مدل‌سازی ترکیب خودکار سرویس‌ها، آگاه از کیفیت سرویس، مبتنی بر مساله پوشش مجموعه

مرتضی خانی دهنویی[۱]، سعید عربان[۲]، مهدی خانی دهنویی[۳]

[۱] دانشجوی دکتری، گروه کامپیوتر، دانشگاه فردوسی مشهد،
morteza.khani2@gmail.com

[۲] استادیار، گروه کامپیوتر، دانشگاه فردوسی مشهد،
araban@um.ac.ir

[۳] دانشجوی دکتری، گروه کامپیوتر، دانشگاه فردوسی مشهد،
m_khani_dehnoi@yahoo.com

چکیده

ترکیب سرویس‌ها در معماری سرویس‌گرا به ایجاد هماهنگی میان تعدادی از سرویس‌های موجود برای فراهم آوردن یک سرویس ترکیبی با ارزش افزوده اطلاق می‌شود و منظور از آن دست‌یافتن به برخی از نیازمندی‌های کاربر است که یک سرویس به تنهایی قادر به برآورده ساختن آن‌ها نیست. در این مقاله مدلسازی مساله ترکیب خودکار سرویس‌ها به صورت پوشش مجموعه پیشنهاد و از الگوریتم تقریبی برای حل آن بهره گرفته شده‌است. روش پیشنهادی، ترکیب سرویس‌های وب را در دو فاز اصلیِ تولید و گسترش بالا به پایین درخت ترکیب، و تولید سرویس ترکیبی از طریق پیمایش پایین به بالای درخت ترکیب مطرح می‌کند.

در فاز اول، تولید یک درخت (مشابه تولید درخت در حل مساله با جستجو) با شروع از خروجی یا پس شرط‌های سرویس درخواستی به سمت ورودی یا پیش شرط‌های آن پیشنهاد می‌شود. هر گره یا حالت از درخت، مجموعه‌ای از ورودی و خروجی‌ها یا شرایط است و سرویس‌ها به عنوان یال‌های درخت امکان انتقال از یک گره به گره دیگر را فراهم می‌کنند.

در فاز دوم، رسیدن از برگ‌های درخت تولید شده به ریشه، معادل رسیدن به خروجی سرویس ترکیبی در نظر گرفته می‌شود و مسیر رسیدن به ریشه، سرویس‌های دخیل و طرح ترکیب را مشخص می‌کند. برگ‌هایی از درخت که محتوای آن‌ها زیر مجموعه ورودی سرویس درخواستی است قابل دسترس هستند. برای رسیدن به هرگره غیر برگ از درخت یک مساله پوشش مجموعه تولید و با استفاده از یک الگوریتم تقریبی حریصانه حل می‌شود. اگر تولید و حل مساله پوشش مجموعه به صورت سلسله مراتبی تا رسیدن به گره ریشه ادامه پیدا کند، طرح و هزینه سرویس ترکیبی مورد تقاضا تولید خواهد شد.

ترکیب زنجیره‌ای و موازی سرویس‌ها به صورت توأم، نگاشت خواص کیفیتی سرویس به هزینه با استفاده از تحلیل مدل برداری ، ترکیب نزدیک به بهینه و آگاه از کیفیت سرویس براساس درخواست، و امکان تولید پویا و توزیع شده سرویس ترکیبی، بیشترین تمرکز این پژوهش را به خود معطوف کرده است.

کلمات کلیدی

وب سرویس، ترکیب سرویس، مساله پوشش مجموعه، الگوریتم تقریبی.

۱- مقدمه و پیشینه

یکی از بزرگترین توانایی‌های بالقوه فناوری سرویس که باید مورد توجه قرار گیرد، قابلیت همکاری بین سرویس‌ها در داخل یا خارج از مرزهای سازمان صاحب سرویس است. اگر سرویسی برای انجام وظایف عملکردی خود سرویس‌های دیگری را فراخوانی کند، به سرویس فراخواننده، سرویس مرکب[1] و به سرویس‌های خوانده شده، سرویس‌های پایه[2] می گویند. برای افزایش کارایی، یک سرویس مرکب به سرویس‌هایی که فراخوانی می کند (سرویس‌های پایه) وابسته نیست [1]. از اهداف مهم ترکیب سرویس‌ها دستیابی به حداکثر انعطاف پذیری ممکن در تطبیق پویا با محیط است، چرا که سرویس‌های شریک (سرویس‌های پایه) از نظر در دسترس بودن، توازن بار یا کاربرد تغییر می کنند [2].

ترکیب سرویس‌ها معمولا به صورت یکی از دو راهبرد هم‌نواسازی[3] یا هم‌خوانی[4] اجرا می‌شود. در هم‌نواسازی یک فرآیند مرکزی که می تواند یک سرویس باشد، سرویس‌های دخیل در ترکیب را کنترل و هماهنگ می کند. سرویس‌های دخیل در ترکیب نباید از این واقعیت که آن‌ها قسمتی از یک فرآیند کسب وکار سطح بالاتر هستند، با خبر شوند. هماهنگ کننده مرکزی[5] باید از هدف کلی فرآیند، تعریف اعمال و ترتیب به خدمت گرفتن وب سروییس‌های دخیل در ترکیب مطلع باشد.

در طرف دیگر، هم‌خوانی بدون یک فرآیند مرکزی عمل می کند. به جای استفاده از هماهنگ کننده، هر سرویس دخیل در عملیات ترکیب می داند که چه زمانی باید همکاری کند، و با چه سرویسی تعامل داشته باشد. عمل همکاری بر پایه رد و بدل پیام‌ها انجام می گیرد. هر یک از سرویس‌های دخیل در ترکیب باید از فرآیند کسب و کار کلی (وظیفه سرویس ترکیبی)، اعمالی که باید انجام گیرد، پیام‌هایی که باید رد و بدل شود و زمان بندی این پیام‌ها مطلع باشد. [۱۳, ۱۴]

سرویس ترکیبی می تواند به صورت خودکار یا توسط عامل انسانی متخصص[6] طراحی شود. بر اساس نظر [3] و [4] ، ترکیب خودکار سرویس‌ها روشی برای ترکیب سرویس‌ها در زمان اجرا (در حرکت[7]) و بدون دخالت انسان در تولید زنجیره ترکیب ارایه می‌کند. در این روش یک درخواست سرویس که شامل مشخصات سرویس درخواستی است، موجب تولید سرویسی متشکل از چند جزء خواهد شد که تا قبل از درخواست وجود خارجی نداشته است.

بر اساس نظر [5] و [4] تکنیک‌های ترکیب خودکار سرویس‌ها در دو دسته تکنیک‌های بر اساس جریان‌کار[8]و تکنیک‌های بر اساس برنامه‌ریزی در هوش‌مصنوعی[9] طبقه بندی می‌شوند. تکنیک‌های بر پایه جریان کاری، یک سرویس ترکیبی را مجموعه ای از سرویس‌های غیر قابل تجزیه (اتمیک) به همراه جریان‌کار که شامل جریان داده و جریان کنترل اجرای سرویس‌های دخیل می‌باشد، در نظر می‌گیرند. تکنیک‌های بر پایه جریان‌کاری، روش‌هایی خودکار برای انقیاد گره‌های مجرد به منابع یا سرویس‌های واقعی و عینی فراهم می کنند.

