



ارائه روشی جهت بهینه‌سازی انتخاب و ترکیب وب‌سرویس‌های آگاه از کیفیت خدمات در سازمان‌های سرویس‌گرا

مهدی فرزندی^۱، فریدون شمس علیی^۲

^۱ دانشجوی کارشناسی ارشد مهندسی فناوری اطلاعات، دانشگاه شهید بهشتی، تهران
mahdifarzandway@gmail.com

^۲ دانشیار دانشکده مهندسی و علوم کامپیوتر، دانشگاه شهید بهشتی، تهران
f_shams@sbu.ac.ir

چکیده

امروزه معماری سرویس‌گرا، انتخاب و ترکیب سرویس‌ها را به عنوان راه‌کاری برای پاسخ‌گویی سریع به نیازمندی‌های پیچیده‌ی مشتریان در دسترس سازمان‌های سرویس‌گرا قرار می‌دهد.

سازمان‌ها برای پاسخ‌گویی سریع‌تر به نیازمندی‌های پیچیده و متغیر مشتریان، به استفاده از فناوری‌های نوظهور در این زمینه، از جمله وب‌سرویس‌ها و انتخاب و ترکیب مناسبی از آن‌ها، پرداختند. با گسترش تمایل سازمان‌ها به استفاده از وب‌سرویس‌ها، به مرور زمان تأمین‌کنندگان وب‌سرویس‌ها افزایش یافتند به همین دلیل وب‌سرویس‌هایی با عملکرد یکسان، ویژگی‌های کیفی متفاوت و روابط بین وب‌سرویس‌ها از جمله، همبستگی، وابستگی و ناسازگاری گسترش یافتند. ولی تاکنون روشی ارائه نشده است که به مسئله‌ی انتخاب و ترکیب وب‌سرویس‌ها با در نظر گرفتن ویژگی‌های کیفی و روابط همبستگی، وابستگی و ناسازگاری بین وب‌سرویس‌ها، بپردازد. در این پژوهش سعی می‌کنیم، از روش‌های قبلی که به وابستگی یا ناسازگاری یا همبستگی در حالت‌های ساده ترکیب وب‌سرویس‌ها (حالت ترتیبی) پرداخته‌اند، استفاده کنیم و به کمک الگوریتم ژنتیک تمام این روش‌ها را در یک روش جامع بهبود و پیشنهاد دهیم تا اینکه همبستگی، وابستگی و ناسازگاری را در حالت‌های پیچیده‌ای که از ترکیب وب‌سرویس‌ها (حالت‌های ترتیبی، موازی، شرطی و حلقوی) ممکن است رخ دهد را نیز پشتیبانی کنیم و وب‌سرویس مرکب مناسب را براساس چندین ویژگی کیفی بیابیم. در انتها به کمک آزمایش‌ها نشان می‌دهیم که به کمک روش پیشنهادی این پژوهش می‌توانیم وب‌سرویس‌های مرکب مناسب‌تری براساس چندین ویژگی کیفی نسبت به روش‌های پیشین بیابیم.

کلمات کلیدی

الگوریتم ژنتیک، وب‌سرویس، انتخاب وب‌سرویس، ترکیب وب‌سرویس، ویژگی کیفی، وابستگی، ناسازگاری، همبستگی

۱- مقدمه

سازمان‌ها تمام فرآیندهای جریان کاری^۱ خود را توسط سرویس‌های درون‌سازمانی برآورده نمی‌سازند، زیرا مطابقت با درخواست‌ها و نیازمندی‌های پیچیده مشتریان و تغییرات زیاد محیط اطراف بسیار سخت و هزینه‌بر است، بنابراین سازمان‌ها قسمتی از جریان کاری درون‌سازمانی خود را توسط سرویس‌های برون‌سازمانی برآورده می‌سازند. یکی از فناوری‌های نوظهور در این زمینه استفاده از وب‌سرویس‌ها است، بنابراین سازمان‌ها برای کاهش هزینه‌های مطابقت خود با نیازمندی‌های متغیر مشتریان از وب‌سرویس‌ها استفاده می‌کنند، به این منظور وب‌سرویس‌هایی با عملکرد مورد نظر خود را شناسایی و انتخاب می‌کنند. با گسترش اینترنت و تمایل سازمان‌ها به استفاده از وب‌سرویس‌ها، به مرور زمان تعداد وب‌سرویس‌هایی با عملکرد یکسان و ویژگی‌های کیفی^۲ متفاوت افزایش یافت، بنابراین مسئله‌ی انتخاب وب‌سرویس‌ها با بهترین ویژگی کیفی برای سازمان‌ها اهمیت پیدا کرد. از طرفی دیگر وب‌سرویس‌های موجود تنها برای یک عملکرد ساخته می‌شوند و تک منظوره هستند بنابراین سازمان‌ها نمی‌توانند برای درخواست‌های پیچیده و چند منظوره مشتریان که نیازمند ترکیب مناسبی از چندین وب‌سرویس هستند، از یک وب‌سرویس که تک منظوره است، استفاده کنند بنابراین مسئله‌ی انتخاب و ترکیب وب‌سرویس‌ها براساس ویژگی‌های کیفی آن‌ها، برای سازمان‌ها به منظور سریع‌تر پاسخ دادن به نیازمندی‌های پیچیده‌ی مشتریان بسیار حائز اهمیت شد.

۲-۲- ترکیب وب‌سرویس‌ها

در ترکیب وب‌سرویس‌های آگاه از کیفیت خدمات (ویژگی کیفی)، از مجموعه‌ی وب‌سرویس‌های ممکن، وب‌سرویس‌های مناسبی را انتخاب و با هم ترکیب می‌کنیم تا اینکه بتوانیم وب‌سرویس مرکب مناسبی را ارائه دهیم، این وب‌سرویس مرکب بایستی ویژگی عملکردی و کیفی مورد نظر مشتری را برآورده سازد، لازم به ذکر است که بهینه‌سازی در انتخاب و ترکیب وب‌سرویس‌های آگاه از کیفیت خدمات (ویژگی کیفی) از مسائل غیرچندجمله‌ای-سخت^۳ است [۲۰،۴].

وب‌سرویس مرکبی که در آژانس‌های مسافرتی استفاده می‌شود، نمونه‌ای کاربردی از مسئله‌ی انتخاب و ترکیب وب‌سرویس‌ها است [۱۸]. هنگامی که مشتری برای مسافرت به وب‌سرویس آژانس مسافرتی مراجعه می‌کند، چندین فعالیت مختلف انجام می‌پذیرد. این فعالیت‌ها شامل رزرو بلیط هواپیما، رزرو هتل، انتخاب نوع بیمه، آگاهی از وضعیت هوا، رزرو اتومبیل و در نهایت پرداخت هزینه‌ی سفر است. هر کدام از این فعالیت‌ها در قالب یک وب‌سرویس انجام می‌پذیرند، ابتدا وب‌سرویس رزرو بلیط هواپیما، صندلی مورد نظر مشتری را در هواپیما رزرو می‌کند. سپس مشتری باید هتل، بیمه و وضعیت هوا را بررسی کند. در مرحله بعد مشتری باید از موسسه‌ی کرایه‌ی اتومبیل در مقصد، خودروی مورد نظر خود را رزرو کند و در نهایت با پرداخت هزینه‌های مربوطه تمامی امکانات ذکر شده برای مشتری رزرو نهایی می‌شوند [۶].



شکل ۲-۱: وب‌سرویس مرکب آژانس مسافرتی

هرچه تعداد فرآیندهای موجود در جریان کار و وب‌سرویس‌های کاندید برای تک‌تک فرآیندها بیشتر باشند، تعداد حالات ممکن ترکیب وب‌سرویس‌ها برای انجام آن جریان کاری بیشتر خواهد شد. اگر تعداد فرآیندهای موجود در جریان کاری برابر m و تعداد وب‌سرویس‌های کاندید برای هر فرآیند برابر n باشد، آنگاه تعداد حالات ممکن ترکیب وب‌سرویس‌ها برای انجام آن جریان کاری برابر با n^m خواهد شد. به همین دلیل انتخاب وب‌سرویس‌های مناسبی که عملکردهای مورد نظر مشتری را فراهم کنند و یافتن ترکیب مناسبی از وب‌سرویس‌ها که علاوه بر برآورده سازی تمامی عملکردهای مورد نظر مشتری از مجموع ویژگی کیفی مناسبی برخوردار باشد، بدون اینکه تمامی حالات ممکن ترکیب را بررسی

۲-۲- کشف، انتخاب و ترکیب وب‌سرویس‌ها

برای یافتن ترکیب مناسبی از وب‌سرویس‌ها نیازمند گذر از دو مرحله هستیم. این دو مرحله عبارتند از: کشف و انتخاب وب‌سرویس‌های مناسب، یافتن ترکیب مناسبی از وب‌سرویس‌ها براساس ویژگی‌های کیفی آن‌ها.

۲-۱- کشف و انتخاب وب‌سرویس

در کشف و انتخاب وب‌سرویس‌ها، ابتدا بایستی وب‌سرویس‌های مورد نظر مشتری از نظر ویژگی عملکردی را کشف کنیم، برای این منظور از UDDI استفاده می‌کنیم، UDDI امکان جستجوی وب‌سرویس را برای ما فراهم می‌آورد [۷].

از میان وب‌سرویس‌های کشف شده، وب‌سرویس‌های دارای ویژگی کیفی مناسب‌تر را انتخاب و در مجموعه‌ی وب‌سرویس‌های ممکن برای بررسی در یافتن ترکیب مناسبی از وب‌سرویس‌ها قرار می‌دهیم [۸].

نمی‌شوند و کمترین یا بیشترین مقدار عددی از بین اعداد ویژگی کیفی انتخاب شده و در روابط قرار داده می‌شوند.

$$QoS = \frac{QoS_{max} - QoS_{neg}}{QoS_{max} - QoS_{min}} \quad (1-2)$$

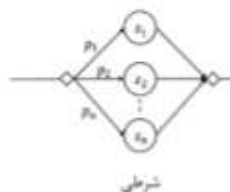
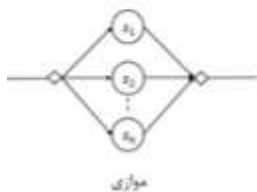
$$QoS = \frac{QoS_{pos} - QoS_{min}}{QoS_{max} - QoS_{min}} \quad (2-2)$$

۲-۵- شکل‌های مختلف ترکیب وب‌سرویس‌ها

وب‌سرویس‌ها با شکل‌های مختلفی، منطبق بر جریان کاری، که نیازمندی عملکردی مشتری را برآورده می‌سازد، با یکدیگر ترکیب می‌شوند. این شکل‌ها شامل: ترتیبی، موازی، حلقوی، شرطی و ترکیبی از این موارد در حالات پیچیده، است.

در شکل ۲-۲، شکل‌های مختلف ترکیب وب‌سرویس‌ها آورده شده است.

- در شکل ترتیبی، ابتدا وب‌سرویس S_1 سپس وب‌سرویس S_2 و همین‌طور روند ادامه پیدا می‌کند تا اینکه در انتها وب‌سرویس S_n اجرا شده و عملیات پایان می‌یابد.
- در شکل حلقوی، وب‌سرویس S_1 به اندازه k بار تکرار می‌شود تا اینکه عملیات مورد نظر به انجام برسد.
- در شکل موازی، وب‌سرویس‌های S_1, S_2, \dots, S_n به طور همزمان اجرا می‌شوند تا اینکه عملیات مورد نظر به انجام برسد.
- در شکل شرطی، برای ادامه‌ی روند، یکی از وب‌سرویس‌ها بایستی اجرا شود که احتمال ادامه‌ی روند با وب‌سرویس S_1 برابر با p_1 است و احتمال ادامه‌ی روند با S_2 برابر با p_2 است و در انتها احتمال ادامه‌ی روند با S_n برابر با p_n است.



شکل ۲-۲: شکل‌های مختلف ترکیب وب‌سرویس‌ها (ترتیبی، حلقه، موازی،

شرطی) [۱۰]

کنیم، تبدیل به یکی از مهم‌ترین مباحث در ترکیب وب‌سرویس‌ها شده است [۶].

۲-۳- ویژگی‌های کیفی وب‌سرویس

گام شناسایی وب‌سرویس‌ها یکی از مهم‌ترین و ضروری‌ترین گام‌ها پیش از انتخاب و ترکیب وب‌سرویس‌ها به شمار می‌آید، باید به نحوی از کیفیت خروجی‌های این گام که همان وب‌سرویس‌های کشف شده در این گام هستند، شناخت پیدا کرد، برای این منظور از ویژگی‌های کیفی برای سنجش کیفیت وب‌سرویس‌های شناسایی و کشف شده، استفاده می‌کنیم.

