بهینه‌سازی ترکیب سرویس‌های ابری با آگاهی از کیفیت سرویس با استفاده از الگوریتم جستجوی موجودات زنده همبسته

وحیده حی‌العالم[۱] – علی اصغر پورحاجی کاظم[۲]

(۱) دانشجوی کارشناسی ارشد – گروه مهندسی کامپیوتر، دانشکده فنی و مهندسی، واحد تبریز، دانشگاه آزاد اسلامی، تبریز، ایران

(۲) استادیار – گروه مهندسی کامپیوتر، دانشکده فنی و مهندسی، واحد تبریز، دانشگاه آزاد اسلامی، تبریز، ایران

تاریخ دریافت: ۱۳۹۶/۵/۶ تاریخ پذیرش: ۱۳۹۶/۱۰/۳۰

خلاصه: امروزه سیستم‌های سرویس‌گرا با توجه به امکان استفاده در محیط‌های توزیع شده ناهمگون و عدم وابستگی به فناوری خاص، و نیز وجود سیستم‌های مقیاس وسیع پویا با نیازهای متغیر، بسیار مورد توجه قرار گرفته‌اند. سرویس‌های ابری نیز یکی از روش‌های پیاده‌سازی مفهوم سرویس-گرایی است. افزایش تمایل کاربران به استفاده از رایانش ابری، فروشندگان سرویس‌های وب را به تولید سرویس‌هایی با ویژگی‌های غیرعملکردی (کیفیت سرویس) متفاوت، تشویق می‌کند. نیازهای کاربران در اغلب اوقات توسط سرویس‌های تکی موجود رفع نمی‌گردد و بایستی با ترکیب تعدادی از سرویس‌ها به سرویس مورد نظر دست یافت. تعداد ارائه‌دهندگان سرویس روز به روز در حال افزایش است و به موازات آن، برای یک درخواست، چندین سرویس با عملکرد یکسان و با ویژگی‌های غیرعملکردی متفاوت پیشنهاد می‌شود، که این باعث می‌شود مساله ترکیب سرویس کاملاً پیچیده شده و جزو مسائل Np-Hard محسوب شود. بنابراین افزایش کیفیت سرویس‌های مرکب چالشی بزرگ می‌باشد و از آنجایی که نتایج کارهای پیشین کیفیت لازم را ندارند لذا در این تحقیق سعی شده است با استفاده از الگوریتم جستجوی موجودات زنده همبسته، این چالش حل می‌شود. تمامی شبیه‌سازی‌ها در محیط متلب اجرا شده و نتایج آنها با الگوریتم‌های ژنتیک، ازدحام ذرات و کلونی مورچگان مقایسه شده که نشان می‌دهد روش پیشنهادی از نظر کیفیت جواب‌ها، پایداری و مقیاس‌پذیری، برتری قابل‌توجهی نسبت به سایر الگوریتم‌ها دارد، همچنین دارای ۱۳٪ بهبود نسبت به سایر روش‌های مورد مقایسه می‌باشد.

کلمات کلیدی: رایانش ابری، ترکیب سرویس، کیفیت سرویس، معماری سرویس‌گرا، بهینه‌سازی، الگوریتم جستجوی موجودات زنده همبسته

QoS-Aware Optimization of Cloud Service Composition Using Symbiotic Organisms Search Algorithm

Vahideh Hayyolalam [1] – Ali Asghar Pourhaji Kazem [2]

(1) MSc - Department of Computer Engineering, Tabriz Branch, Islamic Azad University, Tabriz, Iran,
p.hayyolalam@gmail.com

(2) Assistant Professor - Department of Computer Engineering, Tabriz Branch, Islamic Azad University, Tabriz, Iran,
pourhajikazem@hotmail.com

Abstract

Nowadays, service-oriented systems according to possibility of using in heterogeneous distributed environments and being independent of the specific technology, also with the existence of large-scale dynamic system with changeable requirements, are highly regarded. Cloud services are one of the implementation methods of service-oriented concepts. Increasing the tendency of users to use cloud computing, encourages service vendors to provide services with different non-functionality features. Mostly single services couldn't satisfy users' requirements, so it's necessary to compose some services to achieve the demand service. With the increasing of service providers, also services in the internet service pools, selecting the optimal service from a set of functionality equivalent candidates which are differ in QoS, becomes an important NP-Hard research problem. Therefore increasing the quality of composite services is a vital challenge. Due to insufficient quality of current approaches' results, this research employs Symbiotic Organism Search Algorithm (SOS) to solve aforementioned challenge. Simulations are conducted in Matlab environment and the results are compared to three famous algorithms including GA, ACO and PSO. The comparisons demonstrate the remarkable superiority of SOS in result's quality, stability and scalability, also 13% improvement.

Index Terms: Cloud computing, service composition, QoS, SOA, optimization, Symbiotic Organism Search

نویسنده مسئول: علی اصغر پورحاجی کاظم، استادیار – گروه مهندسی کامپیوتر، دانشکده فنی و مهندسی، واحد تبریز، دانشگاه آزاد اسلامی، تبریز، ایران، pourhajikazem@hotmail.com

۱- مقدمه

با پیشرفت فناوری اطلاعات نیاز به انجام کارهای محاسباتی در همه جا و همه زمان به وجود آمده است. همچنین نیاز به این است که افراد بتوانند کارهای محاسباتی سنگین خود را بدون داشتن سخت‌افزارها و نرم‌افزارهای گران، از طریق سرویس‌ها انجام دهند. رایانش ابری آخرین پاسخ فناوری به این نیازها بوده است. رایانش ابری مدلی است برای فراهم کردن دسترسی آسان بر اساس تقاضای کاربر از طریق شبکه به مجموعه‌ای از منابع رایانشی قابل تغییر و پیکربندی (مثل: شبکه‌ها، سرورها، فضای ذخیره‌سازی، برنامه‌های کاربردی و سرویس‌ها) که این دسترسی بتواند با کمترین نیاز به مدیریت منابع و یا نیاز به دخالت مستقیم فراهم‌کننده سرویس به سرعت فراهم شود [۱]. با رشد سریع معماری سرویس‌گرا، سرویس‌ها به عنوان مولفه‌هایی با قابلیت اتصال سست[1] برای توسعه انواع گوناگون کاربردها به کار گرفته می‌شوند.

