

ترکیب خودکار سرویس‌ها مبتنی بر گراف رنگ‌آمیزی

سپیده شیوندی^۱؛ سیما عمادی^۲

۱- گروه مهندسی کامپیوتر - واحد یزد - دانشگاه آزاد اسلامی - یزد - ایران - ms.sheivandi@iauyazd.ac.ir

۲- گروه مهندسی کامپیوتر - واحد یزد - دانشگاه آزاد اسلامی - یزد - ایران - emadi@iauyazd.ac.ir

چکیده: سرویس‌های وب به‌عنوان مؤلفه‌های نرم‌افزاری مستقل توسط ارائه‌دهندگان سرویس بر روی اینترنت منتشر شده و توسط درخواست‌کنندگان سرویس برای رسیدن به اهداف مختلف فراخوانی می‌شوند. با این حال در بسیاری از موارد هیچ سرویسی به‌تنهایی در مخزن سرویس یافت نمی‌شود که بتواند رضایت درخواست‌کننده را برآورده سازد. ترکیب سرویس، مؤلفه‌های جدیدی را با استفاده از یک مدل تعاملی برای سرعت بخشیدن به برنامه‌ها ایجاد می‌کند. قبل از ترکیب سرویس‌ها با یکدیگر، مهم‌ترین مسئله برای پیدا کردن نمونه سرویس‌های کاندید مطلوب، مطابقت آن سرویس‌ها با نیازمندی‌های غیروظیفه‌مندی است؛ لذا چگونگی طراحی یک روش کارآمد جهت ترکیب زنجیره‌ای از سرویس‌های متصل به هم مهم است. به‌تازگی تحقیقات زیادی جهت کاهش زمان جستجو برای پیدا کردن یک ترکیب سرویس انجام شده است. با این حال بسیاری از این روش‌ها برای پیمایش و بررسی همه وب‌سرویس‌های موجود در یک مخزن وب به مدت‌زمان طولانی نیاز دارند که به‌صورت قابل توجهی وقت کاربر را اشغال می‌کند. این مقاله رویکردی برای ترکیب خودکار سرویس‌های آگاه از کیفیت سرویس و همچنین استفاده از سلاقی کاربران در رسیدن به نتیجه ترکیب بهینه ارائه می‌دهد. بدین منظور از روش پیشنهادی گراف رنگ‌آمیزی اصلاح شده برای فیلتر نمودن سرویس‌ها قبل از ایجاد ترکیب در داده‌هایی با مقیاس بزرگ استفاده می‌شود که مجموعه سرویس‌های انتخابی را کاهش می‌دهد. همچنین استفاده از الگوریتم KPL در این پژوهش باعث گردیده است تا چندین راه‌حل مناسب به کاربر ارائه شود تا در مواقع لزوم از قابلیت جایگزینی این راه‌حل‌ها به‌جای بهترین ترکیب استفاده نماید. نتایج حاصل از تحلیل و ارزیابی روش پیشنهادی، بهبود مطلوبی را در زمان اجرا و مصرف حافظه نمایان می‌سازد.

واژه‌های کلیدی: ترکیب سرویس، گراف رنگ‌آمیزی، الگوریتم Top-K، آگاه از کیفیت سرویس، الگوریتم KPL

Automatic Service Composition Based on Graph Coloring

S. Sheivandi¹; S. Emadi²

1- Department of Computer Engineering, Yazd Branch, Islamic Azad, Yazd, Iran, Email: ms.sheivandi@iauyazd.ac.ir

2- Department of Computer Engineering, Yazd Branch, Islamic Azad, Yazd, Iran, Email: emadi@iauyazd.ac.ir

Abstract: Web services as independent software components are published on the Internet by service providers and services are then called by users' request. However, in many cases, no service alone can be found in the service repository that could satisfy the applicant satisfaction. Service composition provides new components by using an interactive model to accelerate the programs. Prior to service composition, the most important issue in finding suitable candidate services samples is their compliance with non-functional requirements. Thus, designing an efficient way to combine a chain of connected services is important. Recently, numerous studies have been done to reduce the search time in finding a service composition. However, many of these methods to examine and investigate all Web services in a Web repository require a long time, which occupy the user's time significantly. This paper provides an approach for automatic quality-aware service composition as well as the users' preferences in achieving the optimum composition results. For this purpose, modified graph coloring method to filter the data before compositions in large-scale data is used which decreases selected services set. The application of KPL algorithm in this study provided some proper solutions to the user so that these solutions can be used instead of the best composition if necessary. Therefore, the results derived from the analysis of the proposed method, indicates a good optimization in runtime and memory consumption.

Keywords: Service Composition, Graph Coloring, Top-K algorithm, Quality-aware Service, KPL Algorithm.

تاریخ ارسال مقاله: ۱۳۹۵/۱۰/۲۲

تاریخ اصلاح مقاله: ۱۳۹۶/۰۴/۱۵، ۱۳۹۶/۰۱/۱۶ و ۱۳۹۶/۰۸/۱۴

تاریخ پذیرش مقاله: ۱۳۹۶/۱۰/۳۰

نام نویسنده مسئول: سیما عمادی

نشانی نویسنده مسئول: گروه مهندسی کامپیوتر - واحد یزد - دانشگاه آزاد اسلامی - یزد - ایران

۱- مقدمه

امروزه با افزایش تعداد وب‌سرویس‌ها و کاربران آن‌ها [۱]، انتخاب و ترکیب دستی سرویس‌های مورد نیاز با توجه به درخواست‌های متفاوت کاربران غیرممکن است. بنابراین، ترکیب خودکار سرویس^۱ به منظور برآورده نمودن درخواست‌های کاربران ضروری است. این مسئله توجه بسیاری را در زمینه تحقیقاتی و جوامع صنعتی به خود جلب کرده است [۲-۴].

وب‌سرویس‌ها، برنامه‌های کاربردی مستقل از زمان و بستر هستند که از طریق وب قابل فراخوانی و اجرا می‌باشند [۵]. ساختار وب‌سرویس برای پیاده‌سازی تعاملات ساده بین مشتری و یک وب‌سرویس کافی است، اما با پیچیده‌تر شدن روزافزون درخواست‌ها، وب‌سرویس‌های با پیچیدگی بیشتر نیاز است. ترکیب وب‌سرویس به فرآیند الحاق چندین وب‌سرویس با یکدیگر برای ایجاد وب‌سرویس با ارزش افزوده، اطلاق می‌شود [۵]. فرآیند ترکیب، با دریافت درخواست کاربر برای استفاده از یک سرویس مرکب آغاز می‌شود و پس از بررسی این درخواست، سرویس مورد نظر از میان سرویس‌های موجود انتخاب و سپس توصیفی از این سرویس مرکب توسط یک زبان مناسب ارائه می‌گردد. در نهایت موتور اجرای سرویس مرکب این توصیف را اجرا می‌کند [۶]. به عبارت دیگر فرآیند پیدا کردن ترکیب مناسب بر اساس درخواست کاربر دارای سه مرحله ایجاد مدل کیفیت سرویس مناسب، کشف و انتخاب وب‌سرویس‌های مناسب و ایجاد ترکیب مناسب بر اساس وب‌سرویس‌های انتخاب شده است [۶].

رایج‌ترین روش‌های ترکیب سرویس به دو دسته کلی متدهای دقیق یا همان غیرابتکاری و متدهای تقریبی یا ابتکاری تقسیم می‌شوند [۷]. در روش‌های جستجوی گراف که جز متدهای دقیق است، ابتدا یک گراف وابستگی^۲ بر اساس درخواست کاربر ایجاد و با استفاده از الگوریتم جستجو، بهترین راه‌حل پیدا می‌شود. علیرغم شباهت‌های موجود، این روش‌ها در نحوه جستجو، کارایی و کیفیت راه‌حل به دست آمده تفاوت دارند. الگوریتم‌های ابتکاری روش‌هایی هستند که بر پایه الهام گرفتن از پدیده‌های طبیعی، زیستی یا اجتماعی کار خود را انجام می‌دهند. مزایایی چون مقیاس‌پذیری، تطابق‌پذیری، قابلیت اطمینان و زمان اجرای قابل قبول باعث شده تا انواع مختلفی از این الگوریتم‌ها از جمله الگوریتم ژنتیک، کلونی مورچگان، تجمع ذرات یا همان جمعیت پرندگان، رقابت استعماری و غیره را تبدیل به روش‌هایی نیرومند در حل مسائل بهینه‌سازی و به طبع مسئله بهینه‌سازی ترکیب وب‌سرویس‌ها کند [۷].

