ISSN: 2745-3673
http://www.Rcsj.ir

مجله علمی پژوهش در علوم رایانه (سال پنجم)
شماره ۲۰ / زمستان ۱۳۹۹ / ص ۳۲- ۱۶

استفاده از الگوریتم رقابت استعماری چند هدفه مبتنی بر نظریه بازی ها برای بهبود کیفیت سرویس در پایگاه داده بلادرنگ

مریم مجاوری[۱]، حنانه صالحی فر[۲]، منصور اسماعیل زاده[۳]

[۱] دانشجوی کارشناسی ارشد، کامپیوتر، نرم افزار، موسسه آموزش عالی بصیر.
[۲] مهندسی کامپیوتر، نرم افزر، دانشگاه آزاد اسلامی.
[۳] کارشناسی ارشد، هوش مصنوعی، شرکت خاوران فناوری هوشمند.

نام نویسنده مسئول:
مریم مجاوری

تاریخ دریافت: ۱۳۹۹/۸/۴
تاریخ پذیرش: ۱۳۹۹/۱۰/۳

چکیده

امنیت و کیفیت سرویس **(QOS)**[1] هر دو منبع محاسباتی سیستم **IT** را مصرف کرده و در نتیجه ممکن است به وضوح سرویس های نرم افزاری را تحت تاثیر قرار دهند. در مواردی که منبع محاسباتی با محدودیت مواجه است، مدل سازی تاثیر متقابل بین امنیت شبکه و **QOS**، که می تواند به صورت همزمان برای ارائه یک عملکرد بهتر تحت منبع محاسباتی موجود بهینه شود، اهمیت پیدا می کند. در این مقاله، یک مدل ارزیابی بر این اساس برای توصیف تاثیر متقابل امنیت شبکه و **QOS** ارائه می شود و سپس الگوریتم رقابت استعماری چند هدفه مبتنی بر نظریه بازی ها برای بهینه سازی مدل انجام می شود. نتایج شبیه سازی، به اینکه الگوریتم ما قادر است مجموعه جدیدی از سیاست های امنیتی پارتو بهینه[۲] در بارهای کاری مختلف شبکه که می تواند به عنوان تنظیمات امنیتی متفاوت در اختیار کاربران بالقوه قرار بگیرد، اعتبار می بخشد. این سیاست های امنیتی پارتو بهینه حاصل، نه تنها نیاز امنیتی کاربر را تامین می کنند، بلکه **QOS** بهینه را تحت منبع محاسباتی موجود فراهم می آورند.

واژگان کلیدی: امنیت و کیفیت سرویس - الگوریتم رقابت استعماری – نظریه بازی ها.

[1] Quality of Service
[2] Pareto-optimal security policies

مجلـه‌ی علمی پژوهش در علوم رایانه ، شماره ۲۰، زمستان ۱۳۹۹، ص ۳۲- ۱۶

ISSN: 2745-3673
http://www.Rcsj.ir

مقدمه

سیستم های پایگاه داده به طور گسترده در سیستم های کامپیوتری امروزی که برای ذخیره سازی و دسترسی به داده در سرویس های نرم افزاری متعدد به کار گرفته می شوند، مورد استفاده قرار می گیرند. [۴]

سیستم های پایگاه داده مرسوم به گونه ای طراحی شده اند که داده دائماً پایدار را پردازش کنند و یکپارچگی و ثبات داده را حفظ نمایند. اهداف عملکردی آن ها بر توان خروجی بالا و هزینه پایین سیستم تمرکز دارد. در حالی که یک پایگاه داده بلادرنگ برای استفاده از پردازش بلادرنگ طراحی شده است، به طوری که قادر است با بارهای کاری که حالت آن ها دائماً تغییر می کند، کار کند. [۵]

با کاربرد گسترده سیستم های پایگاه داده، از آن جایی که داده ذخیره شده در پایگاه های داده همواره دربردارنده اطلاعات حساسی مانند حریم خصوصی افراد، اطلاعات بانکی و اسرار تجاری هستند این سیستم ها با تهدیدهای درونی و بیرونی بیشتر و بیشتری مواجهند [۶]. در پایگاه داده سرویس های بلادرنگ بیشتر و بیشتری مورد نیاز است که بر کیفیت سرویس (QOS) تأثیر زیادی خواهند داشت. سیستم پایگاه داده بلادرنگ به پلت فرم داده برای اطلاعات سازمانی تبدیل شده است که برای پردازش داده تراکنش بلادرنگ در سیستم تجارت الکترونیک سازمان استفاده می شود تا عملکرد سیستم را برای سیستم شبیه سازی آزمایشگاه شبیه سازی و مانیتور کرده و یا داده تاریخی را برای پلت فرم به اشتراک گذاری داده ذخیره کند و غیره [۷] . به دلیل وجود اثر متقابل بین امنیت شبکه و QOS، علاقه فزاینده ای برای کشف رابطه واقعی آن ها در سیستم های پایگاه داده وجود دارد. برای مثال، با استفاده روز افزون از سرویس های نرم افزاری شبکه بلادرنگ که حاوی اطلاعات حساس می باشند، لازم است امنیت کافی سرویس برای حفظ امنیت و QOS بالای کاربران به منظور برآوردن الزامات بلادرنگ بودن فراهم شود.

در بیشتر موارد، امنیت و QOS به طور مستقل مورد بررسی قرار می گیرند. طی دهه های گذشته، در زمینه بهبود QOS، مطالعات پژوهشی بسیاری روی QOS پایگاه داده بلادرنگ به عمل آمده است [۸]. از آن جایی که مکانیسم های تشخیص ناهنجاری برای حفظ پایگاه داده از تهدیدهای بالقوه مانند تزریق SQL و حملات جعل هویت مورد نیاز هستند [۹]، مکانیسم های امنیت مرسوم مانند مکانیسم های کنترل دسترسی [۱۰] و مکانیسم های اجرای سیاست [۱۱] برای سیستم پایگاه داده به اندازه کافی امن نیستند. بنابراین، سیستم های تشخیص و پیشگیری نفوذ (IDPSs) برای کامل کردن مدل امنیتی مرسوم در سیستم پایگاه داده مورد استفاده قرار می گیرد. IDPS ها اخیراً به منظور حفظ سیستم پایگاه داده از نفوذهای مخرب گسترش یافته اند [۱۲].

اما سیستم پایگاه داده سرویس امنیتی و QOS را به طور همزمان نیاز دارد. زمانی که QOS بالا مورد نیاز است، منابع در دسترس کمتری در اختیار سرویس امنیتی شبکه قرار می گیرد. از طرف دیگر، در بعضی کاربردها سطح بالایی از سرویس های امنیتی شبکه تقاضا می شود که این سطوح بالا منابع بیشتری را مصرف می کنند و می توانند به طور قابل توجهی منبع اختصاص یافته به QOS را کاهش دهند [۱۳]. در نتیجه، محققان اقدام به مطالعه ارتباط بین امنیت و QOS در سال های اخیر نموده اند [۱۴]. یک مدل بهینه سازی تک هدفه بر اساس پلت فرم پایگاه داده در کار [۱۵] طراحی شده است که از سیستم تشخیص نفوذ برای تضمین امنیت سیستم استفاده می کند.

بسته به طبیعت نرم افزارها، الزامات امنیتی ممکن است برای کاربر مشابه متفاوت باشد. برای مثال، تجارت آنلاین همواره نیازمند الزامات بالای امنیتی است در حالی که تماشای ویدئو در اینترنت نیاز به پیکره بندی امنیتی پایینی دارد. علاوه بر این، الزامات امنیتی متفاوت حتی برای یک کاربرد می تواند از سوی کاربران متفاوت درخواست شود. برای مثال، وقتی کاربران به یک پایگاه داده دسترسی دارند، برای کاربری که حق امتیاز روت دارد امنیت بالایی درخواست می شود در حالی که امنیت پایین برای کاربری ارائه می شود که حق امتیاز بازدید کننده دارد. به دلیل نیاز به امنیت متفاوت، سیستم پایگاه داده باید مجموعه ای از راه حل های امنیتی بهینه فراهم کند که بتواند درخواست برای سطوح امنیتی مختلف را برآورده کرده و QOS بالا را حفظ کند. بدون هرگونه اطلاعات بیشتر، این راه حل های بهینه که الزامات امنیتی و QOS را یکپارچه می کنند، راه حل های پارتو بهینه هستند [۱۶]، که نشان می دهد هیچ جواب دیگری بهتر از آن ها چه در QOS و چه در امنیت وجود ندارد. با این روش،

ISSN: 2745-3673
http://www.Rcsj.ir

سیستم پایگاه داده باید تعدادی از جواب های پارتو بهینه را به عنوان نماینده پیدا کند که به عنوان جواب های در دسترس برای الزامات مختلف به کار گرفته شود. به این ترتیب، کاربران می توانند یک جواب پارتو بهینه را برای پیکربندی مکانیسم های امنیتی بر اساس تنظیمات خود انتخاب کنند. با این وجود، حتی با تنظیمات بهینه امنیت و QOS در پایگاه داده، پایش بلادرنگ پارامترهای لازم امنیت و QOS فشار زیادی برای سرپرست سیستم محسوب می شود که کارهای بسیار بیشتری از پایش پارامترهای عملکردی و اجرای روش بهینه به صورت دستی در یک محیط پویا دارد [۱۵]. برای حل مشکل ذکر شده، سیستم اتونومیک یک روش امید بخش می باشد، چرا که توانایی خود مدیریتی از طریق پیکره بندی خود، بهینه سازی خود، محافظت از خود و ترمیم خود با حلقه های بازخوردی را دارد [۱۷]. با الهام از سیستم کامپیوتری اتونومیک [۱۸] که توسط مدل های شبکه های صف [۱۹] طراحی شده است، این روش هم چنین قادر است یک پیکره بندی اتونومیک هم برای امنیت و هم QOS در سیستم پایگاه داده ارائه دهد.