یکی از مهمترین چالش‌های استفاده از سرویس‌های ابری مفهوم ترکیب سرویس می‌باشد. نیاز به استفاده از سرویس‌های بسیار پیچیده متفاوت، باعث شد که آماده کردن و ارائه کردن این سرویس‌ها به صورت مستقل برای ارائه‌دهندگان سرویس‌ها عملاً غیرممکن باشد. بنابراین لازم است تأمین کنندگان ابر از شیوه‌هایی استفاده کنند تا هر نوع سرویس پیچیده مرکبی را که درخواست می‌شود با قرار دادن سرویس‌های در دسترس کنار هم، بتوانند ارائه کنند [۲].

کاربر درخواست پیچیده خود را همراه با محدودیت‌های مورد نظرش به سیستم می‌دهد. این درخواست به صورت یک DAG مدل می‌شود. در این DAG هر کدام از گره‌ها نشان‌دهنده یک وظیفه[2] و یال‌ها نشان‌دهنده روابط بین آنها است. برای هر کدام از وظایف ممکن است چندین سرویس پیشنهاد شود که همه این سرویس‌ها از نظر عملکردی یکسان هستند ولی ویژگی‌های غیرعملکردی (برای مثال ویژگی‌های کیفیت سرویس) متفاوتی دارند. حال مساله اصلی این است که از بین سرویس‌های پیشنهادی کدامیک را انتخاب کنیم تا نیازهای کاربر را با توجه به محدودیت‌های خواسته شده‌اش برآورده سازد. واضح است در صورتی که درخواست فقط شامل یک وظیفه باشد در این صورت انتخاب آن از بین سرویس‌های پیشنهاد شده بر اساس محدودیت‌های مورد تقاضا کار سختی نخواهد بود اما با ازدیاد تعداد وظایف، انتخاب یک ترکیب بهینه از بین سرویس‌های پیشنهاد شده به طوری که تمام محدودیت‌های درخواستی کاربر را به بهترین وجه ممکن تامین کند، بسیار مشکل خواهد بود که با استفاده از روش‌های قطعی نمی‌توان با پیچیدگی زمانی خطی به جواب رسید. پس مساله تبدیل می‌شود به یک مساله Np-hard [۳]، [۴] که برای حل آن از روش‌های هوش مصنوعی و ابتکاری استفاده می‌کنیم تا به جوابی نزدیک به جواب بهینه دست یابیم [۴].

۱-۲- بیان رسمی مسئله ترکیب سرویس

مسئله ترکیب سرویس‌های ابری آگاه از کیفیت سرویس، یافتن یک مجموعه از سرویس‌های نامزدی ابری[3] با ویژگی‌های عملکردی مختلف می‌باشد که اولاً محدودیت‌های مشخص شده توسط کاربر را رعایت کرده و ثانیاً تابع هدف را بهینه نماید. در این بخش مسئله مذکور به طور رسمی بیان می‌گردد. یک نمونه از مسئله ترکیب سرویس‌های ابری آگاه از کیفیت سرویس به طور رسمی به شکل زیر بیان می‌گردد:

- یک درخواست ترکیب سرویس به شکل جریان‌کار[4] که با استفاده از یک گراف جهتدار جهت‌دار[5] مدل می‌گردد G=(V,E).
- $V=\{T_1,T_2,\ldots, T_n\}$ که n تعداد کاربرهای در جریان کار می‌باشد.
- E: مجموعه یال‌ها که اولویت اجرای کارها را نشان می‌دهد.
- هر کار T_i $(1 \le i \le n)$ در جریان‌کار یک مجموعه از سرویس‌های نامزدی دارد $CS_i=\{cs_i^1, cs_i^2, \ldots, cs_i^{m_i}\}$ که در آن CS_i^j $(1 \le j \le m_i)$ یک سرویس نامزدی ابری است.
- m_i: تعداد کل سرویس‌های نامزدی موجود برای کار T_i.
- هر سرویس نامزدی CS_i^j یک مجموعه از اطلاعات کیفیت سرویس مختلف دارد $QoS_i^j=\{Q_1, Q_2,\ldots, Q_k\}$ که در آن Q_l $(1 \le l \le k)$ یک ویژگی کیفیت سرویس از سرویس‌های ابری را نشان می‌دهد.
- اطلاعات کیفیت سرویس مربوط به سرویس‌های ابری در انباره کیفیت سرویس ذخیره می‌گردد.
- K: تعداد ویژگی‌های کیفیت سرویس مربوط به سرویس‌های ابری که در مدل کیفیت سرویس استفاده شده است.
- QC: مجموعه محدودیت‌های سراسری که توسط کاربر تعیین گردیده است $QC=(C_1, C_2, \ldots, C_k)$.

با در نظر گرفتن موارد فوق هدف مسئله ترکیب سرویس آگاه از کیفیت سرویس یافتن سرویس مرکب ابری نزدیک بهینه است به قسمی که برای $\forall j = 1 \ldots K$ داریم:

$$\begin{cases} \sum_{i=1}^{n} S_i.Q_j < C_j & \text{if } Q_j \text{ is additive} \\ \prod_{i=1}^{n} S_i.Q_j > C_j & \text{if } Q_j \text{ is multiplicative} \end{cases} \quad (۱)$$

۲- کارهای پیشین

ترکیب سرویس در واقع ترکیب سرویس‌های تکی برای برآورده کردن درخواست‌های پیچیده کاربران است که با سرویس‌های تکی، انجام آن امکان‌پذیر نیست. در این بخش به مرور برخی کارهای قبلی انجام شده در این زمینه پرداخته می‌شود. بررسی چگونگی ترکیب سرویس‌های کاندید برای دستیابی به سرویس مرکب ابری بهینه، هدف این تحقیق می‌باشد که برای این منظور از الگوریتم بهینه‌سازی سیمبیوتیک بهره برده است.

