بهار ۱۴۰۰، دوره ۶، شماره ۱، پیاپی ۲۰
DOR :20.1001.1.2538161.1400.6.1.6.4

تاریخ دریافت مقاله: ۹۹/۰۲/۰۴
تاریخ پذیرش مقاله: ۹۹/۰۶/۰۱

نوع مقاله: پژوهشی

ترکیب سرویس‌های آگاه از کیفیت سرویس مبتنی بر الگوریتم PCA

بتول زارع
دانشجوی کارشناسی ارشد، گروه مهندسی کامپیوتر، واحد یزد، دانشگاه آزاد اسلامی، یزد، ایران
پست الکترونیکی: batool.zare@gmail.com

سیما عمادی*
گروه مهندسی کامپیوتر، واحد یزد، دانشگاه آزاد اسلامی، یزد، ایران
پست الکترونیکی: emadi@iauyazd.ac.ir

چکیده

در سال‌های اخیر، علاقه به ترکیب سرویس وب، با توجه به اهمیت آن در کاربردهای علمی افزایش‌یافته است. هنگامی‌که هیچ سرویسی به‌تنهایی برای برآوردن نیاز یک درخواست‌کننده سرویس توانایی ندارد، ترکیب سرویس، به‌منظور تحقق هدف درخواست‌کننده سرویس می‌تواند مفید باشد. بنابراین، مسئله چگونگی ترکیب مؤثر و کارآمد سرویس‌های وب، زمینه‌های تحقیقی بسیاری را به خود جلب کرده است. یکی از مسائل چالش‌برانگیز و دانش‌بنیان مهندسی نرم‌افزار، ترکیب سرویس با آگاهی از کیفیت سرویس است. استانداردهای کیفیت سرویس وب بر روی ویژگی‌هایی مانند زمان پاسخگویی، درجه اطمینان سرویس یا قابل‌اعتماد بودن، امنیت و غیره تمرکز نموده‌اند. بنابراین ترکیب سرویس وب آگاه از کیفیت، به‌عنوان یک مسئله بهینه‌سازی چندهدفه بررسی می‌شود. از طرفی، با افزایش تعداد سرویس‌ها، اندازهٔ فضای ترکیب به‌سرعت رشد نموده و زمان اجرا و حافظه مصرفی فرآیند ترکیب افزایش می‌یابد. بنابراین کارایی الگوریتم ترکیب سرویس برای رفع این مشکل به یک مسئله چالش‌برانگیز تبدیل‌شده است. در تحقیق پیش رو، با استفاده از الگوریتم PCA، ویژگی‌های کیفیت سرویس ادغام می‌شوند و با این روش، داده‌های مفید کمتری از دست می‌رود و با انجام عملیات اعتبارسنجی و هرس در سطح سرویس، فضای حافظه کمتری مصرف می‌شود. همچنین، ترکیب سرویس که یک مسئله چند هدفه است، به مسئله ترکیب مؤثر و کارای تک‌هدفه تبدیل می‌شود. به‌علاوه ارزیابی این طرح پیشنهادی نشان می‌دهد که روش پیشنهادی در مقایسه با روش ترکیب سرویس با الگوریتم ترکیب جزئی، زمان پاسخ و فضای حافظه مصرفی را کاهش داده و کارایی ترکیب سرویس را بهبود می‌دهد.

واژه‌های کلیدی: وب‌سرویس، کیفیت سرویس، ترکیب سرویس، PCA.

۱-مقدمه

در بسیاری از مواقع، یک سرویس وب منفرد به‌تنهایی نمی‌تواند نیازهای کاربران را برطرف کند و سرویس‌های

* نویسندهٔ مسئول

وب مختلف برای رسیدن به هدف باید با یکدیگر ترکیب شوند. با افزایش تعداد و مقیاس سرویس‌ها، بسیاری از ارائه‌دهندگان، عرضه سرویس‌های نامزد را آغاز نموده‌اند که ازلحاظ کارکرد معادل هستند، امّا دارای سطوح کیفیت سرویس متفاوت می‌باشند. این امر منجر به یکی از چالش‌های اصلی در زمینه ترکیب سرویس می شود. زیرا، در فرایند ترکیب سرویس بایستی پارامترهای کیفی که بسته به نیاز کاربران متفاوت است و همچنین تفاهم‌نامه سطح سرویس[1]، نیز در نظر گرفته شود [۱]. در نگاه اول انتخاب بهترین سرویس‌ها بر مبنای پارامترهای کیفیت سرویس[2] امر ساده‌ای به نظر رسد، کافی است سرویس‌های نامزد را با یکدیگر مقایسه کرده و آن سرویسی که بهترین مقادیر را دارا باشد انتخاب شود. امّا در واقعیت انتخاب سرویس‌های بهینه، فرایند سخت و پیچیده است. زیرا با توجه به فاکتورهای مختلفی، پارامترهای کیفیت سرویس یک سرویس برای کاربران مختلف مقدار متفاوتی خواهد داشت.

از طرفی ترکیب‌های موجود با رعایت تمام پارامترها امکان‌پذیر نیست زیرا در هر محیط پارامترهای کیفیت سرویس متفاوت که برخی باهم ناسازگارند وجود دارد. همچنین، در محیط واقعی سرویس‌های زیادی با عملکرد یکسان امّا پارامترهای کیفیتی متفاوت وجود دارد، لذا سرویس‌های اولیه برای فرایند ترکیب، باید به‌گونه‌ای انتخاب شوند که عمل ترکیب با بهترین شرایط ممکنه ضمن برآورده‌نمودن محدودیت‌های کاربر انجام گیرد. بنابراین، ترکیب آگاه از کیفیت سرویس، سرویس‌های نامزد را از هر مجموعه سرویس انتخاب و برای دستیابی به کارکرد مطلوب، طبق نظر کاربر ترکیب می‌کند.

استانداردهای کیفیت سرویس وب کنونی بر روی ویژگی‌هایی مانند زمان پاسخگویی، درجه اطمینان سرویس یا قابل‌اعتماد بودن[3]، امنیت[4] و غیره تمرکز نموده‌اند [۱]. براین‌اساس ترکیب آگاه از کیفیت سرویس، به‌عنوان یک مسئله چندهدفه بررسی می‌شود که برای تبدیل این مسئله به یک مسئله تک هدفه، باید از روش‌هایی ازجمله تابع مطلوبیت[5] استفاده‌نمود. درنتیجه به‌کارگیری الگوریتمی که بتواند وب‌سرویس‌ها را با کیفیت سرویس بهینه ترکیب نماید بسیار حائز اهمیت است. تاکنون الگوریتم‌های متعددی برای این مسئله ارائه گردیده است، که هرکدام سعی در رفع یک یا چند چالش داشته‌اند. امّا هریک از این روش‌ها دارای نقایصی همچون سرعت پائین، مصرف حافظه بالا، شکست سرویس‌ها در حین اجرا و نرسیدن به جواب بهینه هستند. از طرفی، چالشی که تمام این روش‌ها با آن مواجه هستند، تعداد زیاد معادلات و پیچیدگی نیازمندی‌های کاربران است. ازاین‌رو انتخاب و ترکیب وب‌سرویس‌ها و همچنین به‌دست آوردن ترکیبی که درخواست موردنظر کاربر را برآورده سازد، از اهمیت بالایی برخوردار است.

همان‌طور که در اغلب روش‌های موجود مشاهده می‌شود، با افزایش تعداد سرویس‌های نامزد، فضای جستجو و زمان ترکیب سرویس‌ها، افزایش می‌یابد که همین مسئله باعث نارضایتی کاربران می‌شود. لذا در این تحقیق، روشی ارائه می‌شود که سعی در کاهش فضای جستجو در ترکیب سرویس دارد. روش پیشنهادی، با هرس سرویس‌های نامزد، مرتب‌کردن سرویس‌های نامزد، هرس مجدد آن‌ها و همچنین استفاده از تابع مطلوبیت در کنار الگوریتم PCA[6]، زمان پاسخ و حافظه مصرفی را کاهش داده و کارایی ترکیب سرویس را بهبود می‌دهد.

این تحقیق در ادامه به شرح زیر سازماندهی شده است: بخش دوم، مروری بر تحقیقات گذشته دارد. در بخش سوم، مفاهیم اساسی موضوع بیان شده است. بخش چهارم، ساختار ترکیب سرویس با الگوریتم PCA را بیان میکند. در نهایت بخش پنجم، ارزیابی و در بخش ششم نتیجه‌گیری مطرح خواهد شد.

1- Service Level Agreement
2- Quality of Service (QoS)
3- Reliability
4- Security

5- Utility Function Method
6- Principal Component Analysis

۲- مرور سوابق

تحقیقـات زیادی در زمینه ترکیب سـرویس‌ها آگاه از کیفیت انجام شده است که در ادامه به برخی از آن‌ها اشاره می‌شود.