لازم به ذکر است که اگرچه شاخص های کلیدی QOS شامل تأخیر (زمان پاسخ)، jitter و نرخ اتلاف بسته هستند، به منظور ساده سازی مدل چند هدفه، این مقاله به طور عمده ارتباط بین زمان پاسخ و امنیت شبکه را در نظر می گیرد. علاوه بر این، تأخیرات بیشتر را نیز می توان با مصارف به راحتی برطرف کرد و به طور قابل توجهی تجربه کاربر را زمانی که از پایانه هایی استفاده می کنند که محدودیت منابع دارند تحت تأثیر قرار داد، مانند پایانه های شبکه و وسایل دستی (موبایل و ..). بعد از آن، الگوریتم رقابت استعماری با رویکرد نظریه بازیها برای رسیدن به مجموعه های پارتو بهینه مدل ارائه می شود، چرا که این الگوریتم کارایی خود را در حل بسیاری از مسائل مهندسی نشان داده است [۲] (این جواب های پارتو-بهینه که توسط کنترلر اتونومیک با این الگوریتم به دست آمده است می تواند امنیت سرویس و تأخیر آن را در یک بازه قابل قبول تضمین کند. برای شفاف کردن نوآوری های این مقاله، آن ها را در لیست زیر گردآوری کرده ایم:

۱) روش پیشنهادی ما ازیک مدل بهینه سازی چند هدفه استفاده می کند تا به طور همزمان امنیت شبکه و QOS را بهینه کند، که متفاوت از مدل تک هدفه برای خلاصه سازی امینت و QOS شبکه با استفاده از یک روش وزنی خطی می باشد. این اولین تلاشی است که برای ایجاد یک مدل بهینه سازی چند هدفه برای بهینه کردن امنیت شبکه و QOS پایگاه داده انجام می شود. جواب های پارتو بهینه موجود بیشتر که امنیت شبکه و QOS را در نظر می گیرند، ارائه می شوند و در نتیجه کاربران میتوانند تنظیمات خود را هم روی امنیت و هم QOS انتخاب کنند.

۲) الگوریتم رقابت استعماری با رویکرد نظریه بازی ها برای بهینه کردن مدل چند هدفه پیشنهاد شده و رسیدن به مجموعه پارتو بهینه برای امنیت شبکه و QOS اصلاح شده است که بسیار بهتر و اثربخش تر از الگوریتم تپه نوردی در کار [۱۵] عمل می کند. با این روش، سیستم پایگاه داده قادر است به سرعت به درخواست های بیشمار کاربران مختلف پاسخ دهد.

در پایان، عملکرد الگوریتم رقابت استعماری با رویکرد نظریه بازی ها برای بهینه کردن مدل چند هدفه با استفاده از مدل چند هدفه پیشنهادی برای بهینه کردن امنیت شبکه و QOS ارزیابی می شود. نتایج شبیه سازی نشان می دهد که الگوریتم رقابت استعماری با رویکرد نظریه بازی ها بهتر از NSGA-II با عملگرهای متقاطع تک نقطه ای و دونقطه ای عمل کرده و مجموعه بهینه حاصل قادر است مدل پارتو بهینه را در بارهای کاری مختلف به کار ببرد.

مدل ارزیابی امنیت و QOS

محیط نرم افزار

IDPS ها به کاربران و رفتارهای قانونی اجازه دسترسی به سیستم را می دهد. در حال حاضر، دو روش عمده شناسایی در IDPS ها وجود دارد: IDPS آماری مبتنی بر ناهنجاری[3] و IDPS مبتنی بر امضا[4]. روش IDPS آماری مبتنی بر ناهنجاری، فعالیت نرمال شبکه را تعیین کرده و هنگامی که ترافیک غیرعادی شناسایی شود به کاربر هشدار می دهد، در حالی که IDPS

[3] Statistical anomaly-based IDPS
[4] Signature-based IDPS

ISSN: 2745-3673
http://www.Rcsj.ir

مبتنی بر امضا بسته های موجود در شبکه را با الگوهای حمله از پیش شکل گرفته و تعیین شده که به عنوان امضا شناخته می شوند، پایش می کند.

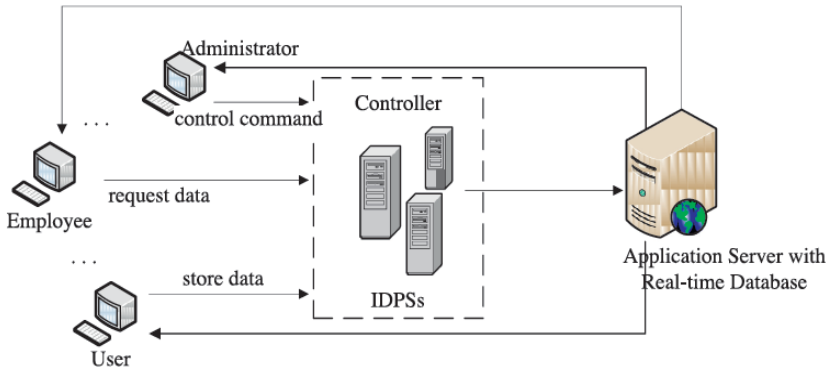
در مدل ما، ترکیبی از IDPS ها برای حفاظت از امنیت سیستم نرم افزاری شبکه استفاده می شود که در آن ادغامی از مکانیسم های مختلف و متنوع، استحکام امنیتی بالاتری فراهم می کند. در تنظیم معمول، IDPS ها بین سرویس گیرنده وب[5] و سرور نرم افزار[6] قرار دارند، که در آن جا IDPS ها درخواست های کاربران را قبل از اینکه اجازه دسترسی به سرور بیابند، ارزیابی می کنند. به دلیل قواعد مختلف IDPS ها، زمان پاسخ IDPS های مختلف برای درخواست مشابه، متفاوت است. بنابراین، زمان پاسخ کلی سیستم بستگی به حجم بار کاری، ترکیب مکانیسم های امنیتی به کار رفته و سربار مرتبط با هر مکانیسم دارد.

طبقه بندی داده[7] برای کار با حجم بالای داده در محیط تولید استفاده می شود. به دلیل حجم بالای کاربران، ایجاد یک پروفایل دسترسی برای هر کاربر غیر عملی است. بنابراین سیستم، کاربران را بر اساس نقش ها طبقه بندی کرده، و نقش ها و رفتارهای مورد انتظار آن ها را تعریف می کند. قاعده طبقه بندی نقش می تواند توسط سرپرست سیستم تنظیم شود یا زمانی که وضعیت سیستم تغییر می کند تغییر داده شود. هم چنین لازم است حملات نیز طبقه بندی شوند. این طبقه بندی بر اساس این سناریو انجام می شود که حملات وجود دارند و کاربر ممکن است در معرض این حملات قرار گیرد. سپس کارایی مکانیسم های امنیتی از IDPS ها برای هر طبقه بندی حمله و سربار مرتبط با هر مکانیسم تعریف می شود. در محیط ما، تعدادی از IDPS های به کار رفته و نرخ شناسایی آن ها برای نشان دادن کارایی مکانیسم ها اتخاذ می شوند.

شکل (۱) یک محیط نرم افزاری سازمانی را برای مدل ارزیابی امنیت و QOS نشان می دهد که در آن سرور نرم افزار، سرورها را در یک سیستم پایگاه داده بلادرنگ نشان می دهد. IDPS ها در لبه خارجی پایگاه داده مستقر شده اند که تمام داده ورودی از خارج سیستم پایگاه داده را بررسی می کند. درخواست های ورودی سیستم از پایانه ها در قالب نقش های مختلف طبقه بندی می شوند، به این ترتیب که فرمان های کنترل سیستم توسط سرپرستان، داده تحلیل توسط کارمندان و داده ذخیره سازی توسط کاربران ارسال می شود. سرپرستان سیستم، فرمان های کنترل را به سیستم که برای حفظ ثبات خود هم نیاز به امنیت بالا و تأخیر پایین دارد، ارسال می کنند. کارمندان سازمان از پایگاه داده بلادرنگ به داده تاریخی دسترسی دارند تا سیستم را شبیه سازی و بهینه کنند، که در مقایسه با بهره وری زمان به امنیت توجه بیشتری می کند. زمانی که کاربران فایل هایی را بارگذاری می کنند که می توانند با هر کسی به اشتراک گذاشته شوند، الزامات امنیت و تأخیر می توانند پایین باشند. کنترلر بین سرویس گیرنده وب و سرور نرم افزار شبکه قرار دارد که برای اینکه درخواست ها را ارزیابی کرده و به IDPS ها ارسال کند، به کار گرفته می شود. کنترلر در یک بازه زمانی خاص اجرا می شود تا نرخ ورود نقش های مختلف را به دست آورد، ترکیب های ممکن از IDPS ها را شناسایی کند، استحکام امنیتی و زمان پاسخ را برای پیکره بندی حاضر با استفاده از مدل ارزیابی پیشنهاد شده محاسبه کند، و جواب بهینه را که بستگی به تنظیمات کاربر دارد پیشنهاد کند. بعد از اجرای کنترلر، سیاست امنیتی بسته به پروفایل انتخاب شده شکل خواهد گرفت.