از طرفی در اغلب روش‌های ترکیب سرویس، اکثر مواقع برای تمایز قائل شدن بین وب‌سرویس‌هایی که وظیفه‌مندی یکسانی دارند از کیفیت سرویس^۳ استفاده می‌شود. نظر به پویایی بالا و رشد سریع در تعداد وب‌سرویس‌هایی 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. با توجه به گستردگی موضوع، روش‌های متعددی برای ترکیب خودکار

سرویس‌ها مطرح شده‌اند که برخی از آن‌ها در این تحقیق بررسی شده است:

Brahmi و همکارانش به منظور ترکیب خودکار سرویس از عامل‌های همکار استفاده نمودند و با ایجاد گراف وابستگی به عامل‌ها قابلیت خودسازماندهی داده شد. این روش با افزایش تعداد سرویس‌ها و انفجار فضای حالت، کارایی لازم را نخواهد داشت، همچنین در زمانی که وب‌سرویس نامعتبر باشد، ترکیب دیگری را نمی‌توان به عنوان جایگزین انتخاب نمود [۸]. Deng و همکارانش روشی نوین برای مسئله ترکیب خودکار آگاه از کیفیت سرویس وب‌سرویس‌های معنایی با استفاده از Top-K ارائه نمودند. دو مسئله مقیاس‌پذیری و ارائه چندین روش به کاربر به جای ارائه یک مورد باعث گردید تا این تیم یک روش موازی برای ارائه راه‌حل‌های مختلف در مجموعه سرویس‌ها با مقیاس بزرگ را تشریح نمایند [۹، ۱۰]. Zou و همکاران رویکردی برای ترکیب پویای سرویس وب با استفاده از طراحی کارآمد مخازن سرویس با مقیاس بزرگ ارائه نموده‌اند. در رویکرد مطرح شده از برنامه‌ریزی و طراحی بر پایه هوش مصنوعی جهت کاهش زمان جستجو برای پیدا کردن یک ترکیب سرویس استفاده نموده است و مخزن وب‌سرویس را فقط یک‌بار به دامنه طراحی PDDL^۴ تبدیل می‌کند. این دامنه تنها موقعی به روزرسانی می‌گردد که تغییری در مخزن سرویس وب ایجاد شود تا منجر به کاهش قابل ملاحظه‌ای در زمان پاسخ و بهبود مقیاس‌پذیری در ترکیب وب‌سرویس‌ها شود [۱۱].

Silva و همکاران روشی برای بهینه‌سازی ازدحام ذرات مبتنی بر گراف، جهت انتخاب و ترکیب وب‌سرویس‌های آگاه از کیفیت سرویس ارائه کرده‌اند. این الگوریتم که از پارامترهای کیفی استفاده کرده است، غیرفازی و پویا بوده و از استراتژی سراسری برای حل مسئله ترکیب وب‌سرویس‌ها استفاده کرده است. نتایج مقایسه بین دو روش الگوریتم حریصانه و الگوریتم ازدحام ذرات مبتنی بر گراف نشان می‌دهد با وجود نیاز به زمان اجرای بیشتر برای الگوریتم ازدحام ذرات، راه‌حل‌های سازگارتری نسبت به روش حریصانه ارائه می‌دهد [۱۲]. در تحقیق دیگری، Silva و همکارانش روش برنامه‌نویسی ژنتیک را برای انتخاب و ترکیب وب‌سرویس آگاه از QoS ارائه کرده‌اند. نتایج آزمایش نشان می‌دهد که ترکیبات تولید شده با کیفیت در روش دوم بهتر از ترکیبات تولید شده توسط روش اول است. در مقایسه با الگوریتم PSO، روش دوم مقدار برازندگی پایین‌تری نسبت به الگوریتم PSO مبتنی بر گراف دارد و همچنین زمان اجرای آن کمتر است [۱۳].

Liao و همکاران نیز از الگوریتم ازدحام ذرات اصلاح شده، برای ترکیب سرویس استفاده نمودند. آن‌ها برای رفع مشکل همگرایی زود هنگام از روش‌های Niching موازی و سریال با عنوان ASPSO^۵ استفاده نمودند [۱۴]. Chen و Yu از الگوریتم کلونی مورچگان برای ترکیب سرویس در محاسبات ابری استفاده نمودند. در این تحقیق یک الگوریتم حریصانه با نام Greedy-WSC و یک الگوریتم بر اساس بهینه‌سازی کلونی مورچگان که ACO-WSC نامیده می‌شود، ارائه شده

یک رابط بین سرویس ترکیبی و سرویس‌های والد خارجی ساخته شده است. روش پیشنهادی علیرغم کاستی‌هایی، به کاربران در ترکیب سرویس‌های عاری از خطا، اطمینان بیشتری می‌دهد [۲۴].

با توجه به بررسی‌های انجام شده، استنباط می‌شود که پژوهش‌های موجود در زمینه یافتن نتیجه بهینه از ترکیب سرویس‌ها، مشکلاتی دارد. ارائه‌دهندگان سرویس باید گزینه‌ای را که موجب بروز اشکال و تبعات آن مشکلات می‌شود را بهبود بخشند. رایج‌ترین این مشکلات، عدم پویایی ترکیب، در نظر نگرفتن دقت و سرعت و عدم وجود حق انتخاب برای کاربران و عدم قابلیت اطمینان در هنگام شکست یک سرویس است. این موضوع می‌تواند محدودیت‌های بسیاری برای کاربران و ارائه‌دهندگان سرویس به ارمغان بیاورد.

در برخی از روش‌ها برای افزایش مقیاس‌پذیری و ترکیب پویای سرویس‌ها از الگوریتم Top-K به صورت انفرادی یا ترکیب با الگوریتم‌های دیگر استفاده نموده‌اند [۹، ۲۵]. هدف استفاده از الگوریتم Top-K، ترکیب پویا و رفع مشکل مقیاس‌پذیری بوده است [۹، ۲۵]. همچنین برای در نظر گرفتن حق انتخاب برای کاربر در رسیدن به ترکیب مناسب از الگوریتم KPL^{۱۱} استفاده شده است [۲۵]. اما از آنجایی که الگوریتم Top-K خود دارای مشکلاتی از قبیل هزینه‌بر بودن و زمان اجرای بالا می‌باشد، در این پژوهش دو روش پیشنهادی بر اساس الگوریتم گراف رنگ‌آمیزی اصلاح شده^{۱۲} با عنوان MGC-TopK و MGC-K ارائه گردید. در روش‌های پیشنهادی، هدف کاهش زمان اجرا و حافظه، با فیلتر نمودن سرویس‌های ناکارآمد برحسب کیفیت سرویس است. با این عمل از تعداد سرویس‌هایی که به عنوان رأس‌های این گراف شناخته می‌شوند کاسته شده و باعث می‌گردد عملیات انتخاب و ترکیب سرویس‌ها با سرعت بیشتر و مصرف حافظه کمتر انجام پذیرد.

در ادامه مقاله، در بخش ۲ با عنوان تعاریف و مفاهیم بنیادی به تشریح مفاهیمی مانند کیفیت سرویس، الگوریتم Top-K و الگوریتم Sim-Dijkstra پرداخته می‌شود. در بخش ۳، روش پیشنهادی (MGC-K و MGC-TopK) به طور کامل تشریح می‌شود. ارزیابی و نتایج راهکار پیشنهادی در بخش ۴ نشان داده شده است. در نهایت، در بخش ۵ نتیجه‌گیری ارائه شده است.

۲- تعاریف و مفاهیم بنیادی

در این بخش ابتدا به تعاریف مورد نیاز در حوزه سرویس پرداخته می‌شود. سپس مفاهیم مربوط به الگوریتم‌های مورد نیاز برای ترکیب سرویس که در روش پیشنهادی استفاده شده، شرح داده می‌شود.

۲-۱- کیفیت سرویس

در ترکیب سرویس وقتی سرویس‌های واقعی انتخاب می‌شوند، نیازمندی‌های وظیفه‌مندی و غیروظیفه‌مندی ملاحظه می‌گردند، هنگامی که بسیاری از سرویس‌ها با وظیفه‌مندی معادل در دسترس هستند، ویژگی‌های کیفیت سرویس (همچون تأخیر، قیمت،

هدف این روش انتخاب مؤثر ترکیبات ابرها با تعداد حداقل بود [۱۵]. Wang و همکاران یک رویکرد مبتنی بر الگوریتم ژنتیک جهت ترکیب وب‌سرویس‌ها در محیط ابر توزیع شده ارائه کردند. این گروه یک مدل ترکیبی که سرویس‌های کیفیت سرویس و محیط شبکه ابری را در نظر گرفته است ارائه می‌دهند. نتایج تجربی نشان می‌دهد که استفاده از الگوریتم ژنتیک در محاسبات ابری جهت ترکیب وب‌سرویس‌ها باعث کاهش زمان و هزینه اجرا می‌گردد [۱۶].

Li و همکارانش یک روش انتخاب بهینه جهانی (GOS)^{۱۳} برای ترکیب وب‌سرویس‌های آگاه از QoS بر اساس مکانیسم پیش‌بینی مقادیر QoS از سرویس‌های محلی پیشنهاد کرده‌اند. نتایج شبیه‌سازی نشان می‌دهد که GOS انتخاب بسیار عالی می‌باشد و نسبت به سایر روش‌های ترکیب، بهره‌وری بهتری دارد و هزینه اجرا نیز کمتر از روش‌های موجود است. با این حال، روش ارائه شده در زمینه مدیریت هستی‌شناسی کارآمد به بهبود بیشتری نیاز دارد [۱۷].

Zou و همکارانش از یک رویکرد مبتنی بر هوش مصنوعی در ترکیب سرویس‌های آگاه از کیفیت سرویس استفاده کردند. این روش با استفاده از الگوهای CSTE^{۱۴} و SCP^{۱۵} ترکیب سرویس بهینه را با استفاده از پارامترهای QoS در زمان اجرای مناسب‌تری انجام داده است [۱۸]. Immonen و همکارانش چارچوبی را برای طراحی ترکیب قابل اعتماد سرویس‌ها ارائه کردند. تا بر اساس آن بتوان از روش‌های ترکیب سرویس در صنعت نیز بهره برد. واضح است که توسعه روش استاندارد و ابزار برای تشویق صنعت به تغییر روش‌های سرویس مهندسی خود و تغییر شیوه برای سرویس‌های ترکیبی مهندسی قابل اعتماد اجتناب‌ناپذیر است [۱۹].

