرا می شوند لذا برای روش پیشنهادی این مقاله قابل استفاده نخواهند بود. به همین دلیل بارکاری جدیدی بر مبنای پلنت لب ایجاد می کنیم که بتواند در هر بازه زمانی قابل اجرا و اندازه گیری باشد.

بارکاری فوق را در بازه های زمانی متفاوت اجرا می نماییم و بهره وری هر بارکاری را با توجه به تعداد ماشین های فیزیکی^{۱۹} و مجازی مشخص می نماییم.

برای محاسبه بهره وری^{۲۰}، مقدار تخصیص داده شده به هر job از بارکاری به پردازنده، حافظه و پهنای باند را با مقدار درخواستی هر job مقایسه می نماید. در صورت بروز اختلاف و عدم تخصیص درخواست job، کمبود درخواست در قالب تخطی از آن پارامتر محاسبه

مقدار بهترین دیگر که توسط الگوریتم بهینه سازی ازدحام ذرات به دست می آید، بهترین مقداری است که تاکنون توسط تمام ذره ها در میان جمعیت به دست آمده است این مقدار بهترین کلی است و $gbest$ نام دارد. پس از یافتن دو مقدار $pbest$ و $gbest$ هر ذره سرعت و مکان جدید خود را با دو رابطه (۳) و (۴) به روز می کند [19].

$$X_i(t+1) = X_i(t) + V_i(t+1) \quad (3)$$

$$A = B \times (C + D \times F) \quad (4)$$

$$A = V_{ij}(t+1)$$

$$B = w$$

$$C = V_{ij}(t)$$

$$D = c_1 r_{1i}(t) [pbest_{ij} - X_{ij}(t)]$$

$$F = c_2 r_{2i}(t) [gbest_{ij} - X_{ij}(t)]$$

به طوری که w وزن اینرسی c_1 و c_2 ضرایب شتاب و r_1 و r_2 اعداد تصادفی در بازه (۰ و ۱) می باشند. برای جلوگیری از واگرایی الگوریتم، مقدار نهایی سرعت هر ذره در بازه $[-V_{MAX}, V_{MAX}]$ محدود می شود c_1 و c_2 و w از پارامترهای الگوریتم بهینه سازی ازدحام ذرات هستند و همگرایی الگوریتم وابسته به مقدار این پارامترهاست.

معمولاً مقدار c_1 و c_2 عددی بین ۱/۵ تا ۲ می باشد ولی بهترین انتخاب $c_1 = c_2 = ۲/۰۵$ می باشد. و $0 \leq w < 1$ انتخاب می شود همگرایی شدیداً به مقدار w وابسته است و بهتر است به صورت دینامیک تعریف شود. در بازه (۰/۲-۰/۹) بدین ترتیب که به صورت خطی در طی روند تکامل جمعیت کاهش می یابد. در ابتدا باید

