



دانشگاه کاشان
University of Kashan

مجله محاسبات نرم

SOFT COMPUTING JOURNAL

تارنمای مجله: scj.kashanu.ac.ir



روشی برای ساده‌سازی الگوها در ترکیب وب‌سرویس‌ها و انتخاب بهینه ترکیب با ساختار احتمالی[✦]

نرجس ظهیری^۱، دانشجوی کارشناسی ارشد، سید مرتضی بابامیر^{۱*}، دانشیار
^۱ دانشکده برق و کامپیوتر، گروه مهندسی کامپیوتر، دانشگاه کاشان، کاشان، ایران.

اطلاعات مقاله

چکیده

تاریخچه مقاله:

دریافت ۱۲ خرداد ماه ۱۴۰۰
پذیرش ۲۷ مرداد ماه ۱۴۰۰

کلمات کلیدی:

ترکیب و انتخاب وب‌سرویس‌ها
وب‌سرویس‌های آگاه به کیفیت
ساختارهای پیچیده احتمالی
ساده‌سازی گراف
الگوریتم‌های تکاملی
الگوریتم NSGAI
الگوریتم SPEAI

یکی از چالش‌برانگیزترین موضوعات مرتبط با وب‌سرویس‌ها مساله ترکیب آن‌هاست که به صورت یک گراف برای نمایش تعامل بین سرویس‌ها ارائه می‌شود. هر گره از این گراف، یک وب‌سرویس انتزاعی با وظیفه مشخص اما ویژگی‌های کیفی نامشخص است. برای هر سرویس انتزاعی، مجموعه‌ای از سرویس‌های کاندید با وظیفه یکسان اما ویژگی‌های کیفی متفاوت وجود دارد. جایگزینی یک وب‌سرویس کاندید برای هر سرویس انتزاعی به نحوی که یک ترکیب بهینه حاصل شود، یک مساله NP-hard است، لذا برای حل آن از الگوریتم‌های هیوریستیک استفاده می‌شود. تاکنون روش‌های متعددی برای ترکیب بهینه وب‌سرویس‌ها ارائه شده اما بیشتر این روش‌ها ساختار احتمالی را پشتیبانی نمی‌کنند. تنها یک روش ساختار احتمالی را پشتیبانی می‌کند که برای گراف‌های بزرگ مقیاس پذیر نیست، مبتنی بر قیود است و به تحلیل هر مسیر گراف به صورت جداگانه پرداخته است. این مقاله، رویکردی مقیاس پذیر و چندهدفه برای تحلیل گراف به صورت یکپارچه ارائه می‌دهد که علاوه بر پوشش دو الگوی جدید حلقه‌های تودرتو و حلقه‌های موازی، با ارائه روشی جهت ساده‌سازی ترکیب وب‌سرویس‌ها موجب بهبود کارایی نیز می‌شود. در این روش برای انتخاب بهینه وب‌سرویس‌ها و حفظ مقیاس پذیری، از الگوریتم‌های تکاملی NSGAI و SPEAI استفاده می‌شود. در روش پیشنهادی ابتدا در گراف‌هایی با الگوی شرطی، هر مسیر بر حسب احتمال آن، چند بار تکرار می‌شود و سپس از الگوریتم NSGAI به منظور تعیین بهترین مسیر در گراف و یافتن راه‌حل‌های بهتر استفاده می‌شود. روش پیشنهادی در مقایسه با بهترین روش مقایسه شده، ۳۰ درصد در پارامتر قابلیت اطمینان و ۱۲۱ میلی ثانیه در زمان پاسخ بهبود داشته است.

© ۱۴۰۰ - مجله محاسبات نرم، کلیه حقوق محفوظ است.

۱. مقدمه

سیستم‌های سرویس‌گرا بازی می‌کند [۱، ۲]. هر وب‌سرویس با عملکرد خاص نیازمند ارتباط با وب‌سرویس‌های دیگر است تا برای پاسخ‌گویی به نیازهای ترکیبی و پیچیده کاربران مناسب باشد [۳]. هم‌چنین تعداد وب‌سرویس‌هایی که دارای عملکرد یکسان ولی کیفیت متفاوت می‌باشند، افزایش یافته است، از این رو انتخاب و ترکیب وب‌سرویس‌ها به طوری که بهترین کارایی را داشته باشند جزء مباحث مهم و پیچیده در این زمینه به شمار

در سال‌های اخیر وب‌سرویس به عنوان یک مدل محاسباتی به سرعت توسعه یافته است و نقش بسیار مهمی در تجارت و

✦ نوع مقاله: پژوهشی

* نویسنده مسئول

پست(های) الکترونیک: nargess.zahiri@gmail.com (ظهیری)

babamir@kashanu.ac.ir (بابامیر)

می‌رود [۴].

در گراف‌های از وب‌سرویس‌ها، برای هر سرویس (که یک سرویس انتزاعی نامیده می‌شود)، تعدادی سرویس کاندید با عملکرد یکسان ولی خصیصه‌های کیفی متفاوت وجود دارد. مساله مورد نظر انتخاب سرویس کاندید برای هر سرویس انتزاعی در گراف‌های دارای الگوهای احتمالی است. برای حل این مسئله، تلاش می‌شود تا گراف به: (۱) یک گره یا (۲) تعدادی از مسیرهای جدا از هم که تشکیل یک درخت را می‌دهد، ساده شود [۵، ۶]. این ساده‌سازی بستگی به: (۱) نوع الگوی گراف (توالی، توازی، حلقه، شرطی)، (۲) احتمالی بودن یا نبودن الگوها، (۳) کیفیت سرویس‌های کاندید دارد، بنابراین ساده‌سازی گراف همراه با جایگزینی سرویس‌های کاندید برای سرویس‌های انتزاعی انجام می‌شود. با توجه به این‌که برای هر سرویس انتزاعی می‌تواند بیش از یک سرویس کاندید وجود داشته باشد، ساده‌سازی‌های متعددی با ساختارهای یکسان ولی کاندیدهای متفاوتی متصور است. روش‌های ساده‌سازی گراف به سه دسته تقسیم می‌شوند: (۱) روش تجمعی^۱ که در این روش با حذف الگوهای توالی، موازی، شرطی و حلقه در آخر به یک گره می‌رسیم. (۲) روش مبتنی بر مسیر^۲ که تمام مسیرهای موجود بین دو گره‌ی ابتدا و انتهای گراف را استخراج می‌کند. (۳) روش ترکیبی^۳ [۵] که ترکیبی از دو روش ذکر شده است. این مقاله از روش سوم بهره گرفته است. همان‌طور که گفته شد ساده‌سازی گراف بستگی به کیفیت سرویس‌های کاندید نیز دارد، بنابراین در ادامه به ویژگی‌های کیفی وب‌سرویس‌ها و مساله انتخاب بهینه آنها می‌پردازیم.

کیفیت وب‌سرویس‌ها شامل ویژگی‌های غیرعملیاتی مانند هزینه اجرا^۴، زمان اجرا^۵، میزان در دسترس بودن^۶، نرخ موفقیت اجرا^۷ و امنیت^۸ وب‌سرویس می‌باشد. ویژگی‌های کیفی وب‌سرویس

به دو دسته طبقه‌بندی می‌شوند. دسته‌ای از این ویژگی‌ها منفی می‌باشند که باید کمینه شوند مانند زمان پاسخ و هزینه. دسته دوم خصوصیات هستند که مثبت هستند که باید بیشینه شوند مانند توان عملیاتی^۹ و دسترس پذیری [۷]. جهت انتخاب بهینه وب‌سرویس‌ها پارامترهای کیفی ذکر شده نقش مهمی در شناسایی بهترین ترکیب وب‌سرویس‌ها در زمان اجرا بازی می‌کنند. یافتن یک راه‌حل بهینه برای مساله ترکیب وب‌سرویس‌ها با اهداف متضاد یک مساله پیچیده است و در زمان چند جمله‌ای قابل حل نیست [۸]. بنابراین برای حل این‌گونه مسائل از الگوریتم‌های تکاملی استفاده می‌شود.

در رویکردهای ارائه شده در مقالات [۹-۱۵، ۳۰-۳۲] گرچه برای انتخاب بهینه وب‌سرویس‌های کاندید، راه‌حلی ارائه می‌دهند اما: (۱) تنها ساختار توالی را در گراف ترکیب وب‌سرویس‌ها در نظر می‌گیرند و در نتیجه ساختارهای احتمالی را پشتیبانی نمی‌کنند و (۲) از روش تجمعی (node-based) استفاده می‌کنند. به طور مشابه، اگر چه در رویکردهای ارائه شده در [۱۶، ۱۷] و [۱۸] گراف ترکیب وب‌سرویس‌ها، به ترتیب ساختارهای توالی، موازی و ساختارهای توالی، موازی و حلقه را در نظر می‌گیرند اما دو مشکل ذکر شده برای موارد بالا را دارند. در رویکردهای ارائه شده در [۱۹-۲۵]، اگر چه در گراف ترکیب وب‌سرویس‌ها، انواع الگوها را در نظر می‌گیرند و الگوی شرطی احتمالی و حلقه‌های مکرر را پشتیبانی می‌کنند، اما (۱) روش تجمعی را برای ساده‌سازی ترکیب ارائه می‌دهند و (۲) ساختارهای موازی و حلقه احتمالی را پشتیبانی نمی‌کنند. همچنین [۲۰، ۲۲] فقط به بحث ترکیب پرداخته‌اند و الگوریتمی به منظور انتخاب کاندید ندارند. با توجه به این‌که کامل‌ترین رویکرد، در مقاله [۵] ارائه شده است، ما این رویکرد را به عنوان مبنای کار در نظر می‌گیریم. در این رویکرد، الگوهای پیچیده از جمله حلقه‌هایی با تعداد بیش از دو گره و انواع الگوهای شرطی پوشش داده می‌شوند و همچنین احتمال هر مسیر نیز محاسبه می‌شود. اما (۱) در این رویکرد، الگوی حلقه‌های تودرتو و موازی منظور نشده است، (۲) مساله، مساله

¹ Node-based

² Path-based

³ Hybrid-based

⁴ Cost

⁵ Execution time

⁶ Availability

⁷ Reliability

⁸ Security

⁹ Through put

نتایج روش ارائه شده پرداخته شده است. در بخش ۶ نتیجه‌گیری آمده است و در بخش ۷ پیشنهادها و کارهای آتی بیان شده است.

۲. مفاهیم اولیه

در این بخش به مفاهیم اولیه مربوط به انواع الگوها و همچنین الگوریتم‌های بهینه‌ساز چندهدفه چون NSGAI و SPEAI پرداخته شده است.

۲.۱. الگوها در ترکیب وب‌سرویس‌ها

در این زیربخش: (۱) تعاریف الگوهای توالی، موازی، شرطی و حلقه در ترکیب وب‌سرویس‌ها را ارائه می‌دهیم، (۲) روش محاسبه خصیصه کیفی هر الگو با استفاده از [۵] نشان داده می‌شود و (۳) با توجه به موارد (۱) و (۲) به شرح روش‌های ساده‌سازی الگوها می‌پردازیم. هر گره در این الگوها، معرف یک وب‌سرویس انتزاعی است که در ساده‌سازی با یک وب‌سرویس کاندید جایگزین می‌شود.

۲.۱.۱. تعاریف و شرح الگوها

یک گراف ترکیب وب‌سرویس به صورت $G = (V, E)$ تعریف می‌شود که در آن $V = \{v_1, \dots, v_n\}$ و $E = \{e_1, \dots, e_m\}$ به ترتیب مجموعه‌ای از رئوس و مجموعه‌ای از یال‌هاست. یال i ام که گره v_x را به گره v_y متصل می‌کند با تاپل $e_i = (v_x, v_y, p_{xy})$ نشان داده می‌شود که p_{xy} معرف احتمالی است که v_y می‌تواند به دنبال v_x بیاید. اگر از v_x به v_y یالی وجود نداشته باشد، $p_{xy} = 0$. درجه ورودی (تعداد یال‌های وارده) به راس v_x و درجه خروجی (تعداد یال‌های خارج‌شده) از این راس است. هر گراف ترکیب وب‌سرویس می‌تواند دارای الگوهای توالی، موازی، شرطی و حلقه باشد. برای بیان رسمی این الگوها، از ماتریس مجاورت دوبعدی A استفاده می‌کنیم که در آن، هر سطر و هر ستون معرف یک گره است.

• الگوی توالی

دو گره v_x و v_y را متوالی گویند اگر و فقط اگر این دو گره

ارضای قیود است و نه بهینه‌سازی، (۳) از الگوریتم غیر تکاملی به منظور انتخاب بهینه سرویس‌های کاندید استفاده شده، بنابراین از نظر زمانی مقیاس‌پذیر نیست، (۴) پس از ساده‌سازی و یافتن احتمال هر مسیر، به تحلیل هر مسیر به صورت مجزا پرداخته است که این موضوع نیز مانعی برای جواب قطعی به مشتری است. ما در این مقاله، به این ۴ مورد پرداخته‌ایم. کار انجام شده در این مقاله به طور خلاصه به دو بخش اصلی تقسیم می‌شود: در بخش اول به مساله ترکیب وب‌سرویس پرداخته شده است که علاوه بر شناسایی دو الگوی جدید، روشی جهت ساده‌سازی گراف ترکیب ارائه می‌شود که نتیجه آن پاسخ دقیق‌تر و قطعی‌تر به مشتریان خواهد بود. در این بخش پس از ساده‌سازی گراف با جایگزینی کاندیدهای متفاوت، جمعیتی تولید می‌شود. در بخش دوم از الگوریتم‌های بهینه‌ساز (SPEAI¹ و NSGAI²) به منظور یافتن راه‌حل‌های بهینه در این مساله استفاده شده است و در پایان، اولویت اعضای جبهه پرتو با استفاده از رویکرد تحلیل سلسله مراتبی، تعیین شود.

۱.۱. نوآوری‌های مقاله

نوآوری‌های این مقاله به شرح زیر هستند:

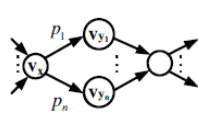
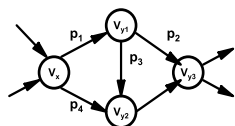
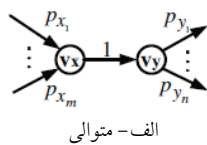
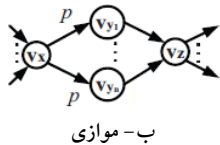
الف- پشتیبانی گراف از الگوهای موازی، شرط و حلقه و گذارهای احتمالی بین سرویس‌ها و همچنین معرفی دو الگوی جدید حلقه‌های موازی و تودرتو
ب- طرح یک روش ترکیبی مبتنی بر تکرار مسیر به منظور پاسخ‌گویی دقیق‌تر به مشتری
ج- استفاده از الگوریتم‌های بهینه‌ساز چندهدفه به منظور انتخاب بهینه وب‌سرویس‌ها

این مقاله شامل بخش‌های زیر است: در بخش ۲ مروری بر مفاهیم مربوط به الگوها در گراف ترکیب وب‌سرویس و همچنین الگوریتم‌های بهینه‌ساز چندهدفه شده است. در بخش ۳ مروری بر کارهای گذشته انجام شده است. روش پیشنهادی در بخش ۴ بیان شده است. در بخش ۵ به ارزیابی عملکرد و

¹ Strength Pareto Evolutionary Algorithm

² Non-dominated Sorting Genetic Algorithm

دارای یالی مشترک باشند و گره v_x تنها دارای یک خروجی و گره v_y تنها دارای یک ورودی باشد و هم‌چنین گره v_x ورودی از گره v_y نداشته باشد (این دو گره تنها یک یال مشترک داشته باشند، شکل (۱-الف)).

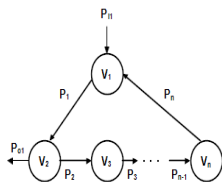
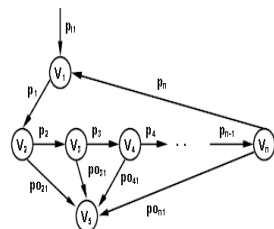


ب- موازی

الف- متوالی

د- شرطی غیر ساخت یافته

ج- شرطی ساخت یافته



و- حلقه با یک ورودی و چند خروجی

ه- حلقه با یک ورودی و یک خروجی

شکل (۱): انواع الگوها در گراف ترکیب وب‌سرویس

۲.۱.۲. محاسبه خصیصه کیفی الگوها

در این بخش، سه خصیصه کیفی دسترس‌پذیری، زمان پاسخ یک سرویس به یک تقاضا و هزینه استفاده از یک سرویس بیان و سپس محاسبه این خصیصه‌ها نشان داده می‌شود. علت انتخاب این سه خصیصه این است که عمومیت این ۳ خصیصه از بقیه بیشتر است. در دسترس بودن یا نبودن یک وب‌سرویس، زمانی که طول می‌کشد تا یک وب‌سرویس به تقاضای کاربر پاسخ دهد و هم‌چنین هزینه‌ای که کاربر باید بابت خدمت مورد نظر بپردازد از مهم‌ترین و اصلی‌ترین خصیصه‌های یک وب‌سرویس هستند که در مقالات [۵، ۶، ۱۲، ۱۳، ۱۵، ۱۶، ۲۰، ۲۳، ۳۱، ۳۲] حداقل یکی از این خصیصه‌ها استفاده شده و در مقالات [۹-۱۱، ۱۷-۱۹، ۲۱، ۲۲، ۲۴، ۳۰] هر سه خصیصه به کار رفته است. در این مقاله نیز هر سه خصیصه به صورت

دارای یالی مشترک باشند و گره v_x تنها دارای یک خروجی و گره v_y تنها دارای یک ورودی باشد و هم‌چنین گره v_x ورودی از گره v_y نداشته باشد (این دو گره تنها یک یال مشترک داشته باشند، شکل (۱-الف)).