چن و همکاران، مدلی بر اسـاس مجموعه پارتو[۷] جهت یافتن سـرویس‌های مناسـب برای ترکیب ارائـه داده‌اند. در ایـن تحقیق تکنیک‌هـای انتخاب جزئی[۸] بـرای کاهش فضای جستجو پیشنهاد شـده‌اند و یک الگوریتم توزیعی بـه نام DPSA[۹] طراحی‌شـده اسـت. این طرح قادر اسـت بهبود قابل‌ملاحظه‌ای در کارایـی ایجاد کند و درعین‌حال مناسـب بودن ترکیب سـرویس را ضمانت می‌نماید [۲]. ژیان و همـکاران، بـرای اطمینان از این‌کـه پارامترهای کیفیت سـرویس، مطابق با نیاز کاربران باشـد، از روش BB4EPS[۱۰] اسـتفاده نموده‌انـد. در این روش یک مدل کلی بـرای ارزیابی پارامترهـای کیفیت سـرویس در ترکیب ارائه‌شـده اسـت [۳]. وانگ و همکاران، روشی را معرفی می‌کنند که در آن سـرویس انتخاب‌شده برای ترکیب آگاه به کیفیت سـرویس، هم‌زمان چندیـن محدودیت پارامتر کیفـی را برای نیازهـای کاربر، برآورده کنـد. این روش به‌طورکلی معادل مسـئله MMKP[۱۱] اسـت، که جزء دسته مسائل NP-hard[۱۲] است. همچنین در این تحقیق، نشان داده می‌شود که مدل پیشنهادی قادر به حل مسائل بزرگ بوده و راه‌حل بهینه را در حداقل زمان ممکن به‌دسـت می‌آورد [٤]. کـوردی و همکاران [٥]، یک مدل جبری[۱۳] بر اسـاس مفهوم بازگشـتی، به‌منظور بررسی شـرایط بن‌بست در ترکیب سرویس پیشـنهاد کرده‌اند. نتایج نشان داده است که استفاده از الگوریتم پیشـنهادی، زمان پاسخ و فضای مصرفی حافظه را برای ترکیب سـرویس کاهش می‌دهد. ولادیمر و همکاران [٦]، الگوریتم تقسـیم و غلبه [۱٤] را برای ترکیب سـرویس پیشـنهاد می‌دهند و الگوریتم تا زمانی تکرار می‌شـود که حداقل یک سرویس مناسب یافت شود. نتایج نشان می‌دهد که روش پیشنهادی، درخواست ترکیب سـرویس را به تعدادی از زیردرخواست‌ها به‌طور مؤثر و ساده تجزیه می‌کند. این فرآیند، ترکیب انعطاف‌پذیرتری را ایجـاد می‌کند. نتایج ارزیابی نشـان می‌دهد که این روش، حتی در موارد شکست الگوریتم، موفق به پیدا کردن راه‌حل درست برای تقریباً، تمام درخواست‌های ترکیب می‌باشد. پوتونـن و همـکاران [۷]، چگونگی ترکیب سـرویس را با الگوریتم ژنتیک بررسـی می‌کنند. در الگوریتم پیشنهادی، برخـی از ایده‌های جدید برای تولیـد کروموزوم، انتخاب و توابع مناسـب پیشـنهاد می‌شـود. نتایج نشان می‌دهد که با اسـتفاده از الگوریتم ژنتیک بسیار سریع‌تر از دیگر روش‌های جسـتجوی تصادفی، می‌توان بـه ترکیب‌بندی مناسب سرویس‌ها دسـت‌یافت، بنابراین زمان اجرا بهبود می‌یابـد. لیو و همکاران [۸]، یـک روش ترکیب داده‌محور برای برنامه‌های کاربردی وب با استفاده از معانی[۱۵] مبتنی بر برچسب[۱٦] ارائه دادند که یک روش پویا است و ترکیب، به‌محض جسـتجوی برچسـب صورت می‌گیـرد. در این روش، برای رسـیدن به ترکیب سـرویس‌های مقاوم، یک الگوریتم برنامه‌ریزی اکتشافی مبتنی بر گراف با پیچیدگی زمانی چندجمله‌ای پیشـنهاد شده اسـت و مسئله ترکیب به‌عنوان تولید چند مدل پیوند برچسـب[۱۷] که خروجی‌های موردنظر جهت رسـیدن به هدف را تولید می‌کنند، در نظر گرفته شده است. در این روش تمرکز اصلی روی قابلیت‌ها و وظیفه‌مندی سرویس‌ها بوده و پارامترهای کیفی در نظر گرفته نشـده اسـت. زینگ ژیان و همکاران [۳]، یک روش رتبه‌بندی فازی مبتنی بر فاصله، برای ترکیب سرویس آگاه از کیفیت سرویس ارائه کردند. نتایج ارزیابی نشان می‌دهد روش پیشـنهادی، عملکرد بهتری در ترکیب سـرویس‌ها بخصوص زمانی که اندازه مسـئله بزرگ‌تر می‌شود دارد. یو و همکاران [۹]، روشـی ارائه کردند که الگوریتم کلونی

7- Pareto Set
8- Partial Selection
9- Distributed Partial Selection Algorithm
10- Branch & Bound For Execution Plan Service
11- Multidimensional Kknapsack
12- Non-Deterministic Polynomial
13- Service Net Algebra
14- Divide and Conquer
15- Semantics
16- tag
17- Tag Link

مورچگان را برای فرایند ترکیب سـرویس در محاسـبات ابری بهینه می‌سازد. در این تحقیق یک الگوریتم حریصانه با نام Greedy-wsc و یک الگوریتم بر اسـاس بهینه‌سازی کلونی مورچگان که ACO-WSC نامیده می‌شود، ارائه‌شده اسـت. هدف این روش انتخاب ترکیبـات عملی و مؤثر با استفاده از حداقل تعداد ابرها است.

لی و همکاران [۱۰]، با ایجـاد یک پایگاه داده رابطه‌ای جدیـد، ترکیـب خودکار سـرویس‌ها را انجـام دادند. در این روش با تولید و ذخیره‌سـازی تمـام ترکیبات ممکن سـرویس‌ها، در یک پایگاه داده رابطه‌ای به هنگامی‌که یک درخواسـت کاربر ارسال می‌شـود، با پرس‌وجوی پایگاه داده و جسـتجو در آن، بهترین راه‌حـل ترکیبی مبتنی بر کیفیت سـرویس را ارائه می‌دهد. ایـن روش، k راه‌حل با بالاتریـن کیفیت معتبـر را برای برآورده شـدن نیازهای کاربـر، پیدا می‌کنـد. ژو و همـکاران [۱۱]، پیش‌بینی‌هایی برای QoS سرویس‌های نامزد برای سازگاری سرویس‌ها در زمان اجرای ترکیب سـرویس انجام داده‌اند. به منظور رسـیدن به این هدف از روش تقسیم‌بندی ماتریس انطباق استفاده کرده اند. از مزیت های این تحقیق، یادگیری برخط و سـازگاری و انتقال تقسـیم‌بندی ماتریس سنتی است. نتایج ارزیابی و مطالعه روی مجموعه داده‌های جهانی وب سرویس‌ها نشـان می‌دهد که این ماتریس انطباق از لحاظ دقت و کارایی و استحکام معتبر است. جاتوس و همکاران [۱۲]، یک الگوریتم مبتنی بر نگاشت/کاهش[18] برای رسیدن به ترکیب سرویس‌های بزرگ آگاه از کیفیت ارائه دادند. این مقاله بهترین عملکرد را از نظر امکان‌سنجی، مقیاس‌پذیری و بهینه‌سـازی بـا ویژگی‌هـای مختلـف QOS برای حل ترکیب سرویس‌های بزرگ دارد. این رویکرد روی مسئله یادگیری ماشـین و الگوریتم‌های فراشناختی[19] کار نکرده است. به دلیل این‌که مسـئله زمانبندی در سیستم‌های ابر پیچیده اسـت بنابراین باید روی این مسـئله و بهره وری انرژی بیشـتر کار کند. مدل قیمت گـذاری باید تکامل یابد و هم چنین روی قیمت گذاری چند بعدی تمرکزی نداشـته اسـت. لی و همکاران [۱۳]، مسـائل هماهنگـی چند عامله در بازی‌های همکارانه بـا اولویت‌های متفاوت کاربران را بررسـی نمودند. در این تحقیق هر عامل نشـان‌دهنده یک ترجیح کاربر اسـت و در نهایت سیاسـتی را که برای هر کاربر مناسـب است انتخاب می‌کند. در عمل در محیط‌های سـرویس به دلیل QoSهـای ناپایدار ممکن اسـت نتواند تضمینی برای به‌دست آوردن سود بدهد. بنابراین با توجه به ترجیحات کاربـر و هدف‌های قطعی به هر عامل اجازه داده شده که در نوبت‌های تصادفی بتوانند جفت خودشان را تغییر دهند. بنابراین عامل توانسته است که با استفاده از ترجیحات متفاوت کاربران راهبرد را یاد بگیرد. این تحقیق نسب به روش‌های یادگیری چندعامله بهتر کارکرده است ولـی باید الگوریتم یادگیری خود را از لحاظ زمان پاسـخ و دقـت بهبود بدهد. لیو و همـکاران [۱۴]، راهکار انتخاب سرویس پویا بر اساس Skyline را برای کاهش افزونگی[20] پیشنهاد دادند. در روش ارائه شده فرایند انتخاب سرویس به دو مرحله تقسـیم شده اسـت: مرحله انتخاب سرویس و مرحله اجرای سـرویس‌های انتخابی. مرحله انتخاب، با اسـتفاده از روش برون‌خـط، محاسـبه skyline را انجام می‌دهد و مسـئول به روز رسانی Skyline سرویس است. بنابراین رونـد برون‌خط هرگز بر عملکـرد زمان مرحلهٔ انتخاب سـرویس‌ها تاثیر نمی‌گذارد. مرحله اجرا، مسئول انتخاب ترکیب مطلوبی از سرویس‌ها است که با محدودیت QoS کاربران مطابقت دارد. نتایج نشـان می‌دهد که روش ارائه شده با دقت بهتر، مناسب‌ترین سرویس‌ها را انتخاب می‌کند.