[5] Web client
[6] Application server
[7] Data classification

ISSN: 2745-3673
http://www.Rcsj.ir



شکل ۱- محیط نرم افزار سازمانی با پایگاه داده بلادرنگ

مفهوم و تعریف

IDPS ها به عنوان مکانیسم های امنیتی در این مقاله به کار می روند. تعداد مکانیسم های امنیتی را با N نشان می دهیم. تعداد نقش ها و تعداد طبقه بندی های حمله نیز به ترتیب با R و A نمایش داده می شوند. $a_{r,j}$ را احتمال حمله j به نقش r، $d_{i,j}$ نرخ شناسایی حمله j توسط مکانیسم امنیتی i، و o_i سربار مکانیسم i در نظر می گیریم.

سیاست کنترلر ρ_r، تخصیص مکانیسم های امنیتی برای نقش r را نشان می دهدکه به صورت ($\varepsilon_{r,1}$, $\varepsilon_{r,2}$, … , $\varepsilon_{r,N}$) = ρ_r تعریف می شود. لازم به ذکر است که اگر $\varepsilon_{r,i}$ معادل ۱ باشد به این معنی است که مکانیسم امنیتی i برای نقش r به کار رفته است، در غیر این صورت $\varepsilon_{r,i}$ معادل صفر خواهد بود. سپس سیاست کلی سیستم توسط بردار(ρ_1 ,ρ_2 , … ,ρ_R) = ρ نشان داده می شود.

تابع کمی

کنترلر، توابع ارزیابی امنیت و زمان پاسخ را بهینه می کند [۱۵]، که عملکرد استحکام امنیت و QOS را توصیف می کند. به دنبال رسیدن به نتایج بهینه هستیم به طوری که زمان پاسخ تا جایی که ممکن است سریع و امنیت تا جایی که ممکن است مستحکم باشد. بدیهی است که دو هدف فوق با یکدیگر تناقض دارند، چرا که سطح بالاتر امنیت، منابع محاسباتی بیشتری را مصرف خواهد کرد و در نتیجه زمان پاسخ طولانی تر خواهد بود. استحکام امنیتی وابستگی بالایی به تعداد IDPS های به کار رفته و نرخ شناسایی آن ها دارد. استحکام امنیتی زمانی که ترکیب های بیشتری از IDPS ها برای تأمین سرویس های امنیتی مورد استفاده قرار می گیرند، افزایش می یابد. تابع امنیت باید شرایطی که در آن استحکام امنیتی کل حداقل بزرگتر از مقدار ماکزیمم نرخ شناسایی به کار رفته در سیاست باشد را برآورده کند و بتواند زمانی که مکانیسم های بیشتر اتخاذ می شوند، افزایش یابد. به همین دلیل از میانگین نمایی برای محاسبه استحکام امنیتی برای رسیدن به یک مدل با ثبات تر استفاده شده است. بنابراین استحکام امنیتی برای نقش r را می توان از رابطه زیر محاسبه کرد:

$$U_r^s(\rho_r) = \sum_{j=1}^{A} a_{r,j}(\ln \sum_{i=1}^{N} e^{d_{i,j}} \times 10 \times \varepsilon_{r,i})/10 \quad (۱)$$

ابزار امنیتی کل می تواند به صورت مجموع وزنی تمام نقش ها به صورت زیر نشان داده شود:

$$U_{total}^s\left(\underset{\rho}{\rightarrow}\right) = \sum_{\forall r} \omega_r^s U_r^s(\rho_r) \quad (۲)$$

که در آن ω_r^s ، فاکتور وزن نقش r است.

زمان پاسخ میانگین سیستم IDPS ها برای نقش r شامل زمان پاسخ IDPS ها (T_{idps}) و نرم افزارهای شبکه (T_{nas}) می باشد که از رابطه زیر حاصل می شود:

ISSN: 2745-3673
http://www.Rcsj.ir

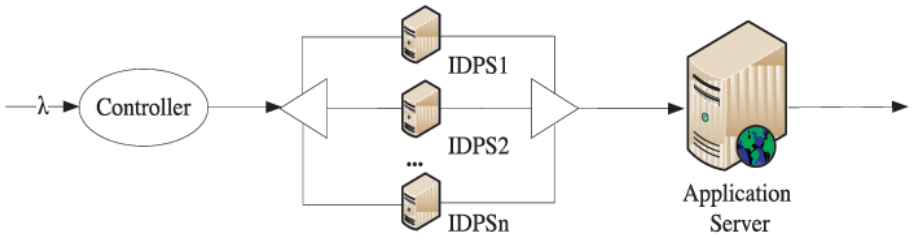
$$T_r = T_{idps} + T_{nas} \quad (۳)$$

زمان پاسخ کل، مجموع وزنی زمان های پاسخ برای تمام نقش ها می باشد و به صورت زیر بیان می شود:

$$T_{total} = \sum_{\forall r} \omega_r^t T_r \quad (۴)$$

که در آن ω_r^t ، فاکتور وزن نقش r است.

مدل شبکه های صف (QN) [۲۰] در این جا برای محاسبه مقدار QOS برای سیاست استفاده می شود. درخواست های کاربران توسط کنترلر ارزیابی می شود و سپس به IDPS ها و سرویس های نرم افزاری شبکه ارسال می شود. مکانیسم های IDPS می توانند به صورت همزمان عمل کنند. با این وجود، سرویس های نرم افزاری شبکه قبل از اینکه بتواند کار کند باید تا تمام شدن IDPS ها صبر کند. صف های fork و join در مدل باز شبکه های صف به عنوان مدل مکانیسم های امنیتی به کار رفته اند. شکل (۲) یک سیستم نرم افزاری شبکه را با مدل fork QN و join نشان می دهد.. IDPS ها در قالب زیر شبکه های fork و join مدل شده اند، و مدل سرور نرم افزار در کنار مدل عملکردی به کار رفته است.[۲۰] درخواست های کاربران تکرار شده و به طور موازی در صف های مختلف سرویس داده می شوند، که در آن زمان های سرویس مختلف برای صف های مختلف مورد نیاز است. اجرای درخواست از نقش شامل ارزیابی کنترلر و اجرای IDPS ها می باشد که توسط سیاست و سرویس های نرم افزاری شبکه پیش برده می شود.



شکل ۲- مدل QN صف های fork و join

فرض کنید که سیستم نرم افزاری شبکه در شکل (۲) از N دستگاه تشکیل شده باشد. طبق قاعده درخواست سرویس [۲۰]، استفاده از دستگاه j توسط نقش r به صورت زیر می باشد:

$$U_{j,r} = \gamma_r \times o_{j,r} \quad (۵)$$

که در آن γ_r نرخ ورود نقش r به دستگاه j است. استفاده کلی از دستگاه j از رابطه زیر به دست می آید:

$$U_j = \sum_{r=1}^{R} U_{j,r} \quad (۶)$$

که در آن R تعداد دستگاه ها است. از آن جایی که مقدار بیشینه استفاده کل صد درصد می باشد، مقدار U_j نباید بزرگتر از یک باشد. با استناد به تئوری ورود[۸] و قاعده کوچک[۹][۲۰]. زمان پاسخ میانگین سیستم نرم افزاری شبکه برای درخواست نقش r را می توان از رابطه زیر به دست آورد:

$$T_{nas} = \sum_{j=1}^{N} \frac{o_{j,r}}{1-U_j} \quad (۷)$$

در اینجا از یک روش برآورد برای محاسبه زمان پاسخ میانگین برای صف های fork و join با سرورهای موازی ناهمگون در شبکه های صف باز استفاده شده است. قبل از برآورد، مکانیسم های امنیتی طبق قاعده زیر مرتب شده اند:

$$o_{1,r} \times \varepsilon_{1,r} \geq o_{2,r} \times \varepsilon_{2,r} \geq \cdots \geq o_{i,r} \times \varepsilon_{i,r}$$

[8] Arrival theorem
[9] Little's Law

ISSN: 2745-3673
http://www.Rcsj.ir

که در آن $o_{i,r}$ تقاضای سرویس درخواست های نقش r در مکانیسم i می باشد، و زمانی که مکانیسم امنیتی i برای نقش r به کار نمی رود، $\varepsilon_{i,r}$ معادل صفر در نظر گرفته می شود. در غیر این صورت، $\varepsilon_{i,r}$ برابر یک خواهد بود. آن گاه زمان پاسخ میانگین برای نقش r با N سرور موازی ناهمگون از رابطه زیر محاسبه می شود:

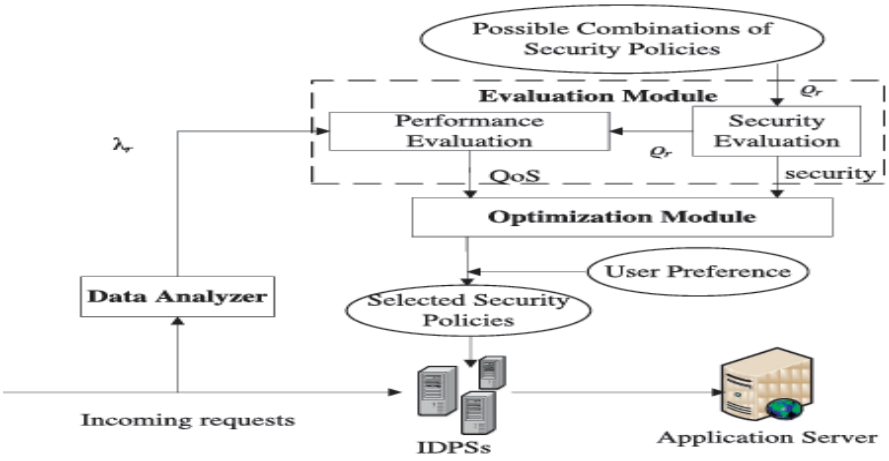
$$T_{idps} = \sum_{i=1}^{N} \frac{1}{i} \times \frac{o_{i,r}}{1-U_i} \quad (۸)$$

که در آن U_i، استفاده از مکانیسم i از سوی درخواست های تمام نقش ها می باشد. استفاده کلی از مکانیسم ها به صورت زیر محاسبه می شود:

$$U_i = \sum_{r=1}^{R} \gamma_r \times o_{i,r} \times \varepsilon_{r,i} \quad (۹)$$

معماری کنترلر

کنترلر، تمام ترکیب های سیاست های امنیتی را جستجو می کند تا به جواب های بهینه ای برسد که به طور همزمان الزامات امنیتی و QoS کاربران را برآورده می کنند. شکل (۳) معماری سیستم کنترلر را نشان می دهد. کنترلر از سه بخش اصلی تشکیل شده است: تحلیل گر داده، ماژول ارزیابی و ماژول بهینه سازی. کنترلر اتونومیک در فاصله زمانی منظم اجرا می شود که فاصله زمانی کنترل برای بهینه کردن نتایج توابع امنیتی و زمان پاسخ نامیده می شود.



شکل ۳- معماری کنترلر

تحلیل گر داده، حجم بار کاری (نرخ درخواست های تمام نقش ها) γ_r سیستم کنونی را در فاصله کنترل به دست می آورد. نرخ درخواست γ_r نقش r و سیاست امنیتی ρ از ترکیب های ممکن سیاست های امنیتی به عنوان پارامترهای ورودی ماژول ارزیابی در نظر گرفته می شوند. ماژول ارزیابی عملکرد از مدل QN برای ارزیابی زمان پاسخ با توجه به سیاست کنونی استفاده می کند. خروجی ماژول ارزیابی عملکرد (زمان پاسخ) و ماژول ارزیابی امنیت (استحکام امنیت) به عنوان پارامترهای ورودی برای ماژول بهینه سازی استفاده می شوند.

به دلیل ترکیب های متعدد موجود از مکانیسم های امنیتی، یافتن تمام ترکیب های بهینه با جستجوی جامع عملی نیست. بنابراین، یک الگوریتم تکاملی تحت عنوان رقابت استعماری بصورت چند هدفه با استفاده از تئوری بازیها ارائه میشود. در این روش در واقع یک مسئله بهینه سازی چند هدفه شامل p هدف, میتواند معادل بازی ای با p بازیکن فرض شود که هر بازیکن معیار بهینه سازی خودش را داشته باشد. هنگامی که یک بازیکن معیار خود را بهینه میکند از اقدامات (استراتژی) دیگر بازیکنان آگاهی دارد و همچنین می داند که اقدامات (استراتژی) دیگر بازیکنان در طول فرآیند تصمیم گیری خود ثابت است. این بهبود برای همه بازیکنان انجام میشود تا جایی که هیچ بازیکنی نتواند بهبود بیشتر از معیار خود داشته باشد. این وضعیت

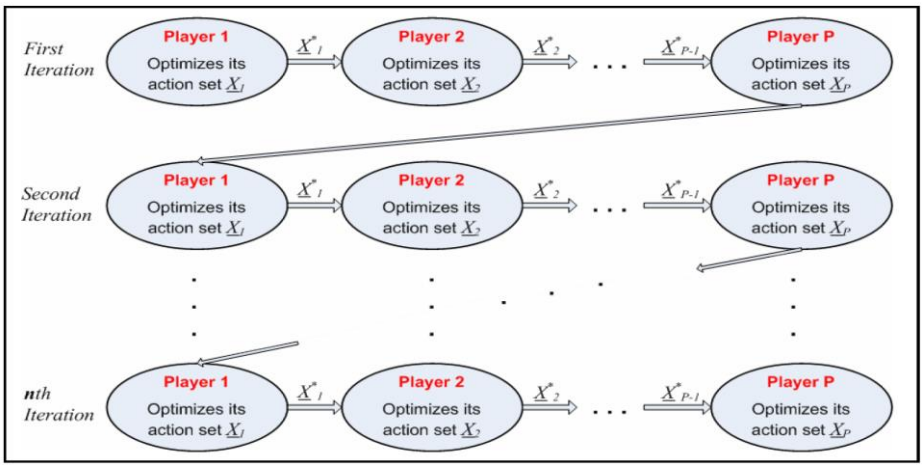
ISSN: 2745-3673
http://www.Rcsj.ir

حاکی از آن است که این سیستم به نقطه تعادل رسیده است و این نقطه تعادل نش بازی و یا نقطه بهینه در بهینه سازی چندهدفه است.[۱]

بهینه سازی چند هدفه مبتنی بر نظریه بازی ها

یک بازی با P بازیکن بدین ترتیب که هر بازی کن با یک مجموعه استراتژی در نظر گرفته میشود و هر یک از X_i ها , راه حل بلقوه بازی $\underline{X}=\{\underline{X}_1, \underline{X}_2, \dots , \underline{X}_p\}$ با مجموعه استراتژی از بازیکنان P_1 تا P_n تعریف میشود. هر بازیکن میتواند معیار خود را بهینه کند و فقط تأثیر بر مجموعه استراتژی(عمل) خودش داشته باشد اما هر عملی میتواند تأثیری بر تابع هزینه دیگر بازیکنان داشته باشد.

در هر تکرار، فرآیند بهینه سازی سود انجام میگیرد. بازیکن i ام تابع هزینه$(J_i(\underline{X}))$ مربوط به معیار بهینه سازی خود را بهینه میکند در حالی که میداند بازیکنان دیگر, انتخابی را انجام داده اند و مجموعه عمل$(\underline{X}_j,\ j=1,\dots,P\ ,\ j\neq i)$ ثابت است. برای شروع هر بازیکن به طور تصادفی استراتژی خودش را انتخاب میکند. سپس هر بازیکن استراتژی خودش را با توجه به استراتژی دیگر بازیکنان بهینه میکند و بعد از این که تمام بازیکنان این کار را انجام دادند تکرار اول تمام میشود. بازیکنان در ادامه از حل بلقوه تکرار n ام استفاده میکنند یعنی: $\underline{X}_n^* = \{\underline{X}^*_{1n}, \underline{X}^*_{2n}, \dots , \underline{X}^*_{pn}\}$ به عنوان استراتژی اولیه برای تکرار بعدی است. وقتی هیچ بازیکنی قادر به تغییر استراتژی نباشد و همچنین تابع هزینه تغییر نکند, ب ازی متوقف میشود و آن نقطه توقف , نقطه تعادل نش بازی است. این مراحل در شکل (۴) نشان داده شده است. [۲]

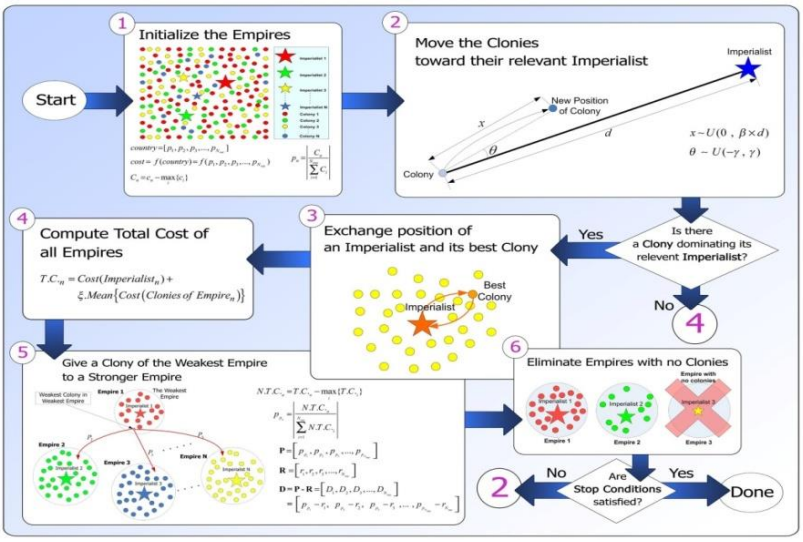


شکل ۴- دیاگرام استرتژی ها و پیدا کردن تعادل نش بازی یا بهینه در بهینه سازی چندهدفه