• الگوی موازی

گره‌های v_{y1} تا v_{ym} را موازی گویند اگر و فقط اگر همگی یک ورودی مشترک با گذر احتمال یکسان و حداقل یک گره خروجی مشترک داشته باشند، در صورت داشتن خروجی غیر مشترک، آن خروجی نباید خود گره باشد، (شکل (۱-ب)).

• الگوی شرطی

به دو دسته شرطی ساخت یافته و شرطی غیرساخت یافته تقسیم می‌شوند.

شرطی ساخت یافته^۱: گره‌های v_{y1} تا v_{ym} را شرطی ساخت یافته گویند اگر و تنها اگر همگی گره ورودی مشترک با احتمال گذر غیریکسان داشته باشند و هم‌چنین هیچ دو مسیری قبل از رسیدن به یکدیگر وظایف یکسانی نداشته باشند و مسیرها کاملاً از یکدیگر مستقل باشند و همگی به یک گره خروجی مشترک برسند (شکل (۱-ج)).

الگوی شرطی غیرساخت یافته^۲: گره‌های v_{y1} و v_{y2} را شرطی غیرساخت یافته گویند اگر و تنها اگر همگی گره ورودی مشترک با احتمال گذر غیریکسان داشته باشند و هم‌چنین دو یا تعداد بیشتری مسیر قبل از رسیدن به یکدیگر وظایف یکسانی را به اشتراک بگذارند، برای مثال (شکل (۱-د)) که وظیفه v_{y2} هم از مسیری که از وظیفه v_{y1} می‌آید و هم از مسیری که از وظیفه v_x می‌آید به اشتراک گذاشته شده است.

• الگوی حلقه

گره‌های v_1 تا v_n تشکیل حلقه می‌دهند اگر و فقط اگر گره اولی با احتمالی به گره دومی و گره دومی با یک احتمالی به گره سومی،، گره $n-1$ با احتمالی به گره n ام و گره n ام با احتمالی به گره اول متصل باشد.

حلقه با یک ورودی و یک خروجی: حلقه‌ای که دارای یک گره

¹ Structured Conditional

² Unstructured Conditional

پاسخ به تقاضا با اجرای سرویس x و هزینه استفاده از سرویس x است. جدول (۱)، محاسبه این سه خصیصه برای الگوی ترتیبی با دو گره x و y ، الگوهای توازی با n گره y_1 تا y_n ، الگوی حلقه با n گره نشان می‌دهد. در رابطه الگوی حلقه، نماد p معرف احتمال یال‌ها در حلقه، k تعداد دورهای حلقه و n تعداد گره‌های حلقه است. جدول (۲) [۵]، احتمال یال‌های ورودی و خروجی را پس از ساده‌سازی هر الگو نشان می‌دهد. برای مقادیر m و n در الگوی ترتیبی شکل (۱)-الف را ببینید. در الگوی حلقه، z اندیس شماره یال‌های خروجی گره k ام حلقه است که از ۱ تا حداکثر m است. بر اساس جداول (۱) و (۲)، در [۵] خصیصه‌های کیفی و احتمال‌ها برای ۳ الگوی پایه توالی، توازی، حلقه ساده منظور شده است.

هم‌زمان در نظر گرفته شده است.
 ۱. دسترس‌پذیری: درصدی از زمان که یک سرویس مشغول به سرویس نیست و آزاد است.
 ۲. زمان پاسخ: فاصله بین زمانی که مشتری برای یک وب‌سرویس درخواست می‌دهد تا زمانی که وب‌سرویس خدمت مورد نظر را انجام می‌دهد را زمان پاسخ گویند. واحد آن بر حسب ثانیه (S) است.
 ۳. هزینه: مقدار پولی است که مشتری جهت استفاده از یک وب‌سرویس باید بپردازد و واحد آن دلار (\$) است.
 با توجه به تعاریف فوق، از روابطی که در [۵] برای محاسبه سه خصیصه کیفی برای سه الگوی ترتیبی، توازی و حلقه آمده است (جدول (۱)) بهره می‌بریم که در آن a_x ، t_x و c_x به ترتیب معرف خصیصه‌های کیفی دسترس‌پذیری سرویس x ، زمان

جدول (۱): محاسبه کیفیت در هر الگو [۵]

خصوصیت کیفی	ترتیبی (۲ گره)	موازی (۲ تا n گره)	حلقه‌ای (۲ تا n گره)
دسترس‌پذیری	$a_x \times a_y$	$a_{y1} \times a_{y2} \times \dots \times a_{yn}$	$\sum_{k=1}^n \frac{(\prod_{i=0}^{k-1} p_i)(1-p_k) \prod_{i=1}^k a_i}{1 - \prod_{i=1}^n (p_i a_i)}$
زمان پاسخ	$t_x + t_y$	$\text{Max}(t_{y1}, t_{y2}, \dots, t_{yn})$	$\sum_{k=1}^n \frac{(\prod_{i=0}^{k-1} p_i)(1-p_k)(\sum_{i=1}^k t_i + \prod_{i=1}^n p_i \sum_{i=k+1}^n t_i)}{(1 - \prod_{i=1}^n p_i)^2}$
هزینه	$c_x + c_y$	$c_{y1} + c_{y2} + \dots + c_{yn}$	$\sum_{k=1}^n \frac{(\prod_{i=0}^{k-1} p_i)(1-p_k)(\sum_{i=1}^k c_i + \prod_{i=1}^n p_i \sum_{i=k+1}^n c_i)}{(1 - \prod_{i=1}^n p_i)^2}$

جدول (۲): احتمال گذر الگوها [۵]

حلقه	موازی	ترتیبی	احتمال
$P'_{okj} = \sum_{i=0}^{+\infty} \left(\prod_{i=1}^n P_i \right)^l \left(\prod_{i=0}^{k-1} P_i \right)$ $(k \in [1, n], j \in [1, m_k])$	$P'_{in} = p$ $P'_{out} = 1$	$P'_{in} = P_x = \{p_{xi} i \in [i, m]\}$ $P'_{out} = P_y = \{p_{yj} j \in [i, n]\}$	احتمال گذر

ادغام و تبدیل به یک گره می‌شود به طوری که در پایان فقط یک گره باقی بماند [۹-۱۵]. از آنجا که در ساده‌سازی الگوی شرطی در این روش، مقدار احتمال یال‌ها در مقادیر خصیصه-های کیفی گره‌ها (سرویس‌های کاندید) ضرب می‌شوند، دقت این روش پایین است.

۲.۱،۳. روش‌های ساده‌سازی الگوها

در این بخش به بیان روش‌های ساده‌سازی الگو می‌پردازیم.

- روش تجمعی^۱

در این روش، گره‌های هر الگو (توالی، موازی، حلقه و شرط)

^۱ Aggregation method

• روش مبتنی بر مسیر^۱

در این روش مانند روش تجمعی گره‌ها حذف و در هم ادغام نمی‌شوند بلکه تنها تمام مسیرهای گراف از گره شروع تا گره پایان استخراج می‌شوند [۱۷]. در این روش اگر گرافی دارای الگوی حلقه با تعداد تکرار زیاد باشد، استخراج مسیرها موجب مصرف زیاد حافظه می‌شود، هم‌چنین این روش را برای بعضی از الگوها، مانند الگوی حلقه احتمالی نمی‌توان استفاده کرد.

• روش ترکیبی^۲

به منظور رفع مشکلات دو روش تجمعی و مبتنی بر مسیر، می‌توان آنها را ترکیب کرد [۵]. در روش ترکیب، الگوهای توالی، توازی و حلقه با روش تجمعی حذف می‌شوند ولی الگوهای شرطی باقی می‌مانند تا: (۱) احتمال هر مسیر گراف حفظ شود و (۲) از کاهش دقت الگوریتم جلوگیری شود (ضرب مقدار احتمال یک گذر در مقادیر خصیصه‌های کیفی گره انجام نمی‌شود).

۲.۲. بهینه‌سازی چندهدفه

یک مسئله مهم در گراف ترکیب وب‌سرویس‌ها، انتخاب مناسب وب‌سرویس کاندید برای هر سرویس انتزاعی در گراف است زیرا انتخاب سرویس‌های کاندید از یک طرف در ساده‌سازی الگوها نقش دارد و از طرف دیگر انتخاب آنها یک مسئله NP-Hard است.

از آنجا که لازم است انتخاب مناسب وب‌سرویس‌های کاندید با توجه به توازن بین خصیصه‌های کیفی مغایر هر وب‌سرویس (بخش ۲،۱،۲) انجام شود، روش بهینه‌سازی چندهدفه در این مقاله استفاده شده است. یافتن راه‌حل‌های بهینه، نیازمند جستجوی کامل در فضای حالت راه‌حل‌ها (جایگشت‌های مختلف ترکیب وب‌سرویس‌های کاندید) است که بسیار زمان‌بر است. یک راه‌حل استفاده از الگوریتم‌های تکاملی است تا راه‌حل‌های نزدیک به بهینه را در زمان مناسبی ارائه دهد. این راه‌حل‌های بهینه موسوم به راه‌حل‌های غالب در یک مجموعه

به نام جبهه پرتو قرار می‌گیرد.

• مفهوم غلبه

مفهوم غلبه با توجه به مقدار بهینه خصیصه‌های یک وب-سرویس تعریف می‌شود. هر خصیصه با یک تابع هدف مشخص می‌شود. بهینه‌سازی تابع هدف برای یک خصیصه مثبت مانند قابلیت دسترسی، ماکزیمم کردن مقدار این تابع و برای یک خصیصه منفی مانند هزینه، حداقل کردن این مقدار است. بنابراین، اگر u_1 و v_1 مقادیر دو تابع هدف برای دو خصیصه مثبت از دو وب‌سرویس s_1 و s_2 باشند، گوییم u_1 بر v_1 غلبه دارد اگر $u_1 > v_1$ در غیر این صورت $u_1 < v_1$. اگر $u = (u_1, u_2, u_3)$ و $v = (v_1, v_2, v_3)$ معرف دو بردار از مقادیر توابع هدف برای خصیصه‌های دو وب‌سرویس کاندید باشد، گوییم u بر v غلبه می‌کند اگر و فقط اگر مقدار هیچ عنصر (بعد) از v نسبت به مقدار عنصر متناظرش در u بهینه نباشد و حداقل مقدار یک عنصر از u بر مقدار عنصر متناظرش از v غلبه کند. رابطه (۱)، غلبه بردار u بر بردار v را نشان می‌دهد که در آن u_1 و u_2 و u_3 به ترتیب معرف مقادیر توابع هدف برای خصیصه‌های دسترس‌پذیری، زمان پاسخ و هزینه است.

$$(u_1 \geq v_1 \wedge u_2 \leq v_2 \wedge u_3 \leq v_3) \wedge (u_1 > v_1 \vee u_2 < v_2 \vee u_3 < v_3) \quad (1)$$

• جبهه پرتو

یک جبهه پرتو، مجموعه‌ای از راه‌حل‌ها با بردارهای غیر مغلوب هستند. در این مجموعه هیچ برداری مغلوب بردار دیگری نمی‌شود.

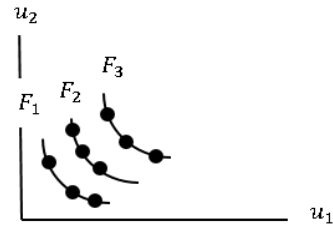
۲.۲.۱. الگوریتم NSGAII

در این الگوریتم هر کروموزوم یک عضو از جمعیت است. اعضای اولیه جمعیت به صورت تصادفی انتخاب می‌شوند (P_t). با تابع مرتب‌سازی نامغلوب اعضای جمعیت در جبهه‌های مختلفی دسته‌بندی می‌شوند. اعضای در جبهه اول (F_1) قرار می‌گیرند که هیچ عضوی از جمعیت آن‌ها را مغلوب نکرده است یعنی بر تمام اعضای دیگر جمعیت غالب هستند. اعضای جبهه دوم (F_2) همه جمعیت بجز اعضای جبهه اول را مغلوب

¹ Path based

² Hybrid based

کرده‌اند. این روال برای بقیه جبهه‌ها نیز وجود دارد، شکل (۲).



شکل (۲): دسته بندی اعضای جمعیت با الگوریتم مرتب سازی نامغلوب

به منظور ایجاد تفاوت بین اعضای موجود در یک جبهه مفهومی به نام فاصله ازدحامی^۱ تعریف شده است. فاصله ازدحامی هر عضو از جمعیت نشان دهنده‌ی این است که فاصله آن عضو با همسایگانش چقدر است که هر چه بیشتر باشد مطلوب تر است. فاصله ازدحامی بین دو عضو از یک جبهه پرتو، طبق روابط (۲) و (۳) محاسبه می شود [۱۴، ۲۶]. در رابطه (۲)، $i, i + 1$ و $i - 1$ سه عضو از جبهه پرتو هستند و u_k^i پارامتر k ام عضو i از جبهه پرتو است.

$$d_{i,k} = \frac{u_k^{i+1} - u_k^{i-1}}{u_k^i - u_k^1} \quad (2)$$

$$CD_i = \sum_{k=1}^m d_{i,k} \quad (3)$$

برای ایجاد جمعیت جدید والدین از بین اعضای جمعیت (F_1) انتخاب می شوند و با اعمال جهش^۲ و تقاطع^۳ روی والدین، جمعیت جدید ایجاد می شود. این جمعیت با جمعیت نامغلوب قبلی (F_1) ادغام می شود و دوباره مرتب سازی جهت یافتن رتبه هر عضو از جمعیت انجام می شود. اولین N عضو با کمترین رتبه و بالاترین فاصله ازدحامی ذخیره می شوند و بقیه جمعیت کنار گذاشته می شود. این فرآیند تا پایان نیافتن شرط خاتمه ادامه می یابد [۲۷].

۲.۲.۲. الگوریتم SPEAII

در این الگوریتم برای یافتن بهترین راه حل ها، به هر عضوی از جمعیت یک مقدار قدرت^۴ که با استفاده از رابطه (۴) بدست

می آید، تخصیص داده می شود که نشان دهنده‌ی تعداد اعضای است که آن عضو مغلوب می کند. مقدار شایستگی^۵ یک عضو که با استفاده از رابطه (۵) بدست می آید، به صورت مجموع مقدار قدرت همه اعضای است که آن عضو را مغلوب کرده‌اند. با توجه به این تعریف بنابراین مقدار تابع شایستگی برای اعضای نامغلوب جمعیت صفر است. این الگوریتم از یک فضای ذخیره سازی^۶ با سایزی از پیش تعیین شده به منظور ذخیره سازی اعضای نامغلوب هر نسل استفاده می کند با این راهکار اعضای نامغلوب از تاثیرات اتفاقی در امان می مانند. جمعیت خارجی که با جهش و تقاطع والدین به دست آمده است با جمعیت جاری ادغام می شوند و مقدار شایستگی جمعیت جدید تعیین می شود. در t بار اجرای فرآیند اعضای تکراری یا مغلوب از آن فضای ذخیره سازی حذف می شوند و فضای ذخیره سازی در هر تولید نسل جدید به روز می شود. حال به منظور تفاوت قائل شدن برای آن دسته اعضای که دارای مقدار شایستگی یکسانی هستند مفهوم چگالی اطلاعات^۷ تعریف می شود. در این الگوریتم روش خوشه بندی k امین نزدیکترین همسایه استفاده شده است. تابع چگالی (رابطه (۶)) معکوس فاصله تا k امین نزدیکترین همسایه است. پس هر چه مقدار تابع چگالی کمتر باشد احتمال باقی ماندنش در فضای ذخیره سازی نیز بیشتر است. بنابراین اگر جمعیت فضای ذخیره سازی از مقدار تعیین شده قبلی بیشتر باشد طبق تابع چگالی، اعضای اضافی حذف می شوند [۲۸].

$$S(i) = |\{j|j \in population \cup archive \& i > j\}| \quad (4)$$

$$R(i) = \sum_{j \in population+archive, j > i} S(j) \quad (5)$$

$$D(i) = \frac{1}{\partial_i^k + 2}, K = \sqrt{N + \bar{N}} \quad (6)$$

در رابطه (۴)، $| \cdot |$ کاردینالیته مجموعه را نشان می دهد و نماد $>$ رابطه غلبه پرتو^۸ را نشان می دهد. در رابطه (۶)، ∂_i^k فاصله i از

⁵ Raw fitness

⁶ Archive

⁷ Density information

⁸ Pareto Dominance

¹ Crowding distance

² Mutation

³ Crossover

⁴ Strength value

k امین نزدیک‌ترین همسایه‌اش است. مقدار k از جذر مجموع تعداد اعضای جمعیت اولیه و آرشیو بدست می‌آید.

۳. مروری بر کارهای گذشته

این بخش بر اساس چگونگی تعامل وب‌سرویس‌ها با یکدیگر در گراف ترکیب وب‌سرویس‌ها به چند زیربخش تقسیم شده است.