توسـط Mezni و همکاران [۱۵]، روشی برای ترکیب سـرویس در محیط چند ابری (MCSC)[21] بر اساس تحلیل مفهومـی رسـمی (FCA)[22] بـرای نشـان دادن و ترکیب سـرویس‌ها در محیط چند ابری پیشـنهاد شده است. در ایـن روش ابتدا محیط چند ابری به‌عنـوان مجموعه‌ای از

18- map reduces
19- Meta-heuristic
20- Redundancy
21- Multi-Cloud Service Composition
22- Formal Concept Analysis

زمینه‌های رسمی مدل‌سازی شده، سپس ابرهای داوطلب از مفاهیم رسمی اســتخراج و ترکیب می‌شوند. در نهایت ترکیب ابری مطلوب انتخاب شــده و MCSC به یک مسئلهٔ ترکیب ســرویس کلاسیک تبدیل می‌شــود. آزمایش‌های انجام شــده اثربخشــی و توانایی روش مبتنی بر FCA را برای گروه‌بندی و یافتــن ترکیب‌های ابری با حداقل تعداد ابرها و هزینهٔ ارتباطات کم نشان می‌دهد. ژائو و همکاران [۱۶]، بــرای ترکیــب وب‌ســرویس‌های مبتنی بــر کیفیت سرویس که مغایرت‌ها و وابستگی‌های بین سرویس‌ها را نیز درنظر می‌گیرد الگوریتم ژنتیک بی‌نظمی به‌نام (CGA)[۲۳] پیشنهاد داده‌اند. الگوریتم بهینه‌سازی بی‌نظم (COA)[۲۴] برای غلبــه بر ضعف الگوریتم ژنتیک مانند گیرکردن در بهینگی محلی و ســرعت همگرایی آهسته اســتفاده شده است. در روش ارائه شده ترکیب خلاقانه‌ای از الگوریتم‌های ژنتیک و COA را برای به‌دست آوردن الگوریتم ژنتیک بی‌نظم CGA که تطبیق‌پذیری و مقیاس‌پذیری بهتری دارد ارائه می‌دهد. در روش پیشنهادی از تئوری بی‌نظمی برای ایجاد جمعیت اولیه الگوریتم ژنتیک اســتفاده کرده است و راهبرد ترمیم به منظور مقابله با جمعیتی که محدودیت‌های درخواستی کاربر را نقض می‌کنند در نظر گرفته شــده است. هیولالام و همــکاران [۱۷]، در تحقیق خود روش‌هــا و راهبردهای موجود جهت ترکیب ســرویس‌های ابــری آگاه از کیفیت ســرویس را مرور نمودنــد. آن‌ها روش‌هــای موجود را ازنظر نوع تکنیک استفاده‌شده و بر اساس عوامل متفاوتی بررســی نمودند. سپس با طبقه‌بندی تحقیق‌ها به دو گروه اصلی متمرکز و توزیع‌شده و بر اساس معیارهایی همچون محیط مســئله، پارامترهای QoS موردبررســی، اهداف از پیش تعریف‌شده و محیط‌های توسعه، نتایج و آمار مناسب به‌دست آوردند که می‌تواند به نتایج کارهای آتی کمک کند. بررسی روش‌های پیشنهادی نشــان داد که ۳۴ درصد از محققان از روش‌های غیرابتکاری و مابقی از الگوریتم‌های ابتکاری اســتفاده نموده‌اند که درصد بیشتری از آن‌ها از الگوریتم ژنتیک استفاده نموده‌اند.

گاوالا و همکاران [۱۸]، در تحقیق خود به منظور ترکیب ســرویس‌های ابری از ترکیب الگوریتــم عقاب و الگوریتم بهینه‌سازی نهنگ استفاده نمودند. الگوریتم پیشنهادی آن‌ها توازن بیشــتری را بین دو مســئله شناسایی و استفاده از سرویس‌های مورد نیاز فراهم می‌کند. نتایج ارزیابی نشان می‌دهد که روش پیشنهادی آن‌ها از الگوریتم‌های ژنتیک و نهنگ، کلونی زنبورها، الگوریتم ژنتیک ترکیبی و الگوریتم جستجوی حریصانه بهتر عمل می‌کند. یعقوبی و همکاران [۱۹]، در تحقیق خود یک الگوریتم بهینه‌ســازی چندوجهی Multi-Verse Optimization (MVO) algorithm بــرای ترکیب ســرویس‌های وب در ابر ارائــه نمودند که مبتنی بر قرارداد ســطح ســرویس بوده و منجر به بهبود QoS می‌شــوند. آن‌ها در الگوریتم IMVO خود مشخص نمودند که میــزان اختلاف فاصلــه تعیین کننــده اختلاف فضای جستجوی پیرامون بهترین راه‌حل فعلی است. نتایج ارزیابی موثربودن روش پیشنهادی را نسبت به الگوریتم‌های دیگر از دیدگاه برآورده نمودن قرارداد ســطح سرویس نشان می‌دهد. کانوچه و همکاران [۲۰]، در روش پیشنهادی خود یک الگوریتم انعطاف‌پذیرترکیب ســرویس آگاه از QoS به نام FQSC ارائه نمودند تا منجر به افزایش ســرعت ترکیب سرویس و افزایش ترکیب‌های سرویس ممکن شود. آن‌ها در روش پیشــنهادی خــود از محدودیت‌هــای روش‌های ترکیب ســرویس مبتنی بر اصل غلبه پارتو و تجزیه QoS اســتفاده نمودند. آن‌ها نشــان دادند که تعداد ترکیبات، با تعداد ســرویس‌ها و ویژگی‌های QoS در روش غلبه پارتو به‌صورت نمایــی افزایش می‌یابــد در حالی‌که روش‌های مبتنــی بر تجزیه QoS منجر به کاهش ترکیبات می‌شــوند بخصــوص زمانی که محدودیت‌هــای عمومی QoS کاربر خیلی محدودکننده باشد.

۳-تعاریف اولیه

فرض کنید با دسته‌بندی وظایف یا انواع درخواست‌های

23- Chaos Genetic Algorithm
24- Service-Oriented Computing

معتبـری که کاربـر می‌توانـد داشـته باشـد، n وظیفه به‌صـورت $T = \{t_1, t_2, \dots, t_n\}$ ایجـاد شـود کـه هر وظیفـه برای پاسـخ به کاربـر حاوی تعدادی سـرویس $t_i = \{s_{i1}, s_{i2}, \dots, s_{im_l}\}$ (m_l تعـداد سـرویس‌ها در t_i اسـت) اسـت که ازنظر وظیفه‌مندی یکسـان امّا دارای پارامترهای کیفیتی متفاوتی هستند. به‌عنوان مثال جدول ۱ یک نمونه از دسته‌بندی سرویس‌ها را نشان می‌دهد.

حال فرض کنید کاربر، متناسب با نیاز خود درخواست t_i را مطـرح می‌کند که m_l مجموعه سـرویس برای انجام آن نیاز است و به‌صورت $t_i = \{s_{i1}, s_{i2}, \dots, s_{im_l}\}$ نشان داده می‌شود. همچنین، برای هر سرویس s_{iml}، n_i سرویس نامزد وجود دارد و همچنین هر سرویس s_{iml} دارای تعدادی پارامتر کیفی است که با $Q^i_{m_l} = \{q_1, q_2, \dots, q_k\}$ نشان داده می‌شود. بنابراین ترکیب سـرویس‌ها برای پاسخ به درخواسـت کاربر به‌صورت $S_P = \{s_{i1}, s_{i2}, \dots, s_{im_l}\}$ (یک گردش کار پیوسـته) نشان داده می‌شـود. از طرفی تمامی ترکیبات، یک فضای ترکیب $S = t_1 \times t_2 \dots \times t_n$ را تشکیل می‌دهند.

۴- ترکیب سرویس برمبنای الگوریتم PCA

در این بخش، مراحل اصلی به‌دست آوردن ترکیب‌های مناسـب بر اساس الگوریتم PCA بیان می‌شود. این روش، چهـار مرحلـه اصلی دارد که سـه مرحلـه اول آن روی مجموعه سـرویس‌های نامزد انجام می‌شود. خروجی این سه مرحله سـرویس‌های نامزد کاهش‌یافته است که برای ترکیب، مناسـب هسـتند. مرحله آخر نیز، عملیات ترکیب سـرویس‌های نامزد، انجـام می‌گیرد. مراحـل الگوریتم عبارتند از:

- اعتبار سنجی و هرس سرویس‌های نامزد
- اعمال الگوریتم PCA
- هرس سرویس‌ها
- ترکیب سرویس‌ها
- جزئیات این روش در ادامه بیان می‌شود.