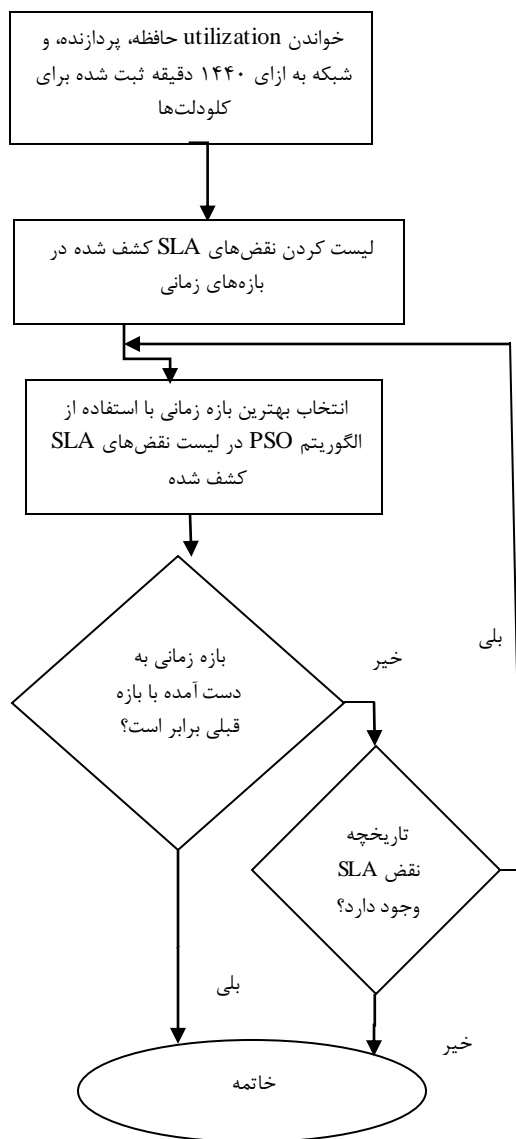
²⁰ Utilization

¹⁹ Host



است که اگر روز را ۲۴ ساعت در نظر بگیریم به ازای هر ۶۰ دقیقه در مجموع خواهیم داشت $24 \times 60 = 1440$ دقیقه است. یعنی هر روز معادل ۱۴۴۰ دقیقه است.

فرض می شود که مقدار بازه بر حسب دقیقه باشد. این فرض یعنی از محاسبه بر حسب ثانیه چشم پوشی شود چرا که به نظر می رسد که بارکاری تحمیل شده به سیستم برای بررسی ثانیه به ثانیه تغییرات، زیاد خواهد شد و محاسبات بر حسب دقیقه منطقی تر است.



(شکل-۱): روندنمای الگوریتم پیشنهادی

(Figure-1): Flowchart of the proposed algorithm in finding the dynamic interval

می گردد. زمان اجرای هر job در آن بازه نیز مشخص می گردد. برای استفاده از جواب‌های بدست آمده از اجرای بارکاری در هر بازه، لازم است تا بهره‌وری بدست آمده از هر job به همراه زمان اجرای متناظر با آن در یک map ذخیره گردد.

نقش map در بهنگام سازی هر ذره خواهد بود. بدین صورت که در ابتدا ذرات به صورت تصادفی تولید می شوند. سپس با توجه به مقداری که برای هر پارامتر کشف تخطی در ذره وجود دارد از طریق map، دوازده job متناظر با آن انتخاب می شود. بر اساس معادله (۲) مقدار تابع برازش مربوط به هر map انتخابی اجرا می گردد. میانگین مقدار تابع برازش برای دوازده map به عنوان بهترین مقدار آن ذره انتخاب می شود. این بهترین مقدار همان طول بازه مورد نظر خواهد بود.

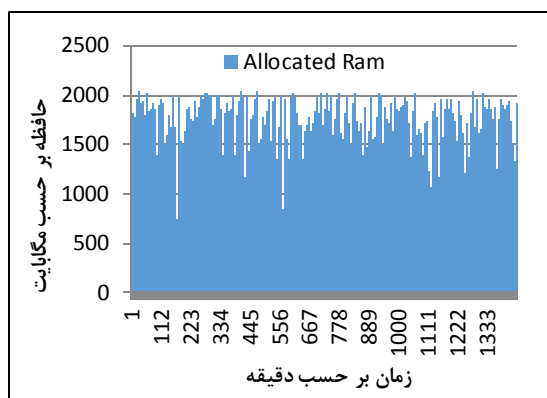
هر ذره براساس سرعت و بهترین مقدار خودش بهنگام می گردد. عملیات تا زمانی که مقدار معادله (۲) برای بهترین ذره در هر بار تکرار الگوریتم از مقدار آستانه کمتر نشود ادامه می یابد. پس از پایان الگوریتم، بهترین ذره انتخاب شده و بهترین مقدار آن به عنوان بازه پویا برای بارکاری تولیدی معرفی خواهد شد. لازم به توضیح است که بهترین ذره در هر بار اجرای الگوریتم، ذره‌ای است که بهترین مقدارش در بین چهار ذره بیشترین باشد. شکل ۱ روندنمای الگوریتم پیشنهادی را نشان می دهد.

۵ - تنظیم پارامترها و نتایج و ارزیابی

به منظور ارزیابی الگوریتم بهینه سازی ازدحام ذرات جهت بهره‌وری از حافظه، پردازنده و شبکه از داده‌های آزمایشگاه پلنت لب بصورت تصادفی ۱۴۴۰ داده به عنوان بارکاری انتخاب گردید. این داده‌ها به این صورت



برای بررسی میزان تخطی های توافقنامه سطح سرویس در الگوریتم بهینه سازی ازدحام ذرات با درخواست کاربر از حافظه فرض بر این گرفته شده است که درخواست کاربر از حافظه حداقل ۷۰۰ مگابایت بوده همچنان که در شکل (۲) ملاحظه می کنید در دقیقه های مختلفی از یک روز مقادیر متفاوتی از حافظه توسط میزبان به کاربر اختصاص داده شده است.