ویژگی‌های کیفی وب‌سرویس‌ها به دو دسته‌ی ویژگی‌های کیفی طراحی و ویژگی‌های کیفی پیاده‌سازی و اجرا تقسیم بندی می‌شوند. لازم به ذکر است در مراحل کشف، انتخاب و ترکیب وب‌سرویس‌ها، مبنای برتری وب‌سرویس نسبت به وب‌سرویس‌های دیگر ویژگی‌های کیفی سطح پیاده‌سازی و اجرای وب‌سرویس‌ها هستند، از جمله‌ی این ویژگی‌های کیفی می‌توان به هزینه اجرایی، زمان پاسخ، قابلیت اطمینان، دسترس‌پذیری، امنیت، تعامل‌پذیری و قابلیت استفاده اشاره کرد.

۲-۴- ویژگی‌های کیفی مثبت و منفی و مصالحه‌ی آن‌ها

ویژگی‌های کیفی وب‌سرویس‌ها به دو دسته‌ی کلی ویژگی‌های کیفی مثبت و منفی تقسیم می‌شوند، ویژگی‌های کیفی منفی به ویژگی‌هایی مانند زمان پاسخ و هزینه گفته می‌شود که هر چه میزان آن‌ها بیشتر باشد، وب‌سرویس ارزش کمتری برای انتخاب شدن دارد. در صورتی که ویژگی‌هایی مانند قابلیت استفاده، تعامل‌پذیری، دسترس‌پذیری، امنیت و قابلیت اطمینان را ویژگی‌های کیفی مثبت می‌نامند، زیرا میزان این ویژگی‌ها هرچه بیشتر باشد وب‌سرویس، کیفیت و ارزش بیشتری برای انتخاب شدن دارد. برای استاندارد کردن ویژگی‌های کیفی منفی (QoS_{neg}) از رابطه‌ی (۱-۲) و برای استاندارد کردن ویژگی‌های کیفی مثبت (QoS_{pos}) از رابطه‌ی (۲-۲) استفاده می‌کنیم، در اصل ویژگی کیفی (QoS) که از روابط (۱-۲) و (۲-۲) به دست می‌آید، مصالحه‌ی بین ویژگی‌های کیفی منفی و مثبت برقرار کرده است و در ادامه‌ی مقاله در محاسبه‌ی مجموع ویژگی‌های کیفی ترکیب وب‌سرویس‌ها از مصالحه‌ی ایجاد شده بین ویژگی‌های کیفی مثبت و منفی (ویژگی کیفی استاندارد شده (QoS)) استفاده می‌کنیم و دیگر دغدغه‌ی منفی یا مثبت بودن ویژگی‌های کیفی را نداریم [۶].

در روابط (۱-۲) و (۲-۲)، بیشینه و کمینه‌ی هر نوع ویژگی کیفی (QoS_{pos}, QoS_{neg}) از اعداد مربوط به آن نوع ویژگی کیفی انتخاب می‌شوند و مثبت یا منفی بودن ویژگی کیفی در انتخاب اعداد در نظر گرفته

۲-۶- محاسبه‌ی ویژگی کیفی وب‌سرویس مرکب

حال بایستی روشی برای محاسبه‌ی مجموع ویژگی‌های کیفی ترکیب وب‌سرویس‌ها، مطابق با هر کدام از شکل‌های ترکیبی ارائه نماییم.

توابع محاسبه‌ی ویژگی کیفی وب‌سرویس مرکب نهایی مطابق با شکل‌های مختلف ترکیبی در جدول ۱-۲ آورده شده است.

روش کیفی	ترکیبی	تصادفی	مجموعی
روش (۱)	$\sum_{i=1}^n T_i(S_i)$	$\sum_{i=1}^n T_i(S_i)$	$k \cdot T(S)$
روش (۲)	$\sum_{i=1}^n C_i(S_i)$	$\sum_{i=1}^n C_i(S_i)$	$k \cdot C(S)$
استمرار پذیری (۳)	$\prod_{i=1}^n A_i(S_i)$	$\prod_{i=1}^n p_i \cdot A_i(S_i)$	$A(S)^k$
قابلیت اطمینان (۴)	$\prod_{i=1}^n R_i(S_i)$	$\prod_{i=1}^n p_i \cdot R_i(S_i)$	$R(S)^k$

جدول ۱-۲: توابع محاسبه‌ی میزان ویژگی کیفی وب‌سرویس مرکب نهایی [۹]

۳- کارهای مرتبط

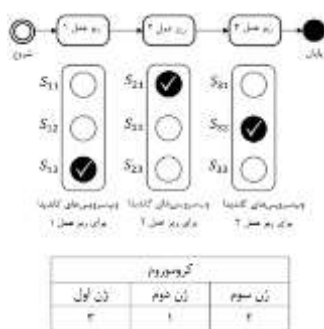
۳-۱- الگوریتم‌های فراابتکاری

الگوریتم‌های فراابتکاری روش‌هایی هستند که بر پایه‌ی الهام گرفتن از پدیده‌های طبیعی، زیستی یا اجتماعی کار خود را انجام می‌دهند. مزایایی چون مقیاس‌پذیری، تطابق‌پذیری، قابلیت اطمینان و زمان اجرای قابل قبول باعث شده تا انواع مختلفی از این الگوریتم‌ها از جمله الگوریتم ژنتیک [۲]، کلونی مورچگان [۳]، تجمع ذرات [۵]، رقابت استعماری [۱۱] و ... را تبدیل به روش‌هایی نیرومند در حل مسائل بهینه‌سازی و به طبع مسئله‌ی بهینه‌سازی انتخاب و ترکیب وب‌سرویس‌ها کند.

۳-۲- الگوریتم ژنتیک

الگوریتم ژنتیک یکی از انواع الگوریتم‌های فراابتکاری است که تصادفی و هوشمند بوده و بر مبنای یک نظریه‌ی زیست‌شناسی به نام داروین بنیان‌گذاری شده است. این الگوریتم از مفاهیم وراثت، جهش، انتخاب ناگهانی و طبیعی و ترکیب ژن‌ها در علم زیست‌شناسی الهام گرفته است. الگوریتم با ایجاد مجموعه‌ای کاملاً تصادفی از موجودیت‌ها شروع به کار می‌کند و نسل‌های بعدی نیز با انجام اعمالی تصادفی روی نسل‌های قبلی ایجاد می‌شوند. از هر نسل، بهترین موجودیت انتخاب می‌شود، اگر شرایط مورد نظر مشتری را داشته باشد، آن موجودیت به عنوان جواب مسئله معرفی می‌شود و الگوریتم پایان می‌یابد، در غیر این صورت این موجودیت به نسل بعد منتقل می‌شود و مجموعه‌ای کاملاً تصادفی از موجودیت‌ها جدید از نسل قبلی ایجاد و روند قبلی تا زمان یافتن موجودیتی که شرایط مشتری را داشته باشد ادامه می‌یابد و اگر موجودیتی مناسب پیدا نشود،

الگوریتم زمانی که به تعداد تکرارهای مشخص شده برای آن برسد، پایان می‌یابد و بهترین موجودیتی که در این تکرارها یافته است را به عنوان جواب مسئله معرفی می‌کند. برای نمایش هر موجودیت در هر نسل از پارامتری به نام کروموزوم^۵ یا ژنوم^۶ استفاده می‌کنیم. هر کروموزوم از تعدادی ژن تشکیل شده است. برای ایجاد جمعیت نسل‌های دوم به بعد، از عملگرهای ژنتیکی روی نسل‌های قبلی استفاده می‌کنیم. انتخاب تصادفی ژن‌ها و ترکیب آن‌ها با هم و ایجاد کروموزوم‌هایی جدید به عنوان فردی در جمعیت نسل جدید یکی از عملگرها است که به آن تقاطع^۷ یا بازترکیبی^۸ می‌گویند و انتخاب تصادفی ژن‌هایی از کروموزوم و جهش آن‌ها، عملگر ژنتیکی دیگری است که به آن جهش^۹ می‌گویند. این عملگرهای ژنتیکی باعث ایجاد نسل جدیدی از کروموزوم‌ها می‌شوند که با نسل قبلی تفاوت دارند. الگوریتم ژنتیک در مسئله‌ی ترکیب وب‌سرویس‌ها برای یافتن وب‌سرویس‌های مناسب برای ترکیب با یکدیگر استفاده می‌شود، کروموزوم‌هایی که الگوریتم ژنتیک می‌یابد نشان دهنده‌ی جریان کاری وب‌سرویس مرکب است و ژن‌های کروموزوم نشان‌دهنده‌ی وب‌سرویس‌های کاندیدایی هستند که انتخاب شده‌اند و بایستی با هم ترکیب شوند [۱۵، ۱۷]. فرض کنید جریان کاری حاوی m فرآیند داریم و برای هر فرآیند n وب‌سرویس کاندیدا داریم، بنابراین کروموزوم ما حاوی m ژن است و هر ژن مقداری بین ۱ تا n را می‌تواند بپذیرد. به عنوان مثال شکل ۱-۳ یک جریان کاری را نشان می‌دهد که دارای سه فرآیند است، بنابراین کروموزوم‌های این مسئله حاوی سه ژن هستند و برای هر فرآیند، سه وب‌سرویس کاندیدا وجود دارد که تنها یک وب‌سرویس برای ترکیب انتخاب می‌شود، پس هر ژن این کروموزوم عددی بین یک تا سه را می‌تواند بپذیرد. با توجه به ساختار کروموزوم در شکل ۱-۳ می‌توان نتیجه گرفت که وب‌سرویس سوم از میان وب‌سرویس‌های کاندیدا برای فرآیند اول، وب‌سرویس اول از میان وب‌سرویس‌های کاندیدا برای فرآیند دوم و وب‌سرویس دوم از میان وب‌سرویس‌های کاندیدا برای فرآیند سوم انتخاب شده‌اند.



شکل ۱-۳: ساختار کروموزوم [۱۷]

از دیگر تفاوت‌ها میان پیاده‌سازی‌های مختلف الگوریتم ژنتیک، بحث روش کدگذاری این الگوریتم است. برخی از پژوهش‌ها [۲۱،۱۹] روش کدگذاری یک کروموزوم یک بعدی را اتخاذ کرده‌اند که البته در صورت افزایش تعداد وب‌سرویس‌های کاندیدا، قابلیت اطمینان کروموزوم‌ها در این روش خیلی ضعیف می‌شود. از طرفی دیگر این روش نمی‌تواند اطلاعات معنایی را نشان دهد. بنابراین برخی از روش‌ها بر پایه الگوریتم کدگذاری ماتریس روابط بنا نهاده شده‌اند. گرچه این روش نیز در موارد زیادی باعث تولید افراد غیرمجاز می‌شود و کارایی الگوریتم را پایین می‌آورد. در [۱۶،۱۷] از کدگذاری درختی که می‌تواند روابط ترکیب‌های مختلف بین وب‌سرویس‌ها را مشخص کند صحبت شده است.

با توجه به مطالب گفته شده دیدیم که مدل استاندارد و یا تعریف شده‌ای برای پارامترهای کلیدی الگوریتم ژنتیک در حل مسئله‌ی انتخاب و ترکیب وب‌سرویس‌ها براساس ویژگی‌های کیفی وجود ندارد و هر کدام از پژوهش‌ها با اتخاذ سیاست‌ها و پارامترهای مختلف این الگوریتم را مورد استفاده قرار داده و در جهت بهبود کارایی آن تلاش کرده‌اند. این پارامترها شامل الگوهای کدگذاری مثل آرایه صحیح، رشته باینری، یک‌بعدی، ماتریسی و درختی، انواع عملگرهای انتخاب شامل انتخاب تصادفی، نخبگان، انواع عملگرهای تقاطع شامل تک نقطه‌ای، دو نقطه‌ای، چند نقطه‌ای و جامع و انواع تابع محاسبه ارزش مجموع ویژگی کیفی همراه با فاکتور جریمه یا حذفی و ... است.