الگوریتم رقابت استعماری با رویکرد نظریه بازی ها در بهینه سازی چندهدفه

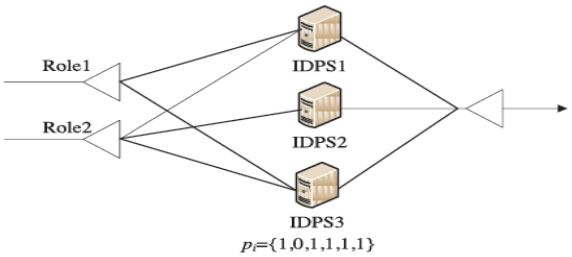
الگوریتم رقابت استعماری (ICA) نیز همانند سایر روش‌های بهینه‌سازی تکاملی، با تعدادی جمعیت اولیه شروع می‌شود. در این الگوریتم، هر عنصر جمعیت، یک کشور نامیده می‌شود. کشورها به دو دسته مستعمره و استعمارگر تقسیم می‌شوند. هر استعمارگر، بسته به قدرت خود، تعدادی از کشورهای مستعمره را به سلطه خود درآورده و آن‌ها را کنترل می‌کند. سیاست جذب و رقابت استعماری، هسته اصلی این الگوریتم را تشکیل می‌دهند. رقابت استعماری، بخش مهم دیگری از این الگوریتم را تشکیل می‌دهد. در طی رقابت استعماری، امپراطوری‌های ضعیف، به تدریج قدرت خود را از دست داده و به مرور زمان با تضعیف شدن از بین می‌روند. رقابت استعماری باعث می‌شود که به مرور زمان، به حالتی برسیم که در آن تنها یک امپراطوری در دنیا وجود دارد که آن را اداره می‌کند. این حالت زمانی است که الگوریتم رقابت استعماری با رسیدن به نقطه بهینه تابع هدف، متوقف می‌شود. شکل (۵) [۳]

ISSN: 2745-3673
http://www.Rcsj.ir



شکل۵- شمای کلی الگوریتم رقابت استعماری[۳]

برای یافتن جواب های پارتو بهینه برای امنیت و QOS، کنترلر الگوریتم رقابت استعماری با رویکرد نظریه بازی ها در بهینه سازی چند هدفه را در شروع هر فاصله اجرا می کند.پارامترهای سیستمی الگوریتم، مانند اندازه جمعیت، میزان جذب و انقلاب همگی توسط سرپرست سیستم از پیش تعریف شده اند و بعد از اجرای آن، یک مجموعه جواب نزدیک به پارتو بهینه با N جواب حاصل شده است که همه جواب های آن با در نظر گرفتن امنیت و QOS، بهینه هستند. کنترلر می تواند یکی از جواب ها را از مجموعه نزدیک به پارتو بهینه بسته به تنظیمات امنیت و QOS کاربر، انتخاب کند.

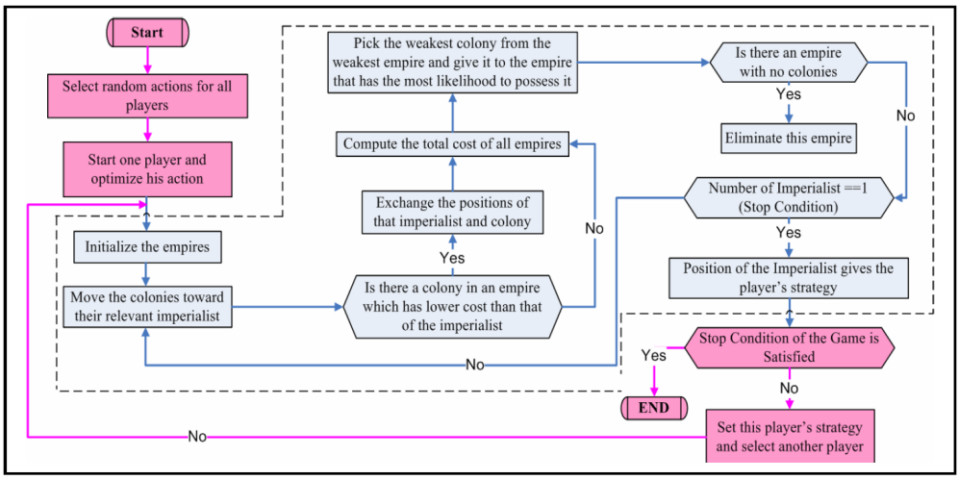


شکل ۶ - نمایش فردی

برای اینکه از عهده مسئله بهینه سازی ناشی از مدل ارزیابی برآییم، (جواب) منفرد استفاده شده در الگوریتم رقابت استعماری با رویکرد نظریه بازی ها در بهینه سازی چندهدفه با کدهای دودویی نشان داده شده است، که نشان می دهد کدام یک از مکانیسم های امنیتی در IDPS ها اتخاذ شده اند. می توان آن را با $\{ x_{r,m} , \dots , x_{r,2} , x_{r,1} , \dots , x_{1,m} , \dots , , \}$ $p_i = \{x_{1,1} x_{1,2}$ نمایش داد، که در آن r تعداد کل نقش ها و m تعداد کل مکانیسم های امنیتی است. هریک از ابعاد جواب منفرد $x_{i,j}$ یک عدد صحیح است که معادل با صفر یا ۱ در نظر گرفته می شود؛ که $x_{i,j} = 1$ به معنی آن ایت که نقش i از مکانیسم امنیتی j استفاده می کند در حالیکه $x_{i,j} = 0$ نشان می دهد که نقش i مکانیسم امنیتی j را اجرا نمی کند. شکل (۴)مثالی را از نمایش یک جواب منفرد ارائه می دهد، که در آن دو نقش توسط سه IDPS شناسایی شده اند. $\{1, 0, 1, 1, 1, 1\} =$ p_i 1} نشان می دهد که نقش اول از اولین و سومین IDPS، و نقش دوم از هر سه IDPS استفاده می کند. مجموعه ای از جواب های منفرد، جمعیت را شکل می دهند که با $P = \{p_1 , p_2 , \dots , p_N \}$ نمایش داده می شود (که در آن N تعداد جواب های منفرد است).

ISSN: 2745-3673
http://www.Rcsj.ir

در واقع در روش پیشنهادی به تعداد r (کل نقش ها) بازیکن وجود دارد که هر کدام میتوانند به تعداد m (تعداد کل مکانیسم های امنیتی) استراتژی برای بازی داشته باشند ، طبق فلوچارت روش پیشنهادی در هر دور یک بازیکن خود را به بهینه ترین حالت سود رسانده و بعد از همگرایی کامل، بازیکن بعدی خود را بهینه میکند و این روند تا جایی انجام میشود که دیگر بازیکنی نتواند سود خود را بهینه تر کند و این همان نقطه همگرایی کل بازیکنان یا همان نقطه تعادل نش و یا همان بهینه چند هدفه است.



شکل ۷- فلوچارت الگوریتم پیشنهادی برای بهینه سازی چندهدفه با رویکرد نظریه بازی ها[۲]

نتایج تجربی

تنظیمات تجربی

در این بخش، آزمایشات جهت بررسی اعتبار سیستم کنترلر اجرا می شوند. الگوریتم رقابت استعماری با رویکرد نظریه بازی ها در بهینه سازی چند هدفه و الگوریتم تپه نوردی حریصانه با استفاده از زبان های برنامه نویسی C++ در پلت فرم VC++ 6.0 اجرا می شوند.

تمام کدهای منبع روی یک CPU با مشخصات Inter® Core™ i5-3230M ، 2.6 GHz، حافظه 4 GB با سیستم عامل ویندوز ۷ اجرا می شوند. در آزمایشات ما، روش بهینه را با دو مسئله در دو اندازه مختلف اجرا می کنیم: یکی با ۳ نقش های درخواست متفاوت و ۸ IDPS مختلف، و دیگری با ۳ نقش های درخواست متفاوت و ۱۰ IDPS مختلف. نرخ های شناسایی IDPS و احتمال حمله برای نقش ها را می توان با استفاده از داده مناسب و اطلاعات تاریخی برآورد نمود.