(شکل-۲): حافظه اختصاص داده شده به میزبان

(Figure-2): Memory allocated to host

شکل (۳) رابطه بین بازه های زمانی و تخطی های توافقنامه سطح سرویس را نشان می دهد. این نمودار مشخص می کند که در اکثر موارد، در بازه زمانی کوچکتر، تخطی بیشتری کشف شده است. به عنوان مثال برای بازه زمانی ۵ دقیقه مقدار ۱۳۰ مگابایت برای حافظه به دست آمده است. این مقادیر قبل از این که به الگوریتم بهینه سازی ازدحام ذرات داده شود به دست آمده است.

در شکل (۴) رابطه بین زمان مصرفی جهت کشف تخطی و بازه های زمانی نشان داده شده است. همانطور که از شکل (۴) پیداست؛ زمان مصرفی برای کشف ۱۳۰ تخطی، برابر ۷۰۵۶۳۴۳ نانوثانیه می باشد.

همانطور که قبلا بیان شد قبل از اعمال الگوریتم PSO، نیاز به تولید map می باشد تعداد ذرات در هر اجرای الگوریتم، چهار و مقدار آستانه ۰.۵ در نظر گرفته شده است.

در الگوریتم PSO بعد از اجرای یکبار الگوریتم، تمام ذرات بر اساس مقدار تابع برازششان به صورت نزولی مرتب می شوند. بر اساس سرعت هر ذره، عمل بهنگام سازی ذرات صورت می پذیرد و جهت حرکت آنها مشخص می شود.

خروجی الگوریتم PSO که در واقع بهترین بازه زمانی پیشنهاد شده است از دو جهت قابل ارزیابی است. یکی از جنبه کشف بیشترین تعداد تخطی ها که به ازای بازه های زمانی صورت گرفته است.

جنبه دوم ارزیابی که مهم تر هم هست سر بار زمانی تحمیل شده به سیستم است که در بعضی از بازه های زمانی کاهش یافته است.

برای شبیه سازی این مقاله از کتابخانه کلودسیم نسخه ۳،۰،۳ استفاده شده است. این شبیه ساز در محیط نرم افزار نت بینز^{۲۱} اجرا شده است. در محیط شبیه سازی پنجاه دستگاه فیزیکی طراحی و در هر دستگاه پنجاه ماشین مجازی طراحی شده است جدول (۱) مشخصات ماشین فیزیکی و مجازی انتخابی را نشان می دهد.

(جدول-۱): مشخصات ماشین های فیزیکی و مجازی

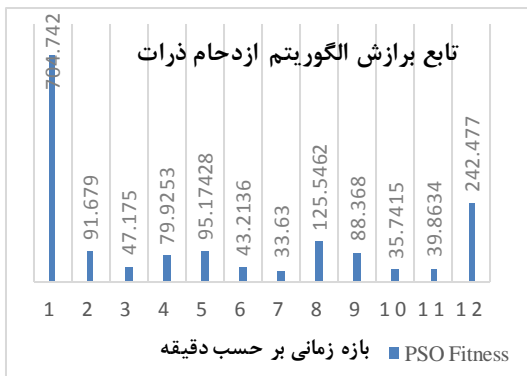
(Table-1): Features of the selected physical and virtual machines

حافظه RAM	پردازنده	سیستم عامل	
۴ گیگابایت	۴ هسته ای	Linux	ماشین فیزیکی
۱ گیگابایت	۱ هسته ای	Ubuntu	ماشین مجازی

²¹ NetBeans IDE 8.1



بازه زمانی هفت بدست آمده است. شکل ۵ دلیل انتخاب این بازه را نشان می‌دهد. در این شکل، محور افقی نمایانگر بازه زمانی و محور عمودی مقدار تابع هزینه را نمایش می‌دهد، همانطور که در شکل پیداست در بازه زمانی هفت کمترین مقدار از معادله (۲) بدست آمده است به عبارت دیگر تعادل بین تعداد تخطی کشف شده و زمان اجرا بدست آمده است.