الگوریتم همه ابرها [۵]، [۶] یک ترکیب ابری پیدا می‌کند، اما با این حال این الگوریتم هیمشه یک ترکیب دارای بیشترین تعداد ابر را به دست می‌آورد. الگوریتم ابر پایه [۵]، [۶] یک ترکیب ابری که حاوی کمترین تعداد ابر است پیدا می‌کند، با این وجود زمانی که تعداد ابرها افزایش پیدا می‌کند مشکل پیچیدگی زمانی قابل توجه می‌شود. الگوریتم ابر پایه و ترکیب ابری حریصانه [۶]، ترکیب مناسب را در بیشتر موارد پیدا می‌کنند ولی در برخی از محیط‌های چند ابری با شکست مواجه

می‌شوند. عملکرد الگوریتم ابر هوشمند و روش پیشنهادی در [۷]، در پیدا کردن ترکیب ابری مناسب مشابه همدیگر است و ترکیب ابری مناسب را در همه موارد نمی‌توانند پیدا کنند. الگوریتم مورچگان [۶] ترکیب ابری مناسب را در همه موارد پیدا می‌کند و نیز زمان اجرایی آن مناسب می‌باشد. الگوریتم پیشنهادی در [۸] که از خوشه‌بندی و الگوریتم کلونی مورچگان استفاده کرده است ترکیب ابری مناسب را در زمانی بیشتر از [۶] به دست می‌آورد. بنابراین الگوریتم‌های کلونی مورچگان [۶]، ابر پایه [۵]، [۶] و خوشه‌بندی [۸] عملکرد تقریباً یکسان و بهتری نسبت به سایر رویکردهای مذکور در پیدا کردن ترکیب ابری مناسب دارند.

در [۹] با بکارگیری عملگر Skyline و الگوریتم ازدحام ذرات به یک ترکیب سرویس بهینه در زمانی بسیار سریع دست می‌یابد ولی پایداری آن کم است. در [۱۰] یک رویکرد جدید بر پایه ازدحام ذرات و با بهره-گیری از یادگیری اجتماعی معرفی شده و از آن برای حل مساله ترکیب سرویس با آگاهی از کیفیت سرویس بهره‌گرفته شده است که پیچیدگی زمانی آن $O(n^2)$ می‌باشد و از مزایای آن می‌توان به همگرایی سریع، استفاده از مقادیر کیفیت سرویس پویا و نیز قابلیت جستجو اشاره کرد. در [۱۱] یک روش پویا با آگاهی از کیفیت سرویس و خصوصیات تراکنشی آن برای ترکیب سرویس با استفاده از الگوریتم بهینه‌سازی کلونی مورچگان معرفی شده است که از مزایای آن می‌توان به پویا بودن گردش کار اشاره کرد. در [۱۲] مساله ترکیب سرویس در محیط‌های ابری، با آگاهی از کیفیت سرویس که در آن مراکز داده[۶] از نظر جغرافیایی توزیع شده هستند مورد بررسی قرار گرفته است. در این رویکرد از عملگر Skyline و نیز از الگوریتم ژنتیک برای حل مساله ترکیب سرویس استفاده شده است، به طوری که علاوه بر پارامترهای کیفیت سرویس، پارامترهای کیفیت شبکه نیز مورد بررسی قرار گرفته‌اند. این الگوریتم نسبت به الگوریتم ژنتیک ساده دارای پیچیدگی زمانی کمتری می‌باشد، ولی در آن ناسازگاری‌ها و استثنائات سرویس‌ها در طول زمان اجرای عمل ترکیب سرویس در نظر گرفته نشده است. رویکرد پیشنهادی در [۱۳] با ترکیب دو الگوریتم توانسته یک رویگرد کارآمدتر به وجود آورد. رویکرد موجود در [۳] با استفاده از خوشه‌بندی پروکلوس به نتایج بهتری دست یافته است. در [۱۴] نیز با استفاده از الگوریتم جهش قورباغه درهم ریخته و در آن مهمترین و اساسی‌ترین ساختار ترکیب را که در استانداردهای ترکیب سرویس به کار رفته است را به وسیله مؤلفه‌های انتزاعی نمایش می‌دهد. در جدول (۱) رویکردهای مذکور از حیث برخی معیارها با یکدیگر مقایسه شده‌اند.

با در نظر گرفتن رویکردهای مذکور در زمینه ترکیب سرویس‌های کاندید مناسب برای دستیابی به یک سرویس مرکب بهینه، مشهود است که اکثر الگوریتم‌ها یا از نظر زمانی دیر به جواب می‌رسند و یا با بهره-گیری از عملگری چون Skyline، خوشه‌بندی و یا عملگرهای دیگری به نتیجه دلخواه در مدت زمان مورد نظر دست می‌یابند. بنابراین در این تحقیق سعی شده است تا با استفاده از الگوریتمی، بدون بهره‌گیری از عملگرهای دیگر به جواب نهایی با کیفیتی بهتر دست یابیم.

جدول (۱): مقایسه روش‌های پیشین

Table (1): Comparison of previous methods

روش	محیط ترکیب	مزایا	معایب	پارامترهای QoS
رستمی و همکاران [۸]	وب	• پیدا کردن ترکیب ابری مناسب • جلوگیری از پیچیدگی پیدا کردن ترکیب ابری مناسب هنگام افزایش تعداد سرویس‌های وب	عدم در نظر گرفتن محیط‌های ابری	• زمان پاسخ • هزینه[۷]
گوبینگ زو و همکاران[۵]	چند ابری	• پیچیدگی زمانی پایین	• عدم دستیابی به ترکیب ابری بهینه • عدم در نظر گرفتن پارامترهای کیفیت سرویس	ندارد
گوبینگ زو و همکاران[۵]	چند ابری	• پیدا کردن ترکیب ابری مناسب	• پیچیدگی زمانی بالا • عدم در نظر گرفتن پارامترهای کیفیت سرویس	ندارد
گوبینگ زو و همکاران[۵]	چند ابری	• پیدا کردن ترکیب ابری مناسب	• عدم موفقیت در برخی مواقع • عدم در نظر گرفتن پارامترهای کیفیت سرویس	ندارد
کیانگ یو و همکاران[۶]	چند ابری	• پیدا کردن ترکیب ابری مناسب	• عدم موفقیت در برخی مواقع • عدم در نظر گرفتن پارامترهای کیفیت سرویس	ندارد
کیانگ یو و همکاران[۶]	چند ابری	• پیدا کردن ترکیب ابری مناسب و زمان اجرای مناسب	در نظر نگرفتن معیارهای کیفیت سرویس	ندارد