Wu و همکارانش، از الگوریتم Skyline پویا، برای ترکیب وب‌سرویس‌ها مبتنی بر کیفیت سرویس ارائه دادند که در آن انتخاب ترکیب سرویس‌های مبتنی بر وب به صورت پویا انجام می‌شد [۲۰]. روش پیشنهادی Lee و همکارانش با کاهش فضای جستجوی سرویس‌ها بهبود قابل توجهی در ترکیب خودکار به وجود آورده است [۲۱].

Jin و همکارانش با استفاده از الگوریتم ژنتیک روشی مؤثر به نام PGA جهت انتخاب سرویس‌های آگاه از همبستگی پیشنهاد داده‌اند. آن‌ها سرویس‌های با کیفیت بالاتر را با استفاده از همبستگی بین آن‌ها ترکیب نموده‌اند، اما در مقابل کارایی کاهش یافته است [۲۲].

Nagamoutou و همکارانش یک مدل جدید به منظور بررسی ترکیب وب‌سرویس‌هایی که از مدل ماشین‌های اتوماتیک پشته‌ای (ESAM)^{۱۶} استفاده می‌کنند، مطرح کرده‌اند. این مدل در شرایطی که فرآیندها شکست می‌خورند یا به بن‌بست می‌رسند اجرای بهتری را از خود نشان داده است [۲۳].

Kholyl و همکارانش چارچوبی برای مقابله با شکست‌های غیرمنتظره (مانند پیام‌های خطا SOAP^{۱۷}) در طول زمان ترکیب سرویس معرفی کردند. این چارچوب پیشنهادی بر روی پشته ترکیب سرویس‌ها به عنوان

است که می‌تواند در فرآیند پاسخ‌دهی به کاربران، به شکل سریع و کارا در دسترس باشد.

یک قانون در این مخزن به صورت چهارتایی^{۱۵} $r = \langle S, C_I, C_O, QoS \rangle$ تعریف شده است که در آن :

- S سرویسی است که قانون را تولید می‌کند.
 - C_I مجموعه مفاهیمی است که قانون به عنوان ورودی نیاز دارد و شامل ورودی‌های وب سرویس می‌باشد.
 - C_O مجموعه مفاهیمی است که قانون به عنوان خروجی نتیجه می‌دهد و شامل خروجی‌های وب سرویس می‌باشد.
 - QoS به کیفیت سرویس اشاره می‌نماید
- زمانی که یک درخواست از کاربر دریافت می‌شود، سرویس‌ها از مخزن قانون واکشی می‌گردند و سرویس‌هایی فیلتر می‌شوند که طبق درخواست کاربر و قوانین موجود در مخزن قانون غیرممکن است در نتایج نهایی ارائه گردند. براین اساس الگوریتم Top-K در پاسخ به پرس‌وجوی کاربر، فرایند فیلترسازی را به صورت موازی و به شرح زیر انجام می‌دهد [۱۰]:

- یک مجموعه Input با عناصر اولیه‌ای شامل ورودی‌های درخواست کاربر ایجاد می‌شوند.
- همه سرویس‌هایی که ورودی‌های آن‌ها شامل عناصر Input می‌باشند، یافت می‌شوند، سپس خروجی این سرویس‌ها به مجموعه Input اضافه و سرویس‌ها نگهداری می‌گردند.
- مرحله ۲ آن قدر تکرار می‌گردد تا سرویس دیگری پیدا نشود که ورودی‌های آن عناصر موجود در مجموعه Input باشد.
- در مرحله بعد همه سرویس‌هایی که خروجی آن‌ها شامل عناصر Output می‌باشند، یافت می‌شوند و سپس خروجی این سرویس‌ها به مجموعه Output اضافه و سرویس‌ها نگهداری می‌شوند.
- ورودی سرویس‌هایی که در مرحله قبل ذخیره شده بودند با خروجی سرویس‌هایی که هنوز در مخزن سرویس موجود هستند مقایسه شده و مرحله قبل آن قدر تکرار می‌شود تا سرویس دیگری جهت پردازش یافت نشود.

بعد از اجرای این الگوریتم یک گراف وابستگی (که در آن اگر یکی از خروجی‌های W_A با یکی از ورودی‌های W_B مطابقت داشته باشد، دو سرویس W_A و W_B به هم متصل می‌شود) ایجاد شده است و سرویس‌هایی که نمی‌توانند به وسیله درخواست کاربر اجرا گردند، فیلتر خواهند شد. این مسئله تا حد زیادی تعداد سرویس‌های کاندیدای ترکیب را کاهش می‌دهد.

۲-۳- الگوریتم Sim-Dijkstra

در این الگوریتم سرویس‌های وابسته بر اساس گراف وابستگی و با انجام جستجوی رو به جلو^{۱۶} از نود ریشه شناسایی شده و در صف اولویت قرار می‌گیرند. در هر مرتبه، سرویسی با بهترین مقدار کیفیت سرویس

دسترس‌پذیری و غیره (بسیار مهم می‌شوند. از آنجاکه ممکن است سرویس‌های واقعی با کیفیت سرویس متفاوت عمل کنند، توسعه‌دهندگان نرم‌افزار نیاز دارند یک مجموعه مناسب از سرویس‌های واقعی را که تفاهم‌نامه سطح سرویس^{۱۷} انجام شده را ارضا می‌کند، انتخاب کنند [۷].

کیفیت سرویس به سطح معینی از کیفیت اطلاق می‌شود که هر یک از شرکت‌ها برای رسیدن به آن سطح مشخص تلاش می‌کنند. به این ترتیب برای هر یک از خدمات باید مشخصات خاصی تهیه شود که در تفاهم‌نامه سطح سرویس، سطح کیفیت خدمات، مسئولیت‌ها و ضمانت‌های اجرای آن‌ها تعیین شده است [۲۶]. با توجه به این که امروزه اکثر شرکت‌ها ادعای ارائه بهترین خدمات را به مشتریان خود دارند، سرویس گرفتن از شرکت‌هایی که تفاهم‌نامه سطح سرویس دارند بهتر است چون میزان ناسازگاری و برخورد میان کاربر و ارائه‌دهنده خدمات به دلیل تعریف شدن سطح کیفی خدمات، کاهش خواهد یافت و کاربر با حقوق قانونی خود که یکی از علت‌های رشد کیفی شرکت ارائه‌دهنده خدمات محسوب می‌گردد، آشنا خواهد شد.

اولین مرحله به دست آوردن ترکیب بهینه وب سرویس‌ها ایجاد یک مدل مناسب برای توصیف ویژگی‌های کیفی آن‌ها می‌باشد. این مدل باید مورد توافق مشتری و فراهم‌کننده سرویس باشد. ویژگی‌های کیفی وب سرویس‌ها به دو دسته کلی، ویژگی‌های مثبت و ویژگی‌های منفی تقسیم می‌شود. مهم‌ترین این ویژگی‌ها عبارت‌اند از زمان پاسخ، هزینه، قابلیت اطمینان و قابلیت دسترسی. از جمله موارد دیگری که باید در مدل کیفیت سرویس مشخص شود چگونگی محاسبه مقدار مجموع ویژگی‌های کیفی وب سرویس مرکب می‌باشد. برای محاسبه مجموع ویژگی‌های کیفی بایستی یکی از ساختارهای الگوریتم‌های ترتیبی، موازی، هم‌زمانی و یا حلقوی را در نظر گرفت [۷]. در صورتی که چندین ویژگی کیفی برای هر سرویس انتخاب شود بایستی از تابع تناسب برای ارائه بهترین ترکیب استفاده نمود.

۲-۲- الگوریتم Top-K و استخراج زیر گراف وابستگی

از آنجایی که انتخاب سرویس مناسب از یک مجموعه سرویس با مقیاس بزرگ، سرعت زمان پردازش را افزایش می‌دهد، بنابراین بهتر است که تعداد سرویس‌های کاندید برای ترکیب کاهش یابد. بدین منظور لازم است که سرویس‌های بی‌فایده^{۱۴} در مراحل آغازین فیلتر شوند.

روش‌های مختلفی برای کاهش تعداد سرویس‌ها در ترکیب وجود دارد که از آن جمله می‌توان به روش خوشه‌بندی فازی C-Means [۲۷]، الگوریتم دسته‌بندی درخت سلسله مراتبی [۲۸]، روش خوشه‌بندی به همراه مدل مارکوف [۲۹] و الگوریتم دسته‌بندی Top-K [۱۰] اشاره کرد. در این مقاله از الگوریتم Top-K، به دلیل رفع مشکل ناسازگاری سرویس‌ها برای کاهش تعداد سرویس‌ها استفاده شده است.

در بخش اول این الگوریتم، سرویس‌ها، برای تبدیل به یک مخزن قانون پردازش می‌شوند. مخزن قانون یک نوع ساختار داده در حافظه

و کار پایان می‌یابد. در غیر این صورت مسیر کلیدی در صف اولویت قرار گرفته و مرحله بعدی به‌صورت بازگشتی انجام می‌شود.

۴- چهارم، مسیر کلیدی موجود در صف اولویت را بازیابی کرده و با استفاده از الگوریتم KPL رأس‌ها را Loose کرده تا بدین‌صورت مسیرهای کلیدی بدتر فراهم شده و در صف اولویت قرار گیرند.

لازم به ذکر است که الگوریتم KPL در صورتی اجرا می‌شود که تعداد گراف‌های بدون دور جهت‌دار تولید شده با استفاده از الگوریتم Sim-Dijkstra کمتر از درخواست کاربر باشد.