از طرف دیگر ترکیب خودکار می تواند بر اساس برنامه‌ریزی در هوش‌مصنوعی بنا شود. در این روش‌ها، فرض بر این است که هر سرویس می تواند به وسیله پیش شرط‌ها و تاثیرات اجرایش روی محیط تعریف و شناسایی شود. به این ترتیب یک طرح یا فرآیند می تواند، به صورت خودکار با استفاده از تئوری منطق یا متدهای برنامه‌ریزی در هوش‌مصنوعی تولید شود، بدون آنکه دانشی درباره جریان‌کاری، از پیش تعریف شده باشد.

از یک دیدگاه روش‌های ترکیب سرویس‌ها را می‌توان به دو دسته کلی ایستا و پویا تقسیم کرد. در ترکیب سرویس‌ها به صورت ایستا باید اطلاعات سرویس‌های در دسترس، که می‌توان از آن‌ها در ترکیب استفاده کرد، قبل از شروع عملیات ترکیب آماده و در یک جا مجتمع باشد. با توجه به تعداد زیاد سرویس‌ها و به روز رسانی دایمی آن‌ها توسط فراهم کنندگان و هم چنین اضافه شدن سرویس‌های جدید به مجموعه سرویس‌های در دسترس، جمع آوری اطلاعات سرویس‌های در دسترس برای انجام عملیات ترکیب مشکل اصلی این روش‌ها است. به همین دلیل روش‌های ایستا برای ترکیب سرویس‌ها همواره با یک نوع عدم انعطاف پذیری ذاتی همراه هستند. هم چنین به دلیل عدم توانایی در توجه به توازن بار روی سرویس‌های شریک در ترکیب، مقیاس پذیری نیز در این روش‌ها محدودیت‌هایی دارد.

در ترکیب پویای سرویس ها، با توجه به وضعیت حاضر دنیای وب زنجیره ترکیب تشکیل می‌شود. این بدان معنی است که دیگر نیازی به جمع آوری همه‌ی اطلاعات سرویس‌های در دسترس به صورت متمرکز و قبل از انجام عملیات ترکیب نیست.

۲- پیش نیازهای راهکار پیشنهادی

۲-۱- حل مساله با جستجوی درخت

در صورتی که فضای مساله را بتوان به صورت یک درخت مدل کرد حل مساله با جستجو می‌تواند یک کاندیدای مناسب برای حل مساله باشد. در این صورت برای فرمول بندی دقیق مساله باید فضای حالت مساله، حالت اولیه، مجموعه حالت‌های نهایی، تابع جانشین و تابع ارزیابی یا هدف به صورت دقیق توصیف شوند. سپس بسته به ماهیت مساله و اطلاعات در دسترس از آن یک استراتژی مناسب برای جستجو در فضای حالت انتخاب می‌شود. رسیدن از حالت اولیه به یک حالت نهایی یک راه حل برای مساله خواهد بود.

در مساله ترکیب سرویس های وب از آنجایی که به خدمت گرفتن هر سرویس مجموعه ای از ورودی ها یا پیش شرط ها را به مجموعه ای از خروجی ها یا پس شرط ها تبدیل می کند و این خروجی می تواند ورودی یک یا چند سرویس دیگر را فراهم کند، فضای مساله می تواند به صورت یک درخت مدل شود و استفاده از حل مساله با جستجو گزینه مناسبی خواهد بود. بدین ترتیب، هر گره یا حالت از درخت مجموعه‌ای از ورودی و

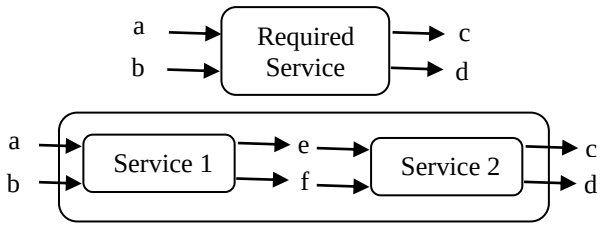
---

[1] Composite Service
[2] Basic Service
[3] Orchestration
[4] Choreography
[5] Coordinator
[6] Domain Expert
[7] On the fly
[8] Workflow Techniques
[9] AI Planning Techniques

خروجی‌ها یا شرایط است و سرویس‌ها به عنوان یال‌های درخت امکان انتقال از یک گره به گره دیگر را فراهم می‌کنند.

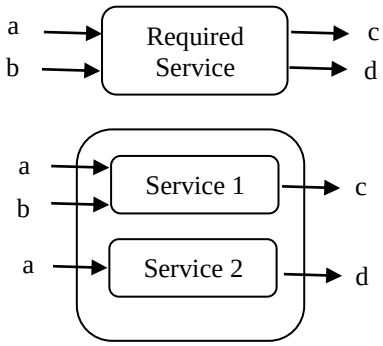
۲-۲- ترکیب زنجیره‌ای و موازی سرویس‌ها

در صورتی که سرویس مرکب درخواستی با به خدمت گرفتن متوالی دو یا چند سرویس حاصل شود، ترکیب زنجیره‌ای یا متوالی است.



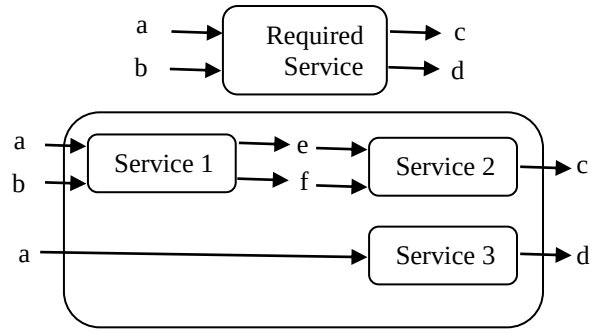
شکل 1 : ترکیب زنجیره‌ای یا متوالی

از طرف دیگر، در صورتی که قسمتی از نیازمندیهای عملکردی سرویس درخواستی با به خدمت گرفتن یک سرویس و قسمتی دیگر از نیازمندی‌های آن با به خدمت گرفتن یک یا چند سرویس دیگر برآورده شود، ترکیب موازی است.



شکل 2: ترکیب موازی

طرح یک سرویس مرکب می‌تواند در چند سطح توالی و توازی توصیف شود.



شکل 3 : ترکیب یک سطح متوالی و یک سطح موازی

هرچند که اکثر روش‌های ترکیب خودکار سرویس‌های وب به ترکیب زنجیره‌ای سرویس‌ها پرداخته‌اند [8]–[6]، اما تلاش‌هایی هم برای ترکیب موازی وب سرویس‌ها انجام گرفته‌است [10] ,[9]. در این پژوهش امکان ترکیب زنجیره‌ای و موازی خودکار سرویس‌ها به صورت توأم درنظر گرفته شده‌است.

۲-۳- مساله پوشش مجموعه

مساله پوشش مجموعه یک مساله کلاسیک در علوم کامپیوتر و تئوری پیچیدگی است و جزء ۲۱ مساله معروف Karp است که NP-Complete بودن آن‌ها در سال ۱۹۷۲ اثبات شده‌است [11].

