

تاریخ دریافت مقاله: ۹۵/۱۱/۲۹  
تاریخ پذیرش مقاله: ۹۶/۰۲/۱۰

## ترکیب وب سرویس خودکار آگاه از QoS بر مبنای عوامل همیار با استفاده از الگوریتم دسته بندی Top-k

سمیه لطف محمدی

کارشناس ارشد مهندسی کامپیوتر، دانشگاه آزاد اسلامی، واحد یزد، یزد، ایران

پست الکترونیکی: s\_lotfemohamady@yahoo.com

سیما عمادی\*

استادیار گروه مهندسی کامپیوتر، دانشگاه آزاد اسلامی، واحد یزد، یزد، ایران

پست الکترونیکی: emadi@iauyazd.ac.ir

### چکیده

تا کنون راهکارهای زیادی برای ارتباطات بین سیستم‌های نرم افزاری مطرح گردیده که ترکیب سرویس‌ها یکی از این روش‌ها است. با توجه به رشد روز افزون و فراگیر معماری سرویس‌گرا و سیستم‌های تحت وب، انتخاب وب سرویس‌های مناسب و نحوه ترکیب آن‌ها مسئله‌ای حائز اهمیت است. با توجه به توانایی‌های عوامل هوشمند می‌توان از این عوامل در برآورده شدن نیازهای کنونی سیستم‌ها و افزایش سرعت، دقت، کیفیت و بازدهی آن‌ها استفاده نمود. به دلیل فراگیری سیستم‌های سرویس‌گرا و توانمندی عوامل هوشمند، مطالعه در مورد وب سرویس‌ها و ترکیب آن‌ها بر اساس عامل‌های هوشمند بسیار ضروری و مورد توجه است.

روش‌های زیادی برای ترکیب وب سرویس‌ها ارائه شده‌است که تنوع در انتخاب، سرعت و مصرف حافظه را به ویژه در هنگام انفجار فضای حالت، در کنار هم لحاظ ننموده‌اند. نظر به اهمیت در دسترس پذیری و پاسخ‌گویی بدون وقفه در سیستم‌های تحت وب، ایجاد چندین

ترکیب بهینه می‌تواند مؤثر واقع شود. بنابراین در این مقاله سعی بر آن گردیده است که روشی مناسب برای ترکیب سرویس‌های اتمی بر مبنای عوامل همیار با لحاظ نمودن پارامتر کیفیت سرویس به کار گرفته شود. این روش با استفاده از الگوریتم Top-k و ساختار موازی آن می‌تواند چندین راه حل مناسب به دست آورد تا در مواقع لزوم از قابلیت جایگزینی این راه‌ها به جای بهترین ترکیب استفاده نماید. همچنین پالایش داده‌ها قبل از ایجاد ترکیب در داده‌هایی با مقیاس بزرگ، مجموعه سرویس‌های انتخابی را کاهش می‌دهد. بنابراین نتایج حاصل از تحلیل و ارزیابی روش پیشنهادی، بهینه‌سازی مطلوبی در زمان اجرا و مصرف حافظه نمایان می‌سازد. رویکرد این تحقیق در تمامی مؤسسات و سازمان‌هایی که به ارائه خدمات خود از طریق وب سرویس می‌پردازند، بسیار مؤثر و کارآمد می‌باشد. بدیهی است استفاده از این روش می‌تواند با سرعت و کیفیت بهینه‌ای پاسخ لازم برای درخواست و پرس و جوی کاربر را فراهم نماید.

واژه‌های کلیدی: وب سرویس، ترکیب وب سرویس‌ها، عامل‌های همیار، آگاه از کیفیت سرویس، الگوریتم Top-k.

\* نویسنده مسئول

تکرار را به حداقل می‌رساند. انتخاب تعداد عوامل به صورت پویا، سرعت و دقت در ایجاد ترکیب‌های سرویس را افزایش می‌دهد [۲].

از آنجایی که ایجاد یک ترکیب از تعداد زیاد وب سرویس‌ها ممکن است در بعضی از موارد باعث عدم دسترس‌پذیری و توقف سیستم‌های تحت وب شود، از این رو با فیلتر نمودن داده‌ها در این الگوریتم از تعداد سرویس‌های نامزد کاسته می‌شود و بهینه‌سازی قابل توجهی در زمان اجرا و مصرف حافظه پدید می‌آید. بنابراین می‌توان بیان نمود استفاده از روشی که با کاهش مجموعه داده‌های انتخابی، چندین ترکیب وب سرویس بر مبنای عوامل با در نظر گرفتن پارامترهای کیفیت سرویس به دست آورد، سبب تضمین کارآیی، انعطاف‌پذیری و بهبود دقت و سرعت در سیستم‌های مذکور می‌گردد [۳].

همان‌طور که بیان شد با در نظر گرفتن پویایی بالا و رشد سریع در تعداد وب سرویس‌هایی که از نظر کارکرد مشابه هستند، یافتن ترکیب سرویس وب بهینه در زمان مناسب که نیازهای کاربر را برآورده سازد به کاری چالش برانگیز تبدیل گردیده است و مورد توجه می‌باشد.

بنابراین بدیهی است که به‌کارگیری روش‌هایی که بتواند مراحل ایجاد سرویس مرکب را بهینه‌تر و کارآتر به انجام برساند بسیار مفید واقع می‌شود. براهمی و گمودی<sup>۵</sup> روشی ارائه داده اند [۲] که مجموعه‌ای از عوامل خودکار همیارانه است و ترکیب سرویس‌ها را با استفاده از خودسازماندهی عوامل در گراف وابستگی به نام عامل شبکه اجتماعی و پروتکل همیارانه بین آن‌ها، انجام می‌دهد. در تحقیق حاضر با بررسی رویکرد ارائه شده توسط براهمی و گمودی مشخص گردید که این روش در داده‌هایی با حجم وسیع، کارایی لازم را نخواهد داشت لذا با افزودن الگوریتم Top-K به آن، راهحل مناسبی برای ترکیب سرویس به دست می‌آید که در مقایسه با روش ارائه شده پیشین از قابلیت‌های بیشتری برخوردار می‌باشد.

در سال‌های اخیر استفاده از سیستم‌های سرویس‌گرا در مؤسسات و سازمان‌ها رو به افزایش بوده است. قابلیت این سیستم‌ها و انعطاف‌پذیری آن‌ها دلیلی بر روند رو به رشدشان می‌باشد.

سیستم‌های سرویس‌گرا به دلیل امکان فعالیت در محیط‌های توزیع شده ناهمگون از اهمیت ویژه‌ای برخوردارند. کاربر این گونه سیستم‌ها از سرویس‌هایی که مؤلفه‌های سیستم ارائه می‌دهند، استفاده می‌کند [۱].

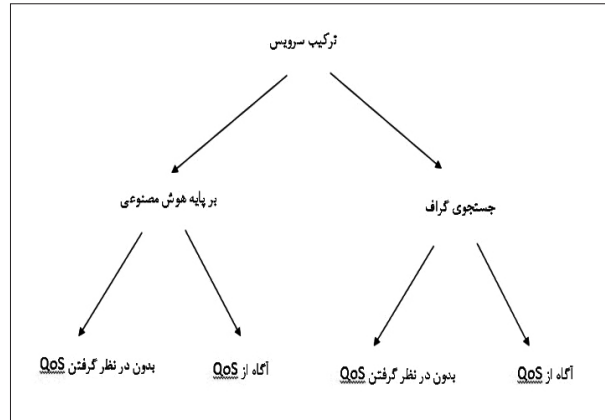
وب سرویس‌ها در محیط‌های تحت وب به عنوان برنامه‌های کاربردی<sup>۱</sup> می‌توانند پاسخگوی بسیاری از نیازهای کاربران متقاضی باشند. در بیشتر مواقع نیازهای کاربر توسط یک وب سرویس اتمی برآورده نمی‌شود، به همین دلیل لازم است که روش‌هایی برای ترکیب وب سرویس‌ها به کار گرفته شود تا با ساختن سرویسی مرکب و با ارزش افزوده از سرویس‌های ساده، به نیازهای رو به رشد کاربران پاسخ داده شود. بدیهی است در نظر گرفتن پارامترهای کیفیت سرویس<sup>۲</sup> به عنوان نیازهای غیرعملکردی<sup>۳</sup> سیستم در حین ترکیب بسیار حائز اهمیت می‌باشد [۱].

علاوه بر این محیط وب سرویس‌ها با تعداد زیاد سرویس‌های وب و پویایی بالا به عنوان دو مسئله اصلی روبرو می‌باشند؛ بدین معنی که برای یک سرویس می‌تواند رویدادهایی مانند تعویق یا تغییر معیار کیفیت سرویس اتفاق بیفتد. به کارگیری یک سری عوامل<sup>۴</sup> در ترکیب وب سرویس‌ها می‌تواند هوشمندی، دقت و سرعت مطلوبی را به این روش‌ها بیفزاید و به ارائه ترکیب سرویس خودکار بهینه‌ای برحسب مقیاس‌پذیری و دقت کمک نماید.