۳- روش پیشنهادی

روش‌های زیادی برای حل ترکیب سرویس‌ها همچون DFS، BFS و غیره ارائه شده است. در این پژوهش هدف، حل مسئله به کمک گراف رنگ‌آمیزی و مقایسه آن با الگوریتم دسته‌بندی Top-K بوده است. راهکار ارائه شده در این پژوهش در مقایسه با الگوریتم Top-K و روش ارائه شده توسط Jiang [۲۵]، استفاده از گراف رنگ‌آمیزی برای کاهش تعداد سرویس‌ها در زمان کوتاه‌تر و ترکیب بهتر سرویس‌های منفرد با هزینه کمتر است. روش‌های پیشنهادی با عناوین MGC-TopK و MGC-K ارائه گردید. در این روش‌ها سرویس‌هایی با کیفیت بهتر رنگ‌آمیزی می‌شوند، لذا از تعداد سرویس‌هایی که به‌عنوان رأس‌های این گراف شناخته می‌شوند کاسته شده و باعث می‌گردد عملیات انتخاب و ترکیب سرویس‌ها با سرعت بیشتر و مصرف حافظه کمتر انجام پذیرد.

در روش‌های پیشنهادی استفاده از گراف رنگ‌آمیزی باعث حذف سرویس‌های ناسازگار می‌شود، لذا از الگوریتم رنگ‌آمیزی گراف همراه و بدون الگوریتم Top-K در ترکیب سرویس‌ها استفاده شد تا کارایی روش پیشنهادی مشخص شود.

در روش MGC-TopK، بر روی خروجی الگوریتم Top-K، فیلتر مجدد سرویس‌ها بر اساس رنگ‌آمیزی انجام شده است، درحالی‌که در MGC-K، الگوریتم Top-K هیچ نقشی نداشته و رنگ‌آمیزی گراف مستقیماً بر روی سرویس‌ها به‌منظور عملیات فیلترینگ اعمال می‌شود. در هر دو روش پیشنهادی، فیلترسازی با استفاده از رنگ‌آمیزی گراف انجام شده است. سپس به‌منظور بررسی این موضوع که آیا استفاده از مزایای الگوریتم Top-K در رسیدن به مسیر بهینه‌تر می‌تواند مؤثر باشد یا خیر، مقایساتی بین این دو روش پیشنهادی انجام شده است.

۳-۱- روش MGC-TopK

در این بخش مرا حل اولین الگوریتم پیشنهادی که ترکیبی از الگوریتم Top-K و گراف رنگ‌آمیزی است شرح داده می‌شود. مراحل این الگوریتم به شرح زیر است:

۱- ابتدا بر روی مجموعه سرویس‌های اولیه الگوریتم Top-k اجرا می‌گردد و زیر گراف وابستگی استخراج می‌شود.

در صف اولویت کنترل می‌شود. درنهایت فرزندان فعال مشخص شده و در صف اولویت بازگشتی قرار می‌گیرند [۲۵].

این الگوریتم با اضافه کردن سرویس‌های فعال توسط نود ریشه در صف اولویت شروع می‌شود. یک سرویس هنگامی فعال است که تمام ورودی‌های آن را بتوان با استفاده از سرویس‌های دیگر فعال کرد. اولویت‌بندی سرویس‌های موجود در صف بر اساس مقدار کیفیت سرویس نهایی آن‌ها است. بنابراین هر بار، سرویسی که دارای بهترین کیفیت در صف اولویت است، برداشته می‌شود. برای هر خروجی از سرویس برداشته شده، از تعداد فرزندان^{۱۷} (ورودی‌های گره) مقدار یک کم می‌شود. سرویس‌های فعال جدید به صف اولویت اضافه شده و فرآیند فوق تا زمان خالی شدن صف به‌صورت بازگشتی ادامه می‌یابد. سپس زیرگراف حاصل از گراف اصلی توسط روش sim-Dijkstra پیمایش شده و مسیرهای بهینه مشخص و به کاربر معرفی می‌گردند.

۲-۴- KPL

همان‌طور که قبلاً بیان شد، پیدا کردن نتیجه ترکیب سرویس مطلوب، (تقریباً) کار چندان آسانی نیست [۲-۴]. بنابراین باید به دنبال یافتن روشی برای ساده کردن مسئله بود. در بخش قبل عنوان شد که روش sim-Dijkstra مسیرهای بهینه را مشخص و به کاربر معرفی می‌کند. حال در این زمان اگر مسیرهای پیشنهادی مورد تأیید کاربر قرار نگرفت لازم است تا مسیرهای بعدی در صورت وجود مشخص شوند بدین منظور الگوریتم KPL به همراه عملیات loose به‌صورت ذیل ایفای نقش می‌کند [۲۵].

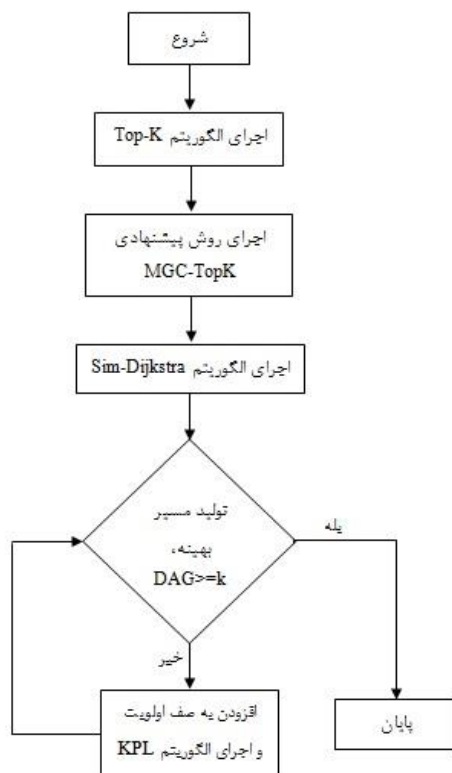
در ابتدا بدترین مسیر از مسیرهای اولیه به‌دست‌آمده، انتخاب و از آخرین گره شروع به بازگشت به اولین گره در مسیری نماید. در مسیر بازگشت هر گره‌ای که قبلاً در مسیر بهینه دسته‌بندی شده بود را نادیده گرفته و مابقی گره‌ها انتخاب می‌شوند و زمان آن‌ها استخراج می‌گردد. در پایان مسیرهای به‌دست‌آمده که اصولاً دارای زمان بدتری نسبت به مسیرهای بهینه اولیه می‌باشند جهت انتخاب در اختیار کاربر قرار می‌گیرند. بدترین مسیر انتخاب شده تحت عنوان مسیر کلیدی نام دارد. مسیر کلیدی یک گراف بدون دور جهت‌دار و زنجیره‌ای از سرویس‌های قبلی است که مقدار کیفیت سرویس کلی آن برابر با بدترین مقدار کیفیت سرویس کلی (allQoS)، در میان تمام ورودی‌های ارائه دهنده‌گان این سرویس است. به‌طور خلاصه، الگوریتم KPL شامل مراحل زیر است [۲۵]:

۱- اول، با استفاده از الگوریتم Sim-Dijkstra یک جستجوی رو به جلو از رأس شروع در گراف وابستگی انجام می‌شود.

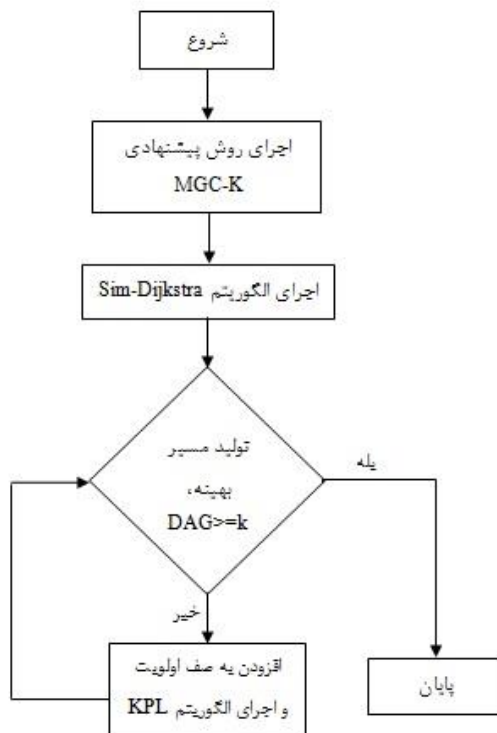
۲- دوم، بر اساس اطلاعات ثبت شده در الگوریتم Sim-Dijkstra مسیر کلیدی بهینه با انجام جستجوی رو به عقب بازیابی می‌گردد.

۳- سوم، با استفاده از مسیرهای کلیدی بهینه، یک گراف بدون دور جهت‌دار تولید می‌شود. اگر تعداد گراف‌های بدون دور جهت‌دار بزرگ‌تر از K باشد جواب به کاربر برگردانده می‌شود

کرده تا بدین‌صورت مسیرهای کلیدی بدتر فراهم شده و در صف اولویت قرار گیرند [۲۵]. شکل ۱ و شکل ۲ مراحل الگوریتم‌های پیشنهادی را نشان می‌دهد.



شکل ۱: چارت اجرای روش MGC-TopK



شکل ۲: چارت اجرای روش MGC-K

۲- گراف وابستگی حاصل از مرحله قبل، ابتدا طبق بخش بعدی رنگ‌آمیزی شده و فیلترینگ با استفاده از جستجوی رو به جلو انجام می‌شود.

۳- با استفاده از الگوریتم Sim-Dijkstra یک جستجوی رو به جلو از رأس شروع در گراف وابستگی که از مرحله دوم به‌دست‌آمده است انجام می‌شود.