مجموعه مرجع U و مجموعه S شامل m مجموعه دیگر را نیز در نظر بگیرید، به صورتی که اجتماع m زیر مجموعه داخل S برابر U باشد. یعنی مجموعه S مجموعه U را بپوشاند. مساله پوشش مجموعه، شناسایی مجموعه C به عنوان کوچکترین زیر مجموعه از S است که اجتماع مجموعه‌های داخل آن برابر U باشد. یعنی مجموعه C نیز مجموعه U را پوشش دهد.

مثلا اگر مجموعه مرجع $U = \{1,2,3,4,5\}$ و $S = \{\{1,2,3\},\{2,4\},\{3,4\},\{4,5\}\}$ در نظر گرفته شود، با حل مساله پوشش مجموعه نتیجه برابر $\{\{1,2,3\},\{4,5\}\}$ خواهد بود.

اگر انتخاب هر مجموعه داخل S یک عدد به عنوان هزینه داشته باشد، مساله از نوع پوشش مجموعه وزندار[10] خواهد بود.

مساله پوشش مجموعه را می‌توان به صورت برنامه‌ریزی خطی صحیح[11] فرمول بندی کرد و الگوریتم‌های تقریبی[12] خوبی از جمله الگوریتم حریصانه جانسون [12] با ضریب تقریب لگاریتمی و مرتبه زمانی چندجمله‌ای برای آن وجود دارد. از روش‌های ابتکاری مانند الگوریتم ژنتیک در [13] و الگوریتم کلونی مورچه در [14] نیز برای بهتر کردن تقریب استفاده می‌شود.

از کاربردهای مساله پوشش مجموعه می‌توان به استفاده در مسیریابی و مدیریت شبکه‌های کامپیوتری [15]، تعیین محل جغرافیایی تجهیزات و سیستم‌های حمل و نقل[16]، زمان بندی [17] اشاره کرد.

با توجه به پیشینه کاوش شده، در حل مساله ترکیب سرویس از مساله پوشش مجموعه استفاده نشده‌است و تنها در تعدادی از پژوهش‌ها مانند [18] و [19] برای اثبات NP-Complete بودن مساله ترکیب از کاهش آن به مساله پوشش مجموعه استفاده شده‌است.

۲-۳-۱- الگوریتم حریصانه برای حل مساله پوشش مجموعه

الگوریتم حریصانه جانسون[12] در هر مرحله مجموعه‌ای را از S انتخاب می‌کند که شامل بیشترین بهره وری یا به عبارتی کمترین هزینه به ازای هر عضو انتخاب نشده باشد. به این منظور برای هر مجموعه، حاصل تقسیم هزینه بر تعداد عضو را محاسبه کرده و مجموعه‌ای که کمترین حاصل تقسیم را داشته باشد، انتخاب می‌کند. مجموعه‌ی انتخاب شده به C اضافه می‌گردد و اعضای آن از U و مجموعه‌های انتخاب نشده حذف خواهد شد. الگوریتم زمانی تمام است که U تهی شده باشد. در [12]

[10] weighted set cover

[11] Integer linear programming

[12] approximation algorithms

اثبات شده‌است که ضریب تقریب این الگوریتم $H(n)$ است که در آن n تعداد اعضای مجموعه مرجع U است. این بدان معنی است که الگوریتم جانسون مجموعه‌هایی را برای پوشش U انتخاب می‌کند که آن‌ها ممکن است حد اکثر $H(n)$ برابر، بزرگتر از هزینه پوشش مینیمم باشد. لازم به ذکراست که $H(n)$ ، nامین عدد هارمونیک است و از رابطه زیر بدست می‌آید.

$$H(n) = \sum_{k=1}^{n} \frac{1}{k} \leq \ln n + 1$$

۲-۴- آگاهی از کیفیت در ترکیب سرویس ها

در پس وظایف تابعی یک سرویس، سرویس‌ها با ویژگی‌های غیر تابعی (کیفیتی) خود شناخته می شوند که معمولا از این ویژگی‌ها تحت عنوان ویژگی‌ها یا خصوصیات کیفیت سرویس (QoS)[۱۳] یاد می شود. اهمیت ویژگی‌های کیفیت سرویس آن جایی خود را نشان می دهد که از بین چند سرویس که از نظر کارکرد یکسان هستند، آن سرویسی که نیازمندی‌های کیفیت سرویس مشتری را بهتر ارضا کند، به خدمت گرفته خواهد شد.

برخی از پژوهشگران [20] کیفیت سرویس و ترجیحات کاربر[۱۴] را دو مقوله جدا در نظر گرفته‌اند و معتقدند بعضی از خواص کیفیتی مانند صداقت[۱۵] قابل ترجیح دادن نیستند و روش‌های ترکیب باید آن‌ها را بدون ذکر نیاز توسط کاربر در نظر بگیرند.

همه جنبه‌های کیفیت سرویس باید با هم و بر اساس نیازمندی‌های مصرف کننده سرویس، در نظر گرفته شود. در بعضی مواقع اگر سرویس‌هایی با جنبه‌های کیفیتی نامناسب استفاده شود، می‌تواند آثار جانبی نامطلوبی به همراه داشته باشد. مثلا در یک تراکنش بانکی، اگر به امنیت در به خدمت گرفتن سرویس توجه نشود، ممکن است منجر به مشکلات جبران ناپذیری شود. یا اگر جنبه هزینه، مورد توجه قرار نگیرد و سرویس گرانی به خدمت گرفته شود، ممکن است منجر به کاهش رضایتمندی مصرف کننده سرویس شود.

با عنایت به این موضوع که انتخاب وب سرویس‌ها برای نقش‌های زنجیره ترکیب باید به صورت خودکار انجام شود و بر عهده عامل نرم افزاری است، عامل انسانی که درخواست کننده اصلی بوده‌است به صورت مستقیم با سرویس‌های به خدمت گرفته شده برای نقش‌ها در معماری سرویس‌گرا ارتباط ندارد. این موضوع نیاز به رابطی را که نیازهای غیر تابعی عامل انسانی را به عامل نرم افزاری منتقل کند، تایید می‌کند. عامل نرم افزاری نیز باید با اطلاعاتی که از عامل انسانی دریافت کرده است، نیازمندی‌های غیر تابعی را در انتخاب وب سرویس‌ها برای نقش‌ها مد نظر قرار دهد.

لازم به ذکر است که عدم توجه به جنبه‌های کیفی سرویس‌ها در غالب روش‌های ترکیب خودکار وب سرویس‌ها برای نقش‌ها در معماری سرویس‌گرا، باعث کاربردی نشدن این روش‌ها شده‌است.

برای انتقال مقادیر کیفیت سرویس از عامل انسانی به عامل نرم افزاری و یا ارزیابی خواص کیفیتی از یک سرویس توسط عامل نرم افزاری، باید بتوان کیفیت سرویس را به صورت رسمی و مکتوب نگاشت کرد. به عبارت دیگر، نیاز به سیستمی است که هم بتواند داده‌های کیفی فراهم شده توسط کاربر را به صورت یکنواخت و استاندارد کمی کند، و هم مکانیسمی برای مقایسه دو سرویس از نظر کیفیت فراهم کند.