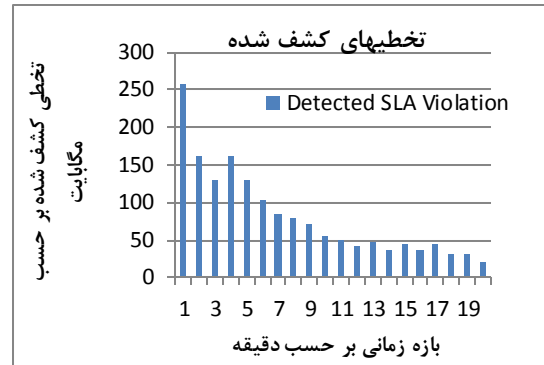
(شکل-۵): تابع برازش الگوریتم ازدحام ذرات

(Figure-5): Fitness function value in the PSO algorithm

علت انتخاب دوازده بازه، این است که در بازه‌های بزرگتر از دوازده تعداد تخطی‌های کشف شده به شدت کاهش می‌یابد و به دنبال آن تعداد تخطی‌های کشف نشده افزایش پیدا می‌کند. همین امر باعث افزایش مقدار تابع برازش در آن بازه‌ها خواهد شد.

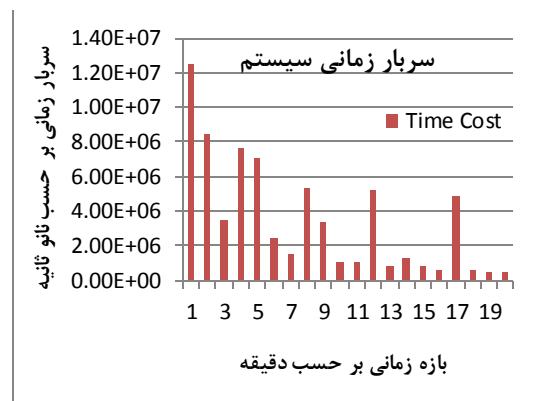
جهت بررسی کارایی الگوریتم بهینه سازی ازدحام ذرات آزمایش دیگری با الگوریتم‌های ژنتیک (GA) و پرندۀ فاخته (Cuckoo) انجام شده است. محیط این آزمایش‌ها با محیط الگوریتم بهینه سازی ازدحام ذرات یکسان در نظر گرفته شده است. بدین منظور ورودی الگوریتم ژنتیک و پرندۀ فاخته همان ورودی الگوریتم بهینه سازی ازدحام ذرات می‌باشد. از طرفی تابع برازش نیز همان معادله (۲) در نظر گرفته شده است.

هدف الگوریتم بهینه سازی ازدحام ذرات به دست آوردن بازه‌ی زمانی است که در آن تخطی توافقتنامه سطح سرویس بیشتر و سربار زمانی اعمالی به سیستم کمتر شود.



(شکل-۳): تخطیهای کشف شده از بازه زمانی

(Figure-3): SLA violations detected in time intervals



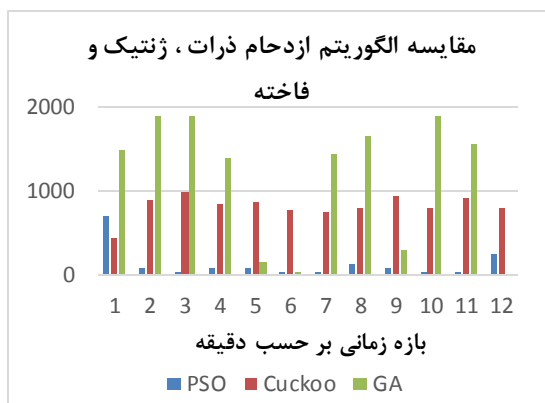
(شکل-۴): سربار زمانی تحمیل شده به سیستم

(Figure-4): Time overhead applied to the system

همانطور که قبلا بیان شد map تولیدی شامل میزان تخطی‌های متناظر با پارامترهای توافقتنامه سطح سرویس و زمان یافتن آنها در بازه مشخص شده برای یک شبانه‌روز می‌باشد که به عنوان ورودی به الگوریتم PSO داده می‌شود. الگوریتم فوق بر اساس تاریخچه map تصمیم می‌گیرد که در چه بازه‌ای تعادل بین مدت زمان اجرا و تعداد تخطی‌های کشف شده برقرار شده است. بعد از اتمام اجرای الگوریتم بهترین ذره با