۴- بر اساس اطلاعات ثبت شده در الگوریتم Sim-Dijkstra مسیر کلیدی بهینه با انجام جستجوی رو به عقب بازیابی می‌گردد.

۵- با استفاده از مسیرهای کلیدی بهینه، یک گراف بدون دور جهت‌دار تولید می‌شود. اگر تعداد گراف‌های بدون دور جهت‌دار بزرگ‌تر از K باشد جواب به کاربر برگردانده می‌شود و کار پایان می‌یابد. در غیر این صورت مسیر کلیدی در صف اولویت قرار گرفته و مرحله بعدی به‌صورت بازگشتی انجام می‌شود.

۶- درنهایت، مسیر کلیدی موجود در صف اولویت را بازیابی کرده و با استفاده از الگوریتم KPL رأس‌ها را Loose کرده تا بدین‌صورت مسیرهای کلیدی بدتر فراهم شده و در صف اولویت قرار گیرند [۲۵].

۲-۳- روش MGC-K

هدف از این روش استخراج زیرگراف وابستگی از گراف اولیه می‌باشد، به‌گونه‌ای که تعداد گره‌های باقی‌مانده بر روی k مسیر مورد نیاز استخراج گردد. مراحل اجرایی این روش مشابه روش اول بوده با این تفاوت که ابتدا الگوریتم Top-K حذف شده و تنها رنگ‌آمیزی گراف برای کاهش تعداد سرویس‌ها استفاده شده است. مراحل این روش به شرح زیر است:

۱- ابتدا بر روی مجموعه سرویس‌های اولیه که خود یک گراف بزرگ را تشکیل داده‌اند رنگ‌آمیزی طبق الگوریتم بخش بعد انجام می‌شود و زیر گراف وابستگی استخراج می‌شود.

۲- با استفاده از الگوریتم Sim-Dijkstra یک جستجوی رو به جلو از رأس شروع در گراف وابستگی که از مرحله دوم به‌دست‌آمده است انجام می‌شود.

۳- بر اساس اطلاعات ثبت شده در الگوریتم Sim-Dijkstra مسیر کلیدی بهینه با انجام جستجوی رو به عقب بازیابی می‌گردد.

۴- با استفاده از مسیرهای کلیدی بهینه، یک گراف بدون دور جهت‌دار تولید می‌شود. اگر تعداد گراف‌های بدون دور جهت‌دار بزرگ‌تر از K باشد جواب به کاربر برگردانده می‌شود و کار پایان می‌یابد. در غیر این صورت مسیر کلیدی در صف اولویت قرار گرفته و مرحله بعدی به‌صورت بازگشتی انجام می‌شود.

۵- درنهایت، مسیر کلیدی موجود در صف اولویت را بازیابی کرده و با استفاده از الگوریتم KPL رأس‌ها را Loose

۳-۳- رنگ‌آمیزی گراف

در این روش ابتدا گراف موجود با استفاده از قوانین گراف رنگ‌آمیزی به‌طور کامل رنگ می‌گردد. به‌گونه‌ای که در آن هیچ دو رأس مجاوری هم‌رنگ نباشند و از حداکثر m رنگ متفاوت استفاده شده باشد. برای گراف موجود، هر سرویس نشان‌دهنده یک رأس و ارتباط بین سرویس‌ها که بر اساس ورودی و خروجی هر سرویس است یک یال را نشان می‌دهد. تفاوت رنگ‌آمیزی پیشنهادی در این است که گره‌های موجود در یک سطح می‌توانند هم‌رنگ انتخاب شوند اما یک گره با فرزندان نباید هم‌رنگ باشد.

مراحل رنگ‌آمیزی گراف بدین‌صورت می‌باشد:

- ۱- در ابتدا یک رأس به‌عنوان گره شروع در نظر گرفته شده و رنگی به آن اختصاص داده می‌شود.
- ۲- در مرحله بعد مابقی رأس‌ها ملاقات شده و هر رأسی که رنگ نشده است، با توجه به قانون عدم هم‌رنگ بودن دو رأس مجاور رنگ‌آمیزی می‌گردد. در این مرحله گره‌های هم‌جوار سطری می‌توانند از یک رنگ باشند، اما فرزندان هر گره باید از قانون رنگ‌آمیزی تبعیت کنند.

- ۳- این چرخه تا زمانی که تمامی رأس‌ها پیمایش و رنگ شوند ادامه پیدا می‌کند.

پس از رنگ‌آمیزی گراف جریان فیلترسازی به شرح زیر اجرا می‌شود:

- ۱- ابتدا از رأس شروع تمامی فرزندان که هم‌رنگ می‌باشند انتخاب می‌گردند و هرکدام در دسته‌بندی‌های جداگانه قرار می‌گیرند.

- ۲- سپس با توجه به تعداد رأس‌های موجود برای هر رنگ بزرگ‌ترین دسته‌بندی انتخاب می‌شود.

- ۳- در این مرحله دسته‌بندی‌های صورت گرفته بر اساس مقادیر پارامترهای کیفیت سرویس به‌صورت صعودی مرتب می‌گردد.

- ۴- در صورتی که یک رأس بیش از ۳ فرزند داشته باشد مرحله بعدی جهت فیلترسازی رأس‌ها اجرا می‌گردد، در غیر این صورت تمامی فرزندان آن رأس مورد پیمایش قرار می‌گیرند.

- ۵- سپس جهت عملیات فیلترسازی بر اساس یک بازه که خود یکی از پارامترهای کیفیت سرویس می‌باشد مطابق با رابطه (۱) برای دسته انتخابی تعریف می‌شود.

(۳)

$$\begin{cases} \text{if } N \text{ is odd Then } Max = W_{(N+1)/2} \\ \text{if } N \text{ is even Then } Max = \frac{\left(\left(W_{\frac{N}{2}}\right) + \left(W_{\frac{N}{2}+1}\right)\right)}{2} \end{cases} \quad 1 \leq N \leq i$$

در این رابطه N تعداد رأس‌های هم‌رنگ از یک پدر می‌باشند و در اینجا به‌عنوان اندیس مقادیر پارامتریک کیفیت سرویس (هر نوع کیفیت سرویسی می‌تواند انتخاب گردد؛ که در روش پیشنهادی زمان پاسخ انتخاب شده است) استفاده گردیده است و همچنین W_i به‌عنوان مقدار

کیفیت سرویس هر رأس می‌باشد.

به‌منظور هرس گره‌های بلااستفاده در گراف رنگ‌آمیزی شده ابتدا یک گره به‌عنوان گره پدر در نظر گرفته شده و تمام فرزندان هم‌رنگ آن انتخاب می‌شوند. در ادامه با توجه به تعداد گره‌های هم‌رنگ فرزند و با استفاده از رابطه (۱) حداقل و حداکثر بازه مورد نیاز برای گره‌های باقی‌مانده مشخص و مابقی گره‌ها به دلیل آن که نمی‌توانند مسیر بهینه‌ای را در دسترس قرار دهند و یا به دلیل بلااستفاده بودن، حذف می‌گردند. بدین‌صورت گره‌هایی با وزن کمتر جهت انتخاب مسیر مناسب باقی می‌مانند. به‌عنوان مثال، اگر پارامترهای کیفیت سرویس رأس‌های هم‌رنگ به‌صورت جدول ۱ باشد، بازه مورد نظر جهت انتخاب رأس‌ها با استفاده از رابطه (۱) به شرح ذیل محاسبه می‌گردد.

از آنجایی که تعداد رأس‌ها ۵ است و فرد می‌باشد با توجه به رابطه (۱)، $W(5+1)/2 = 3$ معادل ۲۰ می‌شود. پس بازه اعداد جهت فیلترینگ رأس‌ها از ۵ تا ۲۰ انتخاب می‌گردد.

جدول ۱: اطلاعات پارامترهای کیفیت سرویس برای پنج رأس هم‌رنگ

N	۱	۲	۳	۴	۵
W _i	۵	۱۵	۲۰	۲۳	۳۸

سپس این بازه بر روی تمامی رأس‌های موجود در دسته حاضر و مابقی رأس‌ها اعمال می‌شود و هر رأسی که مقدار پارامتر کیفیت سرویس آن در بازه به‌دست آمده باشد انتخاب شده و مابقی رأس‌ها حذف می‌گردند.

- ۶- در نهایت مرحله اول الگوریتم تکرار شده و تا زمانی که به نقطه نهایی گراف برسد این روند ادامه پیدا می‌کند.

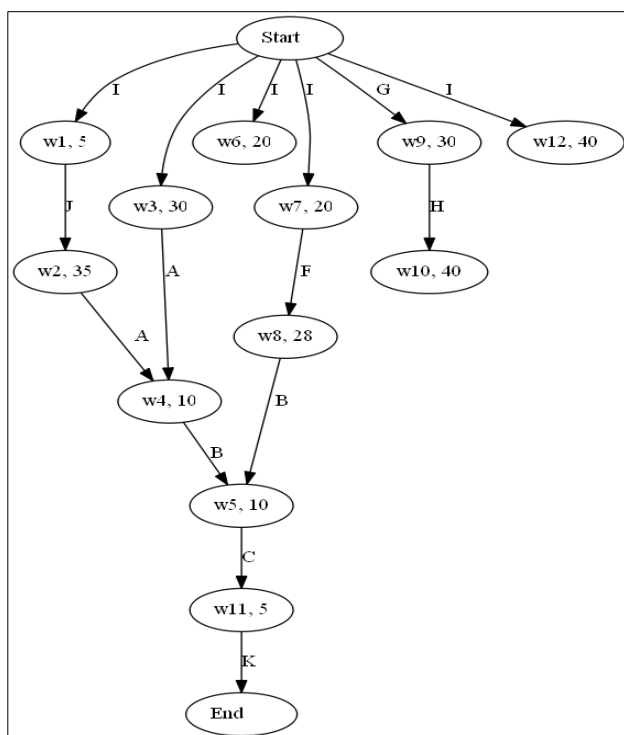
۴- ارزیابی

همان‌گونه که پیش‌تر نیز بیان گردید، ترکیب و فیلترینگ وب‌سرویس‌ها براساس پارامترهای کیفیت سرویس به‌وسیله الگوریتم‌هایی مانند Top-K انجام شده است. در روش MGC-TopK ترکیب سرویس‌های وب با استفاده از گراف رنگ‌آمیزی اصلاح شده و تلفیق آن با الگوریتم Top-K مطرح شده در [۲۵] انجام شده است. اما در روش MGC-K ترکیب سرویس‌های وب تنها با استفاده از گراف رنگ‌آمیزی اصلاح شده انجام شده است. در پایان، الگوریتم Sim-Dijkstra و KPL بر روی خروجی دو روش پیشنهادی اعمال شده و مورد ارزیابی قرار گرفته‌اند.