روش‌های پیشنهاد شده برای این کار را می‌توان به دو دسته اصلی تقسیم کرد. دسته اول سعی دارد سرویس‌ها را از نظر کیفی دسته بندی کند. به این ترتیب جستجوی یک سرویس می‌تواند در کلاس مورد نظر کاربر انجام شود. در این روش‌ها هر سرویس به محض اینکه در دسترس کاربران قرار می‌گیرد، در یکی از کلاس‌های از پیش تعریف شده ثبت نام می‌شود. هم چنین طبق قوانینی سرویس‌های موجود می‌توانند بین کلاس‌های موجود گذار کنند. نمونه‌ای از این روش‌ها در [21]–[23] آمده است. نارسایی‌های اصلی در این روش‌ها، تخمین تعداد و توصیف مناسب کلاس‌ها، و تشخیص کلاس یک سرویس جدید(شروع سرد[۱۶]) است.

دسته دوم از روش‌های انتقال کیفیت سرویس، بر پایه‌ی داخل کردن مقادیر خواص کیفیت سرویس در بسته بندی سرویس است. یعنی در توصیف سرویس، علاوه بر رابط سرویس (توصیف پیش‌نیازهای سرویس و نتایج حاصل از آن) و توصیف تابعی سرویس (معرفی عملیات سرویس)، یک بخش برای توصیف غیر تابعی (کیفی) سرویس نیز گنجانده می‌شود. تفاوت روش‌های این دسته در نحوه‌ی نگاشت از داده‌های کیفی به داده‌های کمی است. بعضی از روش‌های این دسته (مانند [24]) کیفیت سرویس را به یک عدد واحد و بعضی دیگر (مانند [25], [26]) به یک بردار و تعداد اندکی نیز به یک تابع توزیع نگاشت می‌کنند. نگاشت کیفیت سرویس به یک بردار کیفیت از همه معمول‌تر است. روش پیشنهاد شده توسط پژوهش حاضر که در ادامه آمده است، در این دسته جای می‌گیرد.

۲-۵- طراحی رابطی برای انتقال خواص کیفیت سرویس و مقایسه کیفیتی سرویس

در این قسمت رابطی برای انتقال خواص کیفیت سرویس و مقایسه کیفیتی سرویس‌ها با استفاده از تحلیل مدل برداری طراحی می‌شود. کیفیت سرویس، ترکیبی از چندین خاصیت کیفی از یک سرویس است. جنبه‌هایی از کیفیت سرویس که از نظر متخصصان [27] با معماری سرویس‌گرا مرتبط هستند، در زیر لیست شده‌است:

۱- هزینه[۱۷] (q_c): جنبه‌های مرتبط با هزینه یا شارژ یک سرویس را بر می‌گرداند.

۲- کارایی[۱۸] (q_{perf}): سرعت اجرای سرویس را مد نظر قرار می‌دهد. سرعت، علاوه بر کارایی ممکن است به صورت زمان پاسخ[۱۹]، نرخ تولید خروجی[۲۰] و غیره نیز نشان داده شود. کارایی در بعضی مواقع به

[13] Quality Of Service Characteristics
[14] User preferences
[15] Trustworthy
[16] Cold Start
[17] Cost
[18] Performance
[19] Response Time
[20] Throughput

ترکیبی از ویژگی‌های یاد شده که سرعت اجرا در آن‌ها دخیل است، اطلاق می‌شود.

۳- امنیت[۲۱] (q_{sec}) : یک ویژگی ترکیبی است که قابلیت پیاده سازی شناسایی[۲۲]، احراز[۲۳] و تصدیق[۲۴] هویت، مخفی بودن[۲۵]، عدم توانایی انکار[۲۶]، مقاومت[۲۷] در مقابل انواع مشخصی از حملات رمزگشایی[۲۸] و دیگر جنبه‌های مرتبط با امنیت، را نگاشت می‌کند. ورود بیشتر به مقوله امنیت از حوزه این تحقیق خارج است.

۴- شهرت[۲۹] (q_{re}) : معیاری برای صداقت فراهم کنندگان سرویس است. معمولا شهرت یک فراهم کننده یا سرویس، به اظهار نظرات و یادداشت‌های کاربران نهایی وابسته است. شهرت می‌تواند دلیلی بر باور کردن ادعاهای فراهم کننده سرویس در هنگاه تبلیغ[۳۰] یک سرویس باشد.

۵- نرخ اجرای موفق[۳۱] (q_{suc}) : برگرداننده این احتمال است که یک درخواست برای یک سرویس خاص، حد اکثر در یک زمان مورد انتظار، به صورت موفقیت آمیز پاسخ بگیرد.

۶- در دسترس بودن[۳۲] (q_{av}) : به میزان احتمال اینکه یک سرویس برای به خدمت گرفتن قابل دستیابی باشد، دسترس پذیری یک سرویس اطلاق می‌گردد.

فرض کنید که هر جنبه کیفیت سرویس برای هر سرویس می‌تواند مقداری محدود به یک بازه مشخص را به خود بگیرد. بنابراین سرویس S_i از نظر کیفیت سرویس می‌تواند با بردار شش تایی

$$QOS(S_i) = (q_{c}^{S_i}, q_{pref}^{S_i}, q_{sec}^{S_i}, q_{re}^{S_i}, q_{suc}^{S_i}, q_{av}^{S_i})$$

نمایش داده شود. این بردار را بردار شاخص کیفیت برای سرویس Si می‌نامیم.

مصرف کننده سرویس، نیازمندی‌های کیفیتی خود را با دو عدد کمترین و بیشترین مقدار قابل قبول برای هر یک از جنبه‌های کیفیت، مشخص می‌کند. هم چنین بر اساس اهمیت هر جنبه کیفیت برای مصرف کننده، به آن جنبه یک وزن می‌دهد.

از میان سرویس‌های در دسترس که از نظر عملکردی، نیاز مصرف کننده را فراهم می‌کنند، آن سرویسی برای مصرف کننده از همه مناسب‌تر است که اولا مقادیر بردار شش تایی شاخص کیفیت آن از بازه متناظر بیان شده توسط مصرف کننده بیرون نباشد و ثانیا حاصل جمع حاصل ضرب‌های مقدار هر جنبه کیفیتی در وزن اهمیت آن جنبه برای مصرف کننده، از بقیه بیشتر باشد.

در این روش عامل انسانی درخواست کننده سرویس، علاوه بر توصیف سرویس درخواستی به صورت مجموعه ورودی‌های در دسترس و خروجی‌های مورد نیاز، برای بیان نیازمندی‌های غیر تابعی سرویس درخواستی، باید یک توصیف کیفیتی مانند آنچه در بالا تشریح شد، در اختیار عامل نرم افزاری سازنده زنجیره ترکیب قرار دهد. این توصیف برای هر جنبه کیفی شامل یک مقدار بیشینه و یک مقدار کمینه خواهد بود. هم چنین بر اساس اهمیت هر جنبه کیفی، توصیف یاد شده دارای یک وزن برای هر جنبه خواهد بود. به عبارت دیگر، درخواست کننده سرویس برای عامل نرم افزاری باید سه بردار شاخص، یکی برای مقادیر بیشینه، یکی برای مقادیر کمینه و یکی هم برای ضرایب اهمیت تولید کند.