کوانوانگ وو و همکاران [۱۱]	تک ابری	• پویایی در زمان اجرا • دریافت گردش کار در زمان اجرا • دریافت محدودیت های کیفیت سرویس در زمان اجرا	کم بودن تعداد معیارهای کیفیت سرویس	• زمان پاسخ • دسترس پذیری • قابلیت اطمینان
اس ونگ و همکاران [۹]	تک ابری	زمان اجرایی پایین	• عدم پایداری سیستم ترکیب سرویس	ذکر نشده
دادن ونگ و همکاران [۱۲]	چند ابری	• در نظر گرفتن معیارهای کیفیت شبکه علاوه بر معیارهای کیفیت سرویس • زمان اجرایی پایین	عدم در نظر گیری سازگاری سرویس‌ها در زمان انجام فرآیند ترکیب	• زمان پاسخ • دسترس‌پذیری • هزینه • اعتبار
ژو-ژانگ لیو و همکاران [۱۰]	تک ابری	• همگرایی سریع • استفاده از مقادیر کیفیت سرویس به صورت پویا • قابلیت جستجو	پیچیدگی زمانی درجه دوم $O(n^2)$	• هزینه • زمان پاسخ • قابلیت اعتماد • دسترس‌پذیری • نرخ موفقیت
سیمون لودینگ و همکاران [۱۳]	تک ابری	ایجاد تعادل بین زمان اجرا و مقدار برازندگی	• پیچیدگی زمانی مرکب • همگرایی زودرس	• قابلیت اطمینان • دسترس‌پذیری • اعتبار • زمان اجرا • هزینه اجرا
امین جولا و همکاران [۳]	تک ابری	• مقیاس‌پذیر • کارآمد • کاهش زمان اجرا	فقط یکی از معیارهای ترکیب سرویس را در نظر گرفته است	ذکر نشده
علی یونس و همکاران [۱۴]	تک ابری	• نرخ موفقیت بهتر • زمان محاسبه کمتر	• ناسازگاری • عدم انتخاب سرویس بهینه	• زمان پاسخ • هزینه • دسترس‌پذیری • قابلیت اعتماد

۳- الگوریتم جستجوی موجودات زنده همبسته

همانند سایر الگوریتم‌های برپایه جمعیت، الگوریتم پیشنهادی نیز جهت یافتن جواب بهینه سراسری از یک جمعیت اولیه از جواب‌های تصادفی در محدوده جستجوی مساله استفاده می‌کند. SOS[8] با یک جمعیت اولیه که اکوسیستم نامیده می‌شود، شروع می‌شود. اکوسیستم اولیه شامل گروهی از موجودات زنده است که به صورت تصادفی در محدوده جستجوی مسئله تولید می‌شوند. هر موجود زنده نشان‌دهنده جوابی برای مسئله مورد نظر است که مطابق با تابع هدف مسئله، مقدار شایستگی مشخصی دارد.

تقریباً تمامی الگوریتم‌های فرامکاشفه‌ای به منظور بهبود جوابها، عملگرهایی را روی جوابها در هر نسل اعمال می‌کنند تا جواب‌های جدیدی را برای نسل بعدی تولید کنند. برای مثال الگوریتم ژنتیک استاندارد دارای دو عملگر تقاطع و جهش می‌باشد. الگوریتم جستجوی هارمونی برای بهبود جوابها و ایجاد هارمونی‌های جدید سه قانون در نظر گرفته است: در نظرگیری حافظه، تنظیم و انتخاب تصادفی. در SOS راه‌حل جدید با تقلید از تعامل بیولوژیکی بین دو موجود زنده در اکو سیستم ایجاد می‌شود. برای این منظور سه فاز که مشابه مدل‌های تعامل بیولوژیکی در دنیای واقعی هستند معرفی شده‌اند: فاز همزیستی، فاز همسفرگی، فاز زندگی انگلی [۱۵].

قانون کلی هر فاز به‌صورت مقابل تعریف می‌شود. در فاز همزیستی هر دو طرف از تعامل سود می‌برند. در فاز همسفرگی یک طرف از تعامل بهره می‌برد و هیچ اثری برای طرف دیگر ندارد. در فاز زندگی انگلی، یک طرف از تعامل بهره برده و طرف دیگر به طور اساسی آسیب می‌بیند. هر موجود زنده با دیگر موجودات به صورت تصادفی به‌واسطه تمامی فازها در تعامل می‌باشد. این روند تا رسیدن به شرط خاتمه تکرار می‌شود. طرح کلی الگوریتم را می‌توان به صورت زیر نشان داد:

- مقدار دهی اولیه
- تکرار
  - فاز همزیستی
  - فاز همسفرگی
  - فاز زندگی انگلی
- ارزیابی شرط خاتمه

با بررسی مقاله [۱۵] مشاهده می‌شود که الگوریتم مذکور برای توابع محک نتایج خوبی ارائه کرده است. همچنین با توجه به اینکه جواب‌های نامطلوب در فاز سوم این الگوریتم از چرخه خارج می‌شوند، انتظار می‌رود برای مسئله ترکیب سرویس نیز نتایج ارزنده‌ای به دست آید. زیرا که در فاز سوم با حذف سرویس‌های کاندید نامطلوب، فضای جستجو محدودتر شده و امکان انتخاب سرویس‌های کاندید مناسب به خوبی فراهم می‌شود. در نتیجه سرویس مرکب حاصل مطلوب‌تر خواهد بود، که این امکان در سایر الگوریتم‌های فرامکاشفه‌ای بدین صورت مشهود نمی‌باشد، به همین جهت در این تحقیق الگوریتم SOS انتخاب شده است.

۴- ترکیب سرویس‌های ابری با آگاهی از کیفیت سرویس با استفاده از الگوریتم جستجوی موجودات زنده همبسته

ساختار کلی الگوریتم جستجوی موجودات زنده همبسته در بخش قبل، مورد بررسی قرار گرفت و در این بخش به پیاده کردن این ساختار روی مسئله بیان شده خواهیم پرداخت.