۴- مسئله

۴-۱- همبستگی ویژگی‌های کیفی در مسئله‌ی انتخاب

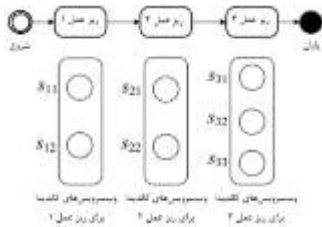
و ترکیب وب‌سرویس‌ها

همبستگی بین ویژگی‌های کیفی زمانی رخ می‌دهد که میزان ویژگی‌های کیفی وب‌سرویس‌ها تنها به خود وب‌سرویس‌های انتخابی وابسته نبوده و به وب‌سرویس‌های دیگری که در جریان کاری انتخاب و با هم ترکیب می‌شوند نیز وابسته هستند. فرض کنید دو وب‌سرویس مجاور هم که به صورت ترتیبی با هم ترکیب شده‌اند اگر از یک شرکت و در یک مکان باشند دیگر نیازی به انتقال اطلاعات بین این دو وب‌سرویس نداریم و در صورت ترکیب این دو وب‌سرویس با هم به زمان پاسخ بهتری برای ترکیب آن‌ها دست می‌یابیم، به این حالت، همبستگی در وب‌سرویس‌های مجاور گویند. برای همبستگی در وب‌سرویس‌های نامجاور می‌توان به تأمین‌کنندگان وب‌سرویس‌های متنوعی همچون میکروسافت^۱ و آمازون^{۱۱} اشاره کرد که برای تشویق مشتریان به خرید تعداد وب‌سرویس‌های متفاوت بیشتری از آن‌ها، قیمت‌های وب‌سرویس‌ها را کاهش می‌دهند، در صورتی که مشتری دو یا تعداد بیشتری وب‌سرویس از آن‌ها خریداری کند.

```
Algorithm 1 Genetic Algorithm for QoS-aware Web Service Selection and Composition
Input: set of web services, set of quality of them and relation (sequence, parallel, loop,...)
between them, best quality of service (BQoS), crossover rate, mutation rate, population size,
number evolution, maximum population generation
Output: set of best web services with the best quality of services (QoS)
1: population = new population(population_size)
2: generation_count = 1
3: repeat
4:   population = new population(population_size - 1) U population.getFittest()
5:   generation_count = generation_count + 1
6:   crossover(population, crossover_rate, number_evolution)
7:   mutation(population, mutation_rate, number_evolution)
8:   until population.getFittest().getFitness() < BQoS or
9:   generation_count < maximum_population_generation
10: return population.getFittest()
```

الگوریتم ۳-۱: الگوریتم ژنتیک

همان‌طور که اشاره شد، الگوریتم ژنتیک به جای جستجو در یک نقطه، مجموعه‌ای از نقاط را جستجو می‌کند و در نتیجه کمتر در دام بهینه محلی گرفتار می‌شود که البته همین میزان اندک گرفتار شدن در دام بهینه محلی نیز باعث پایین آمدن کارایی الگوریتم ژنتیک می‌شود. از طرفی خاصیت تصادفی بودن آن منجر شده تا هنگام رسیدن به جواب مناسب، از آن دور شده و در پیدا کردن جواب بهینه دچار مشکلات شود. از این رو برای رفع معایب ذکر شده در طی سال‌های اخیر کارهای مختلفی در زمینه‌ی بهبود الگوریتم ژنتیک اولیه صورت گرفته است. از آن موارد می‌توان به ایجاد برخی تغییرات در عملگرها و پارامترهای الگوریتم و یا ترکیب این الگوریتم با سایر روش‌های دیگر به منظور افزایش کارایی آن اشاره کرد. به عنوان نمونه در مقاله‌ی [۲۲] این الگوریتم برای حل مسئله‌ی ترکیب وب‌سرویس‌ها ارائه شده است. مؤلفان در این پژوهش اثر پارامترهای مختلف مثل نرخ جهش و تقاطع را روی توانایی بهینه‌سازی الگوریتم ژنتیک بررسی کرده و نشان داده‌اند که پارامترهای مختلف در کارایی این الگوریتم تأثیرهای متفاوتی را دارند. در مقاله [۲۳] نیز با اشاره به تأثیر انتخاب تابع محاسبه ارزش مجموع ویژگی کیفی و قوانین جهش، خاطر نشان شده است که چگونگی انتخاب این پارامترها می‌تواند در بهبود سرعت همگرایی الگوریتم ژنتیک نقش بسزایی داشته باشد. در پژوهشی دیگر [۱۳] از یک بهینه‌ساز محلی برای اصلاح مقادیر ارزش مجموع ویژگی کیفی افراد جمعیت استفاده شده که منجر به اصلاح مقادیر بهینه نهایی می‌شود. کنفرا و همکارانش نیز از این الگوریتم با یک تابع محاسبه‌ی ارزش مجموع ویژگی کیفی پویا استفاده کرده‌اند. در این روش ساختار ژنوم آرایه‌ای از اعداد صحیح است که هر مدخل آن به یک سرویس اشاره دارد. عملگر تقاطع در اینجا دو نقطه‌ای و عملگر جهش به صورت تصادفی عمل می‌کند [۱۴]. اما در کاری دیگر [۱۲] ساختار ژنوم به صورت آرایه‌ای در نظر گرفته شده که هر مدخل آن به جای اشاره به یک وب‌سرویس به یک ویژگی کیفی اشاره دارد. عملگر تقاطع در اینجا تک نقطه‌ای و عملگر جهش مانند روش قبل عمل می‌کند.



شکل ۲-۴: مثالی از انتخاب و ترکیب ترتیبی وب‌سرویس‌ها [۱]

وب‌سرویس	تأمین‌کننده وب‌سرویس	زمان پاسخ وب‌سرویس (میلی ثانیه)	مقدار زمان انتقال اطلاعات به سازمان (میلی ثانیه)	میزان هزینه اجرای وب‌سرویس (میلی ثانیه)	مقدار زمان خروج اطلاعات از سازمان (میلی ثانیه)
S_{11}	Oracle	۲۲۲	۲۲	۲۲	۱۱۲
S_{12}	Amazon	۲۱۰	۲۵	۲۱	۱۲۴
S_{21}	Microsoft	۲۲۲	۲۸	۲۲	۱۲۴
S_{22}	Netrae	۲۰۶	۱۱۱	۲۲	۱۲۱
S_{31}	Amazon	۲۲۲	۱۵۲	۲۲	۱۲۷
S_{32}	Netrae	۲۲۲	۱۲۷	۲۲	۱۱۲
S_{33}	IBM	۲۱۲	۱۲۰	۲۲	۱۲۲

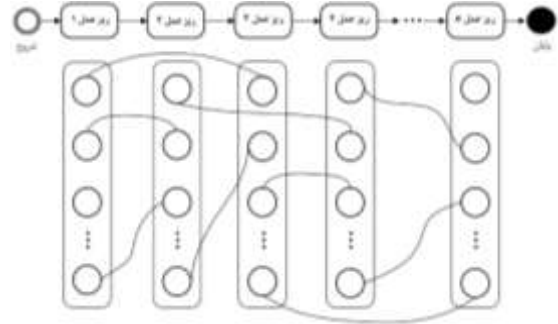
جدول ۱-۴: زمان پاسخ وب‌سرویس‌ها [۱]

۴-۱-۲- همبستگی ویژگی‌های کیفی در وب‌سرویس‌های

نامجاور

در این مثال اگر به صورت محلی برای هر فرآیند بهترین وب‌سرویس را انتخاب کنیم، وب‌سرویس‌های S_{11} ، S_{21} ، S_{31} را انتخاب خواهیم کرد و میزان ویژگی کیفی هزینه‌ی اجرایی برای وب‌سرویس مرکب برابر $50 + 160\$ = 35 + 75$ خواهد شد. حال اگر در این مثال به صورت محلی عمل نکنیم و به صورت سراسری نگاه کنیم و همبستگی بین وب‌سرویس‌ها را در نظر بگیریم و وب‌سرویس‌های S_{12} ، S_{21} ، S_{31} را انتخاب کنیم، میزان هزینه‌ی اجرایی وب‌سرویس مرکب برابر $155\$ = 60 + 35 + 60$ می‌شود، وب‌سرویس‌های انتخابی در این حالت همبسته و نامجاور هستند و یا اگر وب‌سرویس‌های S_{11} ، S_{22} ، S_{32} را انتخاب کنیم، میزان هزینه‌ی اجرایی وب‌سرویس مرکب برابر $140\$ = 50 + 40 + 50$ می‌شود. بنابراین در این مثال با اتخاذ رویکرد بهینه سراسری و در نظر گرفتن همبستگی بین وب‌سرویس‌ها 20 دلار یا $12/5$ درصد در هزینه‌ی اجرایی وب‌سرویس‌ها بهبود ایجاد کرده‌ایم. دلیل این بهبود، همبستگی وب‌سرویس‌های S_{22} ، S_{32} است، زیرا این دو وب‌سرویس از یک سازمان هستند و سازمان تأمین‌کننده وب‌سرویس‌ها به منظور تشویق مشتریان به استفاده از وب‌سرویس‌های بیشتری، در صورت استفاده‌ی بیش از دو وب‌سرویس، هزینه‌ی اجرایی را از وب‌سرویس دوم به بعد کاهش می‌دهند تا مشتری را برای استفاده بیشتر از وب‌سرویس‌های سازمان خود ترغیب کنند، در این

مقاله‌ای که تاکنون در این زمینه ارائه شده‌اند، همگی همبستگی را برای ترکیب وب‌سرویس‌ها در حالت ترتیبی در نظر گرفته‌اند و راه‌حلی برای آن ارائه کرده‌اند. راه‌حل‌های ارائه شده تاکنون قادر به حل مسئله با وجود همبستگی بین تعداد زیادی از وب‌سرویس‌ها و شرایط پیچیده نیستند.



شکل ۱-۴: مثالی از شرایط پیچیده در همبستگی بین تعداد زیادی از وب‌سرویس‌ها

۴-۱-۱- همبستگی ویژگی‌های کیفی در وب‌سرویس‌های

مجاور

به کمک این مثال مفاهیم بهینه‌ی سراسری (ترکیب بهینه) و محلی (انتخاب بهینه) و چگونگی محاسبه‌ی میزان ویژگی کیفی وب‌سرویس مرکب شرح داده می‌شود. اگر به صورت محلی برای هر فرآیند، بهترین وب‌سرویس را از میان وب‌سرویس‌های کاندیدا انتخاب کنیم، وب‌سرویس‌های S_{11} ، S_{21} ، S_{31} را انتخاب خواهیم کرد و میزان ویژگی کیفی زمان پاسخ برای وب‌سرویس مرکب $881ms = 292 + 274 + 315$ خواهد شد. حال اگر در این مثال به صورت محلی عمل نکنیم و به صورت سراسری نگاه کنیم و وب‌سرویس‌های S_{11} ، S_{22} ، S_{32} را انتخاب کنیم، میزان زمان پاسخ وب‌سرویس مرکب $654ms = 292 + 306 + 344 - 141 - 147$ می‌شود. با اتخاذ رویکرد بهینه سراسری در این مثال 227 میلی‌ثانیه بهبود ایجاد کردیم، دلیل این بهبود، همبستگی وب‌سرویس‌های S_{22} ، S_{32} است، زیرا این دو وب‌سرویس از یک سازمان تأمین‌کننده و در یک مکان هستند و در واقع در فعالیت دوم زمانی برای خروج اطلاعات از سازمان تأمین‌کننده و در فعالیت سوم، زمانی برای ارسال اطلاعات به سازمان تأمین‌کننده نیاز نداریم، زیرا اطلاعات در سازمان وجود دارند بنابراین می‌توانیم به کمک بهینه‌ی سراسری و با در نظر گرفتن همبستگی بین وب‌سرویس‌ها، 227 میلی‌ثانیه یا $25/7$ درصد نسبت به روش بهینه‌ی محلی، بهبود ایجاد کنیم. برای این نوع از ویژگی‌های کیفی تنها بایستی وب‌سرویس‌هایی را انتخاب کنیم که از یک سازمان تأمین‌کننده و در یک مکان قرار دارند و مجاور یکدیگر نیز هستند تا بتوانیم در زمان‌های خروج اطلاعات از سازمان و ورود اطلاعات به سازمان صرفه‌جویی کنیم.

(وابستگی‌ها و ناسازگاری‌ها) پشتیبانی می‌نمایند) از کارایی خوبی جهت مواجهه با محدودیت‌ها برخوردار شود. در ادامه یک روش از بین روش‌های مختلف به کارگیری محدودیت‌ها، برای مواجهه با محدودیت‌ها همراه با وابستگی‌ها و ناسازگاری بین وب‌سرویس‌ها مورد استفاده قرار گرفته است، شرح داده خواهد شد.