عامل نرم افزاری در هنگام کاندید کردن وب سرویس‌ها برای ورود به زنجیره ترکیب، باید توجه داشته باشد که اولا برای هر جنبه، مقادیر کیفی منتسب به وب سرویس‌های کاندید از بازه تعریف شده در توصیف خارج نباشد. یعنی برای کاندید کردن یک وب سرویس، باید مقادیر بردار شاخص کیفیت مربوط به آن از مقادیر متناظر در بردار بیشینه کمتر و از مقادیر متناظر در بردار کمینه بیشتر باشد. ثانیا در هنگام اولویت بندی میان سرویس‌های کاندید، مطابق با شاخص کیفیتی بیان شده برای سرویس درخواستی، آن سرویسی که حاصل جمع حاصلضرب‌های مقدار هر جنبه کیفی در وزن اهمیت آن جنبه از بقیه بیشتر باشد شانس بیشتری برای قرارگیری در نقش مورد نظر را داشته باشد. به این منظور باید برای تمامی وب سرویس‌های کاندید، ضرب داخلی بردار شاخص آن سرویس در بردار ضرایب اهمیت محاسبه شود. سپس می‌توان برای گزینش، سرویس‌های کاندید را بر اساس مقادیر بدست آمده به صورت نزولی مرتب کرد (یعنی به سرویسی که مقدار ضرب داخلی بیشتری دارد، شانس بیشتری برای انتخاب داده می‌شود).

لازم به ذکر است در روش بالا وجود یک سیستم ارزشگذاری واحد برای انتساب ارزش‌های کمّی به جنبه‌های کیفی هر وب سرویس ضروری است. در دنیای مهندسی نرم افزار، هریک از خواص کیفیتی یاد شده یک تعریف رسمی مخصوص به خود دارد [27] که در آن چگونگی نگاشت آن کیفیت از نرم افزار به مقدار عددی نیز دیده شده‌است. در دنیای وب سرویس‌ها بهترین منبع برای نگهداری مقادیر منتسب به خواص کیفی یک وب سرویس، قرارداد سطح سرویس[۳۳] از توصیف WSDL آن وب سرویس است.

عامل نرم افزاری که برای انتخاب وب سرویس مناسب برای یک نقش نیاز به توجه به خواص کیفیتی وب سرویس‌های در دسترس دارد، می‌تواند این خواص را در توصیف WSDL سرویس‌ها پیدا کرده و با انجام عملیات مقیاس سازی[۳۴] آن را به مقادیر یکسان و معتبر در بازه مورد نظر تبدیل کند.

---

21 Security
22 Identification
23 Authorization
24 Authentication
25 Secrecy
26 Non-Repudiation
27 Resistance
28 Cryptanalytic attacks
29 Reputation
30 Advertise
31 Successful Execution Rate
32 Availability
33 Service Level Agreement (SLA)
34 Scaling

در روش ترکیب سرویس مورد نظر این پژوهش برای ارزش گذاری مسیرها در درخت جستجو، باید هر یک از وب سرویس‌های دخیل در مسیر جستجو به صورت مجزا نسبت به سرویس درخواستی ارزش گذاری شود و حاصلجمع مقادیر بدست آمده، ارزش کل مسیر را نشان می دهد.

۳- راه‌کار پیشنهادی

در این قسمت راهکاری برای ترکیب خودکار و آگاه از کیفیت سرویس‌ها با مدلسازی به صورت مساله پوشش مجموعه[35] و استفاده از الگوریتم تقریبی[36] پیشنهاد می‌گردد.

نقاط تمایز و نو آوری، و بیشترین تمرکز روش پیشنهادی را می توان در موارد زیر خلاصه کرد:

۱. امکان ترکیب خورکار زنجیره‌ای و موازی سرویس‌ها به صورت توأم (اکثر روش‌های موجود برای ترکیب خودکار سرویس‌ها تنها به ترکیب زنجیره‌ای پرداخته‌اند [8]–[6]، و در برخی موارد به ترکیب موازی سرویس‌های وب پرداخته‌اند [10] ,[9])
۲. مدلسازی ترکیب خودکار سرویس‌های وب به صورت مساله پوشش مجموعه
۳. تولید ترکیب آگاه از کیفیت سرویس بر اساس ویژگی‌های کیفیتی بیان شده در درخواست
۴. خودکاری، پویایی و توزیع شدگی روش ترکیب
۵. تولید ترکیب نزدیک به بهینه در زمان معقول
۶. اطمینان از کامل بودن روش ترکیب

روش پیشنهادی، ترکیب سرویس‌های وب را در دو فاز اصلی طراحی می‌کند:

۱- تولید بالا به پایین درخت ترکیب
۲- تولید پایین به بالای طرح سرویس ترکیبی

۳-۱- تولید بالا به پایین درخت ترکیب

در این پژوهش برای مدلسازی ترکیب زنجیره ای، تولید یک درخت (مشابه تولید درخت در حل مساله با جستجو) پیشنهاد می‌شود. این درخت از خروجی یا پس شرط‌های سرویس درخواستی به سمت ورودی یا پیش شرط‌های آن تولید می‌شود. هر گره یا حالت از درخت مجموعه‌ای از ورودی و خروجی‌ها یا شرایط است و سرویس‌ها به عنوان یال‌های درخت امکان انتقال از یک گره به گره دیگر را فراهم می‌کنند.

ریشه درخت: خروجی‌های سرویس درخواستی محتوای گره ریشه است.

تابع جانشین: تابع جانشین، یک گره را به عنوان ورودی تابع می گیرد و تمامی وب سرویس‌هایی که در همان زمان در دسترس هستند و خروجی آن‌ها شامل قسمتی از محتوای گرهی است که در حال گسترش است، را می یابد. تابع جانشین مجموعه ورودی هر یک از سرویس‌هایی را که یافته است به عنوان گره‌های تولید شده باز می گرداند

آزمون خاتمه تولید گره در یک شاخه: تولید درختان در هر شاخه تا جایی ادامه پیدا می‌کند که به گرهی برسیم که محتوای آن زیر مجموعه‌ای از ورودی یا پیش شرط‌های سرویس درخواستی باشد ویا اینکه محدودیت عمق یا وزن مسیر نقض شود. محدودیت عمق یا وزن مسیر بسته به محدودیت زمانی ترکیب یا محدودیت هزینه سرویس‌ها توسط درخواست می‌تواند تعیین شود.

هزینه سرویس درخواستی: هزینه سرویس درخواستی یک فاکتور مرکب از خواص کیفیتی است که برای سرویس درخواستی و مسیرهای معتبر در درخت‌ها قابل محاسبه است. نگاشت خواص کیفیت سرویس به هزینه بر اساس آنچه در قسمت ۲-۵ گفته شد انجام می‌گیرد. تابع هزینه، بردار خواص کیفیتی را به عنوان ورودی دریافت کرده و با محاسبه شباهت آن با بردار کیفیت مرجع، آن را به یک عدد نگاشت می‌کند.

۳-۲- تولید پایین به بالای طرح ترکیب

رسیدن از برگ‌های درخت تولید شده به ریشه درخت معادل رسیدن به خروجی سرویس ترکیبی است و مسیر رسیدن به ریشه، سرویس‌های دخیل و طرح ترکیب را مشخص می‌کند. برگ‌هایی از درخت که محتوای آن‌ها زیر مجموعه ورودی سرویس درخواستی است قابل دسترس هستند. برای رسیدن به هرگره غیر برگ از درخت باید یک مساله پوشش مجموعه به صورت زیر تولید و حل شود.