۳.۱. گراف ترکیب مورد نظر تنها دارای الگوی توالی

شالی لیو و همکاران (۲۰۰۵)، با استفاده از جستجوی اول عمق به صورت داینامیک شروع به یافتن بهترین کاندید هر وب-سرویس جهت ارضای قیود کلی کرده است. در این جستجو دو پارامتر زمان پاسخ و هزینه را به عنوان اهداف مساله مد نظر قرار داده، سپس از الگوریتم ژنتیک با استفاده از جمعیت اولیه یافت شده، جمعیت جدید را پیدا کرده است. این الگوریتم بارها تکرار شده است و در نهایت بهترین راه‌حل‌ها استخراج شده است [۱۲]. محمد آریفای و همکاران (۲۰۰۹)، از یک الگوریتم هیوریستیک توسعه یافته به منظور یافتن بهترین کاندید برای هر وب‌سرویس استفاده کرده‌اند. روش آن‌ها از آن جهت مقیاس پذیر است که مساله بهینه‌سازی کلی به چندین زیرمساله کوچکتر شکسته شده است و با یافتن راه‌حل بهینه محلی در هر مساله کوچک، راه‌حل کلی پیدا شده است. آزمایش‌ها روی گرافی با توالی چند وب‌سرویس انجام شده است، سه پارامتر زمان پاسخ، هزینه و بازدهی برای هر وب‌سرویس در نظر گرفته شده است. نتایج نشان می‌دهد این روش علاوه بر رسیدن به میانگین نرخ بهینگی بالای ۹۸ درصد، زمان اجرای بسیار پایینی هم دارد. در پایان، این الگوریتم از نظر نرخ بهینگی با الگوریتم هیوریستیک و از نظر زمان اجرا با الگوریتم برنامه‌ریزی خطی مقایسه شده است. نتایج نشان می‌دهد که از هر دو این الگوریتم‌ها بهتر عمل کرده است [۹]. هوان لیو و همکاران (۲۰۱۰)، الگوریتمی حاصل از ترکیب الگوریتم ژنتیک و الگوریتم کلونی مورچگان^۱ ارائه داده‌اند. آن‌ها از مزایای دو

الگوریتم به منظور ارتقای الگوریتم ترکیبی استفاده کرده‌اند. در آزمایش‌های انجام شده یک گراف جریان کار با الگوی متوالی و تعداد هفت وب‌سرویس در نظر گرفته شده است. چهار پارامتر دسترس‌پذیری، هزینه، زمان پاسخ و قابلیت اطمینان برای هر وب‌سرویس در نظر گرفته شده است. احتمال تقاطع را ۷۰٪ و احتمال جهش را ۶۰٪ در نظر گرفته‌اند. در این مقاله به پارامترها وزن داده شده است و مسئله به صورت تک‌هدفه حل شده است. نتایج نشان می‌دهد این روش نه تنها موفق به یافتن راه-حل‌های بهینه می‌شود بلکه سرعت همگرایی بالاتری نیز نسبت به الگوریتم کلونی مورچگان دارد [۱۰]. یوجی یاو و همکاران (۲۰۱۰)، به مقایسه الگوریتم ژنتیک مرتب‌سازی نامغلوب چندهدفه با اعمال قوانین^۲ و بدون قاعده و قانون پرداخته‌اند. در الگوریتم ژنتیک بر اساس قاعده راه‌حل‌های بهینه مرتب می‌شوند و ده تا از بهترین جواب‌های منطبق بر قیود در نظر گرفته می‌شوند و بر اساس این تعداد عضو، الگوریتم ادامه می‌یابد. نتایج نشان می‌دهد که این الگوریتم در مقایسه با الگوریتم ژنتیک بی‌قاعده منجر به یافتن راه‌حل‌های بهینه‌ای می‌شود که رضایت بیشتر کاربر را به همراه دارد [۱۳]. لی لی و همکاران (۲۰۱۰)، از الگوریتم ژنتیک مرتب‌سازی نامغلوب به منظور یافتن راه‌حل‌های بهینه در ترکیب وب‌سرویس‌ها با در نظر گرفتن ۱۰ تابع هدف (پارامتر) استفاده کرده‌اند و با گزارش تعداد راه‌حل‌های غالب و مغلوب به ارزیابی الگوریتم مورد نظر پرداخته‌اند [۱۴]. سورا گاهان و همکاران (۲۰۱۶)، الگوریتم ترکیبی کلونی مورچگان و ازدحام ذرات را با در نظر گرفتن پنج تابع هدف قابلیت اطمینان، دسترس‌پذیری، بازدهی، هزینه و زمان پاسخ و تبدیل این پارامترها به یک هدف به کار برده‌اند. این الگوریتم طی آزمایش‌هایی از نظر زمان اجرا و میزان برازندگی با الگوریتم ازدحام ذرات معمولی مقایسه شده است. است که نشان از عملکرد بهتر الگوریتم ترکیبی است [۱۱]. یینگ هو و همکاران (۲۰۱۷)، الگوریتم کلونی زنبورعسل چندهدفه نخبه‌گرا^۳ را ارائه کرده‌اند که از الگوریتم کلونی زنبور

² Rule based NSGAI

³ Elite-guided Multi-objective artificial bee colony algorithm

¹ Ant colony

ترکیبی ژنتیک و کلونی مورچگان دارد. هرچند زمان اجرای آن بیشتر است. الگوریتم ارائه شده دوم از جستجوی هارمونی^۳ به عنوان اپراتور جهش استفاده کرده است. این الگوریتم نیز از هر دو الگوریتم گفته شده قبلی بهتر است. نرخ همگرایی این الگوریتم از ژنتیک بهتر است و به میزان برازندگی بهتری در نسل‌های پایین‌تر می‌رسد. دیگر مزیت هر دو الگوریتم ارائه شده نسبت به ژنتیک این است که بهترین جواب این الگوریتم‌ها از بهترین جواب الگوریتم ژنتیک خیلی بهتر است [۱۶].

۳.۳. گراف ترکیب وب‌سرویس مورد نظر دارای الگوی متوالی، موازی و حلقه

مالین تانگ و همکاران (۲۰۱۰)، الگوریتم ژنتیک ترکیبی ارائه داده‌اند که از یک بهینه‌ساز محلی به منظور بهبود اعضای جمعیت استفاده می‌کند. تمام کاندیدای هر وب‌سرویس توسط این بهینه‌ساز بررسی شده و بدین طریق با هرس کردن بخشی از کاندیدا فضای جستجو را کاهش می‌دهد. هم‌چنین این بهینه‌ساز علاوه بر افزایش کیفیت کلی، سبب کاهش نقص محدودیت‌های گذاشته شده می‌شود. در پایان این الگوریتم با الگوریتم‌های ژنتیکی مبتنی بر مجازات^۴ و ژنتیکی مبتنی بر ترمیم^۵ از نظر زمان اجرا و تابع برازندگی مقایسه شده‌اند. نتایج نشان می‌دهد که الگوریتم ترکیبی ژنتیک بهتر از دو الگوریتم دیگر عمل کرده است [۱۸].

۳.۴. گراف ترکیب وب‌سرویس مورد نظر دارای انواع الگوها (متوالی، موازی، شرطی و حلقه)

جراردو کانفورا و همکاران (۲۰۰۵)، به منظور انتخاب بهینه وب-سرویس‌ها از الگوریتم تکاملی ژنتیک تک‌هدفه استفاده کرده‌اند و با الگوریتم برنامه‌نویسی عددی^۶ از نظر زمان اجرا مقایسه کرده‌اند. نتایج نشان داده است که با در نظر گرفتن راه‌حل‌های یکسان زمان همگرایی الگوریتم ژنتیک در مقیاس‌های بزرگ

عسل و روش مرتب‌سازی نامغلوب، انتخاب جمعیت، تولید راه‌حل‌های گسسته نخبه‌گرا و روش محاسبه تابع برازندگی چندهدفه تشکیل شده است. آزمایش‌ها روی دو مجموعه داده انجام گرفته است. پارامترهای در نظر گرفته شده برای هر وب-سرویس عبارتند از دسترس‌پذیری، زمان پاسخ، اعتبار و بازدهی که این چهار هدف را تبدیل به یک هدف کرده است و هزینه را به عنوان هدفه دیگر در نظر گرفته است و مسئله را دو هدفه حل کرده است. نتایج نشان می‌دهد این الگوریتم از نظر شاخص‌های کیفی فاصله بین نسلی، پراکندگی، زمان اجرا بهتر از الگوریتم‌های ژنتیک مرتب‌سازی نامغلوب و ازدحام ذرات و کلونی زنبور عسل است [۱۵].

۳.۲. گراف ترکیب وب‌سرویس مورد نظر دارای الگوی متوالی و موازی

دانیلو آرداگنا و همکاران (۲۰۰۵)، الگوریتم برنامه‌نویسی ترکیبی خطی و عددی^۱ ارائه کرده‌اند که با در نظر گرفتن چهار تابع هدف و تبدیل آن به تابع تک‌هدفه کارایی آن ارزیابی شده است. این مساله با تبدیل گراف ترکیب وب‌سرویس‌ها به مسیرهای اجرایی مختلف متشکل از انتخاب کاندیدای مختلف برای هر وب‌سرویس حل شده است. در این روش مسیرهایی منتخب هستند که بهینه‌ترین راه‌حل را ارائه دهند [۱۷]. اردینک ایلماز و همکاران (۲۰۱۴)، دو الگوریتم ژنتیک بهبود یافته ارائه کرده‌اند، در این الگوریتم‌ها به جای استفاده از اپراتور جهش اصلی از اپراتور جهش هوشمند استفاده شده است. در جهش اصلی هیچ وقت میزان برازندگی عضو جدید ارزیابی نمی‌شود ولی اپراتور جهش هوشمند بدین صورت است که در نسل‌های ابتدایی میزان برازندگی اعضای جهش‌یافته ارزیابی نمی‌شود ولی در نسل‌های بالاتر تنها جهش‌هایی پذیرفته شده‌اند که تابع برازندگی آن‌ها بزرگتر باشد. الگوریتم ارائه شده اول از شبیه-سازی ذوب فلزات^۲ به عنوان اپراتور جهش استفاده کرده است. این الگوریتم تابع برازندگی بهتری نسبت به ژنتیک و الگوریتم

³ Harmony search

⁴ Penalty based genetic algorithm

⁵ Repairing based genetic algorithm

⁶ Integer programming

¹ Mixed integer linear programming

² Simulated annealing

سال ۲۰۰۹ الگوی جدیدی پیشنهاد دادند. این الگو، الگوی حلقه غیرساخت‌یافته چند ورودی و چند خروجی^۵ نام دارد. احتمال نیز برای هر مسیر محاسبه می‌شود و به منظور انتخاب بهینه کاندیدا درگراف ترکیب وب‌سرویس از الگوریتم برنامه‌نویسی عددی استفاده شده است [۵]. پروین شریف‌آرا و همکاران (۲۰۱۴)، روشی ارائه داده‌اند که این روش علاوه بر رسیدن به راه‌حل بهینه سرعت اجرای بالایی نیز دارد و این مساله زمانی که تعداد وب‌سرویس‌های در دسترس زیاد است، نمود بیشتری پیدا می‌کند. در روش ارائه شده ابتدا مساله به یک مساله تک‌هدفه تبدیل می‌شود در این حالت به منظور افزایش سرعت، دقت و قابلیت اطمینان ترکیبی از الگوریتم ژنتیک و روش فازی به کار گرفته شده است. سپس مساله تغییر کرده است و به یک مساله چندهدفه تبدیل شده است که با الگوریتم ژنتیک مرتب‌سازی نامغلوب حل شده است. در این روش به منظور بهبود زمان اجرا تابع مرتب‌سازی نامغلوب در الگوریتم ژنتیک تغییر کرده است. تابع جایگزین هر ویژگی در هر عضو از جمعیت را مرتب می‌کند و به آن ویژگی یک اندیس می‌دهد و در پایان اندیس‌های به دست آمده از تمام ویژگی‌های یک عضو را با یکدیگر جمع می‌کند و اعضای جمعیت را بر اساس مقدار آن اندیس مرتب می‌کند. به این ترتیب پیچیدگی زمانی الگوریتم را کم می‌کند. دو شرط توقف نیز برای این روش ارائه شده است. شرط اول: بیشترین تعداد نسل تولید شده نباید از تعداد از پیش تعیین شده تجاوز کند و شرط دوم: تا زمانی الگوریتم ادامه می‌یابد که بهترین راه‌حل در تعدادی نسل ثابت باقی بماند [۲۴]. لی لیو و همکاران (۲۰۱۵)، ابتدا دو الگوریتم چندهدفه مرتب‌سازی نامغلوب و بهینه‌سازی ازدحام ذرات را مقایسه کرده‌اند و با اندازه‌گیری زمان اجرا، فاصله بین‌نسلی معکوس و پراکندگی این دو الگوریتم برتری الگوریتم مرتب‌سازی نامغلوب و بهینه‌سازی ازدحام ذرات را با روش فرآیند تحلیل سلسله‌مراتبی^۶ ادغام کرده‌اند، در این الگوریتم‌ها از فرآیند

بسیار کمتر است [۱۹]. دبدوت موخرج و همکاران (۲۰۰۸)، الگوهای پیچیده‌تری را نسبت به مقاله آقای هولیان (۲۰۱۲) در نظر گرفته است. در همین راستا به منظور پوشش الگوهای شرطی غیرساخت‌یافته کیفیت سرویس‌ها از گره برگ تا ریشه محاسبه می‌شود ولی الگوهای حلقه غیرساخت‌یافته حل نشده باقی مانده است [۲۰]. یوجی یاو و همکاران (۲۰۰۹)، از الگوریتم ژنتیک مرتب‌سازی نامغلوب به منظور یافتن راه‌حل‌های بهینه در گراف با الگوهای پیچیده استفاده کرده‌اند و با الگوریتم ژنتیک از نظر سرعت همگرایی و تولید جواب‌های بهینه در یک زمان خاص مقایسه کرده‌اند. میانگین انحراف کمتر و در نتیجه سرعت همگرایی کمتر این الگوریتم که سبب انتخاب گسترده‌تر مشتریان می‌شود نشان‌دهنده‌ی عملکرد بهتر آن نسبت به ژنتیک است [۲۱]. هولیان ژنگ و همکاران (۲۰۰۹)، علاوه بر پوشش الگوهای توالی، موازی، شرطی ساخت‌یافته و غیرساخت‌یافته، الگوی حلقه با یک ورودی و تعدادی خروجی^۱ را نیز تشریح کرده‌اند. هم‌چنین یک روش سیستماتیک به منظور محاسبه کیفیت الگوها ارائه شده است. در پایان هزینه و زمان پاسخ و احتمال هر مسیر به طور مجزا به دست آمده است [۶]. لی جین ژنگ و همکاران (۲۰۱۰)، از الگوریتم تکاملی مبتنی بر قوت^۲ استفاده کرده‌اند تا مجموعه‌ای از راه‌حل‌های بهینه پرتو را با در نظر گرفتن هم‌زمان مجموعه‌ای از توابع هدف که عبارتند از کاهش هزینه و زمان اجرا و هم‌چنین با رعایت قیود در نظر گرفته شده برای اهداف دسترس‌پذیری و قابلیت اطمینان بدست آورند. نتایج آزمایشات امکان‌سنجی^۳ و کارایی^۴ این الگوریتم را نشان می‌دهد [۲۳]. مارلون دوماس و همکاران (۲۰۱۰)، انواع الگوهای پیچیده را در گراف ترکیب وب‌سرویس‌ها لحاظ کرده‌اند، هم‌چنین با در نظر گرفتن سه تابع هدف و استفاده از وزن‌های داده شده توسط کاربر، مساله را با روش تجمعی و به صورت تک‌هدفه حل کرده‌اند [۲۲]. هویان ژنگ و همکاران (۲۰۱۲)، به منظور توسعه کار پیشین خود در

¹ SEME

² Strength Pareto Evolutionary Algorithm

³ Feasibility

⁴ Efficiency

⁵ Multiple entry multiple exit

⁶ Analytical hierarchy process

SPEAII انجام شده است که نشان‌دهنده کیفیت برتر راه‌حل‌ها در الگوریتم پیشنهادی است [۳۰]. ژی نا و همکاران (۲۰۲۱)، یک روش دو فازی کارآمد با تلفیق الگوریتم خوشه‌بندی و ازدحام ذرات مبتنی بر Chaos-Gauss پیشنهاد داده‌اند. در این روش، در فاز یک الگوریتم Kmeans سرویس‌های نامزد را خوشه‌بندی می‌کند و کاندیدا با مقادیر کیفی ناپایدار هرس می‌شوند. در فاز دو، الگوریتم جدیدی به نام CGPSO برای یافتن ترکیب بهینه کاندیدا پیشنهاد شده است. این رویکرد به دلیل پشتیبانی از تغییر پارامترهای کیفی کاندیدا ارائه شده تا سبب افزایش قابلیت اطمینان در ترکیب و سرویس‌ها شود. در این روش، هرج و مرج به ازدحام ذرات اضافه شده است و همچنین از اپراتور اختشاش گاوس برای حفظ تنوع جمعیت و جلوگیری از همگرایی زودرس استفاده شده است. نتایج آزمایش‌ها نشان می‌دهد که این روش می‌تواند منجر به راه‌حل‌های بهینه‌تری در مساله ترکیب و سرویس‌ها شود [۳۱]. سنگاراج و همکاران (۲۰۲۰)، الگوریتم ترکیبی ژنتیک و جستجوی تابو^۲ را برای یافتن بهترین کاندیدا بر اساس درخواست کاربر و متناسب با پارامترهای زمان پاسخ، هزینه و قابلیت اطمینان ارائه کرده‌اند. در این روش با ارزیابی مقدار آستانه^۳، بهترین مجموعه کاندیدای قابل همکاری با حداکثر قابلیت اطمینان و توان عملیاتی با استفاده از جستجوی تابو به کاربر نهایی پیشنهاد می‌شود. آزمایش‌ها نشان می‌دهد که روش پیشنهادی به طور متوسط ۰/۵٪ بهبود در برازندگی و حدود ۰/۲۵٪ کاهش خطا داشته است [۳۲].

در اغلب مقالات بیان شده، و سرویس‌ها تنها با الگوی توالی با یکدیگر در ارتباط هستند، هم‌چنین تنها [۵] به بررسی الگوهای احتمالی پرداخته است. از نظر استفاده از الگوریتم‌های تکاملی، الگوریتم NSGAI از الگوریتم‌هایی است که به منظور انتخاب بهینه و سرویس‌ها به تعداد زیاد استفاده و مطرح شده است [۱۳، ۱۴، ۲۱، ۲۳، ۲۵، ۲۸]. در مقالات [۲۳، ۲۸] از الگوریتم SPEAII استفاده شده است. در این مقاله برای ساختار گراف موردنظر انواع الگوهای احتمالی در نظر گرفته شده و هم‌چنین به منظور انتخاب بهینه و سرویس‌ها، دو الگوریتم NSGAI و SPEAII که عملکرد خوبی داشته‌اند انتخاب می‌شوند. در جدول (۳) خلاصه‌ای از کارهای مرتبط آمده است.