شکل ۸ مقایسه‌ی روش مبتنی بر الگوریتم ازدحام ذرات و الگوریتم ژنتیک و الگوریتم پرندۀ فاخته را نشان می‌دهد. هدف هر سه روش یافتن مینیمم سراسری در صورت امکان است و اینکه سربار زمانی تحمیل شده به سیستم را کاهش دهند و تعداد تخطی بیشتری پیدا کنند.



(شکل-۸): مقایسه الگوریتم ازدحام ذرات و ژنتیک و فاخته

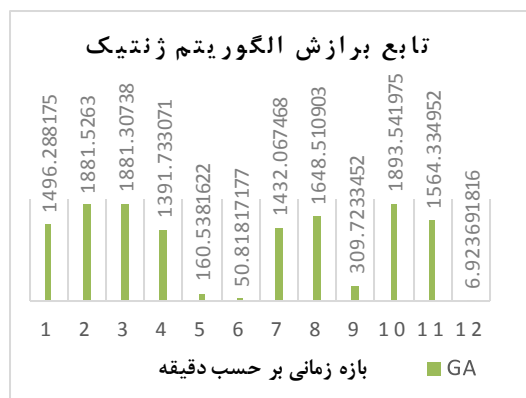
(Figure-8): Comparing the PSO and the GA and the Cuckoo algorithms

۶- نتیجه گیری و پیشنهادهای آینده

در محیط محاسبات ابری، امکانات خوب و مناسبی مانند پردازنده، سیستم‌های ورودی و خروجی، پهنای باند، حافظه و غیره به صورت منابع ناهمگن برای کاربران فراهم می‌شود. در این مقاله سه روش الگوریتم بهینه‌سازی ازدحام ذرات، الگوریتم ژنتیک و الگوریتم فاخته برای یافتن بازه زمانی پویا جهت کشف بیشترین تعداد تخطی از توافقنامه سطح سرویس مورد مقایسه قرار گرفتند، بستر آزمایش برای هر سه روش یکسان در نظر گرفته شده است. هدف هر سه روش کاهش سربار زمانی سیستم و کشف بیشترین تعداد تخطی می‌باشد.

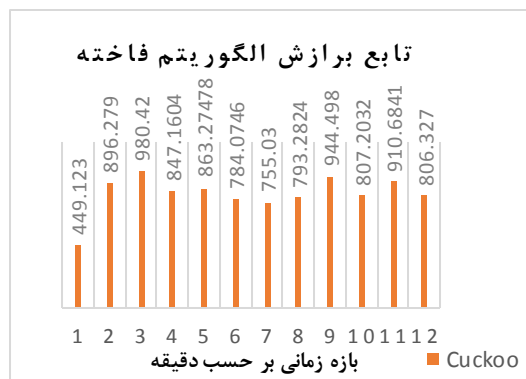
همان‌طور که از نتایج به دست آمد الگوریتم بهینه‌سازی ازدحام ذرات در یافتن بازه زمانی پویا و ایجاد تعادل بین سربار زمانی سیستم و کشف تخطی از

همان‌طور که در شکل ۶ نشان داده شده است الگوریتم ژنتیک در بازه ششم کمترین مقدار تابع برازش را دارا می‌باشد و در شکل ۷ الگوریتم پرندۀ فاخته در بازه یکم کمترین مقدار تابع برازش را دارا می‌باشد. در حالت کلی با مقایسه بین مقادیر تابع برازش سه الگوریتم بهینه‌سازی ازدحام ذرات، ژنتیک و پرندۀ فاخته مشخص می‌شود که الگوریتم بهینه‌سازی ازدحام ذرات توانسته است تعداد تخطی بیشتری را در بازه‌های زمانی نسبت به الگوریتم ژنتیک و پرندۀ فاخته بیابد و پرندۀ فاخته تعداد تخطی بیشتری نسبت به الگوریتم ژنتیک بیابد چرا که مقدار معادله (۲) زمانی بیشتر خواهد شد که حداقل تعداد تخطی‌های کشف نشده بیشتر باشد.