اجزای مساله پوشش مجموعه به صورت زیر تعریف می‌شود:

مجموعه U : شامل محتوای گره جاری خواهد بود.

مجموعه S : به ازای هر فرزند از گره جاری یک مجموعه در S قرار خواهد گرفت که اعضای آن مجموعه، اشتراک مجموعه U با خروجی سرویس شاخه جاری خواهد بود.

مدلسازی وزن مجموعه‌ها: وزن هر مجموعه برابر حاصل جمع هزینه رسیدن به گره فرزند، و سرویس متناظر با گذار از فرزند به گره جاری خواهد بود. یادآوری می‌شود که هزینه هر سرویس بوسیله محاسبه شباهت خواص کیفیتی آن سرویس با سرویس درخواستی (آنگونه در بخش ۵-۲ گفته شد) محاسبه می‌گردد.

بدین ترتیب با حل مساله پوشش مجموعه فوق، هزینه و طرح رسیدن به گره جاری محاسبه می‌گردد. اگر تولید و حل مساله پوشش مجموعه به صورت سلسله مراتبی تا رسیدن به گره ریشه ادامه پیدا کند، طرح و هزینه سرویس ترکیبی مورد تقاضا تولید خواهد شد.

۳-۳- مساله نمونه ترکیب سرویس‌های وب

برای روشن شدن روش، یک نمونه بسیار ساده و قابل نمایش از مساله ترکیب سرویس‌های وب در ادامه توصیف و حل می‌شود.

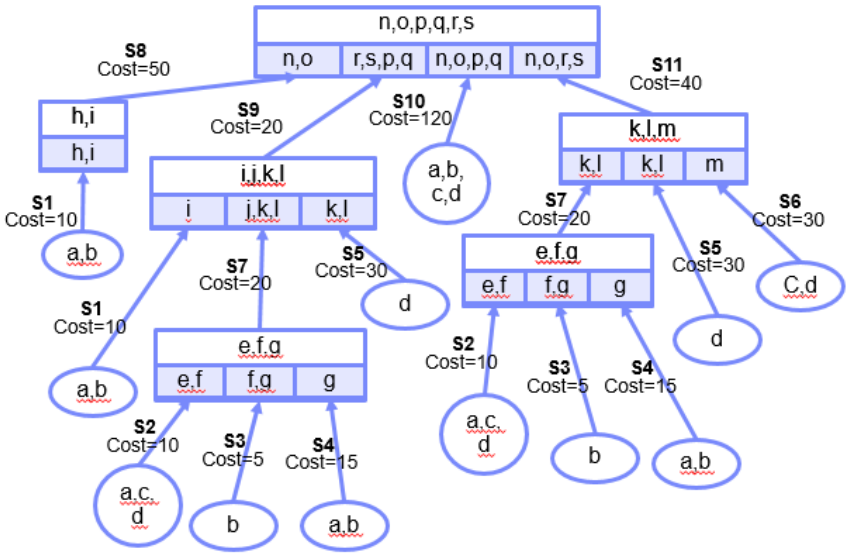
در شکل 4 فضای مساله شامل سرویس‌های در دسترس و درخواست سرویس ترکیبی توصیف شده‌است.

[35] Minimum Set Cover

[36] Approximation algorithms

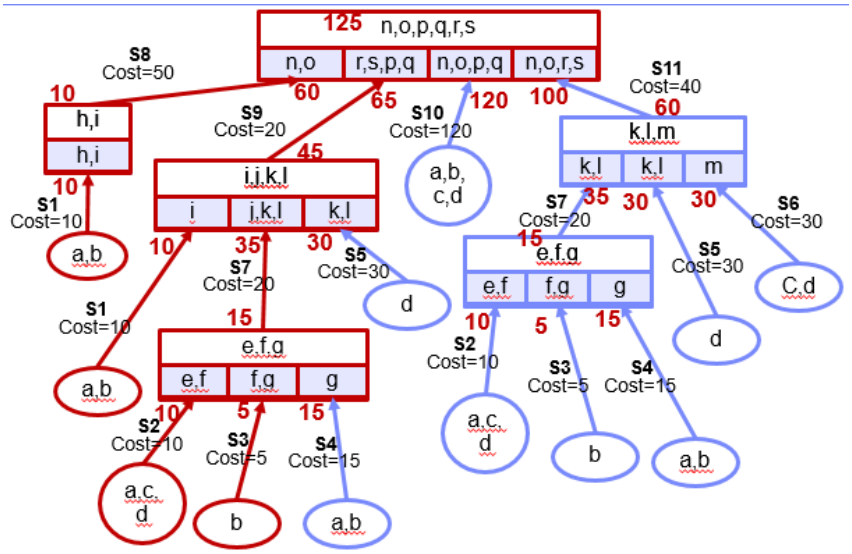
شکل ۴ : فضای مساله نمونه

نتیجه فاز اول روش یعنی تولید بالا به پایین درخت ترکیب در شکل ۵ آمده است.



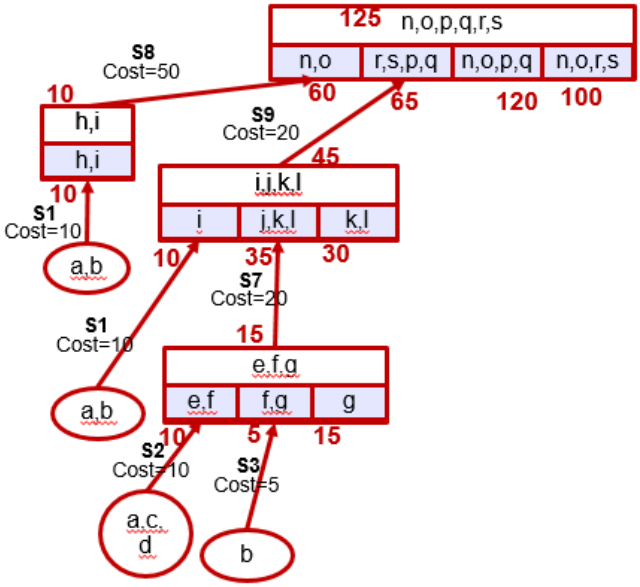
شکل ۵: درخت ترکیب مساله نمونه

اجرای فاز دوم روش یعنی تولید پایین به بالای طرح ترکیب در شکل ۶ آمده است. تولید و اجرای سلسله مراتبی مسائل پوشش مجموعه و مقدار محاسبه شده به عنوان هزینه رسیدن به هر گره در شکل مشخص شده‌است.