تحلیل سلسله‌مراتبی به جای فاصله ازدحامی استفاده شده است و با الگوریتم‌های ژنتیک ادغام شده با فرآیند تحلیل سلسله‌مراتبی و ازدحام ذرات تک‌هدفه ادغام شده با فرآیند تحلیل سلسله‌مراتبی و هم‌چنین الگوریتم‌های چندهدفه مرتب‌سازی نامغلوب و بهینه‌سازی ازدحام ذرات که در انتها روی جواب‌هایشان از تحلیل سلسله‌مراتبی استفاده شده است، مقایسه شده است. نتایج نشان داده است که این رویکرد قادر است به طور دقیق‌تر ترجیحات کاربران را در نظر بگیرد [۲۵]. سامیا چیبانی سادوکی و همکاران (۲۰۱۹)، از الگوریتم فیل چندهدفه گسسته^۱ استفاده کرده‌اند. برای رسیدن به این منظور از الگوریتم فیل اصلی که مناسب فضاهای پیوسته است استفاده شده و با عملگر تقاطع ترکیب شده تا مناسب فضاهای گسسته شود. هم‌چنین از رویکرد پرتو و روش مرتب‌سازی الگوریتم چندهدفه مبتنی بر قدرت الهام گرفته‌اند. قدرت الگوریتم پیشنهادی با فرآیند تقسیم و ترکیب جمعیت با زیر جمعیت (قبیله) فراهم می‌شود که سبب گیر نیفتادن در نقطه بهینه محلی و هم‌چنین سبب تبادل اطلاعات در جستجوهای محلی برای رسیدن به بهینه کلی می‌شود. با مقایسه این الگوریتم با الگوریتم ازدحام ذرات و الگوریتم مبتنی بر قدرت این نتیجه حاصل شد که از نظر معیارهایی چون نرخ پوشش‌دهی، پراکندگی و حجم اضافی به طور قابل توجهی بهتر است [۲۹]. فاتح صغیر (۲۰۲۱) با توجه به پویایی محیط‌ها و نامشخص بودن پارامترهای کیفی، مساله ترکیب و سرویس‌ها را به عنوان یک مسئله بهینه‌سازی چندهدفه فازی فرموله کرده که برای رسیدن به این منظور: (۱) پارامترهای کیفی را با اعداد فازی نشان داده است. (۲) یک الگوریتم گسسته چندهدفه فازی زنبور عسل ارائه کرده که برای این الگوریتم روش جدید رتبه‌بندی فازی برای مرتب‌سازی راه‌حل‌ها و فاصله فازی جدیدی برای کنترل و حفظ تنوع راه‌حل‌ها به صورت فازی تعریف کرده است. (۳) یک روش تصمیم‌گیری چند معیاره فازی برای تعیین بهترین خدمات ترکیبی در بین راه‌حل‌های بهینه پرتو بیان شده است. در پایان مقایسه‌ای بین الگوریتم گسسته چندهدفه فازی زنبور عسل و الگوریتم‌های تک‌هدفه TGA، EFPA و نسخه‌های گسترش‌یافته فازی NSGAI و

² Tabu search

³ Threshold

¹ Multi objective Discrete Elephants Herding Optimization(MO-D-EHO)

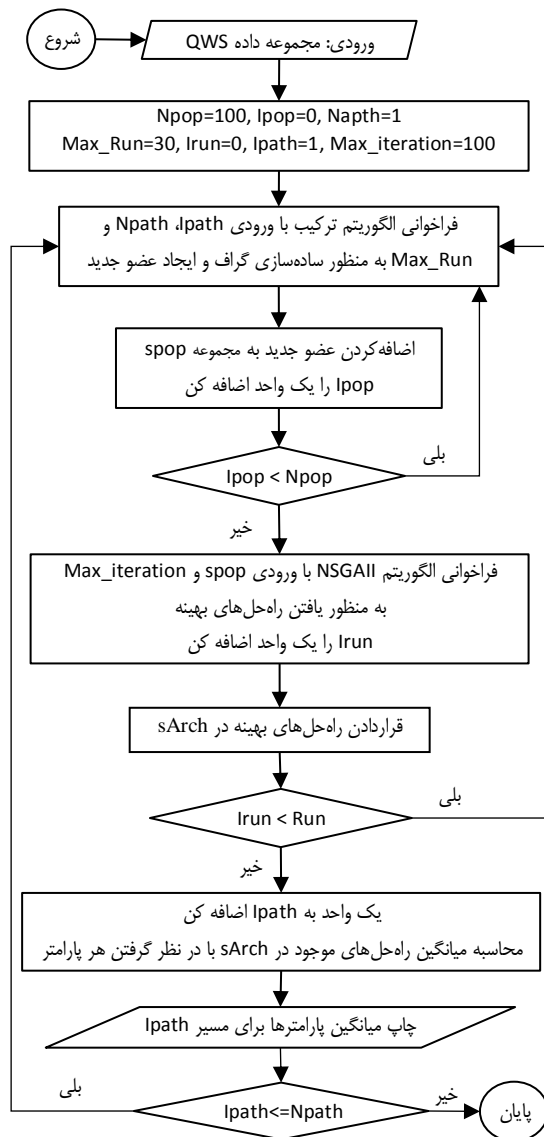
جدول (۳): مقایسه کارهای گذشته

مرجع	الگوریتم مورد استفاده	الگوریتم مقایسه شده	معیار ارزیابی		انبار شبیه‌سازی	گراف احتمالی	مزایا	معایب
			اهداف	شاخص‌ها				
[۵]	IP	-	هزینه، تاخیر، زمان پاسخ	-	✓	✓	۱. پشتیبانی از الگوی توالی، موازی، شرطی و انواع حلقه ۲. ساده‌سازی ترکیبی ۳. چند هدفه بودن	۱. عدم پشتیبانی از الگوی حلقه تو در تو و موازی ۲. مقیاس ناپذیری (زمان اجرا)
[۹]	DIST-HEU	LP, WS-HEU	زمان پاسخ، هزینه، دسترس پذیری	نرخ بهینگی زمان اجرا	✓	✓	۱. پشتیبانی از الگوی توالی، ۲. مقیاس پذیری (زمان اجرا) ۳. ساده‌سازی تجمعی ۴. تک هدفه بودن	۱. عدم پشتیبانی از الگوی موازی، شرطی، حلقه ساده، تو در تو و موازی ۲. گیر کردن در بهینه محلی ۳. ساده‌سازی تجمعی
[۱۰]	A-G(ACO and GA)	ACO	دسترس پذیری، هزینه، زمان پاسخ، قابلیت اطمینان	سرعت همگرایی	✓	✓	۱. پشتیبانی از الگوی توالی ۲. مقیاس پذیری (زمان اجرا) ۳. تک هدفه بودن	۱. عدم پشتیبانی از الگوی موازی، شرطی، حلقه ساده، تو در تو و موازی ۲. ساده‌سازی تجمعی
[۱۱]	PSO-ACO	PSO	هزینه، زمان پاسخ، دسترس پذیری، قابلیت اطمینان	زمان اجرا	✓	✓	۱. پشتیبانی از الگوی توالی، ۲. مقیاس پذیری (زمان اجرا) ۳. تک هدفه بودن	۱. عدم پشتیبانی از الگوی موازی، شرطی، حلقه ساده، تو در تو و موازی ۲. ساده‌سازی تجمعی
[۱۲]	GA	-	هزینه، زمان پاسخ	-	✓	✓	۱. پشتیبانی از الگوی توالی ۲. مقیاس پذیری (زمان اجرا) ۳. چند هدفه بودن	۱. عدم پشتیبانی از الگوی موازی، شرطی، حلقه ساده، تو در تو و موازی ۲. ساده‌سازی تجمعی
[۱۳]	Rule-Based NSGAI	NSGAI	زمان پاسخ، قابلیت اطمینان	-	✓	✓	۱. پشتیبانی از الگوی توالی ۲. مقیاس پذیری (زمان اجرا) ۳. چند هدفه بودن	۱. عدم پشتیبانی از الگوی موازی، شرطی، حلقه ساده، تو در تو و موازی ۲. ساده‌سازی تجمعی
[۱۴]	NSGAI	-	-	تعداد راه‌حل‌های غالب و مغلوب	✓	✓	۱. پشتیبانی از الگوی توالی ۲. مقیاس پذیری (زمان اجرا) ۳. چند هدفه بودن	۱. عدم پشتیبانی از الگوی موازی، شرطی، حلقه ساده، تو در تو و موازی ۲. ساده‌سازی تجمعی
[۱۵]	EMOABC	NSGA-II, MOPSO, MOABC	زمان پاسخ، دسترس پذیری، اعتبار، بازدهی	فاصله بین نسلی پراکندگی نرخ خطا زمان اجرا	✓	✓	۱. پشتیبانی از الگوی توالی، ۲. مقیاس پذیری (زمان اجرا) ۳. چند هدفه بودن	۱. عدم پشتیبانی از الگوی موازی، شرطی، حلقه ساده، تو در تو و موازی ۲. ساده‌سازی تجمعی
[۱۶]	GA-HS, GA-SA	GA-ACO, GA	هزینه، زمان پاسخ، قابلیت اطمینان، بازدهی	بهترین برازندگی میانگین برازندگی، زمان اجرا	✓	✓	۱. پشتیبانی از الگوی توالی، موازی، ۲. مقیاس پذیری (زمان اجرا) ۳. ساده‌سازی تجمعی ۴. تک هدفه بودن	۱. عدم پشتیبانی از الگوی موازی، شرطی، حلقه ساده، تو در تو و موازی ۲. عدم نمایش گراف مورد استفاده ۳. ساده‌سازی تجمعی

<p>۱. مقیاس ناپذیری (زمان اجرا) ۲. عدم پشتیبانی از انواع الگوی شرطی و حلقه ۳. ساده سازی مبتنی بر مسیر ۴. تک هدفه بودن</p>	<p>۱. پشتیبانی از الگوی توالی و موازی</p>	<p>زمان، هزینه، دسترس پذیری، اعتبار</p>	<p>-</p>	<p>MILP [۱۷]</p>
<p>۱. عدم پشتیبانی از الگوی شرطی، حلقه تو در تو و موازی ۲. ساده سازی تجمعی ۳. تک هدفه بودن</p>	<p>۱. پشتیبانی از الگوی توالی، موازی، حلقه با تکرار ۲. مقیاس پذیری (زمان اجرا)</p>	<p>هزینه، زمان پاسخ، دسترس پذیری، قابلیت اطمینان</p>	<p>زمان اجرا، انحراف</p>	<p>RGA PGA HGA [۱۸]</p>
<p>۱. عدم پشتیبانی از الگوی حلقه تو در تو و موازی ۲. ساده سازی تجمعی ۳. تک هدفه بودن</p>	<p>۱. پشتیبانی از الگوی توالی، موازی، شرطی و حلقه با تکرار ۲. مقیاس پذیری (زمان اجرا)</p>	<p>هزینه، زمان پاسخ، دسترس پذیری</p>	<p>زمان همگرایی</p>	<p>IP GA [۱۹]</p>
<p>۱. عدم پشتیبانی از الگوی حلقه تو در تو و موازی ۲. ساده سازی تجمعی</p>	<p>۱. پشتیبانی از الگوی توالی، موازی، شرطی و انواع حلقه ۲. چند هدفه بودن</p>	<p>هزینه، زمان پاسخ، قابلیت اطمینان</p>	<p>-</p>	<p>[۲۰]</p>
<p>۱. عدم پشتیبانی از الگوی موازی، شرطی، حلقه ساده، تو در تو و موازی ۲. عدم نمایش گراف مورد استفاده ۳. ساده سازی تجمعی ۴. تک هدفه بودن</p>	<p>۱. پشتیبانی از الگوی توالی، موازی، شرطی و حلقه با تکرار ۲. مقیاس پذیری (زمان اجرا)</p>	<p>هزینه، زمان پاسخ، دسترس پذیری، بازدهی، اعتبار</p>	<p>سرعت همگرایی میانگین انحراف</p>	<p>GA NSGAI [۲۱]</p>
<p>۱. عدم پشتیبانی از الگوی حلقه تو در تو و موازی ۲. ساده سازی تجمعی ۳. تک هدفه بودن</p>	<p>۱. پشتیبانی از الگوی توالی، موازی، شرطی و انواع حلقه</p>	<p>هزینه، زمان پاسخ، دسترس پذیری</p>	<p>-</p>	<p>[۲۲]</p>
<p>۱. عدم پشتیبانی از الگوی حلقه تو در تو و موازی ۲. ساده سازی تجمعی</p>	<p>۱. پشتیبانی از الگوی توالی، موازی، شرطی و حلقه با تکرار ۲. مقیاس پذیری (زمان اجرا) ۳. چند هدفه بودن</p>	<p>هزینه، دسترس پذیری، قابلیت اطمینان</p>	<p>زمان اجرا</p>	<p>SPEAI [۲۳]</p>
<p>۱. عدم پشتیبانی از الگوی حلقه تو در تو و موازی ۲. عدم نمایش گراف مورد استفاده ۳. ساده سازی تجمعی</p>	<p>۱. پشتیبانی از الگوی توالی، موازی، شرطی و حلقه با تکرار ۲. مقیاس پذیری (زمان اجرا) ۳. چند هدفه بودن</p>	<p>هزینه، زمان پاسخ، دسترس پذیری</p>	<p>زمان اجرا</p>	<p>NSGAI Improved-NSGAI [۲۴]</p>
<p>۱. عدم پشتیبانی از الگوی حلقه تو در تو و موازی ۲. ساده سازی تجمعی</p>	<p>۱. پشتیبانی از الگوی توالی، موازی، شرطی، حلقه ساده با تکرار ۲. مقیاس پذیری (زمان اجرا) ۳. چند هدفه بودن</p>	<p>هزینه، زمان پاسخ، قابلیت اطمینان</p>	<p>فاصله بین نسلی معکوس پراکنندگی زمان اجرا</p>	<p>GA, PSO, NSGAI with AHP, MOPSO with AHP, NSGAI-AHP, MOPSO-AHP [۲۵]</p>
<p>۱. عدم پشتیبانی از الگوی حلقه تو در تو و موازی ۲. گراف مورد استفاده تنها دارای الگوی توالی ۳. ساده سازی تجمعی</p>	<p>۱. پشتیبانی از الگوی توالی، موازی، شرطی، حلقه ساده با تکرار ۲. مقیاس پذیری (زمان اجرا) ۳. چند هدفه بودن</p>	<p>-</p>	<p>نرخ پوشش دهی پراکنندگی حجم اضافه</p>	<p>SPEAI MOPSO EHO [۲۹]</p>
<p>۱. عدم پشتیبانی از الگوی حلقه تو در تو و موازی ۲. گراف مورد استفاده تنها دارای الگوی توالی ۳. ساده سازی تجمعی</p>	<p>۱. پشتیبانی از الگوی توالی، موازی، شرطی، حلقه ساده با تکرار ۲. مقیاس پذیری (زمان اجرا) ۳. چند هدفه بودن</p>	<p>هزینه، زمان پاسخ، دسترس پذیری، قابلیت اطمینان</p>	<p>نرخ پوشش دهی، پراکنندگی، فاصله بین نسلی معکوس</p>	<p>FESPEAI, FENSGAI, TEGA, FPA, FDMOABC [۳۰]</p>