برای ارزیابی عملکرد الگوریتم ها از فاصله نسلی وارونه (IGD) استفاده شده است:

$$IGD(S,P) = \frac{\sum_{x \in P} dis(x,S)}{|P|} \qquad (۱۰)$$

که در آن $dis(x,S)$ نزدیک ترین فاصله اقلیدسی جواب x در P به جواب های S است، در حالیکه $|P|$ تعداد جواب ها در P را نشان می دهد. با فرض اینکه $m = (m_1, m_2, \dots, m_i)$ و $n = (n_1, n_2, \dots, n_i)$ دو نقطه در فضای i بعدی باشند، فاصله اقلیدسی $dis(m,n)$ بین m و n به صورت زیر محاسبه می شود:

$$dis(m,n) = \sqrt{(m_1 - n_1)^2} + \sqrt{(m_2 - n_2)^2} + \cdots + \sqrt{(m_i - n_i)^2} \qquad (۱۱)$$

در این مقاله، زمان پاسخ و امنیت یک جواب با یک نقطه دو بعدی نمایش داده شده است.

به طور کلی؛ مقدار کمتر متریک IGD نشان میدهد که مجموعه حاصل S به مجموعه صحیح پارتو بهینه P نزدیک تر است و به طور یکنواخت تری در جبهه صحیح پارتو بهینه توزیع شده است.

ISSN: 2745-3673
http://www.Rcsj.ir

مزیت های مدل چند هدفه

الگوریتم چند هدفه پیشنهادی (MOICAGA) در ۳ نرخ درخواست مختلف اجرا می شود. این سه نرخ درخواست مختلف γ_1 = {6,8,11}، γ_2 ={12,15,13} و γ_3 ={6,5,4} به ترتیب بیان کننده بارهای کاری کل، بالا و پایین هستند. سه مقدار در مجموعه های نرخ درخواست، نرخ های درخواست سه نقش مختلف را نشان می دهند.

در کار [۱۵]، امنیت شبکه و QOS با یک ابزار جهانی سیستم با استفاده از یک روش وزن دهی خطی جمع شده اند و سپس یک الگوریتم تپه نوردی حریصانه برای جستجوی جواب بهینه با ابزار جهانی بیشینه به کار گرفته شده است. با این وجود، این روش نمی تواند یک مجموعه از جواب های پارتو بهینه نتیجه بدهد و انتخاب بردارهای وزنی برای تعیین اهمیت نسبی امنیت و QOS دشوار است. مدل چند هدفه پیشنهادی ما قادر است به خوبی مسائل بالا را حل نموده و کاربران میتوانند تنظیمات خود را هم روی امنیت و هم QOS تعیین کنند. برای به تصویر کشیدن مزیت مدل چند هدفه خود، روش پیشنهادی در کار [۱۵] با ۵۰.۰۰۰ تکرار در سه مجموعه نرخ درخواست مختلف γ_1، γ_2 و γ_3 اجرا شده است. جدول های ۱ و ۲، جواب های بهینه حاصل از مدل تک هدفه با استفاده از مقادیر وزنی مختلف با نرخ های درخواست γ_1 و γ_3 در مسئله های با اندازه مختلف را ارائه می دهد، که بردارهای وزنی مختلف در آن برای تعیین اهمیت امنیت و QOS به کار رفته اند، مانند ω_1 {0.2,0.8}، ω_2 = {0.3,0.7}، ω_3 = {0.4,0.6}، ω_4 = {0.5,0.5}، ω_5 = {0.6,0.4}، ω_6 = {0.7,0.3} و ω_7 = {0.8,0.2}. همان طور که در جداول ۱ و ۲ مشاهده می شود، مدل تک هدفه تنها می تواند یک جواب بهینه تحت بردارهای وزنی مختلف بدهد. به طور کلی، با افزایش اهمیت امنیت در بردار وزن، مقدار بهینه استحکام امنیت نیز بالا می رود. نقص دیگر این است که مدل تک هدفه ممکن است نتیجه بهینه مشابه بدهد حتی در صورت استفاده از بردارهای وزنی مختلف، مانند (ω_3 , ω_4) و (ω_7 , ω_6 , ω_5) در جدول ۳ و (ω_2 , ω_1) و (ω_4 , ω_3) در جدول ۲. علاوه بر این، از آنجایی که یافتن جوابی که استفاده از $IDPS_i$ U_i را در بار کاری زیاد بزرگتر از یک می کند، راحت است، مدل تک هدفه نمی تواند مسئله را در نرخ های درخواست مختلف γ_3 حل کند. با این وجود، این امر در استفاده کاربردی عملی نیست.

جدول ۱- نتایج با مجموعه مقادیر وزنی مختلف برای ۳ نقش و ۸ IDPS

	ϖ_1		ϖ_2		ϖ_3		ϖ_4	
	security	delay	security	delay	security	delay	security	delay
λ_1	0.739041	0.213123	0.797296	0.233339	0.848868	0.266636	0.848868	0.266636
λ_3	0.819526	0.286701	0.860096	0.312148	0.860096	0.312148	0.860096	0.312148
	ϖ_5		ϖ_6		ϖ_7			
	security	delay	security	delay	security	delay		
λ_1	0.857626	0.271037	0.857626	0.271037	0.857626	0.271037		
λ_3	0.860096	0.312148	0.873403	0.31732	0.873403	0.31732		

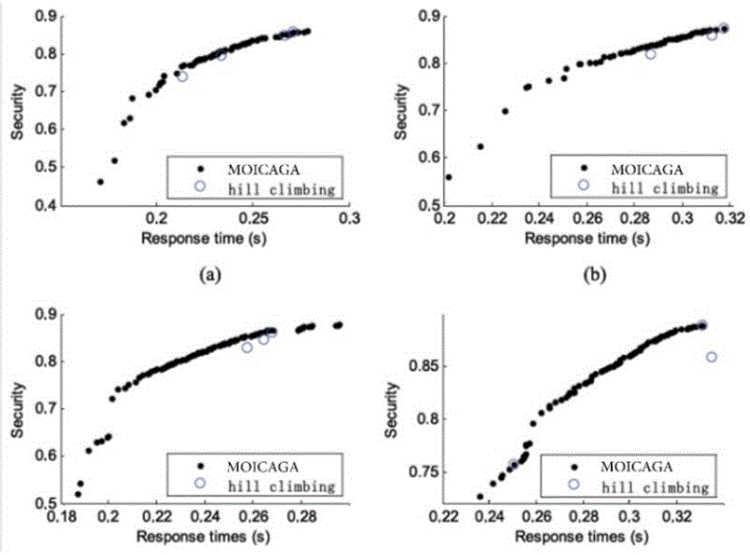
جدول ۲- نتایج با مجموعه مقادیر وزنی مختلف برای ۳ نقش و ۱۰ IDPS

	ϖ_1		ϖ_2		ϖ_3		ϖ_4	
	security	delay	security	delay	security	delay	security	delay
λ_1	0.829517	0.257505	0.829517	0.257505	0.847137	0.264368	0.847137	0.264368
λ_3	0.757392	0.250306	0.757392	0.250306	0.757392	0.250306	0.757392	0.250306
	ϖ_5		ϖ_6		ϖ_7			
	security	delay	security	delay	security	delay		
λ_1	0.862688	0.26812	0.862688	0.26812	0.862688	0.26812		
λ_3	0.859674	0.334973	0.889605	0.33126	0.889605	0.33126		

شکل (۸) (a) و (b) نتایج مقایسه بین مدل چند هدفه ما و مدل تک هدفه [۱۵] که از الگوریتم تپه نوردی با ۸ IDPS و مجموعه نرخ های درخواست γ_1 و γ_3 استفاده می کند را نمایش میدهد. همان طور که در شکل (۸) (c) و (d) مشاهده می شود، الگوریتم تپه نوردی تنها جواب های بهینه متعددی را پیدا می کند، در حالیکه الگوریتم چند هدفه پیشنهادی

ISSN: 2745-3673
http://www.Rcsj.ir

(MOICAGA) قادر است جواب های بهینه موجود بسیار بیشتری را جستجو کند. نتایج حاصل از مدل تک هدفه در مجموعه مقادیر وزنی مختلف، مشابه آنهایی هستند که در مدل ما یافت شدند و یا مغلوب آن ها هستند. شکل (۸) (c) و (d) نتایج مقایسه با ۱۰ IDPS و مجموعه نرخ های درخواست γ_1 و γ_3 را به تصویر می کشد. این نتایج، مشابه نتایج موجود در شکل (۸) (a) و (b) هستند. این نتایج تجربی، مدل چند هدفه ما را که نسبت به مدل تک هدفه اثر بخش تر است، اعتبار می بخشد.



شکل ۸- نتایج مقایسه با ۸ IDPS در a و b و ۱۰ IDPS در c و d

جدول (۳) زمان های اجرا برای الگوریتم های مقایسه شده تحت سه مجموعه نرخ درخواست مختلف γ_1، γ_2 و γ_3 را نشان می دهد که مقادیر میانگین ۳۰ زمان اجرا هستند. برای اجرای الگوریتم چند هدفه پیشنهادی (MOICAGA) ۱.۴ ثانیه زمان لازم است، در حالی که جستجوی تپه نوردی سریعتر به نظر می رسد. با این وجود، الگوریتم تپه نوردی تنها قادر است یک جواب بهینه فرعی بیابد؛ در حالی که الگوریتم MOICAGA می توانند بیش از ۶۰ ترکیب از جواب های بهینه فرعی نتیجه دهند، که نیازهای مختلف کاربران در رابطه با امنیت و QOS را پاسخ میدهند. به طور کلی، بار کاری سیستم یا نقش ها، تحت درخواست های کاربر در دوره زمانی کوتاه مدت، تغییر چندانی نمی کند؛ به طوری که روش بهینه سازی پیشنهادی فقط به صورت دوره ای در یک فاصله زمانی منظم اجرا می شود. زمان های اجرای الگوریتم MOICAGA ، اثر منفی بر سیستم ندارند.