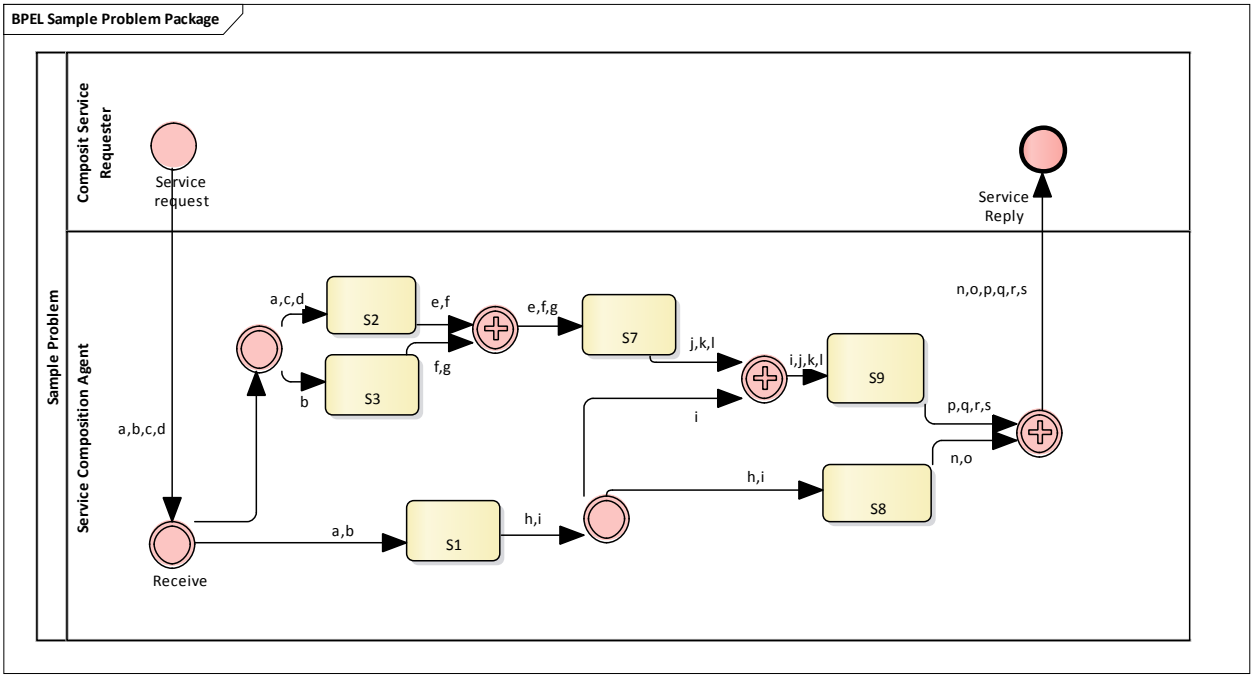
شکل ۶ : تولید و اجرای سلسله مراتبی مسائل پوشش مجموعه در مساله نمونه

در شکل ۷ پس از حذف شاخه‌های استفاده نشده، طرح نهایی ترکیب سرویس درخواستی نمایش داده شده‌است.



شکل ۷ : طرح ترکیب سرویس مسئله نمونه

در شکل ۸ نمودار BPMN سرویس مرکب تولید شده برای مساله نمونه ترسیم شده‌است.



شکل 8 : نمودار BPMN سرویس مرکب تولید شده

۳-۴- ملاحضات مهم

۱. پژوهش حاضر در مرحله ارایه پیشنهاده دکتری می باشد و به همین دلیل اطلاعات پیاده سازی و آزمون روش در آینده، در مقاله بعدی منتشر خواهد شد.

۲. درخت ترکیب تولید شده در روش پیشنهادی، در تطبیق پذیری سرویس ترکیبی (راهکار طراحی مجدد جزیی[37] و تعویض سرویس[38]) هم می‌تواند به کار گرفته شود.

۳. طراحی و استفاده از راهکار مناسبی برای هرس کردن درخت ضروری به نظر می‌رسد.

۴. محاسبه مقدار بعضی از خواص کیفتی بردار کیفیت (مثلا زمان پاسخ) در حالت اتصال دو سرویس به صورت متوالی و موازی با هم متفاوت است.

۵. برای جلوگیری از تکرار در گسترش گره‌های درخت استفاده از مفهوم لیست باز و لیست بسته ضروری است.

۶. تولید و استفاده از یک ابتکار[39] مناسب برای اولویت دادن به گره‌های لیست باز و گسترش هدف گرای درخت ضروری به نظر می‌رسد.

۷. تولید درخت ترکیب می‌تواند به صورت عمیق کننده تکراری صورت پذیرد.

۸. چند بار به خدمت گرفتن یک سرویس در چند درخت متفاوت منجر به تاثیر هزینه‌ی این سرویس در همه مسیرهای ذکر شده می‌شود حال آن که در واقعیت تنها یک بار به خدمت گرفته خواهد شد (این محدودیت در اکثر روش‌های پیشین نیز دیده می‌شود).

۹. ممکن است هنگامی که یک پیاده سازی از یک وب سرویس انتخاب شود لازم باشد حتما یک پیاده سازی مشخص از یک وب سرویس دیگر نیز انتخاب شود، به این حالت محدودیت وابستگی[40] می‌گویند. گاهی اوقات انتخاب یک پیاده سازی از یک سرویس امکان انتخاب برخی از پیاده سازی‌ها از سرویس دیگر را از بین می‌برد. این حالت محدودیت تداخل[41] است. به خدمت گرفتن بعضی از سرویس‌ها می‌تواند روی هزینه سرویس‌های دیگر موثر باشد. به این حالت هزینه آبشاری[42] گفته می‌شود. روش پیشنهادی این قبیل محدودیت‌ها را در نظر نمی‌گیرد.

۱۰. پیاده سازی روش می‌تواند به صورت همکاری چند عامل انجام گیرد. راهبرد توزیع، تقسیم کردن داده[43] است. هر عامل باید به دنبال تولید درخت برای قسمتی از خروجی سرویس درخواستی باشد. هماهنگ ساز[44] وظیفه قسمت کردن خروجی، جمع آوری مسیرهای بهینه و حل مساله پوشش مجموعه در بالاترین سطح را دارد.

۱۱. نیازی به تجمیع اطلاعات سرویس‌های در دسترس قبل از عملیات ترکیب نیست و عملیات ترکیب (شامل تولید بالا به پایین درخت ترکیب و تولید پایین به بالای طرح سرویس ترکیبی) به صورت پویا انجام می‌گیرد.

[37] Partial Re-Planning
[38] Service Exchange
[39] Heuristic
[40] Dependency constraint
[41] Conflict constraint
[42] Cascading cost
[43] Data partitioning
[44] Coordinator

۳-۵- ارزیابی راه‌کار پیشنهادی

ارزیابی روش پیشنهادی شامل دو دیدگاه ارزیابی تئوری و ارزیابی بر اساس آزمایش خواهد بود.

ارزیابی تئوری: ازریابی الگوریتم‌های پیشنهادی برای جمع آوری و پالایش سرویس‌ها، تولید ساختارهای ترکیب ایستا و پویا با روش محاسبه پوشش مجموعه مشخص بصورت تحلیل‌های زیر انجام می‌شوند:

- مرتبه زمانی و مکانی ترکیب قابل محاسبه است.
- کامل بودن روش (یعنی اطمینان از یافتن راه حل توسط روش در صورت وجود) قابل بررسی است.
- فاکتور تقریب، به عنوان معیار نزدیکی ترکیب تولید شده به ترکیب بهینه قابل محاسبه است.

ارزیابی بر اساس آزمایش: مجموعه داده Web Service Challenge 2005 (WSC05) [28] مورد استفاده قرار خواهد گرفت. تا هنگام نگارش این مقاله، نسخه سال ۲۰۰۵ از مجموعه داده یادشده در دسترس بوده‌است. در صورت دریافت نسخه جدید، آن مجموعه داده جایگزین نسخه قدیمی‌تر خواهد شد.