به‌کارگیری یک گروه از عوامل همیار کنار یکدیگر در ترکیب، سبب پاسخگویی به نیازهای مشترک، اشتراک‌گذاری مالکیت و تصمیم‌گیری آزاد می‌گردد. همچنین

1- Applications  
2- Quality of Service(QoS)  
3- Non-functional  
4- Agents

ساختار مقاله در ادامه به این شرح است: در بخش دو، مفاهیم پایه و کارهای مرتبط بیان می‌شود و روش‌های ارائه شده در سال‌های اخیر مورد بررسی قرار می‌گیرد، سپس در بخش سه، روش پیشنهادی شرح داده می‌شود. در بخش بعدی پیاده‌سازی و ارزیابی روش پیشنهادی ارائه می‌گردد. در پایان نتایج حاصل از این تحقیقات، و پیشنهاداتی برای کارهای آینده بیان می‌شود.



شکل ۱: روش‌های ترکیب خودکار وب سرویس

## ۲- کارهای مرتبط

### ۲-۱- روش‌های ترکیب خودکار وب سرویس

روش‌های ترکیب خودکار وب سرویس‌ها، انتخاب و اتصال وب سرویس اتمی را به صورت خودکار توسط برنامه انجام می‌دهند. روش‌هایی ترکیب خودکار وب سرویس‌ها را می‌توان طبق شکل (۱) به دو بخش عمده طبقه بندی نمود:

روش‌های ترکیب بر اساس برنامه ریزی هوش مصنوعی، راه‌حلی بر اساس روش هوش مصنوعی ارائه می‌دهند. مفاهیم برنامه ریزی هوش مصنوعی ساختارهای کنترل پیچیده با حلقه‌های شرطی و غیر شرطی دارند و در زمان اجرا ممکن است اشیاء جدید تولید نمایند.

روش‌های ترکیب بر پایه برنامه ریزی هوش مصنوعی مشکلات زیر را دارد [۴،۵،۶]:

- کارآیی پایین زمانی که فضای جستجو گسترش می‌یابد.
- به دلیل در نظر نگرفتن ساختار موازی، تولید نتیجه ترکیب به عنوان یک ترتیبی از سرویس‌ها راه حل بهینه را پیشنهاد نمی‌دهد.

روش‌های جستجوی گراف در ابتدا یک گراف وابستگی<sup>۷</sup> بر اساس درخواست کاربر می‌سازند و سپس با استفاده از الگوریتم جستجو راه حل را پیدا می‌کنند. علیرغم

هدف از این مقاله ارائه روشی برای ترکیب وب سرویس‌ها بر پایه عوامل با استفاده از الگوریتم Top-k می‌باشد.

با ایجاد ترکیب Top-k می‌توان به صورت موازی عمل نمود و به پرس و جوی کاربر پاسخ داد. همچنین در مواردی که سرویس نامعتبر است، قادر به انتخاب ترکیب دیگری از میان Top-k به عنوان جایگزین خواهد بود. استفاده از این الگوریتم از افزایش بار جلوگیری نموده و عدم تنزل ترکیب سرویس بهینه را در بر دارد.

برای تحلیل و ارزیابی روش ارائه شده در این تحقیق با نام CAUT-K ابتدا پیاده‌سازی رویکرد ارائه شده برامی و گمودی بررسی گردید و سپس نتایج مورد نظر با داده‌های یکسان بر روی الگوریتم‌های ارائه شده پیشین و الگوریتم پیشنهادی CAUT-K تحلیل شد.

با بررسی هر دو روش بر روی مجموعه داده یکسان، مشخص گردید که رویکرد ارائه شده در این تحقیق، علاوه بر کلیه مزایای ایجاد ترکیب مبتنی بر عوامل و استفاده از الگوریتم موازی Top-K، در تمام موارد، زمان اجرا و مصرف حافظه بهتری نسبت به روش ارائه شده در الگوریتم معتبر پیشین دارد.

در حال حاضر کلیه مؤسسات و سازمان‌هایی که ارائه دهنده خدمات از طریق وب هستند و در آینده تمامی مؤسسات و سازمان‌ها، بهره‌وران این رویکرد خواهند بود.

شباهت‌های موجود، این روش‌ها، در نحوه جستجو، کارایی و کیفیت راه حل به دست آمده، تفاوت دارند. به عنوان مثال الگوریتم ارائه شده توسط گونکالوس و دیگران<sup>۸</sup> یک گراف متناظر با درخواست کاربر ایجاد می‌نماید [۷]. عیب این روش این است که به یک ترکیب ساختاری متوالی محدود شده است.

یو و همکاران یک ایده قابل توجه و کارا بر پایه الگوریتم حریصانه برای ایجاد طرح ترکیب پیشنهاد نمودند که تنها ساختار متوالی را لحاظ می‌نماید [۸].

در روشی یک الگوریتم جستجوی A\* ارائه شده است که یک سرویس مرکب اجرایی بدون در نظر گرفتن خصوصیات غیرعملکردی برای انتخاب سرویس تعیین می‌کند [۹].

کاوون و لی<sup>۹</sup> الگوریتمی دو مرحله‌ای پیشنهاد داده‌اند [۱۰] که راه حل‌هایی بدون وب سرویس‌های زاید با به کار بردن نشانه‌ها<sup>۱۰</sup> ارائه می‌دهد. در این سیستم یک شاخص پیوند بر روی وب سرویس‌ها بر طبق ارتباط آن‌ها ساخته می‌شود. این سیستم با وجود کارایی بهتر، حداقل تعداد وب سرویس‌ها در ترکیب را تضمین نمی‌کند و همچنین خصوصیات غیرعملکردی شامل کیفیت سرویس برای وب سرویس‌های ترکیب شده را مورد توجه قرار نمی‌دهد.

## ۲-۲- روش‌های ترکیب وب سرویس آگاه از کیفیت سرویس

خصوصیات کیفیت سرویس یا QoS می‌توانند به دو رده طبقه‌بندی شوند. دسته‌ای از این خصوصیات منفی می‌باشند به این معنی که هر چه مقدار آن‌ها بالاتر باشد، کیفیت کمتر است مانند زمان پاسخ و هزینه. دسته دوم خصوصیات مثبت هستند که مثبت می‌باشند یعنی هر چقدر مقدار آن‌ها بالاتر باشد، کیفیت بیشتر است مانند توان عملیاتی<sup>۱۱</sup> و در دسترس پذیری [۲].

8- Kwon, J., & Lee, D  
9- Goncalves & others  
10-Tokens  
11- Throughput

روش‌های گوناگونی برای حل مسائل ترکیب وب سرویس آگاه از کیفیت سرویس ارائه شده است که بر مبنای الگوریتم‌های مختلفی از جمله الگوریتم ژنتیک<sup>۱۲</sup> [۱۱،۱۲]، تصمیم‌گیری چند معیاره<sup>۱۳</sup> [۱۳]، عامل‌های خودکار<sup>۱۴</sup> [۱۴،۱۵،۱۶]، تجربی<sup>۱۵</sup> [۱۷،۱۸]، برنامه ریزی عدد صحیح<sup>۱۶</sup> [۱۳] هستند. بیشتر روش‌های ارائه شده، مسئله ترکیب خودکار را در نظر می‌گیرند اما با مشکل پویایی و قابلیت گسترش مواجه می‌باشند.

جیانگ و همکاران<sup>۱۷</sup> به مسئله ترکیب سرویس خودکار آگاه از QoS می‌پردازند [۱۹] که مزیت ادغام مراحل ترکیب سرویس و انتخاب سرویس را دارا می‌باشد.

لیو و همکاران<sup>۱۸</sup> یک روش برای ترکیب وب سرویس پویا بر مبنای تجزیه محدودیت‌های کیفیت سرویس کلی ارائه داده‌اند [۲۰] که شامل سه مرحله است. مرحله اول، محدودیت‌های QoS کلی به محدودیت‌های محلی به وسیله یک الگوریتم جدید به نام الگوریتم Culture Genetic تجزیه می‌شوند.

در مرحله دوم، قوانین تخمین مقدار QoS طراحی شده‌اند تا مقدار QoS سرویس‌های نامزد را تخمین بزنند. در مرحله سوم بهترین وب سرویس‌هایی که می‌توانند محدودیت‌های کلی انتخاب شده برای هر وظیفه در طول زمان اجرا را بهبود ببخشند، ترکیب می‌شوند.

روش ارائه شده توسط هاگ و وینهاردت<sup>۱۹</sup> برای سرویس‌های فناوری اطلاعات سفارشی تنظیم شده است [۲۱] که به عنوان یک همکاری مشترک بین ارائه دهنده و مصرف کننده می‌باشد. براساس فرآیند، سرویس سفارشی پیشنهادی شامل سه مرحله درخواست فرمول، مهندسی سرویس و بهینه سازی اقتصادی می‌باشد.

در این تحقیق تغییرات مختلف روش‌های تجربی و

12- Genetic Algorithm  
13- Multiple Criteria Decision Making  
14- Autonomous Agents  
15- Heuristic  
16- Integer Programming  
17- Jiang & others  
18- Liu & others  
19- Haak & Weinhardt

دقیق هر دو از الگوریتم‌های گراف گرا و روش‌های برنامه ریزی عدد صحیح با جزئیات ارائه شده است.