الگوریتم ژنتیک مبتنی بر جریمه از یک تابع جریمه برای مواجهه با محدودیت‌ها در وابستگی و ناسازگاری بین وب‌سرویس‌ها درگیر در مسئله استفاده می‌کند.

۱-۲-۴- الگوریتم ژنتیک مبتنی بر جریمه

شاید الگوریتم جریمه محبوب‌ترین الگوریتم برای مواجهه و پاسخ‌گویی به محدودیت‌ها باشد. الگوریتم ژنتیک مبتنی بر جریمه از یک تابع جریمه جهت پشتیبانی از محدودیت‌های وابستگی و ناسازگاری بین وب‌سرویس‌ها استفاده می‌کند. ایده‌ی اصلی این الگوریتم و تابع آن، دادن جریمه به راه‌حل‌های ناشدنی است. تابع جریمه در صورت مشاهده‌ی انحراف و نقض محدودیت‌های وابستگی و ناسازگاری بین وب‌سرویس‌ها در راه‌حلی که می‌یابد، ارزش راه‌حل یافت شده را کاهش می‌دهد تا این راه‌حل ناشدنی، شانس کمتری برای زنده ماندن نسبت به راه‌حل‌های شدنی دیگری که محدودیت‌ها را نقض نکرده‌اند، داشته باشد. در الگوریتم ژنتیک مبتنی بر جریمه برای کروموزوم‌هایی که ژن‌های ناشدنی دارند و محدودیت‌های وابستگی و ناسازگاری بین وب‌سرویس‌ها را نقض کرده‌اند، تابع محاسبه‌ی برازندگی، از استراتژی جریمه استفاده شده است [۱۸].

باید توجه داشت که عملگرهای جهش و تقاطع که روی والد‌ها اجرا می‌شود، ممکن است راه‌حل‌های (کروموزوم‌های) ناشدنی تولید نمایند. به عنوان مثال فرض کنید که اگر در خانه‌ی اول کروموزوم وب‌سرویس شماره‌ی سه انتخاب شود و قرار گیرد و در خانه‌ی پنجم کروموزوم وب‌سرویس شماره‌ی پنج انتخاب شود و قرار گیرد، به دلیل اینکه این دو وب‌سرویس با هم ناسازگار هستند، راه‌حلی که از ترکیب این وب‌سرویس‌ها به دست می‌آید، در کل به راه‌حلی ناشدنی تبدیل می‌شود. همچنین وب‌سرویس شماره‌ی سه در خانه‌ی شماره‌ی یک با وب‌سرویس شماره‌ی سه در خانه‌ی شماره‌ی شش با یکدیگر ناسازگار هستند.

$WS_{1,3}$ has Conflict with $WS_{5,5}$

$WS_{1,3}$ has Conflict with $WS_{6,3}$

در شکل ۳-۴ والد‌ها محدودیت‌های ذکر شده در بالا را نقض نکرده‌اند. با این حال پس از عمل تقاطع، فرزند اول محدودیت ناسازگاری را نقض می‌نماید و همچنین عملگر جهش نیز منجر به نقض محدودیت می‌شود.

نوع ویژگی کیفی امکان انتخاب وب‌سرویس‌های نامجاور همبسته وجود دارد.

وب‌سرویس	سازمان تأمین‌کننده وب‌سرویس	هزینه‌ی اجرای وب‌سرویس (دلار)	مقایسه‌ی وابستگی با وب‌سرویس دیگر	هزینه‌ی اجرای وب‌سرویس در صورت انتخاب وب‌سرویس همبسته با این وب‌سرویس (دلار)
S_{11}	Netrae	50	تعریف نشده	تعریف نشده
S_{12}	Microsoft	60	تعریف نشده	تعریف نشده
S_{21}	IBM	75	تعریف نشده	تعریف نشده
S_{22}	Amazon	90	تعریف نشده	تعریف نشده
S_{31}	Microsoft	90	S_{12}	60
S_{32}	Amazon	90	S_{22}	90
S_{33}	Oracle	75	تعریف نشده	تعریف نشده

-----: هزینه برابری از یک همبسته
-----: هزینه محلی انتخاب همبسته

جدول ۲-۴: هزینه‌ی اجرای وب‌سرویس‌ها [۱]

۲-۴- مسئله‌ی انتخاب و ترکیب وب‌سرویس‌ها

معمولاً در انتخاب وب‌سرویس‌ها، باید محدودیت‌های وابستگی و ناسازگاری بین وب‌سرویس‌ها را به منظور حفظ یکپارچگی و تضمین درستی و صحت اجرای وب‌سرویس مرکب در نظر گرفت. در واقع انتخاب وب‌سرویس، یک عملیات مستقل نیست، بلکه باید وابستگی و ناسازگاری بین وب‌سرویس‌ها را در نظر بگیرد.

یک وب‌سرویس مرکب ناشدنی، نتیجه‌ی انتخاب دو وب‌سرویس ناسازگار برای یک وب‌سرویس مرکب هستند. انتخاب دقیق وب‌سرویس‌ها، لازمه‌ی اجتناب از رخ دادن یک وب‌سرویس مرکب ناشدنی است که به این موضوع اصطلاحاً صحت و درستی گفته می‌شود.

مسئله‌ای که در این قسمت بررسی می‌شود، یک مسئله‌ی بهینه‌سازی محدودیت‌دار است. چگونگی دست‌یابی به ارزش مجموع ویژگی کیفی مناسب تحت عنوان موضوع بهینه‌سازی تعریف می‌شود و چگونگی حصول اطمینان از برآورده شدن محدودیت‌ها یعنی وابستگی‌ها و ناسازگاری‌ها بین وب‌سرویس‌ها تحت عنوان موضوع درستی و صحت معرفی می‌شود که این دو موضوع دو چالش عمده در ترکیب وب‌سرویس‌ها هستند و الگوریتم‌های ژنتیکی برای مقابله با این چالش‌ها مورد استفاده قرار می‌گیرند.

با توجه به طبیعت و ماهیت الگوریتم ژنتیک، این الگوریتم به تنهایی می‌تواند روی مباحثی که دارای محدودیت نیستند بهینه‌سازی انجام دهد. خوشبختانه این الگوریتم می‌تواند با همکاری و ادغام با برخی از روش‌ها (همچون تابع جریمه، ترمیم و برخی از روش‌های ترکیبی) که از محدودیت‌ها

مسئله‌ی مطرح شده است، بنابراین هنگامی که $V(x)$ برابر صفر است، دلالت بر این نکته دارد که کروموزوم x ، راه‌حلی شدنی است و هیچ موردی از محدودیت‌ها را نقض نکرده است.

با توجه به رابطه‌ی (۱-۴) می‌توان اشاره کرد که اگر کروموزوم x راه‌حلی شدنی باشد، میزان جریمه‌ی آن برابر صفر است. در غیر این صورت مقدار جریمه‌ی داده شده به کروموزوم x به عنوان یک راه‌حل ناشدنی از طریق معادله‌ی $-0.5 - \frac{V(x)}{V_{max}}$ محاسبه می‌شود و این معادله تضمین می‌کند که یک راه‌حل ناشدنی هر چقدر محدودیت‌های بیشتری را نقض کند، جریمه‌ی بالاتری نیز دریافت خواهد نمود.

با توجه به روابط (۱-۴) و (۲-۴) مقدار تابع برازندگی هر کروموزوم شدنی x را می‌توان از طریق معادله‌ی $0.5 + (0.5 * F_{obj}(x))$ محاسبه نمود، که مقدار آن همواره $[0.5, 1]$ است. مقدار تابع برازندگی هر کروموزوم ناشدنی x را می‌توان از طریق فرمول $(0.5 * F_{obj}(x)) - \frac{V(x)}{V_{max}}$ محاسبه نمود، که مقدار آن همواره $[0, 0.5]$ است. بنابراین به کمک این روابط می‌توان تضمین کرد که یک کروموزوم ناشدنی نسبت به هر کروموزوم شدنی، ارزش کمتری از تابع برازندگی کسب خواهد کرد [۱۸].

۵- روش پیشنهادی

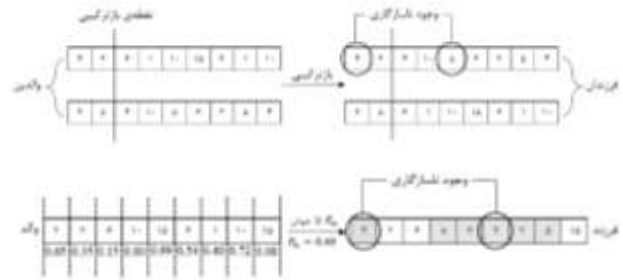
۵-۱- روند عملیاتی الگوریتم ژنتیک برای یافتن جواب

مسئله‌ی انتخاب و ترکیب و بسرویس‌ها

الگوریتم ژنتیک برای شروع کار خود و یافتن جواب مسئله‌ی انتخاب و ترکیب و بسرویس‌ها، جمعیتی از کروموزوم‌ها ایجاد می‌کند، شماره‌ی هر ژن کروموزوم اشاره به شماره‌ی و بسرویس انتزاعی در جریان کاری دارد و مقدار عددی‌ایی که در ژن‌های کروموزوم‌ها قرار می‌گیرد، شماره‌ی و بسرویس کاندیدایی است که برای آن فعالیت انتخاب شده است.

به عنوان مثال، ساختار کروموزوم و ژن‌های آن، برای جریان کاری مورد نظر مشتری که در بالا ذکر شد، در شکل ۱-۵ نمایش داده شده است.

برای پشتیبانی و مواجهه با راه‌حل‌های ناشدنی، یک تابع جریمه در تابع برازندگی این الگوریتم تعریف می‌کنیم.



شکل ۳-۴: ظهور ناسازگاری در فرزندان بعد از عملگرهای تقاطع و جهش

همان‌گونه که در بالا مطرح گردید، برخی از کروموزوم‌های تولیدی توسط عملگرهای تقاطع و جهش ممکن است به راه‌حلی ناشدنی تبدیل گردند، بنابراین در این الگوریتم باید این موضوع بررسی و در نظر گرفته شود که ممکن است کروموزوم‌های ناشدنی، شامل برخی از ژن‌ها که برای ایجاد یک راه‌حل بهینه ضروری هستند، باشند. اگر افراد ناشدنی مستثنی شوند، ممکن است که الگوریتم ژنتیک به راه‌حل بهینه، دست پیدا نکند. در نتیجه استراتژی اتخاذ شده توسط الگوریتم ژنتیک به افراد ناشدنی اجازه می‌دهد در جمعیت قرار بگیرند، اما جریمه‌هایی را به تابع برازندگی اعمال می‌نماید. در ادامه به دو رهنمود اشاره می‌کنیم که موقع تعریف تابع برازندگی استفاده می‌شوند.

باید تضمین شود که راه‌حل (کروموزوم) ناشدنی دارای کمترین ارزش برازندگی نسبت به راه‌حل (کروموزوم) شدنی است.

کروموزومی که از محدودیت‌ها تخلف زیادی نموده است، باید به شدت نسبت به کروموزومی که تخلف کمتری داشته است، جریمه شود.