متغیرهای مستقل آزمایش می‌تواند شامل موارد زیر باشد:

۱- تعداد وب سرویس‌های موجود (اندازه فضای مساله)
۲- تعداد نیازمندی‌های عملکردی سرویس درخواستی (اندازه خروجی)
۳- عمق ترکیب (حداکثر تعداد سرویسی که مجاز هستند در ترکیب پشت سر هم فراخوانی شوند)
۴- روش حل مساله پوشش مجموعه

متغیرهای وابسته آزمایش می‌تواند شامل موارد زیر باشد:

۱- سرعت تولید راه حل (کارایی)
۲- نسبت تعداد آزمایش موفق به تعداد کل آزمایش‌ها (نرخ شکست)
۳- نزدیکی راه حل تولید شده به راه حل بهینه

سپاسگزاری

بدینوسیله از آزمایشگاه ارزیابی کیفی نرم افزار گروه کامپیوتر دانشکده مهندسی دانشگاه فردوسی مشهد که پژوهش پیش رو در آن در حال انجام است، قدر دانی می شود.

مراجع

[1] ف. شمس، ا. مهجوریان, *معرفی اصول، مبانی و روش‌های معماری سازمانی سرویس‌گرا*. انتشارات دانشگاه شهید بهشتی, ۱۳۸۹.

[2] م. خانی دهنویی, "ارایه ی روشی پویا و توزیع شده برای ترکیب خودکار وب سرویس ها با استفاده ازالگوریتم های جستجوی آگاهانه," دانشگاه اصفهان, ۱۳۸۹.

[3] J. Rao and X. Su, "A Survey of Automated Web Service Composition Methods," in *Proceedings of the First international conference on Semantic Web Services and Web Process Composition*, Springer-Verlag, 2005, pp. 43–54.

[4] G. Baryannis and D. Plexousakis, "Automated Web Service Composition : State of the Art and Research Challenges," *ICS-FORTH, Tech. Rep*, no. October, 2010.

[5] "Web Service Choreography Interface (WSCI) 1.0." [Online]. Available: https://www.w3.org/TR/wsci/. [Accessed: 01-May-2017].

[6] Y. Chen, J. Huang, and C. Lin, "Partial Selection: An Efficient Approach for QoS-Aware Web Service Composition," in *2014 IEEE International Conference on Web Services*, 2014, pp. 1–8.

[7] Q. Wu and F. Ishikawa, "Towards Service Skyline for Multi-granularity Service Composition," in *Proceedings of the 2014 International Workshop on Web Intelligence and Smart Sensing - IWWISS '14*, 2014, pp. 1–6.

[8] M. Suresh Kumar and P. Varalakshmi, "A Novel Approach for Dynamic Web Service Composition through Network Analysis with Backtracking," in *Advances in Computing and Information Technology*, 2013, pp. 357–365.

[9] S. R. Ponnekanti and A. Fox, "SWORD : A Developer Toolkit for Web Service Composition," *Proc. Elev. Int. World Wide Web Conf.*, vol. 45, pp. 1–23, 2009.

[10] Y. Yao and H. Chen, "A Rule-Based Web Service Composition Approach," in *2010 Sixth International Conference on Autonomic and Autonomous Systems*, 2010, pp. 150–155.

[11] "Set cover problem." [Online]. Available: https://en.wikipedia.org/wiki/Set_cover_problem. [Accessed: 16-Jul-2017].

[12] D. S. Johnson, "Approximation algorithms for combinatorial problems," *J. Comput. Syst. Sci.*, vol. 9, no. 3, pp. 256–278, Dec. 1974.

[13] J. . Beasley and P. . Chu, "A genetic algorithm for the set covering problem," *Eur. J. Oper. Res.*, vol. 94, no. 2, pp. 392–404, Oct. 1996.

[14] R. Jovanovic and M. Tuba, "An ant colony optimization algorithm with improved pheromone correction strategy for the minimum weight vertex cover problem," *Appl. Soft Comput.*, vol. 11, no. 8, pp. 5360–5366, Dec. 2011.

[15] P. Tague, Jooyoung Lee, and R. Poovendran, "A Set-Covering Approach for Modeling Attacks on Key Predistribution in Wireless Sensor Networks," in *2005 3rd International Conference on Intelligent Sensing and Information Processing*, 2005, pp. 254–259.

[16] H. K. Rajagopalan, C. Saydam, and J. Xiao, "A multiperiod set covering location model for dynamic redeployment of ambulances," *Comput. Oper. Res.*, vol. 35, no. 3, pp. 814–826, Mar. 2008.

[17] T. WANG and D. LIU, “Scheduling algorithm for AND/OR precedence constrained tasks based on solving the set-cover problem,” *J. Harbin Eng. Univ.*, vol. 5, 2007.

[18] S. C. Geyik, B. K. Szymanski, P. Zerfos, and D. Verma, “Dynamic Composition of Services in Sensor Networks,” in *2010 IEEE International Conference on Services Computing*, 2010, no. Scc, pp. 242–249.

[19] V. Gabrel, M. Manouvrier, K. Moreau, and C. Murat, “QoS-aware automatic syntactic service composition problem: Complexity and resolution,” *Futur. Gener. Comput. Syst.*, Apr. 2017.

[20] H. Wang, B. Zou, G. Guo, D. Yang, and J. Zhang, “Integrating Trust with User Preference for Effective Web Service Composition,” *IEEE Trans. Serv. Comput.*, vol. 10, no. 4, pp. 574–588, Jul. 2017.

[21] V. Tewari, U. Thakar, and N. Dagdee, “Classifying Web Services based on QoS Parameters using Extended Dataset,” *Int. J. Comput. Appl.*, vol. 74, no. 8, pp. 33–36, Jul. 2013.

[22] T. Rajendran, P. Balasubramanie, and R. Cherian, “An efficient {WS-QoS} broker based architecture for web services selection,” *Int. J. Comput. Appl.*, vol. 1, no. 9, pp. 79–84, 2010.

[23] M. Rathore, M. Rathore, and U. Suman, “A quality of service broker based process model for dynamic web service composition,” *Proc. 3RD Int. Work. Model. Enterp. Inf. Syst. (EIS’ 07*, pp. 1267--1274, 2011.

[24] S. Chattopadhyay, A. Banerjee, and N. Banerjee, “A Fast and Scalable Mechanism for Web Service Composition,” *ACM Trans. Web*, vol. 11, no. 4, pp. 1–36, Aug. 2017.

[25] A. D’Ambrogio and Andrea, “A Model-driven WSDL Extension for Describing the QoS ofWeb Services,” in *2006 IEEE International Conference on Web Services (ICWS’06)*, 2006, pp. 789–796.

[26] A. D’Ambrogio, “Model-Driven Quality Engineering of Service-Based Systems,” Springer, Berlin, Heidelberg, 2010, pp. 81–103.

[27] “ISO/IEC 13236:1998 Information technology - Quality of service: Framework.” 1998.

[28] “The 2005 IEEE International Conference on e-Technology, e-Commerce and e-Service (EEE-05).” [Online]. Available: http://www.comp.hkbu.edu.hk/~eee05/contest/. [Accessed: 09-Aug-2017].