نتایج ارائه شده در این تحقیق سه محدودیت عمده دارد: اول، نمی‌توان ادعا نمود که روش‌های ارائه شده تنها راه مناسب برای حل مسئله حاضر است. دوم، ثابت شده است که روش‌های بهینه‌سازی و تخمین ارائه شده در مورد زمان اجرا و کیفیت راه حل، بهینه هستند. سوم، یک راه حل برای صفات غیر بلمن جدا از الگوریتم‌های جستجوی فراگیر ارائه شده است. صفات غیر بلمن، صفاتی هستند که از الگوریتم بلمن نمی‌توان کوتاهترین مسیر را در گراف برای آن‌ها به دست آورد. با این وجود از طریق یافته‌های این کار، بهینه‌سازی کارآمد مشکلات پیکربندی سرویس را می‌توان در بسیاری از تنظیمات استاندارد به دست آورد. روش ارائه شده توسط یو و لین<sup>۲۰</sup> [۲۲] موضوع محدودیت کیفیت سرویس نقطه پایانی به پایانی از سرویس مرکب را با استفاده از واسطه‌های QoS که مسئول انتخاب و هماهنگ کردن مؤلفه‌های سرویس فردی هستند، مورد مطالعه قرار می‌دهد.

در تحقیق مذکور دو روش برای حل مسئله انتخاب سرویس ارائه شده است: رویکرد ترکیبی به وسیله مدل سازی مشکل به عنوان مسئله کوله پشتی چند انتخابی (MCKP) و روش گراف به وسیله مدل سازی مشکل به عنوان مسئله کوتاهترین مسیر محدود در تئوری گراف. الگوریتم‌های کارآمدی برای هر دو روش ارائه و عملکرد آن‌ها مقایسه و مطالعه شده است. انتخاب الگوریتم‌ها به اندازه مسئله، ساختار شبکه و فاکتورهای دیگر بستگی دارد. هر دو الگوریتم یک زمان اجرای مناسب برای مسائل با مجموعه داده بزرگ دارند [۲۲].

## ۲-۳- روش‌های ترکیب وب سرویس مبتنی بر عامل‌های همیار

تحقیق ارائه شده توسط کیو و همکاران<sup>۲۱</sup> [۲۳]، یک معماری جدید، با استفاده از اینترنت، عامل و فناوری

وب معنایی ارائه می‌دهد. چهارچوب پیشنهادی با قابلیت تنظیم پویا و بسیار انعطاف پذیر می‌باشد. در این روش، بر اساس مفهوم ارائه سرویس، گروه‌های عامل وب هوشمند، به عنوان ستون فقرات برای سازماندهی اجرای مبتنی بر سرویس سیستم عمل می‌کنند.

شفیق و همکاران<sup>۲۲</sup> رویکردی در رابطه با اتصال سیستم‌های چند عاملی و وب سرویس‌ها پیشنهاد نمودند [۲۴] که بر اساس همکاری بین عامل‌های نرم افزاری و وب سرویس‌های معنایی انجام گردید. این روش برای سازگار نمودن سیستم‌های چند عاملی با استانداردهای موجود برای وب سرویس‌ها بدون تغییر مشخصات، کارکردها و پیاده‌سازی‌های فعلی آن‌ها، راهکاری ارائه می‌دهد.

روش وانگ و همکاران<sup>۲۳</sup>، به رویکرد جدیدی مبتنی بر اجتماع برای انتخاب وب سرویس پرداخته است [۲۵] که عامل‌های برتر با قابلیت‌های بیشتری در آن به عنوان مدیران جامعه فعالیت می‌کنند. تشکیل جوامع می‌تواند به عوامل برای پیدا کردن اطلاعات با ارزش‌تر کمک کند. روش پاداش عملی به عنوان یک عامل پشتیبان با ایجاد انگیزه در عامل‌های برتر، به اشتراک گذاری منابع آن‌ها کمک می‌نماید و اعتباری مبتنی بر اجتماع برای رویکرد پیشنهادی فراهم می‌نماید. یکی از موارد در نظر گرفته نشده در این روش قابلیت اعتماد سوپر عوامل به صورت سرویس گرا، می‌باشد.

ال او هد و همکاران<sup>۲۴</sup> به بررسی یک سرویس اکتشافی برای ترکیب خودکار وب پرداخته‌اند [۲۶]. عملکرد این فرآیند ترکیب شامل دو لایه اکتشاف و لایه اجرا/انتخاب می‌باشد. لایه اکتشاف که قصد دارد یک الگوی ترکیب را ایجاد کند و لایه اجرا که یک سیستم چند عاملی را به کار می‌برد تا وب سرویس‌ها را انتخاب و سرویس مرکب را اجرا نماید. در این روش ترکیبات متنوعی در اختیار کاربران قرار نمی‌گیرد تا ترجیحات آنان را برآورده سازد.

22- Shafiq & others

23- Wang & others

24- El Ouahed & others

20- Yu & Lin

21- Qiu & others

جدول ۱: مقایسه روش‌های بررسی شده

| توصیف روش                                    | مزایا  | معایب  |
|--|--|--|
| ترکیب بر پایه برنامه ریزی هوش مصنوعی (AI)    | ایجاد سرویس مرکب از سرویس‌های اتمی و ساده  | کارایی پایین در هنگام گسترش فضای جستجو به دلیل در نظر نگرفتن ساختار موازی، نتیجه راه حل بهینه را پیشنهاد نمی‌دهد   |
| ترکیب بر پایه جستجوی گراف                    | امکان استفاده از الگوریتم جستجو برای یافتن ترکیب مناسب<br>ارائه راه حل بدون وب سرویس‌های زائد با به کار بردن توکن‌ها   | محدودیت به یک ترکیب ساختاری متوالی<br>عدم توجه به خصوصیات غیرعملکردی شامل کیفیت سرویس<br>عدم تضمین حداقل تعداد وب سرویس‌ها در ترکیب  |
| ترکیب آگاه از کیفیت سرویس                    | در نظر گرفتن خصوصیات غیرعملکردی شامل کیفیت سرویس   | مشکل پویایی و قابلیت گسترش   |
| ترکیب بر پایه عوامل                          | سازگار نمودن سیستم‌های چند عاملی با استانداردهای موجود برای وب سرویس‌ها<br>انتخاب و ترکیب وب سرویس‌ها به وسیله عامل‌های نرم افزاری هوشمند به صورت فعال و پویا<br>یافتن اطلاعات با ارزش تر و انتخاب وب سرویس مؤثرتر<br>ترجیح سرویس‌ها با بیشترین تناسب<br>کاهش تعداد سرویس‌های شرکت کننده در راه حل<br>تقلیل زمان پاسخ<br>ایجاد ترکیبی دقیق به صورت فعال در محیط پویا | قرار ندادن ترکیبات متنوع در اختیار کاربران و برآورده نمودن ترجیحات آنان<br>عدم کارایی به هنگام انفجار فضای حالت<br>عدم امکان انتخاب ترکیبات دیگر در زمان عدم دسترسی به سرویس<br>مصرف حافظه زیاد در داده‌هایی با مقیاس بزرگ |
| ترکیب با استفاده از الگوریتم دسته بندی Top-K | امکان پذیری تنوع تعادل بار   | عدم توجه به تقلیل زمان پاسخ<br>در نظر نگرفتن دقت و سرعت  |

## ۲-۴- روش‌های ترکیب وب سرویس با استفاده از

Top-K

در تحقیق ارائه شده توسط لطف محمدی و عمادی به بررسی روش‌هایی از ترکیب وب سرویس با به کار بردن الگوریتم Top-K پرداخته شده است. چندین روش ترکیب سرویس با استفاده از Top-K مطالعه و بررسی گردید. سپس این روش‌ها با یکدیگر مقایسه و مزایا و معایب هر کدام بیان شد [۲۷].

دنگ و دیگران<sup>۲۶</sup> روشی نوین برای مسئله ترکیب خودکار آگاه از QoS وب سرویس‌های معنایی با استفاده از Top-K ارائه نموده‌اند [۲۸]. روش مذکور شامل سه مرحله متوالی جستجوی رو به جلو<sup>۲۷</sup>، محاسبه کیفیت سرویس محلی بهینه و جستجوی عقب گرد می‌باشد: روش دنگ و دیگران<sup>۲۸</sup> دو موضوع مقیاس بزرگ و

در تحقیق براهمی و گمودی ترکیب وب سرویس به صورت خودکار انجام شد [۲] و روشی پیشنهاد گردید که آگاه از QoS و بر مبنای عامل‌های همیار عمل می‌نماید.

معماری روش ارائه شده شامل سه لایه اصلی سرویس، کارخواه<sup>۲۵</sup> و ترکیب می‌باشد و بر دو مرحله اصلی استوار است:

۱- خود سازماندهی عوامل در گراف وابستگی به نام عامل شبکه اجتماعی.

۲- رایانش توزیع شده ترکیب سرویس‌های وب بهینه به وسیله پروتکل همیارانه بین عوامل.

در این روش، زمان پاسخ به وسیله بهره برداری مشترک توزیع و ایجاد توافق به وسیله فناوری عامل تقلیل می‌یابد. این رویکرد با افزایش تعداد سرویس‌ها و انفجار فضای حالت کارایی لازم را نخواهد داشت، همچنین در زمانی که وب سرویس نامعتبر بوده یا در دسترس نباشد، ترکیب دیگری را نمی‌توان به عنوان جایگزین انتخاب نمود.