۴-۱- کد گذاری اعضا و ایجاد جمعیت اولیه

یکی از مهمترین بخش‌های الگوریتم‌های تکاملی، کدگذاری اعضای جمعیت مسئله می‌باشد. در این الگوریتم هر عضو جمعیت اولیه نشانگر یک راه‌حل مساله است و یک راه‌حل در اینجا همان سرویس ابری مرکب می‌باشد که توسط یک آرایه به طول n (تعداد وظایف در جریان کار) معرفی می‌شود. مقدار ذخیره شده در اندیس i از آرایه (درایه i ام آرایه)، شناسه سرویس کاندید را نشان می‌دهد که وظیفه T_i را اجرا خواهد کرد. با فرض این که تعداد موجودات زنده در جمعیت اولیه P باشد، در این صورت جمعیت اولیه راه‌حل‌ها، یک ماتریس P×n خواهد بود که به صورت تصادفی در محدوده جستجوی مسئله ایجاد می‌شود.

۴-۲- ارزیابی

مهم‌ترین اهدافی که مسئله ترکیب سرویس‌های ابری آگاه از کیفیت سرویس با آن سر و کار دارد، برآورده کردن ترجیحات مورد نظر کاربر و همچنین بهینه‌سازی یک تابع شایستگی می‌باشد. تابع شایستگی بایستی مقادیر پارامترهای کیفیت سرویس را برای سرویس ابری مرکب ساخته شده بهینه کند. با در نظر گرفتن این که در مدل کیفیت پیشنهادی از مجموعه داده‌های QWS [۱۶] استفاده شده، بنابراین؛ شش پارامتر زمان پاسخ[۹]، قابلیت دسترسی[۱۰]، توان عملیاتی[۱۱]، قابلیت موفقیت[۱۲]، قابلیت اطمینان[۱۳] و تاخیر[۱۴] در نظر گرفته شده است، تابع شایستگی برای یک راه‌حل، به صورت رابطه (۲) تعریف می‌گردد که در آن که ضرایب w_1، w_2، w_3، w_4، w_5 و w_6 وزن‌های مثبتی هستند که درجه اهمیت هر یک از پارامترهای کیفیت سرویس را نشان داده و توسط کاربر مشخص می‌گردند. Q^- پارامترهای کیفیت سرویسی را نشان می‌دهد که کاهشی (زمان پاسخ و تاخیر) هستند و Q^+ پارامترهای افزایشی (قابلیت دسترسی، قابلیت موفقیت، قابلیت اطمینان و توان عملیاتی) را نشان می‌دهند.

$$\text{Fitness (sol)}=\sum_{i=1}^{2} w_i\times Q^- + \sum_{j=3}^{6} w_i\times \frac{1}{Q^+}$$

۴-۳- فاز همزیستی

در SOS، x_i در واقع نمونه‌ای است که عضو iام از اکو سیستم می‌باشد. در ادامه نمونه دیگر x_j از اکوسیستم انتخاب می‌شود تا با x_i در تعامل باشد. هر دو نمونه به منظور افزایش بقا در اکو سیستم در ارتباطی متقابل با یکدیگر قرار دارند، راه‌حل‌های پیشنهادی جدید برای x_i و x_j بر اساس روابط (۳)، (۴) و (۵) محاسبه می‌شود:

$$x_{inew} = x_i + \text{Rand}(0,1)\times(x_{best} - mv\times BF_1) \quad (۳)$$

$$x_{jnew} = x_j + \text{Rand}(0,1)\times(x_{best} - mv\times BF_2) \quad (۴)$$

$$mv = \frac{x_i + x_j}{2} \quad (۵)$$

که در این روابط، Rand(0,1) برداری از اعداد تصادفی است و نیز مقادیر BF_1 و BF_2 به صورت تصادفی ۱ یا ۲ در نظر گرفته می‌شود که در واقع این فاکتورها، سطح سودمندی هر نمونه را مشخص می‌کند.

۴-۴- فاز همسفرگی

مشابه با فاز قبلی، نمونه x_j جهت تعامل با نمونه x_i به‌صورت تصادفی از اکوسیستم انتخاب می‌شود. در این فاز، نمونه x_i سعی در بهره‌مندی از تعامل را دارد، در حالیکه نمونه x_j نه سودی می‌برد و نه ضرری متحمل می‌شود. راه‌حل پیشنهادی جدید بر اساس x_i با استفاده از رابطه (۶) محاسبه می‌شود. بر طبق قانون نمونه x_i در صورتی به‌روز رسانی خواهد شد که مقدار شایستگی جدید به‌دست آمده از آن، از مقدار قبلی بهتر باشد.

$$x_{inew} = x_i + \text{Rand}(-1,1)\times(x_{best} - x_j) \quad (۶)$$

۴-۵- فاز زندگی انگلی

در این فاز برداری به نام بردار انگل[۱۵] در فضای جستجوی مسئله با تکرار نمونه x_i بر اساس مقادیر فضای جستجو تغییر داده می‌شود. نمونه x_j به صورت تصادفی از اکوسیستم انتخاب شده و به عنوان میزبان برای بردار انگل استفاده می‌شود. بردار انگل سعی بر این دارد که در اکوسیستم خود را جایگزین x_j کند. برای هر دو نمونه مقادیر شایستگی اندازه گرفته می‌شود. اگر بردار انگل مقدار شایستگی بهتری داشته باشد در این صورت نمونه x_j را از بین برده و جای آن را در اکوسیستم می‌گیرد، اما اگر مقدار شایستگی x_j بهتر باشد در این صورت x_j از انگل در امان مانده و بردار انگل از اکوسیستم حذف خواهد شد.

Input: Composition request as a workflow (DAG) and user preferences
output: near optimal composite cloud service

//initialization
1.ecosystem initialization (initial population of organisms)
each organism is an n-dimensional array of widows for a composition problem with n-Tasks

2. Identifying the best solution x_{best}
3. Mutualism phase by using equations 3,4 and 5
4. Commensalism phase by using equation ۶
5. Parasitism phase
6. Go to step 2 if the current Xi is not the last member of the ecosystem; otherwise proceed to next step
7. Stop if the termination criteria is reached; otherwise return to step 2 and start the next iteration

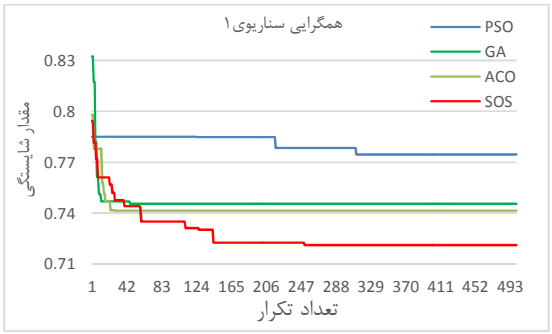
شکل (۱): شبه کد روش پیشنهادی برای حل مسئله ترکیب سرویس
Fig.(1): Pseudo code of symbiotic organism search algorithm for service composition

۴-۶- شرط خاتمه

پس از طی شدن سه فاز فوق، شرط خاتمه بررسی می‌شود و در صورتی که برقرار نشده باشد، اجرای برنامه دوباره به فاز اول برمی‌گردد. در این مقاله شرط خاتمه حداکثر تعداد تکرار در نظر گرفته شده است.