جدول ۱: دسته‌بندی سرویس‌ها در قالب وظایف

t_1	t_2	t_3	t_4	t_5	t_6	t_7
s_{11}	s_{21}	s_{31}	s_{41}	s_{51}	s_{61}	s_{71}
s_{12}	s_{22}	s_{32}	s_{42}	s_{52}	s_{62}	s_{72}
s_{13}	s_{23}	s_{33}	s_{43}	s_{53}	s_{63}	s_{73}
s_{14}	s_{24}	s_{34}	s_{44}	s_{54}		s_{74}
s_{15}	s_{25}			s_{55}		s_{75}
	s_{26}					s_{76}
	s_{27}					s_{77}

۴-۱- اعتبار سنجی و هرس سرویس‌های نامزد

در این تحقیق هر سرویس نامزد دارای پارامترهای کیفی وزن‌دار زمان پاسخ، هزینه و قابلیت اطمینان است. وزن این پارامترها در درخواسـت کاربر به همراه یک کران پارامتر کیفی طبق [۵] به‌عنوان (ماکزیمم و مینیمم مقادیر پارامتر کیفی) مطابق نیاز کاربر تعریف شـده است. به‌عنوان مثال کاربر درخواسـت دارد که زمان اجرای درخواست او در بازه [time1,time2] باشد. در اعتبارسنجی سرویس‌های نامزد، هر سـرویس نامزد s_{iml}، کـه پارامترهای کیفی آن، محدودیت SLA و bound را نقض کنند، جواب‌های مناسب بـرای ترکیب، ایجـاد نمی‌کنند و به‌عنوان سـرویس‌های ترکیب‌نشـدنی در نظر گرفته شده و با مجموعه D نمایش داده می‌شـوند. بعد از هرس کلیه سرویس‌ها، طبق رابطه (۱)، مجموعه D، حذف می‌شود تا فضای مصرفی و زمان ترکیب کاهش یابد.

$$t_i' = t_i - D \qquad (۱)$$

t_i' مجموعه سـرویس‌های هرس‌شـده هسـتند که محدودیت‌هـای تعریف شـده توسـط کاربـر را رعایت نموده‌اند. از آنجایی که هر سـرویس s_{im_l} دارای پارامتر q_k کیفی که به‌صورت بردار $Q^i_{m_l}$ نشـان داده می شود، بردار تابع محدودیت برای آن‌ها به‌صورت رابطه ۲ نشان داده می‌شود.

$$F_{Q^i_{m_l}} = \left(f_1^{Q^1_{m_l}}, f_2^{Q^2_{m_l}}, \dots f_k^{Q^i_{m_l}}\right) \qquad (۲)$$

کـه در آن، $f_k^{Q^i_{m_l}}$ ، تابع تجمعی k محدودیت را نشـان می‌دهـد. بنابراین تابع محدودیـت S_P به‌صورت رابطه ۳ تعریف می شود:

$$F_{Q^i_{m_l}}(S_p) = \left(f_1^{Q^1_{m_l}}, f_2^{Q^2_{m_l}}, \dots f_k^{Q^i_{m_l}}\right) \quad (۳)$$

حال برای بررسی این‌که آیا kامین محدودیت برآورده می شـود یا نه به تابع منطقی $\mathrm{op_r(R_r, bound_r)}$ نیاز اسـت که $\mathrm{bound_k}$ ماکزیمـم و مینیمم مقدار پارامتر k را نشـان می‌دهد که توسـط کاربر برای ویژگی تعریف می‌شـود. تابـع op_r طبق رابطه (٤) مقـدار می‌گیرد بدین صورت که اگر محدودیت برآورده شود مقدار 1 و در غیراین‌صورت مقـدار 0 می‌گیـرد. در نهایت یک ترکیـب S_P می‌تواند با مقایسـه هریک از مقادیر $f_k^{Q^i_{m_l}}$ آن با کـران متناظر خود یعنی bound_k اعتبارسنجی شود.

$$op_r = \begin{cases} 1 \; if \; R_i \; satisfied \\ 0 \; else \end{cases} \quad (٤)$$

٤-۲ عملکرد الگوریتم PCA در ترکیب سرویس

یکی از روش‌های کاهـش ابعاد داده، روش‌های مبتنی بر استخراج ویژگی است که یک فضای چندبعدی را به یک فضای با ابعاد کمتر نگاشت می‌کنند. الگوریتم PCA بهترین روش برای کاهش ابعاد داده به‌صورت خطی اسـت که در آن با حذف ضرایب کم‌اهمیت، اطلاعات ازدست‌رفته نسبت به روش‌های دیگر کمتر اسـت. در این روش، ویژگی‌ها با یکدیگر ادغام‌شـده و از مقادیر پارامترها برای ترکیب‌های مناسب اسـتفاده می‌شـود. ازآنجایی‌که در این روش، از ماتریس کواریانس اسـتفاده می‌شـود و این ماتریس در ابعاد بیشـتر از ۳×۳ کارایی چندانـی ندارد، در این تحقیق سـه بعد داده شامل ویژگی زمان، هزینه و قابلیت اطمینان برای سرویس‌ها در نظر گرفته‌شده است. در ادامه، کاربرد الگوریتم PCA در این تحقیق توضیح داده می‌شود.

مرحله ۱- انتخاب داده

داده‌ها، به‌صورت مقادیر پارامترهای کیفی زمان، هزینه و قابلیت اطمینان برای هر سرویس‌نامزد t_i'، درنظر گرفته شـده‌اند. لذا اگر تعداد سرویس‌های نامزد t_i' را N در نظر بگیریـم، مقادیر پارامترها می‌توانـد، در یک ماتریس N×3 ذخیره شود که به‌صورت نشان داده می‌شود.

مرحله ۲- نرمال‌سازی داده‌ها و محاسبه ماتریس کواریانس

در این مرحله، طبق رابطه (٥)، مقادیر هر بعد برای هر سـرویس، نرمال شـده و ماتریس نرمال شدهٔ A برای هر سرویس ایجاد شود.

$$\bar{x} = \frac{\sum_{i=1}^{n} q_i}{n} \quad (٥)$$
$$q_i' = |\bar{x} - q_i| \; \mathrm{i=1, 2, \dots, k}$$
$$A = [q_1', q_2', \dots, q_k']$$

از آنجایی که با اسـتفاده از مفهوم کواریانس می‌توان ارتبـاط بین ابعـاد مختلف داده‌هـا را پیدا کـرد، در این مرحلـه، برای هر سـرویس نامزد ، ماتریـس کواریانس سـه پارامتـر کیفـی زمـان، هزینـه و قابلیـت اطمینان B=COV(Time, Price, Reliability) محاسبه می‌شود و در نهایت یک ماتریس کواریانس ۳×۳ ایجاد می شود. در انتها نیز مقدار ویژه و بردار ویژه ماتریس کواریانس نیز حاصل می‌شود.

مرحله ۳- انتخاب مؤلفه‌ها و ساختن Feature Vector

بـردار ویژهٔ مرحلهٔ قبل، بر اسـاس مقادیر ویژهٔ آن‌ها از بزرگ به کوچک مرتب می‌شود. بدین ترتیب مؤلفه‌های داده‌هـا از پراهمیت به کم‌اهمیت مرتب می‌شـوند. در این مرحله نیز بـرای کاهش ابعاد داده‌هـا می‌توان مؤلفه‌های کم‌اهمیت را حذف کرد. براین اسـاس سـرویس‌هایی که دارای مقادیـر کیفی پائینی هسـتند حذف می‌شـوند. در این تحقیق، برای رسـیدن به ترکیب مناسـب سرویس و هرس سـرویس‌های نامزد تنهـا بزرگ‌ترین بردار ویژگی در نظر گرفته‌شـده اسـت و ماتریس E نامگذاری شـده است.

مرحله ٤- به‌دست آوردن داده‌های جدید

در آخرین مرحله از PCA، ترانهادهٔ ماتریس بردار ویژه E در ترانهادهٔ داده‌های نرمال‌سـازی شده طبق رابطه (٦)، ضرب می‌شـود. به‌این‌ترتیب در هر سـطر داده‌های جدید استخراج می‌شوند.

$$T_{1*n} = E^T * A^T \quad (٦)$$

نتیجه، یک ماتریس N×1 است که درایه‌های آن، مقادیر

پارامترهای کیفی سرویس‌های نامزد ادغام شده است.

٤-٣ هرس مجدد سرویس‌ها

داده‌های جدید در ماتریس T، به‌صورت نزولی مرتب می‌شوند تا داده‌های مفید کمتری از دست برود. در صورت عدم مرتب‌سازی [٥]، سرویس‌ها به‌صورت جفت‌جفت مقایسه می‌شوند و در این‌صورت بعضی از سرویس‌های مفید نادیده گرفته‌شده و حذف می‌شوند. سپس، یک دوم داده‌های ماتریس مرتب شدهٔ T، از انتهای سطر حذف می‌شود و به‌این‌ترتیب هرس مجدد روی سرویس‌های نامزد انجام می‌شود. لازم به ذکر است که حذف مقدار یک دوم داده ها با مجموعه سرویس‌های متفاوت و با آزمایش‌های متعدد به‌دست آمده است. در قدم بعدی، مقادیر موجود در ماتریس T به سرویس‌های نامزد مربوطه نگاشت می‌شود تا تعداد سرویس‌های نامزد مجددا کاهش یافته و مجموعهٔ جدیدی از سرویس‌های مناسب برای ترکیب ایجاد شود. این مجموعهٔ جدید با $\tilde{t}_i$ نشان داده می‌شوند.