26- Deng & others

27- Forward Search

28- Deng & others

25- client



ترکیب سرویس بر مبنای QoS با راه حل های Top-K را پوشش می دهد [۳]. در این تحقیق مسئله ترکیب سرویس به یک جستجوی گراف تبدیل شده است و الگوریتم ترکیب بر اساس ترکیب جستجوی عقب گرد و جستجوی ژرفای<sup>۲۹</sup> می باشد که می تواند به یک روش موازی اجرا شود؛ تا بهترین ترکیبات K را بر طبق رفتار کیفیت سرویس برگرداند.

در روش های فوق تقلیل زمان پاسخ، دقت و سرعت لحاظ نگردیده است.

## ۲-۵- مقایسه روش ها

در جدول ۱ روش های بررسی شده در زمینه ترکیب سرویس ها به صورت خلاصه بیان گردیده است تا مقایسه آن ها راحت تر و سریع تر انجام شود.

### ۳- مدل پیشنهادی CAUT-K

روش ارائه شده در این تحقیق با عنوان CAUT-K نامگذاری شد. در این روش فرآیند ترکیب وب سرویس ها با در نظر گرفتن ویژگی های کیفی بر پایه عامل ها به نحوی صورت می گیرد که پویایی و سرعت آن افزایش یابد و با کارایی بیشتری به انجام برسد.

مدل پیشنهادی CAUT-K، روش ارائه شده به وسیله رده عوامل در [۹] را پذیرفته و با استفاده از الگوریتم Top-K توسعه یافته است. در این روش درخواست کاربر به صورتی در نظر گرفته می شود که نیاز به ترکیب سرویس وب جهت ارائه پاسخ داشته باشد. با برآورده نمودن کامل نیازهای کاربر می توان درخواست کاربر را کامل کرد.

تعداد عوامل در این روش به صورت پویا انتخاب می شود و مدیریت سرویس ها با توجه به مفاهیم ورودی و خروجی شان به عوامل داده می شود. سپس با استفاده از پالایش و ساختار موازی Top-K ترکیب سرویس انجام می گیرد.

مجموعه ای از تعداد n سرویس وب را می توان با W

نشان داد و آن را به صورت رابطه ۱ تعریف نمود:

$$W = \{W_i | 1 \leq i \leq n\} \quad (1)$$

به صورت رسمی، سرویس وب به صورت رابطه (۲) تعریف می شود [۹]:

$$W_i = (I_{W_i}, O_{W_i}, Q_{W_i}) \quad (2)$$

که در آن  $W_i$  با سه عنصر  $I_{W_i}$  یا ورودی ها،  $O_{W_i}$  یا خروجی ها و  $Q_{W_i}$  یا کیفیت سرویس مشخص می شود. ورودی ها، اطلاعات (پیش شرط های) مورد نیاز برای استفاده از سرویس و خروجی ها، اطلاعات (اثرات) ایجاد شده ناشی از استفاده سرویس را نشان می دهند. کیفیت سرویس می تواند یک مجموعه n تایی  $\langle q_1, q_2, \dots, q_n \rangle$  باشد که یک خصوصیت QoS از وب سرویس مانند هزینه، زمان پاسخ، توان عملیاتی، در دسترس پذیری و غیره را معنی می دهد.

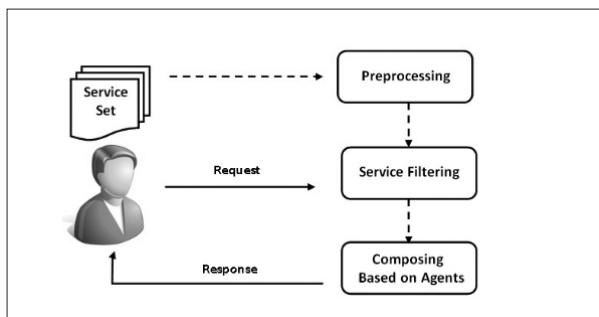
هر پرس و جوی کاربر می تواند به صورت  $R = (I^R, O^R)$  نشان داده شود. ورودی های  $I^R$  اطلاعات فراهم شده توسط کاربر به عنوان ورودی برای سیستم و خروجی های  $O^R$  نتایج مورد نیاز کاربر را تعیین می کند. به عنوان مثال  $I^R$  می تواند یک فایل متنی با قالب «doc» باشد که به عکس تبدیل می شود به این معنی که  $O^R$  یک فایل تصویر با قالب مثلاً «JPG» است. معماری راه حل ارائه شده، همان گونه که در شکل ۲: روش های ترکیب خودکار وب سرویس شکل ۲ نشان داده شده است به سه مرحله پیش پردازش<sup>۳۰</sup>، پالایش سرویس<sup>۳۱</sup> و ترکیب بر مبنای عوامل، تقسیم می گردد.

### ۳-۱- پیش پردازش داده های خام

در مرحله پیش پردازش، مجموعه داده های خام با مقیاس بزرگ، برای تبدیل به یک مخزن قانون پردازش می شوند. به عبارت دیگر می توان گفت سرویس های مجموعه داده به قوانین تبدیل می گردند تا یک مخزن قانون ساخته شود. مخزن قانون یک نوع ساختار داده در حافظه است که می تواند در فرآیند پاسخ دهی به کاربران، به شکل سریع و کارا در دسترس باشد.

30- Preprocessing  
31- Service Filtering

29- Depth First Search



شکل ۲: روش‌های ترکیب خودکار وب سرویس

جدول ۲: نمونه‌ای از مخزن قانون

| Rule           | Web Service     | Input Concept<br>( ) | Output Concept<br>( ) | Response Time (ms) |
|----------------|-----------------|----------------------|-----------------------|--------------------|
| R <sub>1</sub> | WS <sub>1</sub> | A                    | B                     | 40                 |
| R <sub>2</sub> | WS <sub>2</sub> | A                    | B                     | 10                 |
| R <sub>3</sub> | WS <sub>3</sub> | A                    | B                     | 25                 |
| R <sub>4</sub> | WS <sub>4</sub> | A                    | B                     | 30                 |
| R <sub>5</sub> | WS <sub>5</sub> | B                    | D                     | 15                 |
| R <sub>6</sub> | WS <sub>6</sub> | B                    | D                     | 5                  |
| R <sub>7</sub> | WS <sub>7</sub> | D                    | H                     | 10                 |
| R <sub>8</sub> | WS <sub>8</sub> | D                    | H                     | 20                 |
| R <sub>9</sub> | WS <sub>9</sub> | D                    | H                     | 45                 |

جدول ۳: شاخص معکوس مخزن قانونجدول ۲: نمونه‌ای ۱

سرعت یافتن قوانین مربوط به یک مفهوم خروجی بیشتر می‌شود. به‌عنوان مثال زمانی که مفهوم D جستجو می‌شود، می‌توان قوانین R<sub>5</sub>, R<sub>6</sub>, R<sub>10</sub> را به سرعت پیدا نمود.

### ۳-۲- پالایش سرویس

از آن جایی که انتخاب سرویس مناسب از یک مجموعه سرویس با مقیاس بزرگ، به سرعت زمان پردازش را افزایش می‌دهد، بنابراین بهتر است که مقیاس نامزدها کاهش یابد. بدین منظور لازم است که سرویس‌های بی‌فایده<sup>۳</sup> در مراحل آغازین پالایش شوند. زمانی که یک درخواست از کاربر دریافت می‌شود، سرویس‌ها از مخزن قانون واکنشی می‌گردند و سرویس‌هایی پالایش می‌شوند که بر طبق درخواست کاربر و قوانین موجود در مخزن قانون غیرممکن است در نتایج نهایی ارائه گردند.

یک مخزن قانون طبق جدول ۲: نمونه‌ای از مخزن قانون جدول (۲) ساخته می‌شود. به دلیل طولانی بودن جدول قانون فقط بخشی از آن نشان داده شده است. یک قانون به صورت چهارتایی<sup>۳۲</sup>  $r = \langle S, C_i, C_o, QoS \rangle$  تعریف شده است که پارامترهای آن به شرح ذیل می‌باشند:

S سرویسی است که قانون را تولید می‌کند.

C<sub>i</sub> مجموعه مفاهیمی است که قانون به عنوان ورودی

نیاز دارد و شامل ورودی‌های وب سرویس می‌باشد.

C<sub>o</sub> مجموعه مفاهیمی است که قانون به عنوان خروجی

نتیجه می‌دهد و شامل خروجی‌های وب سرویس می‌باشد.

QoS به خصوصیات کیفیت قانون اشاره می‌نماید که

در پیاده‌سازی انجام گرفته در این تحقیق زمان پاسخ به

عنوان پارامتر کیفیت سرویس در نظر گرفته شده است.

مقادیر زمان پاسخ نیز به صورت تصادفی ایجاد می‌گردد.

هر وب سرویس می‌تواند به یک قانون r تبدیل شود که

r.C<sub>i</sub> معادل S.C<sub>1</sub> و r.C<sub>o</sub> ترکیب عطفی S.C<sub>o</sub> و مفاهیم پیشین

S.C<sub>o</sub> می‌باشد. این بدین معنی است که یک قانون می‌تواند

علاوه بر مفاهیم شناخته شده سرویس، پیشینیان آن را نیز

تولید نماید. پس از این که همه سرویس‌ها به قانون تبدیل

شدند، یک مخزن قانون ساخته شده است. مخزن قانون،

داده را در حافظه بیشتر از یک پایگاه داده ذخیره می‌کند

که دسترسی سریع و کارآمد را هنگام ایجاد ترکیب فراهم

می‌نماید.