۵- ارزیابی و نتایج آزمایشات

برای انجام تمامی آزمایشات از یک لپتاپ Samsung با پردازنده Core i5 ,2.5 GHz و ۶ گیگا بایت حافظه اصلی استفاده شده است. همچنین از مجموعه داده QWS به عنوان اطلاعات کیفیت سرویس مربوط به سرویس‌های ابری کاندید استفاده شده است. تمامی آزمایشات و شبیه-سازی‌ها در محیط نرم‌افزار Matlab 2015b انجام شده است.

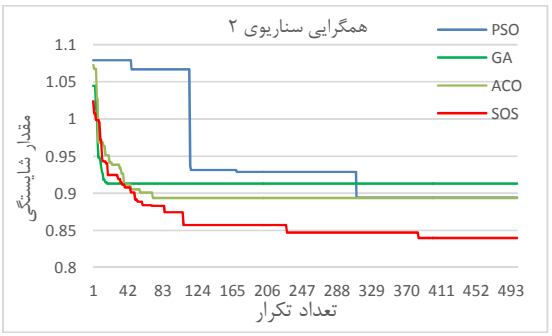


شکل (۲): همگرایی سناریوی ۱
Fig.(2): Convergence of scenario 1

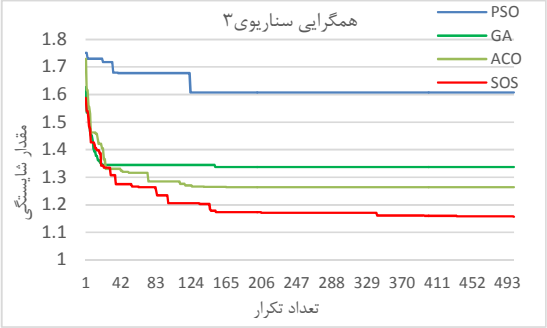
با در نظر گرفتن این نکته که روش پیشنهادی از یک الگوریتم فرا-مکاشفه‌ای استفاده کرده است، بنابراین نتایج حاصل از نظر همگرایی و همچنین پایداری ارزیابی شده‌اند که در ادامه ارائه خواهند شد. همچنین نتایج حاصل از روش پیشنهادی با نتایج سه الگوریتم ژنتیک و کلونی مورچگان و ازدحام ذرات مقایسه شده است.

۵-۱- آزمایشات همگرایی

جهت انجام آزمایشات همگرایی، برای تمامی الگوریتم‌ها تعداد جمعیت اولیه ۱۰۰ و تعداد نسل ۵۰۰ در نظر گرفته شده است. آزمایشات همگرایی برای ۴ سناریو با تعداد وظایف ۱۰، ۵۰، ۱۰۰ و ۱۵۰ به ترتیب با تعداد حداکثر سرویس کاندید ۱۰۰، ۵۰۰، ۱۰۰۰ و ۱۵۰۰ انجام شده است. الگوریتم پیشنهادی و همچنین الگوریتم‌های ژنتیک [۱۷] و ازدحام ذرات [۱۸] و الگوریتم کلونی مورچگان [۱۹] برای ۴ درخواست ترکیب با داده‌های یکسان، اجرا شده و شکل‌های (۲)، (۳)، (۴) ، (۵) به ترتیب نحوه همگرایی به جواب نهایی را نشان می‌دهند. در نمودارهای مذکور محور افقی مرتبه تکرار الگوریتم و محور عمودی نیز بهترین شایستگی هر تکرار را نشان می‌دهد. در شکل (۶) نیز نتایج نهایی هر چهار سناریو در قالب یک نمودار ستونی آورده شده است که با توجه به آن می‌توان دریافت که روش پیشنهادی با افزایش تعداد وظایف همچنان برتری خود را نسبت به سایر الگوریتم‌های مورد آزمایش حفظ کرده است. شکل (۱) نیز نشان‌دهنده شبه کد روش پیشنهادی برای مسئله ترکیب سرویس‌های ابری آگاه از کیفیت سرویس می‌باشد.