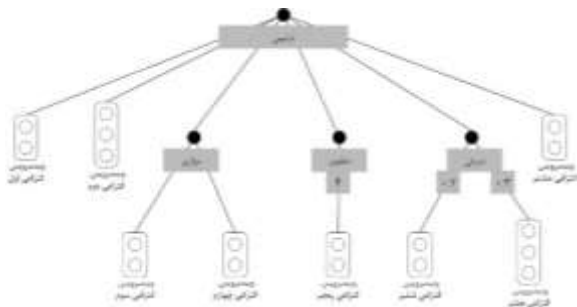
روابط (۱-۴) و (۲-۴) به ترتیب تابع برازندگی و جریمه را بیان می‌نمایند.

$$Fitness(x) = 0.5 + (0.5 * F_{obj}(x)) + P(x) \quad (1-4)$$

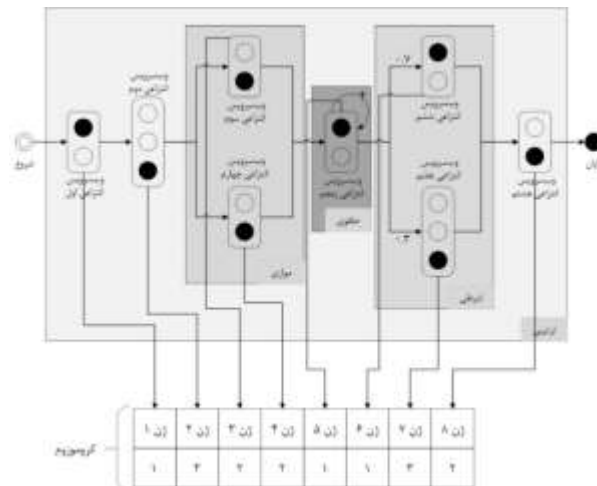
$$P(x) = \begin{cases} 0, & \text{if } V(x) = 0; \\ -0.5 - \frac{V(x)}{V_{max}}, & \text{otherwise} \end{cases} \quad (2-4)$$

در معادله‌ی اول $F_{obj}(x)$ تابع محاسبه‌ی ارزش ویژگی‌های کیفی (تابع برازندگی) برای کروموزوم x است که قبلاً معرفی شد و $P(x)$ مقدار جریمه‌ایی است که برای کروموزوم x محاسبه و نسبت داده می‌شود. در معادله‌ی دوم $V(x)$ مخفف تعداد کل موارد نقض محدودیت توسط کروموزوم x است و V_{max} حداکثر تعداد موارد موجود نقض محدودیت در

به عنوان مثال در ادامه، با توجه به جریان کاری و روابط بین وب‌سرویس‌ها، که مشتری شکل ۵-۱ به مسئله ورودی داده بود، در شکل ۵-۲ درختی که در بردارنده‌ی این اطلاعات است را ترسیم می‌کنیم.



شکل ۵-۲: درخت در بردارنده‌ی جریان کاری و روابط بین وب‌سرویس‌های مورد نظر مشتری



شکل ۵-۱: ساختار کروموزوم و ژن‌های آن برای جریان کاری مورد نظر مشتری

در ادامه، به کمک جدول ۲-۱ و استفاده از الگوریتم جستجوی عمق اول و پیمایش پسوندی آن روی درخت شکل ۵-۲، روابط محاسبه‌ی برازندگی وب‌سرویس مرکب براسا ویژگی‌های کیفی مختلف را بدست آورده و در جدول ۵-۱ نمایش می‌دهیم.

ردیف کیفی	رابطه‌ی محاسبه‌ی برازندگی وب‌سرویس مرکب برای جریان کاری مورد نظر مشتری
زمان (Q_{Time})	$Q_{Time} = \{Q_{Time}(CF_{1,t})\} + \{Q_{Time}(CF_{2,t})\} + \{Max\{Q_{Time}(CF_{3,t}), Q_{Time}(CF_{3,t})\}\} + \{8 \cdot Q_{Time}(CF_{3,t})\} + \{0.7 \cdot Q_{Time}(CF_{3,t})\} + \{0.2 \cdot Q_{Time}(CF_{3,t})\} + \{Q_{Time}(CF_{3,t})\}$
هزینه (Q_{Cost})	$Q_{Cost} = \{Q_{Cost}(CF_{1,t})\} + \{Q_{Cost}(CF_{2,t})\} + \{Q_{Cost}(CF_{3,t})\} + \{Q_{Cost}(CF_{3,t})\} + \{8 \cdot Q_{Cost}(CF_{3,t})\} + \{0.7 \cdot Q_{Cost}(CF_{3,t})\} + \{0.2 \cdot Q_{Cost}(CF_{3,t})\} + \{Q_{Cost}(CF_{3,t})\}$
استقرارپذیری ($Q_{Availability}$)	$Q_{Availability} = \{Q_{Availability}(CF_{1,t})\} + \{Q_{Availability}(CF_{2,t})\} + \{Q_{Availability}(CF_{2,t})\} + \{Q_{Availability}(CF_{3,t})\} + \{Q_{Availability}(CF_{3,t})\} + \{0.7 \cdot Q_{Availability}(CF_{3,t})\} + \{0.2 \cdot Q_{Availability}(CF_{3,t})\} + \{Q_{Availability}(CF_{3,t})\}$
قابلیت اطمینان ($Q_{Reliability}$)	$Q_{Reliability} = \{Q_{Reliability}(CF_{1,t})\} + \{Q_{Reliability}(CF_{2,t})\} + \{Q_{Reliability}(CF_{3,t})\} + \{Q_{Reliability}(CF_{3,t})\} + \{0.7 \cdot Q_{Reliability}(CF_{3,t})\} + \{0.2 \cdot Q_{Reliability}(CF_{3,t})\} + \{Q_{Reliability}(CF_{3,t})\}$
نمای ($Final$)	$Final = Q_{Time} + Q_{Cost} + Q_{Availability} + Q_{Reliability} + \{0.7 \cdot Q_{Time}\} + \{0.2 \cdot Q_{Cost}\} + \{0.7 \cdot Q_{Availability}\} + \{0.2 \cdot Q_{Reliability}\}$

جدول ۱-۵: رابطه‌ی محاسبه‌ی برازندگی وب‌سرویس مرکب برای جریان کاری مورد نظر مشتری

۵-۲- الگوریتم بررسی همبستگی بین وب‌سرویس‌های مجاور برای ویژگی کیفی زمان پاسخ

در بررسی همبستگی برای ویژگی کیفی زمان پاسخ، بایستی وب‌سرویس‌های مجاور بررسی کنیم. ویژگی کیفی زمان پاسخ هر وب‌سرویس شامل مدت زمان انتقال اطلاعات، میانگین زمان اجرای عملیات و مدت زمان خروج اطلاعات است، بنابراین برای هر وب‌سرویس بایستی وجود همبستگی بین وب‌سرویس مورد نظر را با وب‌سرویس‌های مجاور قبل و بعد از وب‌سرویس را بررسی کنیم. اگر همبستگی بین

همان‌طور که قبلاً اشاره شد، الگوریتم ژنتیک مستقل از مسئله کار خود را انجام می‌دهد بنابراین در این مسئله الگوریتم ژنتیک مستقل از روابط بین وب‌سرویس‌ها و جریان کاری‌ای که طبق آن وب‌سرویس‌ها با هم ترکیب شده‌اند، کار خود را انجام می‌دهد. الگوریتم ژنتیک در ابتدا جمعیتی از کروموزوم‌ها با مقادیری تصادفی برای ژن‌های آن‌ها تولید می‌کند. برای ادامه‌ی کار الگوریتم، ابتدا بایستی برازندگی این کروموزوم‌های تولیدی را محاسبه کنیم تا در صورت دارا بودن برازندگی مورد نظر مشتری الگوریتم پایان یابد و مقادیر ژن‌های کروموزوم برتر، وب‌سرویس‌های مناسب برای انتخاب در جریان کاری را برای ما مشخص کند. در غیر این صورت، الگوریتم ژنتیک به کمک عملگرهای جهش و تقاطع و روندی که قبلاً توضیح داده شد، جمعیت جدیدی تولید می‌کند و ادامه می‌یابد تا با یکی از شرایط خاتمه الگوریتم ژنتیک مواجه شود. برای محاسبه‌ی برازندگی کروموزوم‌ها بایستی به روابط بین وب‌سرویس‌ها و جریان کاری‌ای که طبق آن وب‌سرویس‌ها با هم ترکیب شده‌اند و کنار هم قرار گرفته‌اند، توجه کرد، برای این منظور درختی که در بردارنده‌ی شکل جریان کاری و روابط بین وب‌سرویس‌ها است را تشکیل می‌دهیم و همواره برای محاسبه‌ی برازندگی کروموزوم و وب‌سرویس مرکب به این درخت مراجعه می‌کنیم و به کمک الگوریتم جستجوی عمق اول^{۱۲} و پیمایش پسوندی^{۱۳} آن، رابطه‌ی محاسبه‌ی برازندگی را برای وب‌سرویس مرکب می‌یابیم و با توجه به مقادیر ژن‌های کروموزوم‌ها وب‌سرویس‌های انتخابی را یافته و میزان ویژگی کیفی آن‌ها را در رابطه‌ی یافت شده از پیمایش درخت قرار می‌دهیم و برازندگی وب‌سرویس مرکب را محاسبه و با برازندگی که مورد نظر مشتری است مقایسه کرده و برای ادامه کار الگوریتم تصمیم می‌گیریم.



```

Procedure 1 previousAdjacentWebServices(NWS, tree)
1: if NWS.getType() == "sequential" then
2:   if (NWS.leftChild/NWS.getCurrentChild() == null and NWS.getFather() == null)
3:     or (NWS.leftChild/NWS.getCurrentChild() != null) then
4:     return tree.getChildren(NWS.leftChild/NWS.getCurrentChild(), 'rightistChildren')
5:   end if
6: end if
7: father_node = NWS.getFather()
8: father_node.setCurrentChild(NWS)
9: return previousAdjacentWebServices(father_node)

```

الگوریتم روند بررسی وجود همبستگی در هر یک از وب‌سرویس‌ها با وب‌سرویس‌های مجاور بعد از خود در الگوریتم ۵-۲ آورده شده است. در این الگوریتم، درختی که دربردارنده‌ی اطلاعات تمام وب‌سرویس‌های انتزاعی و نحوه‌ی ترکیب آن‌ها است را به این الگوریتم ورودی می‌دهیم.

Algorithm 3 Checking Correlation between Target Web Service and Subsequent Adjacent Web Services
Input: tree of workflow (tree), node of abstract target web service in tree of workflow (NWS), chromosome that shows which candidate web services are selected from abstract web services (chromosome)
Output: percentage of correlation for candidate web service is selected from NWS
1: abstract target web service = NWS
2: subsequent adjacent abstract web services = subsequentAdjacentWebServices(NWS, tree)
3: correlation percentage = DFSPostorderForComputeCorrelation(tree, chromosome, abstract target web service, subsequent adjacent abstract web services)
4: return correlation percentage

الگوریتم ۵-۳: الگوریتم بررسی وجود همبستگی بین وب‌سرویس هدف با وب‌سرویس‌های مجاور بعدی (برای بررسی امکان صرف نظر از مدت زمان خروج اطلاعات از وب‌سرویس هدف، از این الگوریتم می‌توان استفاده کرد).

به کمک تابعی که در پایین نوشته شده است به صورت بازگشتی درختی که به آن ورودی داده شده است را پیمایش کرده و نودی از درخت که در بردارنده‌ی تمام وب‌سرویس‌های انتزاعی که به صورت مجاور دقیقاً بعد از وب‌سرویس مورد بررسی برای همبستگی قرار دارند را یافته و به الگوریتم ۲-۵ برای ادامه‌ی روند محاسبه‌ی ویژگی کیفی نهایی وب‌سرویس مورد بررسی انتقال داده می‌شود.

```

Procedure 2 subsequentAdjacentWebServices(NWS, tree)
1: if NWS.getType() == "sequential" then
2:   if (NWS.rightChild/NWS.getCurrentChild() == null and NWS.getFather() == null)
3:     or (NWS.rightChild/NWS.getCurrentChild() != null) then
4:     return tree.getChildren(NWS.rightChild/NWS.getCurrentChild(), 'leftistChildren')
5:   end if
6: end if
7: father_node = NWS.getFather()
8: father_node.setCurrentChild(NWS)
9: return subsequentAdjacentWebServices(father_node)

```

۴-۵- الگوریتم بررسی همبستگی بین وب‌سرویس‌های نامجاور برای ویژگی کیفی هزینه

در بررسی همبستگی وب‌سرویس مورد نظر برای ویژگی کیفی هزینه بایستی تمام وب‌سرویس‌های قبل از آن را بررسی کنیم و دلیل اینکه می‌توانیم به صورت نامجاور عمل کنیم این است که در همبستگی بین وب‌سرویس‌ها برای ویژگی کیفی هزینه، هدف سازمان‌های تأمین‌کننده‌ی وب‌سرویس، تشویق مشتریان به انتخاب وب‌سرویس‌های بیشتری از سازمان آن‌ها در جریان کاری مورد نظر مشتری است و در این مورد هم نیازی به بررسی همبستگی وب‌سرویس‌ها به صورت مجاور نیست و چندین وب‌سرویس از یک سازمان می‌توانند از مکان‌های مختلف در جریان کاری

وب‌سرویس مورد بررسی با وب‌سرویس‌های مجاور قبل از آن یافت شود می‌توانیم از مدت زمان انتقال اطلاعات به وب‌سرویس هدف صرف نظر کنیم و همچنین اگر همبستگی بین وب‌سرویس مورد بررسی با وب‌سرویس‌های مجاور بعد از آن، یافت شود، می‌توانیم از مدت زمان خروج اطلاعات از وب‌سرویس هدف صرف نظر کنیم بنابراین در صورت یافتن همبستگی می‌توانیم در زمان پاسخ وب‌سرویس به میزان مدت زمان انتقال اطلاعات یا خروج اطلاعات یا هر دو بهبود ایجاد کنیم.