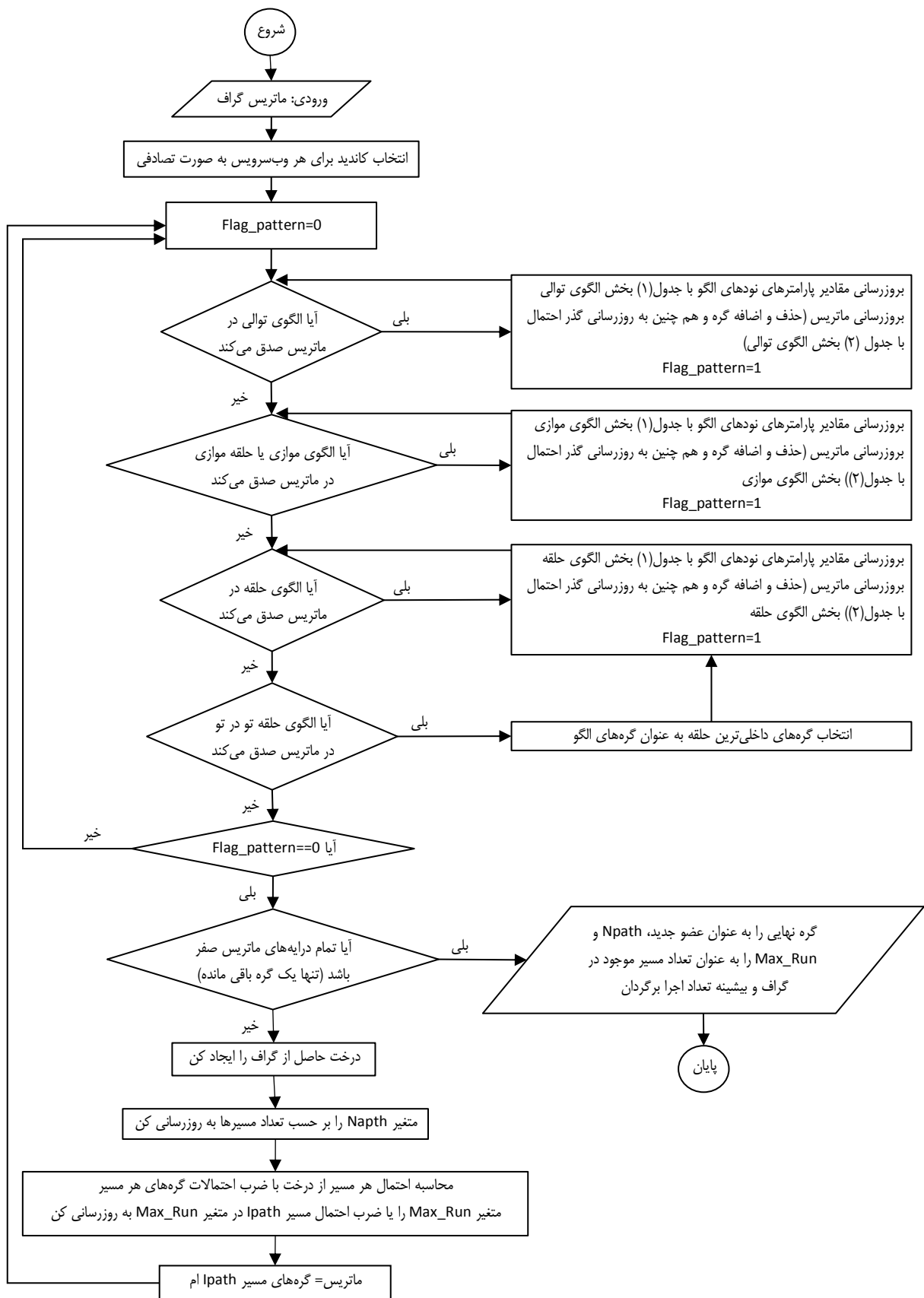
[۳۱]	CGPSO	PMOPSO, MOPSO	دقت، زمان اجرا، نرخ بهینگی	هزینه، زمان پاسخ، قابلیت اطمینان	شکل ۱	۱. پشتیبانی از الگوی توالی، موازی، شرطی، حلقه ساده با تکرار ۲. مقیاس پذیری (زمان اجرا) ۳. ساده‌سازی تجمعی ۴. تک هدفه بودن	۱. عدم پشتیبانی از الگوی حلقه تو در تو و موازی ۲. گراف مورد استفاده تنها دارای الگوی توالی ۳. ساده‌سازی تجمعی ۴. تک هدفه بودن
[۳۲]	GA-Tabu	GA	دقت، حساسیت، نرخ خطا، بهترین برازندگی	هزینه، زمان پاسخ، قابلیت اطمینان	NetBeans	۱. پشتیبانی از الگوی توالی، موازی، شرطی، حلقه ساده، تو در تو و موازی ۲. مقیاس پذیری (زمان اجرا) ۳. ساده‌سازی تجمعی ۴. تک هدفه بودن	۱. عدم پشتیبانی از الگوی موازی، شرطی، حلقه ساده، تو در تو و موازی ۲. ساده‌سازی تجمعی ۳. تک هدفه بودن
روش پیشنهادی	NSGAI	SPEAI, MOPSO, NSGAI with AHP, MOPSO with AHP	فاصله بین نسلی معکوس پراکندگی نرخ پوشش‌دهی، توزیع شدگی زمان اجرا	هزینه، زمان پاسخ، دسترس پذیری	شکل ۲	۱. پشتیبانی از الگوی توالی، موازی، شرطی، حلقه ساده، تو در تو و موازی ۲. مقیاس پذیری (زمان اجرا) ۳. ساده‌سازی ترکیبی ۴. چند هدفه بودن	-

۴. روش پیشنهادی



صورت کلی الگوریتم در شکل (۳) نشان داده شده است. این الگوریتم از دو بخش اصلی تشکیل شده است. بخش اول ساده-سازی گراف ترکیب کاندیدا (شکل (۴)) به منظور ایجاد عضوی از جمعیت و بخش دوم استفاده از الگوریتمی تکاملی به منظور یافتن بهترین کاندیدا که در بخش ترکیب ضمن ارائه دو الگوی جدید در ترکیب وب‌سرویس‌ها، روشی جدید به منظور ساده-سازی ترکیب وب‌سرویس با دقت بیشتر ارائه شده است و در بخش انتخاب بهینه کاندیدا نحوه ایجاد جمعیت اولیه و انتخاب بهترین راه‌حل‌ها با استفاده از تابع برازندگی الگوریتم مورد نظر تشریح شده است. در جریان کاری نمایش داده شده در شکل‌های (۳) و (۴) پارامترهای استفاده شده به شرح زیر هستند. متغیر $Npop$ تعداد اعضای جمعیت الگوریتم، $lpop$ شمارنده تعداد اعضای جمعیت، Max_Run تعداد بار اجرای الگوریتم، $lrun$ شمارنده تعداد بار اجرا که برای شروع مقدار صفر می‌گیرد. پارامتر $Npath$ تعداد مسیرهای موجود در گراف و $lpath$ شمارنده این مسیرهاست که در ابتدا مقدار آن یک است. پارامتر $Spop$ مجموعه‌ای است شامل اعضای الگوریتم، $Max_iteration$ بیشینه تعداد تکرار، $sArch$ مجموعه‌ای شامل بهترین اعضای بدست آمده الگوریتم، $Flag_pattern$ متغیری باینری با مقدار یک یا صفر به منظور چک کردن وجود یا عدم وجود الگوهای توالی، موازی و حلقه در ساختار گراف است.

شکل (۳): جریان کار روش پیشنهادی



شکل (۴): جریان کار الگوریتم ساده سازی ترکیب گراف

۴.۱. شناسایی دو الگوی جدید

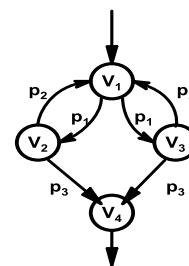
در این زیربخش به شناسایی دو الگوی جدید که از نوع حلقه با یک ورودی و یک خروجی هستند پرداخته شده است.

• الگوی حلقه موازی

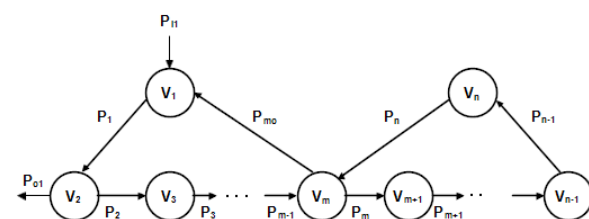
دو گره v_2 و v_3 را حلقه موازی گویند اگر این دو گره: (الف) با یکدیگر موازی باشند (طبق تعریف الگوی موازی) و (ب) هر دو گره یال ورودی به گره مشترک ورودی داشته باشند. مورد الف در الگوی موازی تشریح شده و مورد ب یعنی دو یال ورودی e_{j2} و e_{j3} به صورت $e_{j2} = (v_2, v_1, p_{21})$ و $e_{j3} = (v_3, v_1, p_{31})$ وجود داشته باشد (شکل (۵)).

• الگوی حلقه تو در تو

دنباله گره‌های $v_1, v_2, \dots, v_m, \dots, v_{m+1}, \dots, v_n$ تشکیل دو حلقه تو در تو می‌دهند اگر: (الف) مجموعه گره‌های v_1, v_2, \dots, v_m و مجموعه گره‌های v_{m+1}, v_m, \dots, v_n هر کدام تشکیل حلقه با یک مشترک داشته باشند. برای مورد الف، تعریف الگوی حلقه در بخش ۱، ۱، ۲ را ببینید و مورد ب یعنی گره‌ای از حلقه اول (v_m)، گره ورودی حلقه دوم باشد، در این صورت داریم: درجه ورودی (v_m) با درجه خروجی آن مساوی و برابر ۲ باشد (شکل (۶)).



شکل (۵): حلقه موازی



شکل (۶): حلقه تو در تو

۴.۲. ساده‌سازی گراف ترکیب وب‌سرویس‌ها

در این بخش به نحوه ساده‌سازی گراف ترکیب وب‌سرویس‌ها پرداخته شده است. الگوریتم کلی ساده‌سازی گراف، در جریان کاری شکل (۴) آمده است. در فلوچارت شکل (۴) با فراخوانی تابع ساده‌سازی ترکیب گراف، الگوهای توالی، موازی و حلقه به طور مداوم برای ساده‌سازی گراف فراخوانی می‌شوند تا زمانی که دیگر این الگوها در گراف نباشند (به عبارت دیگر $Flag_pattern = 0$). در این صورت اگر نتایج ساده‌سازی تنها منجر به باقی ماندن یک گره در گراف شد (گراف فاقد الگوی شرطی)، الگوریتم ساده‌سازی ترکیب به پایان رسیده است، در غیر این صورت (گراف حاوی الگوی شرطی)، استخراج مسیر و تبدیل گراف به درخت انجام می‌شود که در این صورت درختی با تعدادی مسیر احتمالی ایجاد شده است. آن‌جا که ما برای پاسخ‌گویی به مشتریان باید در آخر مجموعه نهایی از وب‌سرویس‌ها را با بهترین ویژگی‌های کیفی پیشنهاد دهیم و احتمالی بودن هر مسیر، کار را برای ما دشوار کرده است، بنابراین در روش ارائه شده در این مقاله، احتمالات در هر مسیر، به تعداد بار اجرا در هر مسیر تبدیل شده است و الگوریتم انتخاب (در بخش ۳، ۴ به آن پرداخته شده است) برای هر مسیر به طور مجزا به تعداد Max_Run اجرا شده است (شکل (۴)). با این روش، تعداد اجراهای هر مسیر (که از ضرب احتمال آن مسیر در بیشینه اجرا که عدد ۳۰ است، بدست می‌آید) در تعیین جواب بهینه توسط الگوریتم نقش اساسی داشته بدون اینکه نیاز به ضرب احتمالات در مقادیر کیفی داشته باشیم، هم‌چنین با این روش، راه‌حل‌ها (مجموعه‌ای از کاندیدا و مقدار کیفی حاصل از آن مجموعه) و بهترین مسیر از بین مجموعه‌ای از مسیرها انتخاب شده است.

۴.۳. انتخاب بهینه وب‌سرویس‌ها

پس از ساده‌سازی گراف، به منظور انتخاب بهینه وب‌سرویس‌ها الگوریتم‌های تکاملی چون الگوریتم NSGAI یا SPEAII فراخوانی می‌شوند. برای استفاده از این الگوریتم‌ها باید (۱)

انجام آن سرویس انتزاعی را داراست انتخاب می‌شود. پس از یافتن یک عضو از جمعیت، با فراخوانی الگوریتم ترکیب (شکل (۴)) گراف ترکیب ساده می‌شود که در این حین، مقادیر کیفی گراف ترکیب نیز به روزرسانی می‌شود و *Spop. Quality* مقدار می‌گیرد، این متغیر که حاصل ساده-سازی گراف ترکیب وب سرویس است به صورت $Spop. Quality = (a, r, c)$ تعریف می‌شود که a ، r و c به ترتیب مقادیر دسترس پذیری، زمان پاسخ و هزینه حاصل از ساده‌سازی گراف است. این ساده‌سازی و یافتن مقادیر کیفی، برای تمام *Npop* راه‌حل انجام می‌شود.

۱۰	۶۰	۱۳۰	۱۸۷	۲۳۴	۴۸۰
----	----	-----	-----	-----	-----

شکل (۷): نمایش یک راه‌حل با ۶ وظیفه

• تابع برازندگی

پس از تعیین اعضای جمعیت، به منظور تعیین بهترین راه‌حل‌ها (اعضای جمعیت) نیاز به تابع برازندگی است. از آنجا که در مسئله ترکیب وب سرویس‌ها هر راه‌حل به وسیله کیفیت مجموعه سرویس‌های کاندید آن اندازه‌گیری می‌شود. بنابراین تابع برازندگی برای الگوریتم NSGAII و SPEAII با استفاده از *Spop. Quality* تعریف می‌شود. در ادامه تابع برازندگی در الگوریتم NSGAII و سپس SPEAII تعریف شده است:

برای هر دو عضو i و j از مجموعه *Spop*، عضو i بر عضو j غلبه می‌کند اگر: $Spop_i. Quality < Spop_j. Quality$ (مفهوم غلبه در بخش ۲،۱ در معادله (۱) تعریف شده است). هم‌چنین فاصله ازدحامی برای عضو i ، در معادلات (۱۰) و (۱۱) بازنویسی شده است. مقادیر پارامتر k در این مساله از ۱ تا ۳ است.

$$d_{i,k} = \frac{Spop_{i+1}.Quality_k - Spop_{i-1}.Quality_k}{Spop_{Npop}.Quality_k - Spop_1. Quality_k} \quad (10)$$

$$CD_i = \sum_{k=1}^3 d_{i,k} \quad (11)$$

تابع برازندگی در الگوریتم SPEAII، با رابطه (۱۲) تعریف شده است که R و D در رابطه‌های (۱۳)، (۱۴) و (۱۵) تعریف شده‌اند.

جمعیت اولیه (مجموعه راه‌حل‌ها)، (۲) تابع برازندگی مورد استفاده برای مساله ترکیب وب سرویس‌ها تعریف شوند.

• جمعیت اولیه

قبل از توضیح درباره چگونگی ایجاد جمعیت اولیه، ابتدا مقادیر کیفی هر کاندید را در بازه صفر و یک مقیاس می‌کنیم. مقادیر هر خصیصه کیفی با تقسیم بر بیشترین مقدار آن خصیصه در بازه بین ۰ و ۱ قرار می‌گیرد. برای مثال اگر $C_i = (a_i, r_i, c_i)$ کاندید i ام باشد و مجموعه‌های $A = \{a_1, a_2, \dots, a_{NC}\}$ ، $R = \{r_1, r_2, \dots, r_{NC}\}$ و $C = \{c_1, c_2, \dots, c_{NC}\}$ به ترتیب شامل خصیصه‌های اول، دوم و سوم همه‌ی کاندیدای موجود باشند، آنگاه مقادیر مقیاس شده خصیصه‌های کاندید i ام، طبق روابط (۷)، (۸) و (۹) بدست می‌آیند. در مجموعه‌های A ، R و C متغیر NC تعداد کل کاندیدای موجود را نشان می‌دهد.

$$a_i = \frac{a_i}{Max(A)} \quad (7)$$

$$r_i = \frac{r_i}{Max(R)} \quad (8)$$

$$c_i = \frac{c_i}{Max(C)} \quad (9)$$

در مساله ترکیب وب سرویس‌ها، هر عضو از جمعیت به صورت یک زوج $(Index, Quality)$ تعریف می‌شود. *Spop. Index* برداری با Z بعد است که Z برابر با تعداد سرویس‌های انتزاعی است. هر بعد از این بردار، نشان‌دهنده‌ی شماره اندیس کاندیدای موجود برای آن سرویس انتزاعی است. برای مثال $Spop1.index = \{10, 60, 130, 187, 234, 480\}$ نشان‌دهنده‌ی اندیس کاندیدای عضو اول جمعیت در گراف ترکیب وب سرویسی با ۶ وب سرویس انتزاعی است. در این مثال برای سرویس انتزاعی اول سرویس کاندید شماره ۱۰، سرویس انتزاعی دوم سرویس کاندید شماره ۶۰ و به همین ترتیب تا آخر انتخاب شده است (شکل (۷)). پس از تعریف و نمایش هر عضو از جمعیت، نحوه‌ی مقدار دهی به عناصر جمعیت در ابتدای الگوریتم مطرح می‌شود. در شروع الگوریتم برای هر بعد (سرویس انتزاعی) از یک عضو، به صورت تصادفی یک سرویس کاندید از انباره سرویس‌ها که قابلیت

توالی و موازی برابر $O(|v|)$ و برای الگوی حلقه $O(|v| + |e|)$ است زیرا در شناسایی حلقه از الگوریتم جستجوی اول عمق استفاده شده است. پس از شناسایی و حذف الگوها، برای تبدیل گراف به درخت از الگوریتم جستجوی اول سطح استفاده شده است که پیچیدگی زمانی آن نیز برابر $O(|v| + |e|)$ است. بنابراین برای ایجاد $Npop$ عضو، این پیچیدگی برابر است با:

$$Npop \times O(|v| + |e|)$$

(۲) بخش انتخاب: در این بخش از الگوریتم‌های تکاملی NSGAI و SPEAII برای انتخاب بهینه کاندیدا استفاده می‌کنیم. این الگوریتم‌ها به تعداد $Max_iteration$ بار تکرار می‌شوند و در هر بار برانزندی هر عضو جمعیت محاسبه و سپس بر اساس آن اعضا مرتب می‌شوند. پیچیدگی زمانی این مرتب‌سازی برابر $O(Npop \times \log(Npop))$ است. بنابراین پیچیدگی زمانی الگوریتم انتخاب برابر است با:

$$O(Npop \times \log(Npop) \times Max_iteration)$$

پیچیدگی زمانی کل:

$$O(Npop^2 \times \log(Npop) \times Max_iteration \times (|v| + |e|))$$

۵. آزمایشات و ارزیابی نتایج

به منظور ارزیابی روش پیشنهادی، سه نوع گراف ترکیب وب-سرویس (شکل (۱۰)) در نظر گرفته شده است. شکل (۱۰)-الف و (۱۰)-ب گراف‌های بدون الگوی شرطی که اولی شامل الگوهای موازی و حلقه ساده و دومی دارای الگوی حلقه تودرتو است. شکل (۱۰)-ب نیز گرافی دارای الگوی شرطی است. هم‌چنین برای کاندیدای هر وب‌سرویس انتزاعی از مجموعه داده‌ی QWS [۳۳] استفاده شده است. این مجموعه داده دارای ۲۵۰۷ وب‌سرویس کاندید و مقادیر ویژگی‌های کیفی آن وب‌سرویس‌هاست که در طول سال ۲۰۰۸ با استفاده از چارچوب کارگزار وب‌سرویس^۱ اندازه‌گیری شده است. هر سطر در این مجموعه داده نشان‌دهنده میانگین مقادیر نه ویژگی کیفی برای هر وب‌سرویس است که با استفاده از ابزارهای

$$F(i) = R(i) + D(i) \quad (۱۲)$$

$$S(i) = |\{j|j \in Spop \cup sArch \text{ and } Spop_i.Quality > Spop_j.Quality\}| \quad (۱۳)$$

$$R(i) = \sum_{j \in Sp + sArch, Spop_j.Quality > Spop_i.Quality} S(j) \quad (۱۴)$$

$$D(i) = \frac{1}{\partial_i^k + 2}, K = \sqrt{Npop + sArch} \quad (۱۵)$$

پس از یافتن برانزندی هر عضو از جمعیت، اعضای جمعیت (راه‌حل‌ها) که شامل جفت دوتایی $(Index, Quality)$ هستند مرتب و اعضای اضافی در صورت وجود حذف می‌شوند.