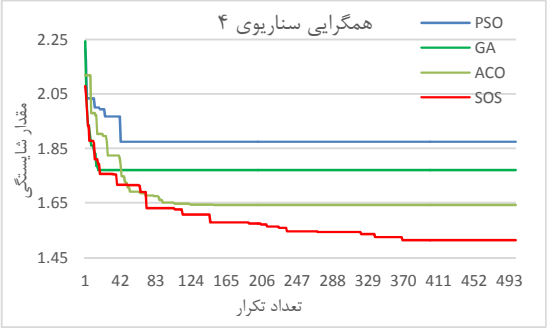


شکل (۳): همگرایی سناریوی ۲
Fig.(3): Convergence of scenario 2



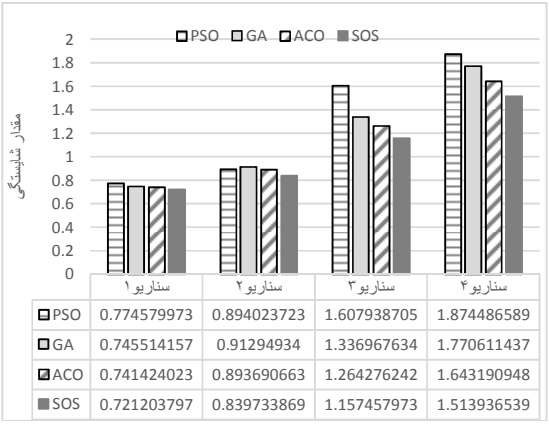
شکل (۴): همگرایی سناریوی ۳

Fig.(4): Convergence of scenario3



شکل (۵): همگرایی سناریوی ۴

Fig. (5): Convergence of scenario 4



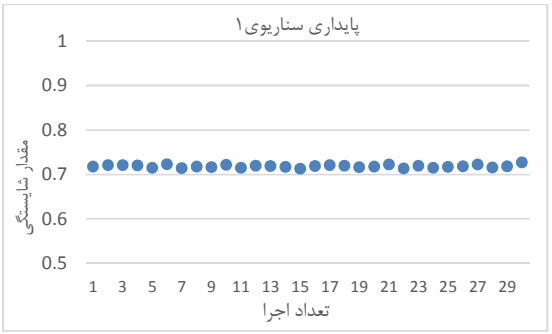
شکل (۶): مقایسه ستونی سناریوها

Fig.(6): Scenarios comparison

بررسی نتایج حاصل از این آزمایشات نشان می‌دهد که روش پیشنهادی برای ترکیب سرویس‌های ابری همگرایی مناسبی داشته و سرویس مرکب ابری نزدیک به بهینه را پیدا می‌کند و نیز با بالا رفتن تعداد وظایف همچنان کیفیت بالای جواب در آن حفظ می‌شود. روش پیشنهادی نسبت به روش‌های ازدحام ذرات، ژنتیک و کلونی مورچگان به ترتیب دارای ۱۹٪، ۱۴٪ و ۰.۷٪ بهبود بوده است که می‌توان گفت به طور متوسط ۱۳٪ نسبت به روش‌های پیشین بهبود داشته است.

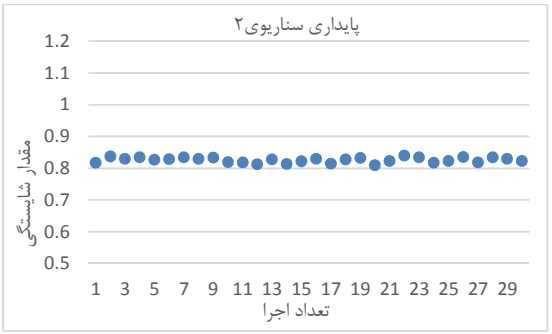
۵-۲- آزمایشات پایداری

الگوریتم‌های فرامکاشفه‌ای ماهیتی غیرقطعی و تصادفی دارند، پس بنابراین لازم است پایداری این الگوریتم‌ها مورد بررسی قرار گیرد. منظور از پایداری یک الگوریتم این است که آیا الگوریتم به ازای اجراهای مختلف، پاسخ‌های یکسان و نزدیکی تولید می‌کند یا نه. برای بررسی پایداری روش پیشنهادی برای چهار درخواست مذکور، الگوریتم ۳۰ مرتبه اجرا شده و مقادیر شایستگی سرویس مرکب ابری ایجاد شده در هر اجرا در شکل‌های (۷)، (۸)، (۹) و (۱۰) نشان داده شده است. محور افقی در نمودارها مرتبه اجرای الگوریتم و محور عمودی مقدار شایستگی سرویس مرکب ابری ایجاد شده در هر مرتبه را نشان می‌دهد.



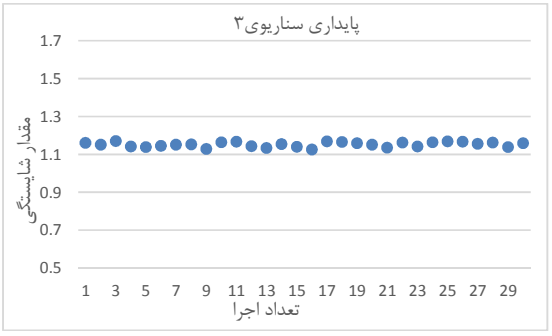
شکل (۷): پایداری سناریوی ۱

Fig.(7): Stability of scenario 1



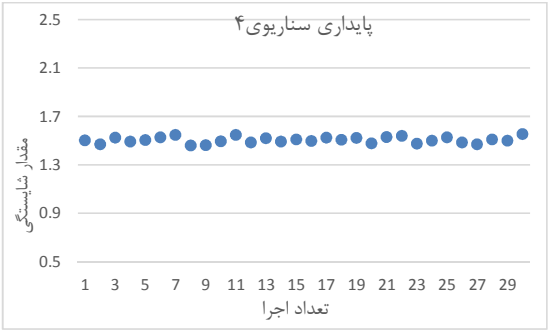
شکل(۸): پایداری سناریوی ۲

Fig.(8): Stability of scenario 2



شکل (۹): پایداری سناریوی ۳

Fig.(9): Stability of scenario 3



شکل(۱۰): پایداری سناریوی ۴
Fig. (10): Stability of scenario 4

بررسی نمودارهای مربوط به پایداری نشان می‌دهد که روش پیشنهادی برای درخواست‌های ترکیب با تعداد وظایف کم، با توجه به نوسان بسیار پایین مقدار شایستگی سرویس مرکب ابری بهینه در اجراهای مختلف، پایدار چشمگیری دارد و برای درخواست‌های با تعداد وظایف بیشتر نیز میزان نوسان پایین است، پس در حالت کلی روش پیشنهادی برای مسئله ترکیب سرویس دارای پایداری نسبتاً بالایی است و همچنین مقدار انحراف معیار برای ۴ درخواست مذکور به ترتیب برابر است با ۰/۰۰۳۳، ۰/۰۰۸۲، ۰/۰۱۳و ۰/۰۲۶که نشان‌دهنده پایداری بالای روش پیشنهادی می‌باشد.