٥- ترکیب سرویس‌ها

از آنجا که هدف ترکیب سرویس آگاه از کیفیت، پیدا کردن بهترین ترکیب است که به درخواست کاربر نزدیک باشد، بنابراین برای رسیدن به یک سرویس ترکیبی مناسب بایستی تابع مطلوبیت تجمیع مقادیر پارامتر کیفی محاسبه شود. در این تحقیق بعد از هرس و کاهش تعداد سرویس‌ها، عمل ترکیب سرویس مطابق با درخواست کاربر انجام می شود.

٥-١- تابع تجمع مقادیر پارامتر کیفی

همان‌طور که قبلا گفته شد، هر سرویس s_{im_l} دارای تعدادی پارامتر کیفی است که با $Q^i_{m_l} = \{q_1, q_2, \dots, q_k\}$ نشان داده می‌شود که در مرحله قبل براساس این پارامترها کاهش یافتند. در ادامه برای ترکیب سرویس‌ها به‌گونه‌ای که مقدار پارامتر کیفی سرویس‌های ترکیبی متناسب با درخواست کاربر باشد باید تابع مطلوبیت و تابع تجمعی را محاسبه نمود. بنابراین برای هر سرویس ترکیبی، هر ویژگی از بردار QoS باید با استفاده از تابع تجمعی برای یک جریان کاری مناسب جمع شود. به‌عبارتی تابع تجمع، عملیاتی است که نحوهٔ به دست آوردن مقادیر پارامتر کیفی ترکیب سرویس را توصیف می‌کند. به‌عنوان مثال برای ترکیبی مانند $S_P = \{s_{i1}, s_{i2}, \dots, s_{im_l}\}$ هرکدام از سرویس‌های نامزد s_{im_l}، پارامترهای کیفی زمان، هزینه و قابلیت اطمینان خود را دارند. برای به دست آوردن مقدار هرکدام از پارامترهای کیفی در ترکیب، از تابع تجمع استفاده می‌شود. تابع تجمع برحسب ساختارهای مختلفی که ترکیب می‌تواند داشته باشد به‌صورت‌های مختلفی محاسبه می‌شود. جدول ٢ لیستی از معیارهای کیفی مورد استفاده در این تحقیق را به همراه نحوه محاسبه ارزش مجموع آن معیار کیفی در ساختارهای مختلف نشان می‌دهد. همان‌طور که جدول ٢ نشان می‌دهد، پارامترهای کیفی، به شش دسته بر اساس خواص توابع تجمع خود رده‌بندی می‌شوند که شامل عملیات جمع، ضرب، مینیمم، ماکزیمم، اجتماع و اشتراک هستند.

برای محاسبه میزان تابع تجمع یک سرویس مرکب، مقادیر نرمال شدهٔ هر یک از پارامترهای کیفی سرویس را با توجه به ساختارهای متفاوت جدول ٢، به طور جداگانه جمع نموده و بدین صورت ارزش مجموع تمامی پارامترهای کیفی در یک سرویس مرکب محاسبه می‌شود.

٥-٢- محاسبه تابع مطلوبیت

همان‌طور که در بخش قبل بیان شد، ترکیب سرویس آگاه به پارامترهای کیفی بر روی ویژگی‌هایی مانند زمان پاسخگویی، هزینه، قابلیت اطمینان، دسترس‌پذیری و... تمرکز نموده است. به‌عنوان مثال، ممکن است یک کاربر ترجیح دهد که درخواستش با زمان پاسخ کمتر و درعین‌حال پرداخت کمترین هزینه انجام شود که دستیابی به آن دشوار است. به‌عنوان مثال، سرویس‌های با حداقل تأخیر معمولاً هزینه بالاتری در پی‌دارند. برای برطرف کردن این چالش، می‌توان از تکنیک‌های SAW

جدول ۲: توابع محاسبه ارزش مجموع هر یک از معیارهای کیفی با ساختارهای مختلف

	ترتیبی	موازی	شرطی	حلقه
زمان پاسخگویی	$\sum_{i=1}^{n} T(S_i)$	$Max(T(S_1)\dots T(S_n))$	$Max(T(S_1)\dots T(S_n))$	$k * T(S)$
هزینه	$\sum_{i=1}^{n} C(S_i)$	$Max(C(S_1)\dots C(S_n))$	$Max(C(S_1)\dots C(S_n))$	$k * C(S)$
قابلیت اطمینان	$\prod_{i=1}^{n} Re(S_i)$	$\prod_{i=1}^{n} Re(S_i)$	$Min(Re(S_1)\dots Re(S_n))$	$(Re(S))^k$

بهره برد و از روش تابع مطلوبیت استفاده کرد که اندازه‌گیری یکپارچه از اهداف متعدد را صرف‌نظر از واحدها و محدوده‌های آن‌ها فراهم می‌نماید [۲۱]. تکنیک‌های SAW مقادیر پارامترهای کیفی را با مقادیر احتمالی متوسط، حداقل و حداکثر مقایسه می‌کند. سپس مطابق با نیاز کاربر به پارامترها وزن داده و مسئله را به یک مسئله تک‌هدفه، مقیاس‌بندی می‌نماید. درنهایت، یک مقدار مطلوبیت برای ارائه عملکرد یک ترکیب به‌کاربرده می‌شود و ترکیب با مقدار مطلوبیت بالاتر، انتخاب می‌شود.

فرض کنید ω_i وزن ویژگی iام باشد، که براساس درخواست کاربر و پارامترهای کیفی مورد نظر او در نظر گرفته شده است ضمن این‌که شرط $\omega_i > 0$ و $\sum_i \omega_i = 1$ باید برآورده شود. از آنجائی که مقادیر پارامتر کیفی نرمال شده‌اند، هدف فرایند ترکیب سرویس، یافتن ترکیبی است که تابع مطلوبیت را مطابق رابطه (۷) بیشینه نماید.

$$maximize \sum_{i=1}^{n} \frac{w_i f_k^{Q^i_{m_l}}(x)_{positive}}{w_i f_k^{Q^i_{m_l}}(x)_{negative}} \qquad (۷)$$

که $f_k^{Q^i_{m_l}}(x)_{positive}$ تابع تجمعی پارامترهای کیفی هستند که مقادیر بیشتر آن‌ها توسط کاربر درخواست می‌شود و $f_k^{Q^i_{m_l}}(x)_{negative}$ پارامترهای کیفی هستند که مقادیر کمتر آن‌ها توسط کاربر درخواست می‌شود. n تعداد سرویس‌های ترکیبی و ω_i وزن‌های مربوط به پارامترهای کیفی زمان، هزینه و قابلیت اطمینان است که کاربر در نظر می‌گیرد. در واقع برای هر ترکیب سرویس تابع مطلوبیت طبق رابطه ۷ محاسبه می شود و در نهایت ترکیب سرویسی انتخاب می شود که بالاترین مقدار تابع مطلوبیت را داشته باشد.

۵-۳- ترکیب سرویس‌های نامزد

در این مرحله عمل ترکیب بر روی سرویس‌های هرس شده یا سرویس‌هایی که بهترین QoS را دارند، انجام می‌شود. مجموعه‌های شامل m سرویس $\tilde{t}_1, \tilde{t}_2, \dots, \tilde{t}_m$ به $|m/2|$ جفت تقسیم‌بندی می‌شود. هر جفت دارای دو مجموعه بیان‌شده به‌صورت $\{\tilde{t}_i, \tilde{t}_{i+1}\}$ یا یک مجموعه $\{\tilde{t}_i\}$ است. به‌عنوان مثال زمانی که m=5 باشد، مجموعه سرویس‌ها می‌توانند به سه جفت تقسیم شوند که عبارتند از $\{\tilde{t}_1, \tilde{t}_2\}$، $\{\tilde{t}_3, \tilde{t}_4\}$، $\{\tilde{t}_5\}$. این مجموعه‌ها در یک جفت، درون یک مجموعه بزرگ‌تر ادغام می‌شوند. به‌عنوان مثال، برای جفت $\{\tilde{t}_i, \tilde{t}_{i+1}\}$، می‌توان $\tilde{t}_{i,i+1} = \tilde{t}_i \times \tilde{t}_{i+1}$ را به‌صورت یک مجموعه ترکیب جدید به دست آورد. در این‌صورت جفت‌هایی نگهداری می‌شوند که محدودیت کاربر را برآورده کنند و بقیه حذف می‌شوند چون ترکیب مناسبی را ارائه نمی‌دهند. برای این کار از تابع مطلوبیت طبق رابطه ۱۰ برای رسیدن به ترکیب سرویس آگاه به پارامتر کیفی متناسب با نیاز کاربر استفاده شده و به‌این‌ترتیب، ترکیب تک‌هدفه حاصل می‌شود. در واقع بعداز اعمال فرآیندهای اعتبارسنجی، هرس، الگوریتم PCA و اجرای تابع مطلوبیت روی تمامی جفت سرویس‌ها، $\tilde{t}_{i,i+1}$ حاصل می‌شود و یک دور خاتمه می‌یابد. سپس دور بعدی این فرایندها تکرار می‌شوند تا وقتی‌که به مجموعه بزرگی دست‌یابیم که دربردارنده تمامی m دسته سرویس باشد. اجرای این فرآیندها بر روی جفت‌های مختلف $\{\tilde{t}_i, \tilde{t}_{i+1}\}$ می‌تواند به‌طورهمزمان بر روی چندین سرویس‌دهنده یا

پردازشـــگر انجام شـــود. در این روش تابع مطلوبیت هر بار بخشـــی از داده‌ها و الگوریتم PCA نیز نیمی از داده‌ها را هـــرس می‌کنـــد و بنابراین فضای مصرفـــی حافظه را کاهش داده و کارایی و راندمان را بهبود خواهد بخشـــید. مراحل اصلی روش پیشنهادی در شکل(۱) نشان داده‌شده است.