به منظور ارزیابی دسترسی سریع به مخزن قانون، یک

شاخص معکوس برای آن ساخته می‌شود، قوانین به

وسیله مفاهیمی که می‌توانند خروجی بدهند، شاخص

گذاری شده‌اند[۹].

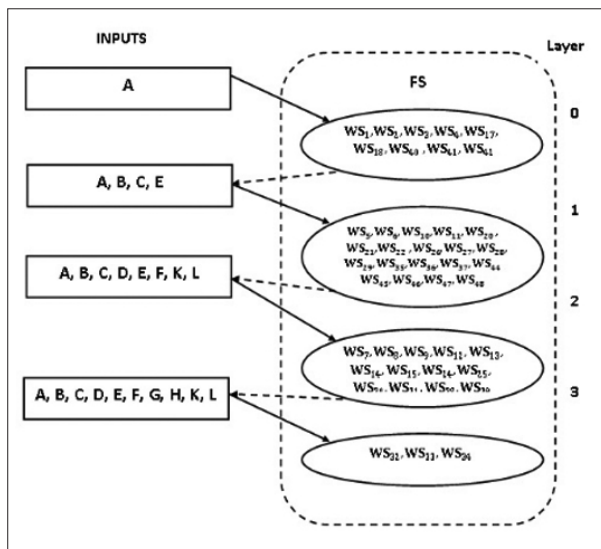
جدول ۳ شاخص معکوس مربوط به مخزن قانون

جدول ۲: نمونه‌ای از مخزن قانونجدول ۲ را نشان می‌دهد.

همان طور که دیده می‌شود یک مفهوم خروجی برحسب

قانون‌های مرتبط با آن نمایش داده می‌شود. بدین ترتیب





شکل ۴: پالایش مخزن قانون جدول ۲: ن

درخواست وجود دارد. در غیر این صورت در خواست کاربر فاقد راه حل می باشد و رویه ترکیب نیازی به اجرا ندارد.

### ۳-۳- ترکیب بر مبنای عوامل

ترکیب بر مبنای عوامل، مرحله اصلی روش CAUT-K در این تحقیق می باشد. همان گونه که بیان گردید پس از این که در مرحله قبل به وسیله عملیات پالایش از تعداد سرویس ها تا حد ممکن کاسته شد، سرویس هایی به دست آمدند که حتماً در ایجاد ترکیب کاربرد دارند. مدیریت مجموعه سرویس های وب که با QoS متفاوت کارکردهای مشابهی دارند یا به عبارت دیگر ورودی و خروجی مشابهی را ارائه می نمایند، بر عهده یک عامل رده<sup>۳۴</sup> می باشد که می توان آن را به صورت رابطه (۳) مشخص نمود:

$$CA_i = (ID_i, CWS_i, I_{CA_i}, O_{CA_i}) \quad (3)$$

که  $ID_i$ ، نام عامل،  $CWS_i$ ، مجموعه سرویس های عامل،  $I_{CA_i}$  ورودی و  $O_{CA_i}$  خروجی مجموعه سرویس های عامل یا همان ورودی و خروجی عامل می باشند. برای ایجاد ترکیب سرویس بر مبنای عامل ها می توان آن ها را در یک شبکه اجتماعی در نظر گرفت. شبکه اجتماعی عامل، گراف وابستگی  $SN = (V, ACE)$  می باشد که در آن رؤس  $V$  رده عوامل و  $ACE$  یال های گراف هستند که هر  $(i, j) \in ACE$

```

//Algorithm Filtering
1) selectedInputSet ← userInputSet
2) selectedServiceSet ← new empty set
3) for each input in selectedInputSet
4)   for each service in allServices
5)     for each output in service.output
6)       if (output contains input)
7)         if (selectedServicesSet.add(service))
8)           selectedInputSet.add(output)
9) return selectedServicesSet

```

شکل ۳: شبه کد عملیات پالایش

در مرحله نهایی، با به کارگیری عوامل، ترکیب های سرویس در پاسخ به پرس و جوی کاربر به صورت موازی تولید می شوند.

جریان فرآیند پالایش به شرح زیر می باشد:

۱- مجموعه INPUT با عناصر اولیه ای شامل ورودی های درخواست کاربر ایجاد می شوند.

۲- همه سرویس هایی که ورودی های آن ها شامل عناصر INPUT می باشند، یافت می شوند. سپس خروجی این سرویس ها به مجموعه INPUT اضافه و سرویس ها نگهداری می گردند.

۳- مرحله ۲ آن قدر تکرار می گردد تا سرویس دیگری پیدا نشود که ورودی های آن عناصر موجود در مجموعه INPUT باشد. به عبارت دیگر این مجموعه قابل گسترش باشند.

بعد از اجرای این الگوریتم سرویس های بی فایده پالایش می شوند. به عبارت دیگر سرویس هایی که نمی توانند به وسیله درخواست کاربر اجرا گردند، پالایش خواهند شد. این مسئله تا حد زیادی تعداد سرویس های نامزد ترکیب را کاهش می دهد. شبه کد عملیات پالایش مطابق با شکل ۳: شبه کد عملیات پالایش شکل (۳) می باشد.

طبق روش پیشنهادی CAUT-K و با توجه به مخزن قانون جدول ۲: نمونه ای از مخزن قانون جدول ۲ چنانچه درخواست کاربر به صورت  $R=(A,H)$  باشد، روند پالایش مربوطه را نشان می دهد.

چنانچه مجموعه INPUT نهایی، شامل مفاهیم وابسته به خروجی درخواست شده کاربر باشد، راه حل هایی برای این

نشان دهنده وجود اتصال اجتماعی بین رده‌های عوامل  $i$  و  $j$  است. دو عامل را در صورتی می‌توان در گراف وابستگی به یکدیگر متصل نمود که خروجی‌هایی از عامل اول وجود داشته باشند که بتوانند با برخی از ورودی‌های عامل دوم تطابق یابند. به عبارت دیگر دو عامل  $CA_1$  و  $CA_2$  با یکدیگر اتصال دارند اگر و تنها اگر خروجی‌هایی از  $CA_1$  وجود داشته باشند که بتواند برخی ورودی‌های  $CA_2$  را تطبیق دهند [۹]:

$$O_{CA_1} \cap I_{CA_2} \neq \emptyset \quad (۴)$$

ترکیب سرویس وب را می‌توان به صورت سه تایی  $\beta = (WS_\beta, P_\beta, QoS_\beta)$  تعریف نمود که در آن  $WS_\beta = \{W_i | 1 \leq i \leq m\} \subseteq W$  زیر مجموعه‌ای از سرویس وب انتخاب شده برای ایجاد ترکیب است و  $P$  ترتیب درخواست  $WS_\beta$  را تعریف می‌کند. دو سرویس وب  $W_1$  و  $W_m$  به ترتیب گره آغازین و پایانی ترکیب را نشان می‌دهد. مقدار  $QoS_\beta$  نیز مجموع  $QoS$  هر سرویس اتمی است:

$$QoS_\beta = \sum_{i=1}^m W_i \quad (۵)$$

بنابراین همان‌طور که پیش‌تر گفته شد ترکیب سرویس وب، کارآمد نامیده می‌شود اگر امکان پذیر و بهینه باشد؛ در این صورت  $QoS_\beta$  بهترین است. به صورت کلی می‌توان گفت که مسئله ترکیب سرویس وب خودکار آگاه از  $QoS$  به صورت سه گانه  $\langle W, R, \beta, \varphi \rangle = QoSAWSP$  تعریف می‌شود:

-  $W$  مجموعه‌ای از سرویس‌های وب است.

-  $R$  نشانه پرس و جوی کاربر است.

-  $\varphi$  تابع، الگوریتم یا روشی است که ترکیب کارآمد  $\beta$

را محاسبه می‌کند [۹].

$$\varphi(W, R): W \times R \rightarrow \beta \quad (۶)$$

گراف وابستگی رده‌های عوامل نمونه در نظر گرفته شده از درخواست کاربر با توجه به مخزن قانون جدول ۲: نمونه‌ای از مخزن قانون جدول ۲ در شکل ۵: گراف وابستگی مخزن قانون جدول ۲:  $n$  شکل ۵ نشان داده شده است. همان‌طور که بیان گردید سرویس‌های به دست آمده با

ورودی و خروجی یکسان که  $QoS$  متفاوتی دارند، توسط یک عامل رده مدیریت می‌شوند بنابراین تعداد عامل‌های مورد استفاده پویا می‌باشند و به تعداد سرویس‌ها بستگی دارند.

عامل‌های رده بر مبنای ورودی و خروجی سرویس‌های خود به یکدیگر پیوند داده و به صورت یک گراف وابستگی در شبکه اجتماعی سازماندهی می‌شوند. به عبارت دیگر در این شبکه اجتماعی، یک عامل رده به عامل رده دیگر در صورتی متصل است که خروجی سرویس‌های تحت مدیریت آن ورودی عامل رده دوم باشد. سپس چندین راه حل بهترین، برای ترکیب سرویس‌ها با استفاده از روش‌های جستجوی گراف به دست می‌آیند.