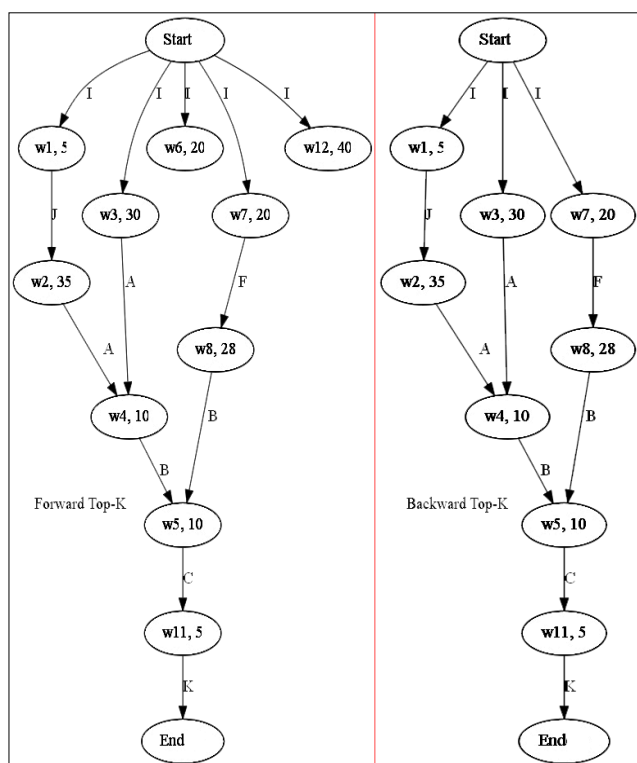
در این بخش به بررسی و سنجش نتایج شبیه‌سازی روش‌های MGC-K و MGC-TopK پرداخته شده است، قبل از بررسی نتایج و آزمایش‌ها، ابتدا شیوه پیاده‌سازی شرح داده می‌شود. در نهایت به بیان تجارب و ارزیابی روش مورد استفاده پرداخته شده و نتایج به‌دست آمده تحلیل می‌گردند.

۴-۱- مشخصات سیستم

پیکربندی سیستم مورد استفاده در پیاده‌سازی و شبیه‌سازی روش پیشنهادی MGC مطابق جدول ۲ می‌باشد.



شکل ۳: گراف سرویس‌های اولیه



شکل ۴: گراف خروجی الگوریتم Top-K

جدول ۲: پیکربندی سیستم مورد استفاده در شبیه‌سازی

Asus Tek N82JQ	مادربرد
Intel(R) Core (TM) i7	پردازنده
4GB DDR3	حافظه اصلی
Microsoft Windows 7 Ultimate 32 Bit	سیستم عامل

۴-۲- مجموعه داده انتخابی

مجموعه داده سرویس‌های مورد نظر (جدول ۳ و شکل ۳) به صورت تصادفی ایجاد شده است. ابتدا بر اساس بازه دریافتی از کاربر، سرویس‌ها تولید می‌گردند و محدوده ورودی و خروجی درخواست کاربر معین می‌شود. نتایج ترکیب سرویس با ایجاد یک گراف وابستگی آغاز می‌شود، که در آن اگر یکی از خروجی‌های W_A با یکی از ورودی‌های W_B مطابقت داشته باشد، دو سرویس W_A و W_B به هم متصل می‌شود. درخواست به صورت دو رأس Start و End در گراف نشان داده شده است. W_1 - W_{11} نشان‌دهنده سرویس‌ها و A-J ورودی/خروجی آن‌ها می‌باشد. سپس برحسب سرویس‌های ایجاد شده و ورودی و خروجی آن‌ها، برنامه در ۳ فاز مجزا اجرا می‌گردد. در فاز اول، ابتدا عملیات فیلترینگ با الگوریتم Top-K بر روی گراف وابستگی (شکل ۳) اجرا می‌شود که خروجی آن در شکل ۴ نشان داده شده است. در فاز دوم، همان‌گونه که در شکل الف-۵ نشان داده شده است گراف حاصل از خروجی Top-K رنگ‌آمیزی می‌شود. سپس با اجرای رابطه (۱) بر روی گراف رنگ‌آمیزی شده عملیات فیلترسازی انجام می‌شود (شکل ب-۵). در فاز سوم، ابتدا گراف اولیه را رنگ‌آمیزی کرده (شکل الف-۶) و سپس رابطه (۱) که مربوط به فیلترسازی سرویس‌ها است بر روی گراف رنگ‌آمیزی شده اعمال می‌شود (شکل ب-۶).

جدول ۳: داده‌های اولیه

Service Name	Input	Output	SelfQos
W1	I	J	5
W2	J	A	35
W3	I	A	30
W4	A	B	10
W5	B	C	10
W6	I	D	20
W7	I	F	20
W8	F	B	28
W9	G	H	30
W10	H	D	40
W11	C	K	5
W12	I	M	40

در نهایت، به منظور استخراج مسیرهای بهینه، الگوریتم KPL را بر روی گراف‌های خروجی دو روش پیشنهادی اعمال کرده و مطابق با درخواست کاربر مسیرهای بهینه مطلوب ارائه می‌شود.

۴-۳- بررسی نتایج شبیه‌سازی

برای تحلیل و ارزیابی روش‌های MGC-K و MGC-TopK، ابتدا پیاده‌سازی رویکرد ارائه شده توسط Jiang و الگوریتم Top-K بررسی گردید و سپس نتایج مورد نظر با داده‌های یکسان بر روی الگوریتم‌های ارائه شده توسط Jiang و روش‌های پیشنهادی MGC-K و MGC-TopK تحلیل شد. از طرفی به دلیل اهمیت الگوریتم‌های مبتنی بر هوش مصنوعی، روش‌های پیشنهادی با الگوریتم ترکیب سرویس مبتنی بر PSO [۱۴] و ژنتیک [۳۰] مقایسه گردید. همچنین میزان مصرف حافظه بر حسب بایت و زمان اجرا بر حسب میلی‌ثانیه محاسبه می‌شود.

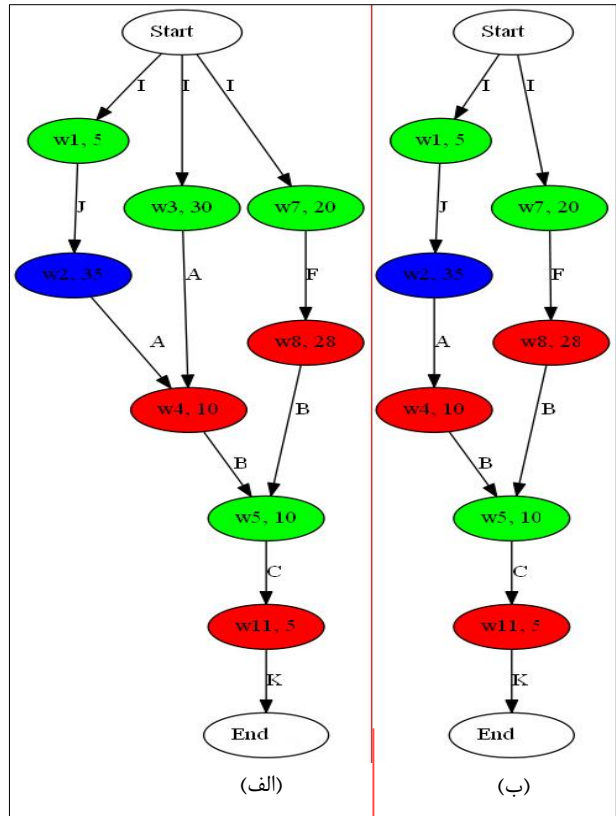
نتایج کاربرد روش‌های پیشنهادی MGC-K و MGC-TopK در مسئله یافتن چندین ترکیب بهینه از وب‌سرویس‌ها بدین شرح است:

۴-۳-۱- زمان اجرای فیلترسازی و کل اجرا

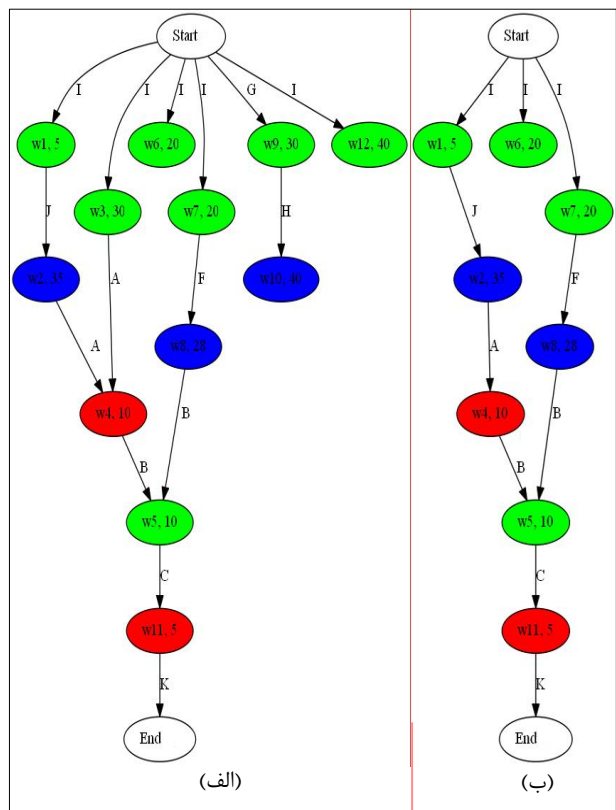
هدف این بخش، بررسی مناسب بودن عملیات فیلترسازی روش‌های پیشنهادی نسبت به الگوریتم Top-K، Jiang، PSO [۱۴] و ژنتیک [۳۰] است. در تحلیل نتایج به دست آمده، مشاهده می‌شود که با افزودن تعداد وب‌سرویس‌های مجموعه داده، زمان اجرا افزایش می‌یابد.

با توجه به این مسئله که در روش‌های پیشنهادی MGC-K و MGC-TopK، ترکیب وب‌سرویس‌ها با استفاده از مرحله فیلترسازی الگوریتم Top-K و همچنین گراف رنگ‌آمیزی اصلاح شده صورت گرفته است، لذا زمان اجرای فیلترسازی در تعداد وب‌سرویس‌های یکسان، با استفاده از الگوریتم Top-K نسبت به روش‌های MGC-TopK و MGC-K کاهش داشته است. علت افزایش زمان اجرای دو روش پیشنهادی این است که ابتدا گراف را رنگ‌آمیزی کرده و سپس عمل فیلترینگ را انجام می‌دهیم، اما در الگوریتم Top-K فقط عملیات فیلترسازی صورت گرفته است. همچنین MGC-K نسبت به MGC-TopK نیز از کاهش زمان بیشتری بهره برده است و بهبود مناسبی در زمان اجرای الگوریتم رخ داده است. همان‌طور که در شکل ۷ مشاهده می‌شود در داده‌هایی با تعداد سرویس‌های مختلف تفاوت زمان اجرا محسوس‌تر می‌باشد. در روش‌های ارائه شده علاوه بر به کارگیری الگوریتم Top-K جهت فیلترسازی سرویس‌ها از روش‌های پیشنهادی MGC-K و MGC-TopK مطابق روال‌هایی که قبلاً بیان شد، عملیات فیلترینگ انجام می‌شود.

زمان اجرا کل توسط روش‌های MGC-K و MGC-TopK با روش‌های Jiang، PSO [۱۴] و ژنتیک [۳۰] در این مرحله بررسی شده است. شکل ۸، زمان اجرای کل را در روش Jiang ژنتیک [۳۰]، PSO [۱۴] و روش‌های پیشنهادی با بازه‌های زمانی متفاوت و تعداد سرویس‌های یکسان نشان می‌دهد.



شکل ۵: گراف خروجی روش پیشنهادی MGC-TopK



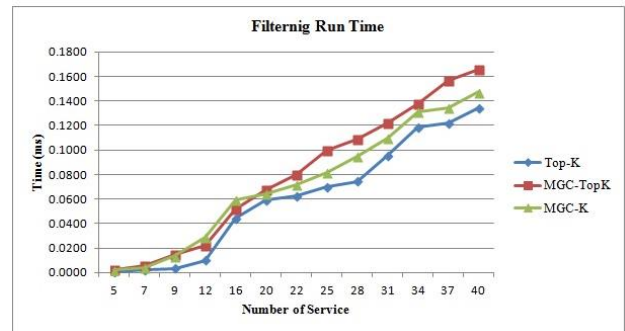
شکل ۶: گراف خروجی روش پیشنهادی MGC-K

گراف رنگ‌آمیزی انجام می‌پذیرد، مجموعه سرویس‌های کاندید جهت ترکیب کاهش یافته است (شکل ۹)؛ لذا میزان مصرف حافظه نسبت به روش Jiang کمتر و در مقایسه با روش‌های ژنتیک و PSO، تفاوت چشمگیری نداشته است و بهبود مناسبی در میزان مصرف حافظه اتفاق می‌افتد. شکل ۱۰، نمودار مقایسه را در بازه کوچک‌تر نشان می‌دهد.

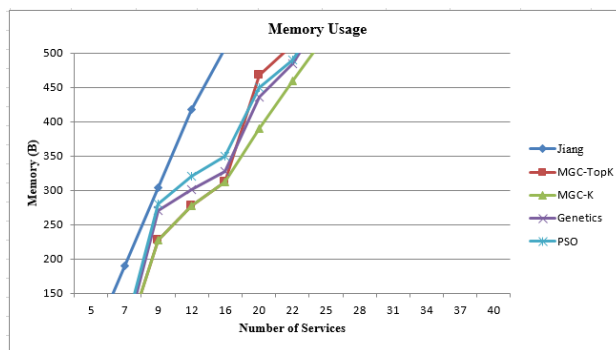


شکل ۹: نمودار مقایسه میزان مصرف حافظه روش Jiang و MGC-TopK و TopK و MGC-K و ژنتیک

نتایج نشان‌دهنده آن است که زمان کل اجرای روش Jiang نسبت به روش‌های MGC-K و MGC-TopK بالاتر بوده و روش‌های پیشنهادی توانسته‌اند تقلیل زمانی بیشتری را به دست آورند. همچنین MGC-K نسبت به ژنتیک و PSO نیز زمان اجرای بهتری دارد.

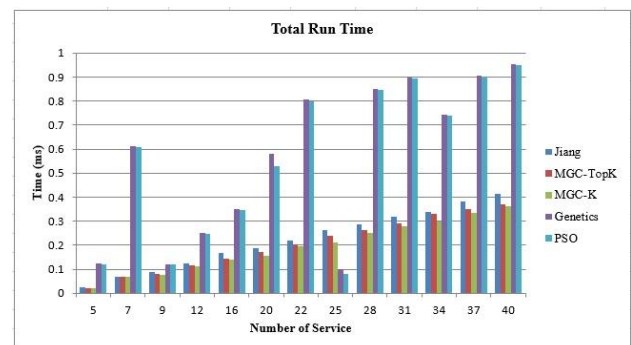


شکل ۷: نمودار مقایسه زمان اجرا در MGC-TopK و Top-K و MGC-K



شکل ۱۰: نمودار مقایسه میزان مصرف حافظه روش Jiang و MGC-TopK و TopK و MGC-K و ژنتیک در مقیاس کوچک‌تر

همان‌گونه که شکل ۸ نشان می‌دهد، الگوریتم ترکیب سرویس مبتنی بر ژنتیک [۳۰] دارای زمان اجرای بالاتری نسبت به مابقی روش‌ها است. زیرا مکان سرویس‌های کاندید متغیر است؛ در واقع ممکن است در برخی از حالات، ترکیب سرویس کاندید در موقعیتی قرار بگیرد که مابقی ذرات به آن نزدیک باشند در نتیجه سرعت همگرایی آن‌ها زیاد شده و سریع‌تر ترکیب انجام می‌پذیرد؛ اما در برخی حالات، سرویس‌های کاندید در محل دورتری از ذرات قرار گرفته و همین مسئله منجر به زمان طولانی برای ترکیب سرویس می‌گردد. در نتیجه زمان اجرای الگوریتم افزایش می‌یابد. البته در برخی موارد سرعت حرکت ذرات را هم بایستی در نظر گرفت.



شکل ۸: نمودار زمان کل اجرا در روش Jiang، ژنتیک و PSO و روش‌های پیشنهادی

۴-۳-۲- مصرف حافظه

مورد قابل بررسی دیگری که در نتایج به‌دست‌آمده قابل تحلیل است، میزان مصرف حافظه توسط روش‌های پیشنهادی MGC-K و MGC-TopK در مقایسه با روش پیشنهادی Jiang و مبتنی بر PSO و ژنتیک است.

نظر به این‌که در روش‌های پیشنهادی عملیات فیلترسازی بر اساس

در طی سال‌های اخیر، تعداد وب‌سرویس‌هایی که دارای عملکرد یک‌سان ولی کیفیت متفاوت می‌باشند، افزایش یافته است. همچنین با گسترش و بهبود خدمات وب، مهندسان سرویس متقاضی الگوریتم‌هایی برای ترکیب سرویس خودکار شدند که نه تنها عملیات ترکیب صحیح سرویس‌ها را برای هزاران سرویس انجام می‌دهند، بلکه الزامات کیفی کاربران را نیز فراهم می‌سازد.

در واقع این وب‌سرویس‌ها هستند که امکان تبادل بین کامپیوترها را در یک محیط ناهمگن که از سیستم‌های مختلفی تشکیل شده ممکن می‌سازند. از طرفی سرویس‌های ساده اغلب قادر به پاسخگویی به نیازهای کاربران نیستند، بنابراین نیاز به ترکیب سرویس‌های وب در جهت افزایش کارایی آن‌ها و انجام خدمات پیچیده‌تر از مباحث مهم در زمینه وب‌سرویس‌ها می‌باشد.