۶- ارزیابی روش پیشنهادی

این بخش به بررســـی و ســـنجش نتایج پیاده‌ســـازی ترکیب ســـرویس بـــر مبنـــای الگوریتـــم PCA می‌پردازد. نمونه‌ها برای ارزیابی روش پیشنهادی به‌صورت تصادفی ایجاد شده‌اند. هر سرویس دارای مشخصاتی همچون نام ســـرویس، پارامترهای ورودی و خروجی و پارامترهای کیفیت سرویس است.

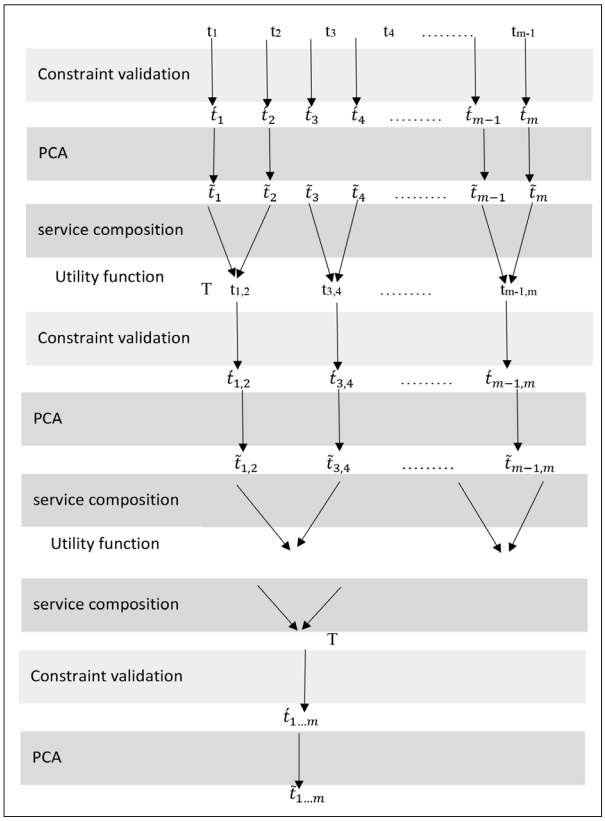
در این بخش روش ترکیب ســـرویس بـــر مبنای PCA و روش ترکیب انتخاب جزئی[۲۵] ارائه‌شـــده توســـط چن و همکارانش [۲]، از دو جنبه زمان و حافظه مصرفی ارزیابی شده‌اند. ابتدا دو روش از نظر زمان و فضای حافظه صرف شـــده با تعداد متغیر Task و تعداد ثابت سرویس نامزد و دوم ازنظـــر زمان و فضای حافظه صرف شـــده در تعداد ثابت Task و تعداد متغیر ســـرویس نامزد مقایسه شده‌اند. در جدول (۲)، کارایی ترکیب سرویس‌های آگاه به پارامتر کیفی مبتنی بر الگوریتم ترکیب جزئی با روش پیشنهادی، مقایسه شده است.

در شکل (۲)، زمان پاسخ و در شکل (۳)، میزان حافظه مصرفی دو روش مقایسه شده که محور افقی تعداد متغیر Task با تعداد سرویس نامزد ۵ را نشان می‌دهد.

بـــرای این‌که بهبود زمان پاســـخ، در این قســـمت بهتر نشان داده شود ؛ شکل (٤)، با تعداد Task کمتر نمایش داده شده است.

همچنین برای نمایش بهتر بهبـــود حافظه مصرفی در الگوریتـــم PCA با تعداد Task متغیر، شـــکل ۵، نمایش داده می‌شود.

25-Partial Selection Methodology



شکل ۱: مراحل اصلی ترکیب سرویس آگاه به معیار کیفی مبتنی بر PCA

از طرفـــی در شـــکل (٦)، تعـــداد Task ثابـــت در نظـــر گرفته‌شـــده و ســـرویس‌های نامزد، از ۳ تا ۳۰ تغییر کرده است.

در شـــکل (۷) فضای حافظه مصرفی در الگوریتم PCA و ترکیب جزئی مقایسه شده است؛ در این قسمت نیز تعداد Task ثابت و تعداد ســـرویس‌ها متغیر در نظر گرفته شده است.

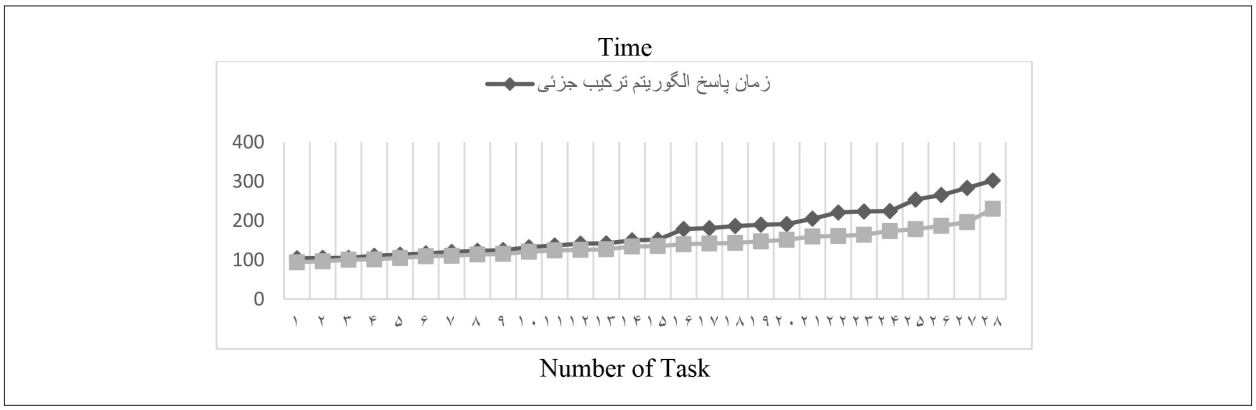
در ادامـــه، برای نمایـــش بهتر بهبود زمان پاســـخ در الگوریتم PCA، شـــکل (۸) با تعداد ســـرویس متغیر نمایش داده می‌شود.

همچنین در شکل (۹)، بهبود فضای حافظه مصرفی در الگوریتم PCA نشان داده‌شده است.

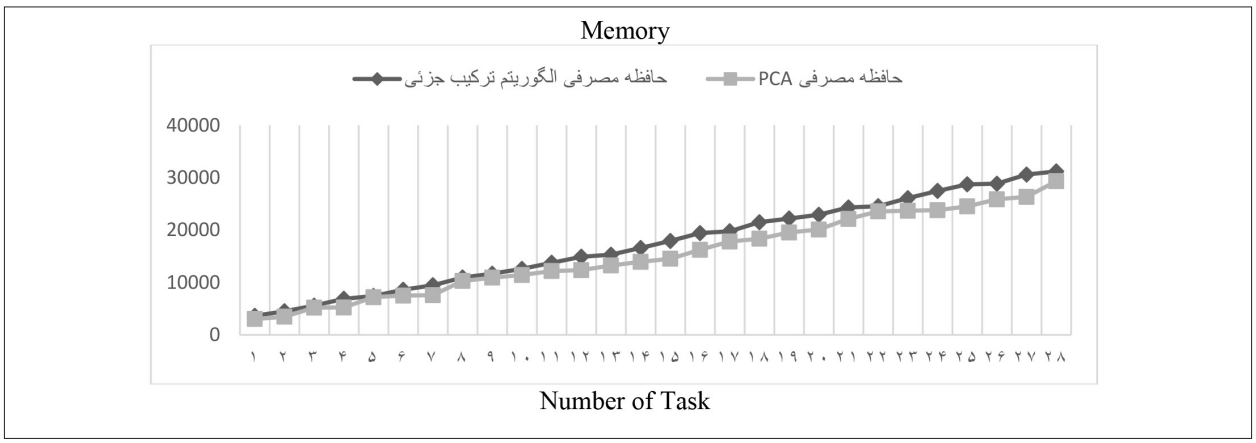
می‌توان مشـــاهده نمود که اندازه‌هـــای فضای حافظه مصرف‌شده و زمان‌های پاسخ هردو رویکرد، با افزایش تعداد سرویس‌ها یا افزایش تعداد سرویس‌های نامزد افزایش می‌یابنـــد. بااین‌حال، اندازه فضـــا و زمان اجرا