وجود عوامل همیار که به صورت عامل شبکه اجتماعی در زمان طراحی سازماندهی می‌شوند، از تکرار جلوگیری می‌نمایند. همچنین این عوامل درخواست‌های کاربران زیادی به صورت همزمان را تأمین می‌نمایند.

در طول فرایند درخواست، ممکن است رویدادهایی رخ دهد. این رویدادها می‌تواند شامل موارد زیر باشد:

- ارائه سرویس‌های وب جدید با  $QoS$  بهتر.

- ناپدید شدن و در دسترس نبودن سرویس‌های وب

- هر رویدادی روی مقدار  $QoS$  کلی و امکان‌پذیری

ترکیب تاثیر می‌گذارد. در این حالت، عامل رده باید

سرویس مشابهی را ارائه کند که در رده سرویس بهترین  $QoS$  را دارد.

### ۳-۴- الگوریتم ترکیب

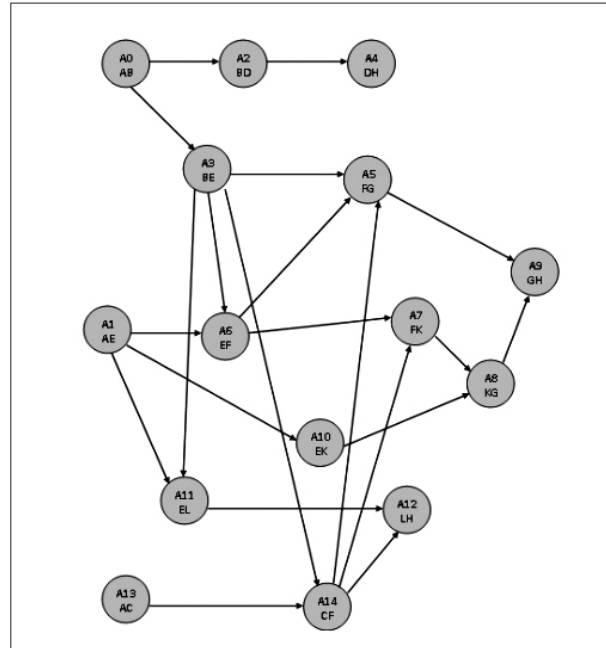
در این بخش الگوریتم مورد استفاده برای ترکیب در روش CAUT-K بیان می‌شود. بدین صورت که ابتدا بازه مورد نیاز برای سرویس‌ها تعیین و سپس ورودی‌ها و خروجی‌ها از کاربر دریافت می‌گردد. در مرحله بعد سیستم به صورت خودکار شروع به پیمایش گراف عامل‌ها می‌نماید و با تطبیق خروجی هر عامل با ورودی عامل بعدی، عامل مورد نظر را انتخاب کرده و سپس سرویسی که دارای بهترین  $QoS$  می‌باشد

جدول ۴: پیکربندی سیستم مورد استفاده در شبیه‌سازی

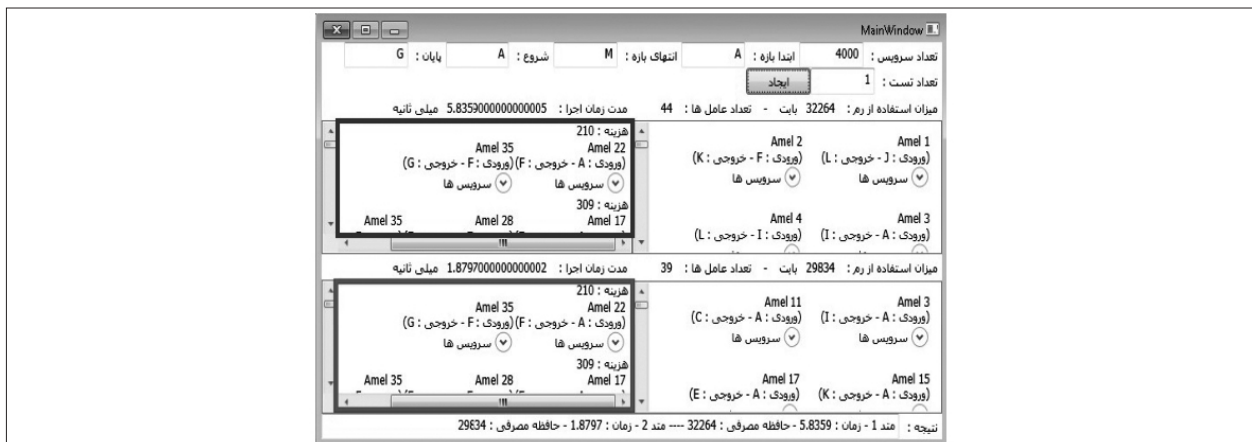
|                |  |
|----------------|--|
| تخته مدار اصلی | Quanta LG4                                   |
| پردازنده       | Intel (R) Core (TM) i5-2430 M CPU @ 2.40 GHz |
| حافظه اصلی     | ۴,۰۰۰ گیگابایت                               |
| سیستم عامل     | ۳۲ بیتی Microsoft Windows 7 Ultimate         |

```
//Algorithm CAUT-K
1) for each root in finalresult
2)   cost := 0
3)   for each Agent in root.Agent
4)     if Agent.Services.is available
5)       cost := cost + Agent.Services.Min (item => item.ResponseTime)
6)   root.cost := cost;
```

شکل ۶: بدنه اصلی الگوریتم ترکیب CAUT-K



شکل ۵: گراف وابستگی مخزن قانونجدول ۲: ن



شکل ۷: نمونه خروجی برنامه از راه حل‌های به دست آمده

و قابل قبول آن جهت شبیه‌سازی وب سرویس‌ها و عوامل است.

#### ۴-۲- مشخصات سیستم

پیکربندی سیستم مورد استفاده در پیاده‌سازی و شبیه‌سازی الگوریتم پیشنهادی CAUT-K مطابق جدول ۴ می‌باشد.

#### ۴-۳- مجموعه داده انتخابی

مجموعه داده سرویس‌های مورد نظر توسط برنامه ایجاد می‌شود. طبق نمونه خروجی نمایش داده شده در شکل ۷ ابتدا

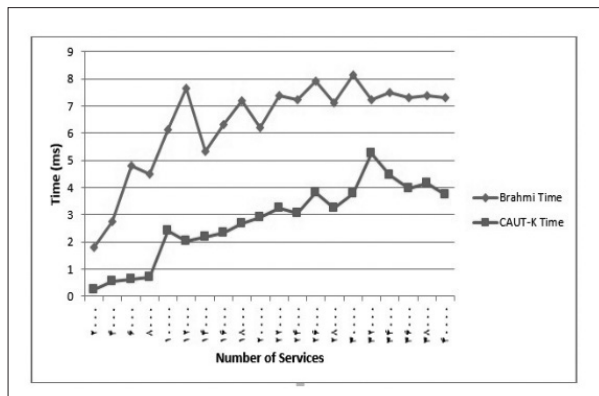
را انتخاب و تا رسیدن به جواب نهایی این چرخه را ادامه می‌دهد.

بدنه اصلی الگوریتم ترکیب CAUT-K در شکل ۶: بدنه اصلی الگوریتم ترکیب CAUT-K شکل ۶ ارائه شده است.

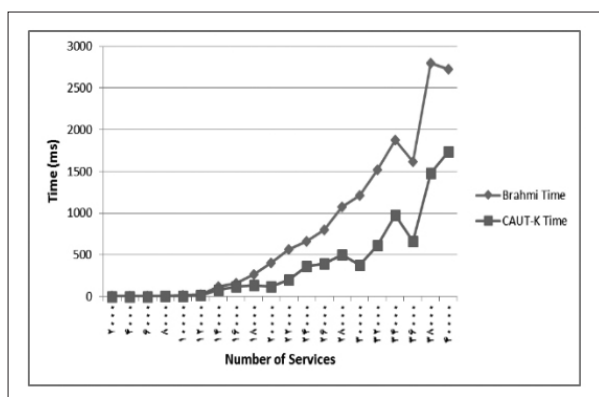
#### ۴-۴- پیاده‌سازی و ارزیابی

##### ۴-۱- ابزار شبیه‌سازی

برای شبیه‌سازی و پیاده‌سازی الگوریتم CAUT-K از نرم‌افزار ویژوال استودیو محصول سال ۲۰۱۳ استفاده شد. یکی از دلایل انتخاب نرم‌افزار مذکور امکانات مختلف



شکل ۸: نمودار مقایسه زمان اجرا در بازه A\_N و ورودی-خروجی A\_G



شکل ۹: نمودار مقایسه زمان اجرا در بازه A\_W و ورودی-خروجی A\_G

جدول ۶: مقادیر مصرف حافظه روش CAUT-K و روش Brahmi با تعداد وب سرویس‌های یکسان

| Interval = A-N<br>I/O = A-G |           |        |            |
|-----------------------------|-----------|--------|------------|
| NO.Agent                    | Memory(B) |        | NO.Service |
|                             | Brahmi    | CAUT-K |            |
| 15                          | 16,167    | 9,935  | 2000       |
| 33                          | 32,271    | 27,189 | 4000       |
| 42                          | 48,319    | 40,814 | 6000       |
| 44                          | 64373.2   | 47,465 | 8000       |
| 60                          | 80,391    | 73,320 | 10000      |