جدول ۳- زمان های اجرا برای الگوریتم های مختلف

	λ_1	λ_2	λ_3
MOICAGA	1.42	1.312	1.504
Hill climbing	0.073	0.065	0.075

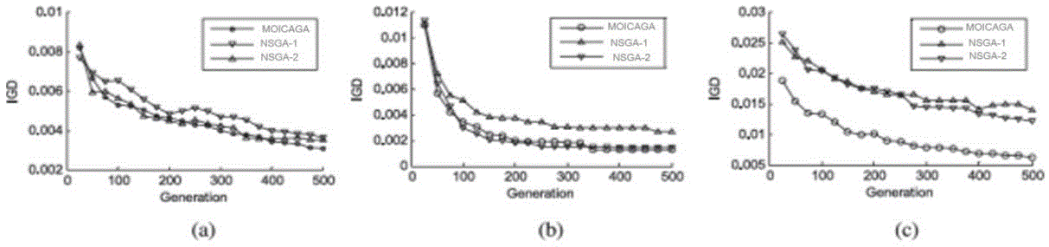
مقایسه الگوریتم MOICAGA با سایر روشها

شکل (۹)، منحنی های تکاملی IGD با ۸ IDPS و سه مجموعه نرخ درخواست مختلف γ_1، γ_2 و γ_3 با استفاده از سه الگوریتم را نشان می دهد ، که در آن MOICAGA ، NSGA-1 و NSGA-2 به ترتیب بیان گر MOICAGA با عملگر همگذری مبتنی بر نقش، NSGA-II اصلی با عملگر همگذری تک نقطه ای و عملگر همگذری دو نقطه ای هستند. در همه بارهای کاری اعمال شده، به وضوح مشاهده می شود که نتایج IGD الگوریتم MOICAGA سریع تر کاهش می یابد و روند

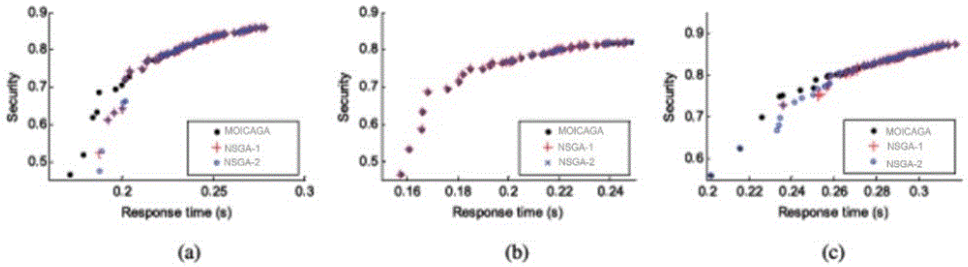
ISSN: 2745-3673
http://www.Rcsj.ir

تکامل روان تر است، که نشان میدهد MOICAGA ما، سرعت همگرایی بیشتری نسبت به نسخه اصلی آن دارد. به ویژه در بارهای کاری کم یا کلی، مزیت روش ما مشهودتر است.

شکل (۱۰) مجموعه های پارتو بهینه حاصل از سه الگوریتم مقایسه شده با ۸ IDPS و سه مجموعه نرخ درخواست مختلف γ_1، γ_2 و γ_3 را نشان می دهد.



شکل ۹- منحنی های تکاملی با ۸ IDPS

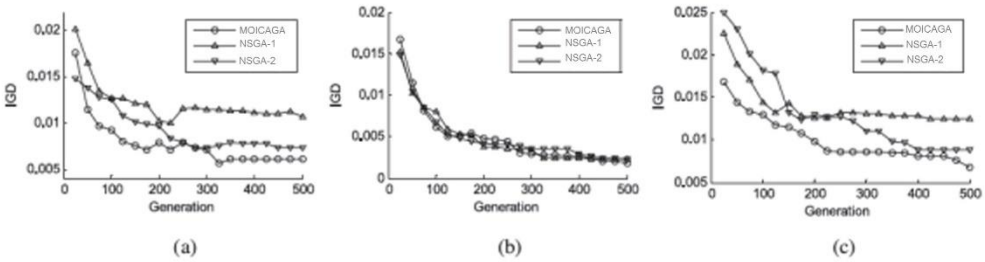


شکل ۱۰- نتایج نهایی با ۸ IDPS

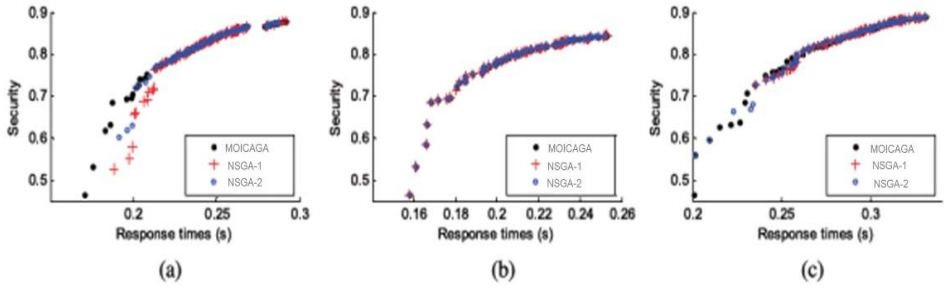
بر اساس مشاهدات منحنی های a تا c شکل (۱۰)، جواب های حاصل از MOICAGA تقریباً بر جواب های به دست آمده از NSGA-2 با دو عملگر همگذری مرسوم غالب هستند. در بار کاری کلی یا پایین، روش ما می تواند جواب های بهتری در سناریویی با امنیت پایین پیدا کند. در حالی که در بار کاری بالا، روش ما عملکردی مشابه با NSGA-1 و NSGA-2 نشان می دهد. این امر منطقی است که پیدا کردن جواب های پارتو بهینه در بار کاری بالا راحت است چرا که تعداد ترکیب های موجود از IDPS ها به مقدار زیادی کاهش می یابد. در نتیجه، NSGA-1 و NSGA-2 بسیار مشابه با روش ما در بار کاری بالا عمل می کنند. علاوه بر این، با افزایش درخواست های کل، نتایج IGD بزرگ تر خواهد شد، که به این معنی است که یافتن تمام جواب های پارتو بهینه به مراتب سخت تر شده و زمان پاسخ کل نیز طولانی تر خواهد شد.

با توجه به آزمایشات با ۱۰ IDPS، شکل (۱۱)، منحنی های تکاملی IGD با ۱۰ IDPS و سه مجموعه نرخ درخواست مختلف γ_1، γ_2 و γ_3 با استفاده از سه الگوریتم را نشان می دهد. مشابه با نتایج IGD با ۸ IDPS، الگوریتم MOICAGA به طور قابل توجهی بهتر عمل می کند، به ویژه در بارهای کاری پایین. شکل ۱۲، هم چنین مجموعه های پارتو بهینه حاصل از MOICAGA ، NSGA-1 و NSGA-2 با ۱۰ IDPS و سه مجموعه نرخ درخواست مختلف γ_1، γ_2 و γ_3 را نشان می دهد. در بار کاری کلی و پایین نیز روش ما میتواند جواب های بهتری در سناریویی با امنیت پایین نتیجه دهد. در حالی که در بار کاری بالا، روش ما عملکرد مشابه با NSGA-1 و NSGA-2 دارد.

ISSN: 2745-3673
http://www.Rcsj.ir



شکل ۱۱- منحنی های تکاملی با ۱۰ IDPS



شکل ۱۲- نتایج نهایی با ۱۰ IDPS

آزمون ویلکاکسون[10] را برای مجموعه های IGD حاصل از MOICAGA ، NSGA-1 و NSGA-2 در دو اندازه IDPS و سه مجموعه نرخ درخواست مختلف γ_1، γ_2 و γ_3 اجرا می کنیم. اندازه هر مجموعه IGD، ۳۰ است. به طور کلی، وقتی p-مقدار آزمون ویلکاکسون برای دو مجموعه داده کمتر از ۰.۰۵ باشد، دو مجموعه داده به لحاظ آماری مشابه نیستند. تمامی p-مقدارها در جدول ۶ گردآوری شده اند. به جز ۱۰ IDPS و مجموعه نرخ های درخواست γ_2، سایر p-مقدارهای آزمون ویلکاکسون بسیار کمتر از ۰.۰۵ هستند. این امر را می توان در شکل (۱۲) (b) هم مشاهده کرد که نتایج IGD برای تمام سه الگوریتم در ۱۰ IDPS و مجموعه نرخ های درخواست γ_2 بسیار مشابهند. از آنجایی که یافتن جواب های پارتو بهینه در بار کاری بالا راحت است، این مشاهده منطقی است.