برای محاسبه‌ی برازندگی کروموزوم یا وب‌سرویس مرکب مورد بررسی، به کمک این الگوریتم برای هر یک از وب‌سرویس‌های انتخاب شده در ژن‌های کروموزوم امکان وجود همبستگی را بررسی می‌کنیم و در صورت یافتن همبستگی در هر یک از وب‌سرویس‌ها در محاسبه‌ی برازندگی از زمان پاسخ جدید و بهبود یافته بعد از اعمال همبستگی استفاده می‌کنیم و زمان پاسخ وب‌سرویس مرکب نیز در مجموع بهبود خواهد یافت.

برای ویژگی‌های کیفی دیگری همچون دسترس‌پذیری، تعامل‌پذیری و ... که بایستی وب‌سرویس‌های مجاور را بررسی کنیم، مشابه زمان پاسخ عمل کرده و از الگوریتم‌هایی که در ادامه آورده شده، استفاده می‌کنیم.

الگوریتم روند بررسی وجود همبستگی در هر یک از وب‌سرویس‌ها با وب‌سرویس‌های مجاور قبل از خود در الگوریتم ۵-۱ آورده شده است. در این الگوریتم، درختی که دربردارنده‌ی اطلاعات تمام وب‌سرویس‌های انتزاعی و نحوه‌ی ترکیب آن‌ها است را به این الگوریتم ورودی می‌دهیم.

Algorithm 2 Checking Correlation between Target Web Service and Previous Adjacent Web Services
Input: tree of workflow (tree), node of abstract target web service in tree of workflow (NWS), chromosome that shows which candidate web services are selected from abstract web services (chromosome)
Output: percentage of correlation for candidate web service is selected from NWS
1: abstract target web service = NWS
2: previous adjacent abstract web services = previousAdjacentWebServices(NWS, tree)
3: correlation percentage = DFSPostorderForComputeCorrelation(tree, chromosome, abstract target web service, previous adjacent abstract web services)
4: return correlation percentage

الگوریتم ۵-۱: الگوریتم بررسی وجود همبستگی بین وب‌سرویس هدف با وب‌سرویس‌های مجاور قبلی (برای بررسی امکان صرف نظر کردن از مدت زمان انتقال اطلاعات به وب‌سرویس هدف، از این الگوریتم می‌توان استفاده کرد).

به کمک تابعی که در پایین نوشته شده است به صورت بازگشتی درختی که به آن ورودی داده شده است را پیمایش کرده و نودی از درخت که در بردارنده‌ی تمام وب‌سرویس‌های انتزاعی که به صورت مجاور دقیقاً قبل از وب‌سرویس مورد بررسی برای همبستگی قرار دارند را یافته و به الگوریتم ۱-۵ برای ادامه‌ی روند محاسبه‌ی ویژگی کیفی نهایی وب‌سرویس مورد بررسی انتقال داده می‌شود.

۴-۵- الگوریتم بررسی وابستگی و ناسازگاری بین وب‌سرویس‌ها

همان‌گونه که در الگوریتم ژنتیک مبتنی بر جریمه اشاره شد، برخی از کروموزوم‌های تولیدی محدودیت‌های وابستگی و ناسازگاری بین وب‌سرویس‌ها را نقض می‌کنند و به راه‌حل‌های ناشدنی تبدیل می‌گردند، بنابراین بعد از بررسی همبستگی بین وب‌سرویس‌ها، بایستی هر یک از کروموزوم‌ها را از نظر نقض محدودیت‌های وابستگی یا ناسازگاری نیز بررسی کنیم و در صورت مشاهده نقض محدودیت‌ها ارزش راه‌حل یافته شده را به کمک الگوریتم ژنتیک مبتنی بر جریمه که قبلاً معرفی شد، کاهش دهیم تا این راه‌حل ناشدنی حتی در صورتی که به کمک همبستگی بین وب‌سرویس‌ها برازندگی مناسب‌تری پیدا کرده باشد، شانس کمتری برای زنده ماندن نسبت به راه‌حل‌های شدنی دیگری که محدودیت‌ها را نقض نکرده باشند، داشته باشد، بنابراین برای هر یک از کروموزوم‌ها بعد از بررسی همبستگی و محاسبه‌ی برازندگی، بررسی می‌کنیم که آیا ژن‌های ناشدنی دارند و محدودیت‌های وابستگی و ناسازگاری را نقض کرده‌اند که در این صورت برازندگی یافته شده شامل یک استراتژی جریمه خواهد شد (برای محاسبه‌ی برازندگی نهایی هر یک از کروموزوم‌ها با در نظر گرفتن جریمه برای راه‌حل‌های ناشدنی از روابط (۱-۴) و (۲-۴) استفاده می‌کنیم).

در انتها بعد از بررسی همبستگی، وابستگی و ناسازگاری، کروموزومی که دارای بهترین برازندگی است را به عنوان راه‌حل برای جریان کاری مورد نظر مشتری اعلام می‌کنیم.

۶- ارزیابی

در این بخش الگوریتم ژنتیک پیشنهادی با الگوریتم ژنتیک پایه براساس تأثیر افزایش همبستگی بین وب‌سرویس‌ها بر میزان برازندگی وب‌سرویس مرکب یافت شده، تأثیر افزایش وابستگی و ناسازگاری بین وب‌سرویس‌ها بر مدت زمان رسیدن به میزان برازندگی مورد نظر برای وب‌سرویس مرکب و تأثیر افزایش تعداد وب‌سرویس‌های انتزاعی و کاندیدا بر مدت زمان رسیدن به میزان برازندگی مورد نظر برای وب‌سرویس مرکب، بررسی می‌شوند.

الگوریتم‌های ژنتیک پایه و پیشنهادی با زبان PHP پیاده‌سازی شده‌اند و بر روی کامپیوتری شخصی با پردازنده‌ی 2.7 GHz Intel Core i5 و 8 GB Ram اجرا شده‌اند. نتایج مقایسه‌ی این دو الگوریتم در این بخش آورده می‌شود.

انتخاب شوند و سازمان تأمین‌کننده نیز به هدف خود یعنی فروش بیشتر وب‌سرویس‌ها رسیده است بنابراین در این مورد برای بررسی همبستگی هر وب‌سرویس، تمام وب‌سرویس‌های انتخابی قبل از آن را بررسی و مشاهده می‌کنیم که آیا از سازمان وب‌سرویس مورد بررسی، وب‌سرویس دیگری قبل از آن در جریان کاری از همان سازمان انتخاب شده است که اگر انتخاب شده باشد هزینه‌ی اجرایی جدید و بهبود یافته‌ای که خود سازمان اعلام کرده است را برای آن وب‌سرویس در نظر گرفته و از این به بعد آن را در محاسبات مورد استفاده قرار می‌دهیم.

برای محاسبه‌ی برازندگی کروموزوم یا وب‌سرویس مرکب مورد بررسی، به کمک این الگوریتم برای هر یک از وب‌سرویس‌های انتخاب شده در ژن‌های کروموزوم امکان وجود همبستگی را بررسی می‌کنیم و در صورت یافتن همبستگی در هر یک از وب‌سرویس‌ها در محاسبه‌ی برازندگی از هزینه‌ی اجرایی جدید و بهبود یافته بعد از اعمال همبستگی استفاده می‌کنیم و هزینه‌ی اجرای وب‌سرویس مرکب نیز در مجموع بهبود خواهد یافت.

برای ویژگی‌های کیفی دیگری که برای بررسی همبستگی بایستی وب‌سرویس‌های نامجاور را بررسی کنیم، مشابه هزینه‌ی اجرایی عمل می‌کنیم و از الگوریتمی که در ادامه آورده شده، استفاده می‌کنیم.

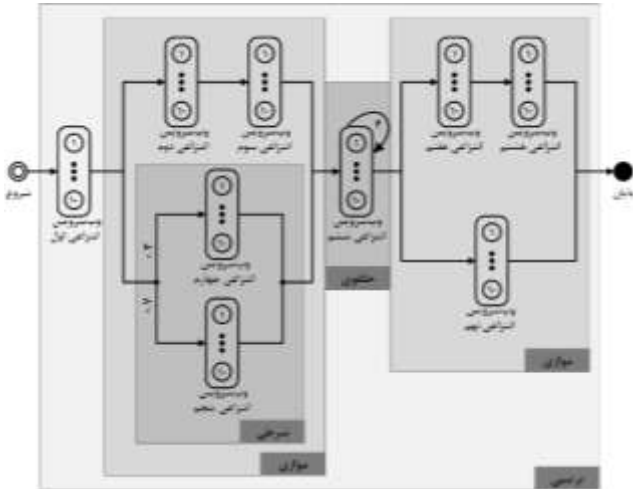
الگوریتم روند بررسی وجود همبستگی در هر یک از وب‌سرویس‌ها با وب‌سرویس‌های نامجاور قبل از خود در الگوریتم ۳-۵ آورده شده است. به این الگوریتم کروموزومی که نشان‌دهنده‌ی وب‌سرویس‌های انتخابی است ورودی داده می‌شود تا برای بررسی همبستگی وب‌سرویس هدف، بتوانیم با مراجعه به این کروموزوم، تمام وب‌سرویس‌های دیگری که انتخاب شده‌اند را یافته و مورد بررسی قرار دهیم و ویژگی کیفی نهایی وب‌سرویس هدف را به راحتی در صورت وجود همبستگی محاسبه کنیم.

Algorithm 4 Checking Correlation between Target Web Service and Previous Nontarget Web Services
Input: node of abstract target web service in tree of workflow (NWS), chromosome that shows which candidate web services are selected from abstract web services (chromosome)
Output: existence of correlation for candidate web service is selected from NWS
1: selected web services = selectedWebServices.feature.Target(NWS, chromosome)
2: for $\forall a \in$ selected web services do
3: if checkCorrelation(a, NWS, chromosome) then
4: return true
5: end if
6: end for
7: return false

الگوریتم ۳-۵: الگوریتم بررسی وجود همبستگی بین وب‌سرویس هدف با وب‌سرویس‌های نامجاور (برای بررسی امکان کاهش هزینه‌ی اجرایی در وب‌سرویس هدف از این الگوریتم می‌توان استفاده کرد).

۳-۶- آزمایش بررسی تأثیر افزایش همبستگی بین وب‌سرویس‌ها بر میزان برازندگی وب‌سرویس مرکب یافت شده

در این آزمایش برای جریان کاری نمونه‌ای با نه وب‌سرویس انتزاعی و ده وب‌سرویس کاندید برای هر یک از وب‌سرویس‌های انتزاعی، ارتباط بین درصد وجود همبستگی بین وب‌سرویس‌های کاندید و میزان برازندگی وب‌سرویس مرکب یافت شده را بررسی می‌کنیم.



شکل ۶-۱: جریان کاری آزمایش تأثیر همبستگی بر میزان برازندگی وب‌سرویس مرکب یافت شده

همانطور که در بخش قبل اشاره کردیم، با توجه به ویژگی کیفی مورد بررسی، نیازمند بررسی همبستگی بین وب‌سرویس‌های مجاور یا نامجاور هستیم که برای هر کدام از این حالت‌ها الگوریتمی پیشنهاد داده شد در ادامه آزمایش را برای هر دو الگوریتم بررسی همبستگی بین وب‌سرویس‌ها مجاور یا نامجاور انجام داده و نمودار نتایج را ارائه می‌دهیم.

برای آزمایش اول، ده عدد تصادفی تولید می‌کنیم و به هر یک از وب‌سرویس‌های کاندید هر وب‌سرویس انتزاعی یکی از این اعداد تصادفی تولید شده که نشان دهنده مدت زمان پاسخ آن وب‌سرویس است را انتصاب می‌دهیم. برای بررسی همبستگی و محاسبه‌ی مجموع ویژگی کیفی در صورت وجود همبستگی، در هر یک از این وب‌سرویس‌ها مدت زمان پاسخ را تقسیم بر سه کرده و مدت زمان انتقال اطلاعات به وب‌سرویس و میانگین زمان اجرایی وب‌سرویس و مدت زمان خروج اطلاعات از وب‌سرویس را بدست آورده تا اعداد مورد نظر برای صرف نظر کردن از آن‌ها در محاسبات مجموع ویژگی کیفی در صورت وجود همبستگی را داشته باشیم.