همان‌طور که در شکل ۸: نمودار مقایسه زمان اجرا در بازه A\_N و ورودی-خروجی A\_G شکل ۸ و شکل ۹: نمودار مقایسه زمان اجرا در بازه A\_W و ورودی-خروجی A\_G شکل ۹ مشاهده می‌شود در داده‌هایی با بازه‌های متفاوت تفاوت زمان اجرا محسوس‌تر می‌باشد. پالایش روش

جدول ۵: مقادیر زمان اجرای روش CAUT-K و روش Brahmi با تعداد وب سرویس‌های یکسان

| Interval = A-N<br>I/O = A-G |          |        |            |
|-----------------------------|----------|--------|------------|
| NO.Agent                    | Time(ms) |        | NO.Service |
|                             | Brahmi   | CAUT-K |            |
| 15                          | 1.8021   | 0.2581 | 2000       |
| 33                          | 2.751    | 0.5281 | 4000       |
| 42                          | 4.7977   | 0.6329 | 6000       |
| 44                          | 4.4869   | 0.6811 | 8000       |
| 60                          | 6.113    | 2.4028 | 10000      |

بر اساس بازه دریافتی از کاربر، سرویس‌ها تولید می‌گردند و محدوده ورودی و خروجی درخواست کاربر معین می‌شود، سپس برحسب سرویس‌های ایجاد شده و ورودی و خروجی آن‌ها، تعداد عامل‌های مورد استفاده مشخص می‌گردد. همچنین میزان مصرف حافظه بر حسب بایت و زمان اجرا بر حسب میلی ثانیه محاسبه می‌شود.

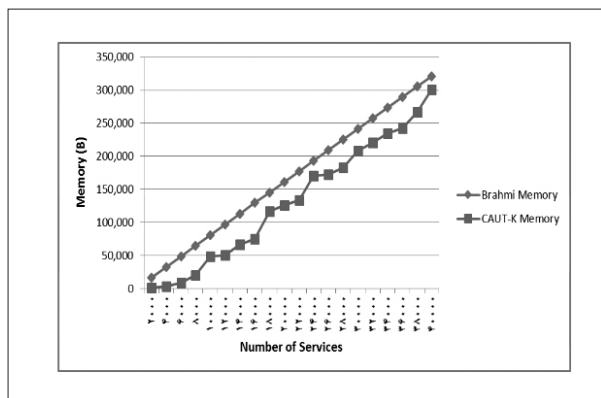
#### ۴-۴-۲- بررسی نتایج شبیه‌سازی مدل پیشنهادی

برای تحلیل و ارزیابی روش CAUT-K ابتدا پیاده‌سازی رویکرد ارائه شده توسط Brahmi [۹] بررسی گردید و سپس نتایج مورد نظر با داده‌های یکسان بر روی الگوریتم‌های ارائه شده توسط Brahmi و الگوریتم پیشنهادی CAUT-K تحلیل شد.

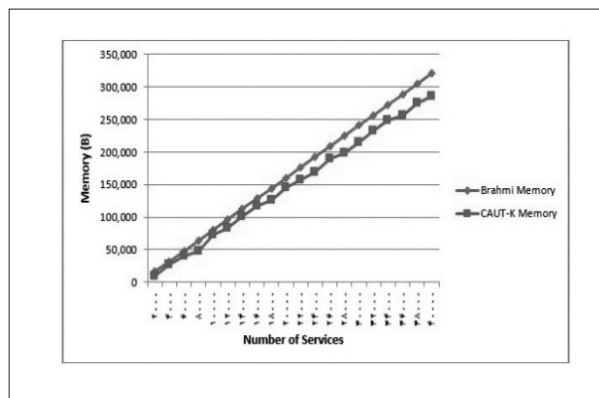
نتایج کاربرد الگوریتم پیشنهادی CAUT-K در مسئله یافتن چندین ترکیب بهینه از وب سرویس‌ها بدین شرح می‌باشند:

۴-۲-۴- زمان اجرا

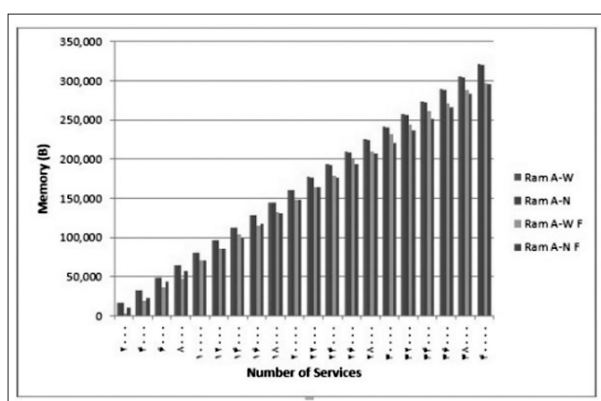
در تحلیل نتایج به دست آمده، مشاهده می‌شود که با افزودن تعداد وب سرویس‌های مجموعه داده، زمان اجرا افزایش می‌یابد. جدول ۵: مقادیر زمان اجرای روش CAUT-K و روش Brahmi با تعداد وب سرویس‌های یکسان ۵ مقادیر به دست آمده از زمان اجرای الگوریتم CAUT-K و الگوریتم Brahmi [۹] را در تعداد سرویس‌های وب یکسان نمایش داده است.



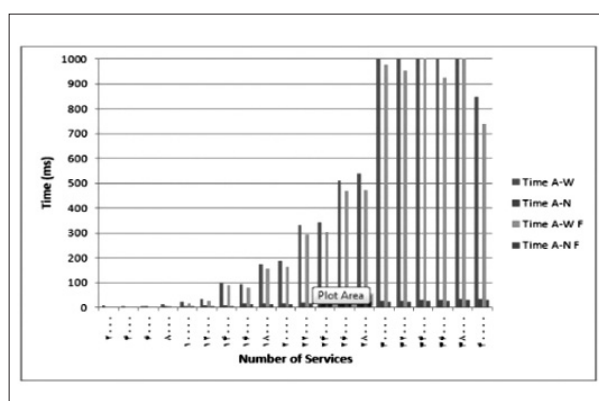
شکل ۱۱: نمودار مقایسه میزان مصرف حافظه در بازه A\_W و ورودی-خروجی A\_G



شکل ۱۰: نمودار مقایسه میزان مصرف حافظه در بازه A\_N و ورودی-خروجی A\_G



شکل ۱۳: نمودار مقایسه میزان مصرف حافظه در بازه‌های A\_W و A\_N با ورودی-خروجی B\_M (دو روش CAUT-K و Brahma)



شکل ۱۲: نمودار مقایسه زمان اجرا در بازه‌های A\_W و A\_N با ورودی-خروجی B\_M (دو روش CAUT-K و Brahma)

۱۰: نمودار مقایسه میزان مصرف حافظه در بازه A\_N و ورودی-خروجی A\_G شکل ۱۰ و شکل ۱۱: نمودار مقایسه میزان مصرف حافظه در بازه A\_W و ورودی-خروجی A\_G شکل ۱۱ ارائه گردیده است.

توجه به این موضوع که در مدل CAUT-K تعداد عواملها به صورت پویا انتخاب می‌شود و بر اساس ورودی و خروجی سرویس‌های نامزد، مدیریت آنها به عوامل محول می‌گردد و همچنین نظر به عملیات پالایش و ساختار موازی الگوریتم Top-K می‌توان نتیجه گرفت که حافظه مصرفی در مدل پیشنهادی CAUT-K کمتر می‌باشد.

همان‌طور که نتایج حاصل از شبیه‌سازی نشان می‌دهد الگوریتم CAUT-K علاوه بر کلیه مزایای ایجاد ترکیب مبتنی بر عوامل و استفاده از الگوریتم موازی Top-K، در

پیشنهادی قبل از برقراری مدیریت سرویس‌های نامزد به وسیله عاملها، سبب گردیده است که زمان صرف شده برای اجرای الگوریتم پیشنهاد شده کمتر از روش [۹] باشد که تنها از عوامل جهت ایجاد ترکیب استفاده نموده است.

#### ۲-۴-۴- مصرف حافظه

مورد قابل بررسی دیگری که در نتایج به دست آمده قابل تحلیل می‌باشد، میزان مصرف حافظه توسط الگوریتم پیشنهادی CAUT-K است. جدول ۶: مقادیر مصرف حافظه روش CAUT-K و روش Brahma با تعداد وب سرویس‌های یکسان ۶ مقادیر میزان مصرف حافظه الگوریتم CAUT-K و الگوریتم Brahma [۹] را در تعداد وب سرویس‌های یکسان نشان می‌دهد.

مقایسه اطلاعات میزان مصرف حافظه روش CAUT-K و روش Brahma برحسب تعداد وب سرویس‌ها در شکل

کیفیت سرویس چندگانه از جمله ترکیب پذیری، در دسترس پذیری و غیره جهت ساخت ترکیب پرداخت. همچنین به مدیریت سرویس‌هایی با ورودی و خروجی چندگانه که تشابهی بین ورودی و خروجی آن‌ها وجود دارد توسط عوامل همیار توجه نمود. قراردادن نتایج ترکیب در حافظهٔ نهان<sup>۲۵</sup> و استفاده مجدد از آن‌ها، بررسی استفاده از عوامل در عملیات پالایش و به‌کارگیری روش CAUT-K در مجموعه داده‌های ناسازگار از موارد دیگری می‌باشند که می‌توان در تحقیقات آینده لحاظ نمود.