(شکل-۶): تابع برازش الگوریتم ژنتیک

(Figure-6): Fitness function value in the GA algorithm



(شکل-۷): تابع برازش الگوریتم فاخته

(Figure-7): Fitness function value in the Cuckoo algorithm



Base.Future Generation Computer
Systems", Vol.78, pp.142-154, 2018.

[5] Hussain, W., Hussain, F.k., Hussain, O., Bagia, R., Change, E., Risk- based Framework for SLA Violation abatment from the Cloud Servise Provider's perspective.The Computer Journal,2018.

[6] Emeakaroha,V.C., Netto, M.A.S., Calheiros, R.N., Brandic, I., Buyya, R., De Rose, C.A, "Towards autonomic detection of SLA violations in Cloud infrastructures,Future Generation Computer Systems", vol.28, no.7, pp. 1017-1029,2012.

[7] Rojas, M.A.T., Redigolo.F.F., Gonzalez, N.M., Sbampato, F. V., de Brito Carvalho, T.C.M., Ullah, K.W., Ahmad,A.S, "Managing the Lifecycle of Security SLA Requirements in Cloud Computing, In Developments and Advances in Intelligent Systems and Applications, Springer", Cham, pp.119-140,2018.

[8] White, M., Melvin, H., Schukat, M., The Impact of Dynamic Monitoring Interval Adjustment on Power Consumption in Virtualized Data Centers,2014.

[9] Long, W., Yuqing, L., Qingxin, X., "Using cloudsim to model and simulate cloud computing environment", In Computational Intelligence and Security, 9th International Conference on IEEE, pp. 323-328, 2013.

[10] Maurer, M., Emeakaroha, V.C., Brandic, I., Altmann, J., "Cost-benefit analysis of an SLA mapping approach for defining standardized Cloud computing goods", Future Generation Computer Systems,vol.28, no.1, pp. 39-47,2012.

[11] Wood, T., Shenoy, P., Venkataramani, A., Yousef, M., "Sandpiper: Black-box and gray-box resource management for virtual machines," Computer Networks, vol.53, no.17, pp.2923-2938, 2009.

[12] Micsik, A., Frutos, H.M., Kotsiopoulos, I., Koller, B., "Semantically supported SLA negotiation," 10th IEEE/ACM International Conference on Grid Computing, pp.169-170, 2009.

[13] Petcu, D., Di Martino, B., Venticinque, S., Rak, M., Máhr, T., Lopez, G.E., Stankovski, V., "Experiences in building a mOSAIC of clouds,"Journal of Cloud Computing: Advances, Systems and Applications,vol.2, no.1, pp. 1-12, 2013.

[14] Boniface, M., Phillips, S., Sanchez-Macian, S., Surrudge, M., "Dynamic service provisioning using GRIA SLAs," In Service-Oriented Computing-ICSO Workshops, pp.56-67, 2009.

[15] Comuzzi, M., Kotsokalis, C., Spanoudkis, G., Yahyapour, R., "Establishing and monitoring SLAs in

توافقنامه سطح سرویس بهتر از الگوریتم ژنتیک و فاخته به جواب رسیده است و الگوریتم فاخته بهتر از الگوریتم ژنتیک عمل کرده است و توانسته اند تعداد تخطی بیشتری را در بازه زمانی بیابند چرا که مقدار تابع هزینه زمانی بیشتر خواهد شد که حداقل تعداد تخطی های کشف نشده بیشتر باشد.

به عنوان چشم انداز آینده می توان الگوریتم بهینه سازی ازدحام ذرات بهبود داده شود. همچنین می توان با استفاده از مهاجرت ماشین های مجازی و خوشه بندی در محیط های توزیع شده همانند اسپارک^{۲۲} و هادوپ^{۲۳} میزان سربار زمانی و تخطی از توافقنامه را کاهش دهیم.

۷- مراجع

[1] رحیمی، ز؛ سعادت جو، ف؛ اسمعیلی زینی، ع.م: «به کار گیری الگوریتم جهش قورباغه در یافتن تخطی ها در کنترل تعهدات SLA در محاسبات ابری». سومین کنفرانس بین المللی وب پژوهی، سوم، تهران، دانشگاه علم و فرهنگ، ۱۳۹۶.