جدول ۴ -مقدارهای آزمون ویلکاکسون برای NSGA-1، NSGA-2 در مقابل MOICAGA

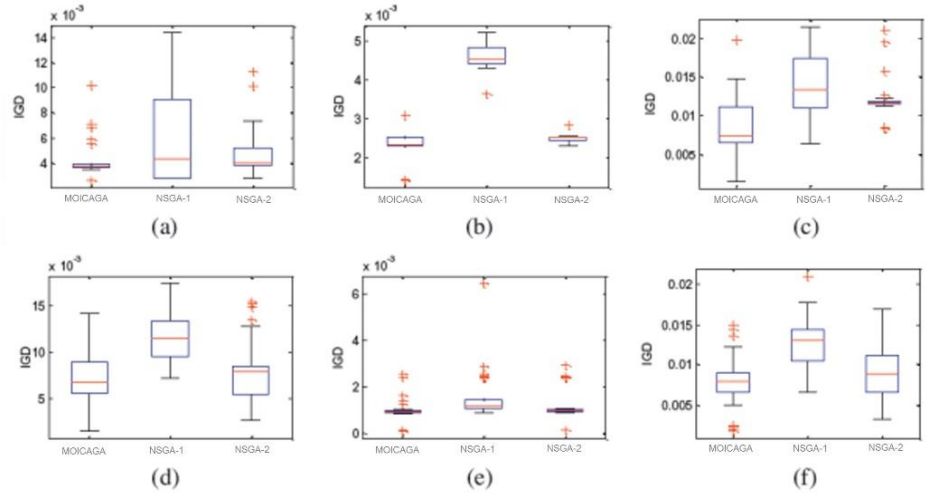
	NSGA-1 **VS** MOICAGA	NSGA-2 **VS** MOICAGA
8 IDPSs with λ_1	9.94E-03	5.79E-04
8 IDPSs with λ_2	2.78E-11	8.40E-03
8 IDPSs with λ_3	4.42E-06	3.26E-07
10 IDPSs with λ_1	3.50E-06	8.40E-03
10 IDPSs with λ_2	8.82E-01	7.35E-01
10 IDPSs with λ_3	1.44E-04	2.80E-03

شکل (۱۳) (a-f) نمودارهای جعبه ای مجموعه های IGD حاصل از سه الگوریتم مشاهده شده در سه مجموعه نرخ درخواست مختلف و دو اندازه IDPS را به تصویر می کشد. اندازه هر مجموعه IGD، ۳۰ است. همانطور که می بینید، در همه

[10] Wilcoxon rank sum test

ISSN: 2745-3673
http://www.Rcsj.ir

بارهای کاری و اندازه های IDPS، مقادیر میانه MOICAGA در نمودار جعبه ای کمتر از NSGA-1 و NSGA-2 می باشد. به ویژه در بارهای کاری کم، مزیت روش ما مشهودتر است.



شکل ۱۳- نمودارهای جعبه ای مجموعه های IGD از سه الگوریتم مقایسه شده با ۸ و ۱۰ IDPS

نتیجه گیری

در این مقاله، یک مدل بهینه سازی چند هدفه اتونومیک پیشنهاد شده است، که می تواند امنیت و QOS را تحت منبع محاسباتی موجود ادغام کند. به منظور بهینه سازی مدل چند هدفه پیشنهادی، یک الگوریتم چندهدفه پیشنهادی (MOICAGA) برای رسیدن به جواب های پارتو بهینه اصلاح شده است،آنجایی که الزامات امنیتی برای نقش های مختلف مستقل هستند، روش همگذری مبتنی بر نقش پیشنهادی هم چنین برای هر نقش به طور مستقل تبادل اطلاعات انجام می دهد. نتایج شبیه سازی نشان می دهد که MOICAGA بهتر از نسخه های NSGA اصلی عمل می کند، و مجموعه بهینه حاصل قادر است به کل جبهه پارتو بهینه تحت بارهای کاری مختلف نزدیک شود. این سیاست های امنیتی پارتو بهینه حاصل نه تنها می توانند الزامات مختلف امنیتی کاربر را برآورده کنند، بلکه QOS بهینه را نیز فراهم می آورند.

در پژوهش های آینده، اهداف دیگری در مدل بهینه سازی چند هدفه خود در نظر خواهیم گرفت. برای مثال، در محیط رایانش ابری، که منابع محاسباتی به سادگی با تقاضا افزایش می یابند، هم چنان نیاز به متوازن کردن امنیت و QOS وجود دارد چرا که کاربران باید برای منابع محاسباتی بیشتر هزینه پرداخت کنند. حسابداری منابع محاسباتی می تواند یک هدف بالقوه دیگر در این سناریوی کاربردی باشد. علاوه بر این، سایر الگوریتم های بهینه سازی برتر که از طبیعت الهام گرفته اند نیز برای بهینه سازی مدل پیشنهادی ما مورد بررسی قرار خواهند گرفت.

ISSN: 2745-3673
http://www.Rcsj.ir

منابع و مراجع

[1] J. Periaux et al, “Multi-Objective EAs and Game Theory”, Evolutionary Optimization and Game Strategies for Advanced Multi-Disciplinary Design, Intelligent Systems, Control and Automation: Science and Engineering,2015.
[2] Ramin Rajabioun, Esmaeil Atashpaz-Gargari, and Caro Lucas,” Colonial Competitive Algorithm as a Tool for Nash Equilibrium Point Achievement”, pringer-Verlag Berlin Heidelberg, O. Gervasi et al. (Eds.): ICCSA 2008, Part II, LNCS 5073, pp. 680–695, 2008.
[3] Atashpaz-Gargari, E., Lucas, C.: Imperialist Competitive Algorithm: An algorithm for optimization inspired by imperialistic competition. In: IEEE Congress on Evolutionary Computation, Singapore ,2007.
[4] Andres, G., Jose, H., Ernesto, R., & Alfredo, M. (2013). Indexing and retrieving in fingerprint databases under structural distortions. Expert Systems with Applications, 40 (8), 2858–2871.
[5] Laura, C., Jorge, H., & Viviana, E. (2006). Real time database systems. In Encyclopedia of database technologies and applications (pp. 524–530). Hershey, Pa.: Idea Group Referene
[6] Al-Sayid, N., & Aldlaeen, D. (2013). Database security threats: A survey study. In 2013 5th International conference on computer science and information technology (CSIT) (pp. 60–64).
[7] system. In Science and information conference (SAI) (pp. 556–563).
[8] Amirijoo, M., Hansson, J., & Son, S. (2006). Specification and management of QoS in real-time databases supporting imprecise computations. IEEE Transactions on Computers, 55 (3), 304–319;
[9] Kamra, A., & Bertino, E. (2009). Survey of machine learning methods for database security. In Machine learning in cyber trust (pp. 53–71). Springer
[10] Bertino, E., & Sandhu, R. (2005). Database security - concepts, approaches, and challenges. IEEE Transactions on Dependable and Secure Computing, 2 (1), 2–19
[11] Jabbour, G., & Menasee, D. (2008). Policy-based enforcement of database security configuration through autonomic capabilities. In International conference on au- tonomic and autonomous systems (pp. 188–197)
[12] Darwish, S., Guirguis, S., & Ghozlan, M. (2013). Intrusion detection in role administrated database: Transaction-based approach. In International conference on com- puter engineering and systems (ICCES) (pp. 73–79).
[13] Taneja, N., Raman, B., & Gupta, I. (2011). Chaos based partial encryption of spiht compressed images. International Journal of Wavelets Multiresolution and Infor- mation Processing, 9 (2), 317–331.
[14] Chen, J., Hu, C., Zeng, H., & Zhang, J. (2009). Impact of security on QoS in communication network. In International conference on networks security, wireless communications and trusted computing: 2 (pp. 40–43)
[15] Alomari, F., & Menasce, D. (2012). An autonomic framework for integrating secu- rity and quality of service support in databases. In 2012 IEEE sixth international conference on software security and reliability (SERE) (pp. 51–60)
[16] Bayon, L., Grau, J., Ruiz, M., & Suarez, P. (2012). The exact solution of the environmental/economic dispatch problem. IEEE Transactions on Power Systems, 27 (2), 723–
[17] Menasce, D., & Kephart, J. (2007). Guest editors’ introduction: Autonomic comput- ing. IEEE Internet Computing, 18–21.
[18] Bennani, M., & Menasce, D. (2005). Resource allocation for autonomic data centers using analytic performance models. In Second international conference on auto- nomic computing (pp. 229–240).
[19] Kleinrock, L. (1975). Queueing systems. New York, USA: Wiley; Menasce, D. (2004). Performance by design computer capacity planning by example. Upper Saddle River, NJ, USA: Prentice Hall PTR.

ISSN: 2745-3673
http://www.Rcsj.ir

[20] Menasce, D. (2004). Performance by design computer capacity planning by example. Upper Saddle River, NJ, USA: Prentice Hall PTR.