• جمعیت جدید

جمعیت جدید در مساله ترکیب وب‌سرویس‌ها با تقاطع و جهش ایجاد می‌شود (شکل‌های (۸) و (۹)). این فرآیند، ۱۰۰ بار تکرار می‌شود. در پایان مجموعه‌ای از بهترین راه‌حل‌ها به عنوان خروجی چاپ می‌شوند (شکل (۴)).

۱۰	۶۰	۱۳۰	۱۸۷	۲۳۴	۴۸۰
عضو اول					
۵	۸۹	۱۷۰	۲۱۰	۳۴۱	۵۹۰
عضو دوم					
۱۰	۶۰	۱۳۰	۱۸۷	۳۴۱	۵۹۰

شکل (۸): جمعیت جدید حاصل از تقاطع دو عضو

۱۰	۶۰	۱۹۰	۱۸۷	۲۳۴	۵۹۰
----	----	-----	-----	-----	-----

شکل (۹): جمعیت جدید حاصل از جهش عضو اول

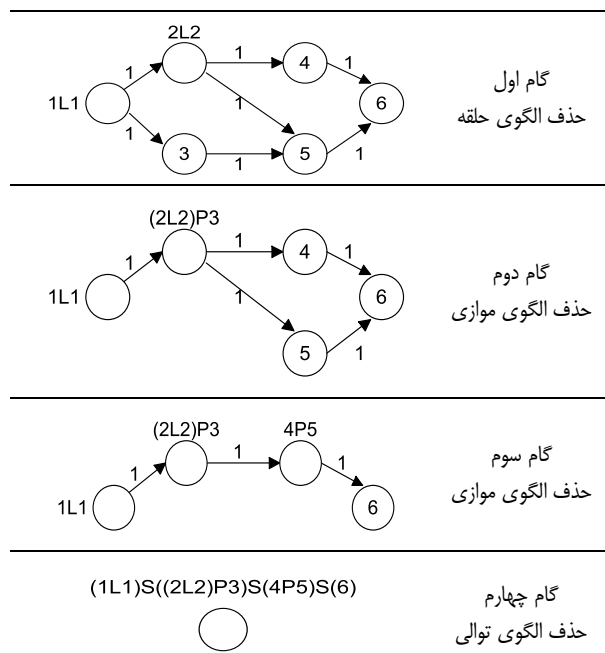
۴.۴. پیچیدگی زمانی

پیچیدگی زمانی روش پیشنهادی در گرافی با v گره (تعداد وب‌سرویس‌ها)، m کاندید برای هر وب‌سرویس و e یال در ۲ بخش محاسبه می‌شود.

(۱) بخش ترکیب: در این بخش ساده‌سازی گراف ترکیب وب-سرویس می‌شود. در ساده‌سازی گراف بحث شناسایی الگوها و محاسبه کیفیت مطرح می‌شود که پیچیدگی زمانی برای الگوهای

^۱ Web Service Broker

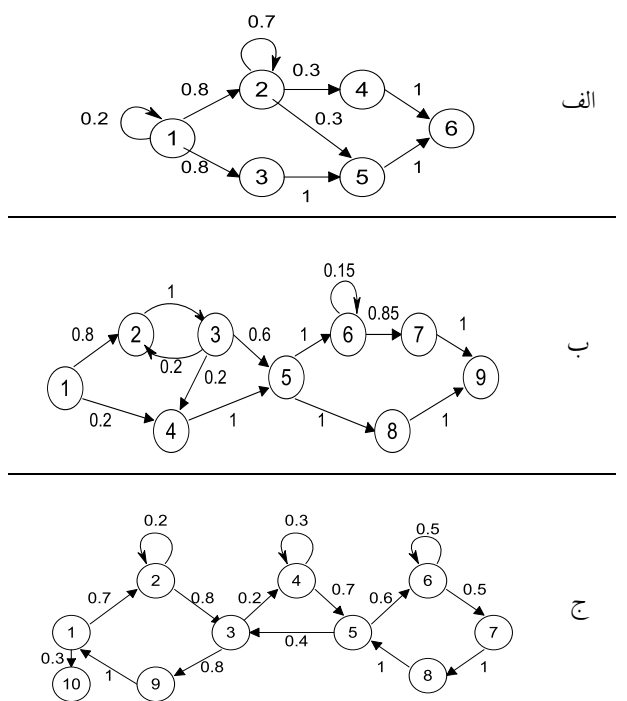
وجود ندارد (گره‌های ۲ و ۳ موازی نیستند چون گره ۲ غیر از خروجی‌های ۴ و ۵، خروجی ۲ نیز دارد و هم‌چنین گره‌های ۴ و ۵ نیز به دلیل اینکه گره ۵ دو ورودی ۲ و ۳ دارد موازی نیستند) بنابراین در مرحله اول الگوی حلقه ساده‌سازی می‌شود (گره ۱ و سپس گره ۲). در مرحله دوم و سوم الگوهای موازی ساده‌سازی می‌شوند و چون این گراف فاقد الگوی شرطی است. در آخر تنها یک گره باقی می‌ماند. نتایج این ساده‌سازی در شکل (۱۱) آمده است.



شکل (۱۱): ساده‌سازی گراف فاقد الگوی شرطی

در شکل (۱۰)-ب، در گام اول الگوی توالی و موازی وجود ندارد بنابراین ابتدا الگوهای حلقه ساده‌سازی می‌شوند. سپس به ترتیب الگوهای توالی، موازی و در آخر نیز الگوی توالی ساده‌سازی می‌شود. از آنجا که این گراف دارای الگوی شرطی است و ساده‌سازی الگوی شرطی منجر به کاهش دقت و صحت الگوریتم می‌شود (به دلیل ضرب احتمالات در مقادیر کیفی کاندیدا در هنگام ساده‌سازی)، بنابراین این الگو ساده‌سازی نمی‌شود و در مرحله آخر مسیرها استخراج می‌شوند که احتمال مسیر اول ۰/۶، احتمال مسیر دوم ۰/۲ و احتمال مسیر سوم ۰/۲ است. نتایج این ساده‌سازی در شکل (۱۲) آمده است. در شکل (۱۰)-ج گراف دارای الگوی توالی، حلقه ساده و حلقه

مخصوص وب‌سرویس در طی شش روز اندازه‌گیری شده‌اند. این ویژگی‌ها شامل زمان پاسخ، دسترس‌پذیری، بازدهی، توانایی موفق بودن، قابلیت اطمینان، انطباق، تاخیر و مستندات است. روش پیشنهادی با استفاده از متلب ۲۰۱۶ در سیستمی با پردازنده Corei7 و رم ۳۲ گیگابایتی و سیستم عامل ۱۰ اجرا شده است. در این بخش ابتدا دو الگوریتم انتخاب (NSGAI)، (SPEAI) بر اساس پارامترهای در نظر گرفته شده برای هر وب‌سرویس، شاخص‌های عملکردی و هم‌چنین روش‌های آماری مقایسه شده‌اند، سپس روش پیشنهادی با استفاده از الگوریتم برتر با روش [۲۵] که بیش‌ترین شباهت را به کار ارائه شده در این مقاله دارد، مقایسه شده است

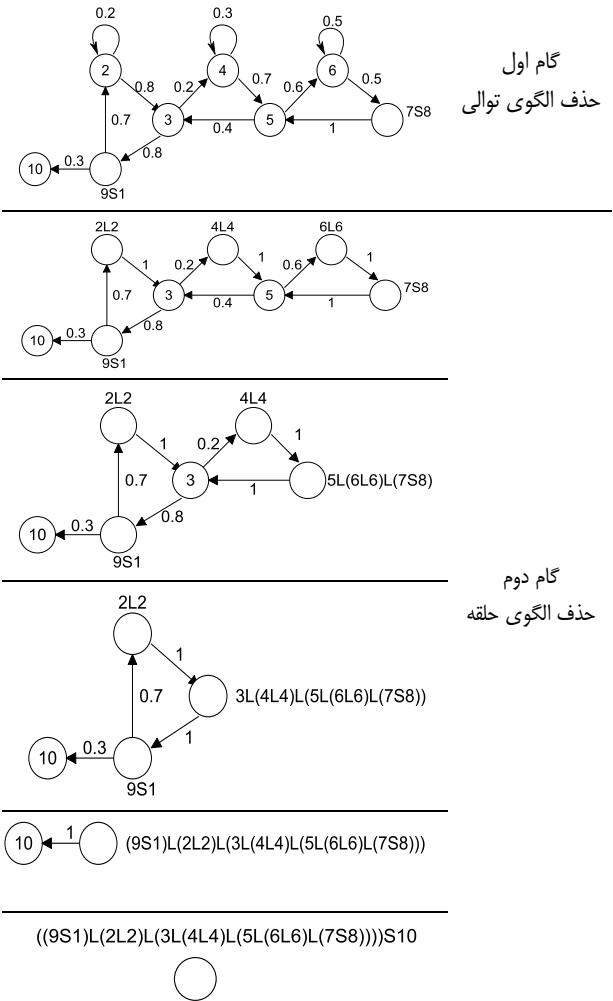


شکل (۱۰): گراف‌های مورد استفاده

۵.۱. ساده‌سازی گراف ترکیب وب‌سرویس

همان‌طور که در بخش ۳، ۱، ۲ گفته شد روش پیشنهادی در این مقاله بر مبنای روش ترکیبی است و بر اساس نوع گراف که فاقد الگوی شرطی (گراف ۱-۱۰ و ۲-۱۰) و یا دارای الگوی شرطی (شکل ۱۰)-ب باشد ساده‌سازی انجام شده است. در شکل (۱۰)-الف الگوی توالی وجود ندارد، الگوی توازی نیز

نود باقی‌مانده در آن مسیر، گراف را ساده‌تر می‌کنیم تا به یک گره برسیم. سپس الگوریتم انتخاب به تعداد Max_Run برای کل گراف اجرا می‌شود. روش پیشنهادی: در این روش بجای ضرب احتمالات در مقادیر کیفی برای هر مسیر استخراج شده، الگوریتم برای هر مسیر به تعداد (احتمال آن مسیر $\times Max_Run$) اجرا می‌شود.

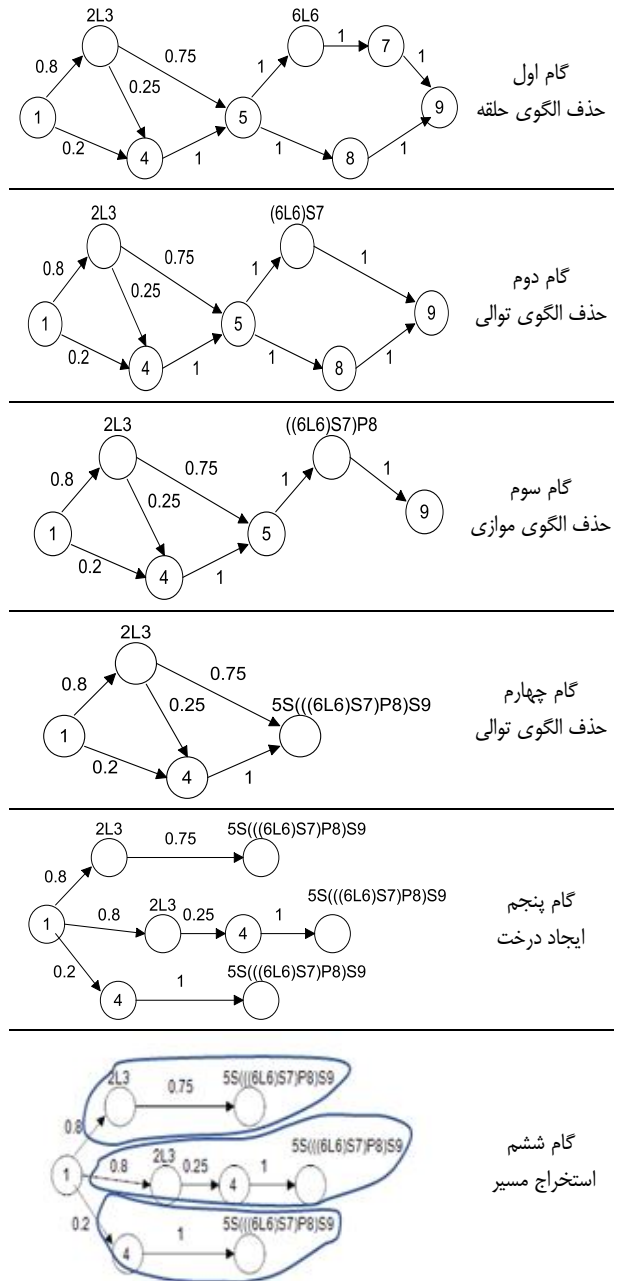


شکل (۱۳): ساده‌سازی گراف فاقد الگوی شرطی (دارای حلقه تو در تو)

۵.۲. مقایسه نتایج الگوریتم‌های NSGAII و SPEAII

مقادیر پارامترهای مورد استفاده در این دو الگوریتم در جدول (۴) آمده است. در این آزمایش هر الگوریتم به تعداد Max_Run اجرا می‌شود [۲۵]. برای مقایسه دقیق‌تر دو الگوریتم، جنبه پرتو هم از لحاظ پارامترها (مشخصه‌های کیفی) و هم بر اساس شاخص‌های سنجش کیفیت ارزیابی شده‌اند.

تودرتو است که ابتدا به حذف الگوی توالی و سپس به حذف حلقه تکی و در آخر به حذف حلقه تو در تو پرداخته شده است. نتایج این ساده‌سازی در شکل (۱۳) آمده است.



شکل (۱۲): ساده‌سازی گراف دارای الگوی شرطی

به طور کلی دو روش برای ساده‌سازی گراف‌های دارای الگوی شرطی استفاده شده است. روش ترکیبی احتمالی: در این روش پس از استخراج مسیرها ابتدا طبق الگوی توالی هر مسیر ساده می‌شود و سپس با ضرب احتمالات هر مسیر در مقادیر کیفی

جدول (۴): مقادیر پارامترهای مورد استفاده در الگوریتم‌های انتخاب

پارامتر	مقدار
اندازه جمعیت اولیه	۱۰۰
اندازه آرشیو (SPEAII)	۱۰۰
بیشترین نسل	۱۰۰
بیشترین تکرار	۱۰۰
تعداد دفعات اجرا	۳۰
احتمال جهش	۰/۲
احتمال تقاطع	۰/۸

۵.۲.۱. مقایسه نتایج بر اساس پارامترها

با مقایسه مقادیر دسترس پذیری، زمان پاسخ و هزینه (شکل‌های (۱۴)، (۱۵) و (۱۶)) جبهه‌های پرتو حاصل از الگوریتم‌های مختلف، این نتیجه حاصل می‌شود که میانگین پارامتر دسترس پذیری در هر دو الگوریتم SPEAII و NSGAII برای هر سه گراف حدوداً یکسان است. میانگین پارامتر زمان پاسخ در NSGAII کمتر از SPEAII و میانگین پارامتر هزینه در SPEAII کمتر از NSGAII است. از آنجا که تنها با مقایسه میانگین پارامترها در *Max_Run* اجرا نمی‌توان برتری الگوریتمی را بر دیگری تشخیص داد بنابراین در ادامه با استفاده از شاخص‌های دیگر سنجش کیفیت و هم‌چنین روش‌های آماری به این مهم رسیدگی می‌شود.

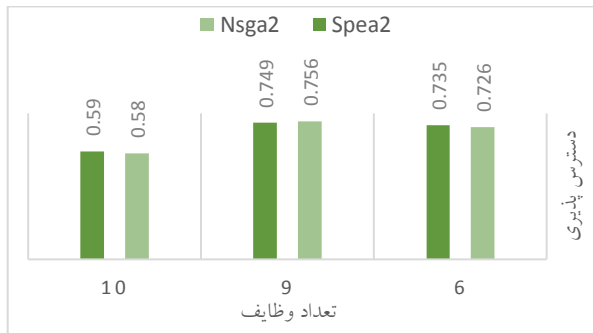
۵.۲.۲. مقایسه نتایج بر اساس شاخص‌های کارایی

در این بخش به مقایسه نتایج بر حسب شاخص‌های مختلف می‌پردازیم.