۱-۶- ویژگی‌های الگوریتم ژنتیک پایه

الگوریتم ژنتیک را به عنوان الگوریتم پایه‌ای که الگوریتم پیشنهادی را با آن مقایسه می‌کنیم، در نظر گرفته‌ایم. دلیل این انتخاب این است که الگوریتم پیشنهادی بر مبنای الگوریتم ژنتیک و ایجاد تغییراتی بر روی آن ارائه شده است.

الگوریتم ژنتیکی که به عنوان الگوریتم پایه برای مقایسه با الگوریتم ژنتیک پیشنهادی در نظر گرفته شده است، الگوریتمی است که در صورت مشاهده‌ی نقض محدودیت‌های وابستگی و ناسازگاری ساده‌ترین روش را اتخاذ کرده و آن کروموزوم‌ی که محدودیت‌ها را نقض کرده، حذف می‌کند و برازندگی آن کروموزوم را صفر قرار می‌دهد، همچنین این الگوریتم نمی‌تواند همبستگی بین وب‌سرویس‌ها را تشخیص داده و تنها ویژگی‌های کیفی اصلی وب‌سرویس‌ها را بدون در نظر گرفتن وجود همبستگی در محاسبه‌های مربوط به برازندگی اعمال می‌کند.

اندازه جمعیت کروموزوم‌ها در هر نسل که تولید و بررسی می‌شوند، ۵۰۰ کروموزوم در نظر گرفته می‌شود. احتمال رخ دادن تقاطع $0/8$ و رخ دادن جهش $0/5$ در نظر گرفته می‌شود. تعداد دفعات تولید نسل برای یافتن بهترین ترکیب ۱۰۰۰۰ در نظر گرفته می‌شود.

۲-۶- ویژگی‌های الگوریتم ژنتیک پیشنهادی

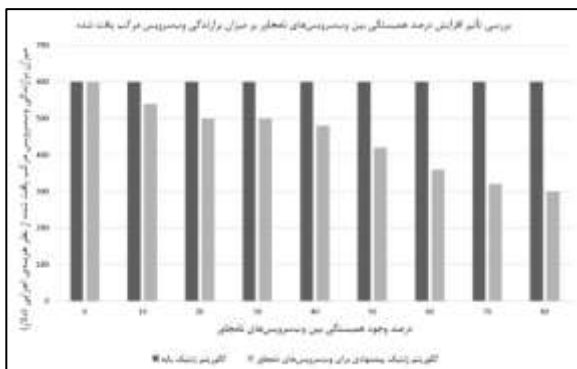
الگوریتم ژنتیک پیشنهادی در صورت مشاهده‌ی نقض محدودیت‌های وابستگی و ناسازگاری توسط کروموزوم‌ها همانطور که در بخش‌های قبل اشاره شد، این کروموزوم‌های ناقص را حذف نکرده و تنها آن‌ها را جریمه کرده و اجازه‌ی ایجاد نسل‌هایی از آن‌ها را می‌دهد، زیرا ممکن است با تقاطع یا جهش روی همین کروموزوم‌های ناقص، کروموزومی با جواب بهینه بیابد ولی برازندگی این کروموزوم‌های ناقص محدودیت‌ها را به اندازه‌ای کاهش می‌دهد تا مطمئن شود که خود این کروموزوم‌های ناقص به عنوان راه‌حل انتخاب نمی‌شوند.

الگوریتم ژنتیک پیشنهادی، می‌تواند همبستگی بین وب‌سرویس‌ها را تشخیص دهد بنابراین در محاسبات مربوط به برازندگی وب‌سرویس مرکب در صورت یافتن همبستگی بین وب‌سرویس‌ها از ویژگی کیفی بهبود یافته‌ی وب‌سرویس دارای همبستگی با دیگر وب‌سرویس‌ها استفاده می‌کند.

الگوریتم ژنتیک پیشنهادی بر پایه‌ی الگوریتم ژنتیک ارائه شده است و برای اینکه تنظیمات پارامترها روی نتایج مقایسه تأثیری نگذارد، برای تنظیم پارامترهای الگوریتم ژنتیک پیشنهادی نیز از پارامترهایی که به کمک آن الگوریتم ژنتیک پایه را تنظیم کرده‌ایم، استفاده می‌کنیم.

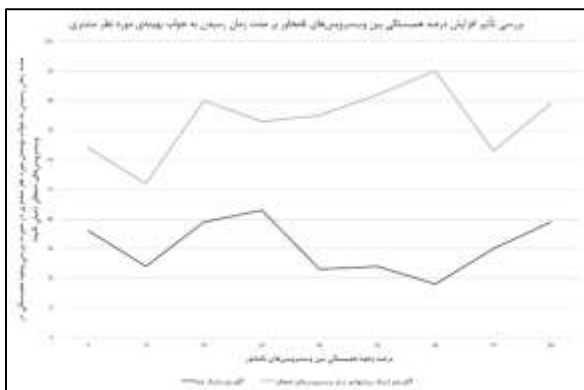
ویژگی کیفی هزینه‌ی اجرایی و سرویس مرکب نهایی مورد بررسی قرار گرفته است.

برای آزمایش دوم، ده عدد تصادفی تولید می‌کنیم و به هر یک از وب‌سرویس‌های کاندیدای هر وب‌سرویس انتزاعی یکی از این اعداد تصادفی تولید شده که نشان دهنده‌ی هزینه‌ی اجرایی آن وب‌سرویس است را انتصاب می‌دهیم. برای بررسی همبستگی و محاسبه‌ی مجموع ویژگی کیفی در صورت وجود همبستگی، در هر یک از این وب‌سرویس‌ها ۱۰ درصد از هزینه اجرایی را کاهش می‌دهیم تا اعداد مورد نظر برای اعمال در محاسبات مجموع ویژگی کیفی در صورت وجود همبستگی را داشته باشیم.



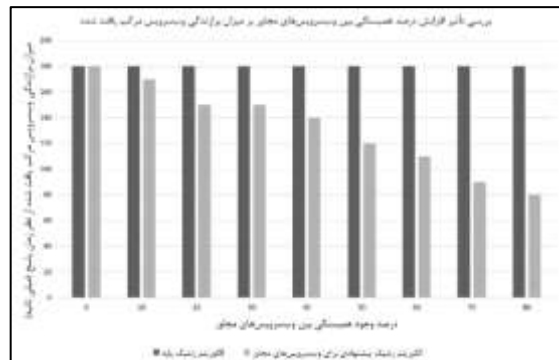
شکل ۴-۶: آزمایش بررسی تأثیر افزایش درصد همبستگی بین وب‌سرویس‌های نامجاور بر میزان برازندگی وب‌سرویس مرکب یافت شده

در این آزمایش مدت زمان رسیدن به جواب بهینه‌ی مورد نظر مشتری را در صورت بررسی وجود همبستگی بررسی می‌کنیم و در نتایج بدست آمده را در ادامه می‌آوریم. در این آزمایش مدت زمان رسیدن به جواب بهینه برای الگوریتم ژنتیک پایه در صورتی که همواره مسئله ثابت است، تغییر می‌یابد که این نتیجه به دلیل ماهیت تصادفی بودن الگوریتم ژنتیک رخ می‌دهد.



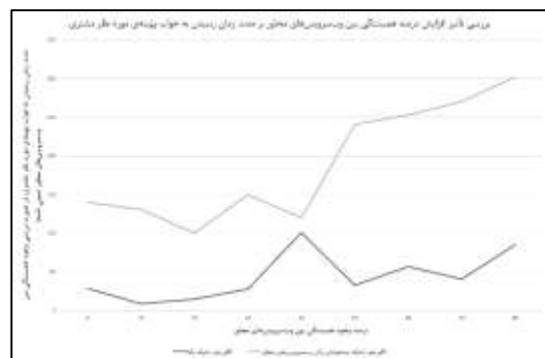
شکل ۵-۶: آزمایش بررسی تأثیر افزایش درصد همبستگی بین وب‌سرویس‌های نامجاور بر مدت زمان رسیدن به جواب بهینه‌ی مورد نظر مشتری

در آزمایش اول برای بررسی تأثیر افزایش درصد همبستگی بین وب‌سرویس‌های مجاور بر میزان برازندگی وب‌سرویس مرکب یافته، ویژگی کیفی زمان پاسخ وب‌سرویس مرکب نهایی مورد بررسی قرار گرفته است.



شکل ۲-۶: آزمایش بررسی تأثیر افزایش درصد همبستگی بین وب‌سرویس‌های مجاور بر میزان برازندگی وب‌سرویس مرکب یافت شده

در این آزمایش مدت زمان رسیدن به جواب بهینه‌ی مورد نظر مشتری را در صورت بررسی وجود همبستگی بررسی می‌کنیم و در نتایج بدست آمده را در ادامه می‌آوریم. در این آزمایش مدت زمان رسیدن به جواب بهینه برای الگوریتم ژنتیک پایه در صورتی که همواره مسئله ثابت است، تغییر می‌یابد که این نتیجه به دلیل ماهیت تصادفی بودن الگوریتم ژنتیک رخ می‌دهد.

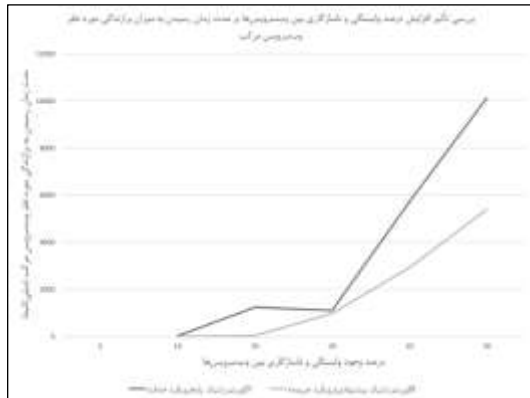


شکل ۳-۶: آزمایش بررسی تأثیر افزایش درصد همبستگی بین وب‌سرویس‌های مجاور بر مدت زمان رسیدن به جواب بهینه‌ی مورد نظر مشتری

همانطور که مشاهده می‌کنید در صورت بررسی وجود همبستگی بین وب‌سرویس‌های مجاور، مدت زمان دستیابی به جواب بهینه‌ی مورد نظر مشتری مقداری افزایش پیدا می‌کند.

در آزمایش دوم برای بررسی تأثیر افزایش درصد همبستگی بین وب‌سرویس‌های نامجاور بر میزان برازندگی وب‌سرویس مرکب یافته،

وب‌سرویس مرکب، عملکرد دو الگوریتم ژنتیک پایه و پیشنهادی را با هم مقایسه می‌کنیم.



شکل ۶-۶: آزمایش بررسی تأثیر افزایش درصد وابستگی و ناسازگاری بین وب‌سرویس‌ها بر مدت زمان رسیدن به میزان برانزندی مورد نظر وب‌سرویس مرکب

همانطور که در نمودار بالا مشاهده می‌کنید، به میزانی که درصد وابستگی و ناسازگاری بین وب‌سرویس‌ها بیشتر می‌شود، الگوریتم ژنتیک پیشنهادی که از رویکرد دوم که راه‌حل ناشدنی را جریمه می‌کند، استفاده می‌کند در مدت زمان بهتری وب‌سرویس مرکب با برانزندی مورد نظر مشتری را می‌یابد.

۷- مقایسه الگوریتم ژنتیک پیشنهادی با الگوریتم ژنتیک پایه

در ادامه جدول ۷-۱ که نشان‌دهنده مقایسه الگوریتم ژنتیک پیشنهادی (که بهبود یافته الگوریتم ژنتیک پایه است) با کارهای پیشین است، آورده شده است.

شاخص‌های مورد ارزیابی	الگوریتم ژنتیک پیشنهادی (الگوریتم پیشنهادی)	الگوریتم ژنتیک پیشنهادی (الگوریتم پیشنهادی)	الگوریتم ژنتیک پیشنهادی (الگوریتم پیشنهادی)	الگوریتم ژنتیک پیشنهادی (الگوریتم پیشنهادی)
فرآیند انتخاب وب‌سرویس‌ها	فرآیند انتخاب وب‌سرویس‌ها	فرآیند انتخاب وب‌سرویس‌ها	فرآیند انتخاب وب‌سرویس‌ها	فرآیند انتخاب وب‌سرویس‌ها
پارامترهای ژنتیک	پارامترهای ژنتیک	پارامترهای ژنتیک	پارامترهای ژنتیک	پارامترهای ژنتیک
امکان بررسی وجود همبستگی بین وب‌سرویس‌ها	امکان بررسی وجود همبستگی بین وب‌سرویس‌ها	امکان بررسی وجود همبستگی بین وب‌سرویس‌ها	امکان بررسی وجود همبستگی بین وب‌سرویس‌ها	امکان بررسی وجود همبستگی بین وب‌سرویس‌ها
امکان بررسی وجود وابستگی و ناسازگاری بین وب‌سرویس‌ها	امکان بررسی وجود وابستگی و ناسازگاری بین وب‌سرویس‌ها	امکان بررسی وجود وابستگی و ناسازگاری بین وب‌سرویس‌ها	امکان بررسی وجود وابستگی و ناسازگاری بین وب‌سرویس‌ها	امکان بررسی وجود وابستگی و ناسازگاری بین وب‌سرویس‌ها

در جدول مقایسه‌ای الگوریتم ژنتیک پیشنهادی و الگوریتم ژنتیک پایه، علامت سبز نشان‌دهنده برتری الگوریتم پیشنهادی و علامت قرمز نشان‌دهنده برتری الگوریتم پایه است.