۵- نتیجه‌گیری

- [7] Klein, A., Ishikawa, F. and Honiden, S. "SanGA: A self-adaptive network-aware approach to service composition." *IEEE Transactions on Services Computing* 7, no. 3: 452-464, 2014.
- [8] Z. Brahmī, "QoS-aware Automatic Web Service Composition based on Cooperative Agents," 22nd International Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE), pp. 27-32. IEEE, 2013.
- [9] S. Deng, B. Wu, J. Yin and Z. Wu, "Efficient planning for top-K Web service composition," *Knowledge and information systems* 36, no. 3: 579-605, 2013.
- [10] S. Deng, L. Huang, W. Tan and Z. Wu, "Top-automatic service composition: A parallel method for large-scale service sets," *IEEE Transactions on Automation Science and Engineering* 11, no. 3: 891-905, 2014.
- [11] G. Zou, Y. Gan, Y. Chen and B. Zhang, "Dynamic composition of Web services using efficient planners in large-scale service repository," *Knowledge-Based Systems* 62: 98-112, 2014.
- [12] A.S. da Silva, H. Ma and M. Zhang, "A graph-based particle swarm optimization approach to qos-aware web service composition and selection," In 2014 IEEE Congress on Evolutionary Computation (CEC), pp. 3127-3134. IEEE, 2014.
- [13] A. S. da Silva, H. Ma and M. Zhang, "Genetic programming for QoS-aware web service composition and selection." *Soft Computing-A Fusion of Foundations, Methodologies and Applications*, no. 10: 3851-3867, 2016.
- [14] J. Liao, Y. Liu, X. Zhu and J. Wang, "Accurate sub-swarms particle swarm optimization algorithm for service composition," *Journal of Systems and Software* 90: 191-203, 2014.
- [15] Q. Yu, L. Chen and B. Li, "Ant colony optimization applied to web service compositions," in *cloud computing Computers & Electrical Engineering* 41: 18-27, 2015.
- [16] D. Wang, Y. Yang and Z. Mi, "A genetic-based approach to web service composition," in *geo-distributed cloud environment Computers & Electrical Engineering* 43: 129-141, 2015.
- [17] M. Li, D. Zhu, T. Deng, H. Sun, H. Guo and X. Liu, "GOS: a global optimal selection strategies for QoS-aware web services composition," *Service Oriented Computing and Applications* 7, no. 3: 181-197, 2013.
- [18] G. Zou, Q. Lu, Y. Chen, R. Huang, Y. Xu and Y. Xiang, "QoS-aware dynamic composition of Web services using numerical temporal planning," *IEEE Transactions on Services Computing* 7, no. 1: 18-31, 2014.
- [19] A. Immonen and D. Pakkala, "A survey of methods and approaches for reliable dynamic service compositions," *Service Oriented Computing and Applications* 8, no. 2: 129-158, 2014.
- [20] J. Wu, L. Chen and T. Liang, "Selecting dynamic skyline services for QoS-based service composition," *Applied Mathematics & Information Sciences* 8, no. 5: 2579, 2014.
- [21] C. H. Lee, S. Y. Hwang, I. L. Yen and T. K. Yu, "A service pattern model for service composition with flexible functionality," *Information Systems and e-Business Management* 13, no. 2: 235-265, 2015.
- [22] H. Jin, X. Yao and Y. Chen, "Correlation-aware QoS modeling and manufacturing cloud service composition," *Journal of Intelligent Manufacturing*: 1-14, 2015.
- [23] D. Nagamoultou, I. Egambaram, M. Krishnan and P. Narasingam, "A verification strategy for web services

اکثر روش‌های موجود برای ترکیب وب‌سرویس‌ها بدون در نظر گرفتن علایق کاربران، تنها یک راه‌حل را برمی‌گرداند که در برخی شرایط معین، بهینه می‌باشد و از نظر انعطاف‌پذیری و تنوع در سطح پایداری قرار دارد. در این تحقیق روشی ارائه شده است که با در نظر گرفتن پارامترهای مختلف کیفیت سرویس، نسبت به ایجاد چندین راه‌حل بر پایه گراف رنگ‌آمیزی اقدام می‌نماید. نتایج حاصل از شبیه‌سازی نشان می‌دهد روش‌های MGC-K و MGC-TopK علاوه بر کلیه مزایای ایجاد ترکیب سرویس بهینه، در موارد زمان کل اجرا و مصرف حافظه نسبت به روش‌های پیشین نتیجه بهتری دارند. در این راستا بعد از بررسی‌های به‌عمل آمده مشخص گردید که دلیل آنکه در روش MGC-K یک مرحله فیلترینگ نسبت به روش‌های MGC-TopK و Jaing کمتر انجام می‌شود و همچنین تعداد گره‌های بلا استفاده هرس شده بیشتر از دو روش دیگر است، سرعت بالاتری برای رسیدن به مسیر مناسب‌تر دارد. هرچند ممکن است مسیرهای پیشنهادی با روش MGC-TopK یکسان باشد.

به‌عنوان کارهای پیشنهادی آینده می‌توان به موارد ذیل اشاره نمود:

- ۱- بهبود الگوریتم از نظر زمان اجرای فیلترینگ.
- ۲- تست روش پیشنهادی با داده‌های واقعی
- ۳- مقایسه روش پیشنهادی با الگوریتم‌هایی مثل BFS و غیره
- ۴- توجه بیشتر به مدیریت سرویس‌هایی با ورودی و خروجی چندگانه که تشابهی بین ورودی و خروجی آن‌ها وجود دارد
- ۵- استفاده از عامل‌ها در عملیات فیلترسازی.
- ۶- استفاده از الگوریتم‌های هیوریستیک جهت انتخاب و ترکیب سرویس‌ها.

مراجع

- [1] W. Jiang, D. Lee, and S. Hu. "Large-scale longitudinal analysis of soap-based and restful web services," *IEEE 19th International Conference on Web Services (ICWS)*, pp. 218-225. IEEE, 2012.
- [2] E. Sirin, B. Parsia, D. Wu, J. Hendler and D. Nau, "HTN planning for web service composition using SHOP2," *Web Semantics: Science, Services and Agents on the World Wide Web* 1, no. 4: 377-396, 2004.
- [3] A. Zhou, S. Huang and X. Wang, "Bits: A binary tree based web service composition system," *International Journal of Web Services Research (IJWSR)* 4, no. 1: 40-58, 2007.
- [4] S.V. Hashemian, and F. Mavaddat. "A graph-based framework for composition of stateless web services," 4th European Conference on Web Services, pp. 75-86. IEEE, 2006.
- [5] De Oliveira, S. B., Balloni, A. J., Nogueira, F., & Toda, F. A., "Information and service-oriented architecture & web services: enabling integration and organizational agility", *Procedia Technology* 5, pp. 141-151, 2012.
- [6] 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.

- Local Feature Weighting,” *Tabriz Journal of Electrical Eng.*, vol. 46, no.2, summer 2016.
- [28] S. Abdollahzadeh, M. A. Balafar, and L. Mohammad Khanli, “Using Clustering and Markov Model in Predicting Web Users' Next Request,” *Tabriz Journal of Electrical Eng.*, vol. 45, no. 3, autumn 2015.
- [29] M. Rafiee, M. abbasi, and M. Nassiri, “An Efficient Method for Parallel Implementation of H-Trie Packet Classification Algorithm on GPU,” *Tabriz Journal of Electrical Engineering*, vol. 46, no. 3, autumn 2016.
- [30] Yuan, Yuan, Xiuguo Zhang, Wenxi Sun, Zhiying Cao, and Hao Wang. "Optimal web service composition based on context-awareness and genetic algorithm." *International Conference on Information Science and Cloud Computing Companion (ISCC-C)*, pp. 660-667. IEEE, 2013.
- composition using enhanced stacked automata model,” *Springer Plus* 4, no. 1: 1, 2015.
- [24] M. El Kholy and A. El Fatatry, “FRWSC: a framework for robust Web service composition,” *Service Oriented Computing and Applications*: 1-23, 2016.
- [25] W. Jiang, S. Hu and Z. Liu, “Top K query for QoS-aware automatic service composition,” *IEEE Transactions on Services Computing* 7, no. 4: 681-695, 2014.
- [26] Klopper B, Ishikawa F, Honiden S, “Service Composition with Pareto-Optimality of Time-Dependent QoS Attributes”, in *Service-Oriented Computing, Lecture Notes in Computer Science*. Berlin, Germany: Springer-Verlag, no. 6470: 635-640, 2010.
- [27] S. Rafiee, and P. Moradi, “Improving Performance of Fuzzy C-means Clustering Algorithm using Automatic

زیر نویس‌ها

¹² Modified algorithm of Graph Coloring

^{۱۳} Service Level Agreement (SLA)، تفاهم‌نامه سطح کیفی خدمات،

قراردادی است حقوقی فی مابین ارائه‌دهنده و استفاده‌کننده از خدمات که

جهت تضمین پارامترهای کیفی سرویس منعقد می‌گردد.

¹⁴ Useless Services

^{۱۵} 4-tuple

¹⁶ Forward search

¹⁷ successor

¹ Automatic Service Composition

² Dependency Graph

³ Quality of Service (QoS)

⁴ Palning Domain Definition Language

⁵ Accurate Sub-swarms Particle Swarm Optimization

⁶ Global Optimization Selection

⁷ Cost Sensitive Temporally Expressive

⁸ SAT-based Cost Planning

⁹ Enhanced Stacked Automata Model

¹⁰ Simple Object Access Protocol

¹¹ Key-Path-Based Loose