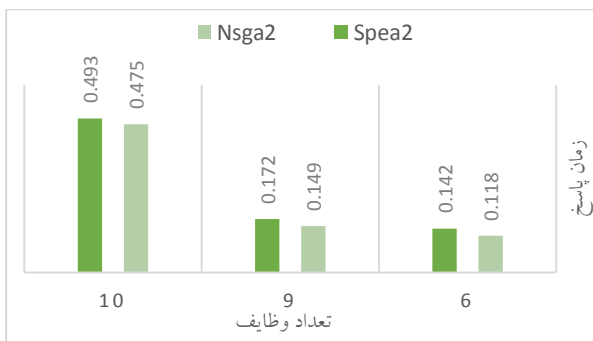
• شاخص نرخ پوشش‌دهی

این شاخص که یکی از مهم‌ترین شاخص‌ها برای مقایسه جبهه‌های پرتو حاصل از الگوریتم‌هاست، نشان می‌دهد چند درصد از راه‌حل‌های الگوریتم دوم توسط راه‌حل‌های الگوریتم اول مغلوب می‌شوند. شکل (۱۷) نرخ پوشش‌دهی دو الگوریتم به یکدیگر را نشان می‌دهد. همان‌طور که در این شکل مشخص است نرخ پوشش‌دهی الگوریتم NSGAII در هر سه گراف از الگوریتم SPEAII بیشتر است. این به این معناست که برای مثال در گرافی با ۶ وب سرویس، الگوریتم NSGAII ۹۱ درصد

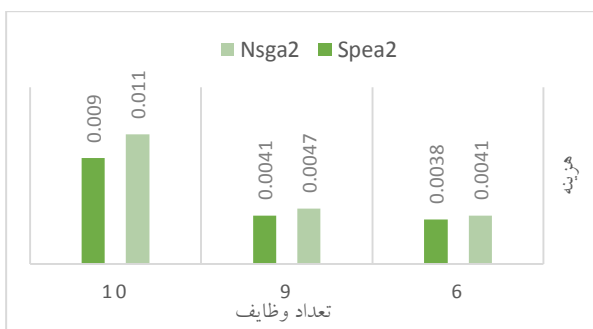
راه‌حل‌های به دست آمده از الگوریتم SPEAII را مغلوب می‌کند در صورتی که این الگوریتم ۸۵ درصد از راه‌حل‌های الگوریتم NSGAII را مغلوب می‌کند. بنابراین در ارزیابی با این شاخص الگوریتم NSGAII بهتر عمل کرده است.



شکل (۱۴): میانگین دسترس پذیری در گراف‌هایی با ۶ و ۹ و ۱۰ وب سرویس انتزاعی



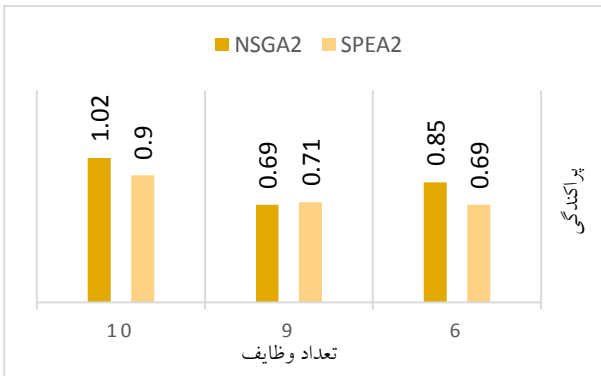
شکل (۱۵): میانگین زمان پاسخ در گراف‌هایی با ۶ و ۹ و ۱۰ وب سرویس انتزاعی



شکل (۱۶): میانگین هزینه در گراف‌هایی با ۶ و ۹ و ۱۰ وب سرویس انتزاعی

• شاخص توزیع شدگی

این شاخص میانگین فاصله راه‌حل‌های بدست آمده در جبهه پرتو را از یکدیگر نشان می‌دهد. هر چه میانگین فاصله راه‌حل‌ها از یکدیگر کمتر باشد، داده‌ها بهتر توزیع شده‌اند. همان‌طور که



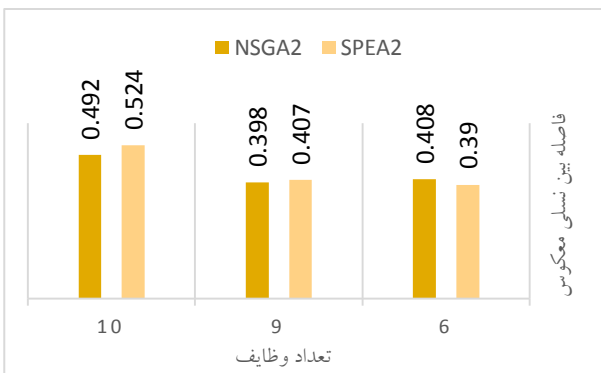
شکل (۱۹): مقایسه پراکندگی در گراف‌هایی با ۱۰ و ۹ و ۶ وب‌سرویس انتزاعی

• شاخص فاصله بین نسلی معکوس

این شاخص میانگین فاصله بین راه‌حل‌های موجود در جبهه‌ی پرتو بهینه و جبهه پرتو بدست آمده از الگوریتم را اندازه می‌گیرد. هر چه این فاصله کمتر باشد نشان‌دهنده‌ی نزدیکی راه‌حل‌های جبهه پرتو به راه‌حل‌های بهینه است. نتایج در شکل (۲۰) نشان داده شده است. این فاصله در گراف‌های ۹ و ۱۰ در الگوریتم NSGAII کمتر از SPEAII است و در گراف با ۶ وب‌سرویس برعکس است که نشان‌دهنده عملکرد بهتر الگوریتم NSGAII نسبت به SPEAII است.

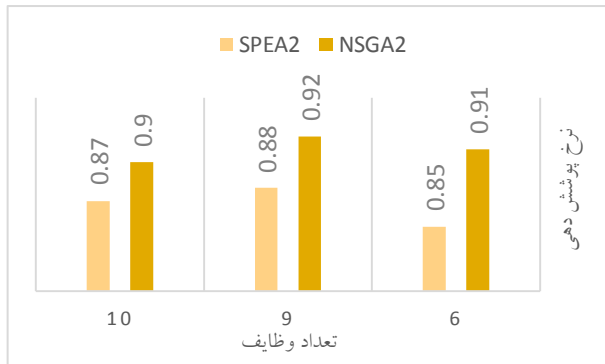
• شاخص زمان اجرا

مدت زمانی که طول می‌کشد تا الگوریتم جواب‌های بهینه را تولید کند. همان‌طور که در شکل (۲۱) مشخص است الگوریتم SPEAII در هر سه نوع گراف زمان اجرای کمتری دارد، البته این اختلاف زیاد نیست.

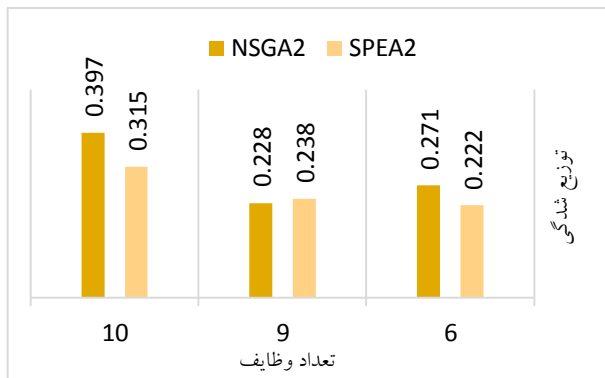


شکل (۲۰): مقایسه فاصله بین نسلی معکوس در گراف‌هایی با ۱۰ و ۹ و ۶ وب‌سرویس انتزاعی

شکل (۱۸) نشان می‌دهد در گرافی با ۶ و ۱۰ وب‌سرویس توزیع شدگی راه‌حل‌های بدست آمده از الگوریتم SPEAII کوچکتر از الگوریتم NSGAII است ولی در گراف با ۹ وب‌سرویس برعکس است. بنابراین در ارزیابی با این شاخص نمی‌توان نظر قطعی داد.



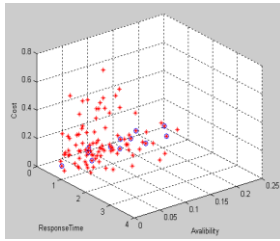
شکل (۱۷): نرخ پوشش دهی در گراف‌هایی با ۱۰ و ۹ و ۶ وب‌سرویس انتزاعی



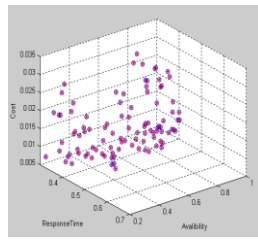
شکل (۱۸): مقایسه توزیع شدگی در گراف‌هایی با ۱۰ و ۹ و ۶ وب‌سرویس انتزاعی

• شاخص پراکندگی

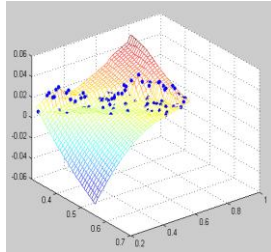
این شاخص فاصله بین راه‌حل‌های مرزی را نشان می‌دهد. هر چقدر این شاخص بیشتر باشد نشان‌دهنده‌ی این است که نقاط مرزی به خوبی پوشش داده شده‌اند. همان‌طور که در شکل (۱۹) مشخص است، در گرافی با ۶ و ۱۰ وب‌سرویس پراکندگی راه‌حل‌های بدست آمده از الگوریتم NSGAII بزرگتر از الگوریتم SPEAII است ولی در گراف با ۹ وب‌سرویس برعکس است، البته این اختلاف در در گراف با ۹ وب‌سرویس بسیار کم است. بنابراین می‌توان نتیجه گرفت که الگوریتم NSGAII تا حدودی در ارزیابی با این شاخص عملکرد بهتری داشته است.



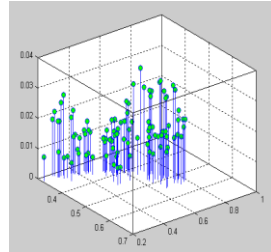
ب: جمعیت اولیه در ابتدای اجرای الگوریتم



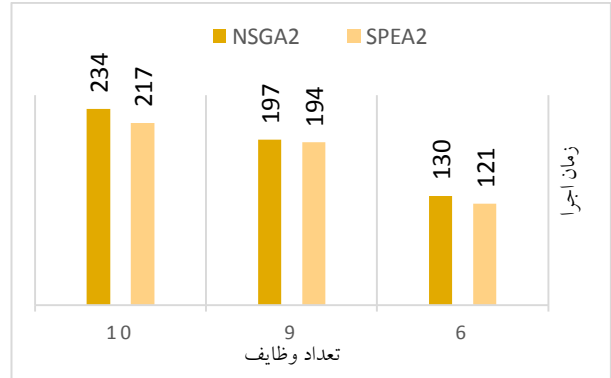
الف: جمعیت بهینه پس از ۱۰۰ دور تکرار الگوریتم



د: نمودار مش جمعیت بهینه پس از ۱۰۰ دور تکرار الگوریتم



ج: نمودار سه بعدی جمعیت بهینه پس از ۱۰۰ دور تکرار الگوریتم



شکل (۲۱): مقایسه زمان اجرا دو الگوریتم در گراف هایی با ۶، ۹ و ۱۰ وب سرویس انتراعی

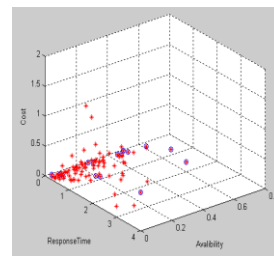
در ادامه به منظور داشتن دید بهتر از مجموعه جواب‌های الگوریتم‌ها، نمودار سه بعدی و مش هر دو الگوریتم در شکل‌های (۲۲) و (۲۳) نمایش داده شده است. ستاره‌های قرمز نشان‌دهنده جمعیت اولیه و ستاره‌های قرمز که دایره آبی دارند اعضای پرتو را نشان می‌دهند. همان‌طور که در شکل‌ها مشخص است در هر دو شکل جمعیت به سمت دسترس‌پذیری بیشتر و زمان پاسخ و هزینه کمتر پیش‌روی کرده است. همچنین شکل‌ها نشان‌دهنده دسترس‌پذیری بیشتر و زمان پاسخ کمتر اغلب اعضای جبهه پرتو الگوریتم NSGAII نسبت به الگوریتم SPEAII و هم‌چنین هزینه کمتر الگوریتم SPEAII نسبت به NSGAII است.

شکل (۲۳): نتایج حاصل از اجرای الگوریتم SPEAII

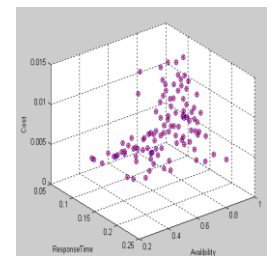
به منظور مقایسه بهتر دو الگوریتم جبهه پرتو حاصل از هر دو الگوریتم در شکل (۲۴) نمایش داده شده است. ستاره‌های آبی جبهه پرتو حاصل از الگوریتم NSGAII و ستاره‌های سبز جبهه پرتو حاصل از الگوریتم SPEAII هستند.

۵.۳. تحلیل آماری نتایج

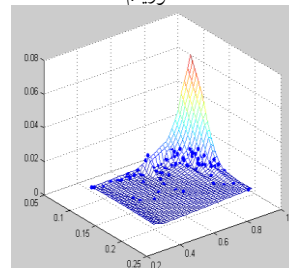
همان‌طور که در شکل (۳) نشان داده شد، الگوریتم را ۳۰ بار با مجموعه داده تکرار و جبهه پرتو حاصل را با آزمون t-test تحلیل می‌کنیم. نتایج حاصل از این آزمون در پیوست مقاله آمده است که در آن شماره‌های ۱ و ۳ به ترتیب نشان‌دهنده الگوریتم NSGAII و SPEAII است. با توجه به اینکه مقدار Sig در پارامتر دسترس‌پذیری، ۰/۳۹۷ و بزرگتر از ۰/۰۵ است، تفاوت معناداری بین این دو روش برای این پارامتر وجود ندارد. مقدار Sig برای پارامتر زمان پاسخ، صفر و معرف تفاوت معناداری قابل ملاحظه بین این دو روش برای این پارامتر است (برتری الگوریتم NSGAII نسبت به SPEAII). مقدار Sig برای پارامتر هزینه، برابر ۰/۰۳ و معرف تفاوت معنادار بین این دو الگوریتم در این پارامتر است (برتری الگوریتم SPEAII نسبت به NSGAII).



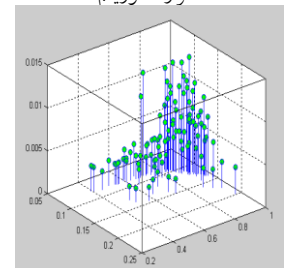
ب: جمعیت اولیه در ابتدای اجرای الگوریتم



الف: جمعیت بهینه پس از ۱۰۰ دور تکرار الگوریتم



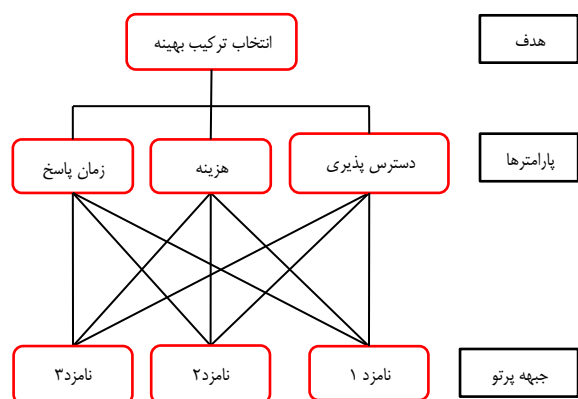
د: نمودار مش جمعیت بهینه پس از ۱۰۰ دور تکرار الگوریتم



ج: نمودار سه بعدی جمعیت بهینه پس از ۱۰۰ دور تکرار الگوریتم

شکل (۲۲): نتایج حاصل از اجرای الگوریتم NSGAII

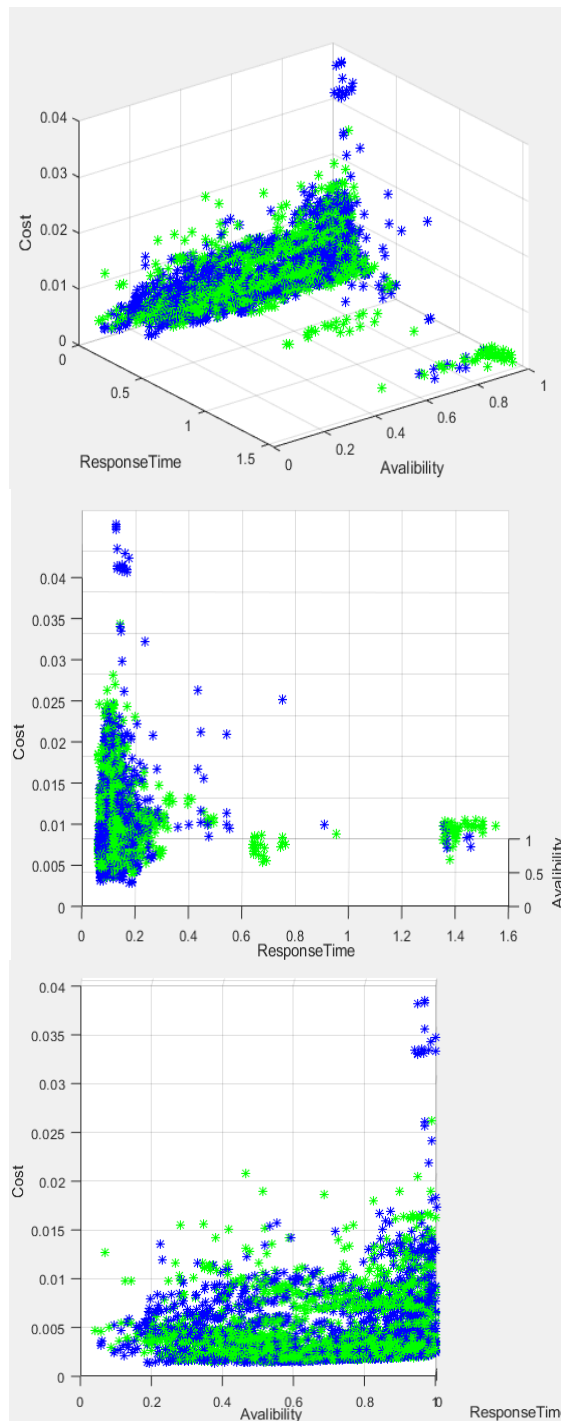
در یک مساله تصمیم‌گیری چند معیاره تعیین می‌کند. در این مساله، سلسله‌مراتب مطابق شکل (۲۵) می‌باشد که در آن سه پارامتر دسترس‌پذیری، زمان پاسخ و هزینه باید بهینه شوند. گزینه‌های پیش‌رو، اعضای جبهه پرتو هستند که در بین آن‌ها یک راه‌حل به عنوان مناسب‌ترین راه‌حل باید انتخاب شود. به منظور تعیین بهترین راه‌حل، به هر پارامتر وزنی می‌دهیم و سازگاری سیستم را بررسی می‌کنیم و با توجه به امتیاز بدست آمده تصمیم‌گیری می‌کنیم. این وزن‌دهی بستگی به اولویت خصیصه‌ها از نظر کاربر دارد، اگر هر ۳ خصیصه دسترس‌پذیری، زمان پاسخ و هزینه برای کاربر از اهمیت یکسانی برخوردار باشند وزن‌دهی یکسان و اگر متفاوت باشند وزن‌دهی متفاوتی برای خصیصه‌ها در نظر گرفته می‌شود. در اینجا ۲ حالت مثال زده شده است. در حالت اول وزن یکسان و در حالت دوم هزینه برای کاربر در تعیین وب‌سرویس‌های کاندید اهمیت خیلی کمی دارد و زمان پاسخ بیشترین اهمیت را دارد. بنابراین به دنبال کاندیدایی در ترکیب وب‌سرویس‌ها هستیم زمان پاسخ کوتاهی داشته باشد که راه‌حل انتخابی برای هر دو الگوریتم در جدول (۶) نشان می‌دهد که برای انتخاب بهترین و مناسب‌ترین کاندیدا ابتدا به زمان پاسخ توجه شده است.