جدول ۷-۱: جدول مقایسه الگوریتم ژنتیک پیشنهادی با الگوریتم ژنتیک پایه

همانطور که مشاهده می‌کنید در صورت بررسی وجود همبستگی بین وب‌سرویس‌های نامجاور، مدت زمان دستیابی به جواب بهینه‌ی مورد نظر مشتری مقداری افزایش پیدا می‌کند.

همانطور که در دو آزمایش بالا مشاهده کردید، به میزانی که درصد همبستگی بین وب‌سرویس‌ها افزایش پیدا می‌کند، امکان انتخاب وب‌سرویس‌هایی که با وب‌سرویس‌های دیگر همبسته بوده و ویژگی کیفی بهتری دارند، نیز افزایش پیدا می‌کند، در نتیجه مجموع ویژگی کیفی یا میزان برانزندی وب‌سرویس مرکب نهایی که از ترکیب این وب‌سرویس‌های همبسته به دست می‌آید، بهبود می‌یابد.

۴-۶- آزمایش بررسی تأثیر افزایش وابستگی و ناسازگاری بین وب‌سرویس‌ها بر مدت زمان رسیدن به میزان برانزندی مورد نظر وب‌سرویس مرکب

همانطور که قبلاً اشاره کردیم در صورتی که کروموزوم، وب‌سرویس‌هایی که محدودیت‌های وابستگی یا ناسازگاری را نقض کرده باشند را برای ترکیب با هم انتخاب کرده باشد، دو رویکرد داریم. رویکرد اول این است که این کروموزوم حاوی راه‌حل ناشدنی را کامل از جمعیت حذف کنیم، این رویکرد در الگوریتم ژنتیک پایه پیاده‌سازی شده است. رویکرد دوم این است که کروموزوم حاوی راه‌حل ناشدنی را جریمه کرده ولی از جمعیت حذف نکنیم زیرا ممکن است در صورت حذف نکردن این کروموزوم با عملگرهای تقاطع و جهش ما سریع‌تر به کروموزوم‌هایی حاوی راه‌حل شدنی دست یافته و جواب مورد نظر مشتری را پیدا کنیم بنابراین در الگوریتم ژنتیک پیشنهادی رویکرد جریمه‌ای را پیاده‌سازی کرده‌ایم.

در ادامه این دو رویکرد را با هم مقایسه کرده و نشان می‌دهیم که با استفاده از رویکرد دوم در مدت زمان کمتری به راه‌حلی برای انتخاب و ترکیب وب‌سرویس‌های مناسب دست می‌یابیم که میزان برانزندی مورد نظر مشتری برای وب‌سرویس مرکب را برآورده می‌سازد.

برای این آزمایش شش وب‌سرویس انتزاعی به صورت ترتیبی با هم ترکیب شده‌اند و برای هر وب‌سرویس انتزاعی شش وب‌سرویس کاندیدا برای انتخاب شدن وجود دارد و وب‌سرویس‌های وابسته و ناسازگار به صورت تصادفی انتخاب می‌شوند. مدت زمان رسیدن به برانزندی مورد نظر برای وب‌سرویس مرکب برای دو الگوریتم ژنتیک پایه و پیشنهادی محاسبه و با یکدیگر مقایسه می‌شوند.

در این آزمایش با بررسی تأثیر افزایش درصد وابستگی و ناسازگاری بین وب‌سرویس‌ها بر مدت زمان رسیدن به برانزندی مورد نظر برای

۸- نتیجه‌گیری

الگوریتم ژنتیک پیشنهادی این مقاله برای مسئله‌ی انتخاب و ترکیب وب‌سرویس‌ها از همبستگی، وابستگی و ناسازگاری در ترکیب ترتیبی، موازی، شرطی و حلقوی پشتیبانی کرده و وب‌سرویس مرکب را با مجموع ویژگی‌های کیفی مورد نظر مشتری یا حتی مجموعی بهتر در صورت امکان و انتخاب وب‌سرویس‌های همبسته، می‌یابد.

الگوریتم ژنتیک پیشنهادی در این مقاله در صورت نقض محدودیت‌های وابستگی و ناسازگاری از رویکرد جریمه استفاده می‌کند و همانطور که در بخش ارزیابی نشان داده شد، استفاده از رویکرد جریمه نسبت به رویکرد حذفی سبب می‌شود که راه‌حل ناشدنی را حذف نکرده و همین راه‌حل‌های ناشدنی به ما در سریع‌تر رسیدن به راه‌حل شدنی کمک کرده و سریع‌تر به وب‌سرویس مرکب مورد نظر مشتری دست بیاییم.

الگوریتم ژنتیک پیشنهادی در این مقاله به دلیل ماهیت تصادفی الگوریتم ژنتیک نسبت به الگوریتم‌های دیگری که از فنون مبتنی بر برنامه‌ریزی خطی استفاده می‌کنند و تمام حالات را چک می‌کنند، سریع‌تر بوده و حافظه کمتری استفاده می‌کند.

۹- کارهای آینده

بررسی امکان استفاده از شبکه عصبی در الگوریتم ژنتیک به طوری که ابتدا الگوریتم ژنتیک به صورت عادی عملیات خود را انجام دهد و میزان برازندگی کروموزوم‌ها را با در نظر گرفتن تمام روابط و حالت‌های بین وب‌سرویس‌ها محاسبه کند و در همین حال کروموزوم و میزان برازندگی آن به الگوریتم شبکه‌ی عصبی ورودی داده می‌شود تا الگوریتم یاد بگیرد بعد از یادگیری کافی الگوریتم شبکه عصبی، دیگر نیازی به محاسبه میزان برازندگی کروموزوم‌ها با در نظر گرفتن تمام روابط و حالت‌های بین وب‌سرویس‌ها به صورت عادی وجود ندارد و الگوریتم شبکه عصبی میزان برازندگی کروموزوم را پیش‌بینی می‌کند، استفاده از این امر سبب می‌شود که تقریباً تا اواسط کار، الگوریتم ژنتیک به صورت معمولی کار خود را انجام دهد ولی از اواسط کار که یادگیری الگوریتم شبکه عصبی کامل شده است دیگر نیاز به زمانی برای محاسبه برازندگی کروموزوم‌ها با در نظر گرفتن تمام روابط و حالت‌های بین وب‌سرویس‌ها نداریم که همین مورد سبب دو برابر شدن سرعت الگوریتم ژنتیک می‌شود.

موازی سازی الگوریتم ژنتیک، به کمک این روش محاسبه‌ی میزان برازندگی چندین کروموزوم به طور همزمان قابل انجام است.

۱۰- مراجع

- [۱] Deng S, Wu H, Hu D, Zhao JL (2016) Service selection for composition with QoS correlations. *IEEE Transactions on Services Computing*, 9(2), 291-303.
- [۲] da Silva AS, Ma H, Zhang M (2016) Genetic programming for QoS-aware web service composition and selection. *Soft Computing*, 20(10), 3851-3867.
- [۳] Yu Q, Chen L, Li B (2015) Ant colony optimization applied to web service compositions in cloud computing. *Computers & Electrical Engineering*, 41, 18-27.
- [۴] Yilmaz AE, Karagoz P (2014) Improved genetic algorithm based approach for QoS aware web service composition. In *Web Services (ICWS), 2014 IEEE International Conference on* (pp. 463-470). IEEE.
- [۵] Chen M, Wang ZW (2007) An approach for web services composition based on QoS and discrete particle swarm optimization. In *Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing, 2007. SNPD 2007. Eighth ACIS International Conference on* (Vol. 2, pp. 37-41). IEEE.
- [۶] Amiri MA, Serajzadeh H (2010) QoS aware web service composition based on genetic algorithm. In *Telecommunications (IST), 2010 5th International Symposium on* (pp. 502-507). IEEE.
- [۷] Lin SY, Lai CH, Wu CH, Lo CC (2014) A trustworthy QoS-based collaborative filtering approach for web service discovery. *Journal of Systems and Software*, 93, 217-228.
- [۸] Wang P, Chao KM, Lo CC (2010) On optimal decision for QoS-aware composite service selection. *Expert Systems with Applications*, 37(1), 440-449.
- [۹] Wang L, Shen J, Yong J (2012) A survey on bio-inspired algorithms for web service composition. In *Computer Supported Cooperative Work in Design (CSCWD), 2012 IEEE 16th International Conference on* (pp. 569-574). IEEE.
- [۱۰] Alrifai M, Risse T, Nejdl W (2012) A hybrid approach for efficient Web service composition with end-to-end



- [۱۸] Ai L, Tang M (2008) A penalty-based genetic algorithm for QoS-aware web service composition with inter-service dependencies and conflicts. In Computational Intelligence for Modelling Control & Automation, 2008 International Conference on (pp. 738-743). IEEE.
- [۱۹] Zhang LJ, Li B (2004) Requirements driven dynamic services composition for web services and grid solutions. Journal of Grid Computing, 2(2), 121-140.
- [۲۰] Ardagna D, Pernici B (2007) Adaptive service composition in flexible processes. IEEE Transactions on software engineering, 33.
- [۲۱] Canfora G, Di Penta M, Esposito R, Villani ML (2005) 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). ACM.
- [۲۲] Jaeger MC, Mühl G (2007) QoS-based selection of services: The implementation of a genetic algorithm. In Communication in Distributed Systems (KiVS), 2007 ITG-GI Conference (pp. 1-12). VDE.
- [۲۳] Su S, Zhang C, Chen J (2007) An improved genetic algorithm for web services selection. In IFIP International Conference on Distributed Applications and Interoperable Systems (pp. 284-295). Springer, Berlin, Heidelberg.
- [۲۴] QoS constraints. ACM Transactions on the Web (TWEB), 6(2), 7.
- [۲۵] Jula A, Othman Z, Sundararajan E (2015) Imperialist competitive algorithm with PROCLUS classifier for service time optimization in cloud computing service composition. Expert Systems with applications, 42(1), 135-145.
- [۲۶] Mardukhi F, Nematbakhsh N, Zamanifar K, Barati A (2013) QoS decomposition for service composition using genetic algorithm. Applied Soft Computing, 13(7), 3409-3421.
- [۲۷] Tang M, Ai L (2010) A hybrid genetic algorithm for the optimal constrained web service selection problem in web service composition. In Evolutionary Computation (CEC), 2010 IEEE Congress on (pp. 1-8). IEEE.
- [۲۸] Canfora G, Di Penta M, Esposito R, Villani ML (2008) A framework for QoS-aware binding and re-binding of composite web services. Journal of Systems and Software, 81(10), 1754-1769.
- [۲۹] Gao F, Curry E, Ali MI, Bhiri S, Mileo A (2014) QoS-aware complex event service composition and optimization using genetic algorithms. In International Conference on Service-Oriented Computing (pp. 386-393). Springer, Berlin, Heidelberg.
- [۳۰] Gao C, Cai M, Chen H (2007) QoS-aware service composition based on tree-coded genetic algorithm. In Computer Software and Applications Conference, 2007. COMPSAC 2007. 31st Annual International (Vol. 1, pp. 361-367). IEEE.
- [۳۱] Wang D, Yang Y, Mi Z (2015) A genetic-based approach to web service composition in geo-distributed cloud environment. Computers & Electrical Engineering, 43, 129-141.

پانویس‌ها

[^] Recombination

[^] Mutation

[^] Microsoft

[^] Amazon

[^] Depth-first Search (DFS)

[^] Post-order

[^] Workflow

[^] Quality of Service (QoS)

[^] NP-Hard

[^] Gene

[^] Chromosome

[^] Genome

[^] Crossover