#### مراجع

- ۱- سپهری فر، م. ک.، زمانی فر، ک. و مردوخی، م.، «الگوریتمی برای یافتن ترکیب بهینه از سرویس‌ها، اولین کنفرانس مهندسی نرم افزار، ۱۳۸۸
- 2- Brahmi, Z., & Gammoudi, M. M., "QoS-Aware Automatic Web Service Composition based on cooperative agents", 22nd International Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE), pp. 27-32, 2013.
- 3- Deng, S., Huang, L., Tan, W., & Wu, Z., "Top-k Automatic Service Composition: A Parallel Method for Large-Scale Service Sets", IEEE Transactions on Automation Science and Engineering, Vol. 11, No. 3, pp. 891-905, 2014.
- 4- Sirin, E., Parsia, B., Wu, D., Hendler, J., & Nau, D., "HTN planning for web service composition using SHOP2", Web Semantics: Science, Services and Agents on the World Wide Web, Vol. 1, No. 4, pp. 377-396, 2004.
- 5- Zhovtobryukh, D., "A petri net-based approach for automated goal-driven web service composition", Transaction of The Society for Modeling and Simulation International, Vol. 83, No. 1, pp. 33-63, 2007.
- 6- Falou, M. E., Bouzid, M., Mouaddib, A. I., & Vidal, T., "Automated web service composition using extended representation of planning domain", International Conference on Web Services Proceedings, IEEE, 2008.
- 7-Goncalves da Silva, E. M., Ferreira Pires, L., & van Sinderen, M. J., "An algorithm for automatic service composition", 1st International Workshop on Architectures, Concepts and Technologies for service Oriented Computing Proceedings, pp. 65-74, 2007.
- 8- Yue, K., Yue, M., Liu, W., & Li, X., "A graph-based approach for type matching in web service composition", Journal of Computational Information Systems, Vol. 6, No. 7, pp. 2141-2149, 2010.
- 9- Rodriguez-Mier, P., Mucientes, M., & Lama, M., "Automatic web service composition with a heuristic-based search algorithm", Proceedings of IEEE International Conference on 35- cache

تمام موارد زمان اجرا و مصرف حافظه بهتری نسبت به روش ارائه شده در الگوریتم معتبر Brahmi [۹] دارد شکل ۱۲: نمودار مقایسه زمان اجرا در بازه‌های A\_W و A\_N با ورودی-خروجی B\_M (دو روش CAUT-K و Brahmi) ۱۲ شکل ۱۳: نمودار مقایسه مصرف حافظه در بازه‌های A\_N و A\_W با ورودی-خروجی B\_M (دو روش CAUT-K و Brahmi) و شکل ۱۳ نمودارهای زمان اجرا و مصرف حافظه را در دو روش CAUT-K و Brahmi [۹] با بازه‌های زمانی متفاوت و ورودی و خروجی یکسان نشان می‌دهد. همان طور که دیده می‌شود برای بررسی جامع‌تر، زمان اجرای بازه‌های سرویس متفاوت در کنار یکدیگر قرار گرفته‌اند.

#### ۵- نتیجه گیری

اکثر روش‌های موجود برای ترکیب وب سرویس‌ها بر پایه عوامل، تنها یک راه حل را بر می‌گرداند که در برخی شرایط معین، بهینه می‌باشد و از نظر انعطاف پذیری و تنوع در سطح پایینی قرار دارد. در این تحقیق روشی ارائه شده است که با در نظر گرفتن معیار کیفیت سرویس و استفاده از چارچوب موازی الگوریتم نسبت به ایجاد چندین راه حل بر پایه عوامل اقدام می‌نماید. الگوریتم ارائه شده با استفاده از عملیات پالایش قبل از ایجاد ترکیب برای داده‌های با مقیاس بزرگ کارایی مطلوبی دارد. در مدل پیشنهادی سرویس‌هایی که در ایجاد ترکیب مورد نظر سهمی ندارند، توسط عملیات پالایش حذف گردیدند، پس از آن سرویس‌های باقیمانده به عنوان نامزد برای مدیریت به عوامل داده شد.

نتایج حاصل از شبیه‌سازی نشان می‌دهد الگوریتم CAUT-K علاوه بر کلیه مزایای ایجاد ترکیب مبتنی بر عوامل و استفاده از الگوریتم موازی Top-K، در تمام موارد بهبود قابل توجهی در زمان اجرا و مصرف حافظه دارد. در پژوهش‌های آینده می‌توان به بررسی دقیق‌تر خصوصیات کیفیت سرویس و لحاظ نمودن ویژگی‌های



dynamic integration environment”, *The International Journal of Advanced Manufacturing Technology*, Vol. 35, No. 1-2, pp. 15-25, 2007.

24-Shafiq, M. O., Ding, Y., & Fensel, D., “Bridging multi agent systems and web services: towards interoperability between software agents and semantic web services”, *Enterprise Distributed Object Computing Conference, EDOC’06, 10th IEEE International*, pp. 85-96, 2006.

25-Wang, Y., Zhang, J., & Vassileva, J., “Effective web service selection via communities formed by super-agents”, *IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology (WI-IAT)*, Vol. 1., pp. 549-556, 2010.

26-El Ouahed, A. K., Erradi, M., & Azzoune, H., “A Discovery Service for Automatic Composition of Web Services Oriented-Agent”, *22nd International Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE)*, pp. 33-35, 2013.

27-Lotf Mohamadi, S., & Emadi, S., “A Survey on Web Services Composition using Top-k Algorithm”, *3rd International Conference on Applied Research in Computer & Information*, 2016.

28-Deng, S., Wu, B., Yin, J., & Wu, Z., “Efficient planning for Top-k Web service composition”, *Knowledge and information systems*, Vol. 36, No. 3, pp. 579-605, 2013.

*Web Services*, 2011.

10- Kwon, J., & Lee, D., “Non-redundant web services composition based on a two-phase algorithm”, *Data & Knowledge Engineering*, Vol. 71, No .1, pp. 69-91, 2012.

11- Cao, L., Li, M., & Cao, J., “Using genetic algorithm to implement cost-driven web service selection”, *Multiagent and Grid Systems*, Vol. 3, No .1, pp. 9–17, 2007.

12- Canfora, G., Di Penta, M., Esposito, R., & Villani, M. L., “An approach for QoS-aware service composition based on genetic algorithms”, *Proceedings of the 7th annual conference on Genetic and evolutionary computation (GECCO’05)*, pp. 1069–1075, 2005.

13-Zeng, L., Benatallah, B., Ngu, A. H., Dumas, M., Kalagnanam, J., & Chang, H., “Qos-aware middleware for web services composition”, *IEEE Transactions on Software Engineering*, Vol. 30, No .5, pp. 311-327, 2004.

14-Vadivelou, G., Ilavarasan, E., & Prasanna, S., “Algorithm for web service composition using multi-agents.” *International Journal of Computer Applications*, 13.8, pp.40–45, 2011.

15- Tong, H., Cao, J., Zhang, S., & Li, M., “A distributed algorithm for web service composition based on service agent model”, *IEEE Transactions on Parallel and Distributed Systems*, Vol. 22, No .12, pp.2008-2021, 2011.

16-Kowalczyk, R., & Braun, P., “Towards agent-based coalition formation for service composition”, *Proceedings of the IEEE/WIC/ACM international conference on Intelligent Agent Technology*, IEEE Computer Society, pp. 73-80, 2006.

17-Alrifai, M., Risse, T., Dolog, P., & Nejdl, W., “A scalable approach for QoS-based web service selection”, *Service-Oriented Computing–ICSOC 2008 Workshops*, Springer Berlin Heidelberg, pp. 190-199, 2009.

18-Wang, X. L., Huang, S., & Zhou, A. Y., “QoS-aware composite services retrieval”, *Journal of Computer Science and Technology*, Vol. 21, No. 4, pp. 547-558, 2006.

19- Jiang, W., Wu, T., Hu, S. L., & Liu, Z. Y., “Qos-aware automatic service composition: A graph view”, *Journal of computer science and technology*, Vol. 26, No. 5, pp. 837-853, 2011.

20- Liu, Z. Z., Xue, X., Shen, J. Q., & Li, W. R., “Web service dynamic composition based on decomposition of global QoS constraints”, *The International Journal of Advanced Manufacturing Technology*, Vol. 69, No. 9-12, , pp. 2247-2260, 2013.

21- Haak, S., & Weinhardt, C., “Optimizing customized services: efficient computation in large Service Value Networks”, *Information Systems and e-Business Management*, Vol. 12, No. 3, pp. 307-335, 2014.

22- Yu, T., & Lin, K. J., “Service selection algorithms for Web services with end-to-end QoS constraints”, *Information Systems and e-Business Management*, Vol. 3, No. 2, pp. 103-126, 2005.

23-Qiu, H. B., Shao, X. Y., Li, P. G., & Gao, L., “An agent- and service-based collaborative design architecture under a