شکل (۲۵): فرآیند سلسله‌مراتبی برای انتخاب ترکیب بهینه

• پارامترها با وزن نسبی یکسان

در این حالت وزن نسبی پارامترها به صورت $W = (0.33, 0.33, 0.33)$ خواهد بود. جدول (۵) نتایج حاصل از آن را نشان می‌دهد. همان‌طور که در جدول مشخص است در این



شکل (۲۴): جبهه پرتو حاصل از دو NSGAII و SPEAII

۵.۴. انتخاب بهترین کاندید با فرآیند تحلیل سلسله

مراتبی

به منظور به‌دست آوردن بهترین راه‌حل بهینه در جبهه پرتو از تحلیل سلسله‌مراتبی استفاده می‌کنیم. این فرآیند بر اساس مقایسه‌های زوجی بنا شده است و اهمیت نسبی چند معیار را

اجرا گزارش شده است. با مقایسه نتایج حاصل از هر مسیر با نتیجه حاصل از ساده‌سازی گراف با روش ترکیبی احتمالی با مقادیر (۰/۰۴۷، ۰/۱۴۹، ۰/۷۵) این نتیجه حاصل می‌شود که مسیر دوم بهترین مسیر برای انتخاب است. نتایج روش پیشنهادی و نتایج روش [۲۵] در جدول (۷) آمده است. علامت "----" معرف عدم گزارش نتیجه به وسیله الگوریتم مربوطه در [۲۵] است. مقایسه نتایج در این جدول نشان می‌دهد که روش پیشنهادی از نظر ویژگی قابلیت اطمینان از همه روش‌ها بهتر و از نظر ویژگی زمان پاسخ فقط از یک روش بدتر عمل کرده است. هم‌چنین در شاخص توزیع شدگی از روش اول و در شاخص فاصله بین نسلی معکوس از روش دوم بهتر است.

جدول ۷: مقایسه روش پیشنهادی و روش تجمعی بالگوریتم‌های انتخاب در [۲۵]

روش‌ها		اهداف		شاخص‌ها	
توزیع شدگی	فاصله بین نسلی معکوس	زمان پاسخ	قابلیت اطمینان	توزیع شدگی	فاصله بین نسلی معکوس
تجمعی	NSGAII	---	---	۰/۵۶۳	۰/۰۱۵
تجمعی	MOPSO	---	---	۰/۶۱۰	۰/۰۲۷
تجمعی	GA with AHP	۱۴/۱	۱۴/۰۸	---	---
تجمعی	PSO with AHP	۱۴/۴	۱۶۴۸	---	---
تجمعی	NSGAII with AHP	۲۸/۷	۱۲۲۴	---	---
تجمعی	NSGAII-AHP	۲۴/۶	۱۱۳۱	---	---
تجمعی	MOPSO with AHP	۱۹/۷	۱۱۹۰	---	---
تجمعی	MOPSO-AHP	۱۹/۳	۱۰۳۶	---	---
ترکیبی با تکرار	NSGAII	۵۸	۱۱۰۳	۰/۴۶۴	۰/۰۱۶

۶. نتیجه‌گیری

در سال‌های اخیر، با رشد سریع ارتباطات و پیچیده‌تر شدن نیازهای کاربران، نیاز به ترکیب وب‌سرویس‌ها بیش از پیش احساس می‌شود. این وب‌سرویس‌ها می‌توانند با الگوهای پیچیده متفاوتی با یکدیگر در ارتباط باشند. بنابراین روشی مناسب برای ساده‌سازی این ترکیبات به منظور انتخاب بهینه وب‌سرویس‌ها برای کمک به کاربران لازم است. بنابراین علاوه

حالت که هر سه خصیصه برای کاربر اهمیت یکسانی دارند، راه‌حل انتخابی الگوریتم NSGAII هم دسترس‌پذیری بیشتر و هم زمان پاسخ و هزینه کمتری نسبت به الگوریتم SPEAII دارد.

جدول (۵): پارامترهای کیفی بر اساس تحلیل سلسله مراتبی با وزن دهی مساوی

الگوریتم	دسترس‌پذیری	زمان پاسخ	هزینه
NSGAII	۰/۸۴	۰/۰۹۷	۰/۰۰۲۳
SPEAII	۰/۸۳	۰/۱۱۶	۰/۰۰۲۴

• پارامترها با وزن نسبی نامساوی

وزن $W = (0.2718, 0.6612, 0.067)$ به عنوان وزن نسبی پارامترها در این حالت در نظر گرفته شده است. نتایج جدول (۶) نشان می‌دهد راه‌حلی که الگوریتم NSGAII پیشنهاد داده است دسترس‌پذیری کمتری و البته هزینه کمتری نسبت به الگوریتم SPEAII دارد.

جدول (۶): پارامترهای کیفی بر اساس تحلیل سلسله مراتبی با وزن دهی نامساوی

الگوریتم‌ها	دسترس‌پذیری	زمان پاسخ	هزینه
NSGAII	۰/۴۰	۰/۰۷۷	۰/۰۰۳۴
SPEAII	۰/۴۱	۰/۰۷۷	۰/۰۰۳۵

۵.۵. مقایسه روش پیشنهادی با سایر روش‌ها

در بخش ۵،۱ به دو روش ساده‌سازی گراف‌های دارای الگوی شرطی (شکل (۱۰) -ب) اشاره شده است. نتایج روش پیشنهادی در شکل (۲۶) نشان داده شده است.



شکل (۲۶): میانگین مقادیر پارامترهای دسترس‌پذیری، زمان پاسخ، هزینه در هر مسیر با روش پیشنهادی

میانگین پارامترها برای مسیر اول پس از $0.6 \times Max_Run$ یعنی پس از ۱۸ بار اجرا و برای مسیر دوم و سوم پس از ۶ بار

بررسی شده است. در آینده می‌توان این مساله را به صورت پویا دنبال کرد، به طوری که محیط دائم در حال تغییر باشد. برای مثال تعداد کاندیدای در دسترس کم و زیاد شوند یا مقادیر کیفی کاندیدا تغییر کنند و یا گراف ترکیب وب‌سرویس‌ها از قبل طراحی نشده باشد. هم‌چنین برای درخواست‌های بلادرنگ باید راهکاری پیدا کرد تا سرعت پاسخگویی به این نیازها بالا رود، برای مثال با موازی‌سازی الگوریتم می‌توان به این مهم دست یافت. هم‌چنین برای انتخاب بهینه کاندیدا می‌توان از الگوریتم‌های تکاملی هدفمندتر با هدف کاهش زمان اجرا استفاده کرد.

تعارض منافع: نویسندگان اعلام می‌کنند که هیچ تعارض منافی ندارند.

بر نیاز به شناخت انواع الگوهای پیچیده در ترکیب وب-سرویس‌ها، نیاز به روش‌هایی جهت ساده‌سازی این ترکیبات و سپس انتخاب بهترین کاندیدا است. در این مقاله علاوه بر معرفی دو الگوی جدید در ترکیب وب‌سرویس‌ها (حلقه‌های تودرتو و موازی)، روشی برای ساده‌سازی ترکیب به منظور افزایش دقت در کیفیت نهایی ترکیب آمده است، هم‌چنین از الگوریتم‌های تکاملی چون NSGAI و SPEAI برای انتخاب بهینه کاندیدا استفاده شده است. در نهایت، برای راحتی کاربر و کمک به او در تصمیم‌گیری، از روش تحلیل سلسله‌مراتبی استفاده شده است که این روش یکی از اعضای جبهه پرتو را به عنوان عضو برتر شناسایی می‌کند و به کاربر پیشنهاد می‌دهد.

۷. کارهای آینده

در این مقاله، مساله ترکیب وب‌سرویس‌ها به صورت ایستا

مراجع

- [۱] رسولزادگان ع.، بصیری م.، «اندازه‌گیری کمی کیفیت در مهندسی نرم افزار سرویس‌گرا: روش‌ها، کاربردها و چالش‌ها»، مجله محاسبات نرم، جلد ۳، شماره ۱، ص ۲-۱۹، ۱۳۹۳.
- [۲] دهقانی م.، عمادی س.، «ارائه یک مدلی جدید برای بلوغ حاکمیت بر معماری سرویس‌گرا»، مجله محاسبات نرم، جلد ۴، شماره ۲، ص ۵۴-۶۷، ۱۳۹۴.
- [3] Chen, F., Dou, R., Li, M., & Wu, H., A flexible QoS-aware Web service composition method by multi-objective optimization in cloud manufacturing, *Computers & Industrial Engineering*, 99, 423-431, 2016.
- [4] Strunk, A., QoS-aware service composition: A survey, In *2010 Eighth IEEE European Conference on Web Services* (pp. 67-74), IEEE, 2010.
- [5] Zheng, H., Zhao, W., Yang, J., & Bouguettaya, A., QoS analysis for web service compositions with complex structures, *IEEE Transactions on Services Computing*, 6(3), 373-386, 2012.
- [6] Zheng, H., Zhao, W., Yang, J., & Bouguettaya, A., QoS analysis for web service composition, In *2009 IEEE International Conference on Services Computing* (pp. 235-242), IEEE, 2009.
- [7] Brahmī, Z., & Gammoudi, M. M., QoS-aware automatic web service composition based on cooperative agents, In *2013 Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises* (pp. 27-32), IEEE, 2013.
- [8] Canfora, G., Di Penta, M., Esposito, R., & Villani, M. L., A lightweight approach for QoS-aware service composition. In *Proc. 2nd International Conference on Service Oriented Computing (ICSOC'04)-short papers*, 2004.
- [9] Alrifai, M., Risse, T., Dolog, P., & Nejdl, W., A scalable approach for qos-based web service selection. In *International conference on service-oriented computing* (pp. 190-199). Springer, Berlin, Heidelberg, 2008.
- [10] Liu, H., Zhong, F., Ouyang, B., & Wu, J., An approach for QoS-aware web service composition based on improved genetic algorithm. In *2010 International conference on web information systems and mining* (Vol. 1, pp. 123-128). IEEE, 2010.
- [11] Gohain, S., & Paul, A., Web service composition using PSO—ACO. In *2016 International*

- conference on recent trends in information technology (ICRTIT) (pp. 1-5). IEEE, 2016.
- [12] Liu, S., Liu, Y., Jing, N., Tang, G., & Tang, Y., A dynamic web service selection strategy with QoS global optimization based on multi-objective genetic algorithm. In *International Conference on Grid and Cooperative Computing* (pp. 84-89). Springer, Berlin, Heidelberg, 2005.
- [13] Yao, Y., & Chen, H., A rule-based web service composition approach. In *2010 Sixth International Conference on Autonomic and Autonomous Systems* (pp. 150-155). IEEE, 2010.
- [14] Li, L., Yang, P., Ou, L., Zhang, Z., & Cheng, P., Genetic algorithm-based multi-objective optimisation for QoS-aware web services composition. In *International Conference on Knowledge Science, Engineering and Management* (pp. 549-554). Springer, Berlin, Heidelberg, 2010.
- [15] Huo, Y., Qiu, P., Zhai, J., Fan, D., & Peng, H., Multi-objective service composition model based on cost-effective optimization. *Applied Intelligence*, 48(3), 651-669, 2018.
- [16] Yilmaz, A. E., & Karagoz, P., Improved genetic algorithm based approach for QoS aware web service composition. In *2014 IEEE international conference on web services* (pp. 463-470). IEEE, 2014.
- [17] Ardagna, D., & Pernici, B., Global and local QoS guarantee in web service selection. In *International Conference on Business Process Management* (pp. 32-46). Springer, Berlin, Heidelberg, 2005.
- [18] Tang, M., & Ai, L., A hybrid genetic algorithm for the optimal constrained web service selection problem in web service composition. In *IEEE Congress on Evolutionary Computation* (pp. 1-8). IEEE, 2010.
- [19] Canfora, G., Di Penta, M., Esposito, R., & Villani, M. L., An approach for QoS-aware service composition based on genetic algorithms. In *Proceedings of the 7th annual conference on Genetic and evolutionary computation* (pp. 1069-1075), 2005.
- [20] Mukherjee, D., Jalote, P., & Nanda, M. G., Determining QoS of WS-BPEL compositions. In *International Conference on Service-Oriented Computing* (pp. 378-393). Springer, Berlin, Heidelberg, 2008.
- [21] Yao, Y., & Chen, H., QoS-aware service composition using nsga-ii. In *Proceedings of the 2nd international conference on interaction sciences: information technology, culture and human* (pp. 358-363), 2009.
- [22] Dumas, M., García-Bañuelos, L., Polyvyanyy, A., Yang, Y., & Zhang, L., Aggregate quality of service computation for composite services. In *International Conference on Service-Oriented Computing* (pp. 213-227). Springer, Berlin, Heidelberg, 2010.
- [23] Li, J. Z., Luo, W. L., Jin-tao, Z., & Jie-wu, X., Application of SPEA2 algorithm in web services selection. In *2010 IEEE Youth Conference on Information, Computing and Telecommunications* (pp. 387-390). IEEE, 2010.
- [24] Sharifara, P., Yari, A., & Kashani, M. M. R., An evolutionary algorithmic based web service composition with quality of service. In *7th International Symposium on Telecommunications (IST'2014)* (pp. 61-65). IEEE, 2014.
- [25] Liu, L., & Zhang, M., Multi-objective optimization model with AHP decision-making for Cloud service composition. *KSI Transactions on Internet and Information Systems (TIIS)*, 9(9), 3293-3311, 2015.
- [26] Deb, K., Pratap, A., Agarwal, S., & Meyarivan, T. A. M. T., A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE transactions on evolutionary computation*, 6(2), 182-197, 2002.
- [27] Radziukynienė, I., & Žilinskas, A., Evolutionary methods for multi-objective portfolio optimization. In *Proceedings of the World Congress on Engineering* (Vol. 2, pp. 1155-1159), 2008.
- [28] Zitzler, E., Laumanns, M., & Thiele, L., SPEA2: Improving the strength Pareto evolutionary algorithm. *TIK-report*, 103, 2001.
- [29] Sadouki, S. C., & Tari, A., Multi-objective and discrete Elephants Herding Optimization algorithm for QoS aware web service composition. *RAIRO-Operations Research*, 53(2), 445-459, 2019.
- [30] Seghir, F., FDMOABC: fuzzy discrete multi-objective artificial bee colony approach for solving the non-deterministic QoS-driven web service composition problem. *Expert Systems with Applications*, 167, 114413, 2021.
- [31] Xie, N., Tan, W., Zheng, X., Zhao, L., Huang, L., & Sun, Y., An efficient two-phase approach for reliable collaboration-aware service composition in cloud manufacturing. *Journal of Industrial Information Integration*, 23, 100211, 2021.
- [32] Thangaraj, P., & Balasubramanie, P., Meta

heuristic QoS based service composition for service computing. *Journal of Ambient Intelligence and Humanized Computing*, 12(5), 5619-5625, 2021.

[33] <https://qwsdata.github.io/>

پیوست

Descriptive		%95 CONFIDENCE INTERVAL FOR MEAN							
	N	Mean	Std. deviation	Std. error	Lower Bound	Upper Bound	Minimum	Maximum	
AVAILABILITY	1	2695	.7260385095	.2244413228	.0043233803	.7175610312	.7345159879	.572839003	1.000000000
	3	2820	.7359804576	.2079333663	.0039156118	.7283027029	.7436582122	.417818478	1.000000000

Scheffe		95% CONFIDENCE INTERVAL						
DEPENDENT VARIABLE	(I) METHOD	(J) METHOD	MEAN DIFFERENCE (I- J)	STD. ERROR	SIG.	Lower Bound	Upper Bound	
AVAILABILITY	1	3	-.009941948	.0049289055	.397	-.025126061	.00052421645	
	3	1	.0099419480	.0049289055	.397	-.005242165	0.0251260606	
RESPONSE TIME	1	3	-.023820666	.0024004172	.000	-.031215453	-.016425879	
	3	1	.023820666	.0024004172	.000	.0164258791	.0312154528	
COST	1	3	0.000268610	.0000663145	.003	0.0000643198	0.0004729002	
	3	1	-.000268610	.0000663145	.003	-.000472900	-.000064320	