۶- نتیجه‌گیری

ترکیب سرویس‌های ابری یک تکنولوژی مهم برای ایجاد سرویس‌های ارزش افزوده می‌باشد [۳]. ترکیب سرویس آگاه از کیفیت سرویس با محدودیت‌های عمومی QoS یک موضوع بسیار مهم در محاسبات ابری [۲۰] و محاسبات سرویس [۲۱], [۲۲] می‌باشد. طراحی یک روش عالی برای حل این مساله با قابلیت یافتن جواب بهینه در مدت زمان مناسب چالشی است که در این مقاله به آن پرداخته شده و با بهره‌گیری از الگوریتم جستجوی موجودات زنده همبسته سعی شده که این مسئله حل شود. نتایج آزمایشات و شبیه‌سازی‌ها و مقایسه آنها با رویکردهای پیشین نشان می‌دهد که روش پیشنهادی از لحاظ کیفیت جواب‌های به دست آمده و نیز مقیاس‌پذیری برتری قابل ملاحظه داشته و نیز از پایداری نسبتاً خوبی برخوردار است، به نظر می‌رسد فاز زندگی انگلی با حذف جواب‌های نامطلوب باعث می‌شود که نسل بعدی با جواب‌های مناسبی به کار ادامه دهد و این باعث شده است که روش پیشنهادی به نتایج بهتری نسبت به سایر روش‌ها دست یابد و به طور متوسط دارای ۱۳٪ بهبود نسبت به سایر روش‌های مورد مقایسه بوده است، ولی با این حال از نظر زمان اجرا کارایی پایینی نسبت به سایر رویکردها دارد، بنابراین برای کارهایی که نیاز به کیفیت بالا داشته و بلادرنگ و برخط نباشند می‌تواند مورد استفاده قرار گیرد.

پی‌نوشت

1. Loosely coupled
2. Task
3. Candidate Cloud Services
4. Workflow
5. Directed Acyclic Graph (DAG)
6. Data centers
7. Cost
8. Symbiotic Organisms Search
9. Response Time
10. Availability
11. Throughput
12. Successability
13. Reliability
14. Lateness
15. Parasite-Vector

References

[1] Wikipedia, “Cloud computing - Wikipedia, The Free Encyclopedia”, 2016.
[2] A. Jula, H. Nilsaz, E. Sundararajan, Z. Othman, “A new dataset and benchmark for cloud computing service composition”, Proceeding of the IEEE/ISMS, pp. 83–86, Langkawi, Malaysia, Jan. 2014.
[3] A. Jula, Z. Othman, E. Sundararajan, “Imperialist competitive algorithm with PROCLUS classifier for service time optimization in cloud computing service composition”, Expert Systems with Applications, Vol. 42, No. 1, pp. 135–145, Jan. 2015.
[4] A. Jula, E. Sundararajan, Z. Othman, “Cloud computing service composition: A systematic literature review”, Expert Systems with Applications, Vol. 41, No. 8, pp. 3809–3824, Jun. 2014.
[5] G. Zou, Y. Chen, Y. Yang, R. Huang, Y. Xu, “AI planning and combinatorial optimization for web service composition in cloud computing”, Proccedding of the International Conference on Cloud Computing and Virtualization, pp. 1–8, 2010.
[6] Q. Yu, L. Chen, B. Li, “Ant colony optimization applied to web service compositions in cloud computing”, Computers & Electrical Engineering, Vol. 41, pp. 18–27, 2015.
[7] H. Kurdi, A. Al-Anazi, C. Campbell, A. Al Faries, “A combinatorial optimization algorithm for multiple cloud service composition”, Computers & Electrical Engineering, Vol. 42, pp. 107–113, 2015.
[8] N.H. Rostami, E. Kheirkhah, M. Jalali, “An optimized semantic web service composition method based on clustering and ant colony algorithm”, ArXiv Prepr. ArXiv1402.2271, 2014.
[9] S. Wang, Q. Sun, H. Zou, and F. Yang, “Particle swarm optimization with skyline operator for fast cloud-based web service composition”, Mobile Networks and Applications, Vol. 18, No. 1, pp. 116–121, Feb. 2013.
[10] Z.-Z. Liu, D.-H. Chu, C. Song, X. Xue, B.-Y. Lu, “Social learning optimization (SLO) algorithm paradigm and its application in QoS-aware cloud service composition”, Information Sciences, Vol. 326, pp. 315–333, 2016.

[11] Q. Wu, Q. Zhu, "Transactional and QoS-aware dynamic service composition based on ant colony optimization", Future Generation Computer Systems, Vol. 29, No. 5, pp. 1112–1119, 2013.
[12] D. Wang, Y. Yang, Z. Mi, "A genetic-based approach to web service composition in geo-distributed cloud environment", Computers & Electrical Engineering, Vol. 43, pp. 129-141, April 2015.
[13] S.A. Ludwig, "Applying particle swarm optimization to quality-of-service-driven web service composition", Proceeding of the IEEE/AINA, pp. 613–620, Fukuoka, Japan, March 2012.
[14] A. Younes, M. Essaaidi, A. El Moussaoui, "SFL algorithm for QoS-based cloud service composition", International Journal Computer Appllications, Vol. 97, No. 17, pp. 42–49, 2014.
[15] M.-Y. Cheng, D. Prayogo, "Symbiotic organisms search: A new metaheuristic optimization algorithm", Computers and Structures, Vol. 139, pp. 98–112, July 2014.
[16] E. Al-Masri, Q.H. Mahmoud, "Discovering the best web service", Proceedings of the IEEE/ICSMC, pp. 4250-4255, San Antonio, TX, USA , 2009.
[17] G. Canfora, M. Di Penta, R. Esposito, M.L. Villani, "An approach for QoS-aware service composition based on genetic algorithms", Proceedings of the GECCO, pp. 1069–1075, 2005.
[18] G. Spezzano, "Using service clustering and self-adaptive MOPSO-CD for QoS-aware cloud service selection", Procedia Computer Science, Vol. 83, pp. 512–519, 2016.
[19] L. Zaki, A. Pourhajikazem, S. Lotfi, "Providing an algorithm for QoS-aware grid service composition using ant colony algorithm", in First National Conference on New Approaches in Computer Engineering and Data Recovery, Rudsar, Islamic Azad University, Ruddersar and Amlash, 1392.
[20] Y. Zhu, R. Y. Shtykh, Q. Jin, "A human-centric framework for context-aware flowable services in cloud computing environments", Information Sciences, Vol. 257, pp. 231–247, Feb. 2014.
[21] J. Yu, Q.Z. Sheng, Y. Han, "Introduction to special issue on cloud and service computing", Service Oriented Computing and Applications, Vol. 7, No. 2, pp. 75–76, June 2013.
[22] M.P. Papazoglou, "Service-oriented computing: Concepts, characteristics and directions", Proceeding of the IEEE/WISE, pp. 3–12, Rome, Italy, Dec. 2003.