[1] Rahimi, z., Saadatjoo, F., Esmaeili Zeini, A.M., " Using the Frog Mutant Algorithm in Finding Violations in Controlling SLA Obligations in Cloud Computing", The third International Conference on Web Research, the Third, Tehran, University of Science and Culture, 2017.

[2] جمالی، ش؛ ملک تاجی، س ؛ آنالویی، م: «مکان یابی ماشین های مجازی با استفاده از الگوریتم رقابت استعماری»، مجله مهندسی برق دانشگاه تبریز: شماره ۱، جلد ۴۶، ۶۲-۵۳، بهار ۱۳۹۵.

[2] Jamali, Sh, Malek Taji, S, Analuei, M., " Positioning the virtual machines using the colonial competitive Algorithm", Electrical Engineering Magazine, Tabriz University, issue 1, Volume 46, 53-62, Spring 2016.

[3] Ghowdhury,A. G., Das, A., "Importance of SLA in Cloud Computing", In Big Data Analyatics, pp.141-147,2018.

[4] Bellini, P., Bruno,I., Cenni, D., Nesi, P., "Managing Cloud Via Smart Cloud Engine and Knowledge

²³ Hadoop

²² Spark



فاطمه سعادت جو در حال حاضر
استادیار و عضو هیئت علمی گروه
مهندسی کامپیوتر دانشگاه علم و هنر
یزد می باشد و مدرک کارشناسی و
کارشناسی ارشد و دکترای خود را از

دانشگاه یزد دریافت کرده است. زمینه‌ی پژوهشی مورد علاقه
ایشان مهندسی نرم افزار، پایگاه داده و داده کاوی و سیستم‌های
فازی می باشد. نشانی رایانامه ایشان عبارت است از:

رایانامه : saadatjou@sau.ac.ir



ولی درهمی در حال حاضر دانشیار
و عضو هیئت علمی گروه مهندسی
کامپیوتر دانشگاه یزد می باشد. ایشان
مدرک کارشناسی خود را از دانشگاه
صنعتی اصفهان و کارشناسی ارشد و

دکتری را از دانشگاه تربیت مدرس تهران دریافت کرده است.
زمینه پژوهشی مورد علاقه ایشان یادگیری ماشین، یادگیری
تقویتی، رباتیک، سیستم‌های چندعامله و سیستم‌های فازی می
باشد. نشانی رایانامه ایشان عبارت است از:

رایانامه : vderhami@yazd.ac.ir

complex service based systems," Proceedings of the 7th International Conference on Web Services, pp.783-790, 2009.

[16] Tordsson, J., Montero, R.S., Moreno-Vozmediano, R., Llorente, I.M, "Cloud brokering mechanisms for optimized placement of virtual machines across multiple providers", Future Generation Computer Systems, vol.28, no.2, pp. 358-367,2012.

[17] Saravanan, S., Venkatachalam, V., Malligai, S.T., "Optimization of SLA violation in cloud computing using artificial bee colony, International Journal". Adv. Eng, vol.1, no.3, pp. 410-414,2015.

[18] Ghafari, N., Saadatjoo, F., " Find the Best Time Intervals in the Control of Service Level Agreement Commitments in Cloud Computing Using Colonial Competitive Algorithm", International Science and Investigation journal, vol.4, no.4, pp. 20-35,2015.

[19] Mokarram, V., Banan, M.R., "A new PSO-based algorithm for multi-objective optimization with continuous and discrete design variables". Structural and Multidisciplinary Optimization. vol.57, no.2, pp.509-533.2018.

[20] Kennedy, J., Eberhart, R.C., "A discrete binary version of the particle swarm algorithm", International Conference on Computational Cybernetics and Simulation, vol.5, pp.4104-4108,1997.



مرضیه رحمانی مدرک کارشناسی را
در رشته مهندسی تکنولوژی نرم افزار
از موسسه آموزش عالی پویا یاسوج
دریافت کرده است و اکنون دانشجوی
کارشناسی ارشد در رشته مهندسی

نرم افزار در دانشگاه علم و هنر یزد می باشد. زمینه‌ی پژوهشی
مورد علاقه ایشان رایانش ابری و سیستم‌های توزیع شده
می باشد. نشانی رایانامه ایشان عبارت است از:

رایانامه : Rahmani69yazdstudent@Gmail.com