جدول ۲: مقایسه کارایی روش ترکیب سرویس مبتنی بر الگوریتم ترکیب جزئی و تحقیق پیش‌رو

تعداد نمونه ها	زمان پاسخ درروش پیشنهادی	حافظه مصرفی درروش پیشنهادی	زمان پاسخ الگوریتم ترکیب جزئی[۵]	حافظه مصرفی الگوریتم ترکیب جزئی[۵]
T=5,Service=10	113.816 ms	10303 byte	123.131 ms	byte10960
T=5,Service=20	143.081 ms	18353 byte	186.417 ms	21516 byte
T=5,Service=30	230.353 ms	29336 byte	302.401 ms	31208 byte
T=10,Service=5	272.577 ms	12953 byte	305.584 ms	3875 byte
T=20,Service=5	294.5 ms	16775 byte	322.916 ms	18012 byte
T=30,Service=5	774.005 ms	26549 byte	900.553 ms	29563 byte



شکل۲: مقایسه زمان پاسخ دو الگوریتم (تعداد Task از ۳ تا ۳۰ متغیر و تعداد سرویس نامزد عدد ثابت ۵)

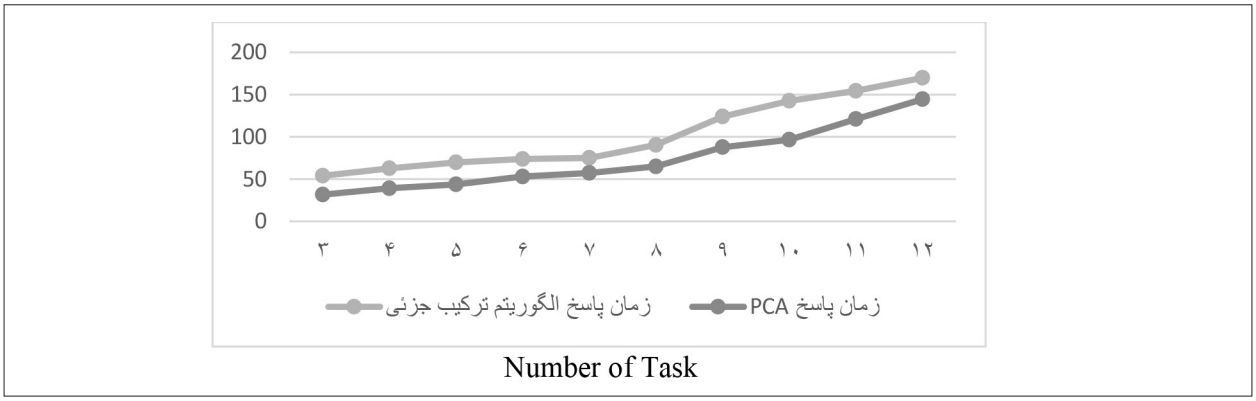


شکل ۳: مقایسه فضای حافظه مصرفی (تعداد Task از ۳ تا ۳۰ متغیر و تعداد سرویس نامزد عدد ثابت ۵)

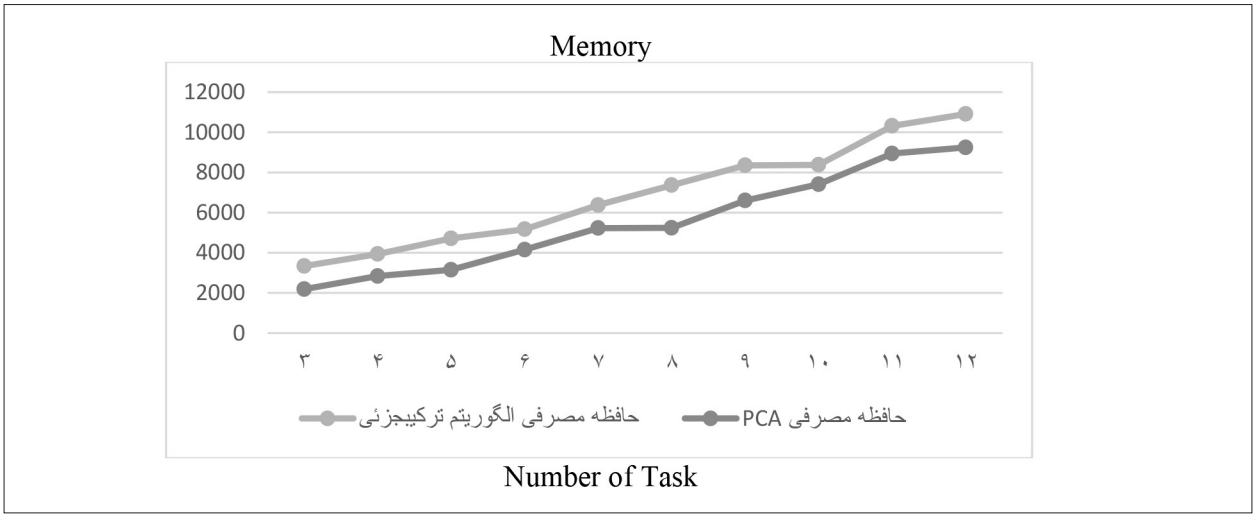
در روش PCA نسبت به الگوریتم ترکیب جزئی رشد کمتری دارد و این بیانگر کارایی بهتر الگوریتم PCA ا ست.

۸-نتیجه گیری

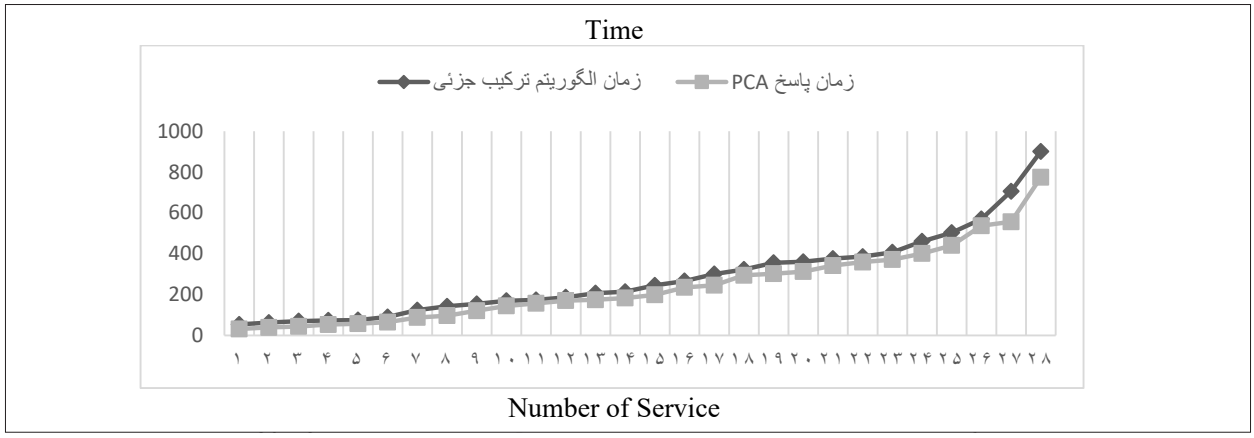
هدف از ترکیب مناسب سرویس، در مرحله اول، انتخاب سرویس‌های مناسب با در نظر گرفتن ارضای محدودیت‌های پارامتر کیفی و ارجحیت نیاز کاربر به‌گونه‌ای است که سرویس‌های مناسب کمتری در عملیات مقایسه و انتخاب سرویس‌ها از بین بروند. لذا در روش پیشنهادی، سرویس‌ها ازنظر ترکیب‌پذیری براساس ورودی‌ها و محدودیت‌های پارامتر کیفی و همچنین محدودیت اعمال‌شده کاربر بررسی و پالایش شده‌اند. با اعمال این محدودیت‌ها، ترکیب فقط روی مجموعه سرویس‌هایی انجام گردید که



شکل ۴: مقایسه زمان پاسخ دو الگوریتم (تعداد Task متغیر تعداد سرویس ثابت)

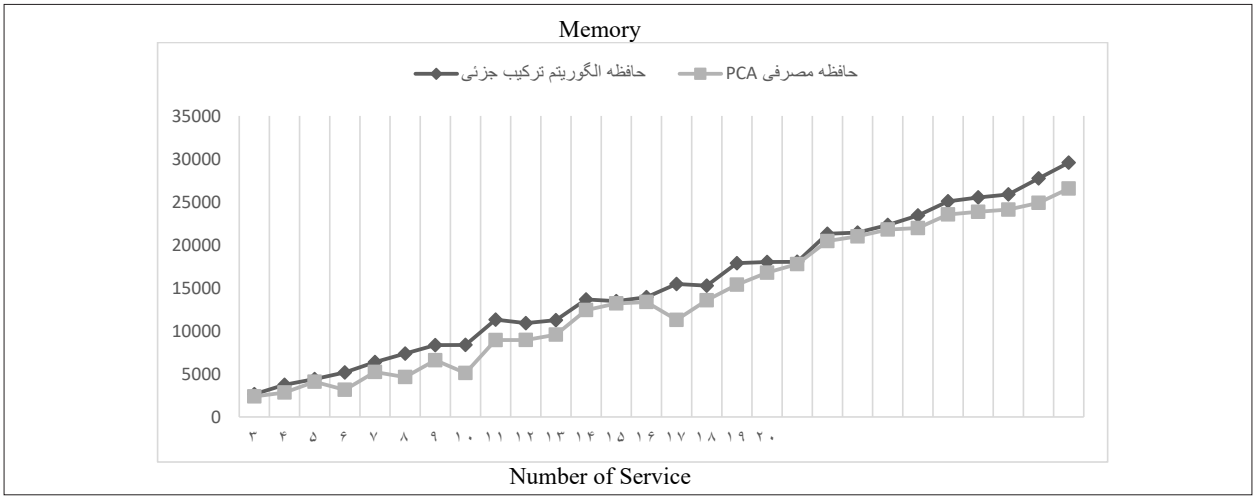


شکل ۵: مقایسه فضای حافظه مصرفی(تعداد Task ثابت و سرویس متغیر)

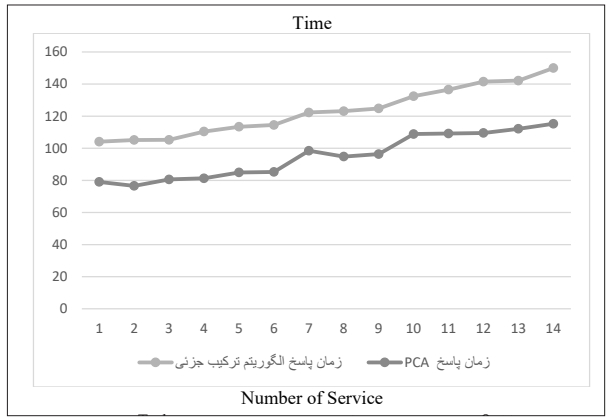


شکل ۶: مقایسه زمان پاسخ دو الگوریتم(تعداد سرویس عدد ثابت ۵ و تعداد سرویس نامزد از ۳ تا ۳۰ متغیر)

مطابق درخواست کاربر است. این امر باعث کاهش تعداد سرویس‌های نامزد و کاهش فضای جستجوی سرویس شد. همچنین استفاده از الگوریتم PCA، ویژگی‌های پارامترهای کیفی در هر سرویس نامزد ادغام‌شده و بعد از عملیات مرتب‌سازی و هرس، تعداد سرویس‌های نامزد و فضای جستجوی ترکیب به‌صورت چشمگیری کاهش یافت. در ادامه، می‌توان شرایطی را در نظر داشت که ویژگی‌های کیفی بیشتری بررسی شود. همچنین از معایب



شکل ۷: مقایسه فضای حافظه مصرفی(تعداد سرویس عدد ثابت ۵ و تعداد سرویس نامزد از ۱ تا ۳۰ متغیر)



شکل ۸: مقایسه زمان پاسخ دو الگوریتم(تعداد سرویس ثابت و تعداد Task متغیر)

شکل ۹: مقایسه فضای حافظه مصرفی(تعداد سرویس ثابت و تعداد Task متغیر)

روش پیشنهادی، عدم مقیاس پذیری سرویس‌های ترکیبی و ترکیب سرویس‌ها به‌صورت ایستا است که جهت بهبود این دو مشـــکل می توان بهبودهایی را در روش پیشنهادی اعمال نمود.

منابع

[1] Sheng, Q. Z., Qiao, X., Vasilakos, A. V., Szabo, C., Bourne, S., & Xu, X. Web services composition: A decade's overview. Information Sciences, 280, 218-238,2014.

[2] Chen, Y., Huang, J., Lin, C., & Hu, J. A partial selection methodology for efficient qos-aware service composition. IEEE Transactions on Services Computing, 8(3), 384-397, 2015.

[3] Jian, X., Zhu, Q., & Xia, Y. An interval-based fuzzy ranking approach for QoS uncertainty-aware service composition. Optik-International Journal for Light and Electron Optics, 127(4), 2102-2110, 2016.

[4] Wang, D., Yang, Y., & Mi, Z. A genetic-based approach to web service composition in geo-distributed cloud environment. Computers & Electrical Engineering, 43, 129-141, 2015.

[5] Kurdi, H., Al-Anazi, A., Campbell, C., & Al Faries, A. A combinatorial optimization algorithm for multiple cloud service composition. Computers & Electrical Engineering, 42, 107-113, 2015.

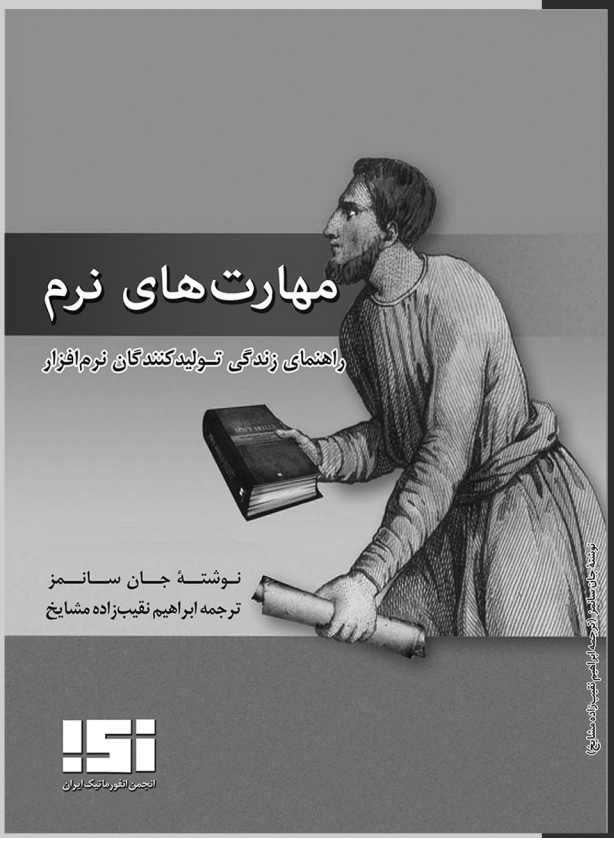
[6] Vladimir, K., Budiselić, I., & Srbljić, S, Consumerized and peer-tutored service composition. Expert Systems with Applications, 42(3), 1028-1038. 2015 .

[7] Puttonen, J., Lobov, A., Soto, M. A. C., & Lastra, J. L. M. , Planning-based semantic web service composition in factory automation. Advanced Engineering Informatics, 29(4), 1041-1054, 2015.

[8] Liu, X., Ma, Y., Huang, G., Zhao, J., Mei, H., & Liu, Y,Data-Driven Composition for Service-Oriented Situational Web Applications. IEEE Transactions on Services Computing, 8(1), 2-16, 2015.

[16] Zhao, Y., Tan, W., & Jin, T. (2017, September). QoS-aware Web Service Composition Considering the Constraints between Services. In Proceedings of the 12th Chinese Conference on Computer Supported Cooperative Work and Social Computing (pp. 229-232). ACM.
[17] Hayyolalam, V., & Kazem, A. A. P. (2018). A systematic literature review on QoS-aware service composition and selection in cloud environment. Journal of Network and Computer Applications.
[18] S.K. Gavvala et al. (2019). QoS-aware cloud service composition using eagle strategy. The International Journal of Advanced Manufacturing Technology, 90 (2019) 273–290.
[19] Yaghoubi, M., & Maroosi, A. (2020). Simulation and modeling of an improved multi-verse optimization algorithm for QoS-aware web service composition with service level agreements in the cloud environments. Simulation Modelling Practice and Theory, 102090.
[20] Khanouche, M. E., Gadouche, H., Farah, Z., & Tari, A. (2020). Flexible QoS-aware services composition for service computing environments. Computer Networks, 166, 106982
[9] Yu, Q., Chen, L., & Li, B, Ant colony optimization applied to web service compositions in cloud computing. Computers & Electrical Engineering, 41, 18-27, 2015.
[10] Li, J., Yan, Y., & Lemire, D. (2018). Full solution indexing for top-k web service composition. IEEE Transactions on Services Computing, vol.11, no.13, 521-533.
[11] Zhu, J., et al., Online QoS Prediction for Runtime Service Adaptation via Adaptive Matrix Factorization. IEEE Transactions on Parallel and Distributed Systems, 2017.
[12] Jatoth, C., et al., QoS-aware Big service composition using MapReduce based evolutionary algorithm with guided mutation. Future Generation Computer Systems, 2017.
[13] Lei, Y. and Z. Junxing, Service composition based on multi-agent in the cooperative game. Future Generation Computer Systems, 2017. 68: p. 128-135.
[14] Liu, Y., Yang, R., & Zhang, S. (2017). Service Selection Method based on Skyline in Cloud Environment. International Journal of Performability Engineering, 13(7).
[15] Mezni, H., & Sellami, M. (2017). Multi-cloud service composition using formal concept analysis. Journal of Systems and Software, 134, 138-152.

جدیدترین کتاب
از انتشارات انجمن انفورماتیک ایران
منتشر شد!



مهارت‌های نرم

برای تهیهٔ کتاب با دفتر انجمن انفورماتیک ایران
تماس بگیرید ۶۶۴۱۲۸۶۱

چاپ اول